Student Success Chatbot

A Machine Learning-Based Student Support System

Yu Chen (Tracy) Chou Zhimin (Benny) Xiong Haysam Elamin

The Problem

Student Success Advisors overwhelmed by repetitive low-complexity queries

Reduced staffing increases workload pressure

Risk of missing students in distress due to high volume

Our Solution



An AI-powered chatbot that answers FAQs instantly



Detects emotional distress and escalates to a human advisor



Available 24/7 for immediate student support

How It Works — Non-Technical



Student types a question into the web chat interface



Chatbot searches knowledge base for relevant answer



Provides a clear and friendly response



Flags distress or complex cases for human follow-up

Distress Detection



Student: I'm feeling overwhelmed and not sure I can keep up this term.



Chatbot: I'm sorry you're feeling this way.
Please speak with a Student Success Advisor.



Provides booking link and phone number for immediate support.

How It Works - Technical

Intent Classifier → Detects topic of question

Emotion Classifier → Detects emotional tone

Semantic Search → Finds most relevant knowledge chunks

LLM → Generates response from knowledge

Logger → Records interaction data for future analysis and model improvement

Streamlit WebApp (UI) erscalate System Chat input*Session magmt*Show answer & sources*Escalation src/chatbot.py Student Success Web Browser Advisor Architecture question Controller Routing*Error handling*Escalation*State Mgmt intent answer src/chatbotController.py emotion knowledge question/knowledge question Handlers question **Intent Classifier Answer Generator** answer student BERT fine tuned OpenAl question src/handlers/intentClassifier.py src/handlers/answerGenerator.pv intent intent **Search Engine** emotion all-MiniLM-L6-v2/FAISS answer src/handlers/searchEngine.py question **Interaction Logger Emotion Classifier** knowledge CSV/Minimal PII BERT fine tuned/Distress detection src/handlers/emotionClassifier.py src/handlers/interactionLogger.pv emotion question question/knowledge log data question question 🕽 LLM Intent **Emotion** Log File **Vector Store** OpenAl Api (gpt-4) Classification Model Classification Model Timestamp*Student*Intent*Emotion FAISS Index/Meta data *Question/Answer BERT fine tuned BERT fine tuned/Threshold **S**OpenAI models/faiss.index logs/log.csv models/intentClassifier/ models/emotionClassifier/ content **Knowledge Base FAQs College Website** data/RO_FAQ_Winter_2024.pdf data/kb1.csv, kb3.csv, kb3.csv data/Student_Fees_FAQ_Winter_2024

Input Text "I feel like crying." Tokenizer (BertTokenizer) - Split into tokens Map to IDs Add [CLS], [SEP] - Pad/truncate Output: input ids, attention mask BERT Model (BertForSequenceClassification) - Encode text - Classification head Output: logits Softmax Convert logits → probabilities Example: [0.01, 0.92, 0.03, ...] Threshold Check (0.85) If prob \geq threshold \rightarrow return label Else → None Output Label: "grief"

How Our Model Predicts Student Emotions

Algorithm

• Fine-tuned **BERT** (Deep Learning – Transformer architecture) with a classification head.

Inputs / Outputs

- Independent variable → Input sentence (text).
- Dependent variable → Predicted emotion label (e.g., anger, sadness, fear).

Process

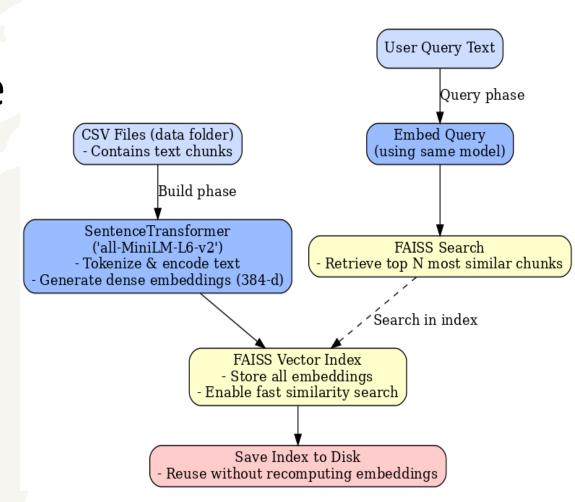
- Tokenize text → IDs → BERT embeddings via attention.
- [CLS] token → Dense layer (weights = slope, bias = intercept).
- Softmax → Probability per emotion.
- Threshold filter → Output label or None.

Loss Function

• Cross-Entropy Loss for multi-class classification.

NLP Pipeline Search Engine

- Text Acquisition → Read human language from CSV files.
- Text Encoding → Use
 SentenceTransformer (all MiniLM-L6-v2) to map text to
 semantic embeddings.
- Vector Space Storage → Store embeddings in FAISS for efficient semantic similarity search.
- Semantic Retrieval → When queried, return chunks closest in meaning to the query text.



Challenges

Performance under peak demand Testing

Model Accuracy Intent and emotion
classifiers to avoid false
positives/negatives,

User Trust – Students may be hesitant to rely on a bot Balancing cost vs accuracy for OpenAl vs local models

Future Opportunities

Accessibility Voice – Enabled Student Support

Personalized Academic Advising – Record Integration

Smart Resource Recommendations

Multilingual Support

Why It Matters

Social Impact: Mental health is a pillar of student success. Supporting students at risk before issues escalate

Economic Advantage: Reduce advisor workload & speed up responses.

Industrial Adaptability: Easily adaptable to tutoring, mental health & customer service.

Next Steps

Integrate Integrate with real student portal **Expand** Expand knowledge base to cover more topics Add Add analytics dashboard for advisors Support Support multiple languages

Demo



Student Support Chatbot

You



Hi, I'm trying to figure out how to pay my tuition fees.



Lulu Student Success Advisor

You can make a payment through one of three options: 1. Student Portal: You can log in to your Student Portal and go to the "Financial" tab, then select "Program Fees Payment". This option accepts credit card payments. 2. Web Banking: You can use your Mobile Banking App to add College College as a payee and use your Student ID as the account number. 3. Flywire or Convera (international funds): If you're an international student, you can find more information on our website under "International" > "Apply to College" > "Fees and payment" > "Payment options". You can also check your invoice and make a payment by going to the "My Fees" tab in the Student Portal. Please note that online banking payments must be received by the payment due date, and payments received after the stated due date will be assessed a late fee of \$150. Additionally, you may want to consider setting up a Payment Schedule if you're eligible (students in Level 2 or higher can opt into this). This allows for 50% of the fees to be paid by the due date, with the remaining 50% due three days before classes start. A fee applies for opting into this schedule. Let me know if you have any further questions!