Course: CS 6320.001 - Natural Language Processing - S25
University: University of Texas at Dallas (UTD)
Department: Computer Science
Instructor: Dr. Tatiana

Project Report: Mental Health Support Chatbot

Youtube Link to Demo: https://www.youtube.com/watch?v=bWrb6nlwzFs

Team Members:

- Nikita Kachane: nsk230001
- Samad Mehndi: sxm230312
- Nikita Ramachandran: nxr200026

Objective

The goal of this project is to develop a mental health chatbot capable of understanding user queries and responding with empathetic, relevant answers. The chatbot uses Natural Language Processing (NLP) techniques, including intent classification and semantic similarity matching. It is built using FastAPI for the backend and incorporates HuggingFace Transformer models and Sentence-BERT for natural language understanding. A React-based frontend allows users to interact with the chatbot via a web interface.

Approach

Our system combines two NLP strategies: intent classification and semantic search. When a user sends a message, the chatbot first predicts the intent using a Transformer-based classifier. If the intent does not yield a confident response, the chatbot falls back on semantic similarity matching against a dataset of mental health FAQs.

The chatbot loads a fine-tuned HuggingFace transformer model for classification and a pre-trained SentenceTransformer (`all-MiniLM-L6-v2`) for computing embeddings. The system compares user query embeddings to a precomputed set of embeddings from our FAQ dataset to retrieve the most semantically similar question and return its corresponding response.

The backend is served using FastAPI and accessible via a RESTful `/chat` endpoint. The frontend, built with React, sends user input to the backend and displays the chatbot's responses in real time.

Enhancements Over Base Models

Course: CS 6320.001 - Natural Language Processing - S25
University: University of Texas at Dallas (UTD)
Department: Computer Science
Instructor: Dr. Tatiana

- **Offline Model Loading:** Supports fast startup using pre-downloaded HuggingFace models.
- **Custom Intent Classifier:** Trained classifier for common mental health intents using HuggingFace.
- **Precomputed Semantic Embeddings:** FAQ embeddings are precomputed for fast semantic search at runtime.
- **Semantic Matching Logic:** Implements cosine similarity fallback when intent classification is uncertain.
- **Modular Design:** Clean separation of backend API logic, chatbot NLP logic, and frontend UI components.
- **React Frontend Integration:** Built and deployed a modern React-based interface for chatbot interaction.

Lessons Learned

- **Data Preprocessing is Critical:** Clean, structured FAQ data improved accuracy and relevance.
- **Model Quality Matters:** `all-MiniLM-L6-v2` provided a good balance of speed and semantic matching quality.
- **FastAPI is Powerful:** Easy to integrate with machine learning logic and lightweight to serve locally.
- **Frontend-Backend Integration Requires Testing:** CORS issues and formatting mismatches needed debugging.
- **Working Offline Has Challenges:** Managing model caches and symlinks was necessary for stability.
- **Team Collaboration Is Key:** Regular meetings and clear version control helped resolve blockers efficiently.

Contributions

Nikita Kachane

- Researched sentence embedding options and selected Sentence-BERT.
- Implemented semantic similarity logic using Sentence-BERT.
- Cleaned and preprocessed the mental health FAQ dataset.
- Assisted with initial embedding generation and Pickle file management.

Samad Mehndi

- Developed the **FastAPI** backend and routing logic.

Course: CS 6320.001 - Natural Language Processing - S25
University: University of Texas at Dallas (UTD)
Department: Computer Science
Instructor: Dr. Tatiana

- Integrated backend with **React frontend** and ensured compatibility.
- Set up Uvicorn and manage the environment setup for deployment.
- Coordinated GitHub repo and managed project dependencies.
- Helped test and debug full-stack integration between frontend and backend.

Nikita Ramachandran

- Developed and fine-tuned the **intent classifier** using HuggingFace.
- Handled **model loading**, cache management, and offline functionality.
- Implemented Pickle-based serialization for embeddings and label encoding.
- Authored major portions of this report and documentation.

Self-Scoring

Nikita Kachane

- **80 points - Significant exploration beyond baseline**
  Explored various sentence embedding models, implemented cosine similarity fallback with Sentence-BERT, and handled preprocessing for semantic search.

- **30 points - Innovation or Creativity**
  Integrated a dual-strategy approach with both semantic search and intent classification for improved coverage.

- **10 points - Highlighted complexity**
  Cleaned and aligned unstructured mental health FAQ data; managed precomputed embeddings for performance.

- **10 points - Lessons learned and potential improvements**
  Contributed to team discussions and documentation of lessons about embedding performance, caching, and integration.

- **10 points - Exceptional visualization/diagrams/repo**
  Contributed to organizing the GitHub repo and data/embedding documentation.

- **10 points - Testing outside team**
  Helped test chatbot outputs with non-team members to validate semantic matching relevance.

Course: CS 6320.001 - Natural Language Processing - S25
University: University of Texas at Dallas (UTD)
Department: Computer Science
Instructor: Dr. Tatiana

- **10 points - Earned money**
  Not applicable

Total: 150 / 160
Deductions: None

Samad Mehndi

- **80 points - Significant exploration beyond baseline**
  Built and tested FastAPI backend, integrated it with React frontend, and resolved cross-component communication issues.

- **30 points - Innovation or Creativity**
  Bridged RESTful backend and React frontend, creating a user-friendly live chatbot experience.

- **10 points - Highlighted complexity**
  Managed local dev environment, dependency resolution, CORS setup, and React state/prop flows.

- **10 points - Lessons learned and potential improvements**
  Documented the value of modular design, runtime testing, and frontend-backend alignment.

- **10 points - Exceptional visualization/diagrams/repo**
  Led repo cleanup and ReadMe structure for clarity.

- **10 points - Testing outside team**
  Shared the chatbot web interface with friends to test usability and detect edge-case behavior.

- **10 points - Earned money**
  Not applicable

Total: 150 / 160
Deductions: None

Nikita Ramachandran

- **80 points - Significant exploration beyond baseline**
  Developed and fine-tuned the intent classifier using HuggingFace, including label encoding and cache handling.

Course: CS 6320.001 - Natural Language Processing - S25
University: University of Texas at Dallas (UTD)
Department: Computer Science
Instructor: Dr. Tatiana

- **30 points - Innovation or Creativity**
  Enabled offline model functionality and optimized model loading with caching and Pickle serialization.

- **10 points - Highlighted complexity**
  Managed multi-model coordination (intent and embeddings), and wrote key integration code.

- **10 points - Lessons learned and potential improvements**
  Wrote the project report's reflections and future improvement sections in detail.

- **10 points - Exceptional visualization/diagrams/repo**
  Contributed to repo documentation and report clarity.

- **10 points - Testing outside team**
  Collected and analyzed feedback from 5 external testers for intent classification accuracy.

- **10 points - Earned money**
  Not applicable

Total: 150 / 160
Deductions: None

Conclusion

This project demonstrates the effectiveness of combining intent classification and semantic search to build an empathetic, responsive mental health chatbot. It features a modular design, optimized inference, and a full-stack implementation with both backend (FastAPI) and frontend (React). The system can be further scaled and improved with session tracking, cloud deployment, and multilingual support.

Future Improvements

- Add user session tracking and memory of past interactions.
- Deploy the full stack to a cloud platform (AWS).
- Expand the current **React frontend** with login, themes, and better UI/UX.
- Include escalation logic for crisis cases with professional referrals.
- Add multilingual support using translation APIs or multilingual models.