

Meet Divya

A Reimagined Virtual USCIS Assistant



Bassim Eledath // Stefan Lane-Nedialkov // Daniel Lazaro // Manasvi Tickoo



Our Goals

US immigration is complex...

Information is plentiful – but
relevant information is hard
to locate...

USCIS is the main US
immigration department...

...whose current chatbot,
Emma, is somewhat basic
and not very helpful.



- **Build a better chatbot: Divya**
 - **Scrape the USCIS website for information**
 - **Create embeddings**
 - **Implement Retrieval-Augmented Generation**
- **Deploy the app via Streamlit on CommunityCloud**



Emma vs Divya

aka USCIS vs AIEdge

Need Help? **Ask Emma >**



Need Help? **Ask DIVYA >**



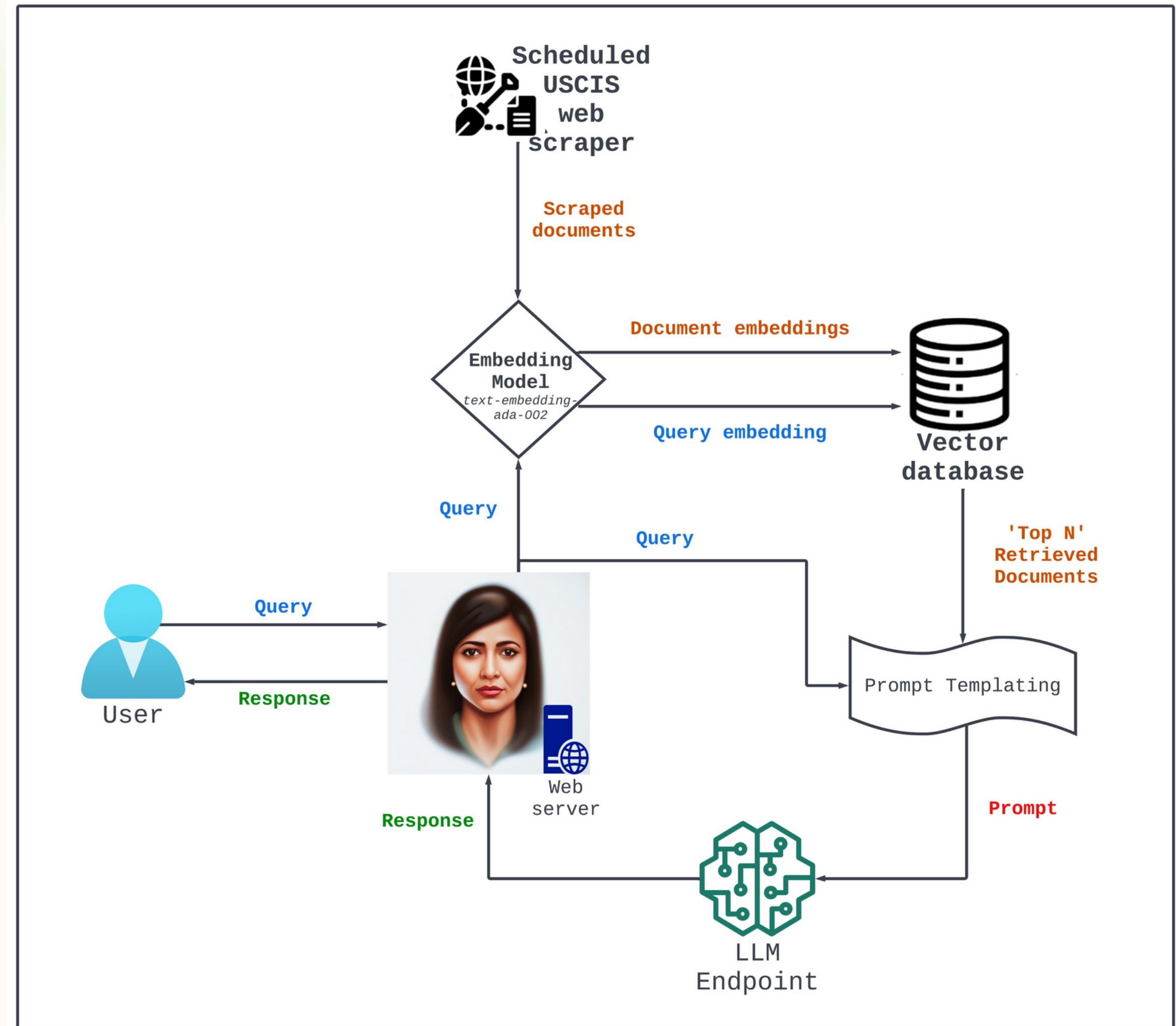
- Formulaic chat, with users picking from a menu of choices
- Relevant information is very high level and light on specifics for individual cases
- The chatbot is often unavailable during 'business hours' (reasons unclear)
- Much more of a 'dialogue' experience
- Relevant information is quickly retrieved; the generated answer is more human-sounding
- Available 24/7



Architecture

Retrieval-Augmented Generation

- Scheduled USCIS web scraper scrapes pages, tables and any relevant information
- Embeddings for scraped documents are generated
- User sends a query and the query embedding is generated
- The 'top N' documents are retrieved based on cosine similarity between query and document embeddings
- The query and the top N documents are passed in a prompt template to an LLM endpoint (ChatGPT)
- The ChatGPT response is returned to the user



Deployment

Streamlit / Community Cloud

- We implemented our solution as a Streamlit app
- We deployed on Community Cloud, one of the deployment options offered by Streamlit

Ask Divya

Disclaimer

For official immigration advice please consult a certified lawyer

 Hello there, what immigration related question can I help you with today?

