LaQual: A Novel Framework for Automated Evaluation of LLM App Quality

YAN WANG, Communication University of China, China
XINYI HOU, Huazhong University of Science and Technology, China
YANJIE ZHAO, Huazhong University of Science and Technology, China
WEIGUO LIN, Communication University of China, China
HAOYU WANG, Huazhong University of Science and Technology, China
JUNJUN SI*, Communication University of China, China

LLM app stores, as platforms for aggregating intelligent applications built on large language models (LLMs), are rapidly emerging, offering users diverse choices for content generation, coding assistance, education, and more. However, current ranking and recommendation mechanisms in LLM app stores predominantly rely on static metrics, such as user interactions and favorites, making it challenging for users to efficiently identify high-quality apps. At the same time, current academic research focuses on specific vertical fields and lacks a general, automated evaluation framework applicable to the diverse LLM app ecosystem. To address the above challenges, we present LaQual, an automated framework for **L**LM **a**pp **qual**ity evaluation. LaQual integrates three key stages: (1) LLM app labeling and hierarchical classification for precise scenario mapping; (2) static indicator evaluation using time-weighted user engagement and functional capability indicators to filter low-quality apps; and (3) dynamic scenario-adaptive evaluation, where LLM generates scenario-specific evaluation metrics, scoring criteria, and tasks for comprehensive quality evaluation. Experiments on a mainstream LLM app store demonstrate the effectiveness of LaQual. Its automated scores show high consistency with human judgments (Spearman's $\rho = 0.62$, $p = 0.006$ in legal consulting; $\rho = 0.60$, $p = 0.009$ in travel planning). Through effective screening, LaQual can reduce the candidate LLM app pool by 66.7% to 81.3%. User studies further validate its significant outperformance over baseline systems in terms of decision confidence, comparison efficiency (mean 5.45 vs. 3.30), and evaluation report value (4.75 vs. 2.25). These results demonstrate that LaQual provides a scalable, objective, and user-centric solution for high-quality discovery and recommendation of LLM apps in real-world scenarios.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; • **Software and its engineering** → *Software verification and validation.*

Additional Key Words and Phrases: LLM app stores, Quality evaluation, Automated evaluation framework

*The corresponding author is Junjun Si (sijunjun@cuc.edu.cn).

Authors' addresses: Yan Wang, wangyan@mails.cuc.edu.cn, Communication University of China, Beijing, China; Xinyi Hou, xinyihou@hust.edu.cn, Huazhong University of Science and Technology, Wuhan, China; Yanjie Zhao, yanjie_zhao@hust.edu.cn, Huazhong University of Science and Technology, Wuhan, China; Weiguo Lin, linwei@cuc.edu.cn, Communication University of China, Beijing, China; Haoyu Wang, haoyuwang@hust.edu.cn, Huazhong University of Science and Technology, Wuhan, China; Junjun Si, sijunjun@cuc.edu.cn, Communication University of China, Beijing, China.

1 INTRODUCTION

Large language models (LLMs) have made breakthrough progress in recent years, which has spurred the emergence of a wide range of LLM-based application forms, such as LLM apps, LLM agents, self-hosted LLM services, and edge-LLM applications. This study focuses on **LLM apps** [57], which specifically refer to intelligent applications built upon LLM capabilities and distributed through various **LLM app stores** such as OpenAI's GPT Store [31], Quora's Poe [34], Baidu's Smart Cloud Qianfan AppBuilder (AppBuilder) [2], and Tencent's Yuanqi Platform [43]. These LLM apps typically feature independent entry points, functional descriptions, and user interfaces, and are widely used in scenarios such as text generation, content creation, education, coding assistance, and emotional support [4, 5, 34, 57]. Compared to traditional apps, LLM apps not only understand and generate natural language but also support complex capabilities such as external knowledge base integration, plugin invocation, and multimodal input, significantly expanding the boundaries of artificial intelligence (AI) applications [16, 47, 51, 52]. Driven by rapid technological advances and lowered development barriers, the LLM app ecosystem has experienced explosive growth, with mainstream LLM app stores hosting a vast number of LLM apps. In just a few months since its launch, the GPT Store has already included over 3 million GPT apps [27], FlowGPT has over 4 million monthly active users [26], and platforms like Poe, and HuggingChat have each exceeded 10 million monthly visits [39, 40]. Furthermore, third-party platforms such as GPTStore.AI [11] and GPTs Hunter [18] have also emerged, with the app inventory on a single platform reaching hundreds of thousands, demonstrating explosive growth in overall market size and activity.

This rapid expansion brings new challenges for users in discovering and selecting high-quality LLM apps. On the one hand, the surge in the number of apps has led to information overload, making it difficult for users to find high-quality apps that truly meet their needs quickly. On the other hand, current mainstream LLM app stores generally adhere to traditional recommendation logic: users input keywords to search, and the platform then returns a list of apps ranked by static metrics such as usage, number of favorites, and release date [7, 24]. As a result, users must manually filter and repeatedly try out various apps, incurring significant cognitive and time costs. In addition, recent user discussions have revealed that app rankings are frequently manipulated through methods such as keyword stuffing, fake reviews, and misleading updates, resulting in low-value or even unused apps appearing at the top of search results [28–30]. This manipulation relies on static metrics, making it difficult for users to accurately identify truly high-quality apps, thus severely damaging the user experience and the health of the ecosystem.

Existing LLM evaluation methods typically focus on specific domains, such as law, mathematics, or code generation, and rely on corresponding benchmarks with predefined metrics [14, 45, 54, 58]. While these benchmarks are effective for evaluating models in well-defined scenarios, they are difficult to scale to the LLM app store context. The diversity of app categories can reach dozens or even more, making a comprehensive evaluation of all types of apps impractical using existing methods [42, 56]. Furthermore, unlike general-purpose or domain-specific LLMs, individual LLM apps often target highly specialized or niche functionalities, which require evaluation at a much finer granularity to accurately reflect their value and effectiveness. Most importantly, the ultimate goal of LLM app stores is to serve users; therefore, factors such as user engagement, experience, and satisfaction must be fully considered in the evaluation process. However, current benchmarks are often model-centric and static, lacking mechanisms to capture real user interactions or feedback. Therefore, a new evaluation framework that can adapt to the multidimensional, fine-grained, and user-centric nature of the LLM app ecosystem is urgently needed.

In response to these unmet needs, we propose LaQual, an automated framework for **L**LM **a**pp **qual**ity evaluation, designed to be broadly applicable across diverse LLM app scenarios. LaQual

first conducts scenario-aware labeling and classification to accurately map LLM apps to their intended use cases. Next, it performs multi-dimensional static filtering based on time-weighted user engagement and functional capability indicators, efficiently excluding low-quality candidates. Finally, LaQual employs LLM-driven dynamic assessment, automatically generating scenario-adaptive evaluation metrics and tasks to enable fine-grained automatic quality measurement.

We systematically evaluate the effectiveness of our framework from three key perspectives: evaluation accuracy and adaptivity, recommendation effectiveness, and user impact. First, we assess whether LaQual can provide accurate and scenario-adaptive quality evaluation by comparing its automated scores with human judgments across different LLM app scenarios, achieving strong consistency (Spearman's $\rho = 0.62$, $p = 0.006$ in legal consulting, and $\rho = 0.60$, $p = 0.009$ in travel planning). Second, we benchmark LaQual against mainstream app store recommendation mechanisms, demonstrating that our framework reduces the candidate app pool by 66.7%–81.3% while surfacing higher-quality apps. Third, through user studies, we examine the impact of LaQual on users' decision quality, cognitive effort, and trust in real-world app discovery tasks. Results show that LaQual significantly improves user decision confidence and comparison efficiency (mean 5.45 vs. 3.30), and increases perceived report value (4.75 vs. 2.25) compared to baseline systems. Collectively, these evaluations validate the practical effectiveness, rationality, and user-centered benefits of our solution in real-world LLM app scenarios.

Contributions. In summary, this work makes the following key contributions:

- We design and implement LaQual, an automated framework that integrates LLM app labeling classification, static indicator filtering, and dynamic LLM-driven evaluation for scalable and systematic LLM app quality evaluation.
- We propose an efficient static indicator evaluation module that utilizes time-weighted user engagement and functional capability indicators, reducing low-quality app candidates by 66.7%–81.3% and significantly improving evaluation efficiency.
- We develop a fully automated dynamic evaluation pipeline, where LLMs generate scenario-specific evaluation metrics and tasks, enabling fine-grained, context-adaptive, and objective quality evaluation across diverse app categories.
- We conduct comprehensive real-world validation, showing that LaQual achieves high agreement with expert ratings, and substantially boosts user comparison efficiency, decision confidence, and perceived report value compared to baseline systems.
- LaQual is easily adaptable to new evaluation scenarios and categories, making it a practical solution for scalable and objective quality evaluation across evolving LLM app ecosystems.

The remainder of this paper is organized as follows. § 2 reviews relevant background on LLM apps and LLM-based evaluation. § 3 introduces the architecture and key components of LaQual. In § 4, we validate the framework's effectiveness through experiments and user studies. § 5 discusses the main findings and their broader significance. § 6 outlines current limitations and future directions. § 7 summarizes related work, and § 8 concludes the paper.

