Language Technologies Institute

SIGIR2026
MELBOURNE | NAARM
AUSTRALIA

Agent Search and Marketplace Evaluation

To Eun Kim

Carnegie Mellon University

SIGIR 2026
Invited talk @ AgentSearch Workshop

Agent Search with the lens of Distributed Systems

- Agent Search is a new problem — but not an unprecedented one.
- Recurring question:

Given a task, find the systems best able to accomplish it

- Predecessors:
  - Distributed IR and AI
    - Route a query to the right collection
    - Heterogeneous agents can advertise and discover (e.g., Blackboard architecture), and coordinate capabilities (MAS)
    - Agent Orchestration

Existing applications

- Agent Search already happens in practice — as routing & selection over competing systems

1. LLM Routing (user → LLM)

ROUTELLM: LEARNING TO ROUTE LLMS WITH PREFERENCE DATA

Isaac Ong[*1] **Amjad Almahairi**[*2] **Vincent Wu**[1] **Wei-Lin Chiang**[1] **Tianhao Wu**[1]
Joseph E. Gonzalez[1] **M Waleed Kadous**[3] **Ion Stoica**[1,2]
[1]UC Berkeley [2]Anyscale [3]Canva

Existing applications

- Agent Search already happens in practice — as routing & selection over competing systems

1. LLM Routing (user → LLM)
2. Retriever Routing (LLM → Retriever; in RAG)

LTRR: Learning To Rank Retrievers for LLMs

To Eun Kim
Carnegie Mellon University
Pittsburgh, PA, USA
toeunk@cs.cmu.edu

Fernando Diaz
Carnegie Mellon University
Pittsburgh, PA, USA
diazf@acm.org

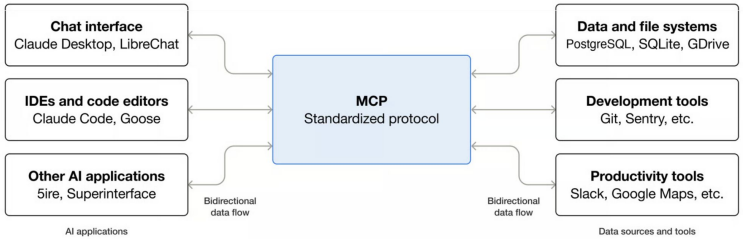
MoR: Better Handling Diverse Queries with a Mixture of Sparse, Dense, and Human Retrievers

Jushaan Kalra[†1*] **Xinran Zhao**[†1] **To Eun Kim**[1] **Fengyu Cai**[2]
Fernando Diaz[1] **Tongshuang Wu**[1]
[1]Carnegie Mellon University, [2]Technical University of Darmstadt

Existing applications

- Agent Search already happens in practice — as routing & selection over competing systems

1. LLM Routing (user → LLM)
2. Retriever Routing (LLM → Retriever; in RAG)
3. Tool / Skill / Agent Selection (task → executable system)



https://modelcontextprotocol.io/docs/getting-started/intro
https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/

Agents live in marketplaces

- Modern agents — LLMs, retrievers, tools — are orchestrated to deliver generated information objects through conversational interfaces.
- Agent–access is increasingly mediated by marketplaces.
- Platforms make it trivial for users and agents to try many providers, *switching* when a (perceived) better options appears.
- Competition between systems is therefore inherent to deployment.

Agents now live in marketplaces — distributed, discoverable, and competing to be selected.
Our evaluation should reflect that.

Language Technologies Institute

SIGIR2026
MELBOURNE | NAARM
AUSTRALIA

Evaluation of Agents under Simulated AI Marketplace Dynamics

To Eun Kim
Carnegie Mellon University

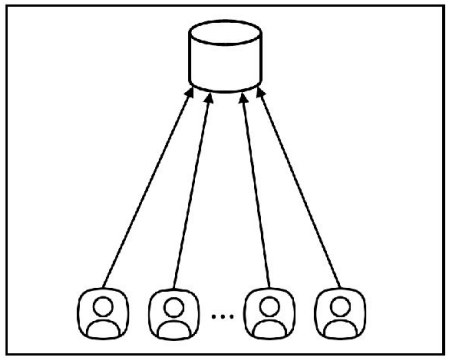
Alireza Salemi
University of Massachusetts Amherst

