Project Details

Jounghee Kim · AI Engineer · LLM Agents

onlyl4youu@gmail.com linkedin.com/in/joungheekim github.com/JoungheeKim Updated Sep 2026

Summary

AI Engineer who automates judgment work with LLM agents. At Toss, built a browser-use merchant review agent that saved 20 FTE in card-company review, and an inspection automation platform in which AI refines the prompts, now used by 10+ departments. At SK Telecom, built A-dot's scheduling agent (handling 67% of schedule registrations) plus movie-booking and main agents. Background in speech/NLP research (Interspeech 2022) and financial AI.

Key Results

20 FTE	**review headcount saved** — 01 Merchant Review Automation
10+	**departments** — 02 Inspection Automation Platform
+25 pp	**average accuracy gain** (single-policy pilot) — 03 Ad Review Agent
67%	**registrations via agent** — 04 A-dot Scheduling Agent
60→80%	**function-call success** — 05 A-dot Main Agent
12%	**booking conversion** — 06 Movie Booking Agent (T Membership & A-dot)
16→31%	**AI auto-answer rate** — 07 Cupid: Answer Recommendation for Similar Questions

Contents

Viva Republica Inc. (Toss) · Dec 2025 – Present

01	**Merchant Review Automation**	May 2026 – Present
02	**Inspection Automation Platform**	Jun 2026 – Present
03	**Ad Review Agent**	Mar 2026 – Jun 2026

SK Telecom · Apr 2022 – Oct 2025

04	**A-dot Scheduling Agent**	Apr 2025 – Oct 2025
05	**A-dot Main Agent**	Apr 2024 – Dec 2024
06	**Movie Booking Agent (T Membership & A-dot)**	Jan 2024 – Dec 2024
07	**Cupid: Answer Recommendation for Similar Questions**	May 2022 – Dec 2023

Project 01

Merchant Review Automation

Period	May 2026 – Present
Organization	Viva Republica (Toss)
Role	Review agent design and development

1. Background & Goals

- Reviewers opened each merchant website by hand to check and capture business details, terms and the payment path; a manual review took about three days on average. The automation targets same-day results.
- Every merchant site is structured differently, so reviewers had to hunt for the right pages one by one.

2. Key Challenges

1) Finding every piece of review evidence on sites that are all structured differently
2) Not burdening or endangering third-party merchant servers
3) Deciding when enough has been collected and stopping

3. Contributions

A. Browser-use review agent (Challenges 1, 3)

a. Designed a browsing loop (observe → reason → act) that reads the screenshot and page text together and decides where to look next
b. Set collection goals per review item (business details, terms, payment path) so the agent picks its next action from what is still missing
c. Made the agent stop on its own once it had what it needed, keeping captured screens and text as review evidence

B. Safety for third-party servers (Challenge 2)

a. Read public pages only, with one browser per merchant going one page at a time, so merchant servers see no more load than a single visitor

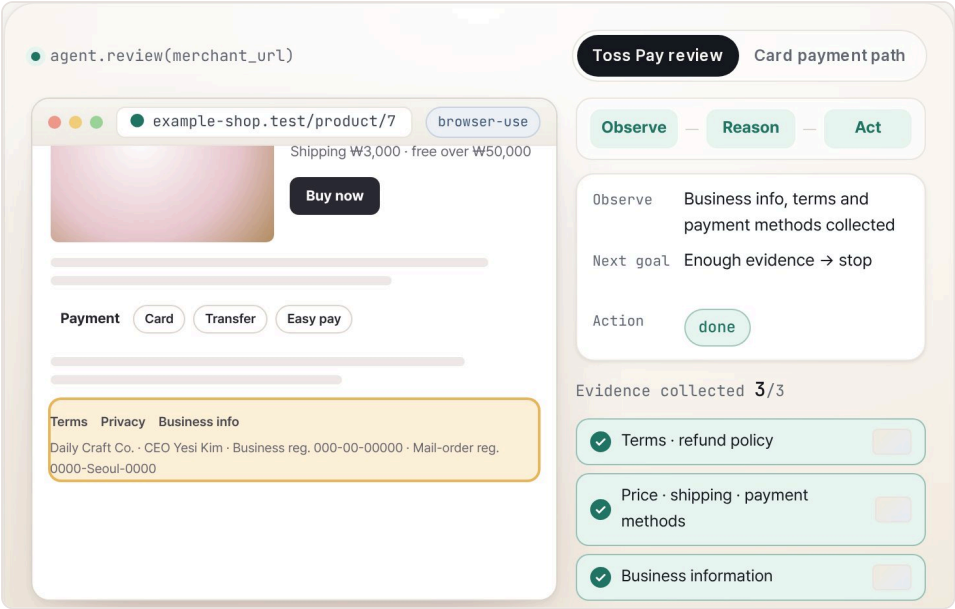
4. Tech Stack

Framework / Platform	browser-use, Headless Chromium, Multimodal LLM
Methodology	Agentic browsing, Prompt engineering

