

Virtual Agent Product Outline

Problem Statement

A well-developed and maintained virtual agent will help users solve problems and complete tasks on their own with little to no human involvement at any time of day.

An assumption: This can result in decreased time Veterans spend waiting for an outcome, and ultimately decreased call center calls – especially for calls involving simple questions and/or solutions.

Ultimately, the problem we want to solve with our virtual agent is to help the Veteran self-serve and find information more easily. While we hope that this may also reduce call volumes, the correlation of the availability of a virtual agent as another channel fielding questions and a reduction in call volumes has not yet been proven to exist. Because of the amount of content that is available and needs to be rewritten in conversational format for the chatbot, we want to target starting with smaller targeted content.

The areas we want to address:

1. Giving the Veteran more direct access to the dense content available at va.gov
2. Providing the Veteran with another channel to interact with the VA that is available all day

Proposed Solution: (PoV) A Learning Bot

Proof of Value (PoV): What is the least we can build into a bot that will be valuable to Veterans and allow us to learn more about what users want from a virtual bot?

The primary goal of this phase is to gather knowledge directly from Veterans (via conversation monitoring, analytics and usability studies) to further finesse the virtual agent's capabilities moving forward. During this phase, we will build a "Learning Model" virtual agent. Responses will focus primarily on breadth of topics over depth in one area. We understand that the virtual agent cannot answer every possible question Veterans have. To that end, we want to gather feedback on what users want to know the most about so we can continue to train our bot in the right areas.

Goals for the PoV:

1. Create a conversational tool to help Veterans self-serve through increased awareness of existing website resources.
2. To learn if/how the Virtual Agent is valuable to the Veteran. This will help us decide the path forward.
3. Understand the effort of managing a virtual agent, including skills, team allocation and content management.

Phase 1: Training the Dialogue Model

The conversation models need to be trained via two formats:

1. Intents and Entities (what the user wants)

2. Responses (what the virtual agent can surface as a response)

For Phase 1 we will look for the following high-topic, and draft high-level (**unauthenticated**) responses only for the following areas:

1. Va.gov Benefits (education, health, etc.)
2. Handling Veteran Crisis scenarios (presenting phone number)
3. Covid Questions

Please Note: We expect many questions to be too complex for a virtual agent to answer. We will surface the best “here is how you can help yourself” response matched to an understood intent.

In Phase 1, the virtual agent will not (always) try to disambiguate a question via extensive conversations with the Veteran. If the virtual agent is not able to answer the Veteran’s question, it will result in a phone number being displayed.

Virtual Agent Phase 1 Features:

1. Location TBD, somewhere on va.gov
2. Free form chat experience to the users
 - a. Greetings & “Chit Chat” dialogues
 - b. Conversations will be logged for analysis
 - c. Handling errors in conversations
 - d. System wide technical errors
 - e. Understanding user intents
 - f. Collecting feedback- embedded feedback survey
3. Handling Veteran Crisis scenarios
 - a. Present users the number for Veteran Crisis hotline and disable chatbot if a crisis keyword is used
4. Knowledge Management Content use
5. Connecting with Covid Chatbot
 - a. Allow users to ask questions about Covid and present a link to go to the Covid Chatbot

Phase 2 Approach:

- Gather and access feedback from live users to refine our roadmap
- Connect with a live agent from the chat
- Partner with LOB’s to identify additional use cases for the Virtual Agent
- Build features to enable users to authenticate and perform tasks (ie. pay bills, process claims)

Assumptions

- We will have access to the Government instance of the cloud and will be using a commercial instance till it is ready- We are assuming the GCC will be ready by June
- We will build the Virtual Agent learning model using PVA, but we acknowledge there are some technical limitations in regards to version control and automated testing

- *We will launch to a small subset of users for a limited study*

Measuring Success

KPIs depend on the roll-out plan. In an ideal scenario, we would release the POV to about 5% of Va.gov users, and measure the following:

- Engagement rates
 - How many users interact with the new virtual agent?
 - How many left without interaction?
 - How many users ask multiple questions in one session?
 - How many ask the same question (in variations) during one session?
- Satisfaction rates
 - How many times was the virtual agent able to answer a question?
 - How many times was it not able to answer a question?
 - How do veterans feel about the interaction? (feedback)
 - What could be improved?
 - Are they likely to use this again?
- Perceived human-requirement
 - How many users directly ask to speak with a human (e.g., phone number) ?
 - How many still want to speak with a human after an interaction?
 - How many posed a crisis-related statement/question?
- Content requests
 - What are the most asked about topics by users?
 - What types of tasks did users ask the bot to do?

Things we've learned

PVA Technical Analysis

- What do we lose if we go PVA over custom development?
 - Expect that long-term maintainability of the core bot will be significantly harder if it keeps getting more complex.
- What do we gain if we go PVA over custom development?
 - Probably very smooth integration with omnichannel later
 - Probably very smooth integration with any of the out-of-the-box capabilities that Microsoft already offers
 - Less/no dev effort required for some core dialogue sequences (i.e. welcome, start over, escalate)
 - Expect a quicker working version of the final product
 - Expect more analytics stuff out of the box - transcripts, etc
- What do we think won't change much either way?

- We expect that we still have to potentially write software to get FAQ content into QnAMaker
 - Complicated multi-turn steps will have to be code (i.e. checking claim status)
 - We expect we still have to do a custom frontend to be sure accessibility concerns are met
 - We expect a need to draft content directly (instead of automation)
 - Why does maintainability suffer with PVA if complexity keeps going up?
 - Version control is severely hampered
 - Harder to analyze differences between two versions of the bot (i.e. answering questions like "why did this bug occur?")
 - Maybe difficult to release with unfinished work inside it (i.e. topic A is ready but topic B is not)
 - We should invest some dev effort in giving us the ability to roll back / restore from older versions
 - Unit testing is just not possible
 - A. unit tests are traditionally the most cost-effective testing tool.
 - B. May be able to write higher-level / more expensive tests using power automate?
 - C. expensive and may require dedicated dev effort
 - D. Downsides to no/few tests:
 - E. refactoring becomes very difficult and risky.
 - F. "Simple" bugs can leak through.
 - G. We might become reliant on manual testing which is expensive and can slow down release cycles.
 - H. Chances of regression are high.
-

Release Plan

- *Current Launch: July 2021*

Key Decisions

1. Using PVA to build and train our bot
2. Release in the July timeframe based on GCC