Hamed Zamani
University of Massachusetts Amherst

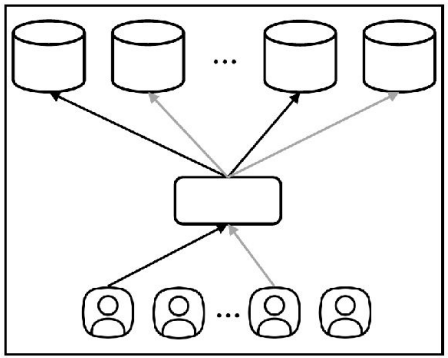
Fernando Diaz
Carnegie Mellon University

SIGIR 2026 Perspectives

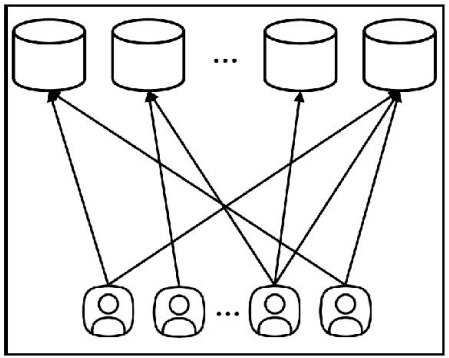
Evaluation Paradigms



Cranfield



Arena



Marketplace

new!

- **Cranfield**: multi-user, single system.
  - All queries routed to a fixed system. *No competition, no user choice.*
- **Arena**: multi-user, multi-system.
  - A platform mediates anonymous pairwise comparisons. There is a certain-level of competition, but *users cannot form persistent preference or switching* (Arena platform decides).
- **Marketplace (ours)**: multi-user, multi-system.
  - *Users actively select* among competing providers, with no central mediator required — enabling competitive dynamics, behavioral adaptation, and market share over time.

Why competitive dynamics matter

- Network effects
  - a retrieval system's value depends on its adoption by the generator population and its differentiation from rivals, not just intrinsic quality.
- Winner-take-all
  - small performance differences can produce dramatic market-share disparities.
- Portfolio & path dependence
  - agents diversify across providers; early interactions alter future routing and resource allocation.
- Therefore, a deployment success is not a single offline number
  - it shows up as query traffic, market share, and sustained usage.

Motivating study (Cranfield model)

- Seven systems: Qwen3, Kimi K2.5, Llama 3.3, DeepSeek V3.2, Grok 4.1, Gemini 2.5, GPT-OSS.
- 500 factoid questions from SimpleQA; agents produce generated information objects (not short phrases).
- Score each answer for correctness and rank by F1.

Model	F1
Qwen 3	60.24
Kimi K2.5	42.51
Llama 3.3	27.74
DeepSeek V3.2	27.59
Grok 4.1	19.21
Gemini 2.5	16.83
GPT-OSS	14.42

F1 scores on SimpleQA Benchmark (Cranfield Evaluation)

Motivating study (Cranfield model)

- Fair share: market share you would expect — proportional to Cranfield benchmark capability (F1)
- Market Concentration: measure of skewness of share
  - Herfindahl–Hirschman Index (HHI): Square of ℓ_2 norm of share
  - Effective number of Firms (EF): Reciprocal of HHI

Model	F1	Fair Share (%)	Fair Share (%) w/o Qwen3	Fair Share (%) w/o DeepSeek V3.2
Qwen 3	60.24	28.89	–	33.30
Kimi K2.5	42.51	20.38	28.66	23.49
Llama 3.3	27.74	13.30	18.71	15.33
DeepSeek V3.2	27.59	13.23	18.60	–
Grok 4.1	19.21	9.21	12.95	10.62
Gemini 2.5	16.83	8.07	11.35	9.30
GPT-OSS	14.42	6.91	9.72	7.97
HHI	–	0.18	0.19	0.22
EF	–	5.56	5.24	4.63

Lightly concentrated market

Heavily concentrated market

Motivating study (Marketplace Simulation)