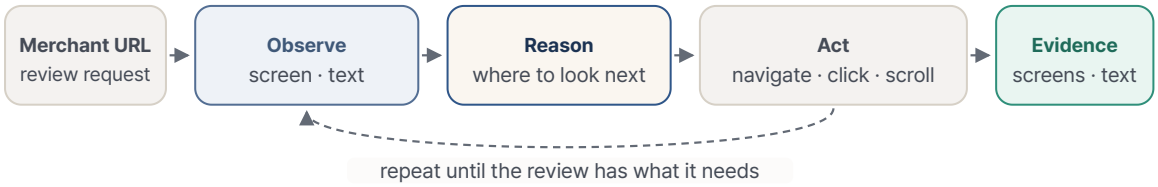
5. Results

- Saved 20 FTE by automating card-company merchant review (payment-path capture)
- Applied the same agent to Toss Pay merchant review (an 8-person workload)

6. Reference Material



Reconstructed screen · synthetic data



browser-use review agent · reads public pages only and hands the captured screens and text to the review as evidence

Project 02

Inspection Automation Platform

Period	Jun 2026 – Present
Organization	Viva Republica (Toss)
Role	Platform design and development

1. Background & Goals

- Every inspection task needed its own agent, and when results were off, people rewrote the prompts by hand and ran them again.
- When results were off, it was hard to tell whether the prompt was wrong or the policy and criteria (the direction) themselves needed to change.
- Requirements, policies and past cases were scattered across chat threads, documents and sheets, so even preparing to build an agent took a long time.

2. Key Challenges

1) Telling prompt problems apart from problems where the policy (the direction) must change
2) Cutting repeated manual steps such as organizing requirements, assembling agents and editing prompts
3) Managing each task's policies, data and formats together and improving them by version

3. Contributions

A. AI-driven improvement cycle (Challenge 1)

a. AI finds recurring format and evidence errors in run logs and review data, drafts prompt changes that fit the agent's structure, and re-runs the same data to compare
b. Errors a prompt cannot fix are shown to users as places where the data conflicts with the current direction (policy) or the policy has gaps, with supporting cases, affected scope and a proposed fix
c. Users set the direction and approve; AI does the editing, re-running and comparing, so the cycle is easy to repeat

B. AI features that cut the steps in between (Challenge 2)

a. AI gathers scattered material (chat threads, documents, sheets, owner notes), drafts the inspection requirements, and turns gaps into questions
b. From the requirements, AI proposes the components and their order and builds a draft inspection agent, flagging what it cannot build and what is missing
c. AI generates each task's prompts and supports versioned experiments, so prompt versions can be compared on results

C. **Platform architecture and shared module** (Challenges 2, 3)

a. Ran inspection agents on LangGraph; built the intermediate AI features (requirements, agent drafts, prompt learning, policy refinement) as Claude Agent SDK services

b. Kept policies, data, cases and input/output formats together as reference information, and connected each task's agent in a develop → run → improve flow

c. Stored each run's verdict, evidence, run history and version, and fed reviewers' final decisions (confirm, edit, hold) into the next improvement

