

How we implemented Generative A.I. into our business.

Rich Swier

r@roofre.com

Constellation Roof LLC

January 3, 2024

Summary

This document serves as a detailed guide on implementing AI applications in business. It shares our company's success in using machine learning and Large Language Models like ChatGPT, focusing on how AI can improve internal operations and enhance products. The paper divides AI into two categories: Active and Passive, each with distinct roles and strategies in business applications. It addresses the challenges in deploying AI systems and suggests best practices, especially in managing the balance between risks and rewards. The document concludes by discussing the business impacts of AI, particularly in generating revenue and reducing costs, and offers resources for developers working with ChatGPT and its API.

The document also discusses various challenges in leveraging AI for business. These include training AI for accuracy, managing AI's human-like interactions, dealing with the 'black box' nature of AI, and addressing security, compliance, and intellectual property concerns. It also explores the importance of oversight and feedback in AI applications, the risks of misuse, and the challenges in integration and user acceptance. The paper advises on strategies to mitigate these risks, such as implementing training and security policies, disclosing AI interactions, and optimizing user acceptance.

Overall, the document is an insightful exploration of the nuances and complexities of deploying AI in business, offering practical advice and strategic guidance to companies looking to harness the power of AI technologies.

Introduction.....	3
Business Challenges.....	4
Our Implementation of Generative A.I.....	7
Technical Architecture.....	10
Bot Framework.....	13
Prompt Engineering.....	16
Fine-Tuning.....	19
Using Functions.....	21
Threads (Conversations).....	22
Other Use Cases to Consider.....	23
Dealership.....	23
Salon / Spa / Health Services.....	23
Food Service & Hospitality.....	23
E-Commerce / Retail.....	24
Insurance.....	25
Home Services.....	25
The Impact on our Business.....	26
The Launch of R.A.I.A.....	27
Conclusion.....	28
Appendix: Resources.....	29

Introduction

In this document, we explore how Artificial Intelligence (AI) is like a participant in conversations, offering two main ways to interact: actively or passively. With Active AI, it's like consulting an expert for advice and decisions. Passive AI, on the other hand, works more like a response mechanism to user inputs, similar to how ChatGPT operates. Understanding these roles is key for applying AI effectively in various business scenarios.

Passive AI relies on user-driven conversations. For example, in the ChatGPT framework, the user starts and steers the dialogue by asking questions, and the AI responds. This setup is pretty straightforward and often seen in web chats via browsers or mobile apps. Companies like OpenAI provide easy-to-use platforms for creating AI assistants that can be customized to meet specific business needs, making Passive AI a versatile choice for various applications.

However, the success of Passive AI largely depends on the user's engagement and clarity of intent. It's like using Google Search: the user must initiate the conversation by asking a question. This approach is especially effective in customer support or information retrieval scenarios.

Passive AI is most valuable when it replaces less efficient or costlier methods of user interaction. It assumes that users will engage with it digitally, through websites or apps, rather than traditional methods like customer support calls.

Active AI, in contrast, takes the lead in conversations to achieve specific goals. It can start interactions through emails, texts, notifications, calls, or direct messages. Active AI needs special training to align with its objectives. For example, in a customer satisfaction survey, it would first get permission to engage, then ask a series of questions. This involves more complex coding and design, as it not only starts the conversation but also guides it to achieve specific outcomes. The data collected can then be integrated into systems like CRM for further analysis.

Active AI scenarios also include tracking specific data, like customer ratings, which may require custom coding within the GPT framework. It's about finding the right balance in the application's design, whether using a single AI for multiple tasks or managing several simpler bots.

A critical aspect of using AI is deciding when to switch from AI-led interactions to human involvement. In some cases, AI can handle the entire conversation, while in others, it might start the conversation and then hand it over to a human. This handoff ensures oversight and less reliance on AI alone. For example, an AI could qualify a sales lead and then alert a salesperson to take over. Both scenarios underscore the importance of monitoring AI performance and integrating human feedback, particularly important in Active AI applications.

Business Challenges

After exploring the complex architecture needed for Active AI, let's focus on the business challenges it brings.

Businesses can use AI in two main ways: for internal operations and for external, customer-facing services. Using AI internally is generally less risky because access to these systems can be controlled and their impact is more manageable. However, when AI interacts with external parties like customers or suppliers, the risks are higher due to its public nature.

Despite these risks, the growing expectation for businesses to incorporate AI into their offerings is becoming more pressing. Companies must carefully weigh the risks of not using AI against the risks that come with adopting these technologies.

In AI deployment, there's a clear link between risk and reward. Low-risk AI uses tend to offer smaller rewards. For example, an AI chat tool for employees to learn about company policies might not save much money, but it can improve the employee experience. This is a low-risk, low-reward situation.

On the other hand, using AI to automatically create and send priced proposals carries higher risks and rewards. The main risk is the AI setting the wrong prices, but the potential benefits are significant. Such automation could greatly reduce the costs associated with human staff doing this work. The most effective approach often combines AI with human oversight to increase both efficiency and accuracy.

In the context of AI application deployment, it's essential to consider various risks:

1. **AI Training and Accuracy:** Using a base Large Language Model (LLM) gives a wide range of knowledge, but it might not fully understand your specific business needs. Improving the AI with

fine-tuning or prompt engineering is possible, but there's a risk of the AI giving outdated or incorrect information, like old website content or incorrect pricing. This can also lead to inconsistent answers, particularly with things like product prices, and not enough training can result in a bad user experience.