2 BACKGROUND

2.1 LLM App

LLM apps are specialized applications built on LLMs that provide intelligent, task-oriented services through natural language interaction [57]. Unlike traditional software, LLM apps leverage the reasoning, comprehension, and generation abilities of foundation models to support diverse scenarios, such as content creation, code assistance, consultation, translation, and data analysis. LLM apps include metadata like app name, description, instructions, categories, developer details, and sample conversations, helping users understand each app's core features and use cases.
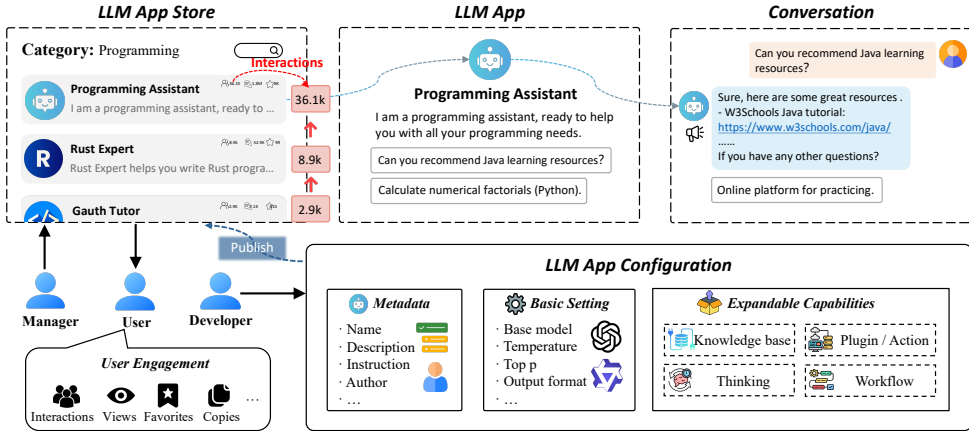
Fig. 1. Overview of the LLM App Ecosystem.

LLM app stores are digital platforms designed for the aggregation, distribution, and management of various LLM apps. Notable examples include GPT Store [31], Coze [5], and Poe [34]. These platforms provide developers with efficient tools for publishing, configuring, and hosting LLM apps, as well as supporting streamlined creation, deployment, and maintenance workflows [57]. As shown in Fig. 1, developers can incorporate functionalities such as knowledge bases, plugins, and thinking, while enriching app metadata to improve clarity and usability. For users, LLM app stores offer intuitive interfaces to search, browse, and discover apps by categories or tags. Users can interact with selected apps through multi-turn natural language conversations, experiencing a variety of intelligent services. The platforms also collect and utilize user behavioral data and feedback, including ratings, usage statistics, and number of conversations, as important factors in app recommendation and ranking.

In recent years, the low technical barrier for LLM app development has led to an explosive growth in the number of available apps on these platforms. This rapid proliferation creates significant challenges for users, who often struggle to efficiently identify and select high-quality apps that best meet their needs. Addressing this challenge and enabling effective discovery and evaluation of LLM apps is a critical motivation for our work.

2.2 LLM-based Evaluation Paradigms

Traditional evaluation methods often lack flexibility and semantic understanding when assessing complex language outputs. To address this, researchers have begun using LLMs as evaluators, a paradigm known as **LLM-as-a-Judge**. This approach leverages the LLM's reasoning abilities to overcome the rigidity, limited scalability, and semantic shortcomings of traditional methods. A line of pioneering studies has demonstrated the potential of this paradigm. For example, Chiang et al. [8] showed that LLM-based evaluations are often highly consistent with expert human judgment, especially in open-ended tasks like story generation. In the education domain, Chang et al. [53] and Grévisse et al. [12] leveraged GPT-family and Gemini models, respectively, for automatic grading of short-answer questions, achieving promising results in terms of accuracy and efficiency compared to human annotators. These advances indicate that LLM-as-a-Judge could substantially reduce labor costs and improve handling of edge cases in various vertical applications.

Nevertheless, a growing body of research also reveals significant challenges in generalizing this approach. As summarized by Gu et al. [13] and others, the reliability and stability of LLM-based

evaluation are often sensitive to prompt design and model sampling parameters. Furthermore, many current studies rely on limited or single-source test sets and lack systematic scoring standards, which makes it difficult to extend their findings to diverse, real-world LLM application scenarios involving complex interactions. Motivated by these gaps, this work presents the LaQual framework, which systematically optimizes the LLM-as-a-Judge paradigm and, for the first time, applies it to the comprehensive evaluation of complex LLM apps. To overcome the core limitations of previous approaches, particularly the lack of explicit evaluation criteria and insufficiently diverse test sets, LaQual introduces a dynamic generation mechanism that automatically produces task-specific scoring metrics and evaluation questions before assessment.

3 PROPOSED FRAMEWORK

This section introduces the workflow of LaQual. As shown in Fig. 2, LaQual consists of three key stages: *LLM app labeling and classification* (§ 3.1), *static indicator evaluation* (§ 3.2), and *dynamic scenario-adaptive evaluation* (§ 3.3).
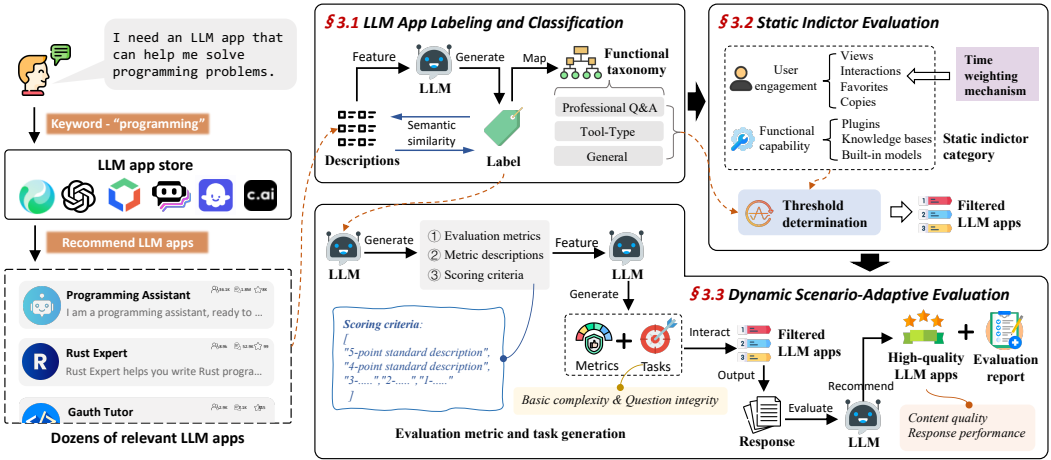

Fig. 2. The Workflow of LaQual.

3.1 LLM App Labeling and Classification

3.1.1 LLM-based Labeling. Before the formal evaluation begins, an *LLM-based annotator* automatically generates labels for all LLM apps to be evaluated. In LLM app stores, each app is associated with a description. The LLM-based annotator performs semantic analysis on this textual information to extract key domain entities, functional elements, and typical usage scenarios. Based on this analysis, the annotator generates a candidate label that summarizes the core functionality of the app. To ensure that labels are concise and accurate, LaQual limits their length. For example, if it is Chinese, it can be limited to a phrase containing 6-9 Chinese characters. Fig. 3 illustrates the prompt of the LLM-based annotator.

We then use the *text2vec* model to further verify the accuracy of the generated labels. Specifically, this model calculates the semantic similarity between each candidate label and the original description of the LLM app. A similarity threshold of 0.7 is set; if the calculated similarity falls below this threshold, a feedback mechanism is immediately triggered, prompting the LLM to iteratively refine the label until the similarity requirement is met or exceeded. It is important to note that

the threshold of 0.7 is empirically determined, based on extensive experimental validation and a comprehensive assessment of both label quality and iteration efficiency under different thresholds.
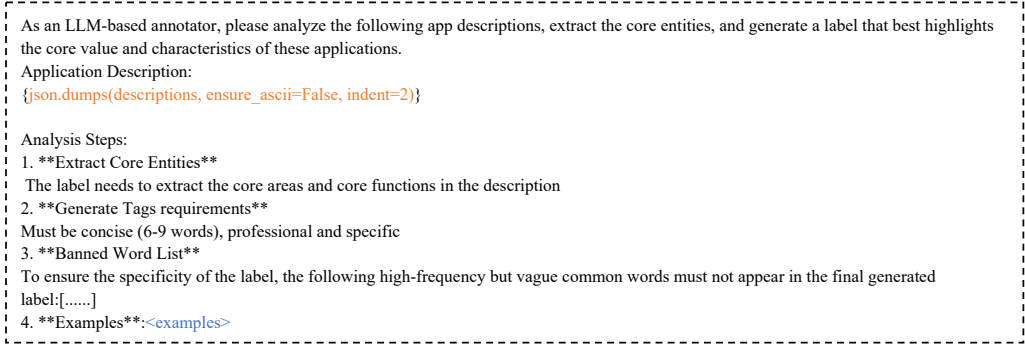
As an LLM-based annotator, please analyze the following app descriptions, extract the core entities, and generate a label that best highlights the core value and characteristics of these applications.
Application Description:
{json.dumps(descriptions, ensure_ascii=False, indent=2)}

Analysis Steps:
1. **Extract Core Entities**
 The label needs to extract the core areas and core functions in the description
2. **Generate Tags requirements**
Must be concise (6-9 words), professional and specific
3. **Banned Word List**
To ensure the specificity of the label, the following high-frequency but vague common words must not appear in the final generated label:[......]
4. **Examples**:<examples>

Fig. 3. Prompt Template for LLM-based Annotator.

3.1.2 Hierarchical Classification. After generating labels for each LLM app, the next step is to determine its specific category. As illustrated in Table 1, we designed a three-level hierarchical taxonomy, consisting of *category*, *subcategory*, and *tag*, to categorize LLM apps based on their core functional characteristics. The design of this taxonomy draws on Su et al. [41]'s empirical study on GPT Store, systematically examining various classification methods used in official and third-party app stores. Based on insights into the functional distribution of tens of thousands of apps, we refined and established three highly generalizable and distinct categories: *professional Q&A*, *tool-type*, and *general*. This taxonomy encompasses a total of ten subcategories and 62 tags.

Table 1. Functional Taxonomy of LLM Apps.