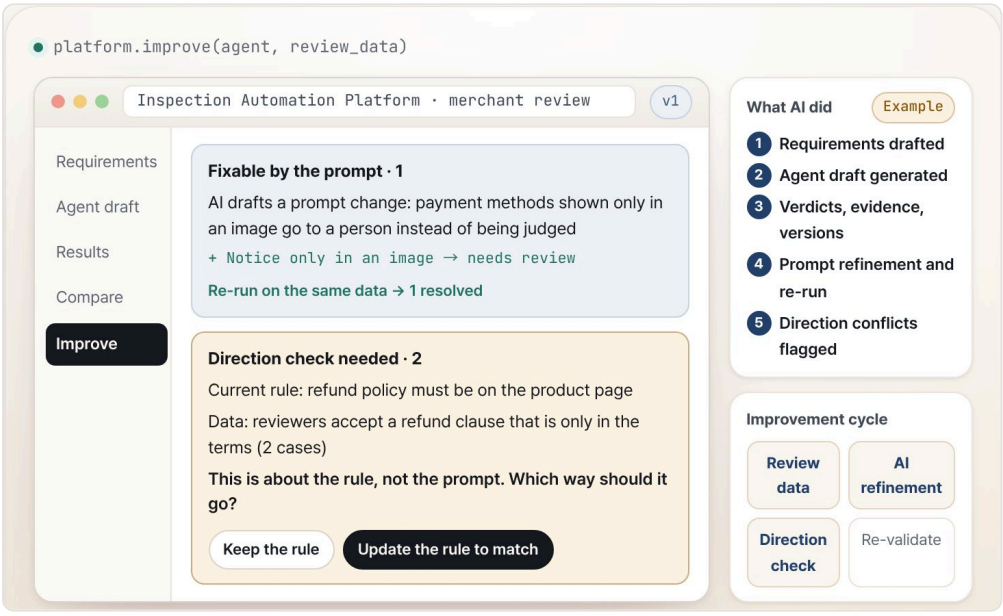
4. Tech Stack

Framework / Platform	LangGraph, Claude Agent SDK
Methodology	Requirement drafting, Agent drafting, Prompt learning, Policy refinement, Versioned experiments

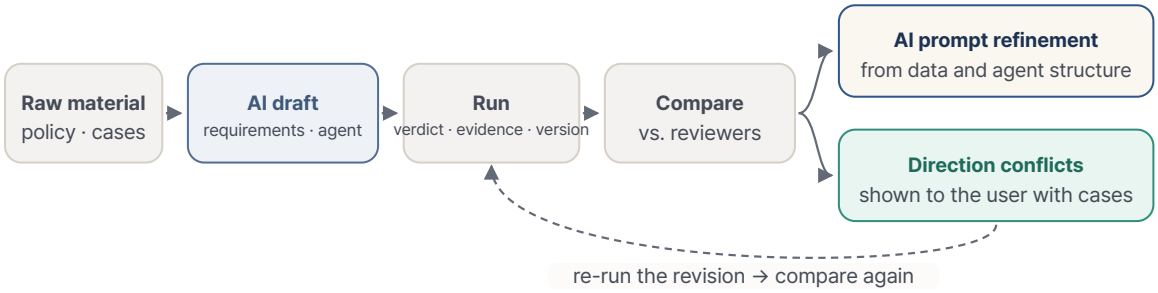
5. Results

- Used as a shared inspection module across inspection tasks in 10+ departments

6. Reference Material



Reconstructed screen · synthetic data



AI handles requirements, agent drafts and prompt refinement; conflicts a prompt cannot fix go to the user with supporting cases, which keeps the improvement cycle easy

Project 03

Ad Review Agent

Period	Mar 2026 – Jun 2026
Organization	Viva Republica (Toss)
Role	Design and development

1. Background & Goals

- Ad policies are often a line or two, but an AI reviewer needs a guideline with criteria, boundaries and examples, and people had to write one for every policy.
- When the guideline was stricter or looser than real reviewers, it rejected good ads or missed violations, and finding where it went wrong was hard.

2. Key Challenges

1) Turning a short policy into workable review criteria
2) Closing the gap between the guideline and reviewers with data
3) Avoiding overfitting to the refinement data

3. Contributions

A. Automatic guideline generation (Challenge 1)

a. Found the basis in the policy source text and turned a short policy into a guideline with decision criteria, violation/OK boundaries and policy references

B. Data-driven refinement (Challenges 2, 3)

a. Had AI compare the guideline's decisions with reviewers', find the patterns behind the disagreements and revise the guideline (e.g. marking compounds, amount units and ad-style phrasing common in ads as OK)
b. Split refinement and held-out data to confirm the improvement holds on unseen ads
c. People only checked the revisions; nobody edited the guideline by hand