2. **AI's Human-Like Interaction:** Many people are used to interacting with computers, but AI's ability to mimic human conversation can sometimes be an issue. If users think they're talking to a human and later find out it was AI, they might react negatively. This is different from traditional automated systems, which don't pretend to be human.

3. **AI as a Black Box:** When you choose an LLM for your AI application, you face a 'black box' scenario. You have limited control over how the AI responds and sticks to scripts, unless you do specific training and programming. While human sales and support teams are driven by job security concerns, an LLM's broad knowledge and conversational skills can sometimes lead to inappropriate responses, especially if users don't know they're interacting with AI. To reduce this risk, you should do thorough testing with users.

4. **Security, Compliance, Intellectual Property:** Implementing AI comes with its own technology risks, including issues around security. If you're training AI models with sensitive data, you need strict access control. Also, you must consider compliance and intellectual property just like with any other software product.

5. **Lack of Inbuilt Oversight and Human Feedback:** It's important for AI applications to have oversight and improve based on human feedback. Each interaction with the AI should be monitored, sometimes with direct user feedback, and other times with internal reviews.

6. **Misuse and Abuse:** AI applications like GPT that face the public can be misused or abused. Users might try to use the bot for things it wasn't meant to do, like getting around paying for unrelated services.

7. **Integration, Cost, and Dependency Risks:** Relying on AI/GPT and third-party APIs can lead to risks like system downtime and fluctuating costs. These issues need to be carefully managed.

8. **User Acceptance:** Even though ChatGPT is growing quickly, changing how users behave and experience things can be risky, particularly if they're not ready for it. It's important to remember how long people have been using the web and the slow shift to using mobile apps.

Addressing the risks associated with deploying AI applications involves strategic measures. Here are some tactics to mitigate these risks:

1. **Establish a Training Policy:** Set up a process to carefully select data for AI training. Be cautious not to use too much data, which might be incorrect or confidential. For the best AI performance, you might need to review and format data specifically for training your Large Language Model (LLM).

2. **Develop a Security Policy:** If you're using AI internally, you can adjust your current app policies to control who has access. Since interactions with LLMs aren't stored, this reduces the risk of exposing public data. For better security, think about developing your app with an API and an extra layer of security, especially if your standard user experience doesn't track access or activities.

3. **Be Transparent About AI Interactions:** To prevent users from feeling deceived when they realize they're talking to an AI, make it clear from the start of the conversation that they're interacting with a bot. As AI becomes more common, people will likely get used to it and appreciate how it improves their experience.

5. **Incorporate Oversight and User Feedback:** It's important to add features to your app that collect feedback. This could be direct from users or through active learning, where the AI asks for comments on its responses. Feedback helps improve the AI model.

6. **Prevent Misuse and Abuse:** To stop people from misusing the AI, set limits on how often they can make requests and keep an eye on how often they use it. This should be done outside of the GPT model, in the app itself, and it can also help control unexpected cost increases.

7. **Make User Acceptance Smooth:** Use well-known methods like SMS and email to get people to engage with your AI app. This makes them more comfortable and helps them adopt the technology more easily. When adding AI to websites or web apps, check that it fits in seamlessly and doesn't interrupt the usual way users interact with your site or app.

To effectively manage the risks of deploying AI-powered apps, it's necessary to construct an application layer around the GPT. This layer should encompass functionalities for managing security, compliance, training, oversight, and human feedback.

Our Implementation of Generative A.I.

Since starting in 2014, our company has used machine learning (ML) to predict home sales and prices, providing valuable data to the real estate industry. Initially, we experimented with early Large Language Model (LLM) versions in 2017 but found they weren't quite right for our main operations. However, the introduction of GPT and OpenAI was a game-changer for us.

We discovered many ways to use Generative AI (LLM) in our business, which helped increase our revenue, keep our customers, and reduce costs. Our main focus shifted to improving our primary product by incorporating Generative AI for qualifying leads both inside and outside the company.

Before we started using Generative AI, we faced several challenges:

1. **Ineffective Lead Nurturing:** Our product didn't do a good job of nurturing leads. We tried using sequence-based marketing and external drip applications, but these were expensive and didn't work well. Traditional marketing methods weren't engaging our leads or encouraging them to interact with us.

2. **Poor Lead Qualification:** We didn't have a strong feature for qualifying leads, so we experimented with call centers. But the high costs and low success rates in making connections made this approach unfeasible. Also, the call centers had trouble training their staff and having meaningful conversations with potential leads.

Faced with these issues, we decided to add conversational AI to our product, moving away from one-way marketing strategies and the limitations of relying solely on human resources.

Our AI deployment aimed to achieve:

- AI's comprehension of our product offerings.
- Flexibility in guiding leads towards a call to action, like scheduling appointments.
- Prompt responses anytime.
- Asking pertinent questions for lead qualification.

- Communication through SMS, Email, or Web Chat.
- Seamless integration with our existing systems and sales processes.
- Scalability for client customization.
- Conversation scoring to reduce human involvement in qualification and monitoring.
- Scalable solutions for numerous clients.
- Cost-effective scaling.

