PL
REDEFINING SALES WITH AI-DRIVEN CX FROM ETIYA
ENGINEERING DIGITAL CX WITH ERICSSON
ADVOCATING FOR ENTERPRISE CX
AVOIDING CRM PITFALLS
HELP DESK CX
CX DATA DISCONNECT
DATA-DRIVEN DISCOVERY
CLOSING THE CX-EX GAP
MOBILE DEVICE MANAGEMENT
MAXIMIZING CX VALUE & ROI
CX & DX

www.pipelinepub.com
Volume 22, Issue 5

Enhancing the Customer Experience with the AI Help Desk

By: Mark Cummings, Ph.D.

Most everybody has known a frustrating experience engaging with an automated help system. With Generative AI (GenAI) this experience can be greatly improved while reducing the costs of its delivery. The key challenge is a well implemented GenAI model. To deliver a better experience, organizations need to move up the AI learning curve and develop optimal implementations. One of the best ways is to join a domain focused user group.

Background

Early attempts to reduce costs of providing help desk and customer service took one of two directions: off shoring or end user automated data entry.

Offshoring involves staffing in countries with lower labor costs. Unfortunately, it was often prone to poor user experiences because of accents, unfamiliar vocabulary, or general cultural gaps.

Automated data entry, whether voice or data driven, uses a fixed script to prompt a customer to input each piece of data serially. While this can be cost effective it too can deliver a poor experience. And such systems can result in a dead-end if an end user doesn't have the requested data or doesn't fully understand the request.

Similar automated approaches were also used in an attempt to drive down the cost of sales. They intended to reduce time per session by automating the front-end collection of basic customer data to optimize the interaction time with live staff.

Early AI Help Desk Implementations

With the introduction of GenAI new opportunities for reducing costs are emerging. The challenge is that we are early on the learning curve with this new technology.

Earliest implementations sought to lower training costs by having staff use an AI chat system to get key information to end users. This has two notable downsides: possible delays in answering the questions and delivering the wrong information.

The delay comes from the time it takes to manage the multi-step request loop: First, a staff member receives the prompt and engages the AI; then AI processes the prompt and communicates the inference to the staff member; then the staff member reads and processes the inference and finally, communicates the inference to the customer.

Wrong information can be more subtle. Often, this originates from not accurately structuring the prompt, i.e., asking the wrong question. For example, the author recently engaged with a help desk staff member about pricing, who then responded with the pricing of a specific plan. However, while it was an accurate description of a plan, it was not the desired plan that was eligible for a discount that the author qualified for. As a result, a sale was lost.

One of the easiest ways to implement an AI-based help desk is to use an AI intelligent agent to perform the same end user data entry used in earlier generations of help desk technology. This has the advantage of leveraging support staff with a lot of experience with these scripts and thus able to focus on the infrastructure changes involved in replacing deterministic code with an AI intelligent agent. The downside is that such implementations have the same tendency to produce a poor customer experience. That is to still get stuck if the end user doesn't fully understand what is being requested or doesn't have the requested information readily available.

Optimal AI Implementation

In prior generations of technology, the best customer experience routinely came from a properly trained staff member with a friendly demeanor, who was well informed, and had ready access to the necessary information.

GenAI presents the opportunity of creating an effective agent that has these characteristics while being much less expensive. That is, systems that fully leverage AI's powerful capabilities intended to maximize its performance with an implementation focused on replicating this more human-like behavior. This can be achieved by an AI application having an appealing persona that operates against specified objectives, algorithms, and constraints.

The development tools available to achieve those four types of characteristics (pleasing persona, clear objectives, algorithms, and constraints) are context management, prompt articulation, bridge building, and Large Language Model (LLM) selection.

Context Window

GenAI LLM’s typically have two ways to create input: prompt creation and loading of the context window. The context window is ‘background data’ that the LLM uses in conjunction with specific prompts to produce inferences. Fully specifying a personality can be done in a prompt. However, it may be easier to get the desired results when creating a personality when there is a very detailed description in the context window.

Giving that personality description a name that can be routinely referenced can also be helpful. If this personality approach is used, there may be an option of informing the customer that they are interacting with AI. Some environments may require such notification. In environments that don’t require it, there is a choice to be made.

Many LLM’s deliver the results of each inference session into their context window. This can amplify errors and increase the probability of hallucinations or inaccurate results. So, consideration should be given to how to manage the context window over time. There are many strategies ranging from erasing and reloading the base (including personality) context window at the end of each inference session, to not deleting any information in the context window until it hits an overflow threshold and then allowing the algorithm the LLM uses to make space.

Prompts

The prompt describes to the LLM what is desired. The more explicit and detailed the prompt is, the better the result. The prompt doesn’t have to be written in a programing language, Boolean logic statement, mathematical equation, or other structure, though those types can be used. The fundamental objective is to make it as clear as possible to the LLM what the desired output should be.