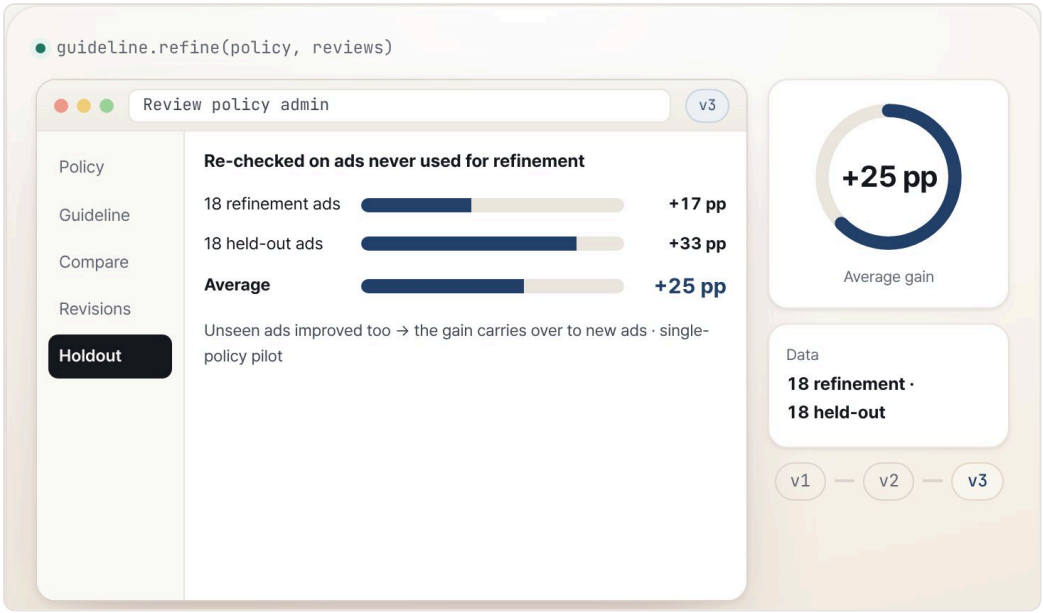
4. Tech Stack

Framework / Platform	LLM, Prompt engineering
Methodology	Data-driven prompt refinement, Holdout validation

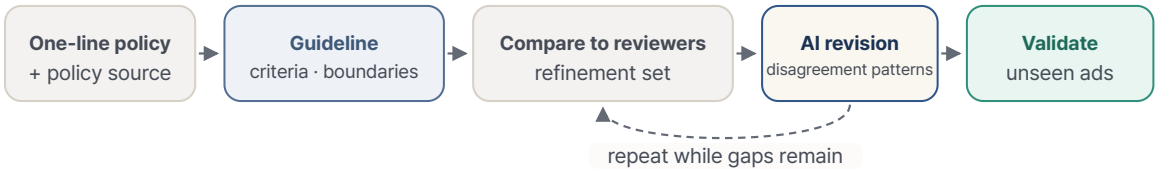
5. Results

- Raised average accuracy by 25 pp on an ad-copy typo policy pilot (36 ads)
- Accuracy also rose on held-out ads never used for refinement, so the gain holds on new ads
- This approach became the prompt-refinement cycle of the Inspection Automation Platform

6. Reference Material



Reconstructed screen · synthetic data



Policy → guideline → data-driven AI revision → check on unseen ads · +25 pp average accuracy on the pilot policy

Project 04

A-dot Scheduling Agent

Period	Apr 2025 – Oct 2025
Organization	SK Telecom
Role	Agent design and development

1. Background & Goals

- We built an agent that handles A-dot's events and reminders through conversation.
- The goal was to handle multi-intent requests such as "add a dentist appointment at 3 tomorrow and push Friday's meeting back 30 minutes" quickly and accurately.

2. Key Challenges

1) Multi-intent requests and response latency
2) Integration with external calendars (Outlook, Google Calendar) and subscription calendars
3) Accurate handling of time information such as recurring events and reminders

3. Contributions

A. Agent architecture with the Plan-and-Execute pattern and a Refine step (Challenge 1)

a. Plan: splits the request into sub-tasks and orders them
b. Execute (sequential): completes dependent sub-tasks in order
c. Execute (parallel): runs independent sub-tasks in parallel to cut latency
d. Refine: re-plans from the execution results to raise task accuracy

B. External and subscription calendar integration (Challenge 2)

a. Designed the flow so the LLM distinguishes external calendars (Outlook, Google Calendar) from subscription calendars (benefit and event schedules) and decides whether each is linked and can be read, changed or deleted

C. RAG for schedule retrieval (Challenge 1)

a. Filtering: narrows candidate events by owner, subscription and date range
b. Parallel search: splits candidates into N chunks and calls the LLM in parallel to find the requested events within the latency budget
c. Verification: checks that each extracted event matches the request and extracts the supporting evidence to improve accuracy

D. Prompt engineering for time information (Challenge 3)

a. Designed prompts that use the iCalendar standard and ISO 8601 durations so recurring events and reminders are handled precisely

