ChemNovus **HR Assistant chatbot**

May / 2025

Presented by

Eugene Hao

Content

- What is built
- Tech Stack
- RAG workflow
- Prompt engineering
- Demo
- Production roadmap
- Key challenges
- Business impact
- Q&A

What is built

- Gradio web interface for HR policy Q&A
- Powered by OpenAI GPT-4o with RAG architecture
- Ingest DOCX, PDF, PPTX, XLSX files as the ground truth
- Region-aware responses (Global/US/EU)
- Conversational history support

Tech stack

1

Gradio App

open-source Python package to build web app for ML model, API

2

LLM

OpenAI GPT-4o

3

Embedding

text-embedding-ada-002

4

Search

Cosine similarity for KNN

RAG workflow

1

Ingestion

- Process documents by region (Global / US / EU)
- Chunk text (2000 chars), stored in memory

2

Embedding

- Generate ADA-002 embeddings
- Store the embedding in memory

3

Query

- Embed user question
- Find top-20 similar chunks

4

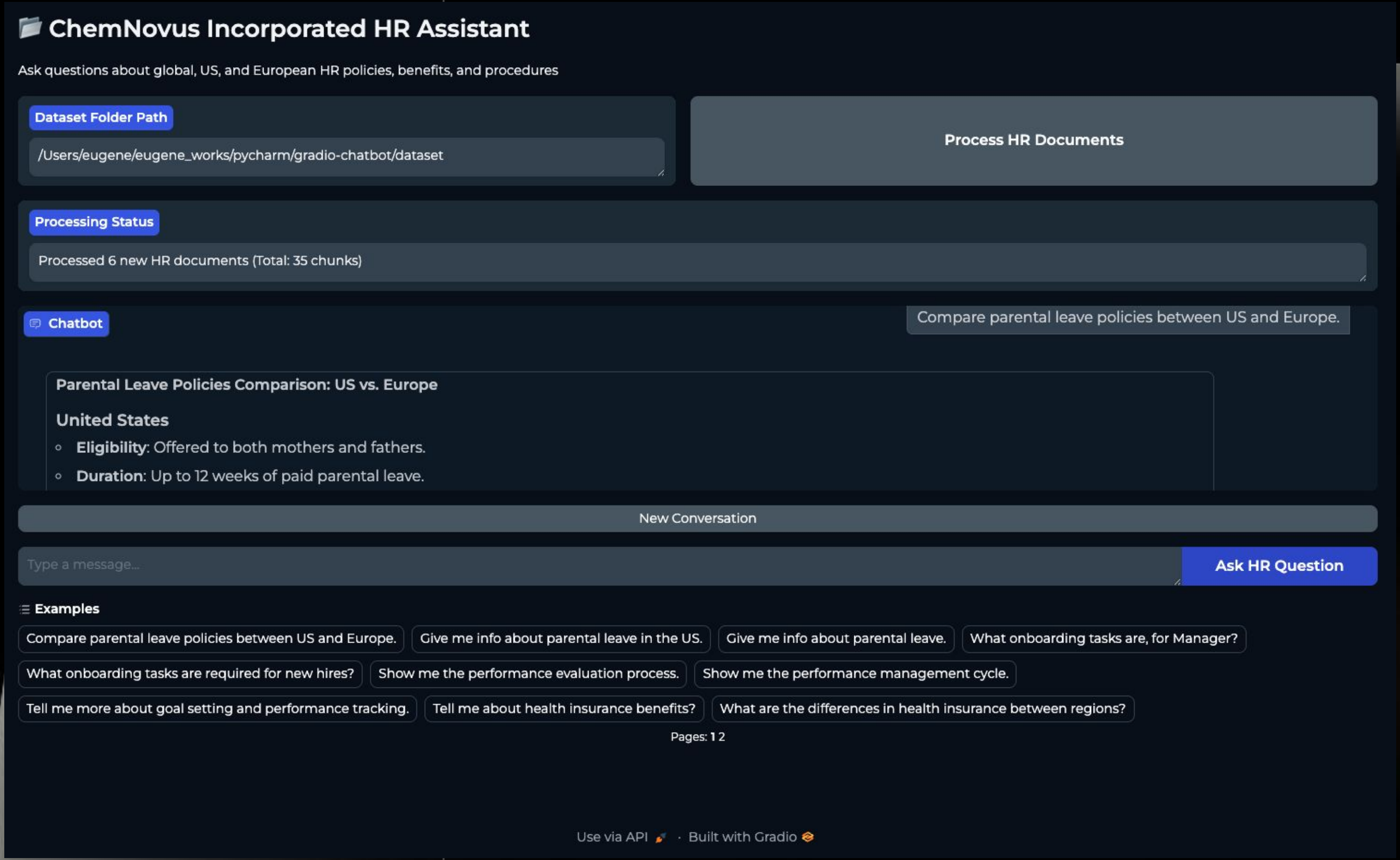
Generation

- Inject chunks + history into GPT-4o prompt
- Return formatted response