Category	Description	Subcategory	Tags
Professional Q&A	Focus on professional knowledge services in vertical domains.	Medical Health	Medical, Health, Disease, Diagnosis, Treatment, Psychology
		Legal Consultation	Laws, Regulations, Contracts, Litigation
		Financial Management	Finance, Financial Management, Investing, Stocks
		Education	Education, Training, Courses, Mathematics, Chinese, Biology, Physics, Chemistry, History, Geography, Politics
		Language Learning	Translation, Vocabulary, Translation, Grammar, Language, Chinese, English, Japanese, German, Korean, French, Russian, Italian
		Professional Consulting	Consulting, Consultant, Expert
Tool-type	Designed to accomplish specific tasks as the goal.	Development Tools	Code, Programming, Development, Debugging, Algorithms, Front-end, Back-end, Python, Java, Javascript, Matlab
		Analysis Tools	Data Analysis, Statistical Analysis, Data Forecasting
		Planning Tools	Planning, Planner, Planning Design
		Design Tools	Interior Design, Branding
General	Default. / Others.	–	–

Rather than directly mapping the app descriptions to the predefined classification tags, LAQUAL instead maps these tags using the generated labels in the previous step. **These labels serve as a semantic bridge, enabling more accurate and robust mapping to our fine-grained taxonomy by reducing the noise and ambiguity of the original descriptions.** LAQUAL

employs a hierarchical matching strategy, from fine-grained to coarse-grained, prioritizing tag over subcategory. In particular, LaQual first attempts to match label with the most specific tag keywords, such as "litigation" and "contract" under the "law" subcategory, to ensure precise classification. If no match is achieved at the tag level, the process falls back to broader subcategory keywords, such as "legal consultation" or "financial management", for subsequent matching. This hierarchical retrieval approach not only ensures classification accuracy but also significantly improves the efficiency and robustness of the identification process.

3.2 Static Indicator Evaluation

3.2.1 Static Indicator Selection. The first step in static indicator evaluation is to determine the scope of indicators to consider. We selected two-dimensional evaluation indicators based on objective and readily available platform data. Our goal is to provide a comprehensive evaluation that reflects both user interaction patterns and the inherent technical capabilities of each LLM app.

User engagement indicators, which measure the app's popularity and recognition among the user community; and **functional capability indicators**, which assess the app's configuration richness and versatility. This dual-perspective approach enables us to distinguish between apps that attract broad user attention and those that offer advanced technical features. Table 2 lists the detailed indicators used in our evaluation framework. For example, *copies*, or configuration replication frequency, reflects an app's influence among developers, while *plugins* indicates its extensibility and the ability to integrate with external tools or services.

Table 2. Static Evaluation Indicator for LLM Apps.

Category	Indicator	Definition
User Engagement	Views	The total number of times the app's main page is viewed by users.
	Interactions	The total number of conversions or interactions with the app.
	Favorites	The number of times users add the app to their personal favorites.
	Copies	The number of times users duplicate the app's configuration to create their version.
Functional Capability	Plugins	The number of external services integrated into the app to extend its functionality.
	Knowledge Bases	The count of distinct knowledge bases connected to the app.
	Built-in Models	The number of built-in models in the app.

3.2.2 Time weighting and indicator standardization. After determining the static indicators, we observe that newer apps in the LLM app store are often disadvantaged in evaluation because they have had less time to accumulate indicator values, whereas older apps may appear artificially strong due to the accumulation of historical data. This temporal accumulation bias can therefore obscure the real-time competitiveness of apps. To address this issue, we introduce a **time-decay dynamic weighting mechanism** that adjusts static indicators according to each app's operational age. The core idea is to ensure that the contribution of any cumulative indicator decreases as time progresses, thereby placing greater emphasis on recent performance rather than historical accumulation.

We first segment the lifecycle of each app into discrete evaluation periods, using the quarter as the basic analysis unit. Let M denote the total number of months since an app's release; the corresponding number of operating quarters Q is computed as:

$$Q = \max(1, \lfloor M/3 \rfloor) \tag{1}$$

where $\lfloor \cdot \rfloor$ represents the floor function, M is the total operation months and $\max(1, \cdot)$ ensures that even apps with less than three months of operation are counted as having completed one quarter. This guarantees that both new and long-standing apps are fairly included in the evaluation.

Next, for each of the user engagement indicators (i.e., views, interactions, favorites, and copies), we sequentially calculate its quarterly average and apply a time-based decay coefficient. We let X_{total} denote the total cumulative value of any given user engagement metric recorded over its entire operational period. The standardized, time-weighted value X_{weighted} is given by:

$$X_{\text{weighted}} = \left(\frac{X_{\text{total}}}{Q}\right) \cdot \omega_t \tag{2}$$

This formula first calculates the quarterly mean of the indicator (X_{total}/Q) to eliminate the total amount difference caused by the operating time. Then, it reflects the timeliness by multiplying by a decay coefficient (ω_t) that decreases over time. The decay coefficient ω_t is defined as an exponential function of the number of operating quarters Q:

$$\omega_t = \beta^{Q-1} \tag{3}$$

where $\beta \in (0, 1)$ is the decay base that determines the rate at which past performance is discounted. In our implementation, we set $\beta = 0.99$, meaning each additional quarter reduces the weight of historical data to 99% of the previous period. For example, a new app ($Q = 1$) retains full weight ($\omega_t = 1$), while an app active for four quarters ($Q = 4$) is weighted by $\omega_t = 0.99^3 \approx 0.9703$.

This exponential decay mechanism effectively suppresses the inertia of historical data, thereby making the evaluation results more responsive to recent user feedback and app store dynamics. As a result, LaQual significantly improves both the timeliness and predictive power of the evaluation process. The decay base β serves as a critical hyperparameter that governs the rate at which historical data are discounted. The choice of β should be tailored to the specific characteristics of the app store, such as platform popularity, the total number of available apps, and the level of user activity. A smaller value of β leads to a faster decay, placing greater emphasis on the most recent market performance. In contrast, a β value closer to one retains more of the historical performance, favoring long-term stability. By fine-tuning β, LaQual can be flexibly adapted to different evaluation scenarios and platform characteristics.

3.2.3 Threshold Determination. Building upon the LLM app classification in § 3.1.2 and the time-weighted static indicator evaluation in § 3.2.2, this step focuses on the scientific determination of admission thresholds for each evaluation indicator. Once each LLM app is categorized, we establish a set of `Category-Indicator-Threshold` mapping rules to ensure that the entry criteria are both differentiated by category and grounded in empirical data. Using the AppBuilder as a representative platform, we set baseline thresholds for key indicators across different LLM app categories, as shown in Table 3. For example, *professional Q&A* apps are required to meet a minimum knowledge base size (≥ 1), while *tool-type* apps are primarily assessed by the number of functional components (≥ 2). These thresholds are designed to guarantee that candidate LLM apps possess the essential capabilities, user engagement, and professionalism expected for their respective categories.

Table 3. Differentiated Admission Thresholds for Different App Categories (AppBuilder).

Metric Dimension (Quarterly)		Professional Q&A	Tool-type	General
User Engagement	Views	≥ 20	≥ 20	≥ 20
	Interactions	≥ 20	≥ 20	≥ 20
	Favorites	≥ 20	≥ 20	≥ 20
	Copies	≥ 20	≥ 20	≥ 20
Functional Capability	Plugins	≥ 0	≥ 2	≥ 0
	Knowledge Bases	≥ 1	≥ 0	≥ 0
	Built-in Models	≥ 1	≥ 1	≥ 1

The thresholds are determined through multiple rounds of empirical validation and data analysis. In particular, user engagement metrics are closely aligned with the specific characteristics of the LLM app store, such as platform popularity and user activity levels. Therefore, when adapting this framework to other platforms, we recommend calibrating these thresholds based on the actual data distribution of the target environment to ensure optimal evaluation performance. Once the thresholds are established, we apply the time-weighted evaluation algorithm described in § 3.2.2 to compute weighted user engagement scores for each LLM app. These weighted indicators, along with functional capability metrics, are then compared against their respective thresholds to assess qualification. At this stage, we intentionally adopt relatively inclusive thresholds to efficiently filter out apps with clear functional shortcomings, while minimizing the risk of excluding promising candidates. This strategy helps maintain a sufficiently large and high-quality candidate pool for subsequent, more detailed dynamic evaluations.

3.3 Dynamic Scenario-Adaptive Evaluation

Dynamic scenario-adaptive evaluation is the core of LaQual, which addresses the limitations of traditional fixed metrics by dynamically constructing evaluation criteria for a variety of application scenarios. Based on the labels generated for each LLM app by an LLM-based annotator (§ 3.1), the system systematically designs relevant evaluation metrics and scoring criteria. It then generates corresponding evaluation tasks, ensuring that the quantitative evaluation of LLM apps accurately reflects their real-world functions and usage contexts.

3.3.1 Evaluation Metric and Task Generation. For each LLM app, the label generated by the LLM-based annotator (§ 3.1) serves as the anchor for designing targeted evaluation metrics. Guided by the principles of functional uniqueness, quantifiability, and content orientation, LaQual constructs a set of metrics that directly correspond to the app's core capabilities. The prompt template shown in Fig. 4 is used to instruct the LLM to generate evaluation metrics based on the given label, resulting in three core metrics for each label. Corresponding scoring criteria are also generated for each evaluation metric. By providing segmented descriptions for each score level, these metrics are translated into clearly structured and quantifiable scoring guidelines.

As a top-tier LLM app quality assessment expert, please design three core content quality assessment metrics for apps labeled "{label}"

Please ensure that each generated metric and its scoring criteria strictly adhere to all of the following specifications.
1. **Basic Requirements**:[......]

2. **Core Constraints on Assessment Scope**:[......]
In principle, metrics must be based entirely on subjective quality assessments of generated text content. Therefore, any metrics that require external objective data or non-textual content for evaluation are strictly prohibited.
Specifically, the following aspects are prohibited from being evaluated:[......]