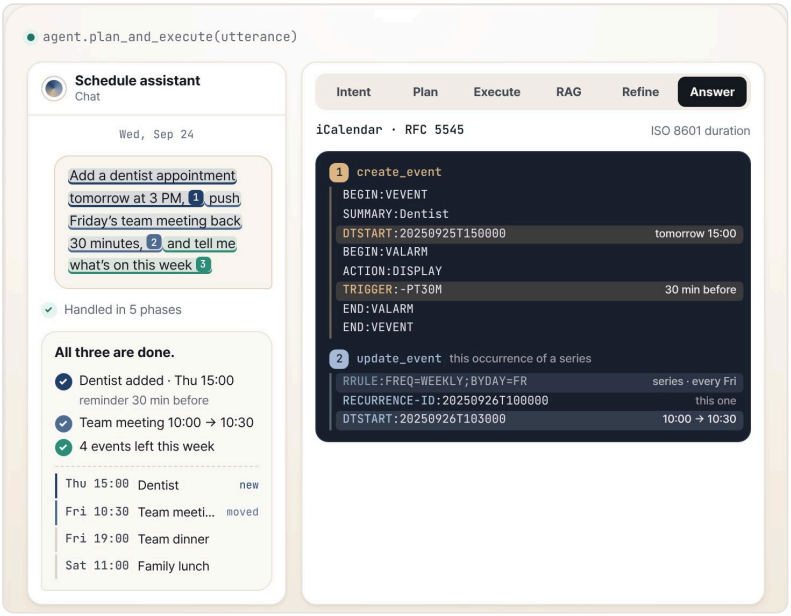
4. Tech Stack

Framework / Platform	LangGraph, Gradio, FastAPI
Methodology	Prompt engineering, RAG, Plan-and-Execute

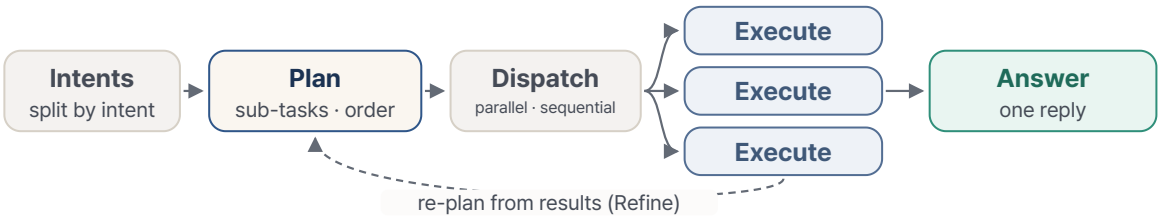
5. Results

- 67% of schedule registrations came through the agent (66K of 98K)
- Service-wide schedule MAU grew 2.5× (203K in Jan → 511K in Sep 2025)

6. Reference Material



Reconstructed screen · synthetic data



Plan-and-Execute: independent tasks in parallel, re-plan from results

- A-dot adopts agentic workflows (SKT Newsroom) news.sktelecom.com
- Recurring events in A-dot Calendar (DEVOCEAN) devocean.sk.com
- Plan-and-Execute Agents (LangChain) blog.langchain.com
- SKT–TimeTree partnership (SKT Newsroom) news.sktelecom.com
- A-dot 4.0 agentic workflows (SKT Newsroom) news.sktelecom.com

Project 05

A-dot Main Agent

Period	Apr 2024 – Dec 2024
Organization	SK Telecom
Role	Agent design, prompt engineering, training-data design, fine-tuning

1. Background & Goals

- A-dot's flagship LLM agent, offering everyday conversation plus 17 functions such as exchange rates, weather, time, directions, news and subway congestion.

2. Key Challenges

1) Function-call accuracy and hallucination
2) API cost and latency
3) Multi-turn conversation quality

3. Contributions

A. Agent design for accuracy and UI/UX integration (Challenge 1)

a. Consolidated similar functions to simplify function selection
b. Let the conversation use in-app context such as the current playlist and UI screen
c. Split functions with low argument accuracy into sequential calls

B. Prompt engineering for cost and answer accuracy (Challenges 1, 2)

a. Kept prompts minimal to cut API cost and used fine-tuning to instill answer style
b. Switched argument extraction for hallucination-prone time functions to NER

C. Fine-tuning and evaluation (Challenge 3)

a. Defined the training-data format and generated multi-turn dialogue data
b. Fine-tuned GPT models on Azure OpenAI
c. Ran quantitative evaluation (function selection and accuracy, argument selection and extraction) and qualitative evaluation (multi-turn fluency, accuracy of function-result answers, handling of sensitive and inappropriate requests)

