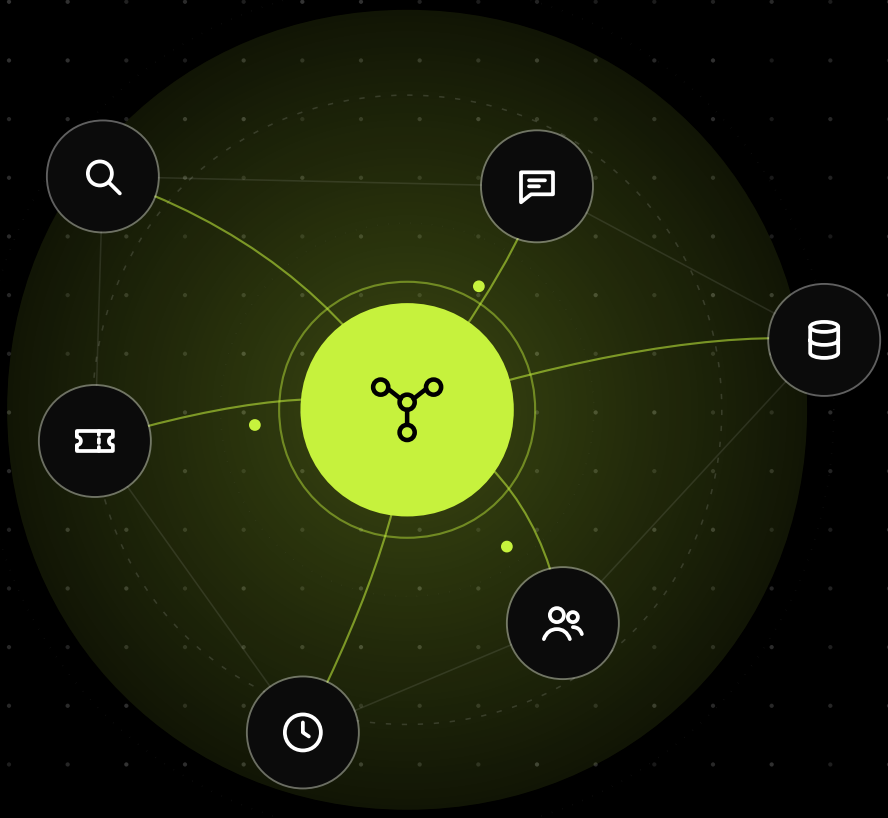


LinkedIn

Support tickets resolved 28.6% faster with AI search

LinkedIn's customer service team used an AI system that answers new questions from a knowledge graph of past tickets. In a randomized test, the group using it resolved issues faster.

Technology | Customer service, technical support | Knowledge graph | Retrieval-augmented AI

77.6%
better retrieval accuracy (MRR score)

28.6%
lower median time to resolve an issue

40 to 15 h
mean resolution time per issue

AT A GLANCE

COMPANY	WHO USED IT	AI MODEL NAMED	REPORTED
LinkedIn	LinkedIn's customer service team, across multiple product lines	GPT-4	Research paper, SIGIR 2024 (arXiv, Apr. 26, 2024)

01

The challenge

In customer service technical support, finding the right past ticket quickly and accurately is critical. The answer to a new question is often sitting in an issue someone already solved.

Standard AI search treats a large pile of past tickets as plain text. That ignores how each ticket is organized inside, and it can disconnect related content, which leads to incomplete answers.

LinkedIn's researchers set out to keep that structure: the sections inside each ticket, and the links from one ticket to another.

7 h
median time to resolve an issue without the system

40 h
mean time to resolve an issue without the system

0.522
retrieval score (MRR) of the plain-text baseline

BEFORE

- × Past tickets searched as plain text
- × Structure inside each ticket ignored
- × Related content disconnected, answers incomplete
- × Median 7 hours to resolve an issue

AFTER

- ✓ Past tickets stored as a knowledge graph
- ✓ Sections and links between tickets kept
- ✓ GPT-4 answers from the matching sub-graph
- ✓ Median 5 hours to resolve an issue

02

The solution

The team built a knowledge graph with two levels: relations inside each ticket, and relations between tickets.

When a customer question comes in, the system parses it to find the named entities and the intent, retrieves the related part of the graph, and GPT-4 writes the answer from it.

