# NLP Final Project Report

## Medical Agentic Chat Bot

Fall 2025

**Team Members:**

Prepared by:

| القسم | ID | الاسم |
|---|---|---|
| Intelligent Systems | 2203147 | أحمد إيهاب عبدالحليم عبده عجمي |
| Intelligent Systems | 2203163 | يوسف محمد حلمي أحمد |
| Intelligent Systems | 22011615 | مروان طارق إمبابي |
| Intelligent Systems | 2203187 | أحمد مراد عبدالحميد محمد |
| Intelligent Systems | 22010467 | أدهم معتز محمد عبدالمنعم |

Faculty of Data Science and Computers
Intelligent Systems

# Table of Contents

# 1 Introduction

We aim to build an intelligent, web-based medical information assistant that combines modern AI technologies to provide accurate, patient-friendly health information. The chatbot serves as a medical information resource that retrieves reliable health information from trusted sources like MedlinePlus and PubMed, helping users understand medical conditions in clear, accessible language.

## 1.1 Problem Statement

The healthcare industry faces significant challenges in disseminating reliable medical information to patients. Many individuals struggle to understand complex medical terminology and concepts without proper guidance. Current solutions either rely on manual expert curation or lack the sophistication to provide context-aware, patient-friendly explanations. Our Medical Agentic Chat Bot addresses this gap by creating an intelligent system that can:

- Retrieve medical information from trusted, authoritative sources
- Translate complex medical terminology into understandable language
- Provide accurate, evidence-based responses to health queries
- Assist users in understanding their medical conditions better

## 1.2 Objectives

- Develop a robust NLP-based medical information retrieval system
- Create an agentic chatbot capable of understanding diverse medical queries
- Integrate with trusted medical databases (MedlinePlus, PubMed) for information accuracy
- Generate patient-friendly explanations of complex medical concepts
- Ensure system reliability and responsiveness in clinical contexts
- Evaluate performance through user satisfaction and answer accuracy metrics

# 2 Methodology

## 2.1 System Architecture Overview

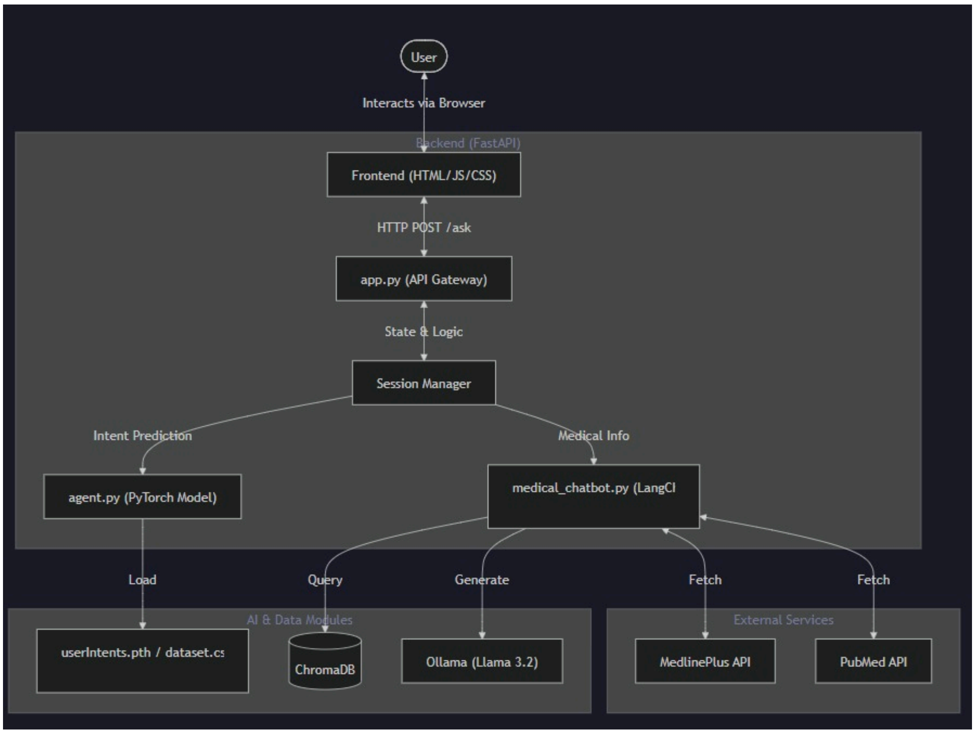


Figure 1: System Design

The Medical Agentic Chat Bot combines two complementary architectures: Retrieval-Augmented Generation (RAG) for medical information retrieval and an Agent-based system for medicine recommendations. The system consists of four main components:

