

Two Simple Models for Making Intuitive Conversational AI Experiences

Author: Jonathan Dexter, Monkeyjump Labs CEO & Partner

If time is money, shouldn't we consider how much time is spent looking for answers to questions that impact our jobs? Angst builds when we utter, "Let me do some digging."

With the exponential growth of conversational AI, spending minutes and hours following trails of information on multiple search engines should be considered a crime. The internet is programmed for instant results, and your job or employees will benefit the most if you take advantage.

Conversational AI like Siri and Google Assistant use voice recognition, parse your statement, and pipe the concept into many pre-programmed results — like opening Maps, adding a To-Do Item, or sending a text message. They have changed how we interact with our digital devices and organize our lives in a small way (or large, depending on your usage).

This problem has three "basic" parts (and by basic, we mean simplified for legibility — each portion consists of many steps). These parts are:

1. The conversion of voice to meaning (strings for computers), the representation of language and text that computers understand;
2. The parsing and conversion of that content into intent; and
3. The movement of intent into action

Various companies may play in all places, but recent entrants have focused on innovating in the second stage.¹ ChatGPT and other players have lowered the barrier of access for independent producers to create meaningful experiences along this journey.

This explosion of interest and availability has caused a 7x growth since 2015.³ That doesn't even include ChatGPT impacts, which went public on November 30, 2022. All industries continue to explore this technology, including E-Commerce, Manufacturing, Human Resources, Health Care, Real Estate, Travel & Hospitality, Auto, Education, and Insurance.

This innovation is directly happening through [Large Language Models \(LLM\)](#),

¹"Chat GPT", Open AI, August 13, 2023, <https://chat.openai.com/>

²"Google Bard", Google, August 13, 2023, <https://bard.google.com/>

³Thormundsson, Bergur, "Total Global AI Investment 2015-2022", Statista, August 13, 2023, <https://www.statista.com/statistics/941137/ai-investment-and-funding-worldwide/>

generally a type of Recurrent Neural Networks, the most popular of which is ChatGPT. These language models have ingested an absolutely insane amount of information.⁴ This allows them to build and surface information for every kind of company.

Integrating conversational AI like ChatGPT may feel challenging, as the technology of Conversational AI (and its underpinning on Deep / Recurrent Neural Networks) can sometimes seem like magic. However, we wanted to share two simple models for making intuitive conversational AI experiences that are realistic for many corporations to implement right now.

Conversational AI Model #1:

Capability: Type in a question and receive the information you need.

This conversational AI model provides detailed information a user is looking for.

Most organizations have places where they store knowledge that provides a competitive advantage, consistency, efficiency, or some other category of important information. These knowledge bases (or wikis) often experience an ever-expanding growth, with the continual addition of

⁴ Hughes, Alex, "ChatGPT: Everything you need to know about OpenAI's GPT4 Tool", BBC: Science Focus, August 13, 2023, <https://www.sciencefocus.com/future-technology/gpt-3>

information making finding the 'right answer' more complex over time. This model makes it easier and faster for customers and employees to get answers to their specific company-related questions. Think: reducing operational expenses of time and cost, whether solving repetitive customer questions with an automated chatbot response to common questions or solving an issue of widely-dispersed information with a single source query.

Creating the Model

Three inputs that come together in a conversational AI model:

1. System prompt

A system prompt is a text or code you manually apply to retrieve the information you want to display. Example; in our [internal ChatGPT-powered chatbot](#), we provided a prompt along the lines of “You are a chatbot for Monkeyjump labs, a digital agency” to allow for setting a baseline for our LLM's behavior.

2. User prompt

A user prompt frames the intended output so the query or question the user asks can be accurate. Essentially, the user prompt builds the type of conversation the user wants to have. For example, an MJL employee could ask, “How do I use the MJL conference rooms?” This user prompt would then pull information from the knowledge base.

3. Knowledge base

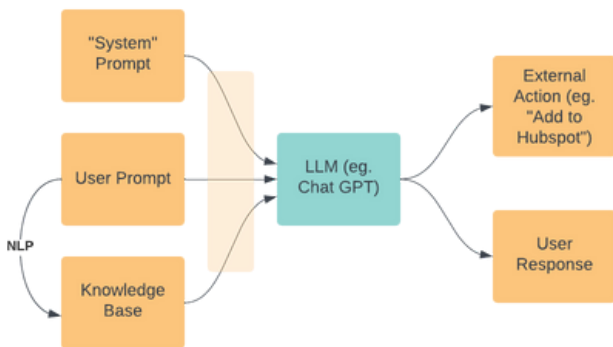
The user prompts are connected through the back channel to the knowledge base.

Connecting the pieces together

Once we have these three input components, we want to stitch them together to combine them into a single, more informed, response. This is termed Retrieval Augmented Generation⁵(RAG).

A naive approach may be the following:

- The user prompt is parsed via important terms through NLP (natural language processing).
- Next, our knowledge base is directly queried with these terms
- Then the results are combined with the user prompt and submitted to the LLM to get an actual response



⁵"Retrieval Augmented Generation using Azure Machine Learning prompt flow (preview) - Azure Machine Learning." Microsoft Azure Machine Learning, 31 July 2023, <https://learn.microsoft.com/en-us/azure/machine-learning/concept-retrieval-augmented-generation?view=azureml-api-2>

Figure 1: Block diagram of the process for a knowledge-based integration of an LLM into an existing platform.