The best way to do this for an intelligent agent is to describe the Objectives, Algorithms, and Constraints for the agent.

Objectives for a help desk can be generic, and some will be specific. For example, a generic objective might be something like, “Always try to make the customer comfortable.” Or “Always try to understand exactly what the customer is saying.” Domain specific objectives generally have to do with key aspects if the domain.

Algorithms are ways of describing processes that the intelligent agent can or should use to achieve the objectives. Some examples of algorithms include the following. Using the data entry scripts described above. Repeating back to the customer in slightly different wording than what the customer says and asking if that accurately reflects what the customer was trying to address. Some algorithms may be mathematical or Boolean (if then, else chains, etc.). In the Boolean case, one set of algorithms may include conditions where the agent transfers the session to a human staff member.

Constraints are limits on the action of the intelligent agent. Examples might include the following. “Never express, or appear to express, frustration with the customer.” “Never quote a price over $500”. “Never say that something is the organizations’ fault.” etc. Constraints may also be expressed in Boolean terms. Constraints may be expressed mathematically in some pricing or terms of conditions discussions or in highly technical domains.

Bridges and Umbrella Data Model

Bridges are the subsystems that intelligent agents use to connect with other subsystems that they obtain information from, send information to, or seek to control. The bridge provides data translation, API implementation, etc. to do this job. There may be many bridges because organizations often have sub systems that came from different generations of technology, cultural contexts, and ways of representing data.

It is generally helpful to have an overall data model with standard terminology. This overall data model is called an Umbrella Model. The Umbrella Data Model is a superset of all of the local data models. The bridge translates from/to the Umbrella Model and the local data models. An example of such is the use of Fahrenheit and Celsius measures of temperature. The Umbrella Model may represent all temperatures in Fahrenheit while some sub systems local data model represents temperature in Celsius. In such a situation, the bridge would translate Celsius into Fahrenheit for the agent and then back to Celsius for the sub system.

Choice of LLM

Some might think that the choice of LLM should be first in the development process. However, it is important to develop enough of the application to understand what is required of the LLM. At the current stage of the technology, it appears that smaller and simpler LLM's have less of a tendency to hallucinate. A smaller model also requires less resources. An understanding of the application will help drive this. It also can help to divide an application into several separate intelligent agents that interact with each other. This can have many benefits including allowing each of the LLMs powering each of the agents to be smaller.

AI Infrastructure Alternatives

There are two basic approaches to AI application infrastructure: Edge or Data Center.

There are a broad range of local hardware alternatives. The Apple Mac Mini or Mac Studio and similar Windows-Linux form factor machines provide an interesting alternative. They can be loaded with an intelligent agent AI help desk system designed to be plugged into the existing help desk workstation. It can be designed such that it doesn't take technical expertise to plug it in. This allows the AI to be installed quickly and easily.

Maintenance is simplified by considering the installed package a bundle and not planning on complex software updates in the field. This provides a graceful path to prototype, then determines the optimal mix of intelligent agents and human staff.

Generally, local operation involves using an open-source model. Many of the current open-source models come from China leading to concerns about privacy and intellectual property.

There are two types of AI data centers: IaaS (Inference as a Service) and organization dedicated. An IaaS provider has the advantage of taking on the installation and support of the LLM itself. This may include automatic updates to new more advanced LLMs. Such updates can be a double-edged sword as they may produce unexpected behavior changes. IaaS providers also have the disadvantage of potential problems with privacy and intellectual property leakage.

Organization dedicated AI data centers assume responsibility for LLM installation and maintenance, but they don't suffer from LLM version control, privacy, and IP leakage problems. In some cases, large organizations may be able to license LLMs from frontier LLM developers. Others will use open-source models. Some large companies have talked about developing and training their own LLMs. How practical is open to debate.

Over time, small organizations are likely to use the edge architecture. While large organizations use the more centralized data center.

Security

Some may be tempted to think that by taking the staff member out of the help desk equation, the security risk of social engineering and other forms of attack may be reduced. Unfortunately, this is not the case. Special consideration needs to be given to security.

Role of User Groups

Because GenAI technology is so new and evolving so quickly, we are still early on the learning curve in how best to use it to achieve optimal results. As a result, it is important for developers, deployers, managers and users to find a way to move up the learning curve quickly. One of the best ways of doing this is to become a member of a user group focused on the technology in your application space. One organization working on creating such domain specific AI user groups is the AIWG. Such user groups provide a way for people working in the domain to share lessons learned, best practices, recent developments, etc. This can be very effective in moving up the learning curve.

Conclusion

With GenAI, the help desk customer experience can be made better while reducing costs. To capture the opportunity, organizations need to move up the learning curve and develop optimal implementations. One of the best ways to prepare to do this is to join a domain focused user group.