# JPMC #2: Self-Hosted End-to-End Question Answering Bot

## AI Studio Final Presentation

Break Through Tech New York @Cornell Tech
[12.13.2023]

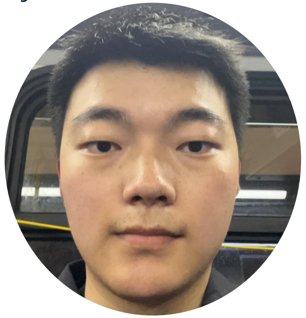
## Introductions

## Meet Our Team!

**Alyssa Choi**
Stony Brook University

**Lauretta Martin**
Rutgers University

**Sonia Tan**
Stony Brook University



**Hanqi Lin**
CUNY Queens College

**Fayha Farooqi**
Stevens Institute of Technology

# Our AI Studio TA and Challenge Advisors

**Sakshi Pandey**
AI Studio TA

**Adedapo Alabi**
Challenge Advisor

## Presentation Agenda

1. **AI Studio Project Overview**
2. **Architecture**
3. **Implementation**
4. **Vector Search LLM**
5. **Final Implementation and Demonstration**

## AI Studio Project Overview

“

Self-Hosted Product-Centric Project

Utilizing a pre-trained LLM of public data

End-to-end implementation

## Our Goal

1. Design an **end-to-end self-hosted application** that allows users to retrieve information from a private pdf document
2. Redesign to implement **multiple documents** to be inputted into the chatbot
3. Allow JPMC to deliver recommendations and information to clients in-house without third-parties involved
    a. Companies with financial data don’t have the liberty to send data to OpenAI due to the necessity of **data security** regarding financial statements
    b. Federal Trade Commission - Financial Privacy Regulations

## Business Impact

- Worked with JP Morgan Chase on this project
- With explosions of LLMs such as ChatGPT, chatbots are now able to have human-like conversations
- Companies like JPMC would benefit from chatbots that allow users to talk to these documents without sharing any data with external services

## Our Approach

Decided on the following approach based on Challenge Advisor guidance. Before starting the project, we researched different architectures and frameworks that are feasible for us to implement.

Sept

**Finding Database**

Using a pretrained LLM but finding options for chatbot creation (ex. Streamlit, React, Flask etc.)

Late Sept - Early Oct

**Integrating to Streamlit**

Integrating Huggingface Document QnA Model in Streamlit

Mid Oct

**Researching Vector Search LLM**

Finding best option to create a vector database from pdf documents to run queries of nearest matches

Nov

**Implementation**

User can upload multiple documents and system figures out best document that fits the question and outputs response

## How can financial companies abide by data regulations while still utilizing AI/ML/DL to make processes more efficient?

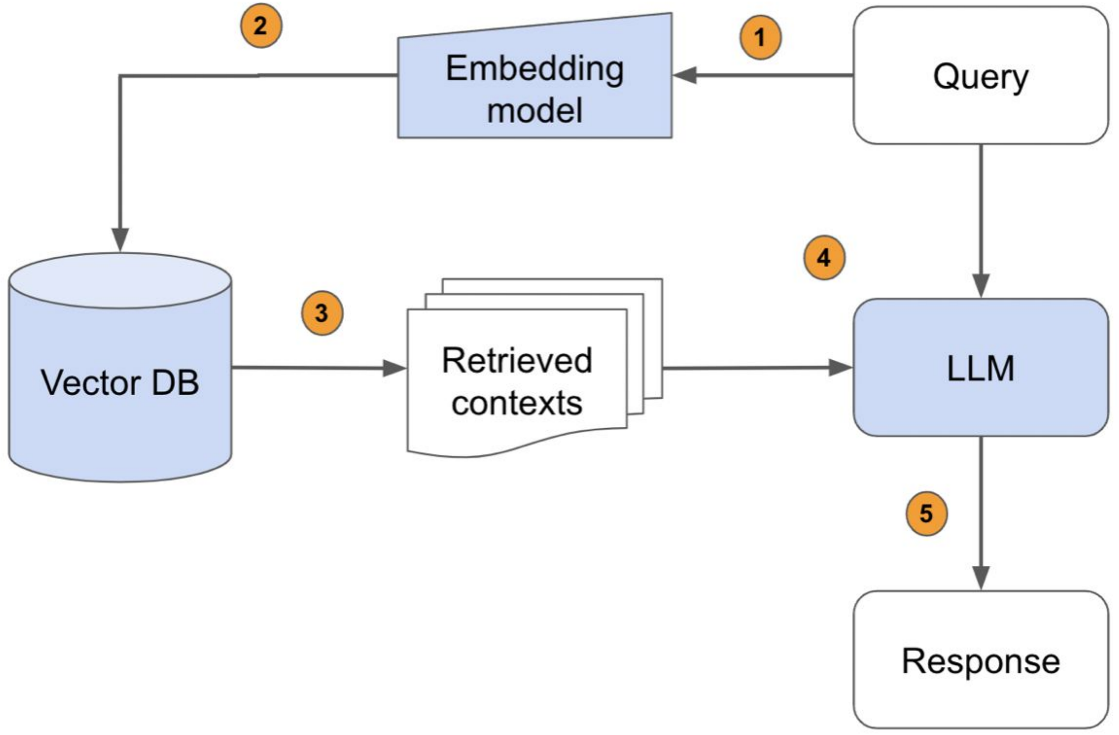
- Create data processing programs in-house!
  - Maintains **data integrity** and safety
  - Abides by financial institution data privacy laws
  - Gives firms the **opportunity to cater their created software product** to their own needs
  - **Streamlined** data collection and processing

