11/14 Client Meeting

Nexteer (AI Bots) Capstone Team

Agenda

Progress in the Past Week

Goals for Next Week

Deliverable

Questions

Progress in the Past Week

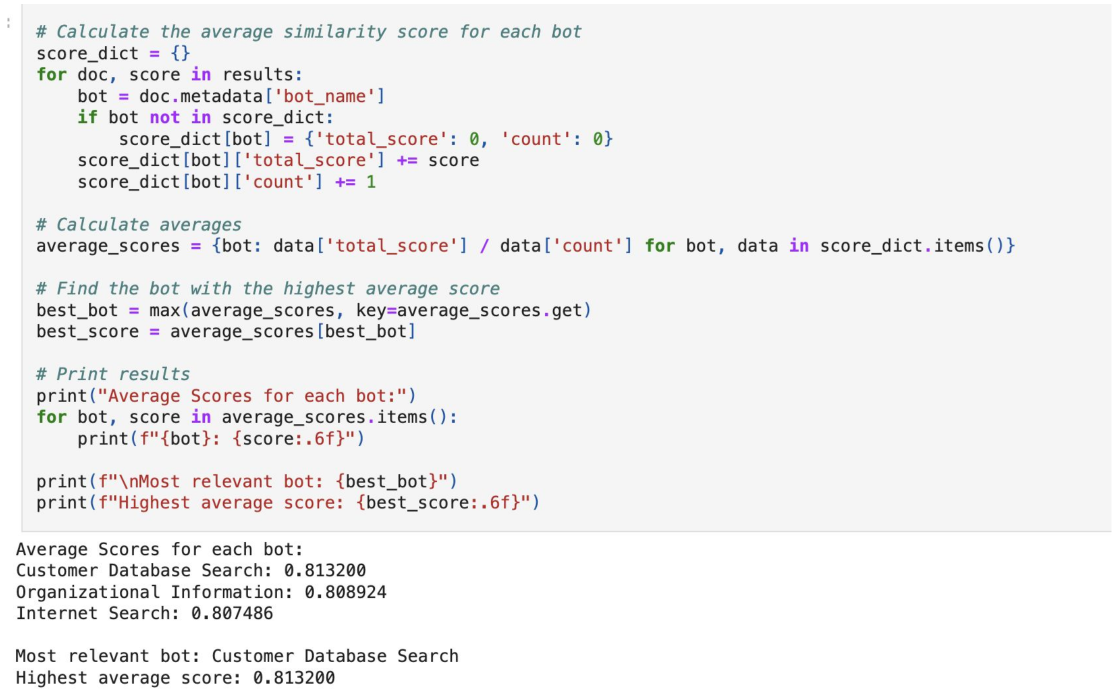
1. Merging all code and building an usable service out of it
2. Built web based dashboard for uploading documents and generating metadata
3. Performed similarity search experiments with dummy data to evaluate initial model behavior.

Similarity Search with Json Metadata

```python
# Perform similarity search with scores
query = "Can you provide a list of customers who've been featured in our marketing materials?"
results = docsearch.similarity_search_with_score(query, k=5)

# Print results
print(f"Query: {query}\n")
print("Results:")
for doc, score in results:
    print(f"Bot: {doc.metadata['bot_name']}")
    print(f"Similarity Score: {score}")
    print(f"Content: {doc.page_content[:50]}...")  # Truncate content for readability
    print()
```

```
Query: Can you provide a list of customers who've been featured in our marketing materials?

Results:
Bot: Customer Database Search
Similarity Score: 0.813858
Content: total, decisions, channels, average, metric, measu...

Bot: Customer Database Search
Similarity Score: 0.812541604
Content: total decisions channels average metric measuring ...

Bot: Organizational Information
Similarity Score: 0.808923662
Content: hires, requesting, schedule, positive, providing, ...

Bot: Internet Search
Similarity Score: 0.808375895
Content: growing, patients, analyze, consumers, adopt, comm...

Bot: Internet Search
Similarity Score: 0.80659622
Content: growing patients analyze consumers adopt commerce ...
```