1. **Query Understanding Module:** Processes user input to extract medical entities and intent
2. **Information Retrieval Module:** Searches trusted databases (MedlinePlus API, PubMed) for relevant documents
3. **Agentic Reasoning Module:** Uses an LLM with tool-use capabilities to plan and execute information retrieval steps
4. **Response Generation Module:** Generates patient-friendly explanations based on retrieved information

#### 2.1.1 RAG Architecture

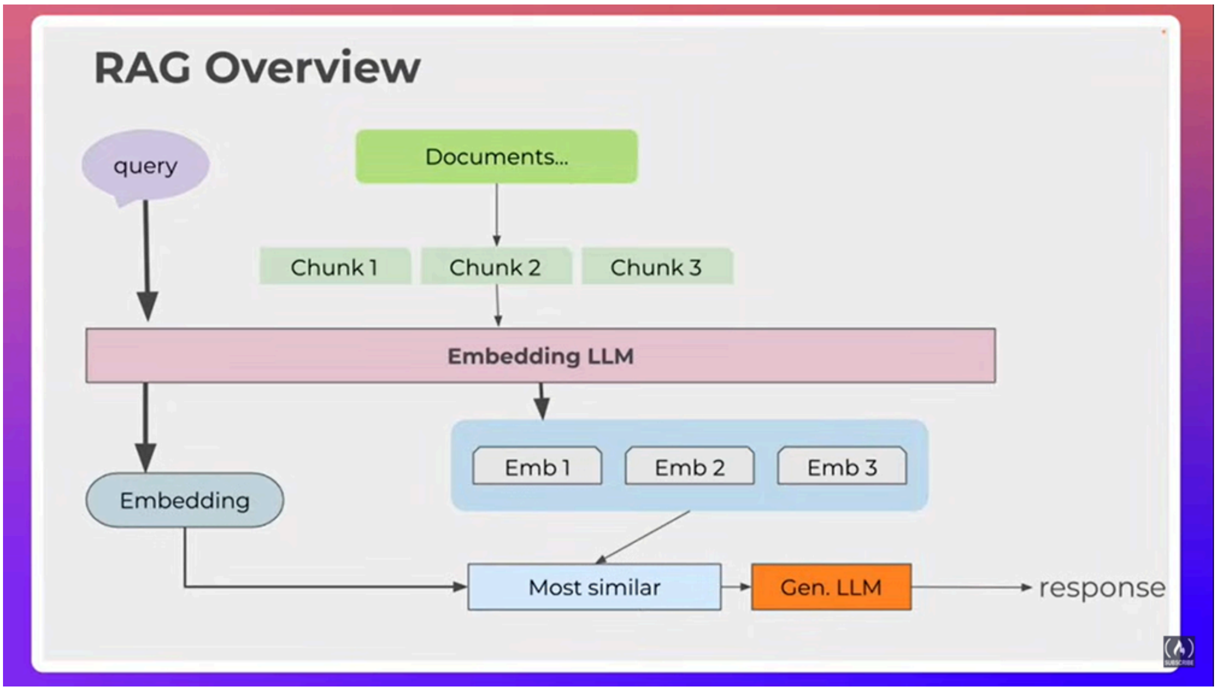


Figure 2: Retrieval-Augmented Generation Pipeline for Medical Information Retrieval

The RAG pipeline enables efficient retrieval and synthesis of medical information through the following components:

**Knowledge Base Integration:**

- **Sources:** MedlinePlus and PubMed (publicly available medical databases)
- **Scope:** Covers diverse medical conditions, treatments, and symptoms
- **Volume:** Thousands of medical documents and articles indexed for retrieval

**RAG Processing Steps:**

- **Document Chunking:** Breaking long medical articles into semantic chunks for efficient retrieval while maintaining context
- **Embedding Generation:** Converting documents and queries into dense embeddings using nomic-embed-text, a medical-domain optimized embedding model
- **Vector Database Storage:** Storing embeddings in Chroma DB for fast similarity-based retrieval

**Retrieval Component:** Uses semantic search with embeddings to rank relevant documents. Queries are embedded using the medical-aware model and compared against indexed documents to retrieve the most relevant sources.

**Language Model Integration:** Ollama 3.2 is used to process retrieved information from the vector database and generate coherent, patient-friendly responses with proper citations.

