JRISA Journal of Research and Innovation for Societal Advancement
2026, Issue 2
https://doi.org/10.69052/jrisa.2026.1020

REVIEW ARTICLE

From Passive Mining to Proactive Matching

A Survey of AI-Driven User Need Discovery and the Case for Agent-Mediated Demand-Supply Coordination

Xinglang Wang
High Bluff Academy, 5531 Cancha de Golf, Suite 202, Rancho Santa Fe, CA 92091, United States
Correspondence: hywgmem@gmail.com

MANUSCRIPT
JRISA-2026-1020

RECEIVED
15 October 2026

ACCEPTED
28 October 2026

PUBLISHED
16 November 2026

DOI
https://doi.org/10.69052/jrisa.2026.1020

CITE AS
Wang, X. (2026). From Passive Mining to Proactive Matching: A Survey of AI-Driven User Need Discovery and the Case for Agent-Mediated Demand-Supply Coordination. *Journal of Research and Innovation for Societal Advancement*, 2026, Issue 2. https://doi.org/10.69052/jrisa.2026.1020

Open access. This article is distributed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). Copyright remains with the author.

ABSTRACT

Firms have long struggled to learn what users want before deciding what to produce. This survey reviews research published from 2024 to 2026 on using artificial intelligence (AI), in particular large language models (LLMs) and LLM-based agents, to discover user needs from user-generated content such as product reviews and social media posts. We searched Google Scholar with a bilingual English and Chinese query matrix and applied deduplication, bibliographic verification, rule-based relevance scoring and citation snowballing, which produced a verified corpus of 98 papers. Each paper was classified by technique, application domain and year. Within this corpus, LLMs are the most common technique, agent-based work is growing from a small base, and retrieval-augmented generation is rarely used. Applications are concentrated in software requirements engineering and e-commerce, while other industries remain thinly covered. We propose that this pattern reflects how much public user-written text each industry produces, and we present this as a proposition for future testing because our search design favors such industries. Reported accuracy is promising on narrow tasks such as sentiment and requirement classification, but few studies compare their systems with human analysts, so it remains open whether accuracy is sufficient. We also argue that acquiring need data deserves as much attention as analyzing it. Building on these findings, we outline a design in which agents acting for users describe their needs and coordinate with agents acting for suppliers. We close with a research agenda on data availability, data acquisition, agent protocols, privacy and incentives.

Keywords: user need discovery, requirements elicitation, large language models, demand-supply matching, agentic commerce

1 Introduction

Every firm has to decide what to make and for whom, and mistakes in this decision are costly in both directions. Overproduction leaves inventory and waste, while underproduction or the wrong product

© 2026 The Author. Published by JRISA under a CC BY 4.0 licence.

leaves demand unmet and customers lost. Much of twentieth-century management thinking can be read as a series of answers to this single question under changing technological conditions. Lean production, mass customization and personalized recommendation are examples of such answers.

These approaches share a weakness that is easy to overlook because it is so common. Information about what people want is gathered passively, with delay and with loss. Firms either ask through surveys, interviews and focus groups, which are limited by sample size and by how well respondents can articulate their wants, or they infer demand from sales data, which only shows demand that existing products already meet. Xu and Yang make this point in their study of e-commerce reviews, observing that consumer feedback contains many implicit needs that traditional surveys miss because of sample bias and expression barriers [1].

At the same time, people have spent two decades writing down what they want in product reviews, app-store feedback, forum posts and social media. This user-generated content (UGC) holds a large amount of information about needs, but its unstructured, colloquial and fragmentary form long made systematic analysis impractical. Natural language processing (NLP) was expected to close this gap, yet Norheim et al. note that NLP has long been seen as promising for requirements engineering productivity without demonstrating substantive benefits [2]. Since 2023 the situation has changed. Large language models (LLMs) can read colloquial text, classify it with little or no labeled data and produce structured output, and agent architectures built on LLMs can chain these steps into end-to-end pipelines.

If artificial intelligence (AI) can discover needs from UGC at low cost and acceptable quality, the problem of matching supply to demand changes in structure. Need discovery would move from a firm asking its customers to a system that listens to them, and possibly further to a user's own agent that speaks on the user's behalf. Research on this shift is growing quickly but is scattered across software engineering, e-commerce, marketing and information systems, and it is published in both English and Chinese. Existing reviews, which we discuss in Section 4, focus mainly on requirements engineering and app-review analysis. To our knowledge, no study has yet compared adoption across industries or asked what the results imply for the way demand and supply are coordinated.

This paper addresses that gap with a systematic survey of AI-driven user need discovery published between 2024 and 2026. We searched the English and Chinese literature with a two-axis query matrix, verified every candidate against open bibliographic databases or its own full text, and classified the resulting 98 papers by technique, application domain and year. We then use this landscape to ask why some industries adopted AI need discovery early while others lag, and where the technical bottleneck now lies as methods move from topic models and Bidirectional Encoder Representations from Transformers (BERT) to LLMs and agents.

The paper makes three contributions. First, to our knowledge it provides the first cross-industry and bilingual analysis of how AI techniques are used to discover user needs, based on a verified corpus with an auditable evidence trail. Second, it offers two propositions about this landscape. We propose that the availability of public user-expressed data, rather than industry characteristics alone, helps explain which industries led, and we show that reported accuracy is promising on narrow tasks but rarely benchmarked against human analysts. Third, it proposes agent-mediated demand-supply coordination, in which user-side and supply-side agents exchange structured needs and responses in real time, and it sets out the preconditions and open research problems for such a system.

The rest of the paper is organized as follows. Section 2 describes the survey method, and Section 3 reports the distribution of the corpus by technique, domain and year. Section 4 analyzes the demand

side and Section 5 the supply side. Section 6 presents the proposal for proactive matching, Section 7 sets out a research agenda, and Section 8 concludes.

2 Survey Method

We designed the survey so that every selection decision follows a deterministic procedure and every included paper can be traced to a verifiable bibliographic record. This design serves two purposes, namely a reproducible method and a reference list free of fabricated or misattributed entries. Figure 1 summarizes the selection pipeline and the number of papers kept at each stage. The subsections below describe each step and the limitations of the method.

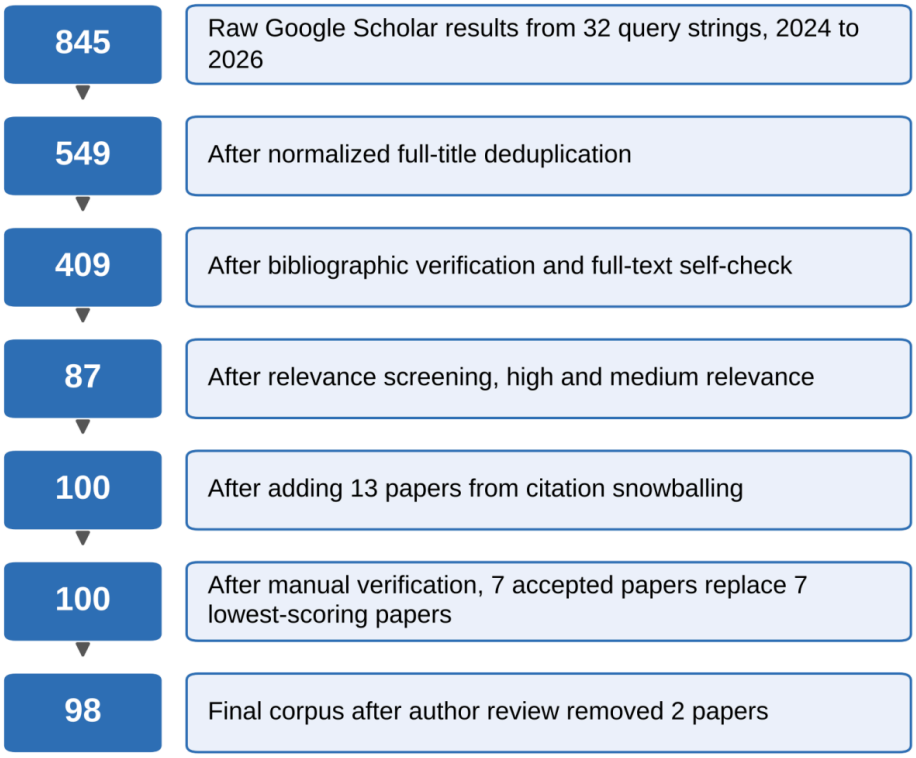


Figure 1. Selection pipeline and sample size at each stage. Each row gives the number of records that remain after the step described on its right, from 845 raw Google Scholar results to the final corpus of 98 papers. Every rule was fixed before it was applied and can be re-run.