## Resources We Leveraged

- HuggingFace
- Streamlit
- Github

## Architecture

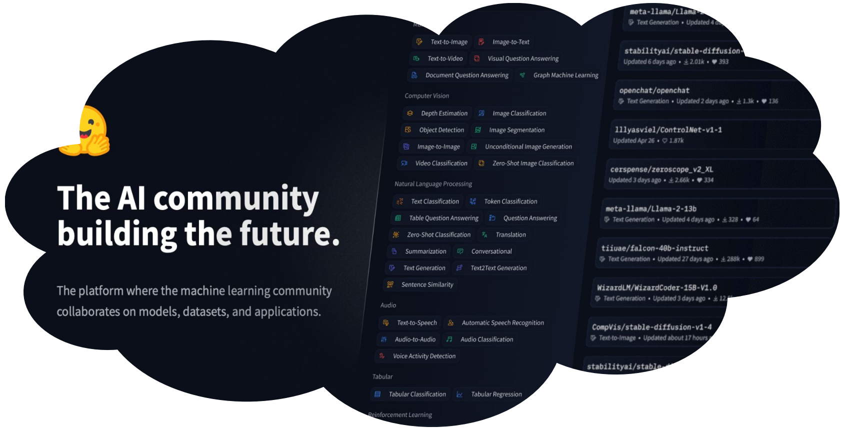
## Program Architecture



## Implementation

# Model Research

- **HuggingFace**
  - Building and deploying machine learning models
  - Includes open-source pre-trained models
  - Ideal to test chatbot on in regards to basic information
  - Involves a wide range of datasets to test different types of questions on



**The AI community building the future.**

The platform where the machine learning community collaborates on models, datasets, and applications.

Text-to-Image Image-to-Text
Text-to-Video Visual Question Answering
Document Question Answering Graph Machine Learning

Computer Vision
Depth Estimation Image Classification
Object Detection Image Segmentation
Image-to-Image Unconditional Image Generation
Video Classification Zero-Shot Image Classification

Natural Language Processing
Text Classification Token Classification
Table Question Answering Question Answering
Zero-Shot Classification Translation
Summarization Conversational
Text Generation Text2Text Generation
Sentence Similarity

Audio
Text-to-Speech Automatic Speech Recognition
Audio-to-Audio Audio Classification
Voice Activity Detection