We achieved our goals by building an application layer around ChatGPT, using their API. Here's how we approached each objective:

1. **Understanding the Product:** We used ChatGPT's Assistant feature to make sure the bot understood our product. We did this by giving it detailed instructions and uploading a formatted JSONL file with FAQs and User/Assistant Prompts..
2. **Encouraging Action:** We programmed the Assistant to ask questions that would lead the conversation towards actions we wanted to happen.
3. **Immediate Responses:** We set up the system to send instant SMS and email replies, activated when someone submits a web form.
4. **Targeted Questions:** We added specific questions to the Assistant's guidelines, mainly to help with scheduling appointments.
5. **Communicating Across Channels:** Our app, developed with the OpenAI API, handles both SMS and email communications. We use Twilio for SMS and a third-party service for emails.
6. **Integrating with Existing Systems:** We used the OpenAI API to link our AI bots with our current systems, allowing them to talk to leads over phone or email.
7. **Client-Specific Scalability:** We created a system that supports multiple AI bots or Assistants for each client, using the OpenAI API's functionality to create Assistants programmatically.
8. **Evaluating Conversations:** We set up a separate bot to review and rate completed conversations, integrating it with other functions of our app.

9. **Handling Large Volumes:** Our system is designed to manage thousands of bots, users, and clients. The only real limit is ChatGPT's rate limiting.

10. **Cost-Effectiveness:** The main expense is for the ChatGPT API, which charges per query. Based on how we use it, we expect the monthly costs to stay under \$500 if we deployed to all current customers..

In our project, we explored various features and how they can be applied:

1. **Using OpenAI Assistant:** This tool greatly sped up our deployment. Most of the prompt engineering is done in the Assistant's settings, which makes it easy to make changes in real time. If your situation requires linking to current applications or data, you can use the Assistant's functions feature to integrate these actions.

2. **Fine-Tuning in the Assistant's Settings:** This process is similar to prompt engineering but allows for using more data. The main advantages are:

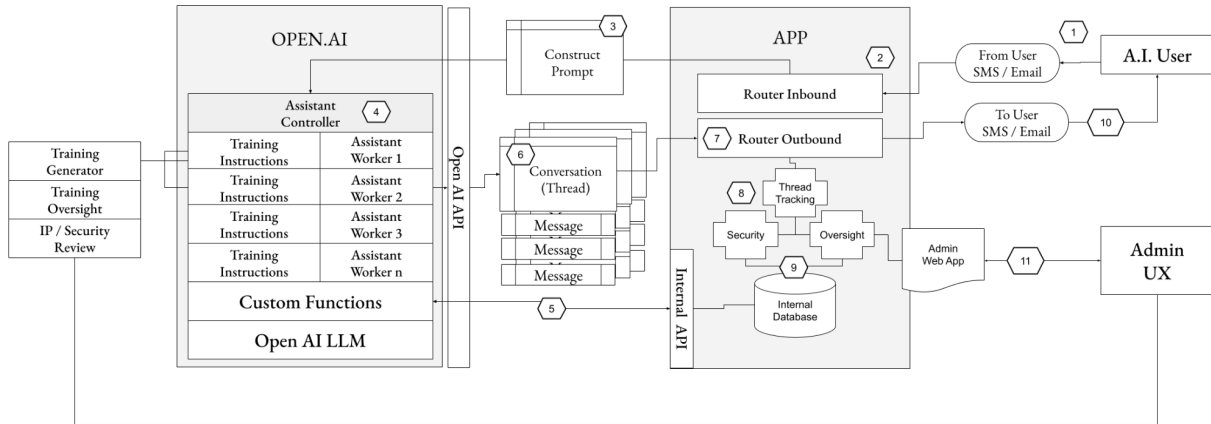
- Better quality results compared to standard prompts.
- Training with more examples than the prompts' limits.
- Using fewer tokens for shorter prompts, which saves resources.
- Faster response times.

We gathered all our Frequently Asked Questions (FAQs) from various sources into one spreadsheet. After our team reviewed it, we converted this data into JSONL format for ChatGPT. While we know this can improve how well the responses fit our needs, the full impact of this approach is still unclear. There's ongoing debate about the best way to use Fine Tuning—whether it's better for setting instructions and tone or for feeding in specific data to make the AI smarter. The general view is that Fine Tuning works best for guiding the AI in a way that's similar to using prompts, rather than for remembering exact facts.

In short, our detailed strategy of incorporating Generative AI effectively tackled significant business challenges and met our key objectives.

Technical Architecture

Diagram



Basic Flow of User Experience (see numbers in Diagram)

1. A.I. User (in our use case this was a lead) responding to the initial message.
2. Router Code receives the message from the User.
3. Message is transformed into Prompt and sent to the Open AI API (creates or updates Thread)
4. Assistant routes Message to appropriate Assistant and submits Prompt.
5. Open.AI LLM response with message (and potentially calls function)
6. Message is added to open Thread.
7. Message sent to Router Code Outbound.
8. Router interfaces Security / Oversight.
9. Results stored in DB.
10. Message delivered to User.
11. Admin able to monitor activity for oversight.

The framework diagram above provides some context on how we implemented A.I. into our product. For simpler use-cases, you may not require all of the components.

Creating a ChatGPT Assistant

The ChatGPT Assistant is versatile and can be used for a variety of purposes, simplifying the process by eliminating the need for more complex development. For detailed specifications, you can visit the OpenAI documentation. The Assistant has a character limit of 32,000 for its instructions, where you define the bot's main goals. If you need more than this, Fine-Tuning is an option.