Model	Fair Share (%) w/o Qwen3	Fair Share (%) w/o DeepSeek V3.2
Qwen 3	–	33.30
Kimi K2.5	28.66	23.49
Llama 3.3	18.71	15.33
DeepSeek V3.2	18.60	–
Grok 4.1	12.95	10.62
Gemini 2.5	11.35	9.30
GPT-OSS	9.72	7.97
HHI	0.19	0.22
EF	5.24	4.63

Fair shares of **Lightly** and **Heavily** concentrated market based on **Cranfield**

Motivating study (Marketplace Simulation)

Simulation Procedure

- Assume a fixed population of users, each with a preference distribution over systems
- Initialize uniform preference over systems
- Simulation steps (loop)
  - Sample k users
  - For each user
    - Sample a question
    - Select a system based on their pref. distribution
    - Observe the response from the selected system
    - Update pref. distribution based on the observed correctness

Model	Fair Share (%) w/o Qwen3	Fair Share (%) w/o DeepSeek V3.2
Qwen 3	–	33.30
Kimi K2.5	28.66	23.49
Llama 3.3	18.71	15.33
DeepSeek V3.2	18.60	–
Grok 4.1	12.95	10.62
Gemini 2.5	11.35	9.30
GPT-OSS	9.72	7.97
HHI	0.19	0.22
EF	5.24	4.63

Fair shares of **Lightly** and **Heavily** concentrated market based on **Cranfield**

Motivating study (Marketplace Simulation)

Simulation Procedure

- Assume a fixed population of users, each with a preference distribution over systems
- Initialize uniform preference over systems
- Simulation steps (loop)
  - Sample k users
  - For each user
    - Sample a question
    - Select a system based on their pref. distribution
    - Observe the response from the selected system
    - Update pref. distribution based on the observed correctness

1. *System ranking changes*
2. *Market shares are not proportional to benchmark scores*

Information that Cranfield CANNOT give you

Model	Fair Share (%) w/o Qwen3	Fair Share (%) w/o DeepSeek V3.2
Qwen 3	–	33.30
Kimi K2.5	28.66	23.49
Llama 3.3	18.71	15.33
DeepSeek V3.2	18.60	–
Grok 4.1	12.95	10.62
Gemini 2.5	11.35	9.30
GPT-OSS	9.72	7.97
HHI	0.19	0.22
EF	5.24	4.63

Model	Share	ΔFS
Kimi K2.5	29.40	(+0.74)
DeepSeek V3.2	21.00	(+2.4)
Llama 3.3	14.40	(-4.31)
Grok 4.1	13.80	(+0.85)
GPT-OSS	12.80	(+3.08)
Gemini 2.5	8.60	(-2.75)
HHI	0.19	0
EF	5.15	(-0.09)

Model	Share	ΔFS
Qwen3	51.60	(+22.71)
Kimi K2.5	14.80	(-5.58)
Llama 3.3	10.60	(-2.70)
Grok 4.1	9.60	(+0.39)
GPT-OSS	7.40	(+0.49)
Gemini 2.5	6.00	(-2.07)
HHI	0.32	(+0.10)
EF	3.15	(-1.49)

10 users, **100 simulation steps**,
5 users per step, 500 test queries in total
ΔFS: Fair share difference

Motivating study (Marketplace Simulation - Market Entry)

- We already pre-warmed the simulation with 6 models (t=1 - 100)
- Market Entry:
  From t=101, we introduce a 7th model, and let the simulation run (t=101 - 200) with the same set of questions.

- Two contrasting entry scenarios:
  - a strong model (Qwen3) enters a **lightly** concentrated market;
  - an average model (DeepSeek V3.2) enters a **highly** concentrated market.

Motivating study (Marketplace Simulation - Market Entry)

Qwen3 late entry.