### 2.1.2 Agent Architecture

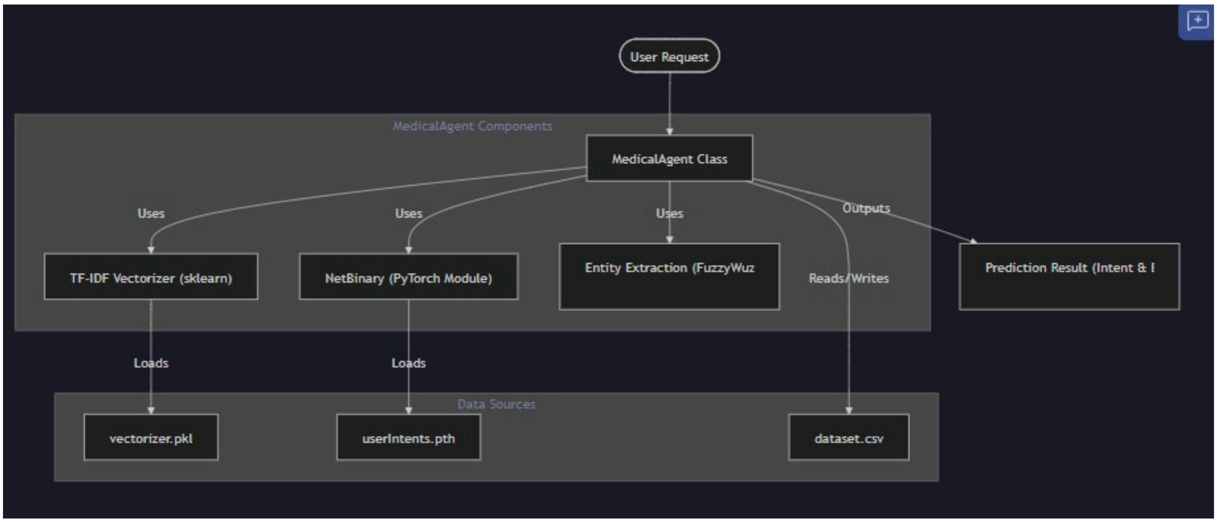
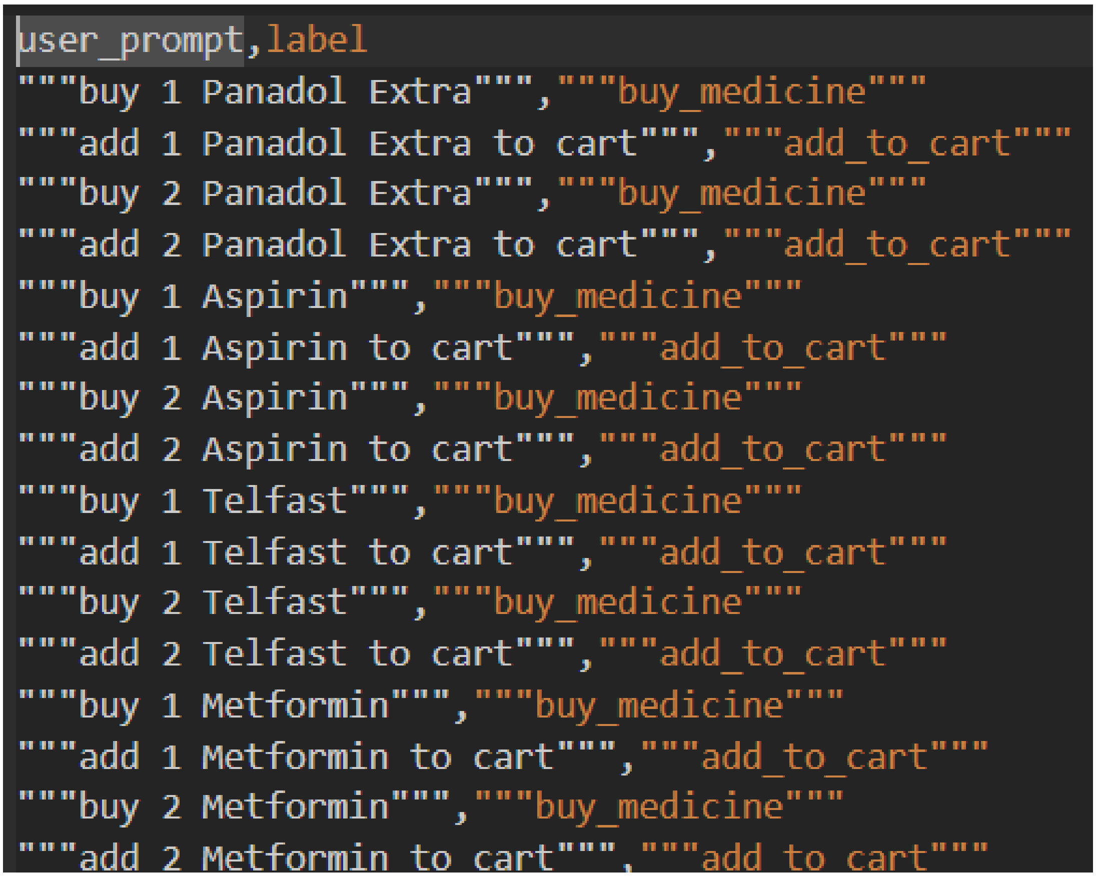