Tabular
Tabular Classification Tabular Regression

Reinforcement Learning

meta-llama/Llama-
Text Generation • Updated 4 d

stabilityai/stable-diffusion-
Updated 6 days ago • 2.01k • 393

openchat/openchat
Text Generation • Updated 2 days ago • 1.3k • 136

lllyasviel/ControlNet-v1-1
Updated Apr 26 • 1.87k

cerspense/zeroscope_v2_XL
Updated 3 days ago • 2.66k • 334

meta-llama/Llama-2-13b
Text Generation • Updated 4 days ago • 328 • 64

tiiuae/falcon-40b-instruct
Text Generation • Updated 27 days ago • 288k • 899

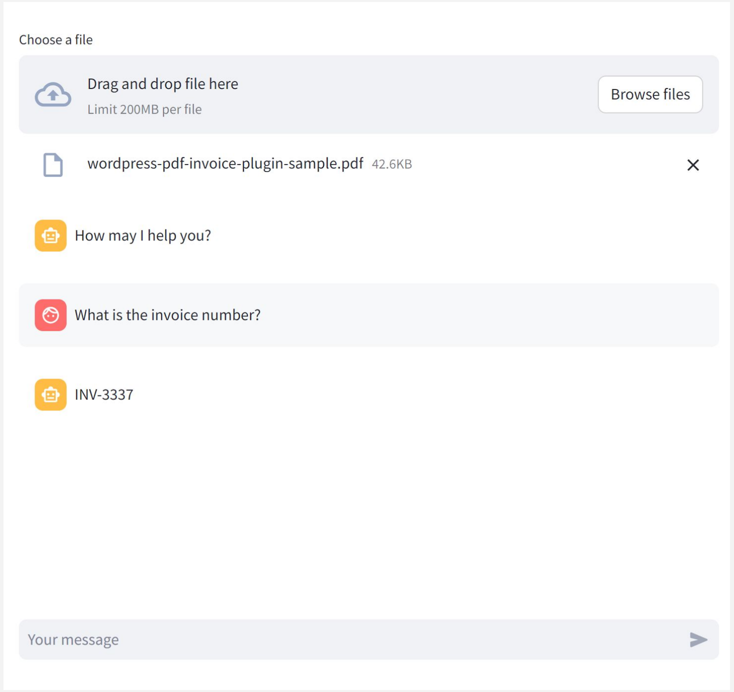
WizardLM/WizardCoder-15B-V1.0
Text Generation • Updated 3 days ago • 12.

CompVis/stable-diffusion-v1-4
Text-to-Image • Updated about 17 hours

stabilityai/stable-

**Streamlit** is an open-source app framework for machine learning and data science web apps

Streamlit


Choose a file

Drag and drop file here

Limit 200MB per file

wordpress-pdf-invoice-plugin-sample.pdf 42.6KB

How may I help you?

What is the invoice number?