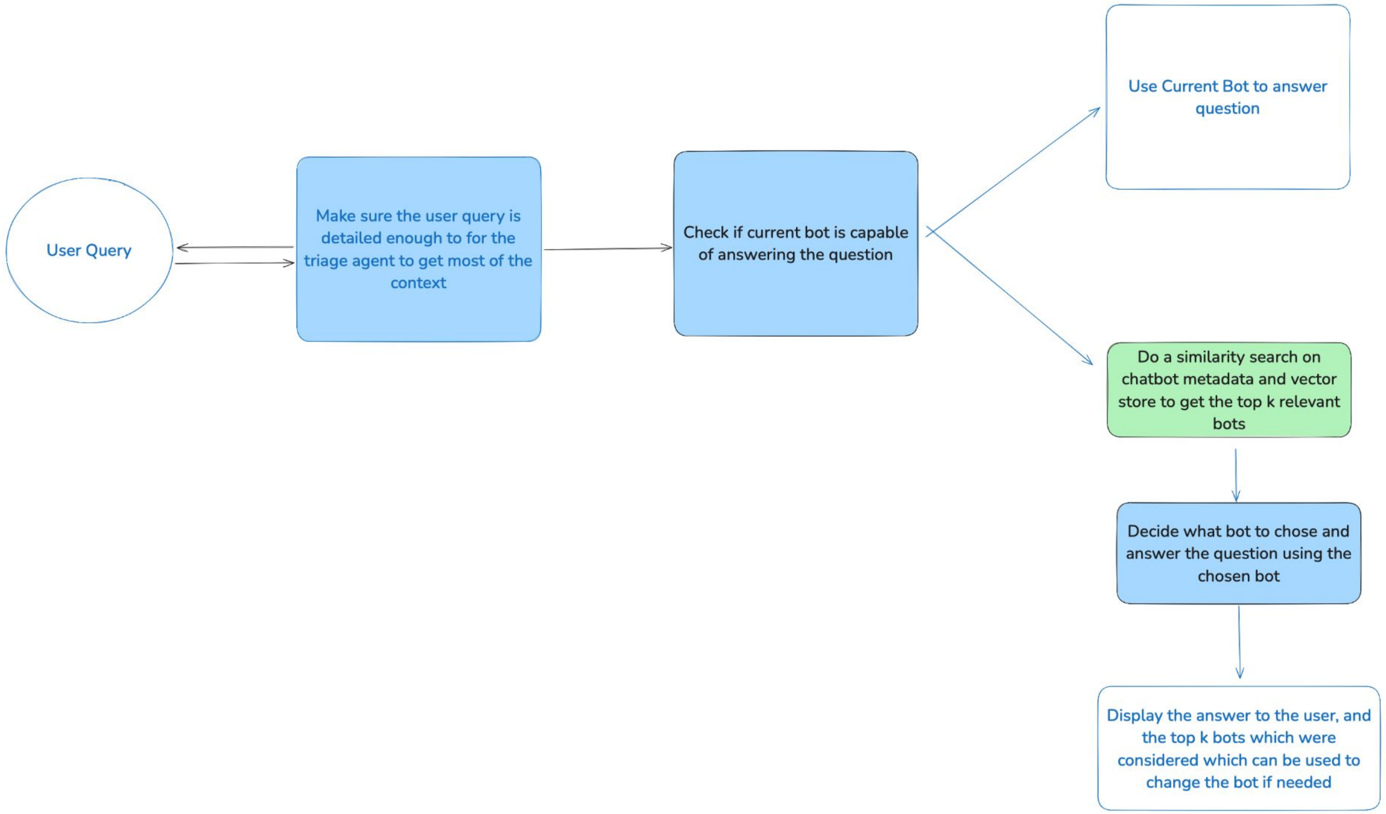
Similarity Search with Json Metadata



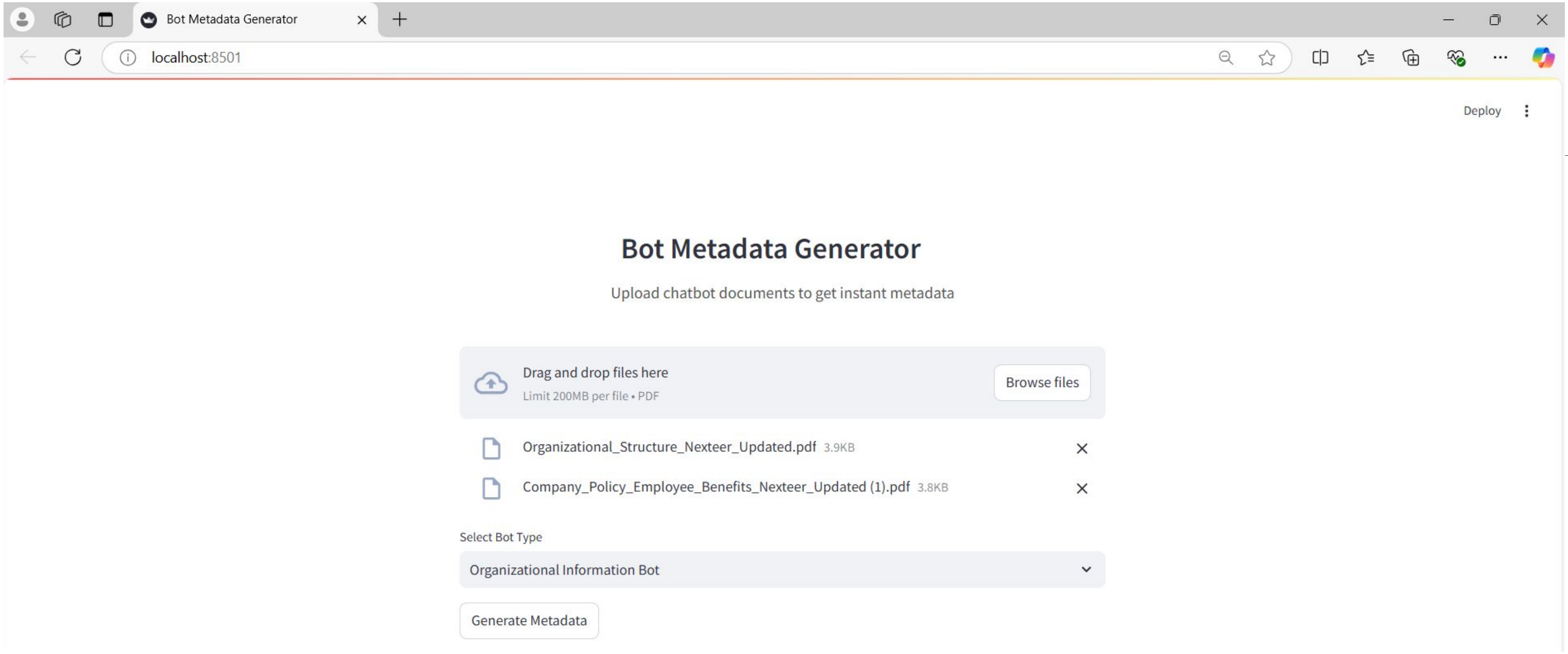
```python
# Calculate the average similarity score for each bot
score_dict = {}
for doc, score in results:
    bot = doc.metadata['bot_name']
    if bot not in score_dict:
        score_dict[bot] = {'total_score': 0, 'count': 0}
    score_dict[bot]['total_score'] += score
    score_dict[bot]['count'] += 1

# Calculate averages
average_scores = {bot: data['total_score'] / data['count'] for bot, data in score_dict.items()}

# Find the bot with the highest average score
best_bot = max(average_scores, key=average_scores.get)
best_score = average_scores[best_bot]

# Print results
print("Average Scores for each bot:")
for bot, score in average_scores.items():
    print(f"{bot}: {score:.6f}")

print(f"\nMost relevant bot: {best_bot}")
print(f"Highest average score: {best_score:.6f}")
```

```
Average Scores for each bot:
Customer Database Search: 0.813200
Organizational Information: 0.808924
Internet Search: 0.807486

Most relevant bot: Customer Database Search
Highest average score: 0.813200
```

Triage Agent - WIP


User Query
Make sure the user query is detailed enough to for the triage agent to get most of the context
Check if current bot is capable of answering the question
Use Current Bot to answer question
Do a similarity search on chatbot metadata and vector store to get the top k relevant bots
Decide what bot to chose and answer the question using the chosen bot
Display the answer to the user, and the top k bots which were considered which can be used to change the bot if needed