2.1 *Query design*

The research question has two conceptual axes. The first is the AI technique, which covers LLMs, generative AI, agents, and fine-tuning or reinforcement learning from human feedback (RLHF). The second is the need-discovery task, which covers user needs, requirements elicitation, pain points and the voice of the customer. We expanded each axis into four keyword groups and crossed them to form a four-by-four matrix, shown in Table 1.

Table 1. English-language query matrix. Each cell is the conjunction of its row group, its column group, and the channel constraint.

Technique \ Need task	B1 user need / user requirement	B2 product requirement / requirement elicitation	B3 pain point / unmet need	B4 voice of customer / customer feedback analysis
A1 large language model / LLM	A1×B1	A1×B2	A1×B3	A1×B4
A2 generative AI	A2×B1	A2×B2	A2×B3	A2×B4
A3 AI agent / multi-agent / agentic	A3×B1	A3×B2	A3×B3	A3×B4
A4 RLHF / fine-tuning	A4×B1	A4×B2	A4×B3	A4×B4

Each cell's query joins three parts with the Boolean operator AND, namely the technique group of its row, the need group of its column and a channel constraint. The channel constraint requires one of the phrases "social media", "user review", "app review" or "user-generated content", which restricts results to work that discovers needs from UGC rather than from interviews or specifications. Because Google Scholar does not match across languages, we built a parallel Chinese-language matrix with equivalent terms and ran the two tracks independently, giving 32 query strings in total. All searches were restricted to publications from 2024 to 2026.

2.2 *Retrieval and deduplication*

For each English cell we retrieved the first four result pages, which hold up to 40 records, and for each Chinese cell the first three pages. This yielded 845 raw records, and the number of query cells that retrieved each paper was recorded as a rough indicator of topical centrality.

Deduplication used normalized full-title matching. Before comparison we removed the format tags that Google Scholar adds to titles, decoded HyperText Markup Language (HTML) entities, applied Unicode Normalization Form Compatibility Composition (NFKC), converted the text to lower case, and removed all punctuation and whitespace. This step left 549 unique papers.

2.3 *Bibliographic verification and full-text self-check*

To confirm that each paper exists and that its metadata are correct, we queried OpenAlex, Crossref and arXiv by title. Returned candidates were compared using the larger of two similarity scores on normalized titles, the sequence ratio computed by the Python difflib module and the token Jaccard index. Preset thresholds defined the outcome. A score of at least 0.95 with a consistent year counted as a strong match, a score of at least 0.85 as a weak match, a score between 0.60 and 0.85 was referred to a human reviewer, and any lower score counted as no match. Every application programming interface (API) request and raw response was saved as an evidence file.

This step exposed a systematic gap, because 145 Chinese-language papers had no record in any of the three databases. To close it, we extracted the first three pages of each downloaded full-text file and checked whether the text contained the paper's own title, either as an exact substring or with a token coverage of at least 0.8. We call this procedure a full-text self-check. Together the two procedures verified 409 of the 549 papers, as Figure 2 shows.

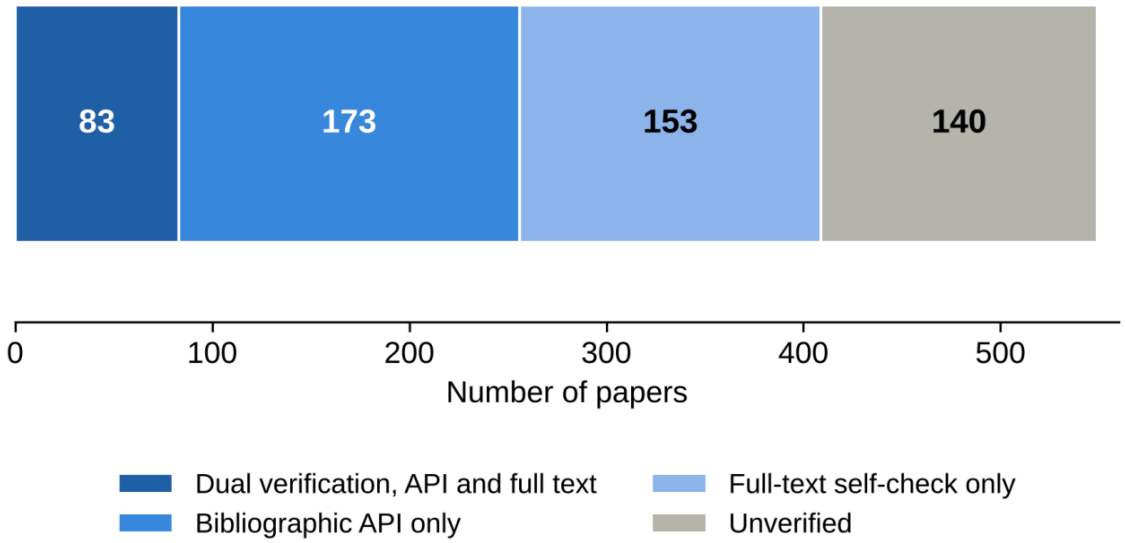


Figure 2. Verification status of the 549 deduplicated records. Dual verification means that a record has both a bibliographic API match and a full-text self-check. The self-check mainly recovers Chinese-language papers that are absent from international databases.

2.4 *Relevance screening and citation snowballing*

Verified papers were scored from 0 to 100 by a deterministic keyword rule. The rule gave the title a weight of 3 and the first three pages or the abstract a weight of 1, and it scored four dimensions, namely need discovery, AI technique, data source and product development, each with its own cap. Off-topic domain terms in the title, such as medical, educational or legal terms, incurred a penalty, and a paper could score no more than 25 unless both the need dimension and the AI dimension were non-zero. Papers scoring 60 or more were rated as high relevance, which gave 37 papers, and papers scoring from 40 to 59 as medium relevance, which gave 50, for 87 in total.

To recover papers that keyword search misses, we used 39 verified papers of high and medium relevance as seeds for citation snowballing in OpenAlex in both directions, covering their references and the works that cite them. Candidates had to be published in 2024 or later and not already be in the pool, and they were scored with the same rule. The 13 highest-scoring candidates brought the sample to 100. The author then manually verified 9 high-relevance papers that the tool could not verify automatically, accepting 7 and rejecting 2. Under a preset rank-and-replace rule, the 7 accepted papers displaced the 7 lowest-scoring papers in the sample. In a final review, the author removed 2 papers of low relevance, leaving 98.

2.5 *Classification*

Each of the 98 papers was classified by a deterministic keyword rule. Technique categories were ordered from the newest generation to the oldest, beginning with agents and followed by retrieval-augmented generation (RAG), LLMs and generative AI, fine-tuning with RLHF or prompting, BERT and other Transformer models, topic models, deep learning, and finally classical machine learning (ML) and NLP. The first category matched became the paper's primary technique, and all matched categories were recorded as multiple labels. For domain, a match in the title took precedence. If the title

matched no domain, the body had to match at least twice, and papers matching nothing were labeled general or cross-domain.