Prompt Engineering

```
"role": "system",
"content":
"You are an HR assistant for ChemNovus
Incorporated."
 ● Answer questions using ONLY the
   provided documents.
 ● Pay special attention to
   region-specific information (Global,
   US, Europe).
 ● When information differs between
   regions, present it in a clear
   comparison format.
 ● For global policies, indicate they
   apply worldwide.
 ● Format responses with clear section
   headers and bullet points.
```

Demo



📂 ChemNovus Incorporated HR Assistant

Ask questions about global, US, and European HR policies, benefits, and procedures

Dataset Folder Path: /Users/eugene/eugene_works/pycharm/gradio-chatbot/dataset

Processing Status: Processed 6 new HR documents (Total: 35 chunks)

Chatbot

Compare parental leave policies between US and Europe.

Parental Leave Policies Comparison: US vs. Europe

United States

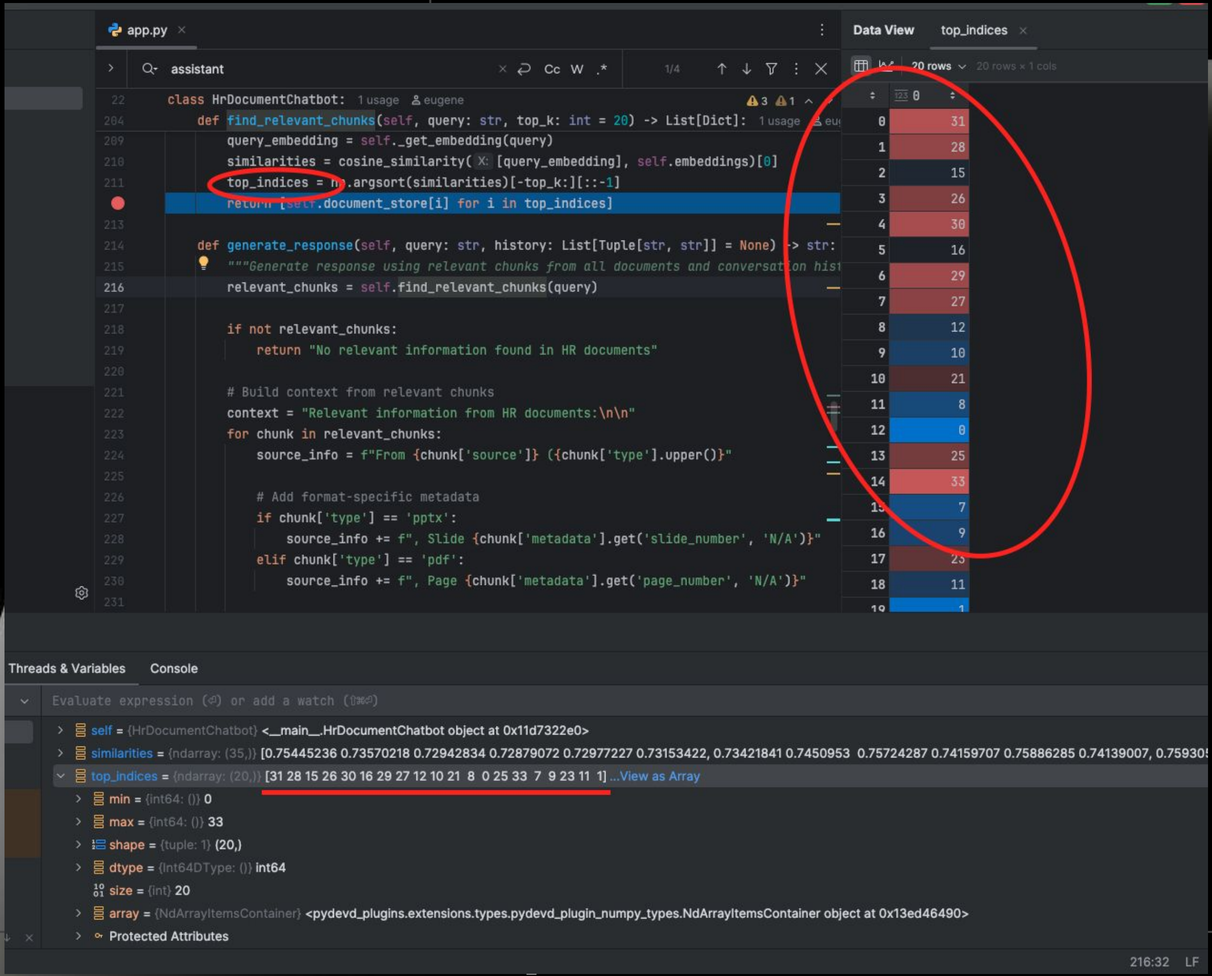
- **Eligibility**: Offered to both mothers and fathers.
- **Duration**: Up to 12 weeks of paid parental leave.

