Ampersand

&

Ampersand

Executive Summary

Ampersand is a conversational agent leveraged for data science. What makes Ampersand different is that it handles context deficits inherent to "AI" assistant bots by having a reciprocal, interrogative conversations with the user (think ChatGPT meets 20 Questions). **This idea on the merits is something I have great reservations about**: User-forward interfaces work because the individual knows what they want to get out of the conversation more than anyone else. Making assumptions about user intent is either the X-factor of an application ("*it just works!*") or it's the worst experience in the world.

It all comes down to my definition of success. It's easy to create an AI that exposes endlessly but learns less effectively; given the tools, environments, APIs, data management, etc., how do you even find the ideal balance? What constitutes the "most aggressive" learning behavior is still open to interpretation, and I need to define clear success metrics. Moreover, a truly novel learning strategy could lead to behaviors and outcomes that are hard to predict. I don't know what I don't know.

> ESSENTIAL IDEA
>
> Ampersand is a web app. Users sign in and have text-based conversations. Like ChatGPT, the main conversation populates top-to-bottom, but when users want clarification or Ampersand brings up something they don't know, users select a "magic highlighter," specify the parts of Ampersand's answer they don't understand, an input box appears for users to comment or clarify, and an ancillary conversation panel opens up horizontally.
>
> Aggregate user data could be a novel way to help Ampersand understand intent by comparing the nature of the initial response versus how many parts of the following answer needed to be clarified. Ampersand could then modulate the depth and specificity of each succeeding answer.

The due date for this project is March 12th at 12:00 pm. Based on the outline of this project with the resources I have listed below, I need to know how much of this is feasible by the deadline.

- I am predisposed to big-picture thinking and often get lost in the minutia because of this. It's completely unrealistic, but how I plan to structure this project is outlined in:
    - `architecture.ipynb < docs <` GitHub
- I am at the beginning of my career in terms of understanding software development. I am extremely concerned about where I have placed things on the hierarchy. You will clearly see from this graph that I don't know what I'm talking about.

- DU gave each of us an Azure Subscription for Students. I have $100 and I've spent 11 cents. I might ask the University to provide more but I don't know the process yet.
- The bulk of this project (if not all of it) needs to be in Python. My IDE is VSCode, the changes from which are committed to the GitHub, which then builds in an Azure Web Service called "asterisk."
- I want to keep as much of this project within **Microsoft and Azure** as possible. As you will see in the GitHub, I have been unable to successfully build Ampersand within an Azure environment. I suspect this is because of the versions of Python I have installed on my computer, and I want to be delicate about how I fix this without bricking the functionality of my environment on my MacBook. Also, I need help clarifying when and how to implement virtual environments.
- I have applied for **Microsoft Cognitive Services** and **Azure OpenAI,** but have not received an email or anything and it's been almost a week at this point. Honestly, I don't know if it's even possible to do any meaningful ML by March 12th.
- At my professor's suggestion, I have been using **Microsoft Bot Framework Composer**. I don't like using it and it's more difficult to use it than actually get what I need out of it. But if I were to not use it, I need to figure out (1) the interface, and (2) dialog control (the secret sauce that governs how Ampersand becomes a conversational chatbot). I am leaning towards **Django REST Framework with PostgreSQL** (rather than Flask). What do you think?
  - I am still learning about Microsoft's Bot Framework SDK (which is the parent to BFC). What I could really use your input on is whether I should choose .NET or node.js
- I am in the process of mocking up the UX in Figma. I'm also considering making custom CSS styles and just bought a subscription to Spline.
- Ampersand is an enterprise bot that comes with industry-specific knowledge that it then builds upon with proprietary information uploaded to my service. **For the purposes of this assignment, the industry I have chosen is: telecommunications.**
- As for the data science functionality, this is another rabbit hole in and of itself. I need help distinguishing between the data science Ampersand offers from the data analytics I am running to measure the application's functionality. On that matter, I need help with designing the following behaviors and functionalities:
  - **User employee uploads**: The types of files Ampersand is capable of inspecting and how I translate this information to the bespoke business intelligence it continuously builds. How would it articulate this to managers?
  - **Conversation management**: Microsoft documentation covers this more broadly **here**.
  - **Memory and learning**: I have gone down many rabbit holes.
- For the purposes of this project, I am more than happy with Ampersand providing the most superficial, like object-permanence-level-of-obvious answers.