The rule was revised twice during development. Substring matching was replaced by word-boundary matching after “app” matched “application” and “rag” matched “storage”, and tolerance for plural forms was then added after word boundaries caused “large language models” to be missed. Both revisions and their effects on the counts are recorded.

2.6 *Limitations of the method*

Six limitations should be kept in mind. First, Google Scholar was the only search source. Studies indexed mainly in Scopus, Web of Science, IEEE Xplore, the ACM Digital Library or the Chinese databases CNKI, Wanfang and CQVIP may therefore be missing.

Second, every query paired a need term with an LLM-era technique term, so the corpus over-represents studies that name these techniques, and the LLM share reported in Section 3 is likely higher than in the wider literature. Every query also required a UGC channel term, which favors industries with public reviews, as Section 4.3 discusses. Technique-neutral queries, with technique classified afterwards, would reduce the first bias.

Third, retrieval sampled only the first three or four Google Scholar pages per cell, so large cells were sampled rather than exhausted. Fourth, keyword classification cannot distinguish using a technique from merely mentioning it, which affects the multi-label counts more than the primary-technique counts.

Fifth, 11 of the 98 papers were classified from the title alone because no abstract or full text could be obtained. Sixth, many Chinese-language papers come from a single open-access publisher whose review standards may differ from those of mainstream English-language venues, so we interpret the Chinese-language counts with caution.

3 The Landscape: Technique, Domain, and Time

This section describes the composition of the corpus. Of the 98 papers, 20 were published in 2024, 43 in 2025 and 33 in 2026 up to the search date, while 2 have no determinable year. Sixty-six are written in English and 32 in Chinese.

We first report the distribution of techniques, both as a single primary technique per paper and as multiple labels. We then report the distribution of application domains and its relation to publication language. These distributions form the empirical basis for the demand-side and supply-side analyses in Sections 4 and 5.

3.1 *Techniques*

By primary technique, LLMs and generative AI lead with 45 papers. They are followed by LLM agents and multi-agent systems with 12 papers, BERT and other Transformer encoders with 11, classical ML, NLP and sentiment analysis with 10, topic models with 9, deep learning with 5, fine-tuning, RLHF and prompting with 4, and RAG with 2. Because every query named an LLM-era technique, these shares describe the corpus rather than the whole field, as Section 2.6 explains. Figure 3 shows how these papers are distributed across the three years.

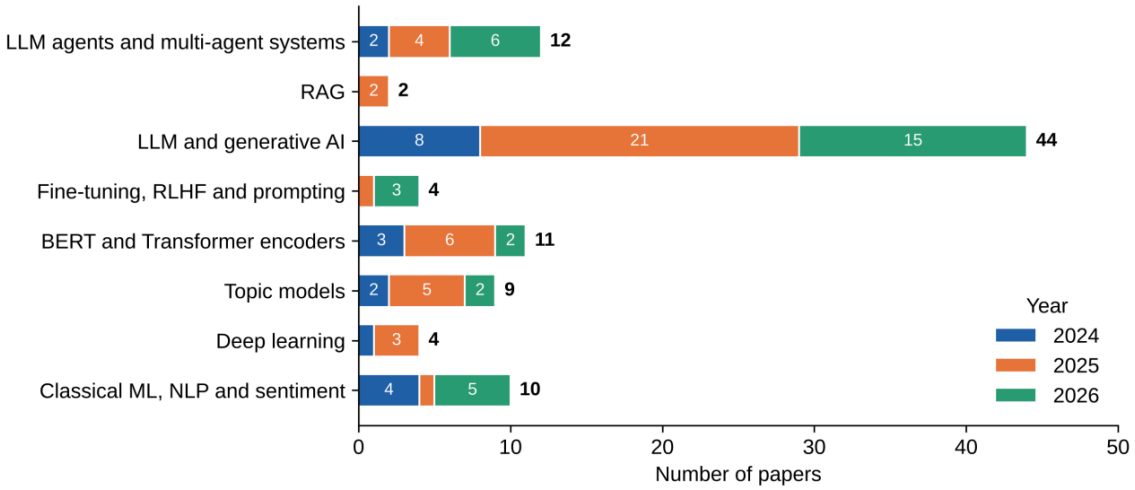


Figure 3. Primary technique by year. Each paper is assigned to the newest-generation technique it uses. The two papers with unknown year are excluded, which leaves 96 papers, so the deep learning and LLM totals are one lower than in Figure 4. Numbers inside the bars give the papers per year, and the number at the end of each bar gives the total.

Three trends are visible. LLMs were already the leading technique in 2024 with 8 papers and peaked in 2025 with 21. BERT-class encoders as a primary technique rose to 6 papers in 2025 and fell to 2 in 2026, which is consistent with their displacement by LLMs. Agent-based work grew from 2 to 4 to 6 papers across the three years, a steady increase from a small base. RAG is the primary technique of only two papers, both from 2025, and we return to this gap in Section 5.4.

The multi-label view in Figure 4 adds a further observation. Classical ML, NLP and sentiment-analysis techniques are used or mentioned in 60 papers, slightly more than the 58 papers that use or mention LLMs, so most LLM-based studies keep classical components as pipeline stages or baselines. You et al., for example, use an LLM to build a requirement taxonomy and annotate training data and then hand large-scale classification to a BERT classifier to balance capability against deployment cost [3]. Similarly, a structured-analysis method for e-commerce reviews uses BERT embeddings, principal component analysis (PCA) and k-means clustering to select representative samples that an LLM then receives as few-shot examples [4].

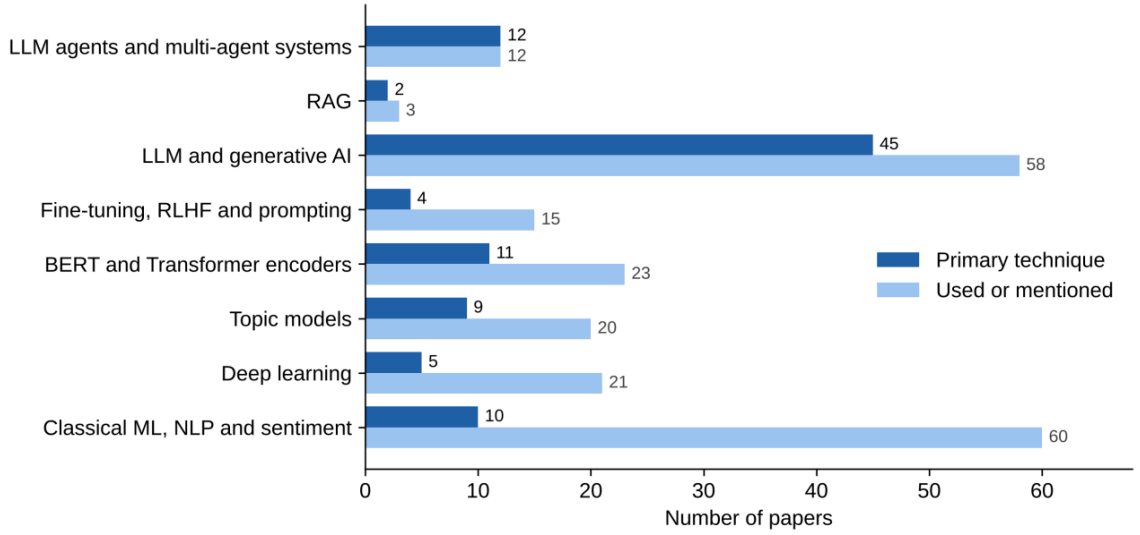


Figure 4. Primary technique compared with techniques used or mentioned. Dark bars count papers whose primary technique falls in each category, with one category per paper and 98 papers in total. Light bars count papers that use or mention the technique, with multiple labels allowed and 212 labels in total. A large gap between the two bars indicates a technique that serves mainly as a pipeline component.