Examples

- Compare parental leave policies between US and Europe.
- Give me info about parental leave in the US.
- Give me info about parental leave.
- What onboarding tasks are, for Manager?
- What onboarding tasks are required for new hires?
- Show me the performance evaluation process.
- Show me the performance management cycle.
- Tell me more about goal setting and performance tracking.
- Tell me about health insurance benefits?
- What are the differences in health insurance between regions?

Pages: 1 2

Demo



```python
class HrDocumentChatbot:  1 usage  eugene
    def find_relevant_chunks(self, query: str, top_k: int = 20) -> List[Dict]:  1 usage
        query_embedding = self._get_embedding(query)
        similarities = cosine_similarity( X: [query_embedding], self.embeddings)[0]
        top_indices = np.argsort(similarities)[-top_k:][::-1]
        return [self.document_store[i] for i in top_indices]

    def generate_response(self, query: str, history: List[Tuple[str, str]] = None) -> str:
        """Generate response using relevant chunks from all documents and conversation hist
        relevant_chunks = self.find_relevant_chunks(query)

        if not relevant_chunks:
            return "No relevant information found in HR documents"

        # Build context from relevant chunks
        context = "Relevant information from HR documents:\n\n"
        for chunk in relevant_chunks:
            source_info = f"From {chunk['source']} ({chunk['type'].upper()}"

            # Add format-specific metadata
            if chunk['type'] == 'pptx':
                source_info += f", Slide {chunk['metadata'].get('slide_number', 'N/A')}"
            elif chunk['type'] == 'pdf':
                source_info += f", Page {chunk['metadata'].get('page_number', 'N/A')}"
```

	0
0	31
1	28
2	15
3	26
4	30
5	16
6	29
7	27
8	12
9	10
10	21
11	8
12	0
13	25
14	33
15	7
16	9
17	23
18	11
19	1

```
> self = {HrDocumentChatbot} <__main__.HrDocumentChatbot object at 0x11d7322e0>
> similarities = {ndarray: (35,)} [0.75445236 0.73570218 0.72942834 0.72879072 0.72977227 0.73153422, 0.73421841 0.7450953  0.75724287 0.74159707 0.75886285 0.74139007, 0.75930
v top_indices = {ndarray: (20,)} [31 28 15 26 30 16 29 27 12 10 21  8  0 25 33  7  9 23 11  1] ...View as Array
    > min = {int64: ()} 0
    > max = {int64: ()} 33
    > shape = {tuple: 1} (20,)
    > dtype = {Int64DType: ()} int64
      size = {int} 20
    > array = {NdArrayItemsContainer} <pydevd_plugins.extensions.types.pydevd_plugin_numpy_types.NdArrayItemsContainer object at 0x13ed46490>
    > Protected Attributes
```

Productionize Roadmap

Infra

- Dockerize app + Kubernetes deployment
- Persistent vector database (Pinecone/Weaviate)
- Redis for conversation history

Performance

- Cache common queries
- Async document processing
- Rate limiting
- PCA for embedding (shorter vector)

Enhancements

- PDF text extraction improvement (OCR for scans)
- User history and recommendation
- Feedback mechanism (thumbs up/down)
- Doc gen: work verification
- ElasticSearch to replace in-mem storage
  - disk
  - ANN (faster than sklearn linear knn scan)
  - filter by region
- CI / CD:
  - Regression test
  - versioned release
  - Deploy (k8s rollout)

Monitoring

- Log LLM token usage/costs
- Track unanswered questions
- Alert for document changes
- 4xx / 5xx error
- CPU / Mem usage

Key Challenges

Chunk size

- chunk size 2000 chars works ok
- ada-002 token limit: 8191 tokens
  - avg 4 chars / token
  - 2000 / 4 = 500 token per chunk

Top k

- top_k = 20
- when used top_k = 3, missed some answers for question ***"What onboarding tasks are required for new hires?"***

Temperature

- temperature=0.3
- Lower values (e.g., 0.3) make the output more focused and deterministic.
- Higher values (e.g., 1.0 or above) make the output more random and creative.

Region

- should categorize the region for the ingested unstructured documents

Business impact

- Huge reduction in HR policy human communication
- Instant access to latest documents
- Quick response then email HR (24 x 7)

Q & A