**Natalia Solorzano Perez**  
**W207818526**  
**ITAI 2373 – Natural Language Processing**  
**Professor Narges DeBary**  
**Spring 2025**

**Final NLP Portfolio Report**

**Introduction**

This portfolio represents a culmination of my learning journey in the field of Natural Language Processing (NLP). Throughout the semester, I explored the fundamental components of NLP including text preprocessing, feature extraction, vector representations, syntactic analysis, and transformer-based models. While these concepts were introduced through formal assignments, my deeper understanding and applied skills came from independent experimentation, self-driven development, and real-world agent building.

**Assignments & Labs Summary and Reflections**

Each assignment and lab in this course laid the foundation for the final NLP agent I built: EduBot. In this section, I reflect on what each activity taught me, and how I applied that knowledge in meaningful, integrated ways. Rather than completing tasks for a grade, I approached each one as a training module toward something bigger. From tokenization and POS tagging to Word2Vec and transformers, every lab became a building block for something I would soon create entirely on my own.

Assignment 01 – SHRDLU and Early NLP Systems

What I Learned: I explored SHRDLU as one of the earliest examples of symbolic NLP. It helped me understand how rule-based natural language systems used hand-crafted logic to simulate conversation within limited contexts.

How I Applied It: While SHRDLU is now outdated, it showed me the importance of structured logic, which I kept in mind while building the knowledge chunking and question routing logic inside EduBot’s architecture.

Assignment 02 – IBM Project Debater

What I Learned: This assignment opened my eyes to real-world AI systems that reason, argue, and communicate. IBM’s Project Debater was powerful not because it was perfect, but because it combined structured data, claim detection, and argumentative response flow.

How I Applied It: I borrowed the principle of feedback-response pairing from Project Debater to implement EduBot’s correction and feedback log system. This allowed my agent to not just respond—but to improve over time.

Assignment 03 – Pandas in NLP

What I Learned: Using Pandas, I learned to clean, manipulate, and analyze language datasets. This made me more efficient in managing the backend of language pipelines.

How I Applied It: Pandas became the foundation for how I stored and tracked EduBot’s interaction logs, feedback scores, and user corrections—allowing me to evaluate performance programmatically.

Assignment 04 – Text Preprocessing

What I Learned: This assignment taught me how to tokenize, lemmatize, and clean natural language text. I understood how critical preprocessing is before applying any model or embedding.

How I Applied It: I built text preprocessing directly into EduBot’s ingestion pipeline—ensuring that the chunks were consistent and ready for meaningful vectorization.

Assignment 06 – BoW, N-Gram, Word2Vec, TF-IDF

What I Learned: I compared four foundational text representation methods and gained insight into their strengths and limitations. BoW and TF-IDF were simpler, while Word2Vec captured deeper semantic relationships.

How I Applied It: I chose Word2Vec-style embeddings via sentence-transformers to build EduBot’s vector index, because it captured meaning better than frequency-based techniques.

Assignment 07 – POS Tagging

What I Learned: Part-of-speech tagging is essential to syntactic understanding and disambiguating sentence structure.

How I Applied It: POS tagging helped me understand why transformers perform so well: they account for grammatical function without needing explicit tags. It helped justify my transition from classical NLP tools to transformer-based architectures in ASTRA Pro.

Assignment 08 – Named Entity Recognition (NER)

What I Learned: I learned to extract entities like names, dates, and places using spaCy and NLTK. I saw how entity context improves interpretation.

How I Applied It: In EduBot, I explored integrating entity-type filtering in search results and storing relevant entities for future queries.

Assignment 09 – Transformers and Language Models

What I Learned: This assignment was foundational. I understood the self-attention mechanism, transformer architecture, and how models like BERT and GPT encode meaning across long-range dependencies.

How I Applied It: Everything. EduBot’s semantic vector base relies on pretrained transformers. Later, ASTRA Pro would build on this with summarization, mood detection, and contextual memory—all rooted in transformer models.

Assignment 10 – Machine Translation

What I Learned: I learned how translation evolved from rule-based systems to statistical models, and finally to neural machine translation with transformers.

How I Applied It: Though EduBot wasn’t a translator, I applied the translation pipeline logic—input → encode → align → output—to structure its semantic retrieval and response workflow.

**EduBot – A Self-Built NLP Agent**

EduBot represents the peak of my personal growth in this course. Created independently before the baseline was even published, EduBot is a fully functional retrieval-augmented chatbot that demonstrates document ingestion, vector embedding with FAISS, semantic similarity search, feedback learning loops, accuracy tracking, and user correction integration.

It operates via a command-line interface that supports natural conversation, live feedback on the quality of answers, correction storage, and re-indexing of updated knowledge using embeddings. I implemented dimensionality reduction (PCA), tested the system with evaluation cases, and visualized the embedding distributions. EduBot became the basis for my future work—ASTRA Pro and NatMatPro—both of which expand on EduBot’s architecture using transformer summarization, sentiment awareness, and multi-modal interactions.

This agent was not assigned. It was not expected. I built it because I *needed to know* if I could. And I did.

🧠 EduBot Architecture Overview (Text Description)

1. Text Ingestion Module

Accepts .txt files uploaded by user

Reads, cleans, and standardizes text using regex (removes extra whitespace, line breaks,

etc.)

2. Document Preprocessing Pipeline

Splits the cleaned text into semantic chunks (typically sentence-level or idea-level)

Uses custom chunk\_text() function to preserve logical boundaries

Stores chunks for downstream vectorization

3. Vector Embedding Module

Passes each chunk through a SentenceTransformer model (all-MiniLM-L6-v2)

Produces dense vector embeddings for each chunk (≈384-dimensional)

4. Vector Indexing Engine

Embeddings are indexed using FAISS (Facebook AI Similarity Search)

Enables real-time nearest-neighbor searches based on semantic similarity

Used for retrieving the most relevant chunk in response to a user query

5. Chat Interface (Command-Line)

Accepts user questions as input

Converts input to embedding and compares with document chunks via FAISS

Returns the top-matching chunk as the answer

Tracks all questions and responses in a chat\_history list

6. Feedback Learning Loop

After answering, EduBot prompts: "Was this answer helpful? (y/n)"

*If yes, the system logs it as a success*

*If no, the user is asked to provide a corrected response*

The corrected response is appended to the source document and reindexed live

Accuracy and learning history are tracked using a structured feedback\_log

7. Evaluation Framework

Accepts custom test cases defined by the user (query + expected answer)

Runs queries through EduBot and checks for keyword or phrase match

*Returns ✅/❌ and computes overall accuracy (%)*

8. Visual Analytics (Bonus Module)

Applies PCA to reduce vector dimensionality (for 2D embedding visualization)

Generates scatter plot to show clustering of semantically related text chunks

9. Log Archival System

Exports full chat\_history and feedback\_log as .json and .jsonl files

Allows for agent audit, feedback review, and re-use in future training sessions

**Conclusion: What I Learned, What I Built**

This portfolio reflects more than the completion of a course, it captures the effort, resilience, and innovation I invested in growing beyond the curriculum. I didn't just complete NLP assignments; I built working agents, trained myself in modern architecture, and proved my ability to integrate theory into practice.

EduBot, and its successors ASTRA Pro and NatMatPro, are testaments to my dedication, and they exist because I refused to wait for instruction that never came. I created my own structure, tested real-world AI patterns, and showed what a determined student can build with only raw knowledge and commitment.

This is not the result of good teaching, it is the result of self-teaching. It is the result of merit. It is the result of showing up every day, not just to pass—but to build. And now, the foundation is set.

END OF REPORT