3.2 *Application domains*

By primary domain, software and app requirements engineering leads with 29 papers. It is followed by general or cross-domain work with 27 papers, e-commerce and retail with 23, publishing, media and marketing with 6, hospitality and tourism with 4, and apparel and consumer goods with 3. Automotive, consumer electronics and healthcare have 2 papers each. Figure 5 shows the domain distribution by language, and Figure 6 shows how domains and techniques combine.

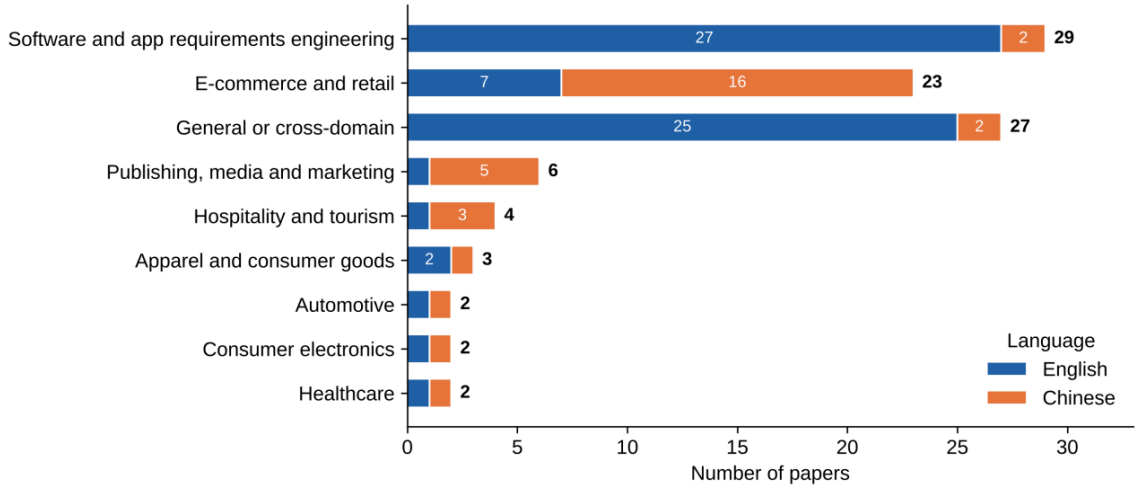


Figure 5. Application domain by publication language. Each paper is counted once under its primary domain. The general category contains papers whose title and body match no industry-specific term.

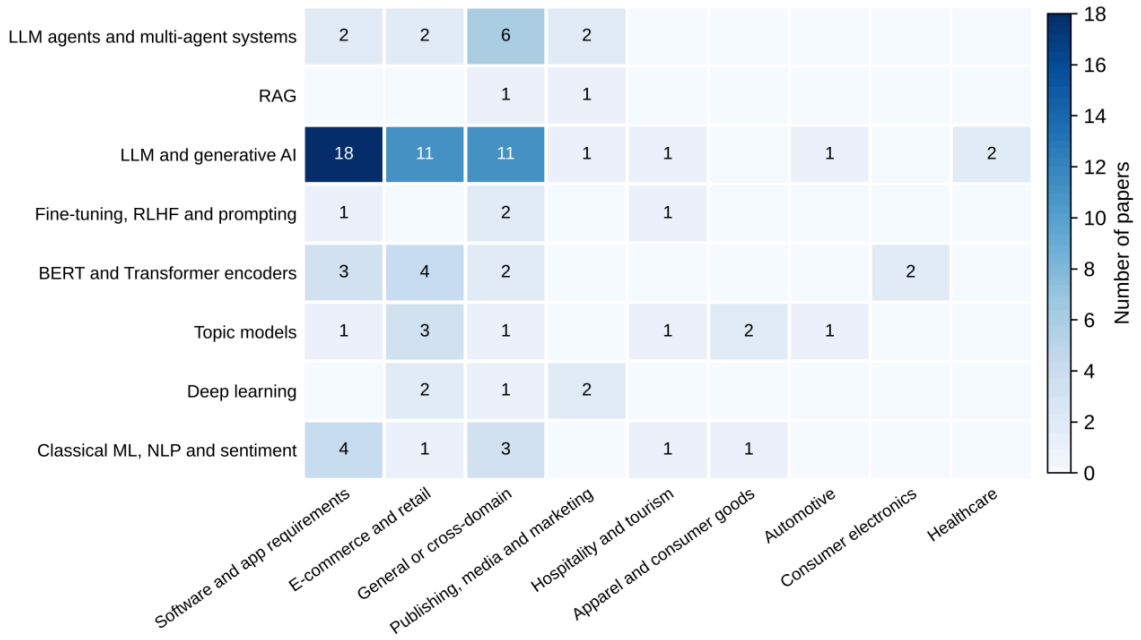


Figure 6. Primary technique by primary domain. Each cell gives the number of papers with that combination of technique and domain, and darker shading indicates more papers. LLMs lead in all three large domains, while agent-based work is concentrated in the general category and has not yet reached specific industries.

Language and domain are strongly associated, which matters for the argument in Section 4. Of the 29 software papers, 27 are in English, while 16 of the 23 e-commerce papers are in Chinese. The English-language literature thus centers on software requirements engineering, and the Chinese-language literature centers on e-commerce review mining.

4 Demand Side: Why These Industries Led

This section asks why AI need discovery has been adopted unevenly across industries. We first describe the two largest application areas, software requirements engineering and e-commerce, which together account for more than half of the corpus. We then assess the usual industry-based explanation for their lead and propose an alternative based on the availability of data, which we frame as a proposition for future testing.

4.1 *The two largest applications*

In software and app requirements engineering, app-store reviews are the main data source. Arambepola et al. trace the development of app-review analysis from conventional methods to LLMs [5], and the mapping study by Massenon et al. covers 180 primary studies and catalogs the main technique families, including topic modeling, collocation finding, association rules, aspect-based sentiment analysis and hybrid methods [6]. Recent work has moved to LLMs. Chaudhary et al. explore zero-shot classification of app reviews into functional and non-functional requirements with ChatGPT [7]. Sarhan and Alhoshan use grounded theory to derive a scheme of 14 characteristics that separates explicit, implicit and irrelevant reviews. They label 84,721 reviews to build the REV4RE dataset and report that a fine-tuned Robustly Optimized BERT Pretraining Approach (RoBERTa) model reaches a weighted F1 score of 87%, which is comparable to human annotators [8]. BERT-based

elicitation frameworks for market-driven requirements engineering have also been proposed [9]. Several systematic reviews and a meta-analysis now chart the use of LLMs across the requirements lifecycle [10][11][12].

In e-commerce, product reviews are the data source and the analysis pipeline is notably uniform. A topic model such as Latent Dirichlet Allocation (LDA) or BERTopic extracts the product attributes that users discuss, sentiment analysis measures their attitude, and a Kano model sorts the resulting needs into priority levels. Xu and Yang build a five-layer framework on 62,716 reviews, identify 36,486 implicit-need records and assign them to three Kano priority tiers [1]. A chained framework of three LLM agents extracts need statements, performs fine-grained sentiment analysis and classifies needs along several dimensions [13]. An LDA study of 11,299 reviews of a second-hand trading app identifies three need themes, namely inspection, pricing and after-sales service [14]. Further studies apply topic models, sentiment classifiers or both to reviews of hearing aids [15], apparel [16], laptops [17] and earphones on Amazon [18].

4.2 *How much do industry characteristics explain?*

The intuitive explanation rests on industry characteristics. Software has three structural advantages. It both produces and consumes AI tools, so its learning curve is the shortest. App-store reviews arrive already structured with ratings, version numbers and timestamps, and short release cycles let the value of need discovery be realized quickly. E-commerce, for its part, holds the largest body of public reviews of any industry, and each review is tied directly to a stock-keeping unit (SKU).

