Part B – Sentiment Analysis System

---

1. Problem Statement

The goal of Part B is to classify incoming customer emails as **positive**, **negative**, or **neutral**.
The assignment requires:

- Running the sentiment system on **10 sample email rows** from the dataset
- Implementing a **weak prompt (Prompt V1)**
- Implementing an **improved prompt (Prompt V2)**
- Comparing their outputs
- All done **locally without API usage**

---

2. Dataset Preparation

From the dataset:

- Combined subject + body into a single text field
- Selected the **first 10 rows** as required
- These form the input to the sentiment model

Example input:

"Unable to access shared mailbox I am getting a..."

---

3. Model Used

A **fully local transformer model** was used:

Model:

cardiffnlp/twitter-roberta-base-sentiment-latest

Reasons for choosing this model:

- Lightweight
- High-quality sentiment detection
- Zero external API usage
- Runs on CPU
- Includes sentiment-specific fine-tuning

---

4. Local Sentiment Pipeline

Steps:

1. Tokenize input email
2. Forward pass through RoBERTa model
3. Apply softmax
4. Return:
    - sentiment
    - confidence
    - raw scores

Example output:

{
"sentiment": "negative",
"confidence": 0.83,
"scores": {...}
}

---

5. Prompt V1 (Weak Prompt Simulation)

Prompt V1 mimics an **unstructured prompt**:

- No evidence extraction
- No reasoning
- Simply returns the model output
- This simulates how a naive LLM prompt would behave

Example:

Sentiment: negative

Confidence: 0.76

Reasoning: "Basic model output without evidence."

---

6. Prompt V2 (Improved Prompt Simulation)

Prompt V2 includes:

Evidence extraction

Keywords like:

- "unable"

- "not"
- "slow"
- "error"
- "thanks"
- "resolved"

Confidence calibration

- Low confidence → scaled down
- Medium confidence → slightly reduced
- High confidence → unchanged

Natural language reasoning

Explains *why* a sentiment was chosen.

Example:

Sentiment: negative

Confidence: 0.846

Evidence: ["unable"]

Reasoning: "Detected negative sentiment based on keywords: ['unable']"

---

7. Results Comparison

A final table compares:

- Original email
- V1 sentiment + confidence
- V2 sentiment + improved confidence
- V2 evidence
- V2 reasoning

This is included in screenshots.

Observation:

- V2 is more explainable
- V2 confidence is more stable
- V2 handles ambiguous/neutral cases better

---

8. Error Analysis

1. Very short emails

One-word replies sometimes become "neutral" by default.

2. Keyword bias

Models may classify emails with “not” as negative even if context is neutral.

3. Domain shift

Model is trained on Twitter, not support emails — reasons for misclassifications.

4. Evidence extraction limits

Some negative phrases (e.g., “this is bad”) are not in the keyword list.

---

9. Production Improvements

1. Fine-tune model on Hiver-specific emails

Better domain adaptation.

2. Domain-aware sentiment rules

Custom lexicons for:

- tagging issues
- workflow issues
- mobile issues

3. Add contextual memory

Improve predictions using previous message history.