3. **Specific Requirements for Scoring Criteria**:[......]

Output format:
Please return in the following JSON format:{{
"Metric 1": {{
"Description": "Metric Description",
"Rating Criteria": [......]
}},
{{
"Metric 2": {{
......
}},
{{
"Metric 3": {{
......
}}

}}

Fig. 4. Prompt Template for Dynamic Evaluation Metric Generation.

Based on these metrics, LaQual dynamically generates evaluation tasks tailored to each LLM app. Leveraging the core label and corresponding metrics, LaQual simulates real user interaction to create tasks that reflect authentic user needs and diverse usage contexts. LaQual can produce

detailed and targeted evaluation tasks, with multiple rounds of quality verification to ensure both accuracy and reliability. As illustrated in Fig. 5, each evaluation prompt integrates the metric name and description, explicit task requirements, instructive examples, and relevant content constraints. This ensures that every task is well-aligned with specific evaluation objectives.

You are a professional LLM app evaluation expert. Please generate a specific, actionable evaluation question based on the following information.

Evaluation Information
* **Evaluation Target:** {metric_name} in the {label} domain
* **Metric Description:** {description}

1. **Task Requirements:**Please generate a natural, high-quality user question that strictly meets all of the following criteria:<Criteria>
2. **Reference Examples**:<Examples>
** Output Format**

Please strictly follow the following format and do not include any other content:

Question: [Question Content]

Fig. 5. Prompt Template for Dynamic Evaluation Tasks Generation.

To guarantee task quality, LaQual employs a two-stage filtering process. First, syntactic integrity inspection is performed to eliminate structurally flawed tasks, such as those with incomplete sentences, incorrect punctuation, or ambiguous wording. Second, complexity inspection assesses whether the tasks contain sufficient information, context, and challenge to enable effective evaluation of the target metric. Tasks that are overly simplistic, vague, or lacking in necessary detail are filtered out at this stage. If a candidate task fails either check, a backoff and retry mechanism is triggered: LaQual regenerates the task, with potential adjustments to prompts or constraints, until all quality criteria are satisfied and a high-quality evaluation task is produced. The transformation of abstract metrics into concrete, context-specific evaluation tasks enables LaQual to establish a standardized and adaptable framework. This design supports flexible and comprehensive evaluation across a wide range of LLM apps.

3.3.2   Response Quality Evaluation. Building on the evaluation metrics and evaluation tasks generated in § 3.3.1, we next focus on systematically evaluating the quality of the dynamic interactions between LaQual and LLM apps. LaQual integrates both **content quality evaluation** and **response performance evaluation**, forming a closed feedback loop from app evaluation to actionable improvement suggestions. Such a design provides a scientific basis for the functional iteration and performance tuning of LLM apps.

Content quality evaluation. This evaluation adopts an LLM-based evaluation paradigm, in which a structured prompt template guides the evaluation model to make quality judgments on LLM app responses. The prompt template, as illustrated in Fig. 6, is constructed using a context embedding approach. It explicitly incorporates three essential elements: *original task*, *response content*, and *scoring criteria*, thus forming a logically complete evaluation context. Semantic alignment between the task and response is emphasized to ensure the evaluation model accurately understands the correspondence between user intent and generated content. Meanwhile, explicit scoring criteria transform abstract evaluation metrics into concrete judgment rules, thereby reducing subjective bias in the evaluation process.

To ensure the stability and consistency of evaluation results, model parameters are finely configured during the LLM invocation. Specifically, we set `temperature=0` to eliminate randomness and enhance reproducibility; `frequency_penalty=0.5` to discourage repetitive expressions and

Fig. 6. Prompt Template for Content Quality Evaluation.

increase informativeness; `max_tokens=5000` to accommodate complex evaluation scenarios, and `timeout=60` seconds to prevent abnormal response delays. Besides, five-level scale values [1, 5] are used to quantitatively rate content quality, with each level clearly defined. Alongside the score, the evaluation model outputs a structured analysis of advantages, weaknesses, and targeted improvement suggestions. The final quantified score obtained through this entire systematic evaluation process is defined as the *content quality score*, denoted as S_{CQ}. This comprehensive framework ensures systematic and objective evaluation, providing precise guidance for app optimization.

Response performance evaluation. This evaluation quantitatively analyzes the real-time interaction capability of LLM apps, with a core focus on ensuring a fluid user experience. As low latency and high throughput are critical for smooth real-time interactions, our design centers on a comprehensive performance metric: *response efficiency* (η). The metric is defined as:

$$\eta = \frac{\text{Tokens}}{\text{R\_time}} \tag{4}$$

where `R_time` denotes the total elapsed time (in seconds) from the user's query submission to the receipt of the last token in the LLM-generated response, and `Tokens` refers to the total number of tokens in the response. By collecting real-time performance data during LLM app operation, response efficiency provides a direct measure of the system's speed and capability in processing and generating outputs. To translate this continuous metric into intuitive and comparable scores, we adopt a discretization strategy based on predefined efficiency thresholds. Specifically, a lookup table, as shown in Table 4, is designed to map each measured response efficiency value to a categorical score within [1, 5]. The final quantified score obtained through this method is defined as the *response performance score*, denoted as S_{RP}. This module offers a quantifiable basis for performance diagnosis, resource allocation optimization, and service quality assurance, substantially enhancing the practical and engineering value of the overall evaluation framework.

Table 4. Response Efficiency Scoring.

Metric Dimension (Quarterly)	Excellent (5 pts)	Good (4 pts)	Average (3 pts)	Poor (2 pts)	Very Poor (1 pt)
Response Efficiency	≥25 tokens/sec	≥20 tokens/sec	≥15 tokens/sec	≥10 tokens/sec	<10 tokens/sec

3.3.3 Weighted Composite Scoring. After obtaining the independent *content quality score* (S_{CQ}) and *response performance score* (S_{RP}), a composite scoring mechanism is applied to ensure that the evaluation results comprehensively and objectively reflect the overall quality of LLM apps. The final composite score is calculated as follows:

$$\text{Score} = \alpha \times S_{CQ} + \beta \times S_{RP} \tag{5}$$

where the weighting coefficients satisfy $\alpha + \beta = 1$ and $\alpha > \beta$. Based on empirical analysis, α is set to 0.8 and β to 0.2. The reason is, although response performance is important for user experience, its marginal utility is generally lower than that of content quality. By prioritizing content quality in the weighting, the evaluation system encourages the optimization of substantive output while still considering improvements in response efficiency. The composite evaluation score serves not only as a quantitative indicator of overall application performance but also as an important basis for iterative optimization. This comprehensive evaluation enables quick identification of both the overall performance level and the specific strengths or weaknesses of an LLM app, providing robust decision support for the continuous improvement and targeted tuning of LLM apps.

4 EVALUATION

This section systematically examines the effectiveness of the LaQual framework in real-world LLM app scenarios. We focus on evaluation accuracy, recommendation rationality, and user experience.

Our evaluation is guided by the following research questions (RQs):

RQ1: Can LaQual provide accurate and scenario-adaptive quality evaluation for various types of LLM apps? This question aims to examine whether the automated evaluation results from LaQual are consistent with user judgments and dynamically adapt to different LLM app scenarios.

RQ2: How does LaQual perform in terms of recommendation quality and rationality compared to recommendation mechanisms in mainstream LLM app stores? Here, we investigate whether LaQual can offer a more comprehensive and reasonable recommendation mechanism that helps users identify truly high-quality LLM apps, surpassing existing LLM app store recommendation mechanisms.

RQ3: What is the impact of LaQual on user decision quality, cognitive effort, and trust in real-world LLM app discovery tasks? We focus on determining if the adoption of LaQual leads to concrete improvements in user decision-making, reduces mental workload, and enhances trust during the LLM app selection process.

4.1 RQ1: Accuracy and Adaptivity

4.1.1 Experimental Setup. All experiments were conducted on the AppBuilder [2], a mainstream LLM app store, which provides a diverse set of LLM apps and a rich array of functional configurations. Two distinct scenarios were selected as experimental subjects to ensure coverage of both *professional Q&A* and *tool-type* LLM app categories: (1) *legal consulting* and (2) *travel planning*. For each scenario, a single scenario-specific keyword was used to retrieve candidate LLM apps from the platform, reflecting the typical user search and recommendation logic in LLM app stores. Specifically, the keyword "lawyer" was used for the *legal consulting* scenario, which yielded 18 candidate apps, while "smart travel planning assistant" was used for the *travel planning* scenario, yielding 32 candidate apps. These numbers represent the original set of apps to be evaluated for each scenario.

Subsequently, each set of candidate apps was independently evaluated by LaQual. For all LLM-based evaluation modules in LaQual, we used "Qwen-32B". It is worth noting that the underlying LLM in LaQual is modular and can be easily replaced with other LLMs as needed. In parallel, a panel of 28 human evaluators, including several domain experts, assessed the same LLM apps using the same metrics. All evaluators were experienced LLM app users aged 18–50, with several possessing relevant domain expertise.

4.1.2 Automated Evaluation by LaQual. The automated evaluation process in LaQual comprises three main stages: (1) LLM app labeling and classification, (2) static indicator evaluation, and (3) dynamic scenario-adaptive evaluation.

LLM app labeling and classification. Before formal evaluation, LaQual employs an LLM-based annotator to automatically generate concise, scenario-specific labels for each app based on its name and description. These labels distill the core functionality and typical usage scenario of the app, facilitating accurate downstream classification. Among the 18 LLM apps retrieved using the keyword "lawyer" in the legal consulting scenario, the annotator consistently generated the label "laws consulting analysis" for these apps. According to the functional taxonomy, this label corresponds to the "Laws" tag, which is categorized under the *professional Q&A* category. Similarly, for the 32 LLM apps retrieved with the keyword "smart travel planning assistant" in the travel planning scenario, the annotator generated the label "travel itinerary planning". This label maps to the "Planning" tag, which falls under the *tool-type* category in the taxonomy. Each app is thus mapped to a fine-grained hierarchical taxonomy, including category, subcategory, and tag. This process ensures that subsequent evaluation criteria and thresholds are precisely aligned with the app's intended usage context, thereby enabling scenario-adaptive evaluation.