We consider this explanation incomplete for three reasons, all of which point toward sampling bias. First, AI need mining is at most three years old, so the current domain distribution may reflect who adopted first rather than who needs it most. Second, much of the software literature is written by requirements-engineering researchers who study software requirements as part of their disciplinary identity, which concentrates publication without implying concentrated demand. Third, as Section 3.2 showed, the software papers are almost all in English and the e-commerce papers mostly in Chinese. This split at least partly reflects the research traditions of two academic communities rather than the intensity of need in two industries.

4.3 *The data-availability proposition*

We propose an alternative explanation, stated as a proposition for future testing rather than as a finding. Adoption may depend less on an industry's intrinsic characteristics than on the availability of its need data, that is, whether a public, high-volume body of user-authored text exists that can be linked to specific products.

Our search design favors this proposition, so the corpus cannot test it. Every query required a UGC channel term such as user review or social media, as Section 2.1 describes, so industries with public reviews entered the corpus more easily, and work using interviews, sales records or support tickets was excluded. The patterns below are consistent with the proposition but cannot separate its effect from that of the search design.

The proposition is consistent with several features of the corpus. The automotive industry, and electric vehicles (EVs) in particular, combines high product complexity with highly engaged users, yet it contributes only two papers. One is a staged framework for mining requirements from vehicle reviews [3], and the other is a BERTopic study of how the needs of EV users evolve [19]. One possible reading is that the EV category is young enough that its review corpus, and the needs it would reveal,

have not yet accumulated. Consumer electronics shows the same pattern. A BERTopic analysis of smart-ring reviews finds user needs moving from basic experience toward health management and connected features [20], and its data again come from e-commerce reviews.

Finance is nearly absent from the corpus. One likely reason is that needs for financial products are expressed in business-to-business or private channels and do not appear as public UGC, so existing methods have little to mine. This case is the mirror image of the proposition, although the channel constraint in our queries would also have excluded studies of financial needs that rely on private data.

If the proposition holds, industries that have not yet adopted these methods may do so as channels for user expression expand, or as the user-side mechanism described in Section 6 takes shape, while industries with private need expression will require a different mechanism. A direct test would need a search without the channel constraint and a measure of public UGC volume by industry.

5 Supply Side: From BERT to Agents, and Where the Bottleneck Sits

This section turns from the industries that use AI need discovery to the techniques that deliver it. We trace three generations of technique in the corpus, review the accuracy that the included studies report, and argue that data acquisition deserves more attention than it receives. We also discuss why RAG is almost absent from the corpus.

5.1 *Three generations of technique*

The corpus suggests three generations of technique, the first of which consists of classical ML and topic models. LDA-family topic models and sentiment analysis based on dictionaries or classifiers were the mainstream before 2024 and remain widely used. They are interpretable and cheap, but their semantics are shallow and their topic outputs require human interpretation. One recent paper summarizes this critique, noting that keyword matching lacks semantic understanding and that topic-model outputs are hard to read [4].

The second generation consists of BERT and other pre-trained encoders. Fine-tuned BERT-class models raised classification accuracy on requirements tasks to the level of human annotators [8]. The cost is labeled data, since the REV4RE dataset behind that result required five annotators with experience in requirements or software engineering to label 84,721 reviews [8]. This labeling burden is the main reason the BERT route is hard to scale, and the corpus shows BERT declining as a primary technique in 2026.

The third generation consists of LLMs and agents. Zero-shot classification with LLMs reduces dependence on large labeled domain datasets [7], and generative output can take the form of actionable business recommendations rather than category labels [21]. On this basis, task-oriented agent architectures have begun to appear. Examples include the chained three-agent framework described above [13], the use of LLM agents by Wang et al. to extract customer needs from UGC [22], and the four-level agent architecture of Yin et al. for multi-source heterogeneous demands, whose stages handle data standardization, requirement parsing, conflict resolution and priority ranking [23]. Agents thus extend need discovery from a single classification step to a full pipeline that covers collection, cleaning, extraction, ranking and conflict resolution.

5.2 *What the reported accuracy shows*

A central question is whether current accuracy on need-discovery tasks is good enough for practical use. To answer it, we extracted the evaluation metrics reported by all 98 included studies, listed in Supplementary Table S1. Only 31 evaluate a need-discovery sub-task quantitatively on their own data, 9 evaluate other tasks, 21 apply methods without an accuracy measure, 25 are reviews or conceptual papers, and 12 could not be checked because no full text was available. Table 2 groups the 31 studies by task.

Table 2. Reported performance on need-discovery sub-tasks, grouped by task. Ranges summarize the headline metric of each study, so values are not directly comparable across studies. One study appears in two rows. Studies, datasets and values are listed individually in Supplementary Table S1.

Task	Studies	Typical data	Reported performance	Baselines	Human reference
Sentiment polarity	8	Product and app reviews	Accuracy 0.75 to 0.94	Lexicon tools or classical ML, none in four studies	Manual labels in two studies
Need or requirement classification	8	App reviews and feedback forums, 1,880 to 84,721 items	F1 or accuracy 0.78 to 0.95	Classical ML at F1 0.35 to 0.49, BERT family at accuracy 0.52 to 0.83	Annotator agreement in three studies, kappa 0.71 to 0.76
Need or requirement extraction and elicitation	9	Reviews, tweets, news, interview transcripts, simulated users	F1 0.40 to 0.92, other measures 0.58 to 0.81	Rule-based or keyword extraction, earlier tools	No comparison with professional analysts
Topic discovery and summarization	4	Reviews, social media posts	Topic coherence 0.70, BERTScore F1 0.86	LDA, BERTopic	None
Need prediction and generation	3	Forum posts, design data, industrial personas	Top-3 accuracy 0.84, expert approval 0.97	LSTM, ARIMA, ablations	Expert review in one study

These numbers do not show whether accuracy is sufficient, because sufficiency depends on the task. For sentiment and classification, we treat a system as sufficient when it matches the agreement between trained annotators on the same data. For need extraction and elicitation, the reference should be professional analysts, as in the study by Timoshenko and Hauser, who compared needs identified from UGC with those that analysts found through experiential interviews [24]. A system is sufficient there when it recovers a comparable share of analyst-identified needs at lower cost. For topic discovery and prioritization, the reference is whether analysts judge the output usable and whether the priorities agree with survey-based classification or later market outcomes.

Against these references the evidence is limited. On classification, a fine-tuned RoBERTa model reaches a weighted F1 score of 87%, which its authors describe as comparable to human annotators [8], and zero-shot ChatGPT reaches an F1 score of about 0.82 on app reviews whose annotators agreed with a Fleiss kappa of 0.76 [7]. No study compares need extraction with professional analysts. An LLM-based voice agent identified 77.5% of relevant requirements in pre-testing without a human benchmark [25], and Norheim et al. caution that the real-world success of LLMs on requirements tasks needs careful assessment [2]. We therefore conclude only that accuracy looks promising on narrow classification tasks and has not been established for implicit-need identification, conflict resolu-

tion and prioritization. The corpus also contains no direct comparison of how LLMs and agents handle sentiment.

5.3 *From analysis to acquisition*

Even where accuracy is promising, a second constraint deserves attention, namely the acquisition of data rather than its analysis. Almost every paper in the corpus assumes that the data have already been scraped, cleaned and loaded, and starts from that point. Deciding where to look, how to keep collecting and how to integrate data across platforms lies outside the research frame, although in practice it is the most labor-intensive step. Yin et al. are among the few to address it, dedicating the first agent layer of their multi-source framework to standardizing heterogeneous inputs [23]. SocialBuddy, a search agent built for social-media ecosystems, addresses the retrieval problem itself by finding relevant content in large social streams [26].