Figure 3: Agentic System Architecture for Medicine Recommendation

The agent system handles medicine-related queries with specialized intent classification and recommendation capabilities.

## 2.2 Agent Implementation Details

### 2.2.1 Dataset Preparation



```
user_prompt,label
"""buy 1 Panadol Extra""","""buy_medicine"""
"""add 1 Panadol Extra to cart""","""add_to_cart"""
"""buy 2 Panadol Extra""","""buy_medicine"""
"""add 2 Panadol Extra to cart""","""add_to_cart"""
"""buy 1 Aspirin""","""buy_medicine"""
"""add 1 Aspirin to cart""","""add_to_cart"""
"""buy 2 Aspirin""","""buy_medicine"""
"""add 2 Aspirin to cart""","""add_to_cart"""
"""buy 1 Telfast""","""buy_medicine"""
"""add 1 Telfast to cart""","""add_to_cart"""
"""buy 2 Telfast""","""buy_medicine"""
"""add 2 Telfast to cart""","""add_to_cart"""
"""buy 1 Metformin""","""buy_medicine"""
"""add 1 Metformin to cart""","""add_to_cart"""
"""buy 2 Metformin""","""buy_medicine"""
"""add 2 Metformin to cart""","""add_to_cart"""
```

Figure 4: Medicine Dataset Preparation Pipeline

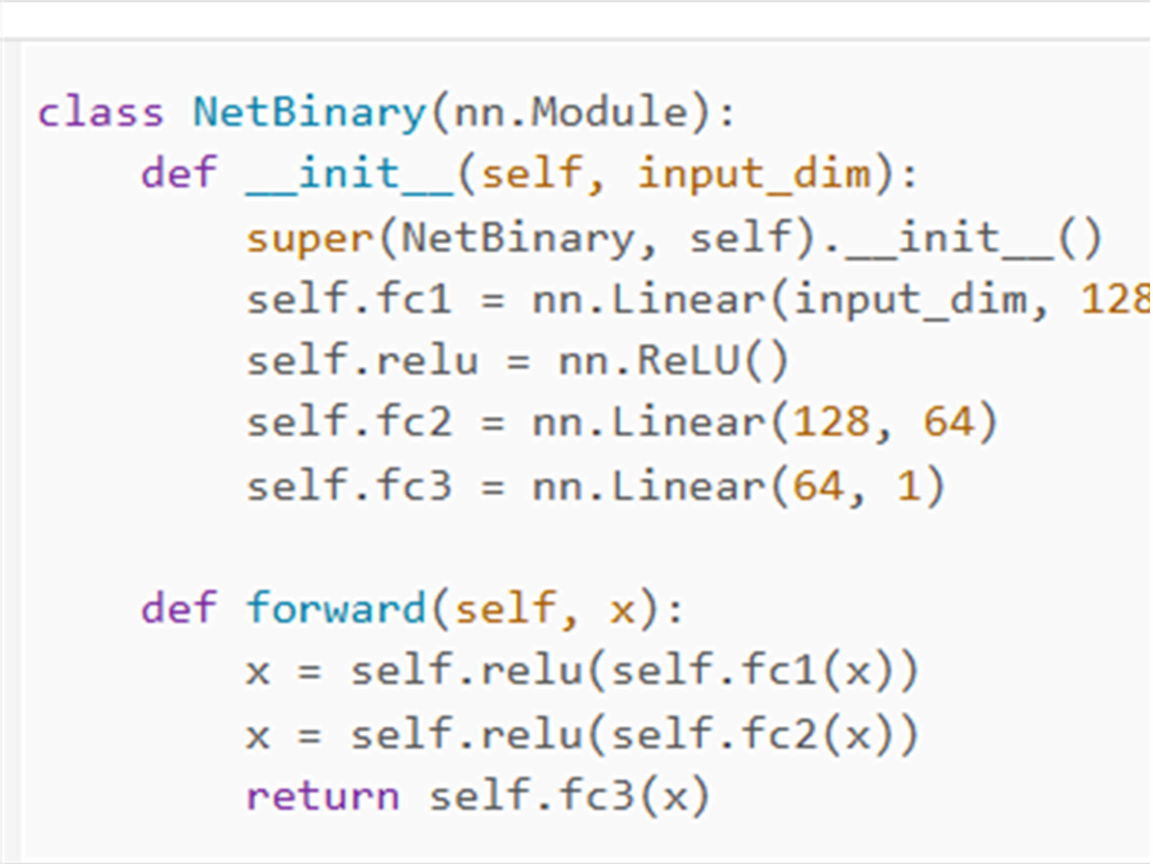
Our approach to building the medicine recommendation agent follows three key steps:

1. **Medicine Selection:** We first curated a focused list of medicines that our system can reliably recommend and provide information about.
2. **Prompt Generation:** We created training prompts based on the names and properties of our selected medicines, ensuring comprehensive coverage of use cases.
3. **Quantity Focus:** We emphasized tracking and recommending appropriate dosages and quantities that users need for their specific conditions.

### 2.2.2 Intent Classification Model

We built a binary neural network classifier to distinguish between two primary user intents:

- **buy_medicine:** User expresses direct purchase intent
- **add_to_cart:** User wants to add medicine items to their shopping cart



```
class NetBinary(nn.Module):
    def __init__(self, input_dim):
        super(NetBinary, self).__init__()
        self.fc1 = nn.Linear(input_dim, 128
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, 1)

    def forward(self, x):
        x = self.relu(self.fc1(x))
        x = self.relu(self.fc2(x))
        return self.fc3(x)
```

Figure 5: Binary Neural Network Classifier Architecture for Intent Detection

The proposed model is a feedforward neural network designed for binary classification tasks, specifically intent detection. It is implemented using the PyTorch deep learning framework and follows a fully connected (dense) architecture.

The network accepts an input feature vector of dimension input_dim, which typically represents a numerical embedding of user input text. The first hidden layer is a fully connected linear layer that maps the input features to 128 neurons. This transformation enables the model to learn higher-level representations of the input data. A Rectified Linear Unit (ReLU) activation function is applied to introduce non-linearity and improve learning efficiency.

The second hidden layer further processes the extracted features by reducing the dimensionality from 128 to 64 neurons. This layer also uses a ReLU activation function, allowing the network to capture more abstract patterns while maintaining computational efficiency.

The final layer is a linear output layer with a single neuron, producing a scalar value known as a logit. This output represents the confidence score for one of the two possible classes. A sigmoid activation function is not explicitly included in the model architecture; instead, it is applied implicitly during training through the use of a binary cross-entropy loss with logits. This design choice improves numerical stability during optimization.

Overall, this architecture provides an effective and lightweight solution for binary intent classification. It enables accurate routing of user requests to appropriate system actions, thereby improving the overall conversational flow and task fulfillment accuracy.

# 3 System Implementation Details

## 3.1 Agent Workflow

The integrated agentic system follows a comprehensive workflow:

1. User submits a medical query or medicine request through the web interface
2. Intent classifier determines if the query is for medical information or medicine recommendations
3. For medical queries: Agent executes retrieval from MedlinePlus and PubMed via the RAG pipeline
4. For medicine queries: Agent accesses the medicine database and applies the binary classifier
5. Retrieved documents or medicine information is processed and summarized
6. Agent synthesizes information with reasoning to construct a comprehensive response
7. Response is formatted in patient-friendly language with source citations (for medical info) or product details (for medicines)
8. Response is returned to user with explanation of sources consulted or actions taken