Static indicator evaluation. Static evaluation aims to efficiently reduce the candidate pool size and improve baseline app quality before dynamic evaluation. For the legal consulting scenario, the initial set of 18 apps was filtered using a combination of user engagement metrics (views, interactions, favorites, copies) and functional capability metrics (plugins, knowledge bases, bulit-in models), together with category-specific thresholds and time-decay weighting. As a result, LaQual selected 6 high-potential candidates. Similarly, in the travel planning scenario, the candidate pool was reduced from 32 to 6 apps. Table 5 summarizes the static indicator values for all shortlisted apps. All retained apps met or exceeded the required thresholds across both user engagement and functional capability dimensions, ensuring a high-quality pool for downstream evaluation.

Table 5. Static Indicator Values for Shortlisted LLM Apps.

Category	Metric	La1[1]	La2	La3	La4	La5	La6	Ta1[2]	Ta2	Ta3	Ta4	Ta5	Ta6
User Engagement	Views	154	26	19	28	271	70	20	103	17	31	12	14
	Interactions	425	157	78	234	15,000	68	132	59	117	345	169	36
	Favorites	1	0	0	0	4	0	2	0	0	0	0	0
	Copies	0	1	0	0	0	0	0	1	0	1	1	0
Functional Capability	Plugins	3	0	4	2	0	6	5	3	4	6	4	4
	Knowledge Bases	15	1	1	1	2	1	2	1	1	1	1	1
	Built-in Models	2	2	2	2	2	2	2	2	2	2	2	2

[1] LaX denotes apps in the *legal consulting* scenario (La = Legal consulting app).
[2] TaX denotes apps in the *travel planning* scenario (Ta = Travel planning app).

Dynamic scenario-adaptive evaluation. Following static screening, the dynamic evaluation module of LaQual automatically adapts its metrics to the application's scenario, leveraging the previously generated labels. For every shortlisted app, the framework dynamically constructs three core evaluation metrics tailored to the scenario, and for each metric, it defines a five-level quantitative scoring metric, values set as [1, 5], with detailed criteria for each level. Representative scenario-specific evaluation tasks are designed to ensure the assessment's authenticity and relevance. For the legal consulting and travel planning scenario, LaQual establishes three core evaluation metrics for each context, as presented in Table 6. The scoring metrics ensure that different levels of app performance are clearly and fairly distinguished.

LLM apps demonstrated varying performance across core metrics. In the legal consulting scenario, most apps received scores of 3–4, indicating generally good performance, while a few had lower scores on certain metrics, such as legal citation accuracy. In the travel planning scenario, top apps consistently scored higher, while others showed weaknesses in itinerary logic and personalization.

Table 6. Scenario-Adaptive Dynamic Evaluation Metrics and Tasks in LaQual.

Scenario	Metric	Definition	Task Example	Score
Legal Consulting	Legal Citation Accuracy	Assesses whether legal terms cited in the answer are accurate and relevant to the question.	"If an employee is dismissed without notice, which labor law articles apply?"	1–5
	Terminology Clarity	Evaluates whether legal terminology is used appropriately and clearly explained.	"Explain the meaning of 'force majeure' in a contract dispute to a layperson."	1–5
	Answer Completeness & Logic	Judges if the answer addresses key legal points, is logically organized, and has a well-founded conclusion.	"I'm a tenant and I need to break my lease early. What are the legal steps I need to follow? What risks should I be aware of?"	1–5
Travel Planning	Personalization Match	Measures whether the plan meets all personalized requirements (e.g., budget, interests, time constraints).	"Plan a 3-day trip to Chengdu for a family with kids, with a budget under 3000 RMB."	1–5
	Itinerary Logic & Flow	Assesses the reasonableness and flow of the itinerary, including transportation and activity planning.	"Design a 5-day Japan itinerary covering Tokyo, Kyoto, and Osaka with minimal backtracking."	1–5
	Content Innovation	Evaluates whether recommendations include unique and insightful local attractions or activities.	"Recommend lesser-known attractions or local experiences in Florence for art lovers."	1–5

These results highlight the framework's ability to provide fine-grained, scenario-relevant evaluation and to identify both strengths and weaknesses of each app.

4.1.3 Human Evaluation and Consistency Analysis. To benchmark the accuracy and adaptivity of LaQual, the six shortlisted apps from each scenario were independently evaluated by the panel of 28 human raters, including domain experts, using the same dynamically generated metrics and a 5-point scale. We then compared the scores from LaQual with the average human ratings.
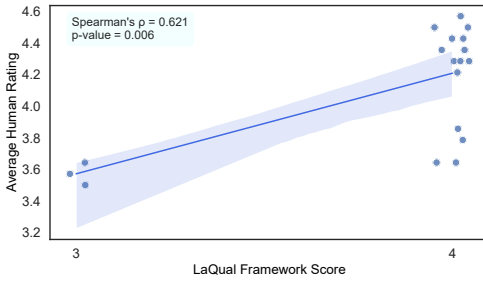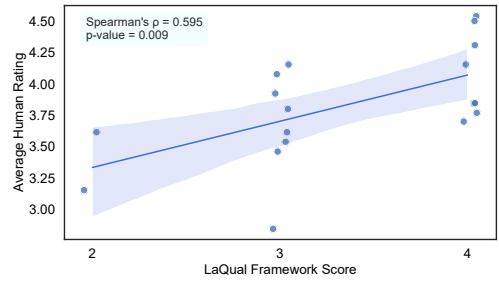
(a) *Legal Consulting* Scenario.   (b) *Travel Planning* Scenario.

Fig. 7. Correlation Analysis between LaQual and Human Evaluation.

For the legal consulting scenario, Spearman's rank correlation analysis was conducted between the human average scores and LaQual's content scores across all 18 evaluation items (six apps, three metrics each). As shown in Fig. 7a, the correlation coefficient (ρ) [36] was **0.621** with a p-value of **0.006**, indicating a moderately strong and statistically significant positive correlation. This demonstrates that LaQual's automated evaluation is highly consistent with human expert judgment. Regarding response performance, it's noted that no direct correlation with human

ratings was assessed due to differences in measurement: human scores reflect subjective fluency perceptions, while LaQual computes objective response efficiency. Nevertheless, the response performance score (S_{RP}) remains a critical component of the final composite evaluation, ensuring that both content quality and interaction fluency are reflected in the overall assessment.

Similarly, for the travel planning scenario, the consistency between LaQual and human raters was evaluated using the same method. As shown in Fig. 7b, Spearman's correlation coefficient was **0.595** with a p-value of **0.009**, again indicating a moderately strong and statistically significant positive correlation. This further validates LaQual's adaptability and accuracy across different application domains.

Answer to RQ1: The results show that LaQual achieves high consistency with human judgments across diverse LLM app scenarios. In the legal consulting case, the Spearman correlation coefficient between LaQual's automated scores and user ratings reached 0.621 (p = 0.006); in the travel planning case, the coefficient was 0.595 (p = 0.009). This indicates that LaQual not only provides accurate evaluations, but also dynamically adapts its assessment criteria to fit different domains.

4.2 RQ2: Recommendation Effectiveness

RQ2 conducts a further analysis based on the experimental results of RQ1 to verify that LaQual significantly outperforms mainstream LLM app store recommendation mechanisms in both recommendation quality and rationality. In the legal consulting scenario, for instance, the app store directly recommended 18 relevant apps for the keyword "lawyer", but LaQual's static indictor evaluation automatically filtered this pool down to 6, eliminating 66.7% of the candidates. Similarly, in the travel planning scenario, the initial 32 apps retrieved by the app store were narrowed to just 6 (an 81.25% reduction) after screening with differentiated, domain-specific rules. Manual review of the 12 and 26 apps filtered out in these two scenarios confirmed that most excluded apps suffered from one or more of the following issues:

- **Lack of market vitality and user recognition:** Many filtered-out apps were zombie apps with very low or declining user activity, or early-stage apps that had not yet gained meaningful user recognition.
- **Absence of domain-specific optimization:** Some apps provided only generic responses based on foundation models, lacking enhancements such as tailored knowledge base, resulting in shallow, outdated, or unstable answers.
- **Functional deficiencies:** Especially in the travel planning scenario, several eliminated apps lacked essential tool integrations (e.g., map navigation, weather plugins), significantly limiting their practical utility.
- **Superficial app design:** A portion of filtered apps demonstrated minimal differentiation in content or user interaction, and failed to address core user needs within the scenario.

Notably, several of the apps ranked among the top recommendations in the app store were identified as low-quality and filtered out by LaQual, further demonstrating the shortcomings of existing recommendation mechanisms and highlighting the rationality and effectiveness of LaQual's approach in surfacing truly high-quality apps.

Answer to RQ2: LaQual outperforms mainstream LLM app store recommendation mechanisms in both recommendation quality and rationality. For example, in the legal consulting scenario, LaQual reduced the candidate pool from 18 to 6 apps (a 66.7% reduction), and in the travel

planning scenario, from 32 to 6 apps (an 81.25% reduction), by systematically filtering out low-quality or irrelevant apps. Manual review further confirmed that the excluded apps typically lacked user recognition or essential domain features, demonstrating the rationality and effectiveness of LaQual's recommendation process.