Agent capabilities are beginning to change this situation. An agent that can browse, search and call tools can in principle acquire need data continuously from many channels without a pre-assembled dataset. In the corpus, however, this capability is still at an early stage. Most of the 12 papers with agents as the primary technique divide analytical work within a pipeline rather than acquiring data from outside it. We regard proactive acquisition, together with task-specific benchmarks, as a key supply-side direction for the next phase and as a technical precondition for the proposal in Section 6.

5.4 *The absence of RAG*

RAG is the primary technique in only two papers. This is surprising, because need discovery is essentially a task of retrieving relevant passages from a large corpus and summarizing them, which is exactly the RAG paradigm. A plausible explanation is that most existing studies process a static set of reviews once, which does not call for on-demand retrieval.

Once need discovery becomes continuous and interactive, for example when a product manager asks what users have said about a given feature in the past week, RAG becomes the natural choice. AllHands, which supports natural-language question answering over large volumes of verbatim feedback, is an early step in this direction [27].

6 Toward Proactive Matching

The previous sections described what current research does. This section asks what it cannot do under its present design and proposes an alternative. We first identify the structural limits of need mining initiated by suppliers and then describe a system of user-side agents and two-way coordination. Finally, we discuss its consequences for product definition, the preconditions it requires and its implications for the market mechanism.

6.1 *The structural limits of the current paradigm*

All the work surveyed above, however sophisticated, shares one premise. Need discovery is initiated by the supplier, carried out passively and conducted on an ad hoc basis. A firm decides at some point to analyze a batch of reviews that were written in the past for reasons unrelated to that analysis.

This paradigm has three structural limits. The first is delay, because every stage between the emergence of a need and a product decision, including expression, collection, analysis and conversion, adds latency. The second is loss, because users write down only a small fraction of what they want and what they write loses its context. The third is one-way flow, because users never learn whether

their expression was heard, how it was interpreted or whether it changed anything, and so they have little reason to keep expressing their needs.

6.2 *The proposal: user-side agents and two-way coordination*

We propose a forward-looking system design. If need-mining tools reach the reference levels set out in Section 5.2, the main question would become institutional, namely whether passive and ad hoc need discovery can be turned into proactive, real-time coordination between the two sides of the market. Figure 7 contrasts the current paradigm with the proposed one.

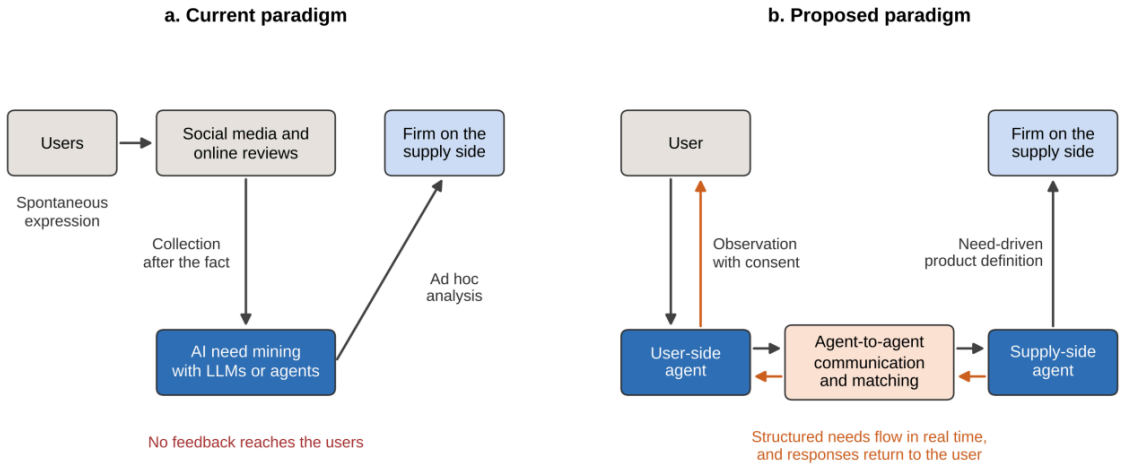


Figure 7. From passive need mining to proactive demand-supply coordination. Panel a shows the current paradigm, in which users express needs spontaneously on social media and review sites, firms collect this content after the fact and analyze it ad hoc with AI tools, and no feedback reaches the users. Panel b shows the proposed paradigm, in which a user-side agent observes the user with consent, summarizes the user's needs and matches them with supply-side agents through an agent-to-agent communication layer, while responses from the supply side flow back to the user.

The core idea is to place an agent on the user's side as well. The user-side agent continuously observes the user's behavior and expression and generates structured descriptions of needs. With the user's consent, it sends these descriptions through an agent-to-agent channel to supply-side agents, which match them in real time. In this arrangement, needs are stated by users rather than mined by firms, and suppliers respond to them instead of guessing.

Several papers in the corpus already point in this direction. Nakata et al. use a virtual agent for interactive needs extraction in personalized service development aimed at Society 5.0 [28]. In AREAs-Lab, an assistant asks clarifying questions over several rounds to uncover a user's latent intent [29]. Voice-based elicitation agents have been prototyped and tested [25], and multi-agent architectures have been applied to requirements refinement [30].

Personal agents recently launched by large firms show how the user-side half of such a system might look. Box 1 summarizes one example, which also illustrates the access problems discussed in Section 6.4.

Box 1. Illustrative case: Meta Muse

In September 2026 Meta announced Muse, a personal AI agent that can send email, book travel, fill out forms and make purchases on a user's behalf, and that remembers what matters to a user in order to make suggestions unprompted [31]. According to Meta, each user's agent runs on a dedicated virtual machine, and a separate component called Sentinel is the only authority that can permit connector actions and network traffic [32]. Meta also states that credentials shared with Muse go into secure storage, so that the agent can use them without seeing them [31].

Within weeks, Amazon blocked Muse from its store. According to press reports, an Amazon spokesperson said that third-party agents shopping for customers should operate openly, and Amazon alleged that Meta had not informed it that Muse would access the store, that the agent did not identify itself when browsing, and that it appeared to capture and store customer credentials [33][34]. These claims are Amazon's allegations. The Register reported Meta's position that the Muse model cannot see users' passwords or payment methods [34]. The case concerns a single product launch and is not evidence about need discovery, but it illustrates both the user-side capability described in Section 6.2 and the access question raised in Section 6.4.

6.3 *Consequences for product definition*

If this mechanism works, the order of product definition is reversed. Today a firm defines a product, launches it, collects feedback and iterates. Under the proposed mechanism, needs are expressed and aggregated in structured form before the product is defined, so the firm can see the distribution, intensity and conflicts of demand at the design stage, much as in a product forum that runs automatically and at scale.

A direct corollary is that, in the age of agents, recording and expressing one's own needs acquire economic value. A user-side agent can summarize only what it can observe. Users who record and express their needs will therefore see them reach suppliers earlier and more accurately than users who do not.

6.4 *Preconditions for each layer*

For the mechanism to work, each layer of the system must meet certain conditions. On the user side, users need a trustworthy personal agent under their own control that can observe and summarize continuously while protecting privacy. Privacy and data sovereignty are preconditions rather than optional features, since users must decide which needs are expressed, at what level of detail and to whom. The design described in Box 1, which isolates the agent and its data and lets a separate component control outbound actions, is one early attempt to meet this condition. Whether users will trust such designs is not yet known.

The communication layer requires agents from different vendors to interoperate, which in turn requires protocols and standards for agent-to-agent communication. A central design question is whether a centralized platform should broker demand and supply. A platform is efficient but concentrates data and invites monopoly, while peer-to-peer protocols respect data sovereignty better but raise coordination costs. We do not prejudge the answer, but the dispute described in Box 1 shows what is at stake. The outcome will influence whether agentic commerce runs on open protocols or on bilateral permissions, and therefore whether the user-side agent we describe can reach suppliers at all.