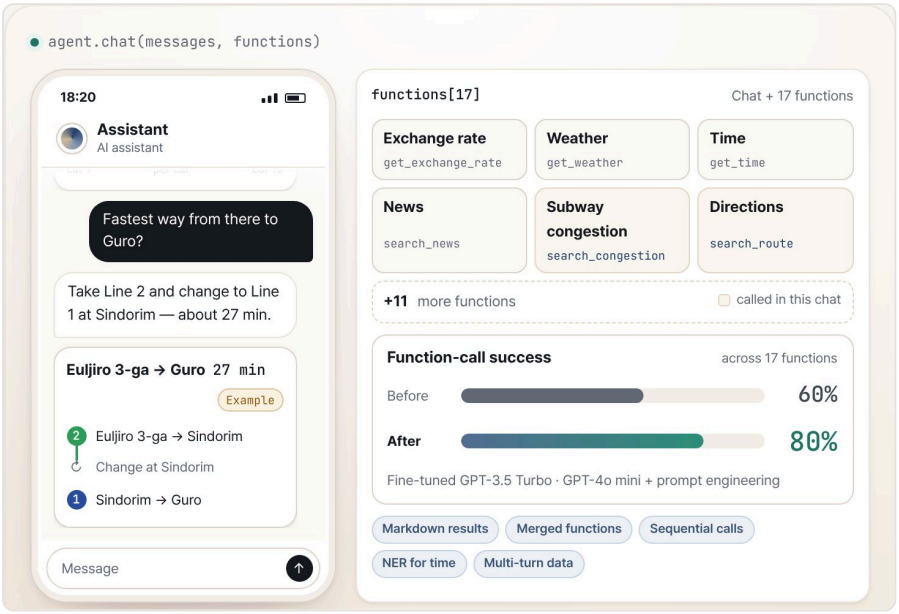
4. Tech Stack

Framework / Platform	Azure OpenAI
Methodology	Prompt engineering, Function calling, Fine-tuning, NER

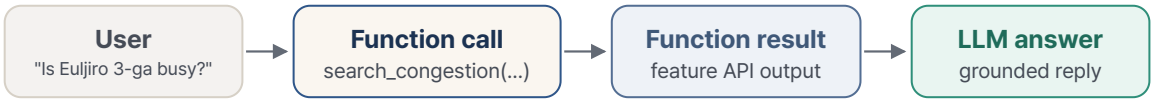
5. Results

- Raised function-call success across 17 functions from 60% to 80% through fine-tuning and prompt engineering
- Reduced hallucination and improved time-parsing accuracy by extracting time arguments with NER and analyzing them sequentially

6. Reference Material



Reconstructed screen · synthetic data



An agent that answers through function calls · data design, fine-tuning and prompt work raised function-call success across 17 functions from 60% to 80%

– A-dot LLM agents in practice (SKT Newsroom)
news.sktelecom.com

Project 06

Movie Booking Agent (T Membership & A-dot)

Period	Jan 2024 – Dec 2024
Organization	SK Telecom
Role	UX & LLM workflow design, RAG development

1. Background & Goals

- A conversational agent for T Membership movie booking, meant to make getting recommendations, searching and booking as easy as talking to cinema staff.
- The service combines chat with UI elements such as buttons, so the UI/UX and the LLM workflow had to flow into each other naturally.

2. Key Challenges

1) Slot filling through conversation, with personalization
2) Integrating the UI/UX with the LLM workflow
3) Movie search accuracy, latency and API cost

3. Contributions

A. **Slot-filling agent** (Challenge 1)

a. Designed a function-calling LLM workflow that informs the user while collecting what the booking needs
b. Personalized recommendations using the user's location, recently visited cinemas and favorite theaters
c. Added a state-management module so slots collected in long conversations are not lost

B. **RAG for movie search** (Challenge 3)

a. Preprocessing: tags movie titles in the request with a Trie built from a title-synonym dictionary
b. Keyword extraction: uses the LLM to extract movie metadata and keywords from the request
c. Retrieval: vector search with metadata filtering on the extracted keywords
d. Final pick: uses the LLM to choose the movies that fit the request

C. **Automated movie-metadata pipeline** (Challenge 3)

a. Extracted keywords, summaries and metadata from daily movie updates with Airflow, an LLM and preprocessing modules
b. Converted the results to vectors and synced them to the vector DB

D. **UI/UX and LLM workflow integration** (Challenge 2)

a. Fed the results of UI actions (buttons, etc.) into the LLM workflow's prompt so both flow into each other naturally

E. Log-based data collection and training (Challenge 3)

a. Collected training data from service logs and fine-tuned GPT models on Azure OpenAI to raise function-calling accuracy and multi-turn fluency

