ManipulatorAI

Stage

- The first direct communication with a potential customer.

Goal

- Proceed a new potential customer to the onboarding agent.

Description

Suppose I have a facebook page for consumer products and I have put on different ads there. Suppose another user clicks on our product or likes it or comments on the ad then I want the AI agent I need to build to be triggered at this comment like or click. Now, what I need is to build an AI Agent as a micro-service which will conduct an introductory conversation with the user where it will try to cordially greet the person and then try to talk to him more about the fact that the user interacted with

our ads and if the person is interested in our product. It will also try to convince the user in a humanly fashion to actually try to make him convinced to register onto our site. Suppose, For the portion I already have built another agent named onboarding agent.

The background knowledge is that I am building a whole platform where there will be multiple agents which will work for both attracting the consumers and also the business owners are my clients too who can observe the whole thing from our platform. Basically what we are trying to do is remove the human interaction or human-powered communication to attract the consumers for the business owners and instead replacing them with different AI agents built by us. Now here the main goal of the agent I am trying to build will do couple of things. It will first be triggered at the event of any kind of interaction of the consumer to the ads of our client's page and then initiate a conversation. Then it will gradually do what I described up above to properly guide the consumer through the process of convincing him or her in a humanly fashion by showing the plus or key points and relevant details about the product he or she showed interest in and also related products too if that person wants to browse for something else too. If the person seems uninterested, it will try it's best to show the good points of the products to make the person interested in buying the product. Then if it convinces the person to onboard or create an account with us to proceed with the browsing in our website, then it will hand the whole thing over to another agent we have built which is the Onboarding Agent and it will handle the rest from this point onwards.

Control Flow

Overview

- The facebook or instagram webhook will send a notification as soon as an user interacts with the ad or our post (through like, comment or clicking on the ad)

- There will be a Redis queue for us to be able to process the things as a queue so that no data is lost in process if multiple request come at once or while processing one request. This keeps the data receive and processing part separate.

- Then the AI Agent will start conversation with the potential consumer. [We will start the conversation far later in the workflow but for now this is just an overview]

- The LLM behind the conversation will be carried out by API calls. We have access to Azure account from where we will use OpenAI models for generating the responses.

Product Database [Knowledge-Base]

- Now to start the conversation the key-point will be towards a warm greeting and then the focus will shift to the specific product / related ad or post the potential customer interacted with. So the later responses are going to focus on what product(s) is/are at discussion here. To integrate that, first we will keep a separate knowledge-base with all the product informations and description of all the products we own. We will build this knowledge-base using postgresql where we will make a central schema containing
    - product_id (Unique),
    - product_attributes(JSON data which will contain the key-value pairs of all the attributes like - 'Price': '$20'. These attributes can be a lot of things ranging from colour, price, type etc),
    - product_tag,
    - product_description.
- The product_tag column is very very important for us because it will contain multiple tag-words for the corresponding product which will be a big factor in cross-referencing products.
- Lastly, the product_description column will have the detailed description about the product.

Branch of Control

- Next stage that we have to take into consideration that the conversation can be initiated by two different ways.
    - One is that the webhook sent us notification and our AI Agent approached the customer in a human-like manner pretending to be a sales manager trying to promote. [Manipulator]
    - Another is that the user himself/ herself actually sent us an inquiry or texted us and we have to pass the text through the LLM and then have to respond to the text our potential customer has sent us. [Convincer]

Manipulator

- For the first branch of the control flow [I will name it the Manipulator branch] (where the user interacted with the ad or post himself), we can directly get to know which product he / she is interested in.
- We can simply take the product description of that object from our database against that unique product ID and use that for future use.

Convincer

- For the second branch [I will name it Convincer branch] (where the customer reached out to us with a text), we have to take in the text that the customer has sent us and feed that into the LLM and prompt it to actually generate the key-words related to our products in the whole text he sent us. So, I will create a sub-system named **keyRetrieve**r and it will work like a function calling.

keyRetriever

- Here the parameters of my function will be the prompt or text our potential customer has sent us and also a summary that we have kept of our client's business about which kind of product is being sold by us.

- This will help the LLM to infer from the text the consumer has sent to the page that which of the whole text of the consumer it should focus on. Because the part on which it should focus on is the one which is related to our products.