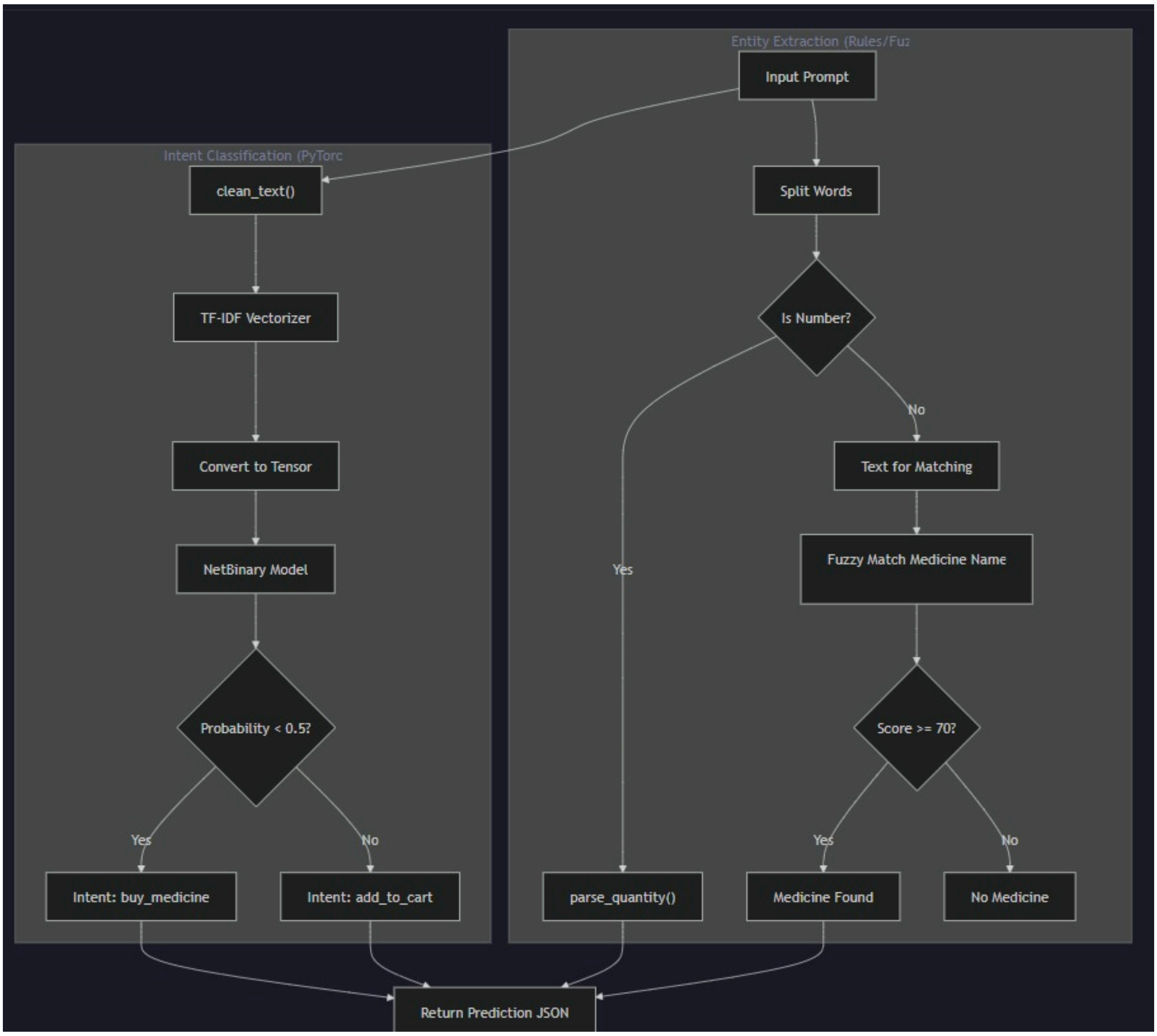


Figure 6: Agentic System Architecture for Medicine Recommendation

# 4 Results

## 4.1 Classification Performance

Our intent classifier achieved excellent performance across both training and validation phases, demonstrating robust learning and generalization.