 What is OPT?

 OPT stands for Optional Practical Training. This refers to a program that allows international students with F-1 visas to work in the United States for a certain period of time after they have completed their academic studies. The given context explains that students who have earned a STEM degree and meet certain criteria may apply for a 24-month extension of their post-completion OPT employment authorization.

 What are the options for immigration if I would like to sponsor my spouse who is a foreign national?

 If you would like to sponsor your spouse who is a foreign national, the options for immigration depend on the specific circumstances. In the given context, it is mentioned that if your I-140 petition is approved, your spouse and unmarried children under the age of 21 may be eligible to apply for admission to the United States in certain categories. These categories include E34 and EW4 (spouse of a "skilled worker" or "professional") and E35 and EW5 (child of a "skilled worker" or "professional"). Additionally, the context mentions the possibility of sponsorship for certain special immigrant categories, such as religious workers, special immigrant juveniles, certain broadcasters, and more. It is advised to review the category-specific information and form instructions provided by USCIS to determine the eligibility and requirements for sponsoring your spouse.



Divya Is Better than Emma

Select Metrics

- Dataset - Curated Immigration FAQ dataset from leading law firms
- Evaluation - Use Sentence Embeddings Cosine similarity to human 'expert responses'
- Compare responses from Emma, Chat GPT and Divya

```
question = "Can I work in the U.S. while waiting for my green card?"  
reference = "Anyone who already has a valid work visa (for example, an H-1B or L-1 visa) can usually continue working in the United States while waiting for your green card."  
  
emma = "The best way to track the delivery of your document is to check Case Status Online (link below) to get the status of your USCIS highly recommends that you register for Informed Delivery through USPS. This allows you to track all mail and packages being delivered to you. For more information on how to register for this service, see the link below."  
  
chatgpt = "As of my last update in September 2021, the ability to work in the U.S. while waiting for your Green Card depends on your individual circumstances. Here are a few common scenarios: Adjustment of Status (AOS) Applicants: If you are in the U.S. and have filed an Adjustment of Status application, you may be able to work legally in the U.S. while your application is being processed. If you are pending if you have a valid Employment Authorization Document (EAD). To obtain an EAD, you need to file Form I-765, Application for Employment Authorization, with USCIS. Once approved, the EAD will allow you to work legally in the U.S. while your Green Card application is being processed."  
  
divya = "Yes, you can work in the U.S. while waiting for your green card under certain circumstances. If you have applied for an Employment Authorization Document (EAD) by filing Form I-765, Application for Employment Authorization, you can work in the U.S. while your application is being processed. The EAD provides temporary work authorization. However, please note that the specific eligibility and requirements for employment authorization depend on your individual circumstances and the type of visa you hold. It is important to consult with an immigration attorney or refer to the official USCIS website for the most up-to-date information."  
  
print('SIM score emma -> {}'.format(util.pytorch_cos_sim(model.encode(reference, convert_to_tensor=True), model.encode(emma, convert_to_tensor=True))))  
print('SIM score chatgpt -> {}'.format(util.pytorch_cos_sim(model.encode(reference, convert_to_tensor=True), model.encode(chatgpt, convert_to_tensor=True))))  
print('SIM score divya -> {}'.format(util.pytorch_cos_sim(model.encode(reference, convert_to_tensor=True), model.encode(divya, convert_to_tensor=True))))  
  
SIM score emma -> tensor([[0.1240]])  
SIM score chatgpt -> tensor([[0.7861]])  
SIM score divya -> tensor([[0.8260]])
```



Augmenting base knowledge with domain specific information not limited to the base model training period.

Q. Has EB5 program ended ?

Chat GPT - As of my last update in September 2021, the EB-5 program had not ended. However, please note that immigration policies and programs are subject to change, and it's essential to verify the current status of the EB-5 program as of the current date in 2023. The EB-5 Immigrant Investor Program is a U.S. employment-based immigration program that provides a pathway to lawful permanent residency (Green Card) for foreign investors who invest a certain amount of capital in a new commercial enterprise that creates jobs for U.S. workers. To get the most accurate and up-to-date information about the EB-5 program's status, its requirements, and any potential changes or updates, I recommend checking the official website of U.S. Citizenship and Immigration Services (USCIS) or consulting with an immigration attorney who specializes in EB-5 investments and visas. They will be able to provide the latest details on the program's current status and eligibility criteria.