If the graph lookup comes up short, the system falls back to ordinary text search, so there is always material to answer from.

HOW IT WORKS



STEP 01

Read each past ticket

A language model parses each ticket into its sections, guided by templates.



STEP 02

Build the graph

Sections form a tree for each ticket, and links connect related tickets.



STEP 03

Embed for search

Ticket content is turned into embeddings so similar issues can be matched.



STEP 04

Understand the question

For a new question, the system identifies the named entities and the intent.



STEP 05

Retrieve the right tickets

It pulls the related sub-graph; plain text search is the fallback.



STEP 06

Write the answer

GPT-4 drafts the answer from what was retrieved.

BUILT WITH

Knowledge graph · Neo4j · Qdrant · BERT and E5 embeddings · GPT-4 · Text search fallback

03

The results

The method was deployed in LinkedIn's customer service team for about six months. The team was split randomly into two groups, and the group using the system cut median time per issue by 28.6%.

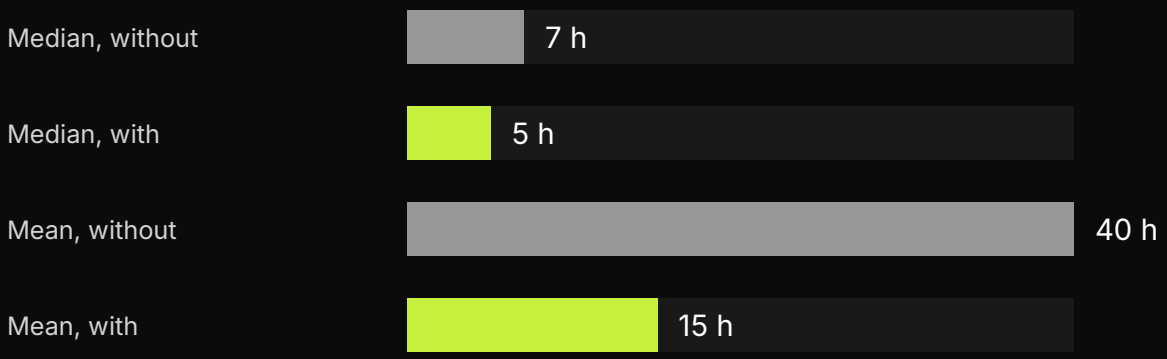
77.6%
higher retrieval accuracy (MRR 0.522 to 0.927)

28.6%
lower median resolution time, 7 to 5 hours

40 to 15 h
mean resolution time per issue

87 to 47 h
90th percentile resolution time

HOURS TO RESOLVE AN ISSUE, WITHOUT VS WITH THE SYSTEM



TIMELINE

Apr. 26, 2024
Paper posted on arXiv by LinkedIn researchers

July 14 to 18, 2024
Presented at SIGIR 2024, Washington, DC

04

What made it work, and what to watch

> "... has improved retrieval and answering metrics, and overall service effectiveness."
>
> Xu et al., LinkedIn, in the conclusion of their SIGIR 2024 paper

1	**Keep the shape of your data**	Treating tickets as plain text lost the structure that made them useful. Keeping sections and links improved retrieval.
2	**Measure against a control group**	One group used the system and one did not, which gave a fair measure of the time saved.
3	**Always have a fallback**	When the graph lookup misses, ordinary text search still supplies material to answer from.
4	**Plan for upkeep**	The researchers list automatic graph templates and dynamic updates as future work, so the graph keeps pace with new tickets.

IS THIS A FIT FOR YOUR BUSINESS?

- ✓ Does your team answer support tickets or customer questions every day?
- ✓ Have you already solved most of those problems somewhere in your ticket history?
- ✓ Do new team members take a long time to learn where the answers live?
- ✓ Would saving hours on each ticket matter to your customers?

05

How Agent Victoria could do this for you

We could connect your help desk history so your team gets suggested answers drawn from tickets you've already solved. Your team spends less time searching and more time solving.

01

Connect your help desk

Your solved tickets, notes and fixes.

02

Map what you know

Past cases organized so related issues link up.