Market Share (ΔFS) (%)	System Ranking	→	System Ranking	Market Share (%) (ΔFS)
Before Entry HHI=0.19 $t = (1 - 100)$			After Entry HHI=0.24 $t = (101 - 200)$	
		new	Qwen3	36.00 (+7.11)
(+0.74) 29.40	Kimi K2.5		Kimi K2.5	29.40 (+8.62)
(+2.4) 21.00	DeepSeek V3.2		DeepSeek V3.2	11.60 (-1.63)
(-4.31) 14.40	Llama 3.3		Gemini 2.5	6.60 (-1.47)
(+0.85) 13.80	Grok 4.1		GPT-OSS	6.40 (-0.51)
(+3.08) 12.80	GPT-OSS		Llama 3.3	5.60 (-7.7)
(-2.75) 8.60	Gemini 2.5		Grok 4.1	4.40 (-4.81)

DeepSeek V3.2 late entry.

Market Share (ΔFS) (%)	System Ranking	→	System Ranking	Market Share (%) (ΔFS)
Before Entry HHI=0.32 $t = (1 - 100)$			After Entry HHI=0.38 $t = (101 - 200)$	
(+22.71) 51.60	Qwen3		Qwen3	57.80 (+28.91)
(-5.58) 14.80	Kimi K2.5		Kimi K2.5	15.80 (-4.58)
		new	DeepSeek V3.2	12.0 (-1.23)
(-2.70) 10.60	Llama 3.3		Llama 3.3	4.60 (-8.70)
(+0.39) 9.60	Grok 4.1		GPT-OSS	4.20 (-2.71)
(+0.49) 7.40	GPT-OSS		Gemini 2.5	3.20 (-4.87)
(-2.07) 6.00	Gemini 2.5		Grok 4.1	2.40 (-6.81)

Realized shares ≠ expected benchmark "fair" shares.

- A strong model entering a lightly concentrated market is a shock to a market and reshuffles everyone
- An average model entering a highly concentrated market barely moves the market

Motivating study (Marketplace Simulation - Market Entry)

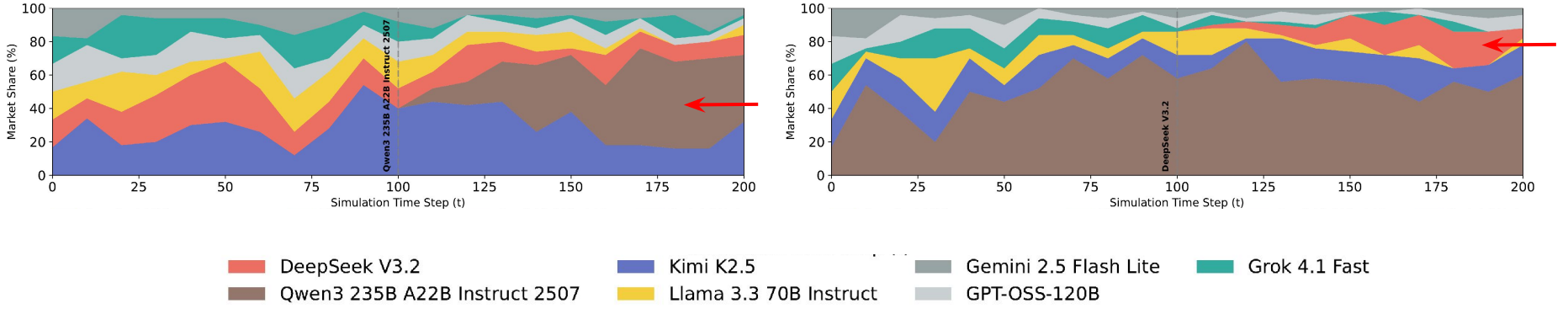
Qwen3 late entry.

Market Share (ΔFS) (%)	System Ranking	→	System Ranking	Market Share (%) (ΔFS)
Before Entry HHI=0.19 $t = (1 - 100)$			After Entry HHI=0.24 $t = (101 - 200)$	
		new	Qwen3	36.00 (+7.11)
(+0.74) 29.40	Kimi K2.5		Kimi K2.5	29.40 (+8.62)
(+2.4) 21.00	DeepSeek V3.2		DeepSeek V3.2	11.60 (-1.63)
(-4.31) 14.40	Llama 3.3		Gemini 2.5	6.60 (-1.47)
(+0.85) 13.80	Grok 4.1		GPT-OSS	6.40 (-0.51)
(+3.08) 12.80	GPT-OSS		Llama 3.3	5.60 (-7.7)
(-2.75) 8.60	Gemini 2.5		Grok 4.1	4.40 (-4.81)

DeepSeek V3.2 late entry.

