AI Community Assignment

May 7, 2025

General Guidelines for Submission

- The assignment consists of **Technical** (3 problems) and **Non-Technical** (2 problems).
- **Deadline:** 28th May 2025, 11:55 PM IST. Late submissions incur heavy penalty.
- Join the WhatsApp group for queries; all further communications will be via that.
- **All problems are compulsory** (bonus parts optional for extra credit).
- Submit a single public GitHub repository named `<your_roll>_AIC`. You must be the sole contributor.
- Technical code must be in Jupyter notebooks (`.ipynb`), with each code cell accompanied by markdown explanations and visible outputs.
- Include a comprehensive `README.md`:
  - Setup steps for local execution.
  - Detailed documentation of analysis, experiments, and final comments.
  - References to any external resources used.
  - Notes on error handling and troubleshooting.
- Non-Technical answers must be in separate markdown files: `NT_Q1.md`, `NT_Q2.md`, respecting word limits.
- If you need any help with learning the appropriate Tech Stack, contact the Managers.
- Start this Assignment as soon as possible as the questions and lengthy and requires a bit of research.
- Try to Complete all the questions but Submit whatever you've done with proper documentation.
- Hint : Use GPU based training using libraries like PyTorch and TensorFlow as The Datasets are huge.

1. Technical Problems

1.1 Q1: Text Classification with BERT & From-Scratch Attention (Bonus)

Task: Build a state-of-the-art text classifier to assign one of 43 labels.

Implementation:

a) Fine-tune a pretrained BERT model on `train.csv` (`Category`, `Text`). Find the train.csv file here.

- Preprocessing: stop-word removal, lemmatization, normalization, augmentation.
- Report: training/validation loss curves; accuracy, precision, recall, F1 Scores.
- Hyperparameter tuning: learning rate, batch size, epochs. Document results.

b) **Bonus:** Implement a Transformer-style classifier *from scratch*:

- Word2Vec embeddings.
- Sinusoidal positional encodings.
- Multi-headed self-attention and Feed-forward classification head.
- Train end-to-end; compare performance and compute cost vs. BERT.

c) **Extra Bonus:** Efficient Attention Exploration
Choose and Implement ONE of the following efficient attention variants:

- Linear Attention
- Sliding Window Attention
- Local-Global Attention
- Random Feature Attention

Compare your efficient implementation with standard attention:

- Mathematical basis for efficiency gains
- Theoretical advantages and limitations
- Implementation complexity

1.2 Q2: Transfer Learning for Fashion-MNIST

Objective: Adapt a pretrained CNN (e.g., ResNet50, VGG16) to classify 28×28 grayscale Fashion-MNIST images into 10 classes.

Implementation:

a) Data pipeline:

- Resize to 224×224.
- Convert 1→3 channels (duplication or learnable adapter).

b) Model:

- Load pretrained backbone without top layers.
- Freeze backbone; add new FC head.
- Train head only; record validation metrics.

c) Fine-tuning:

- Unfreeze selected deeper blocks.
- Fine-tune with lower learning rate.
- Experiment: data augmentation, LR scheduling, dropout, weight decay.

Find the Fashion-MNIST Dataset here

1.3 Q3: Retrieval-Augmented Generation (RAG) over PDF

Goal: Build a RAG chatbot over a chosen PDF using open-source LLMs.

Implementation:

a) **Ingestion & Indexing:**

- Parse PDF; chunk into semantic units.
- Build a FAISS (or similar) vector index.

b) **Retrieval & Generation:**

- Retrieve top-k relevant chunks for a query.
- Generate answers with an open LLM (e.g., Groq API) conditioned on retrieved context.

c) **Advanced Techniques (Bonus):**

- KV-cache for multi-turn speedup.
- Extract entities/relations; build a mini knowledge graph.
- Implement history-aware responses by passing dialogue history to the prompt.

d) **Agentic Architecture (Extra Bonus)**

- Design and implement the following specialized agents:
  - Information Extraction Agent: Extract key entities, facts, and relationships from text
  - Synthesis Agent: Summarize and organize extracted information
  - Query Agent: Handle natural language questions about processed documents
- Create a simple communication protocol between your agents (using JSON)
- Design a basic coordinator that manages the workflow between agents
- Implement error handling for when agents fail or provide incomplete information

2. Non-Technical Problems

2.1 NT Q1: Hackathon Preparation Timeline

Draft a detailed 1-month preparation plan for an AI hackathon, covering:

- Problem selection and quantitative formulation.
- Data sourcing and preprocessing techniques.
- Exploratory Data Analysis.
- Model selection, training strategies, compute requirements.
- Team roles, milestones, deliverables, risk mitigation.

Minimum 500 words

2.2 NT Q2: Statement of Purpose (SOP)

Write an SOP for joining the Artificial Intelligence Community, including:

- Your goals and motivations.
- Relevant background and knowledge.
- How you will contribute and what you hope to learn.

Minimum 200 words

Bonus

Propose ideas for AI products/projects. Describe:

- Problem statement and real-world impact.
- Technical approach and feasibility.
- Roadmap from concept to prototype.

(Completely optional; extra credit for practical, materializable ideas.)

Resources

Fashion-MNIST Dataset:
https://drive.google.com/drive/folders/1qZNwYOW53GZYZjpmsSpZMBNh1PEQumnb?usp=sharing
BERT Text Classification Dataset:
https://drive.google.com/file/d/19o5KeyLL0Hio-OHJyxpUKd$_y mZMSfIjc/view?usp = sharing$
Complete Collection of Files:
https://drive.google.com/drive/folders/1kxA4bKERW8UbZERQCjMpe3C3Rr5eNr4R?usp=sharing
WhatsApp Group:
Join for more information on the Selection Process and Queries
https://chat.whatsapp.com/IWu7Ij8f0G9Li1nTrDAK0T