On the supply side, firms must rebuild their product-definition processes to accept real-time, structured input on needs. This involves aggregating needs and resolving conflicts among them, for

which the corpus offers early work [23], generating design candidates automatically from needs, as has been attempted for textile product design [35], and linking need priorities to production planning.

Trust and incentives complete the list. Users will keep expressing needs only if they see evidence that they were heard and that something changed, so feedback loops and incentive design are part of the mechanism itself.

6.5 *Implications for the market mechanism*

Hayek argued that the price system should be understood "as a mechanism for communicating information," whose chief virtue is "the economy of knowledge with which it operates, or how little the individual participants need to know in order to be able to take the right action" [36]. That mechanism remains indispensable. The price signal, however, arrives after the fact, is aggregated and has few dimensions. It reports what was bought at what price, not what was wanted and why.

We suggest that AI-mediated expression of needs could add a second information channel alongside the price system. This channel would operate before purchase, at the level of the individual and with many dimensions. It would not replace the market but would increase the information the market can carry. This is a conjecture whose realization depends on technical and institutional conditions being met together, and we present it as a research agenda rather than a conclusion.

7 Research Agenda and Open Problems

The analyses and the proposal above leave a number of questions open. We group them into four areas, namely the demand side, the supply side, system and institutional design, and survey methodology.

On the demand side, the data-availability proposition needs direct empirical testing. A key test is whether the availability of public UGC predicts the adoption of AI need discovery once industry characteristics are controlled for. Such a test should draw on a search without a UGC channel constraint, so that industries without public reviews can enter the sample. For industries in which need expression is inherently private, such as finance and enterprise services, alternative discovery mechanisms remain to be identified.

On the supply side, the accuracy of LLMs and agents on each need-discovery sub-task has not been benchmarked against human analysts, and implicit-need identification and sentiment sensitivity in particular lack systematic benchmarks. Shared datasets with analyst-identified needs, in the spirit of Timoshenko and Hauser [24], would make the references in Section 5.2 testable. Proactive acquisition agents, which decide for themselves where to look, what to look for and when to refresh, have yet to be designed and evaluated. The role of RAG in continuous, interactive need discovery also deserves study.

At the level of systems and institutions, it is not yet known how a user-side agent can reliably generate structured needs from behavior and expression while respecting privacy. A first step is a formal specification of the need schema, the matching problem and the agent protocol, followed by agent-based simulation that reports need recall, match quality, latency, privacy leakage and robustness to manipulation. Whether agent-to-agent communication should run through a centralized platform or a decentralized protocol is also open, as is the question of who sets the terms of access. Further work

is needed on incentives that let users see a return on expressing needs, and on how firms must change their product-definition processes to absorb real-time input.

For survey methodology, the deterministic verification and classification tools used here can be applied to other fields, but their keyword rules must be redesigned for each domain. Future surveys should also search several bibliographic databases with technique-neutral queries, and raising semantic accuracy without sacrificing auditability remains an open problem.

8 Conclusion

We surveyed AI-driven user need discovery from 2024 to 2026 with a reproducible and auditable method and classified 98 core papers by technique, domain and year. Within this corpus, LLMs are the most common technique, agent-based work is growing from a small base, and RAG is rarely used. Applications are concentrated in software requirements engineering and e-commerce, while vertical industries remain thinly covered.

From this landscape we draw three arguments, each of which needs further testing. On the demand side, we propose that the availability of public user-expressed data helps explain which industries adopted these methods first, although our search design favors this pattern. On the supply side, reported accuracy is promising on narrow classification tasks, but few studies compare their systems with human analysts, so task-specific benchmarks are needed before accuracy can be called sufficient, and data acquisition deserves more attention. At the system level, we outline how user-side agents could express needs and coordinate with supply-side agents in real time, turning passive mining into proactive matching. If realized, this would add an individual, high-dimensional information channel that operates before purchase alongside the price system. Recent personal-agent products and the first disputes over their access to commerce platforms show that parts of this design are being built, but whether the full system will emerge, and under which rules, remains open.

STATEMENTS AND DECLARATIONS

Data Availability Statement. No new primary data were generated for this article. The evaluation metrics extracted from the included studies are provided as Supplementary Table S1. The bibliographic records, relevance scores and classification results that underlie the survey are available from the author on reasonable request.

Declaration of Generative AI Use. During preparation of this manuscript, the author used Claude to help develop the software tools used in the survey for literature search, citation verification, full-text download and figure generation, to assist with extracting reported metrics from the included studies, and to assist with language polishing and with translating Chinese-language literature. The author reviewed and verified the outputs of these tools and the resulting material, and takes full responsibility for the content of the manuscript.

Funding. This research received no external funding.

Conflict of Interest. The author declares no conflict of interest.

REFERENCES

[1] W. Xu and J. Yang. 2026. Intelligent Identification of Consumers' Implicit Needs and Quality Adaptation Strategies Based on Text Mining. Zenodo (preprint). https://doi.org/10.5281/zenodo.22055016

[2] J. J. Norheim, E. Rebentisch, D. Xiao, L. Draeger, A. Kerbrat, and O. L. de Weck. 2024. Challenges in applying large language models to requirements engineering tasks. Design Science 10, e16. https://doi.org/10.1017/dsj.2024.8

[3] Z. You, S. Peng, W. Zhang, and H. Tan. 2026. A Staged Framework for Mining User Requirements from Vehicle Online Reviews Based on Large Language Models and Sentiment Weighting. Systems 14, 6, 645. https://doi.org/10.3390/systems14060645

[4] D. Xiong. 2026. Research on Structured Analysis Method of E-Commerce User Reviews Based on Large Language Model [in Chinese: 基于大语言模型的电商用户评论结构化分析方法研究]. E-Commerce Letters 15, 3, 970–979. https://doi.org/10.12677/ecl.2026.153357

[5] N. Arambepola, W. L. Wimalasena, and L. Munasinghe. 2025. From Conventional Methods to Large Language Models: A Systematic Review of Techniques in Mobile App Review Analysis. Interdisciplinary Journal of Information, Knowledge, and Management 20, 016. https://doi.org/10.28945/5491

[6] R. Massenon, I. Gambo, R. O. Ogundokun, E. A. Ogundepo, S. Srivastava, S. Agarwal, et al. 2024. Mobile app review analysis for crowdsourcing of software requirements: a mapping study of automated and semi-automated tools. PeerJ Computer Science 10, e2401. https://doi.org/10.7717/peerj-cs.2401

[7] M. Chaudhary, C. Jain, and P. R. Anish. 2025. Exploring Zero-Shot App Review Classification with ChatGPT: Challenges and Potential. In Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering (EASE '25). ACM, 672–677. https://doi.org/10.1145/3756681.3757036

[8] M. A. Sarhan and W. Alhoshan. 2026. Eliciting Explicit and Implicit Requirements from App Reviews via Grounded Theory and BERT-Based Classification. Applied Sciences 16, 16, 7997. https://doi.org/10.3390/app16167997

[9] F. A. Mihany, G. H. Galal-Edeen, E. E. Hassanein, and H. Moussa. 2025. Data-Driven Requirements Elicitation from App Reviews Framework Based on BERT. Applied Sciences 15, 17, 9709. https://doi.org/10.3390/app15179709

[10] H. Cheng, J. H. Husen, Y. Lu, T. Racharak, N. Yoshioka, N. Ubayashi, et al. 2026. Generative AI for Requirements Engineering: A Systematic Literature Review. Software: Practice and Experience 56, 2, 141–170. https://doi.org/10.1002/spe.70029