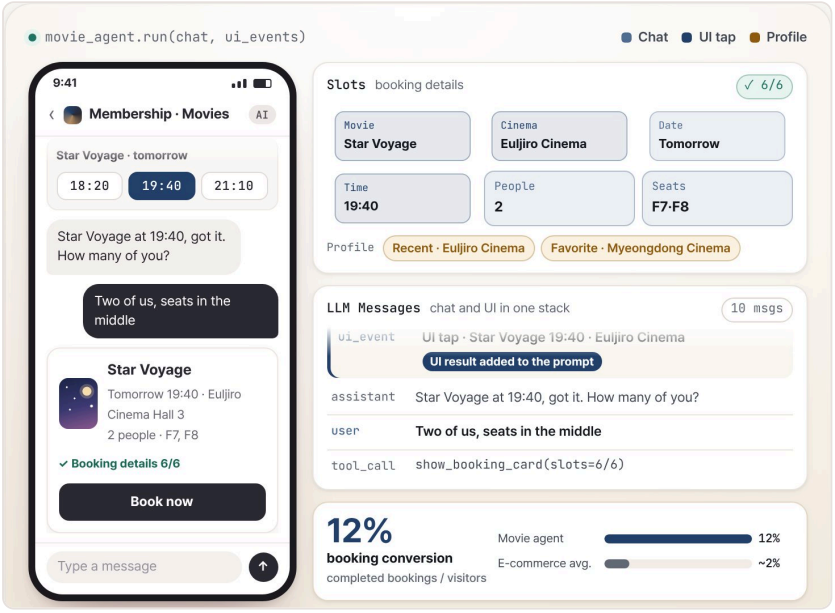
4. Tech Stack

Framework / Platform	Azure OpenAI, Datadog, Airflow
Methodology	Prompt engineering, Function calling, RAG

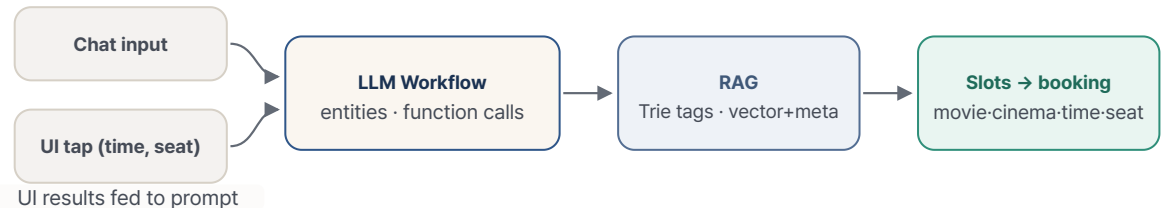
5. Results

- 8,000 MAU for T Membership movie booking; 110K cumulative users of the movie-booking agent
- 12% booking conversion (T Membership movie booking: completed bookings / visitors)

6. Reference Material



Reconstructed screen · synthetic data



Chat and UI flow into one LLM workflow for booking

– A-dot LLM agents in practice (SKT Newsroom)
news.sktelecom.com

– Movie agent LLM workflow (SKT Enterprise)
sktenterprise.com

Project 07

Cupid: Answer Recommendation for Similar Questions

Period	May 2022 – Dec 2023
Organization	SK Telecom
Role	Model development, recommendation and evaluation pipelines

1. Background & Goals

- Cupid is a community service where nearby users ask and answer each other's questions.
- We wanted an AI feature that finds answers already given to similar questions.

2. Key Challenges

1) Question–answer matching accuracy across topics and regions
2) Automated evaluation of the main models

3. Contributions

A. Content-based recommendation pipeline (Challenge 1)

a. Classification and extraction: extracts location and category from the question (BERT-based classifiers and NER)
b. Filtering: narrows candidates by location, category and stop words
c. Encoding: masks place names and converts the question to a vector with the embedding model
d. Retrieval: finds similar questions in Elasticsearch with BM25 + cosine similarity
e. Re-ranking: scores question containment and question–answer consistency and returns the top N

B. Models behind the pipeline (Challenge 1)

a. Embedding model: trained a BERT-based model with a Siamese network so similar sentences sit close together
b. Hierarchical multi-class classifier: built hierarchical topic data and trained a BERT-based classifier
c. NER model: trained with BIO tagging to extract places and keywords

C. Model evaluation pipeline (Challenge 2)

a. Sent items whose category changed plus N sampled items to an external vendor for ground-truth labeling
b. Evaluated with F1 and accuracy and extracted items with low prediction accuracy
c. Gated deployment on a golden-set evaluation after retraining