When setting up custom instructions for ChatGPT, it's important to follow some best practices to get the best results:

- **Be Clear and Precise:** Make sure your instructions are straightforward and to the point. Avoid long and complex sentences that might confuse the AI. Instead, use brief and clear instructions that specify what you want the bot to do or achieve.
- **Use Examples:** Including examples in your instructions can really help the AI understand what you need. These examples can guide the AI to produce more accurate and relevant responses or actions.
- **Consider Special Cases:** Think about potential problems or common misunderstandings and address them in your instructions. Being specific about these cases helps the AI avoid mistakes and unwanted responses.
- **Iterate and Test:** You might need to try different versions of your instructions to see what works best. Don't be afraid to change and test various approaches to fine-tune the AI's behavior.
- **Backup Before Major Changes:** If you're planning big changes to your Assistant, it might be wise to make a copy of the current version first. This way, you can go back to the previous version if the new changes don't work out as expected.

In our situation, we listed specific questions we wanted the bot to ask, with clear directions to guide the conversation effectively.

Bot Framework

Using a Multi-Agent Framework, each agent is designed to perform a specific task. If you need to handle multiple tasks, you can send requests to different assistants. One assistant's main role can be to decide which other assistant to use in different situations.

For instance, you might have a system message like, “You are a helpful assistant that selects the right agent for a given task. Here is a list of agents and what they do.” The main agent uses a 'choose_agent' function to determine and then direct the user's request to the right sub-agent, each with its own set of functions.

In our application, we have a bot specifically designed to summarize and rate all conversations. This helps us decide which leads or conversations are most important to follow up on.

When you have one ChatGPT assistant activating another, it creates a multi-layered system of AI assistants. While setting up this kind of system can be intricate, it's very effective when properly implemented.

Primary Interface Assistant

- Role: This is the front-facing assistant that interacts directly with the user.
- Functions: It handles general queries, user interface interactions, and directs specialized requests to the appropriate secondary assistant.
- Capabilities: It should be capable of understanding a wide range of queries and determining the context or domain of each query.

Secondary Domain-Specific Assistants

- Role: These assistants are specialized in specific domains or tasks.
- Function: When the primary assistant identifies a query as belonging to a specific domain, it forwards this query to the relevant secondary assistant.
- Specialization: Each secondary assistant is fine-tuned or programmed to handle specific types of queries (e.g., technical support, sales inquiries, etc.).

Communication Protocol

- **API Calls:** The primary assistant communicates with secondary assistants via API calls.
- **Data Format:** Use standardized data formats (like JSON) for communication to ensure seamless data exchange.
- **Security:** Ensure secure communication channels to protect data integrity and privacy.

Here is an example format of the JSON that the primary assistant would send to the secondary assistant:

```
{
  "session_id": "123456789",
  "user_query": "My laptop has been overheating frequently. What should I do?",
  "context": "Technical support query regarding laptop issues.",
  "user_details": {
    "user_id": "user123",
    "user_preferences": {
      "communication_style": "concise",
      "technical_knowledge_level": "beginner"
    }
  },
  "assistant_details": {
    "primary_assistant_id": "assistant_xyz",
    "domain": "technical_support"
  },
  "timestamp": "2024-01-01T12:00:00Z"
}
```

For the primary ChatGPT bot to read a configuration file and understand the roles and capabilities of secondary bots, the process generally involves a few steps, considering ChatGPT's API-based operation:

1. Configuration File Design

- **Format:** Choose a format for the configuration file that's easy to parse, like JSON or XML.

- Content: The file should include details of each secondary bot, such as its name, description, capabilities, and any specific keywords or triggers associated with it.

2. Accessing the Configuration File

- Hosting: The configuration file should be hosted in a location accessible by the application running the primary bot, such as a cloud storage bucket or a web server.
- API Endpoint: If the file is hosted on a web server, create an API endpoint to fetch the configuration file.

3. Reading the File

- File Fetching: When the primary bot starts or at regular intervals, it should fetch the configuration file from its location.
- Parsing: Use server-side code (like Node.js for JavaScript environments or an equivalent in Adobe Coldfusion) to parse the file and translate it into a format the primary bot can understand.

4. Updating ChatGPT's Context

- Contextual Information: After parsing, the server-side application should pass the relevant information from the configuration file to the ChatGPT bot as part of its context. This can be done through the API call to ChatGPT, where you include additional information that ChatGPT uses to make decisions.
- Dynamic Updates: Ensure the primary bot can periodically update its understanding by re-fetching and re-parsing the configuration file.

5. Application Logic

- Decision Making: Implement logic in the server-side application that uses the parsed configuration data to help the primary bot decide which secondary bot to route a request to, based on the user's input and the bot capabilities defined in the file.

6. Security and Error Handling

- Security: Secure the access to the configuration file to prevent unauthorized modifications.
- Error Handling: Implement robust error handling for scenarios where the file might be unreachable or contain parsing errors.

Example Workflow:

1. User Input: The user sends a request to the primary bot.
2. Fetch Configuration: The application fetches and parses the configuration file.
3. Contextualization: The application includes this information in the API request to ChatGPT.
4. Processing and Routing: ChatGPT processes the request, uses the configuration information to decide the appropriate secondary bot, and the primary bot communicates this decision back to the application.
5. Forwarding the Request: The application then forwards the request to the chosen secondary bot.