[11] A. Hemmat, M. Sharbaf, S. Kolahdouz-Rahimi, K. Lano, and S. Y. Tehrani. 2025. Research directions for using LLM in software requirement engineering: a systematic review. Frontiers in Computer Science 7, 1519437. https://doi.org/10.3389/fcomp.2025.1519437

[12] C. Ciano, M. Pelosi, A. Del Prete, and F. Amato. 2026. LLMs for Requirements Engineering: A Meta-Analysis on Trends and Challenges. In Advanced Information Networking and Applications (AINA 2026). Lecture Notes on Data Engineering and Communications Technologies. Springer, Cham, 151–161. https://doi.org/10.1007/978-3-032-23335-6_14

[13] D. Zhou. 2025. Research on Demand Extraction and Sentiment Analysis from User Reviews on E-Commerce Platforms Based on Large Language Models [in Chinese: 基于大模型的电商平台用户评论需求获取与情感分析研究]. E-Commerce Letters 14, 12, 5450–5458. https://doi.org/10.12677/ecl.2025.14124510

[14] Y. Ma. 2025. User Demand Mining for Second-Hand Trading E-Commerce Platform APP Based on LDA Model [in Chinese: 基于LDA模型的二手交易电商平台APP用户需求挖掘]. E-Commerce Letters 14, 12, 1062–1071. https://doi.org/10.12677/ecl.2025.14123960

[15] X. Li, E. Luo, and Z. Jia. 2026. Research on User Demand Mining Based on E-Commerce Reviews: Taking Hearing Aids as an Example [in Chinese: 基于电商评论的用户需求挖掘研究：以助听器为例]. E-Commerce Letters 15, 2, 338–347. https://doi.org/10.12677/ecl.2026.152164

[16] Z. Li and Y. Chen. 2024. Research on Consumer Online Reviews Based on BERT-LDA Model [in Chinese: 基于BERT-LDA模型的消费者在线评论研究]. E-Commerce Letters 13, 3, 6385–6392. https://doi.org/10.12677/ecl.2024.133787

[17] X. Dou. 2024. Text Mining Analysis of Online Reviews of Commodities Based on LDA Topic Model [in Chinese: 基于LDA主题模型的商品在线评论文本挖掘分析]. E-Commerce Letters 13, 3, 8710–8718. https://doi.org/10.12677/ecl.2024.1331066

[18] U. Maidar, M. Ra, and D. Yoo. 2024. A Cross-Product Analysis of Earphone Reviews Using Contextual Topic Modeling and Association Rule Mining. Journal of Theoretical and Applied Electronic Commerce Research 19, 4, 3498–3519. https://doi.org/10.3390/jtaer19040170

[19] F. Yue, Q. Li, C. Pan, W. Dai, and W. Lu. 2025. Research on User Demand Characteristics and Evolution of New Energy Vehicles: Topic Mining Based on the BERTopic Model [in Chinese: 新能源汽车用户需求特征与演化：基于BERTopic模型的主题挖掘]. Science Focus 20, 5, 58–74. https://doi.org/10.15978/j.cnki.1673-5668.20250402

[20] W. Qiang, X. Luo, and X. Zhang. 2025. Research on Product Requirement Evolution and Function Mapping Based on BERTopic Model [in Chinese: 基于BERTopic模型的产品需求演化与功能映射应用研究]. Frontiers of Data & Computing 7, 5, 198–211. https://doi.org/10.11871/jfdc.issn.2096-742X.2025.05.016

[21] S. Krothapalli, K. S. Bhandari, T. K. Das, P. Kumar, N. Suravarpu, and P. Narang. 2025. ReviewSense: Transforming Customer Review Dynamics into Actionable Business Insights. arXiv preprint arXiv:2510.16466. https://doi.org/10.48550/arXiv.2510.16466

[22] C. Wang, X. Jiang, Q. Li, Z. Hu, and J. Lin. 2024. Leveraging LLM Agents to Extract Customer Needs from User-Generated Content. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4973155

[23] D. Yin, Z. Cui, and L. Luan. 2025. LLM+Agent-Based Requirement Engineering for Multi-Source and Heterogeneous User Demands. In Electronics, Communications and Networks: Proceedings of the 15th International Conference (CECNet 2025). Frontiers in Artificial Intelligence and Applications 415. IOS Press. https://doi.org/10.3233/FAIA251514

[24] A. Timoshenko and J. R. Hauser. 2019. Identifying Customer Needs from User-Generated Content. Marketing Science 38, 1, 1–20. https://doi.org/10.1287/mksc.2018.1123

[25] O. Weerakoon, T. Mäkilä, E. Kaila, and S. Oyedeji. 2026. Building LLM-Based Voice Agents for Requirements Elicitation: An Experience Report on Early Prototypes. In Proceedings of the 2026 International Workshop on Agentic Engineering (AGENT '26). ACM, 173–180. https://doi.org/10.1145/3786167.3788420

[26] M. Li, Y. Mao, F. Zhang, H. Yao, R. Hu, and W. Que. 2026. SocialBuddy: Tailoring Search Agent for Social Scenarios. arXiv preprint arXiv:2609.01641. https://doi.org/10.48550/arXiv.2609.01641

[27] C. Zhang, Z. Ma, Y. Wu, S. He, S. Qin, M. Ma, et al. 2025. AllHands: Ask Me Anything on Large-scale Verbatim Feedback via Large Language Models. In 2025 IEEE 41st International Conference on Data Engineering (ICDE). IEEE, 43–57. https://doi.org/10.1109/ICDE65448.2025.00011

[28] T. Nakata, S. Chen, S. Saiki, and M. Nakamura. 2025. Enhancing Personalized Service Development with Virtual Agents and Upcycling Techniques. International Journal of Networked and Distributed Computing 13, 1, 5. https://doi.org/10.1007/s44227-024-00043-y

[29] P. Cai, Z. Zhang, T. Jin, C. Zhu, K. Chawla, S. Cho, et al. 2026. AREAs-Lab: An Interactive Environment for AI-driven Requirement Elicitation for AI Systems. arXiv preprint arXiv:2608.28979. https://doi.org/10.48550/arXiv.2608.28979

[30] K. Gallaba, A. Arabat, D. Lin, M. Sayagh, and A. E. Hassan. 2025. Towards Conversational Development Environments: Using Theory-of-Mind and Multi-Agent Architectures for Requirements Refinement. arXiv preprint arXiv:2505.20973. https://doi.org/10.48550/arXiv.2505.20973

[31] Meta. 2026. Introducing Muse: The World's First Personal AI Agent Built for Everyone. Meta Newsroom, 8 September 2026. https://about.fb.com/news/2026/09/introducing-muse-personal-ai-agent/

[32] T. Sheasha. 2026. How We Built Safety Into Muse. Meta AI Blog, 8 September 2026. https://research.meta.ai/blog/security-and-safety-for-ai-agents-our-approach-with-muse

[33] T. Bishop. 2026. Amazon Blocks Meta's Muse AI Assistant in New Standoff over Agentic Shopping. GeekWire, 20 September 2026. https://www.geekwire.com/2026/amazon-blocks-metas-muse-ai-assistant-in-new-standoff/

[34] C. Page. 2026. Amazon Shows Meta's Muse AI Shopping Agent the Door. The Register, 21 September 2026. https://www.theregister.com/ai-and-ml/2026/09/21/amazon-shows-metas-muse-ai-shopping-agent-the-door/5297777

[35] N. Zhu. 2026. User Demand Prediction and Solution Generation in Textile Product Development and Design Based on Machine Learning. Advanced Electromagnetics 15, 3, 2851–2864. https://doi.org/10.7716/aem.v15i3.3343

[36] F. A. Hayek. 1945. The Use of Knowledge in Society. The American Economic Review 35, 4, 519–530. https://www.jstor.org/stable/1809376