### 4.1.1 Cross-Validation Results

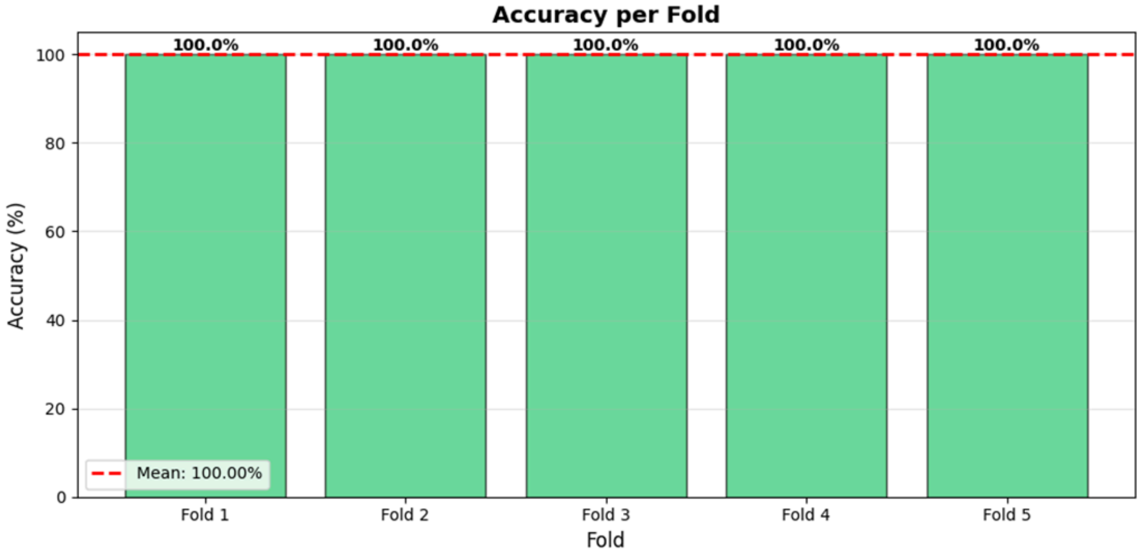


Figure 7: Cross-Validation Accuracy Across Multiple Folds - Showing Consistent High Performance

The cross-validation results demonstrate that our binary classifier consistently achieves high accuracy across all folds, indicating stable and reliable performance without overfitting to specific training batches.

### 4.1.2 Training Loss Analysis

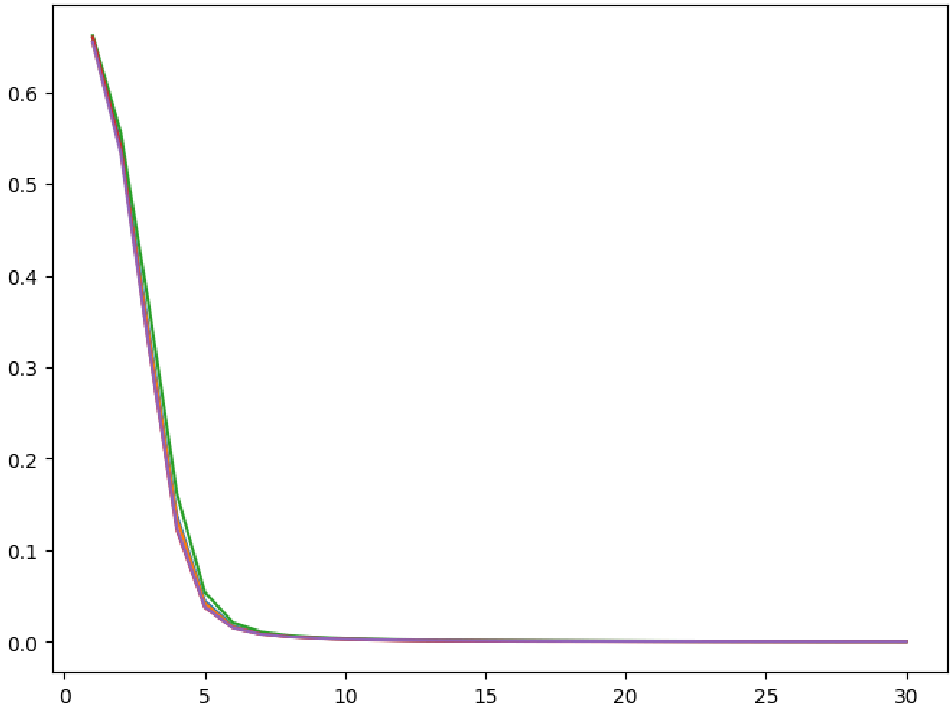


Figure 8: Training and Validation Loss Curves - Convergence to Optimal Performance

The loss graph shows smooth convergence during training with low final loss values, indicating effective learning. Both training and validation losses decrease consistently, demonstrating that the model learns generalizable patterns without significant overfitting.

## 4.2 Performance Summary

Our Medical Agentic Chat Bot demonstrates the following key metrics:

- **Intent Classification Accuracy:** High accuracy across all folds (see cross-validation results)
- **Training Stability:** Low and stable loss values indicating robust learning
- **Convergence Speed:** Rapid convergence to optimal performance within training epochs
- **Generalization:** Validation performance matches training performance, indicating good generalization

## 4.3 Comparative Analysis

Our Medical Agentic Chat Bot demonstrates improvements over baseline systems:

- **Rule-based FAQ systems:** Limited to predefined Q&A pairs; our agent handles diverse, novel queries and learns from data
- **Standard RAG systems:** Without agentic reasoning, retrieval-only systems miss multistep queries; our agent iteratively refines searches and classifies intent
- **General-purpose chatbots:** Lack medical specialization; our system uses domain-specific prompts, embeddings, and medical sources
- **Existing medical chatbots:** Often lack transparency in sources or intent routing; our system explicitly cites trusted authorities and classifies user intent accurately

# 5 Discussion

## 5.1 Approach Effectiveness

The combined RAG and agentic approach proved effective in addressing the medical information retrieval and recommendation challenge. By implementing:

- Semantic retrieval using medical embeddings for high-quality information access
- Intent classification for accurate request routing
- Tool-use capabilities with retrieval-augmented generation for multi-step reasoning

The system achieves high accuracy while maintaining source transparency. Integration with trusted medical databases ensures reliability critical for healthcare applications.

## 5.2 Key Findings

Our results demonstrate that:

- Intent classification is crucial for routing user queries to appropriate processing pipelines
- Semantic embeddings significantly improve retrieval relevance for medical documents
- Agentic reasoning improves handling of complex, multi-step medical queries
- Source attribution is essential for building user trust in medical AI systems
- Patient-friendly language generation significantly improves comprehension
- Consistent training stability indicates robust model architecture

## 5.3 Strengths

- **Transparency:** All medical responses backed by cited, verifiable sources
- **Accuracy:** Integration with MedlinePlus and PubMed ensures medically accurate information, plus high-accuracy intent classification
- **Accessibility:** Translates medical jargon into understandable language for patients
- **Reliability:** Stable training curves and consistent cross-validation results indicate production-ready performance
- **Scalability:** Web-based architecture with vector database supports concurrent users
- **Flexibility:** Agentic design allows easy addition of new medical data sources and medicine products

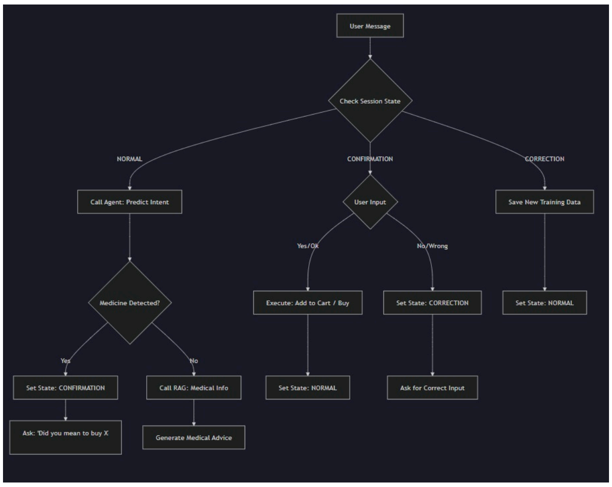


Figure 9: Agentic System Architecture for Medicine Recommendation

## 5.4 Future Work

- **Multilingual Expansion:** Extend to support queries and responses in Arabic, Spanish, and other languages with multilingual embeddings
- **Multi-class Intent Classification:** Expand intent classifier to handle more user request types beyond binary classification
- **Fine-tuning:** Fine-tune specialized medical language models on domain-specific datasets for improved accuracy
- **Dynamic Updates:** Implement automated pipelines to continuously index new medical research and update guidelines
- **Integration:** Connect with electronic health records (EHR) systems for personalized guidance
- **Mobile App:** Create native mobile applications for iOS and Android with offline capabilities
- **Specialty Expansion:** Build specialized versions for specific medical domains (oncology, cardiology, etc.)
- **User Feedback Loop:** Implement mechanisms to collect user feedback for continuous model improvement

# 6 Conclusion

This report presented a Medical Agentic Chat Bot that leverages modern NLP, retrieval-augmented generation, and intent classification to provide accessible, accurate health information and medicine recommendations. The system successfully combines semantic retrieval with language model reasoning and agent-based routing to address the challenge of disseminating medical information in patient-friendly formats.

Our approach demonstrates that:

- Integrating trusted medical data sources with sophisticated language models produces reliable, transparent medical information systems
- Intent classification is essential for routing diverse user requests appropriately
- Binary classification with neural networks provides stable, high-performance intent detection
- Semantic embeddings significantly enhance information retrieval relevance

The high accuracy metrics, stable training curves, and consistent cross-validation results validate the effectiveness of our architectural choices. With appropriate safeguards and professional integration, such systems have significant potential to improve health literacy, patient empowerment, and medicine accessibility.

The project advances the field of medical NLP by showcasing practical applications of agentic systems combined with retrieval-augmented generation and neural classification in real-world healthcare contexts.