By utilizing server-side logic and API interactions, the primary ChatGPT bot can effectively read and use the configuration file to manage and route tasks to various secondary bots.

This architecture leverages the strengths of both generalist and specialist AI models, offering a comprehensive solution that can handle a wide range of queries with a high degree of expertise.

Prompt Engineering

ChatGPT prompts are the instructions or questions you give to the AI to get answers. These prompts are made up of keywords and phrases that are designed to elicit a response. When you ask ChatGPT a question or give it a command, it replies as if it's part of a conversation. In Passive AI, it's the user who comes up with these prompts.

In Active AI, where the bot leads the conversation, you can use the API to set up prompts by sending system messages. However, the Assistant API currently doesn't have the capability to send system messages directly. As a workaround, you can input the message as if it were from the user. For example, you might enter something like, "I just answered 'xxxx' to your question 'yyyy', please don't reply but remember my answer." This way, the system treats it as user input but from the perspective of the user.

Constructing a Prompt

Every use case is different in how you may construct your assistant instructions and your prompts. The general structure should include

[Your persona]
[Your knowledge]
[Your traits]
[Steps to the task]
[Your task]
[Goal]
[Format]

Here's a clearer breakdown of how to instruct the bot:

- 1. Define the Bot's Relationship with the User:** Explain to the bot who it is in relation to the user. For example, you might tell it, "You are a sales assistant speaking with people interested in buying or selling a home."
- 2. Clarify the Bot's Role and Objective:** Make sure the bot understands its role in your business and what it's supposed to achieve. For instance, "Your job is to qualify leads for real estate agents by asking a set of questions." Providing this context is crucial for the bot to function correctly.
- 3. Specify What the Bot Should Say:** In most cases, the bot should be guiding the conversation. Clearly state what you want the bot to discuss or ask about. Place these instructions at the start of the prompt and use symbols like ### or "" to differentiate between instructions and the conversation context.
- 4. Instruct on Tone and Format:** Tell the bot how to frame its responses. Be specific about the desired tone, format, and style. For example, "Respond briefly, professionally, and kindly, using emojis where suitable." This approach is particularly effective for mediums like SMS, where a natural conversational style is beneficial. Be as detailed as possible in describing the context, desired outcome, length, and style of the responses.

5. **Be Direct:** No need to be polite with LLM so there is no need to add phrases like “please”, “if you don’t mind”, “thank you”, “I would like to”, etc., and get straight to the point.

6. **Define your Audience:** Integrate the intended audience in the prompt, e.g., the audience is an expert in the field.

7. **Outline Sequence of Prompts:** Break down complex tasks into a sequence of simpler prompts in an interactive conversation.

8. **Don’t use Negative Language:** Employ affirmative directives such as ‘do,’ while steering clear of negative language like ‘don’t’.

9. **Use Explain Prompt:** When you need clarity or a deeper understanding of a topic, idea, or any piece of information, utilize the following prompts:

- Explain [insert specific topic] in simple terms.
- Explain to me like I’m 11 years old.
- Explain to me as if I’m a beginner in [field].
- Write the [essay/text/paragraph] using simple English like you’re explaining something to a 5-year-old.

10. **Give Examples:** Implement example-driven prompting (Use few-shot prompting).

11. **Use Special Formatting:** When formatting your prompt, start with “###Instruction###”, followed by either “###Example###” or “###Question###” if relevant. Subsequently, present your content. Use one or more line breaks to separate instructions, examples, questions, context, and input data.

12. **Tell the Bot what to do:** Incorporate the following phrases: “Your task is” and “You MUST”.

13. **Use Negative Reinforcement:** Incorporate the following phrases: “You will be penalized” for anything you absolutely don’t want the Bot to say or do.

Fine-Tuning

Fine-Tuning allows you to enhance the assistant's instructions, making it more attuned to your specific business needs and situations that might not be covered by the general knowledge in a Large Language Model (LLM). Here's what you can include:

- **Industry-Specific Terminology:** Incorporate jargon or technical language unique to your industry.
- **Frequent Customer Questions:** Add common questions your customers or users ask.
- **Details About Products or Services:** Include key information about what you offer. Be cautious not to add details that change often to prevent spreading wrong information.
- **Handling Vague Questions:** Teach the bot how to respond to unclear queries, like those about pricing where you might not want to give specific details. You can program it to give a different kind of response instead.
- **Relevant Internal Information:** If there are important texts or documents within your company that the bot should know, include these to give it better context when answering prompts.

The format of the file to upload for fine tuning we used is JSONL.

The JSONL (JSON Lines) format for fine-tuning data is crucial for training models like ChatGPT. Each line in a JSONL file is a separate JSON object, making it an efficient format for processing large datasets. Here's a breakdown of the format:

1. Basic Structure: Each line in a JSONL file represents a single training example. The file itself is a collection of such lines.

2. JSON Object: Every line is a JSON object containing key-value pairs that represent different aspects of the training data.

3. Key Components:

``prompt``: This key holds the input text that you want the model to respond to. It can include questions, statements, or any other form of text.

``completion``: This key contains the model's expected response or output to the corresponding prompt.