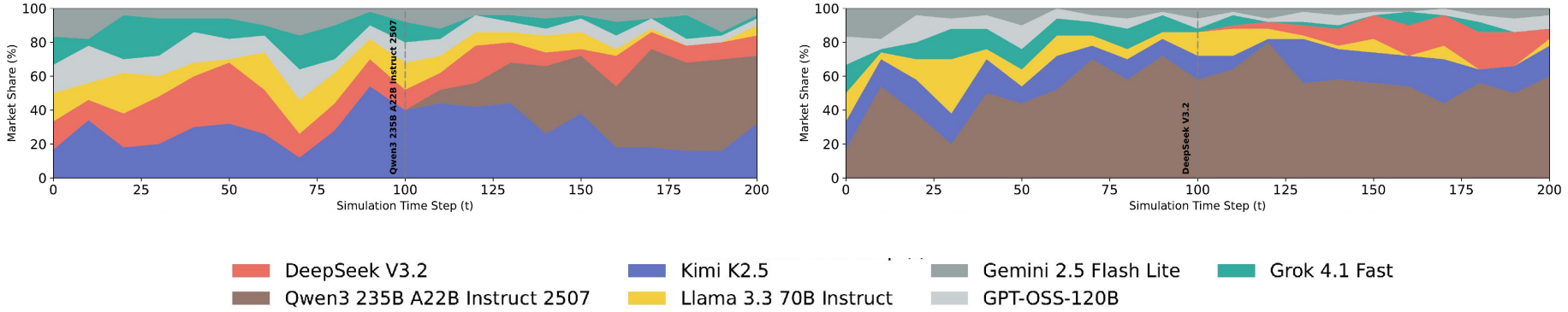
Market Share (ΔFS) (%)	System Ranking	→	System Ranking	Market Share (%) (ΔFS)
Before Entry HHI=0.32 $t = (1 - 100)$			After Entry HHI=0.38 $t = (101 - 200)$	
(+22.71) 51.60	Qwen3		Qwen3	57.80 (+28.91)
(-5.58) 14.80	Kimi K2.5		Kimi K2.5	15.80 (-4.58)
		new	DeepSeek V3.2	12.0 (-1.23)
(-2.70) 10.60	Llama 3.3		Llama 3.3	4.60 (-8.70)
(+0.39) 9.60	Grok 4.1		GPT-OSS	4.20 (-2.71)
(+0.49) 7.40	GPT-OSS		Gemini 2.5	3.20 (-4.87)
(-2.07) 6.00	Gemini 2.5		Grok 4.1	2.40 (-6.81)

Realized shares ≠ expected benchmark "fair" shares.



Motivating study (Lessons)

- Benchmark capability does not translate proportionally into market share
- Order of entry and market concentration changes who wins
- Dominance and winner-take-all effects emerge and this cannot be inferred from static benchmark alone
- A strong entrant can compress mid- and low-ranked models until the static ranking becomes uninformative
- **Need evaluation that explicitly models interaction, adaptation, and competition.**