Web Dashboard for Auto MetaData Generation



Bot Metadata Generator

Upload chatbot documents to get instant metadata

Drag and drop files here
Limit 200MB per file • PDF

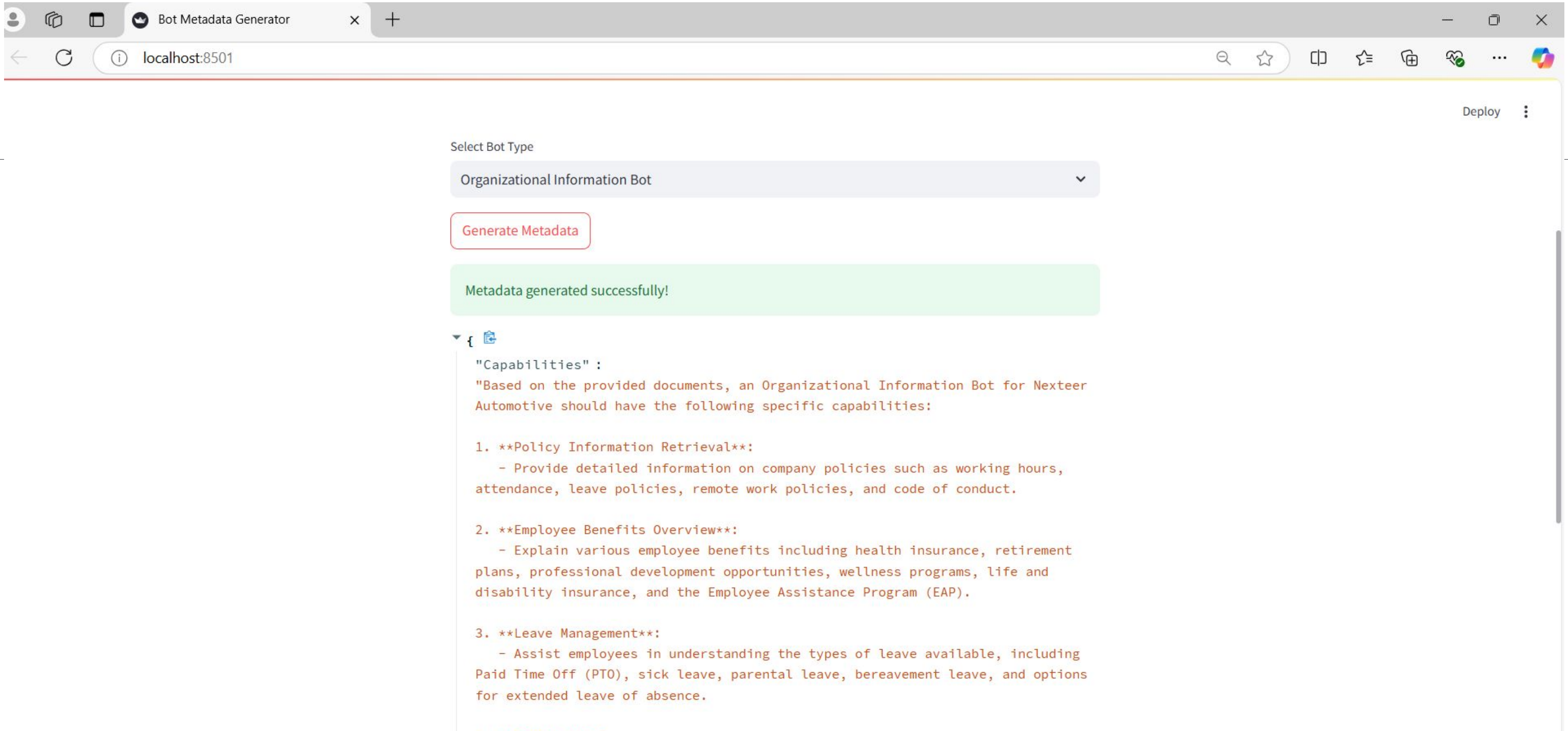
Organizational_Structure_Nexteer_Updated.pdf 3.9KB

Company_Policy_Employee_Benefits_Nexteer_Updated (1).pdf 3.8KB

Select Bot Type

Organizational Information Bot

Web Dashboard for Auto MetaData Generation



Select Bot Type

Organizational Information Bot

Generate Metadata

Metadata generated successfully!

```
{
"Capabilities" :
"Based on the provided documents, an Organizational Information Bot for Nexteer
Automotive should have the following specific capabilities:

1. **Policy Information Retrieval**:
   - Provide detailed information on company policies such as working hours,
attendance, leave policies, remote work policies, and code of conduct.

2. **Employee Benefits Overview**:
   - Explain various employee benefits including health insurance, retirement
plans, professional development opportunities, wellness programs, life and
disability insurance, and the Employee Assistance Program (EAP).

3. **Leave Management**:
   - Assist employees in understanding the types of leave available, including
Paid Time Off (PTO), sick leave, parental leave, bereavement leave, and options
for extended leave of absence.
```

Scoring System

- Detailed agent metadata for informed decision-making
- Performance tracking for continuous improvement
- Flexible scoring system for optimal agent selection
- Consideration of both immediate context and historical performance

Goals for Next Week

1. Documentation for usage and deployment of web based tool for metadata generation
2. Refine and optimize similarity search mechanisms, incorporating metadata to improve relevance and accuracy of results.
3. Test the service over the various synthetic data, improve latency and correctness of each individual component
4. Generate more complicated data to test it further

Deliverable

POC:

1. A model that understands user query domain and skills
2. Redirection logic/demo of redirecting user to specialized chatbots

Deliverable items:

1. Code (for testing model/redirection logic)
2. Documentation/Report

Final Presentation Reminder

December 10th 4-5pm (Tuesday)

Questions