4. Example Entry:

```
{"prompt": "What is the capital of France?", "completion": "The capital of France is Paris."}
```

In this example, the prompt is a question about geography, and the completion is the correct answer.

5. Multiple Entries: A JSONL file for fine-tuning would contain multiple such entries, each on a new line.

6. Consistency: Ensure consistency in the format of the `prompt` and `completion` across the dataset to facilitate effective learning.

7. No Extra Commas: Unlike standard JSON arrays, JSONL does not use commas at the end of each line/object.

8. Size Consideration: Each line should ideally represent a complete instance for training. However, be mindful of the size of each instance to prevent truncation issues, especially if you are working with lengthy prompts or completions.

9. Encoding Context: If the responses depend on specific contexts or conversational history, include that context within the `prompt`.

Here's a more complex example that includes context in the prompt:

```
{"prompt": "User: How do I reset my password? Assistant: You can reset your password by following these steps...", "completion": "First, go to the settings page. Then click on 'Reset Password' and follow the instructions sent to your email."}
```

This format is ideal for training language models to understand and respond to specific types of queries or dialogues, making it highly suitable for creating specialized ChatGPT assistants.

To make it easy to generate JSONL documents we built a program to convert a CSV file into a JSONL file. This made it easier for us to collect information from our team on a spreadsheet and export and transform into a JSONL.

Using Functions

To harness the power of function calling, follow these steps:

1. Define the Function: When calling the model, specify the functions you want to use along with the user's input. For example, if you want to know the current weather in a specific location, you can use a function like `get_current_weather`.
2. Model Interaction: The model, upon receiving the function and user's input, will process the information. If the function is recognized and the input matches its requirements, the model will return a structured response adhering to the function's signature.
3. Third-Party Integration: In some cases, like the weather example, you might need to integrate with a third-party API. Use the model's response to call this API and fetch the required data.
4. Send Data Back to Model: Once you have the data from the third-party API, you can send it back to the model for further processing or summarization. For a practical illustration, consider the scenario where a user asks, "What's the weather like in Boston right now?" Using function calling, the process would look like:

Call the model with the `get_current_weather` function and user's input. The model responds with a function call to `get_current_weather` for "Boston, MA". Use this response to call a third-party weather API. Send the weather data back to the model.

The model then summarizes the data, e.g., "The weather in Boston is currently sunny with a temperature of 22 degrees Celsius." Function calling, in essence, bridges the gap between natural language processing and structured data retrieval, making the OpenAI API a more powerful tool for developers.¹

¹ View code for above example here:

https://colab.research.google.com/drive/1yi5vbnOz2eT-FcxuivXb_NePqfvMyqO4#scrollTo=yQuyWoV1Hbva

Threads (Conversations)

Once your Assistant (Bot) is set up and running, you can start a conversation or 'Thread'. In Passive AI, this usually begins with the user asking a question or giving a command. In Active AI, on the other hand, your app needs to be set up to send the first message.

In our case, we start the Thread by setting the scene for the conversation and letting the bot know that the user has acknowledged the opening statement. We include the User ID right from the start to keep track of who's involved in the conversation. Our goal is to keep the whole conversation's context, even over several interactions. This way, the conversation stays relevant and connected throughout. By using the User ID, we can tailor responses to each user, keeping the conversation flowing smoothly no matter how long or complex it is. This method is especially good for keeping users engaged and making the Assistant more effective.

Building an AI-Powered Application: A New Approach

The structure of an AI-powered app is quite different from regular software, mainly because of three key elements. First, you need to train the AI to meet the goals of your product. Second, you have to figure out the best way for the software to interact with users.

Finally, the design should make it easy to keep an eye on the software and make changes based on how it performs and what users say. These unique features can be challenging for traditional software developers. It means moving away from a typical 'coding first' approach to a 'consult AI first' strategy, deciding what needs to be coded after considering the AI's role.

Other Use Cases to Consider

Dealership

A pivotal element of dealership software revolves around the sales and service aspects. The implementation of Artificial Intelligence can play a transformative role in this realm, particularly in identifying new sales prospects and engaging current customers in need of services. The integration of AI-powered sales and support assistants brings forth several key advantages. These include enhanced speed in lead engagement, round-the-clock availability, and the capability to effectively qualify interactions. Such functionalities are instrumental in optimizing the efficiency of the sales team, ensuring that their time is dedicated to the most promising opportunities.

Salon / Spa / Health Services

A.I. technology offers a multifaceted solution in the beauty and personal care industry, where resources can be limited. Beyond simply responding to leads and booking appointments, both during and after working hours, AI can transform how these businesses operate. For instance, AI can be utilized for personalized product recommendations based on customer preferences and purchase history, enabling a more tailored customer experience. It can also assist in managing inventory efficiently, predicting trends, and automating routine administrative tasks. Furthermore, AI can play a crucial role in engaging customers through personalized marketing campaigns and providing insights from customer feedback to improve services. In the beauty and health market, where customer experience is paramount, these AI applications are not just advantageous but essential for staying competitive and enhancing operational efficiency.

Food Service & Hospitality

In the dynamic world of food and hospitality, engaging with customers goes beyond their physical visit to your establishment. Maintaining an ongoing conversation with patrons after they have visited your restaurant is vital for ensuring customer satisfaction and loyalty. AI technology steps in to bridge the gap between customer experience and business response. By facilitating real-time feedback through AI-driven conversations, businesses can proactively address any issues before they escalate to negative reviews, enhancing the overall customer experience.