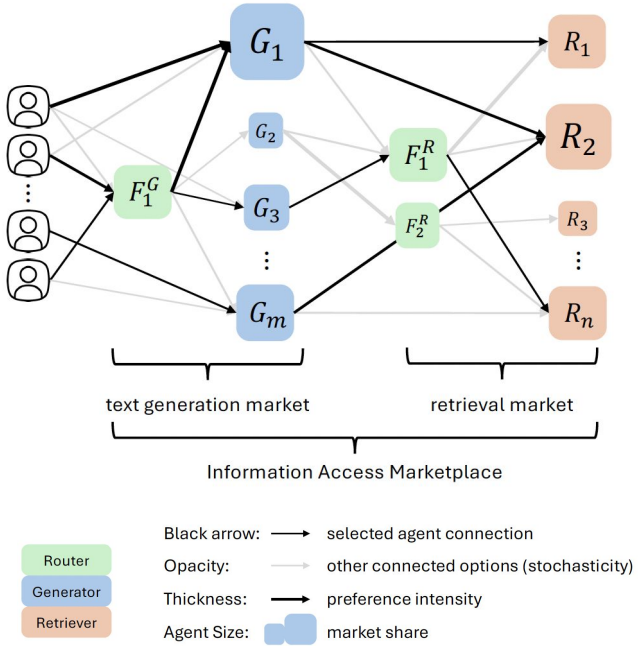
pip install marketplace-eval

- Graph-based simulation framework.
  - Node = market participants
  - Edge = preference-based selection(s)

- One example interaction:
  - user → generator → retriever
  - a judge scores the output as utility μ.

- Participants then update preferences from μ; repeated over many users and steps, traffic, preferences, and market share evolve.

- Focus on longitudinal outcomes:
  how agents attract demand, retain usage, and stay viable under competition.
  - Supports Market metrics
    - E.g., Customer retention, Dominance, Market Share, …



An example simulation snapshot of a RAG marketplace.
Users, generators, routers, and retrievers as participating stakeholders

pip install marketplace-eval

pip install marketplace-eval
https://github.com/kimdanny/marketplace-eval

```
marketplace_eval/       # Installable Python package (pip install marketplace-eval)
├── agents/             # Generator, router, retriever, and judge agent implementations
├── core/               # Node and link primitives plus IO helper nodes
├── humans/             # User taxonomy configurations and automatic profile generation
├── post_simulation/    # Market share, CRR metrics, plotting, and analysis CLI
├── prompts/            # Prompt templates for LLM judges and user simulation
├── system/             # System orchestrator, logging, types, and user population
└── utils/              # LLM client utilities for OpenAI-compatible APIs
```

- Simple configuration YAML
  - Configurations of simulation params, user population, update size frequency, marketplace graph, judge, etc.
- Highly flexible code extension.
  - You can override existing classes (update rules, routers, retrievers, and judges) without touching the core simulation engine.

What one simulation step does

- sample users → each picks a generator (over evolving preferences)
- generator calls routers / retrievers per graph topology
- Metric scores the answer → utility μ
- update preferences & router stats → log a StepRecord

Three Research Thrusts in the paper!

- RQ 1: “How should a marketplace be simulated?”
- RQ 2: “What metrics should be developed for marketplace evaluation?”
- RQ 3: “How to integrate marketplace eval. into evaluation campaigns like TREC?”

arXiv

TEKnologyy

toeunkim@cmu.edu

github.com/kimdanny/marketplace-eval

Language Technologies Institute

SIGIR2026
MELBOURNE | NAARM
AUSTRALIA

Research Agenda

A Long-Term Research Agenda

RQ1 · Simulation

- How should marketplaces be simulated?
- Model users: utility functions, choice models, evolving persistent preferences.
- Model agents: generators, routers, retrievers competing under quality–cost constraints.
- Couple dataset design with preference dynamics.

RQ2 · Metrics

- What metrics fit a marketplace?
- Agent-level: windowed market share, customer retention.
- Market-level: HHI concentration, dominance gap, expected-exposure / fairness.
- **Beyond Accuracy**

RQ3 · Adoption to Eval Campaigns

- How to fold into evaluation campaigns?
- Bring marketplace simulation to TREC / CLEF.
- Axis 1: peer vs. benchmark competition.
- Axis 2: run submission vs. agent submission.
- Plus: validation & meta-evaluation of the simulation.

RQ1: Simulating the marketplace

How should a marketplace be simulated?
Model each stakeholder as a strategic actor — with a goal, an adaptation strategy, and market-level evaluation signals.

RQ1.1: Modeling users (demand side)

- **Objective:** maximize utility from interactions — answer quality, cost, and trust.
- **Utility function:** $U = \alpha \cdot \text{Quality} - \beta \cdot \text{Cost} - \gamma \cdot \text{Latency}$, with heterogeneous per-user sensitivities.
- **Choice model:** stochastic (softmax) selection over estimated utility.
- preferences are coupled to dataset design — topic-dependent vs. persistent, evolving users.
- behavioral effects (framing, exposure/network effects) can amplify market concentration.