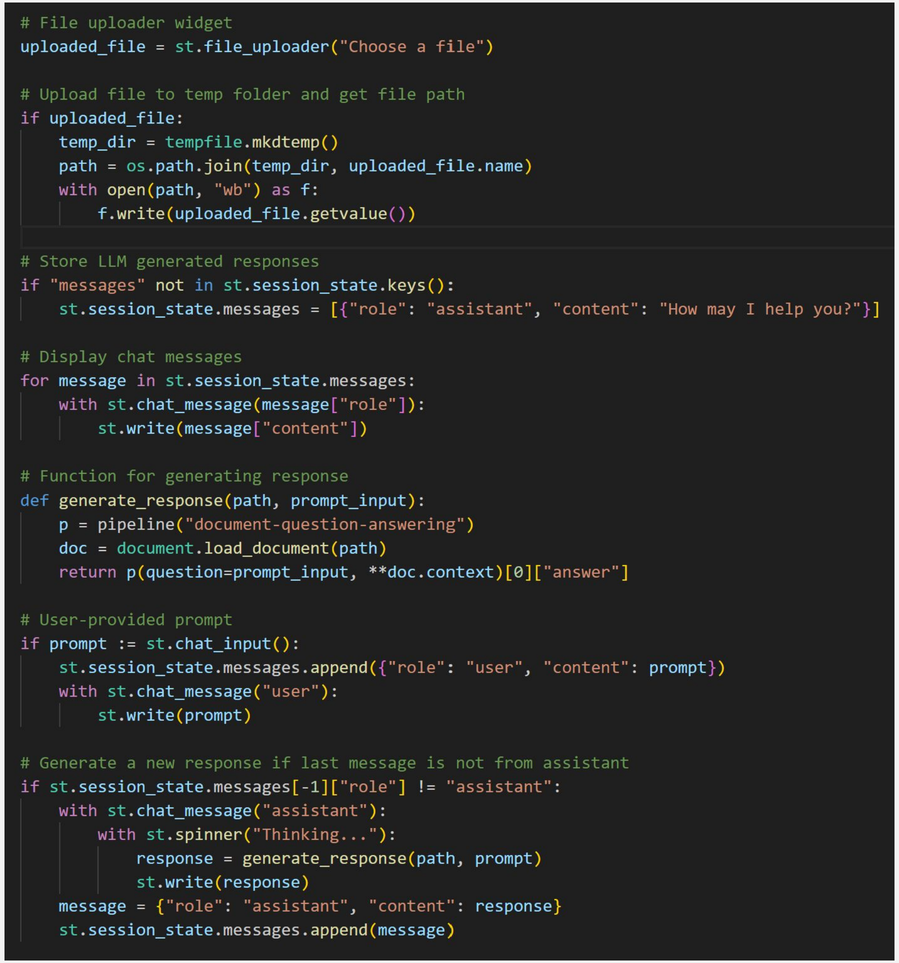
INV-3337

# DocQuery Implementation

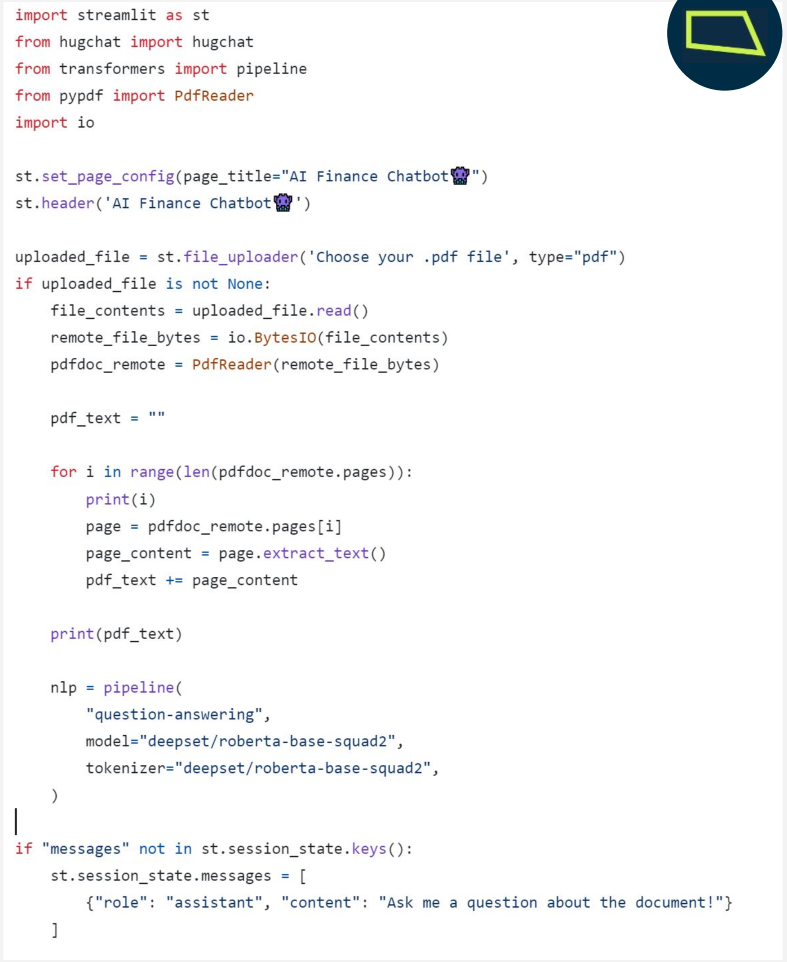
- Takes a single image file or single page PDF
- Takes a question from the user
- Returns an answer using DocQuery

## RoBERTa Implementation

- **RoBERTa Overview**
  - Self-supervised transformers model pretrained on large English texts
  - Bidirectional representation learning
- **Implementation**
  - Takes a single PDF file and extracts the text from the file
  - Takes a question from the user
  - Returns an answer using the RoBERTa model
  - Demo



```python
# File uploader widget
uploaded_file = st.file_uploader("Choose a file")

# Upload file to temp folder and get file path
if uploaded_file:
    temp_dir = tempfile.mkdtemp()
    path = os.path.join(temp_dir, uploaded_file.name)
    with open(path, "wb") as f:
        f.write(uploaded_file.getvalue())

# Store LLM generated responses
if "messages" not in st.session_state.keys():
    st.session_state.messages = [{"role": "assistant", "content": "How may I help you?"}]

# Display chat messages
for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.write(message["content"])

# Function for generating response
def generate_response(path, prompt_input):
    p = pipeline("document-question-answering")
    doc = document.load_document(path)
    return p(question=prompt_input, **doc.context)[0]["answer"]

# User-provided prompt
if prompt := st.chat_input():
    st.session_state.messages.append({"role": "user", "content": prompt})
    with st.chat_message("user"):
        st.write(prompt)

# Generate a new response if last message is not from assistant
if st.session_state.messages[-1]["role"] != "assistant":
    with st.chat_message("assistant"):
        with st.spinner("Thinking..."):
            response = generate_response(path, prompt)
            st.write(response)
    message = {"role": "assistant", "content": response}
    st.session_state.messages.append(message)
```



```python
import streamlit as st
from hugchat import hugchat
from transformers import pipeline
from pypdf import PdfReader
import io

st.set_page_config(page_title="AI Finance Chatbot👾")
st.header('AI Finance Chatbot👾')

uploaded_file = st.file_uploader('Choose your .pdf file', type="pdf")
if uploaded_file is not None:
    file_contents = uploaded_file.read()
    remote_file_bytes = io.BytesIO(file_contents)
    pdfdoc_remote = PdfReader(remote_file_bytes)

    pdf_text = ""

    for i in range(len(pdfdoc_remote.pages)):
        print(i)
        page = pdfdoc_remote.pages[i]
        page_content = page.extract_text()
        pdf_text += page_content

    print(pdf_text)

    nlp = pipeline(
        "question-answering",
        model="deepset/roberta-base-squad2",
        tokenizer="deepset/roberta-base-squad2",
    )

if "messages" not in st.session_state.keys():
    st.session_state.messages = [
        {"role": "assistant", "content": "Ask me a question about the document!"}
    ]
```

