Assignment Report: NLP Tasks

Task 1: Train an RNN Model for Text Classification

1. Goal

The objective of this task was to train a deep-learning Recurrent Neural Network (RNN) from scratch on a subset of the IMDB movie review dataset to classify reviews as either "Positive" or "Negative".

2. Methodology

- **Dataset:** IMDB Movie Reviews (via keras.datasets.imdb).
- **Data Selection:** A subset of 100 reviews was selected to simulate a low-resource data environment.
- **Data Split:**
    - **Training Set:** 80 reviews (80%) used for model learning.
    - **Testing Set:** 20 reviews (20%) reserved strictly for final evaluation.
- **Preprocessing:**
    - Text converted to integer sequences using a vocabulary of 10,000 words.
    - All sequences padded to a fixed length of 100 tokens.
- **Model Architecture:**
    - **Embedding Layer:** Transforms integer indices into dense vectors.
    - **LSTM Layer:** A Long Short-Term Memory layer (64 units) to capture sequential dependencies.
    - **Dense Output Layer:** Uses a Sigmoid activation function for binary classification.

3. Results

The model was trained for 15 epochs. The final evaluation was performed on the 20 unseen test reviews.

- **Final Training Accuracy:** 100.00%
- **Final Test Accuracy: 75.00%**
- **Loss:** 0.8754

4. Observation & Analysis

The model achieved perfect accuracy (100%) on the training data but dropped to 75% on the test data. This discrepancy is a classic sign of **overfitting**, which is expected when training deep neural networks on such a small dataset (80 samples). However, the model still generalized well enough to correctly classify 15 out of 20 unseen reviews, demonstrating the effectiveness of LSTM layers even with limited data.

Submitted By: Kamran Wahab [2025(S)-MS-AI-03]

Task 2: Run LLaMA Locally Using Ollama & Compare

1. Goal

The objective was to run a pre-trained Large Language Model (LLaMA 3) locally on a laptop using Ollama and compare its zero-shot performance against the custom RNN trained in Task 1.

2. Methodology

- **Software:** Ollama (Local Inference Server).
- **Model:** LLaMA 3 (Pre-trained).
- **Input Data:** The exact same 20 test reviews used in Task 1.
- **Process:** Each review was fed into LLaMA with a specific prompt: *"Classify the sentiment of this review as Positive or Negative."*

3. Results

- **Total Test Samples:** 20
- **Correct Predictions:** 17
- **Wrong Predictions:** 3
- **Final LLaMA Accuracy: 85.00%**

4. Comparative Analysis (RNN vs. LLaMA)

Model	Strengths	Weaknesses	Accuracy
RNN (Custom Model)	Lightweight, fast training, simple architecture.	Struggles with complex sentence structures and sarcasm; overfits on small data.	**75.00%**
LLaMA (Pre-trained)	Deep understanding of context, nuance, and sarcasm; utilizes massive pre-training knowledge.	Computationally expensive; requires higher RAM/CPU; slower inference speed.	**85.00%**

5. Conclusion:

LLaMA outperformed the custom RNN by **10%**. This highlights the advantage of Transfer Learning—using a model trained on billions of parameters versus one trained on just 80 examples.

Task 3: Build a Simple RAG System (Retrieval + LLaMA)

1. Goal

To build a Retrieval-Augmented Generation (RAG) system that enables a local LLaMA 3 model to answer questions based on **private, custom documents** that were not present in its original training data.

Submitted By: Kamran Wahab [2025(S)-MS-AI-03]

2. Methodology

- **Custom Knowledge Base:** Three proprietary text files were created:
    1. mars_colony_rules.txt (Rules for a fictional Mars colony).
    2. cyber_pet_manual.txt (Manual for a "Cyber-Dog 3000").
    3. secret_history_z.txt (Details of a classified mission "Operation Z").
- **Retrieval Mechanism:**
    - **Embeddings:** Used SentenceTransformer ('all-MiniLM-L6-v2') to convert text into vector embeddings.
    - **Chunking:** Documents were split into individual sentences.
    - **Search:** Implemented **Cosine Similarity** to find the single most relevant sentence for each query.
- **Testing:** 10 questions were formulated based on these documents.

3. Results (LLaMA Alone vs. LLaMA + RAG)

The following table summarizes the performance on the 10 test questions.

Question ID	Question Topic	LLaMA Alone (No Context)	LLaMA + RAG (With Context)	Result
Q1	Mars Colony Date	Hallucinated / Unknown	"2045"	**Correct**
Q2	Mars Currency	Hallucinated / Unknown	"Red Credit"	**Correct**
Q3	Water Ration Limit	Hallucinated / Unknown	"20 liters per day"	**Correct**
Q4	Water Fine	Hallucinated / Unknown	"500 Red Credits"	**Correct**
Q5	Cyber-Dog Charging	Hallucinated / Unknown	"Every 12 hours"	**Correct**
Q6	Cyber-Dog Reset	Hallucinated / Unknown	"Press blue button for 5 secs"	**Correct**
Q7	Cyber-Dog Languages	Hallucinated / Unknown	"English, Spanish, Binary"	**Correct**
Q8	Operation Z Findings	Hallucinated / Unknown	"Alien probe 'The Watcher'"	**Correct**
Q9	**Mars Water Location**	Hallucinated	**"Context unclear about location"**	*Partial Failure*
Q10	**Operation Z Leader**	Hallucinated	**"Unknown"**	*Retrieval Limit*

Submitted By: Kamran Wahab [2025(S)-MS-AI-03]

4. Deep Analysis of RAG Performance

The RAG system achieved **80% accuracy** (8/10 correct). The system successfully grounded the AI, preventing hallucinations for 8 questions. However, the analysis of the two "failures" provides critical insight into RAG architecture:

- The Success (Grounding):

For questions like "What is the currency?", LLaMA alone would guess "Dollars" or "Credits." With RAG, it retrieved the exact sentence defining "Red Credits" and answered with 100% precision.

- The Failure (Retrieval Granularity):

For Question 10 ("Who led the team in Operation Z?"), the system failed to answer.

  - **Reason:** The retrieval logic was based on sentence-level similarity.
  - **Sentence A:** *"Operation Z was a secret mission..."* (High similarity to query "Operation Z").
  - **Sentence B:** *"The team was led by Commander Sarah O'Neil."* (Low similarity to query because it lacks the keyword "Operation Z").
  - **Outcome:** The system retrieved Sentence A, which did not contain the Commander's name. LLaMA correctly stated "Unknown" based on the provided context.
  - **Correction Strategy:** A production RAG system should use **Context Windowing** (retrieving the top match *plus* the surrounding sentences) to capture this missing information.

5. Overall Conclusion

This assignment demonstrated the progression from training simple models to utilizing state-of-the-art LLMs. While the RNN (Task 1) requires significant data to generalize, LLaMA (Task 2) offers powerful out-of-the-box performance. Task 3 proved that while LLMs are knowledgeable, they require RAG systems to access private or specific data. The implementation highlighted that **retrieval quality is just as important as the model quality**—if the retrieval step misses the context (as seen in Q10), even the most powerful LLM cannot answer correctly.

Submitted By: Kamran Wahab [2025(S)-MS-AI-03]