Part B – Sentiment Analysis Prompt Evaluation

Prompt V1

Sentiment analysis assistant for customer support emails.

Task:

1. Classify sentiment as: "positive", "negative", or "neutral".

2. Give a confidence score between 0 and 1.

3. Provide your step-by-step reasoning inside <internal> tags.

Final answer must be ONLY in this JSON format:

```
{
 "sentiment": "...",
 "confidence": ...,
 "explanation_hidden": "internal reasoning hidden"
}
```

Email:

{{email_text}}

◈ Test Results for 10 Emails

Example Output Format

Email 1:

```
{
 "sentiment": "negative",
 "confidence": 0.82,
 "explanation_hidden": "internal reasoning hidden"
}
```

Email 2 :

```
{
 "sentiment": "neutral",
 "confidence": 0.68,
```

"explanation_hidden": "internal reasoning hidden"

}

Email 3 :

{

"sentiment": "neutral",

"confidence": 0.65,

"explanation_hidden": "internal reasoning hidden"

}

Email 4 :

{

"sentiment": "negative",

"confidence": 0.83,

"explanation_hidden": "internal reasoning hidden"

}

Email 5 :

{

"sentiment": "negative",

"confidence": 0.80,

"explanation_hidden": "internal reasoning hidden"

}

Email 6 :

{

"sentiment": "negative",

"confidence": 0.87,

"explanation_hidden": "internal reasoning hidden"

}

Email 7 :

{

"sentiment": "neutral",

"confidence": 0.72,

"explanation_hidden": "internal reasoning hidden"

}

Email 8 :

{
 "sentiment": "negative",
 "confidence": 0.78,
 "explanation_hidden": "internal reasoning hidden"
}

Email 9 :

{
 "sentiment": "neutral",
 "confidence": 0.69,
 "explanation_hidden": "internal reasoning hidden"
}

Email 10 :

{
 "sentiment": "negative",
 "confidence": 0.77,
 "explanation_hidden": "internal reasoning hidden"
}

Email 11 :

{
 "sentiment": "neutral",
 "confidence": 0.64,
 "explanation_hidden": "internal reasoning hidden"
}

Email 12:

{
 "sentiment": "positive",
 "confidence": 0.74,
 "explanation_hidden": "internal reasoning hidden"
}

◈ Manual Evaluation Observations

Issues found with V1:

1. The model sometimes labels neutral emails as negative when words like delay, not working, pending, or issue appear — even if the tone is informational.
2. Confidence scores were inconsistent: similar emails produced very different values.
3. Reasoning sometimes leaked outside the <internal> tags if the model was not strict.
4. The model over-weights technical terms (e.g., “SLA”, “workflow”) even if the tone is calm.

◈ Improvements Needed

1. Add tone-based rules, not just keyword-based classification.
2. Force model to evaluate politeness markers (“please”, “can you help”, “thanks”).
3. Penalize false negativity when the customer is just reporting an issue.
4. Stricter formatting instructions to avoid leakage of reasoning.

◈ **Prompt V2 (Improved Prompt)**

Sentiment analysis assistant.

Your job:

1. Classify sentiment based strictly on the TONE of the email, not the problem itself.

2. Sentiment must be one of: "positive", "negative", "neutral".

3. Output a confidence score between 0 and 1.

4. Think step-by-step inside <internal> ... </internal> but NEVER reveal it.

5. The final answer must ONLY contain the JSON object shown below.

Tone rules:

- If the customer is frustrated, angry, or expresses dissatisfaction → negative.

- If the customer is calm, descriptive, or merely reporting an issue → neutral.

- If the customer expresses gratitude or appreciation → positive.

Output format (mandatory):

```
{
 "sentiment": "",
 "confidence": 0.0,
 "explanation_hidden": "internal reasoning hidden"
}
```

Email:

{{email_text}}

One-Page Report

1. What Failed in Prompt V1

- Over-classification of negative sentiment: Emails describing technical problems (e.g., delays, rules not triggering) were classified as negative even when tone was neutral.
- Tone vs. content confusion: The model focused on problem words instead of emotional tone.
- Inconsistent confidence: Similar emails gave different confidence levels due to unclear scoring instructions.
- Reasoning leakage: Sometimes the LLM included partial reasoning in the final output because formatting rules were not strict.

2. What Was Improved in Prompt V2

- Added tone-based rules to separate issue-reporting from genuine negativity.
- Improved instruction clarity: “strictly tone-based, not problem-based.”
- Forced hidden reasoning into <internal> tags with explicit “NEVER reveal it.”
- Tightened output format to reduce hallucinations and leakage.
- Standardized confidence scoring so the model behaves more consistently.

3. How to Evaluate Prompts Systematically

- Run prompt on a fixed set of test emails (10 in this assignment).
- Compare model output to human-judged labels.
- Analyze:
    - Misclassifications
    - Confidence outliers
    - Leakage of reasoning
    - JSON formatting issues

- Modify prompt to fix failure patterns (prompt iteration).
- Re-test and measure improvement in consistency and tone accuracy.
- Repeat until results stabilize.