Divya - According to the provided context, the EB-5 Immigrant Investor Regional Center Program is in effect through **Sept. 30, 2027**. Therefore, the program has not ended. However, it is important to note that the context does not provide further details or updates on the current status or any potential changes to the program. For the most accurate and up-to-date information, it is advisable to refer to the official USCIS website or consult with an immigration attorney.



Further Development

- Tuning the Token Length
- Conversational buffer memory
- Once documents repository increases in size and complexity, transition to a higher-dimension/capacity vector database
- To improve privacy, implement LLM endpoint on own server
- Front-end website development with JavaScript
- Make Divya bilingual (Spanish) or multilingual



Q&A



Deployment Code

```
1  import openai
2  import streamlit as st
3  import pandas as pd
4  from scipy import spatial
5  import tiktoken
6  import ast
7
8  st.title("Ask Divya")
9  with st.expander("i Disclaimer"):
10    st.caption(
11      "For official immigration advice please consult a certified lawyer"
12    )
13
14  ### Initial message ###
15  message = st.chat_message("assistant", avatar="./ask_divya.png")
16  message.write("Hello there, what immigration related question can I help you with today?")
17  #####
18
19  ### Embedding ###
20
21  df = pd.read_csv("embeddings.csv")
22  # convert embeddings from CSV str type back to list type
23  df['embedding'] = df['embedding'].apply(ast.literal_eval)
24  df['title_embedding'] = df['title_embedding'].apply(ast.literal_eval)
25
26
27  # Embedding model
28  EMBEDDING_MODEL = "text-embedding-ada-002"
29
30  # search function
31  def strings_ranked_by_relatedness(
32    query: str,
33    df: pd.DataFrame,
34    relatedness_fn=lambda x, y: 1 - spatial.distance.cosine(x, y),
35    top_n_titles: int = 5,
36    top_n: int = 100
37  ) -> tuple[list[str], list[float]]:
38    """Returns a list of strings and relatednesses, sorted from most related to least."""
39    query_embedding_response = openai.Embedding.create(
40      model=EMBEDDING_MODEL,
41      input=query,
42    )
43    query_embedding = query_embedding_response["data"][0]["embedding"]
44
45    # First, sort titles by relatedness
46    title_relatednesses = [
47      (row["title"], relatedness_fn(query_embedding, row["title_embedding"]))
48      for i, row in df.iterrows()
49    ]
50    title_relatednesses.sort(key=lambda x: x[1], reverse=True)
51    top_titles, _ = zip(*title_relatednesses[:top_n_titles])
52    print(top_titles)
53
54    # Filter dataframe to include only top titles
55    df_filtered = df[df['title'].isin(top_titles)]
56
57    # Then within that dataframe, sort texts by relatedness
58    strings_and_relatednesses = [
59      (row["text"], relatedness_fn(query_embedding, row["embedding"]))
60      for i, row in df_filtered.iterrows()
61    ]
62    strings_and_relatednesses.sort(key=lambda x: x[1], reverse=True)
63    strings, relatednesses = zip(*strings_and_relatednesses)
64
65    return strings[:top_n], relatednesses[:top_n]
66
67  def prompt_template(prompt):
68    prompt_extra = "Address or answer the question by retrieving the information from the following table:  
"
69    strings, relatednesses = strings_ranked_by_relatedness(prompt, df, top_n=15)
70    encoding = tiktoken.encoding_for_model("gpt-3.5-turbo")
71    token_limit = 1200 # set token limit here
72    total_tokens = 0
73    for string in strings:
74      string_tokens = len(encoding.encode(string)) # get number of tokens in string
75      if total_tokens + string_tokens <= token_limit:
76        total_tokens += string_tokens
77        prompt_extra += "\n" + string
78      else:
79        break
80    return prompt + " " + prompt_extra
81
82
83  ##### API CALL #####
84  openai.api_key = st.secrets["OPENAI_API_KEY"]
85
86  if "openai_model" not in st.session_state:
87    st.session_state["openai_model"] = "gpt-3.5-turbo"
88
89  if "messages" not in st.session_state:
90    st.session_state.messages = []
91
92  for message in st.session_state.messages:
93    if message["role"] == "assistant":
94      with st.chat_message(message["role"], avatar="./ask_divya.png"):
95        st.markdown(message["content"])
96    else:
97      with st.chat_message(message["role"], avatar="🔴"):
98        st.markdown(message["content"])
99
100 if prompt := st.chat_input("Ask a question"):
101   st.session_state.messages.append({"role": "user", "content": prompt})
102   with st.chat_message("user", avatar="🔴"):
103     st.markdown(prompt)
104
105   with st.chat_message("assistant", avatar="./ask_divya.png"):
106     message_placeholder = st.empty()
107     full_response = ""
108
109     for response in openai.ChatCompletion.create(
110       model=st.session_state["openai_model"],
111       messages=[{"role": "user", "content": prompt}],
112       stream=True,
113     ):
114       for m in st.session_state.messages:
115         if m["role"] == "assistant":
116           full_response += response.choices[0].delta.get("content", "") + "\n"
117
118     message_placeholder.markdown(full_response)
119     message_placeholder.markdown(full_response)
120
121   st.session_state.messages.append({"role": "assistant", "content": full_response})
```