Moreover, AI's applications in this industry are diverse and transformative. For instance, AI can analyze customer preferences to personalize menu recommendations, improving customer satisfaction. It can also streamline kitchen operations by predicting busy periods and optimizing food preparation and inventory management, reducing waste and increasing efficiency. AI-driven chatbots can handle reservations and answer common queries, freeing up staff to focus on more complex tasks. In addition, AI can play a critical role in analyzing customer data to tailor marketing strategies, ultimately driving more business. By integrating AI into various facets of operation, from front-of-house interactions to back-end logistics, restaurants and hospitality businesses can revolutionize their service delivery, making them more responsive, efficient, and customer-centric.

E-Commerce / Retail

One of the pivotal challenges in e-commerce is effectively converting visitors into customers. This process is often hindered by the sheer volume of potential interactions and the need for real-time, quality engagement. The cost of human resources to manage these interactions can be prohibitive, making AI an essential tool. AI-driven chatbots and virtual assistants can handle thousands of simultaneous conversations, providing instant responses to customer queries. This not only supports conversion by answering key questions but also enhances the overall customer experience.

Beyond customer interaction, AI can revolutionize other aspects of e-commerce. Personalized shopping experiences can be created by using AI algorithms that analyze browsing habits and purchase history, recommending products tailored to individual customer preferences. AI can also optimize inventory management by predicting trends and demand, reducing overstock and stockouts. In logistics, AI can streamline shipping and delivery processes, offering customers real-time updates and reducing operational costs. Additionally, AI can assist in fraud detection, identifying and preventing suspicious transactions, thereby safeguarding both the business and its customers. AI's role in analyzing customer data can provide invaluable insights for targeted marketing campaigns and product development, further driving sales and business growth.

In summary, the integration of AI in e-commerce transcends mere customer interaction. It encompasses a comprehensive transformation of the shopping experience, inventory management, logistics, security, and marketing strategies, making businesses more efficient, responsive, and aligned with customer needs.

Insurance

Insurance companies often grapple with high volumes of leads and the pressing need for effective lead qualification to optimize insurance brokers' time. In an industry where most leads are generated online, the concept of speed-to-lead is paramount to ensure that a potential customer doesn't turn to a competitor. AI can play a pivotal role here by automating the lead qualification process, rapidly analyzing lead data to prioritize those with the highest potential, and ensuring quick follow-up.

Beyond lead management, AI can revolutionize several other aspects of the insurance sector. For instance, AI can enhance risk assessment by analyzing vast data sets, leading to more accurate policy pricing and underwriting decisions. In claims processing, AI can automate routine tasks, reduce processing time, and detect fraudulent claims, thereby increasing efficiency and reducing costs. Chatbots and virtual assistants can provide 24/7 customer service, handling inquiries and guiding customers through the policy selection process, improving customer experience and engagement.

Furthermore, AI can assist in predictive analytics, helping insurance companies to foresee trends, understand market dynamics, and develop new insurance products tailored to emerging needs. It can also play a role in personalized marketing, using customer data to tailor marketing messages and offers, thereby improving conversion rates. The integration of AI into these various facets of the insurance business not only streamlines operations but also provides a more personalized and efficient service to customers.

Home Services

The home services industry, encompassing areas like HVAC, plumbing, and electrical work, highly prioritizes speed-to-lead and constant availability due to the often urgent nature of service requests. AI can be instrumental in managing these demands by providing immediate conversational responses to leads, effectively qualifying them, and facilitating swift dispatch. This rapid engagement can be crucial in emergency situations where timely response is key.

Beyond lead management, AI can revolutionize various facets of home services. For example, AI can schedule and dispatch service professionals more efficiently, optimizing routes and service times based on urgency and location data. It can also assist in predictive maintenance, using data from smart home devices to alert homeowners of potential issues before they become emergencies, thereby offering a proactive service approach.

Additionally, AI-powered chatbots can assist customers in troubleshooting common problems, reducing the need for service calls for minor issues. This not only enhances customer satisfaction but also frees up professionals for more complex tasks. In inventory management, AI can predict the need for parts and equipment, ensuring that service vehicles are well-stocked and ready to address most issues on the first visit.

AI can also play a role in customer feedback analysis, helping businesses to improve their services based on real-time customer input. This feedback can be crucial for quality assurance and for tailoring services to meet customer needs more effectively.

Overall, the integration of AI in the home services industry goes beyond just responding to emergency requests. It encompasses scheduling, dispatch optimization, predictive maintenance, customer self-service, inventory management, and feedback analysis, fundamentally transforming how home services operate and enhancing both customer experience and operational efficiency.

The Impact on our Business

1. **Generating Revenue:** Incorporating artificial intelligence into our product suite has significantly enhanced its value proposition to our clientele, concurrently paving the way for novel revenue avenues. This technological integration has enabled us to introduce a groundbreaking, stand-alone product for lead conversion, exclusively available to our existing customer base. Prior to this advancement, the absence of a scalable and dependable solution had precluded us from offering such a service. We are projecting a 25% increase in our revenue streams this fiscal year, attributed to the sales of this AI-driven solution to both our existing and prospective clients. Furthermore, our strategic expansion plans include venturing into diverse market segments within the small and medium-sized business domain, extending beyond our current focus in the real estate sector. The long-term financial impact of these new revenue channels is yet to be fully quantified.