4.3 RQ3: User Impact

This RQ examines the practical value of the framework in real-world LLM app discovery scenarios. Specifically, it investigates whether the dynamic evaluation mechanism provided by LaQual can substantially improve user decision quality, cognitive efficiency, and trust, compared to a baseline system. The overarching goal is to assess the framework's effectiveness as a decision support tool for high-quality LLM app selection.

4.3.1 Research Hypotheses. Based on the core challenges users face in the LLM app store, such as information overload and difficulty discerning quality, we propose the following hypotheses:

- **RH1: Decision quality advantage.** Users perceive that apps recommended by LaQual more closely meet their complex or personalized core needs, resulting in better final choices compared to the baseline system.
- **RH2: Cognitive efficiency advantage.** While LaQual may require slightly longer initial evaluation time, its structured and in-depth reports significantly reduce users' mental effort during app comparison and selection, enhancing decision confidence.
- **RH3: Trust and transparency.** The evaluation reports generated by LaQual are regarded as more rational, transparent, and credible than the baseline system's star ratings and fragmented user reviews, thus fostering greater user trust.

4.3.2 Task Design. To empirically validate these hypotheses, we conducted a controlled user study with a within-subjects design. Twelve participants were recruited to complete six representative app discovery tasks using both the LaQual framework and a baseline system (i.e., AppBuilder). The tasks spanned different application types, with two tasks per type. To mitigate order effects, participants were randomly assigned to two groups in a counterbalanced sequence:

Group A (n=6): Used the baseline system first, then LaQual.
Group B (n=6): Used LaQual first, then the baseline system.

After completing all tasks, participants filled out a post-task questionnaire based on a seven-point Likert scale (1 = Strongly Disagree, 7 = Strongly Agree). The questionnaire was designed around the three research hypotheses and covered four key evaluation dimensions, as detailed in Table 7.

4.3.3 Results. We analyzed the questionnaire data using a paired-samples t-test, as each participant experienced both systems. The results, summarized in Table 8, demonstrate that LaQual significantly outperformed the baseline system across all evaluation dimensions.

As visualized in Fig. 8, LaQual achieved notably higher mean scores (all statistically significant at the p < 0.001 level, as indicated by *** in the figure). The advantage was particularly pronounced in the dimensions of *comparison efficiency* (M=5.45 vs. M=3.30) and "Value of Evaluation Report" (M=4.75 vs. M=2.25). These results highlight that the structured evaluation information provided by LaQual substantially aids users in making efficient and confident app comparisons. Furthermore, LaQual also demonstrated marked advantages in *decision confidence*, *result credibility*, and the *discovery of "surprise" apps*, further validating its effectiveness in improving decision quality and user trust.

Table 7. Quantitative Evaluation Dimensions and Metrics for the User Study.

Dimension	Weight	Metric	Description
Decision Support Benefits	15%	Decision Confidence	With the assistance of the evaluation information provided by LaQual framework, I can make a final choice with confidence.
	15%	Comparison Efficiency	The framework helps me easily compare the pros and cons of different LLM apps, reducing my cognitive and decision-making load.
Evaluation Quality & Credibility	15%	Value of Report	The evaluation rationale/report provided by the framework is substantively helpful for my decision-making.
	15%	Result Credibility	I believe the ratings given by LaQual are objective and reliable.
Depth of Demand Matching	20%	Deep Demand Fulfillment	The LaQual's recommendations not only met my surface-level needs but also considered potential aspects I hadn't thought of.
Exploration & Discovery Value	20%	Discovery of "Surprise" Apps	The system helped me discover high-quality apps that I might have overlooked in a traditional recommendation mechanism but are perfectly suited for me.

Table 8. Paired-Samples T-Test Results Comparing LaQual and the Baseline System.

| Evaluation Category	Sub-dimension	LaQual Mean	Baseline Mean	LaQual SD	Baseline SD
Decision Support Benefits	Decision Confidence (Dim. 1)	5.050	4.400	0.686	0.503
	Comparison Efficiency (Dim. 2)	5.450	3.300	0.887	0.733
Evaluation Quality & Credibility	Value of Report (Dim. 3)	4.750	2.250	0.716	0.716
	Result Credibility (Dim. 4)	4.850	3.900	0.587	0.852
Depth of Demand Matching	Deep Demand Fulfillment (Dim. 5)	4.300	3.700	0.733	0.571
Exploration & Discovery Value	Discovery of "Surprise" (Dim. 6)	4.950	3.550	0.686	0.510
Weighted Total Score	**Overall Experience Score**	**4.873**	**3.555**	-	-
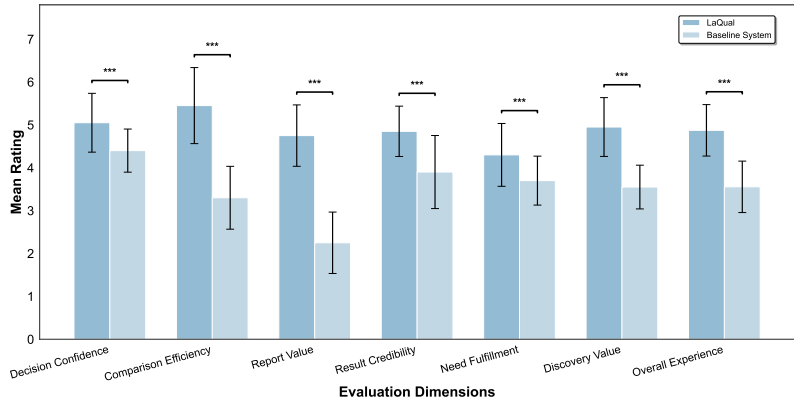
Fig. 8. Mean Score Comparison of LaQual and the Baseline System Across All User Evaluation Dimensions.

Answer to RQ3: User studies indicate that LaQual significantly improves decision quality, reduces cognitive effort, and enhances trust. Quantitative results show that LaQual achieved higher

mean scores than baseline systems across all user evaluation dimensions, with particularly strong advantages in comparison efficiency (M=5.45 vs. M=3.30) and value of evaluation report (M=4.75 vs. M=2.25). All differences were statistically significant, supporting the conclusion that LaQual leads to more confident, efficient, and trustworthy LLM app selection experiences.

5 IMPLICATIONS

Our evaluation framework provides actionable benefits and concrete guidance for all key stakeholders in the LLM app ecosystem, supporting the establishment of a more transparent, high-quality, and trustworthy environment.

LLM app store managers. For store managers, our findings demonstrate that relying solely on static metrics such as downloads or user ratings can easily result in ranking manipulation and poor user experience. By integrating LaQual into routine platform operations, managers can implement automated quality evaluations at both the onboarding and periodic review stages. For example, only apps that meet minimum quality thresholds in scenario-adaptive evaluation can be listed, and regular re-evaluations can help identify apps with declining performance for timely intervention or removal. Additionally, publishing composite and detailed evaluation scores alongside traditional popularity indicators allows users to make more informed decisions, while also encouraging developers to focus on genuine quality improvements. This approach helps build a fairer and more credible app ecosystem.

LLM app developers. For developers, LaQual offers a set of actionable, scenario-specific quality metrics that can be integrated directly into development and release workflows. Developers can use these evaluations during prototyping to quickly identify weaknesses in content, functionality, or user experience, and address them before public release. Incorporating LaQual into continuous integration pipelines enables automatic quality checks with each update, reducing the risk of quality regressions. Furthermore, detailed evaluation reports can help developers prioritize improvements in areas that matter most to users, such as response accuracy or feature completeness. By transparently sharing evaluation results, developers can also strengthen user trust and better position high-quality LLM apps in a competitive market.

LLM app users. End users benefit from our framework through greater transparency and confidence in the LLM apps they select. With access to multi-dimensional evaluation results, users can choose apps that best match their individual needs. For instance, prioritizing response accuracy in legal apps or customization in travel apps. The availability of sub-scores and scenario-based explanations also makes it easier for users to understand the strengths and weaknesses of each app, reducing decision fatigue and the risk of disappointment. As a result, users are empowered to make more informed and satisfying choices, and can provide valuable feedback to further refine quality standards.

Researchers. For the research community, this research contributes two core concepts to the field of LLM app quality evaluation. The first is the adaptability of the evaluation model. We revealed that the inherent judgment logic of the LaQual evaluation framework varies depending on the attributes of the metric. Although all evaluations ultimately output a unified five-point scale, the framework's internal logic for deriving this score shifts based on the inherent differences in the app scenario. When evaluating metrics with clear objective criteria or knowledge boundaries, such as "accuracy of legal references" the criteria are highly quasi-binary, making qualitative grading easier. However, when evaluating metrics that essentially measure "completeness" or "match" such as "personalized match of travel plans" the focus is more on the completeness of the multi-functional

package or personalized satisfaction, resulting in a wider distribution of final scores. We believe that it is precisely through this flexibility in its inherent judgment logic that LaQual achieves accurate evaluation across scenarios.

Secondly, we reveal that tool acts not just as an evaluator, but also as a **stabilizer** for the evaluation ecosystem. It should be considered a stabilizer within the evaluation ecosystem. A recurring phenomenon in our empirical research is that the ratings from human evaluation panels often exhibit substantial dispersion, reflecting significant differences between evaluators due to subjective experience and preferences. In contrast, LaQual provides an objective and stable rating for app evaluations, acting as a stabilizer.

6 THREATS TO VALIDITY

Despite the comprehensive and systematic design of the LaQual framework, several threats to validity remain. We recognize these limitations and have taken targeted measures to mitigate their impact where possible. This section discusses threats from the perspectives of content, internal, and external validity.