A more advanced approach, required especially when your data source is large (for example, the cumulative knowledge of your enterprise over the last 20 years) generally involves creating "embeddings" from these data sources:

- Part 1 — Preparation
 - The information is parsed via an ETL (or ELT) process
 - Each "item" is converted into an embedding (vector representation of the content)
 - Those embeddings are stored in a vector database
- Part 2 - Retrieval
 - The user prompt is converted into an embedding
 - We search our vector database for similar items
 - We retrieve those similar items from the knowledge base to pass into the LLM
 - Then the results are combined with the user prompt and submitted to the LLM to get an actual response

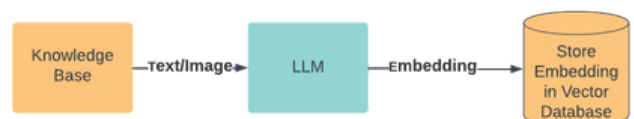


Figure 2: Block diagram of the process for a knowledge-based extraction of embeddings from a knowledge base

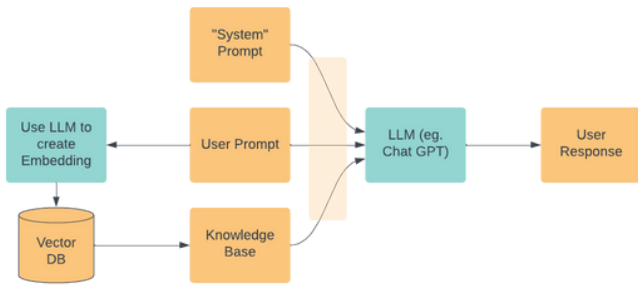


Figure 3: Block diagram of the process of utilizing an embedding search for rapid knowledge access

These three inputs that come together into the Large Language Model (LLM) are combined into a single input.

This combination of content produces a good feedback mechanism for users, allowing the discoverability of information to produce rapid results. Additionally, the knowledge base need not be only one: you can connect multiple systems (for the sake of example, let's say Confluence and Jira) to produce similar results. You do have to begin to worry about cross-system ranking at that point, which is a more difficult problem to discuss than the scope of this piece.

Conversational AI Model #2:

Capability: Enlist the help of AI to take action on a request.

Two inputs come together to create the action model:

1. External action

- a. Example: "Add a meeting to my calendar"
- b. The external action comes through an agent or decision-making source that is exposed to an environment in which it can replicate an intended response. Based on its options, the agent can assess the probable outcomes and value and make a decision. The decision will be based on the overall expected applicability of the action to the desired outcome.

2. User response

- a. When the action is completed, the AI model will return to the user with a response, confirming the outcome with something like, "I've added the meeting to your calendar."

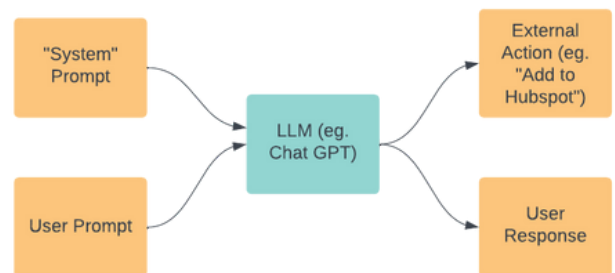



Figure 4: Block diagram of an action-based integration of an LLM into an existing platform.



The action model doesn't need the knowledge base that the conversational AI model requires, it just takes action. Ultimately, the action model AI is an opportunity to do even more with the information it has been given.

This model focuses on taking the resulting information from the LLM, and piping that into a system that produces some sort of action. Examples might be:

- Sending a deal to your CRM (e.g. Salesforce, Hubspot, Netsuite) in order to track it
- Cataloging action in order to summarize items humans have talked about into a ticket tracker (e.g. ServiceNow, Jira)
- Scheduling a meeting for later

As a caveat to the reader: unlike the previous model, which relies primarily on the final delivery to the reader of the output of the LLM, this step has an intrinsic action. We highly recommend providing capabilities like this into your platform tied to a user review — a "does this look right?" step for your user. This can prevent mistakes from going uncaught, as even the best models have an 80 to 90% success rate.⁶ This review step is a required step for good user experiences.

Finally, while independent, these models may be combined to produce a full-featured experience. There is no reason why a SaaS platform or product could not combine its own knowledge base with internal platform actions.

⁶ Lynch, Shana, "AI Benchmarks Hit Saturation", Stanford University: Human-Centered Artificial Intelligence, August 13, 2023, <https://hai.stanford.edu/news/ai-benchmarks-hit-saturation>

AI will continue to evolve and improve over the years. It is likely to provide more rapid access to information in a new and novel way (although with risks associated with misinformation). It's imperative for businesses where to leverage this new technology — and where to hold off. A technology partner like Monkeyjump Labs can help you determine what kind of AI software is best for your business and growth strategy.

Set up a [consultation call](#) or a workshop to ideate and design your AI experience.