2. **Reducing Costs:** Implementing AI has reduced our annual costs for lead qualification and nurturing by 80% which equates to six figure savings.

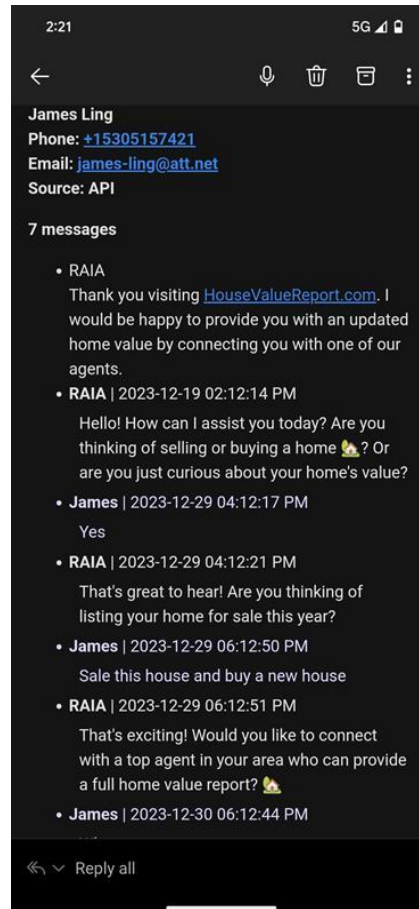
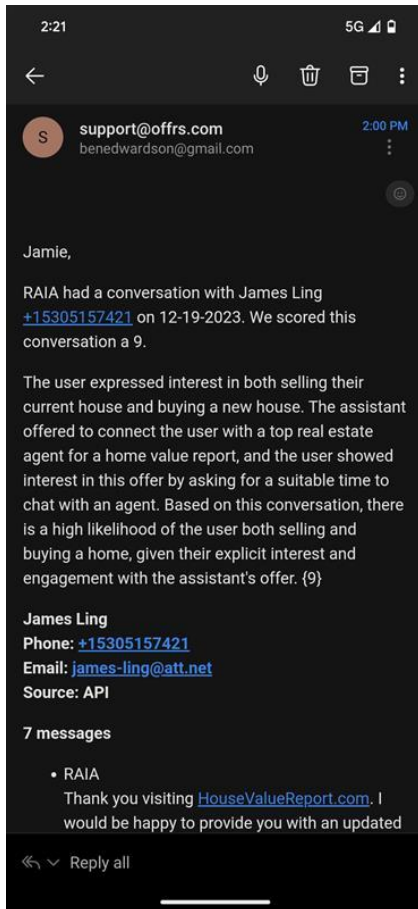
The Launch of R.A.I.A

We launched R.A.I.A (Responsive Artificially Intelligent Assistant) as a stand-alone product in late 2023. The product has now become a core feature for two businesses (offrs.com and ROOFre.com).

Our intention is to leverage this new product to diversify outside of the Real Estate sector.

As with many innovations, you develop a solution for a problem in a specific use-case, and realize that the problem is present in almost every business. The opportunity to bring a new product to new markets is now key to our strategy for organic growth in 2024.

Example Conversation Report RAIA sends to Client. The conversation is summarized, scored for likelihood to transact and provides a copy of the conversation.



Conclusion

Our experience with machine learning and generative AI has made our internal processes and products more efficient and improved what we offer in the market. But to successfully implement AI, careful planning is essential, taking into account the size and structure of your organization and how AI can be used specifically for your needs. Our small team (less than 30 full-time employees) and our own efficient systems have made it easier for us to introduce AI applications without much trouble. However, for companies with different setups, incorporating AI might require more resources and time.

Some of the challenges in developing AI applications are:

- Figuring out which business processes and tasks can be automated with AI and how to integrate these into your current workflows.
- Choosing a system architecture that works well with your existing applications and provides the AI functionalities you need.
- Changing the user experience to incorporate AI into your internal tools or products.
- Including ways to oversee and get human feedback on the AI applications.
- Finding the most suitable approach for your particular situation.

Even with these challenges, the benefits of integrating AI into business operations are significant, making it a key element to consider in any business strategy.

Appendix: Resources

Here are some valuable resources and documentation for developers interested in ChatGPT and its API:

1. **ChatGPT Model Documentation:** This provides detailed information about the ChatGPT model, which can be crucial for understanding its capabilities and limitations.

(<https://platform.openai.com/docs/models/chatgpt>).

2. **Chat API Reference:** This section of the documentation gives developers insights into how to create chats using the API, including necessary parameters and usage guidelines.

(<https://platform.openai.com/docs/api-reference/chat/create>).

3. **Introduction to OpenAI Platform:** For a broader understanding of the OpenAI platform, including its various offerings and how to get started, this introduction is quite helpful.

(<https://platform.openai.com/docs/introduction>).

4. **OpenAI Developer Forum.** <https://community.openai.com/>

These resources should provide a comprehensive starting point for developing applications leveraging ChatGPT and its API.

CODE REFERENCES

Assistants API Code Example:

https://cookbook.openai.com/examples/assistants_api_overview_python

Sample code on Google Colab.

https://colab.research.google.com/drive/1Mkn6cz3wA49qkAz-oFTLTJq4pT_Syw1X?usp=sharing