Content validity. A major threat to content validity lies in the framework's inability to perform real-world fact-checking. LaQual evaluates content quality based on logical coherence, completeness, and structural integrity, using dynamically generated and scenario-adaptive criteria. However, all assessments rely on the internal knowledge and reasoning abilities of the underlying LLM. For instance, as seen in the *smart travel planning assistant* scenario, the system may assign a high score for itinerary logic even if a recommended restaurant has permanently closed, since the framework cannot verify facts in real time. To address this, our evaluation metrics emphasize aspects that LLMs can reliably assess, such as clarity, relevance, and logical flow, while deliberately avoiding metrics that require external validation. Nevertheless, the lack of real-time accuracy validation represents a crucial area for future enhancement, and integrating fact-checking or retrieval-augmented mechanisms will be important for further improvement.

Secondly, in the static indicator evaluation, content validity may be affected by the subjectivity of threshold setting. Since complete platform data is unavailable, the admission threshold in this framework is determined by practical experience, and its optimality requires further validation. Additionally, fixed thresholds may inadvertently exclude apps targeting niche scenarios. To minimize these threats, we adopt a low-threshold filtering strategy to first eliminate unqualified applications, reducing the risk of filtering out potential high-quality apps. We also set differentiated functional indicator requirements for different app categories. While this approach is preliminary, it considers the characteristics of various apps and maintains filtering efficiency, enabling the framework to achieve scenario-based and accurate evaluation while retaining overall generality.

Internal validity The internal validity of our evaluation may be affected by inherent bias within LLMs. Since the evaluation process is powered by a LLM, it is susceptible to political, cultural, or social biases present in its training data or architecture [1, 6]. These biases can influence scoring results, particularly in subjective or open-ended scenarios, and may compromise fairness and objectivity. To reduce this risk, we employ several mitigation strategies. First, we use carefully designed prompt instructions to constrain LLM behavior. Second, we adopt standardized and scenario-adaptive scoring metrics to minimize ambiguity. Third, we set deterministic decoding parameters, such as temperature at zero, to enhance the consistency of results. Despite these measures, it is not possible to fully eliminate model-internal bias, and interpretation of evaluation outcomes should take this limitation into account.

External validity. Another threat concerns the generalizability of our findings. The current LaQual is limited to evaluating text-based outputs of LLM apps. As the LLM app ecosystem is

evolving toward multi-modal applications, including those involving image generation and audio-visual content, our evaluation methods may not be directly applicable to these new forms. To partially mitigate this limitation, we designed our evaluation tasks and metrics to be scenario-adaptive and structurally extensible, providing a foundation for future adaptation to multi-modal or hybrid applications. However, the current conclusions should be interpreted within the scope of text-centric LLM app scenarios, and extending the framework to multi-modal evaluation remains an important direction for future research.

7 RELATED WORK

7.1 Research on LLM Apps

The explosive growth of LLM apps has given rise to a novel software ecosystem centered on LLM app stores. Although academic research on this ecosystem is still in its infancy, significant strides have already been made in several key areas. Current studies primarily concentrate on three dimensions: ecosystem analysis, security vulnerabilities, and data privacy.

At the ecosystem level, pioneering efforts have begun to map the landscape of this rapidly developing field. Zhao et al. [57] laid the groundwork by proposing the first comprehensive research roadmap for the LLM app ecosystem. Foundational datasets and initial exploratory analyses were contributed by Hou et al. [17], who constructed the large-scale GPTZoo dataset, and by Yan et al. [52], who investigated patterns in LLM app distribution. In terms of security, Hou et al. [16] were among the first to expose the security threats facing LLM app stores, systematically identifying several attack vectors such as app hijacking and prompt injection. Building on this, Xie et al. [50] conducted an in-depth analysis of app squatting and cloning, highlighting the severe risks these imitation attacks pose to the overall trust and stability of the ecosystem. Regarding data privacy, researchers have revealed that the flexibility inherent in app configuration can also introduce significant risks of data leakage. Li et al. [51] demonstrated that private application knowledge files are susceptible to theft via prompt engineering, while Wu et al. [47] uncovered the widespread issue of excessive user data collection throughout the ecosystem. While existing research has made notable advances in understanding the ecosystem's structure as well as its security and privacy risks, most efforts remain focused on external threats and vulnerabilities. Little attention has been paid to systematically evaluating the intrinsic quality of content generated by LLM apps. This represents an important research gap. The framework proposed in this study, LaQual, seeks to fill this gap by enabling systematic and automated evaluation of LLM app content quality.

7.2 Domain-Specific LLM Evaluation

A growing body of research has focused on evaluating LLMs within specialized domains, resulting in a variety of tailored frameworks and benchmarks that address the unique requirements and challenges of different professional fields. In the healthcare domain, recent work has addressed both the opportunities and challenges of applying LLMs to medical tasks. Krolik et al. [20] explored the use of LLMs for automated medical Q&A, while Nazi et al. [25] argued that evaluation should consider not only generative ability but also practical applicability and safety in real-world scenarios. Building on this, Ei et al. [46] introduced a dedicated evaluation framework for medical queries, finding that ChatGPT achieves high accuracy in this context. In finance, Guo et al. [15] proposed FinLMEval to systematically evaluate LLMs on a range of financial tasks. Xie et al. [49] further developed FinBen, the first comprehensive benchmark for financial LLM evaluation, enabling more detailed assessment of model performance in complex financial domains. In education, Xiao et al. [48] built a system for generating reading comprehension exercises using LLMs, validating its effectiveness through both automatic and manual evaluation. Kavadella et al. [19] examined

ChatGPT's role in undergraduate dental education, showing improved student performance with model assistance. The legal field has also seen robust evaluation efforts. Wang et al. [44] compared LLM-generated legal content to reference materials and expert judgments. Li et al. [21] presented LegalAgentBench, which uses metrics such as success rate, progress rate, and BERT-Score for hierarchical legal tasks. Haitao Li et al. [22] introduced LexEval for systematic evaluation in the Chinese legal context, while Savelka et al. [35] showed that GPT-3.5 achieves high accuracy in zero-shot legal text annotation.

Collectively, these domain-specific evaluation frameworks underscore the substantial promise of LLMs in professional and expert-driven contexts. However, their scope is typically confined to individual scenarios, lacking the scalability and adaptability required for broader application landscapes. This limitation highlights a central motivation for our work: to move beyond fixed-scenario evaluations and develop a universal, automated framework capable of dynamically generating scenario-aware evaluation criteria for LLM apps.

7.3 Metrics for LLM evaluation

Mainstream evaluation of LLM content quality largely relies on traditional natural language processing metrics. Surface-level semantic matching is typically assessed using BLEU [33] and ROUGE [23], which measure text similarity based on n-gram statistics such as word order and vocabulary overlap. METEOR [3] improves upon these by enhancing the ability to capture semantic equivalence between texts. More recently, metrics like BERTScore [55] and BLEURT [37] have advanced the field by leveraging pre-trained language models to assess deep semantic similarity, representing a shift from pure surface matching to semantic-level evaluation. However, these methods still exhibit notable limitations in the context of LLM evaluation. Specifically, they often lack domain adaptation and deep semantic understanding, making it difficult to accurately assess outputs that involve creative expression or specialized domain knowledge. As a result, there can be significant discrepancies between automated evaluation scores and the actual semantic quality perceived by users in real-world LLM applications.

7.4 Evaluation Based on Human Preferences

While traditional algorithm-based evaluation methods offer efficiency, they often struggle to capture the nuanced preferences and subjective judgments of human users. Belz and Reiter [10] provided a comprehensive review of human evaluation techniques for natural language generation, underscoring the indispensable value of human judgment in assessing generated text quality. To better align models with user expectations, Ouyang et al. [32] introduced reinforcement learning from human feedback (RLHF), where models are trained using human preference rankings for different outputs. Similarly, Chatbot Arena [9] employs crowdsourcing to evaluate LLMs based on large-scale human preferences, while Shankar et al. [38] proposed EvalGen, a hybrid framework aiming to better align evaluators with human expectations. In the context of code generation, Yang et al. [53] highlighted the critical importance of model alignment with human preferences through systematic evaluation. Although these approaches mark a shift from traditional semantic matching to preference-based assessment, they generally require extensive human annotation and face challenges in maintaining annotation consistency. Consequently, their scalability and practicality for automated, large-scale evaluation in LLM app ecosystems remain limited.

8 CONCLUSION

In this work, we present LaQual, a unified and automated framework for large-scale LLM app quality evaluation. Our approach systematically integrates scenario-aware hierarchical classification,

multi-dimensional static indicators filtering, and LLM-driven dynamic evaluation to provide fine-grained and context-adaptive assessment across diverse LLM app scenarios. Extensive experiments on real-world app store data demonstrate that LaQual achieves strong alignment with expert judgments, substantially reduces the candidate app pool, and significantly improves user decision quality, efficiency, and trust compared to existing recommendation mechanisms. Furthermore, our framework is highly extensible and can be adapted to new evaluation scenarios as the LLM app ecosystem evolves. Moving forward, we plan to further enhance the interpretability of automated evaluation results and explore integration with real-time user feedback, aiming to further enhance the robustness and trustworthiness of LLM app discovery experiences.

9 DATA AVAILABILITY

The implementation details and code for LaQual are publicly available at https://github.com/wangyanainn-ainn/LaQual.