- The output of this function will be the key-words from the whole text that the LLM will extract using the help of the context of the product summary that we will provide too alongwith. This can also be compared to the process of 'topic selection' where if given a whole passage the task is to give the passage a topic. We will instead feed the whole text sent to us by the user along with the context of our products summary and the LLM will return us the related or important key-words.

- After getting the output key-words from keyRetriever, our agent will do is cross-reference them with out database to find matches. For

this, we can do one thing. We will keep another sub-system called tagMatcher.

tagMatcher

- the Sub-system will do another function calling where the parameters will be the key-words generated by the previous function and the product_tag column elements from our database. Then our LLM has to infer which one the elements in our database match closest with the key-words generated by the previous function output.

- We can keep an evaluation score of how correlated matches we are getting. After this we can use a similar procedure like the top-K query type of system where it shows the top-k retrieved products over >=80% correlation between the key-words and the product_tags.

- Output of this sub-system will be the unique product_id(s) of the matched products via the tag-matching.

Now question arises why we need this sub-system?

1. Because to continue the conversation with the potential customer, we need to find the relevant and related product(s) the person is looking for.
2. And secondly, we need another sub-system to work as the main modular section of our whole Agent. This will actually perform the conversation with the client.

- The parameter of this sub-system / module will be the conversation with the user so far. If the parameter has nothing included then it means the start of a fresh conversation where we are the ones initiating the conversation.

Before we go into the conversation phase, we got to focus on some important things:

- Initially, it will take into account the product that the person interacted with.
- Secondly, depending on the control flow it will do either of two things.
    - One is if the user interacted with us [Manipulator Control Flow] then we already have the product_id(s) that is/are related with the post. So it will already have the product_id(s) based on which it has to actually continue the conversation.

    - Another is if the control flow went to the Convincer Control Flow (the potential customer texts in our page) it will cross check the Knowledge-Base for the existing product description using the two keyRetriever and tagMatcher sub-systems we built previously. The first one will get invoked based on the text that the user sent us and generate key-words from there. The second one will then take those key-words and then generate the product_id(s) that is/are related to what the customer's text is about.

- Now regardless of the workflows we have the product_id(s) that we need.
- Then we have to make a query to our database to fetch the product_description corresponding to the product_id(s) and contain it in a list of strings.
- Coming to the most important part of the workflow, now we have to generate a series of prompts for smaller modules and a master prompt which will focus on convincing the user to buy the products related. We will feed these prompts to the Azure's OpenAI model and it will produce us the responses which we will send to the user.

Prompt Engineering

- Here the prompt will be custom engineered so that instead of a generic response the OpenAI model gives a more custom-tailored fine-grained vibe for our use.
- The prompt engineering also has to contain couple of other things-
- For our first time responding / interacting with an user, a welcoming prompt will get triggered only for once. The prompt will contain two parts. firstly, The welcome part which should be warming so that it feels more human-like. and secondly, [regardless of coming from Convincer / Manipulator controlflow] based on the product_id(s) we already have the product_descriptions in the list of string I listed before. This string will be also fed to the welcoming prompt for the LLM to summarize the genre in one or two lines. So the prompt will start with a warm welcome and then in the end it will contain the summary of the genre which was previously generated to ask the customer if we can help with the interest of the person shown for the specific type of products.
- Secondly, it should try to convince the human being in a very polite yet persuasive manner.
- Thirdly, it should persist a bit if the user seems uninterested.
- The next one is a bit important and we will do it in case the customer seems uninterested. We will check our Knowledge-base where we kept all our product related datas and try to fetch some data which has 70-80% correlation and ask if they are interested in our other products which are somewhat a bit similar.

- Lastly, if the user still seems uninterested then it should show gratitude to the user for interacting and express that we will be glad to serve him/ her in future and to stay tuned with us.

- The conversation will be custom tailored only according to our products and also the prompt engineering goals mentioned above because the goal is to attract the customer to convince him to buy our products.

Database

- This whole conversation will be saved in a noSQL database. For this, we will use MongoDB. We need the conversation history for the whole on going conversation since everytime we evoke the OpenAI model, we have to feed the context to it from the previous chats or responses of the user and our responses too. So everytime we are making an API request from the Azure platform's OpenAI model, we are feeding the whole context which we are saving in our MongoDB.