03

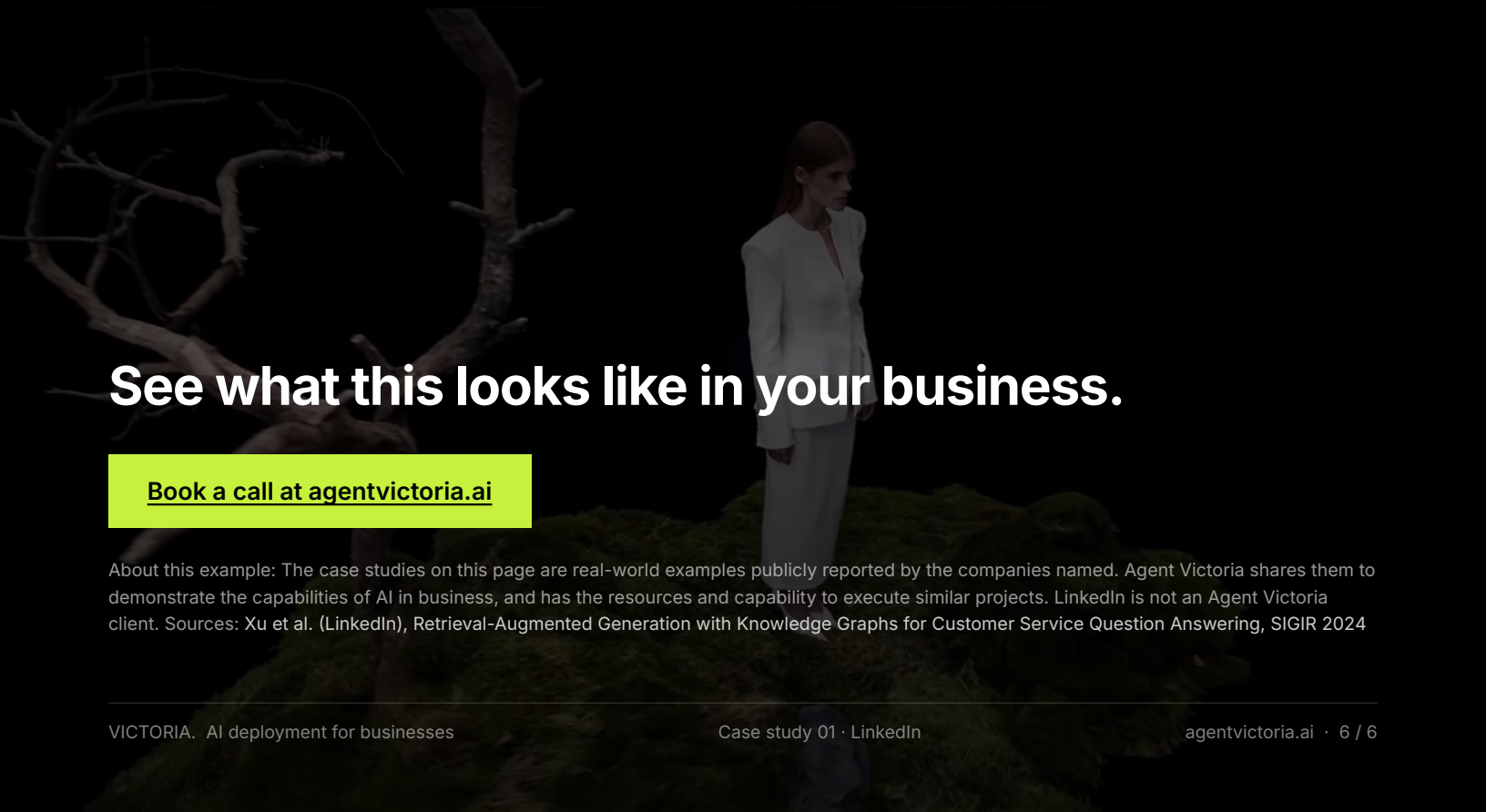
Suggest answers

Each new ticket gets a draft drawn from past fixes.

04

Your team decides

Agents review, edit and send.



See what this looks like in your business.

Book a call at agentvictoria.ai

About this example: The case studies on this page are real-world examples publicly reported by the companies named. Agent Victoria shares them to demonstrate the capabilities of AI in business, and has the resources and capability to execute similar projects. LinkedIn is not an Agent Victoria client. Sources: Xu et al. (LinkedIn), Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering, SIGIR 2024