REFERENCES

[1] Xuechunzi Bai, Angelina Wang, Ilia Sucholutsky, and Thomas L Griffiths. 2024. Measuring implicit bias in explicitly unbiased large language models. *arXiv preprint arXiv:2402.04105* (2024).
[2] baidu. 2024. Baidu app center. https://console.bce.baidu.com/ai_apaas/appCenter.
[3] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.
[4] ByteDance. 2024. Cici. https://www.ciciai.com/chat/.
[5] ByteDance. 2024. Coze. https://www.coze.com/.
[6] Yong Cao, Li Zhou, Seolhwa Lee, Laura Cabello, Min Chen, and Daniel Hershcovich. 2023. Assessing cross-cultural alignment between ChatGPT and human societies: An empirical study. *arXiv preprint arXiv:2303.17466* (2023).
[7] Ning Chen, Steven C.H. Hoi, Shaohua Li, and Xiaokui Xiao. 2016. Mobile App Tagging. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (San Francisco, California, USA) *(WSDM '16)*. Association for Computing Machinery, New York, NY, USA, 63–72. https://doi.org/10.1145/2835776.2835812
[8] Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937* (2023).
[9] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*.
[10] Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2022. A survey of natural language generation. *Comput. Surveys* 55, 8 (2022), 1–38.
[11] GPTStore.AI. 2024. GPTStore.AI. https://gptstore.ai/gpts.
[12] Christian Grévisse. 2024. LLM-based automatic short answer grading in undergraduate medical education. *BMC Medical Education* 24, 1 (2024), 1060.
[13] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594* (2024).
[14] Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, et al. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems* 36 (2023), 44123–44279.
[15] Yue Guo, Zian Xu, and Yi Yang. 2023. Is chatgpt a financial expert? evaluating language models on financial natural language processing. *arXiv preprint arXiv:2310.12664* (2023).
[16] Xinyi Hou, Yanjie Zhao, and Haoyu Wang. 2025. On the (In)Security of LLM App Stores . In *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 317–335. https://doi.org/10.1109/SP61157.2025.00117
[17] Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. 2024. GPTZoo: A Large-scale Dataset of GPTs for the Research Community. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering* (Sacramento, CA, USA) *(ASE '24)*. Association for Computing Machinery, New York, NY, USA, 2329–2333. https://doi.org/10.1145/3691620.3695309
[18] GPTs Hunter. 2024. GPTs Hunter. https://www.gptshunter.com/.

[19] Argyro Kavadella, Marco Antonio Dias Da Silva, Eleftherios G Kaklamanos, Vasileios Stamatopoulos, Kostis Gian-nakopoulos, et al. 2024. Evaluation of ChatGPT's real-life implementation in undergraduate dental education: mixed methods study. *JMIR Medical Education* 10, 1 (2024), e51344.

[20] Jack Krolik, Herprit Mahal, Feroz Ahmad, Gaurav Trivedi, and Bahador Saket. 2024. Towards leveraging large language models for automated medical q&a evaluation. *arXiv preprint arXiv:2409.01941* (2024).

[21] Haitao Li, Junjie Chen, Jingli Yang, Qingyao Ai, Wei Jia, Youfeng Liu, Kai Lin, Yueyue Wu, Guozhi Yuan, Yiran Hu, et al. 2024. Legalagentbench: Evaluating llm agents in legal domain. *arXiv preprint arXiv:2412.17259* (2024).

[22] Haitao Li, You Chen, Qingyao Ai, Yueyue Wu, Ruizhe Zhang, and Yiqun Liu. 2024. Lexeval: A comprehensive chinese legal benchmark for evaluating large language models. *Advances in Neural Information Processing Systems* 37 (2024), 25061–25094.

[23] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[24] M. H. Maqbool, Umar Farooq, Adib Mosharrof, A. B. Siddique, and Hassan Foroosh. 2023. MobileRec: A Large Scale Dataset for Mobile Apps Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 3007–3016. https://doi.org/10.1145/3539618.3591906

[25] Zabir Al Nazi and Wei Peng. 2024. Large language models in healthcare and medical domain: A review. In *Informatics*, Vol. 11. MDPI, 57.

[26] PR Newswire. 2024. FlowGPT, The Open Ecosystem AI Platform, Raises $10M. https://www.prnewswire.com/news-releases/flowgpt-the-open-ecosystem-ai-platform-raises-10m-302070574.html.

[27] OpenAI. 2023. Introducing GPTs. https://openai.com/blog/introducing-gpts.

[28] OpenAI. 2024. GPT Store Search Change Step Back. Available online. https://community.openai.com/t/gptstore-search-change-step-back/600782, Accessed: 2025-08-12.

[29] OpenAI. 2024. Has anyone else noticed changes in the GPTs search ranking algorithm lately? Available online. https://community.openai.com/t/has-anyone-else-noticed-changes-in-the-gpts-search-ranking-algorithm-lately/1258050, Accessed: 2025-08-12.

[30] OpenAI. 2024. Ranking of top 500 GPTs by # conversations. Available online. https://community.openai.com/t/ranking-of-top-500-gpts-by-conversations/578607, Accessed: 2025-08-12.

[31] OpenAIapts. 2023. ChatGPT. https://chatgpt.com/gpts.

[32] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[34] Quora. 2024. Poe. https://poe.com.

[35] Jaromir Savelka. 2023. Unlocking practical applications in legal domain: Evaluation of gpt for zero-shot semantic annotation of legal texts. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*. 447–451.

[36] Patrick Schober, Christa Boer, and Lothar A Schwarte. 2018. Correlation coefficients: appropriate use and interpretation. *Anesthesia & analgesia* 126, 5 (2018), 1763–1768.

[37] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696* (2020).

[38] Shreya Shankar, JD Zamfirescu-Pereira, Björn Hartmann, Aditya Parameswaran, and Ian Arawjo. 2024. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–14.

[39] SimilarWeb. 2025. Website Analysis: huggingface.co. Available online. https://pro.similarweb.com/, Accessed: 2025-08-12.

[40] SimilarWeb. 2025. Website Analysis: poe.com. Available online. https://pro.similarweb.com/, Accessed: 2025-08-12.

[41] Dongxun Su, Yanjie Zhao, Xinyi Hou, Shenao Wang, and Haoyu Wang. 2024. Gpt store mining and analysis. *arXiv preprint arXiv:2405.10210* (2024).

[42] Dongxun Su, Yanjie Zhao, Xinyi Hou, Shenao Wang, and Haoyu Wang. 2025. GPT Store Mining and Analysis.

[43] Tencent. 2024. Tencent agent shop. https://yuanqi.tencent.com/agent-shop.

[44] Jiaqi Wang, Huan Zhao, Zhenyuan Yang, Peng Shu, Junhao Chen, Haobo Sun, Ruixi Liang, Shixin Li, Pengcheng Shi, Longjun Ma, et al. 2024. Legal evalutions and challenges of large language models. *arXiv preprint arXiv:2411.10137* (2024).

[45] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding

benchmark. *Advances in Neural Information Processing Systems* 37 (2024), 95266–95290.

[46] Qiuhong Wei, Zhengxiong Yao, Ying Cui, Bo Wei, Zhezhen Jin, and Ximing Xu. 2024. Evaluation of ChatGPT-generated medical responses: a systematic review and meta-analysis. *Journal of biomedical informatics* 151 (2024), 104620.

[47] Yuhao Wu, Evin Jaff, Ke Yang, Ning Zhang, and Umar Iqbal. 2025. An In-Depth Investigation of Data Collection in LLM App Ecosystems *(WWW '25)*. Association for Computing Machinery.

[48] Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia. 2023. Evaluating reading comprehension exercises generated by LLMs: A showcase of ChatGPT in education applications. In *Proceedings of the 18th workshop on innovative use of NLP for building educational applications (BEA 2023)*. 610–625.

[49] Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, et al. 2024. Finben: A holistic financial benchmark for large language models. *Advances in Neural Information Processing Systems* 37 (2024), 95716–95743.

[50] Yinglin Xie, Xinyi Hou, Yanjie Zhao, Kai Chen, and Haoyu Wang. 2025. LLM App Squatting and Cloning. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*. 64–74.

[51] Chuan Yan, Bowei Guan, Yazhi Li, Mark Huasong Meng, Liuhuo Wan, and Guangdong Bai. 2025. Understanding and Detecting File Knowledge Leakage in GPT App Ecosystem. In *Proceedings of the ACM on Web Conference 2025* (Sydney NSW, Australia) *(WWW '25)*. Association for Computing Machinery, New York, NY, USA, 3831–3839. https://doi.org/10.1145/3696410.3714755

[52] Chuan Yan, Ruomai Ren, Mark Huasong Meng, Liuhuo Wan, Tian Yang Ooi, and Guangdong Bai. 2024. Exploring ChatGPT App Ecosystem: Distribution, Deployment and Security. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering* (Sacramento, CA, USA) *(ASE '24)*. Association for Computing Machinery, New York, NY, USA, 1370–1382. https://doi.org/10.1145/3691620.3695510

[53] Jian Yang, Jiaxi Yang, Ke Jin, Yibo Miao, Lei Zhang, Liqun Yang, Zeyu Cui, Yichang Zhang, Binyuan Hui, and Junyang Lin. 2024. Evaluating and aligning codellms on human preference. *arXiv preprint arXiv:2412.05210* (2024).

[54] Jing Yao, Xiaoyuan Yi, and Xing Xie. 2024. CLAVE: An Adaptive Framework for Evaluating Values of LLM Generated Responses. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 58868–58900. https://proceedings.neurips.cc/paper_files/paper/2024/file/6c1d2496c04d1ef648d58684b699643f-Paper-Datasets_and_Benchmarks_Track.pdf

[55] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).

[56] Zejun Zhang, Li Zhang, Xin Yuan, Anlan Zhang, Mengwei Xu, and Feng Qian. 2024. A First Look at GPT Apps: Landscape and Vulnerability. arXiv:2402.15105 [cs.CR] https://arxiv.org/abs/2402.15105

[57] Yanjie Zhao, Xinyi Hou, Shenao Wang, and Haoyu Wang. 2025. LLM App Store Analysis: A Vision and Roadmap. *ACM Trans. Softw. Eng. Methodol.* 34, 5, Article 125 (May 2025), 25 pages. https://doi.org/10.1145/3708530

[58] Qinkai Zheng, Xiao Xia, Xu Zou, Yuxiao Dong, Shan Wang, Yufei Xue, Lei Shen, Zihan Wang, Andi Wang, Yang Li, et al. 2023. Codegeex: A pre-trained model for code generation with multilingual benchmarking on humaneval-x. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5673–5684.