RQ1.2: Modeling agents (supply side)

- **Generators:** attract & retain users under quality–cost tradeoffs; adapt retrieval depth and orchestration (no / single / multi / agentic retrieval).
- **Routers:** intermediaries allocating queries (user→generator, generator→retriever); exclusive vs. open routing.
- **Retrievers:** compete for repeated selection via domain specialization, personalization, and robustness; maximize marginal contribution.

Stakeholders as strategic actors

- Each stakeholder is defined by a functional role, not an architecture — the same LLM can be a generator or a retriever.
- Every stakeholder has a goal, an adaptation strategy, and market-level evaluation signals that complement accuracy.

Stakeholder	Goal	Adaptive Behavior in the Marketplace	Evaluation Focus
Users	Sustained utility from the marketplace over time	Adapt preferences over generators based on past satisfaction, cost, latency, or trust	Longitudinal utility, retention rate, switching behavior
Generators	Maximize user demand under quality–cost tradeoffs	Adapt interaction protocols and retrieval configurations to remain competitive	User utility, cost efficiency, market share, user retention
Routers	Efficient allocation of queries to downstream services	Exclusive routing (vertical integration) vs. open routing (market-wide discovery)	Counterfactual regret, allocation efficiency, diversity, and fairness
Retrievers	Remain selected by generators under competition	Compete via domain specialization, personalization, or robustness	Marginal utility contribution, selection frequency, long-term survival

primary stakeholders in an information access agent marketplace.

RQ2: Marketplace metrics

Accuracy stays foundational, but doesn't capture performance as a market participant. Two complementary classes of metrics, read off the simulation logs.

Agent-level metrics

Market Share (windowed) — fraction of traffic in a sliding window.

$$\mathrm{MS}_a(t;w) = \frac{\sum_{u \in U} \sum_{k=t-w+1}^{t} \mathbf{1}[q_u(k) = a]}{\sum_{u \in U} \sum_{k=t-w+1}^{t} 1}$$

Customer Retention — do users keep choosing an agent after first adoption?

$$\mathrm{CR}_a(m) = \frac{1}{|U_a|} \sum_{u \in U_a} \mathrm{CR}_{u,a}(m)$$

High market share does not imply high retention.

Marketplace-level metrics

HHI — market concentration; rises toward monopoly.

$$\mathrm{HHI}(t;w) = \sum_{a \in A} \mathrm{MS}_a(t;w)^2$$

Market Dominance — top agent's share above its fair share target ε*.

$$\Delta(t;w) = \max_{a \in A} \mathrm{MS}_a(t;w) - \epsilon_a^*$$

Expected Market Exposure — using market share as a proxy for exposure

$$\mathrm{EE}_{\mathrm{marketplace}}(t;w) = \|\mathrm{MS}_a(t;w) - \epsilon_a^*\|_2^2$$

RQ3: Adoption to evaluation campaigns

How can TREC / CLEF evolve beyond static, Cranfield-style run submission to capture marketplace dynamics?
Two orthogonal design axes.

Axis 1: Peer vs. benchmark competition

- **Peer:** submissions compete directly for traffic and exposure in one shared simulation
  - Can reveal systems' behavior by their dominance, concentration, and switching, etc.
  - Downside: outcomes depend on the year's participant mix.
- **Benchmark:** each submission competes against a fixed, organizer-provided simulated user/agent population
  - better reproducibility and clearer attribution.

Axis 2: Run vs. agent submission

- **Run submission:** participants provide static outputs that are treated as cached outputs; only the user population adapts
  - low cost and reproducible, but systems cannot adapt.
- **Agent submission:** participants provide executable agents that make sequential decisions under competition
  - enables adaptive routing and strategic behavior, but complicates stochasticity, reproducibility, and gives high computational burden to track organizers

Task design: target interaction-heavy tracks:

- **distributed IR (TREC Million LLMs)**
- **generator–retriever coordination (TREC RAG)**
- **interactive information access (TREC iKAT, CAsT).**