## Vector Search LLMs

## Vector Search in Large Language Models (LLMs): A Comprehensive Overview

- **What is Vector Search?**
  - Turns text into numerical vectors representing its meaning.
  - Helps find relevant answers in a huge amount of data.
- **How Does it Work?**
  - Input Conversion: Changes your question into a vector.
  - Matching: Compares your vector with stored vectors to find the best match.
- **Benefits**
  - Fast: Quickly sifts through large data.
  - Accurate: Understands context, not just keywords.

## Implementing Vector Search in Chatbots

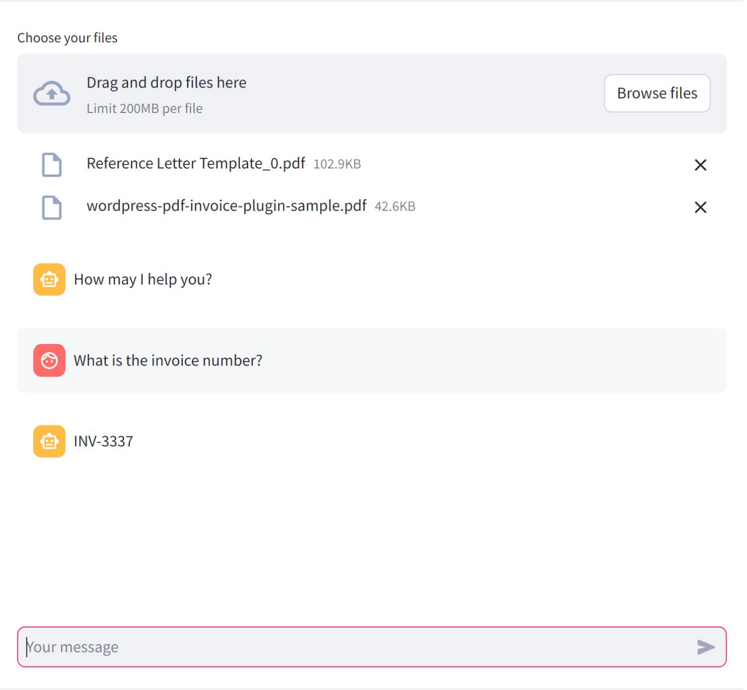
**Input as Query:** AI interprets request as a query. Question is now converted into a numerical vector.

**Semantic Understanding:** LLM analyzes the text's content, breaking down into simpler concepts.

**Response Generation:** The AI then generates a response, akin to how vector search matches a query vector with the most relevant document vectors, the essence of the original text is preserved.

student input: “ to make my HW more “readable and to the point”
gpt respond: In order to make your HW more “concise and reader-friendly”......

## Final Implementation and Demonstration



Choose your files

Drag and drop files here
Limit 200MB per file

Reference Letter Template_0.pdf 102.9KB

wordpress-pdf-invoice-plugin-sample.pdf 42.6KB

How may I help you?

What is the invoice number?

INV-3337

# Txt-AI Implementation

- Converts pdf files into txt
- Creates vector embeddings of all files and stores in a vector database
- Searches database for file with closest context

## References and Resources

- impira/layoutlm-document-qa · Hugging Face
- GitHub - impira/docquery: An easy way to extract information from documents
- deepset/roberta-base-squad2 · Hugging Face
- GitHub - neuml/txtai: 💡 All-in-one open-source embeddings database for semantic search, LLM orchestration and language model workflows
- How to build an LLM-powered ChatBot with Streamlit
- GitHub - py-pdf/pypdf: A pure-python PDF library capable of splitting, merging, cropping, and transforming the pages of PDF files

# Thank You!

Please advise of any questions or comments, we'd love to hear your feedback!