D. Batch processing of new data (Challenge 1)

a. Automated an Airflow batch that encodes new question–answer pairs and re-indexes the vector DB so they become recommendation candidates

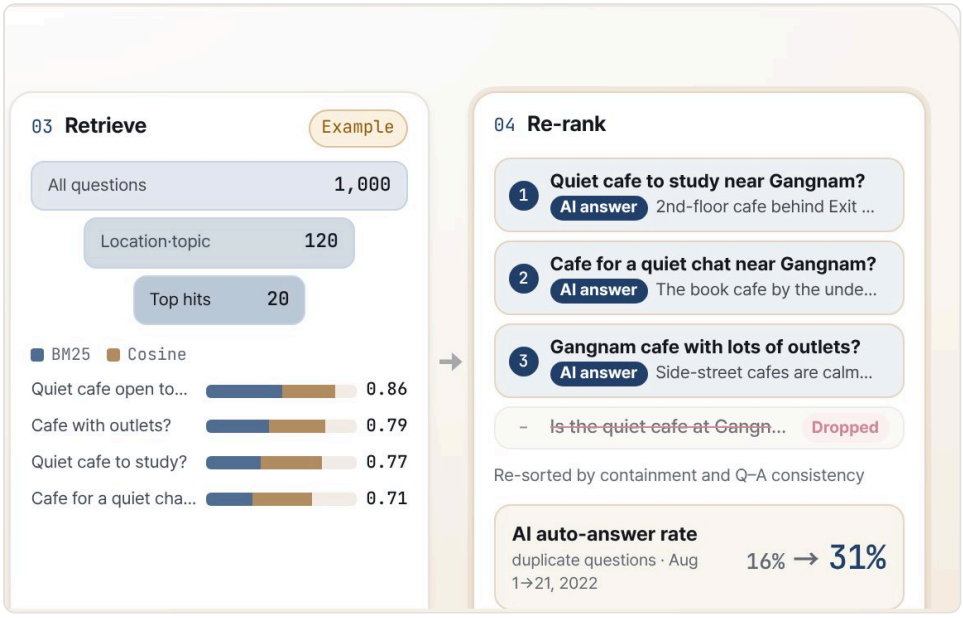
4. Tech Stack

Framework / Platform	PyTorch, Transformers, Elasticsearch, Airflow
Methodology	Siamese network, Hierarchical multi-class classification, BIO NER

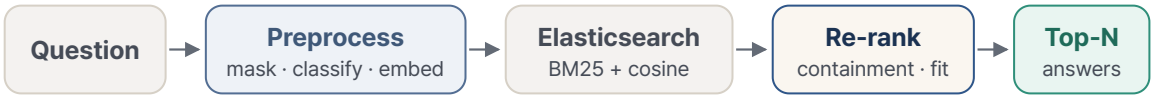
5. Results

- AI auto-answer rate for duplicate questions rose from 16% to 31% (Aug 1 → Aug 21, 2022)
- Positive ("helpful") feedback on AI answers rose from 38% to 57%

6. Reference Material



Reconstructed screen · synthetic data



Content-based recommender that finds existing answers to similar questions

– The team behind Cupid (SKT Newsroom)
news.sktelecom.com

Other Projects

K-Wav2vec 2.0: Korean ASR with Grapheme–Syllable Joint Decoding — Sep 2022
Interspeech 2022 · First author

In-vehicle & Mobile Speech Recognition — Feb 2022 – Apr 2022
Hyundai Motor Company AIR Lab

Multi-modal Korean Emotion Recognition with Consistency Regularization — Dec 2021
JKIIE 2021 · First author

Back-Translated Task Adaptive Pretraining — Jul 2021
arXiv 2021 · Second author

Open-Domain Korean Question Answering — Apr 2020 – Jul 2020
Korea University (graduate project)

Predictive Maintenance for Chemical Processes — Jan 2020 – Sep 2021
Korea Univ. DSBA × Hanwha Systems (ICT) industry–academia project

Market Caster: News-based Market Prediction — Jul 2019 – Mar 2020
SK C&C

Portfolio Robo-advisor — Jul 2018 – Feb 2019
SK C&C

Early Warning for Corporate Distress — Jan 2018 – Jul 2018
SK C&C (client KDB)

KDB E-Finance Services — Jan 2016 – Mar 2017
SK C&C (client KDB)

Details: joungheekim.github.io/portfolio/en

※ Screens and diagrams under Reference Material are illustrative reconstructions, not production screens or data.