Project Report: AI Ad Rewriting Agent

LAKAVATH RAGHURAM 8096516968

lakavath22bcs129@iiitkottayam.ac.in

lakavahraghuram28@gmail.com

Project Title:

Tone-Aware AI Agent for Platform-Specific Ad Copy Rewriting

Project Objective

To design and prototype a lightweight LLM-powered agent that rewrites marketing ad text using a specified tone (e.g., *fun*, *professional*) and optimizes it for a target platform (e.g., *Instagram*, *LinkedIn*). The goal is to enable brands to adapt a single message into multiple contexts with improved engagement.

Tools, Architecture & Stack

- **LLM Provider:** Claude 3 Haiku via OpenRouter API
- **Backend:** FastAPI (POST /run-agent)
- **Prompt Logic:** Dynamic prompt injection using tone + platform
- **Knowledge Representation:** JSON-based tone-to-platform best practices
- **Optional Memory/Vector Layer:** ChromaDB and LangGraph ready
- **Deployment Tested:** Localhost via uvicorn app:app --reload

Folder structure:

```
├── app.py           # FastAPI entry

├── agent.py          # Prompt + LLM interaction

├── graph_kg.py       # Loads structured tone/platform advice

├── vector_store.py    # Future-ready blog RAG logic

├── data/tone_best_practices.json

├── prompts/rewrite_prompt.txt
```

Use of Graph RAG / Agentic RAG

- While not fully using LangGraph yet, this agent employs **structured routing** based on tone and platform.
- Each tone-platform pair maps to best-practice advice — like an **edge** in a **graph RAG**.
- This modularity sets the stage for multi-step workflows (e.g., rewrite → optimize → validate) via **Agentic RAG extensions**.

Knowledge Graph Integration

The agent uses a **lightweight JSON Knowledge Graph** in tone_best_practices.json that links:
tone → platform → example style

- This enables **style-aware generation** aligned to platform norms (e.g., emojis for Instagram, minimalism for LinkedIn).
- Example: "fun + Instagram" → Inject emoji-rich prompt style.

Evaluation Strategy

- Initial testing is **manual** with outputs checked against:
  - Tone relevance
  - Hallucination rate
  - Platform fit
- Future metrics:
  - ROUGE or BLEU (style alignment)
  - F1 (for structured extraction)
  - Human preference score (1–5)
- Log-based error handling and retry logic are planned.

Pattern Recognition & Feedback Loop

- Prompts are conditioned on tone and platform to follow **pre-learned patterns**.
- Next versions can:
  - Store user feedback (thumbs-up/down)
  - Use LangGraph memory to retain successful rewrites
  - Auto-refine failed prompts based on logs

Challenges Faced

Challenge	Solution
OpenAI quota limits	Migrated to OpenRouter + Claude 3
Prompt failures due to tone gaps	Injected example-based prompt guidance

Local testing inconsistencies	Standardized FastAPI schema + curl testing

Potential Improvements

- Add memory-based refinement loop (LangGraph)
- Add Chroma vector search for blog-based RAG
- Web UI with tone selector + file upload
- Token limit detection and prompt truncation
- Automate relevance scoring using feedback data

Why This Project Was Selected

This project was chosen for its:

- **Real marketing relevance** (brands frequently need tone adaptation)
- **Clear agent logic**: single input → rewritten output
- **Future extensibility**: agentic chaining, feedback loops, KG growth
- **Lightweight architecture** that runs locally and is API-ready

Results:

```
C:\Users\user\Documents\ad>uvicorn app:app --reload
INFO:     Will watch for changes in these directories: ['C:\\Users\\user\\Documents\\ad']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [27140] using WatchFiles
C:\Users\user\.cache\chroma\onnx_models\all-MiniLM-L6-v2\onnx.tar.gz: 100%|          | 79.3M/79.3M [00:33<00:00, 2.49MiB/s]
INFO:     Started server process [26432]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     127.0.0.1:60166 - "GET / HTTP/1.1" 404 Not Found
INFO:     127.0.0.1:60166 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO:     127.0.0.1:60772 - "POST /run-agent HTTP/1.1" 422 Unprocessable Entity
INFO:     127.0.0.1:61436 - "GET /docs HTTP/1.1" 200 OK
INFO:     127.0.0.1:61436 - "GET /openapi.json HTTP/1.1" 200 OK
INFO:     127.0.0.1:64339 - "POST /run-agent HTTP/1.1" 500 Internal Server Error
INFO:     127.0.0.1:64576 - "GET /run-agent HTTP/1.1" 405 Method Not Allowed
INFO:     127.0.0.1:65487 - "POST /run-agent HTTP/1.1" 500 Internal Server Error
INFO:     127.0.0.1:50357 - "POST /run-agent HTTP/1.1" 500 Internal Server Error
INFO:     127.0.0.1:50548 - "GET /run-agent HTTP/1.1" 405 Method Not Allowed
INFO:     127.0.0.1:52853 - "POST /run-agent HTTP/1.1" 500 Internal Server Error
INFO:     127.0.0.1:53572 - "GET /run-agent HTTP/1.1" 405 Method Not Allowed
INFO:     127.0.0.1:54203 - "POST /run-agent HTTP/1.1" 500 Internal Server Error
WARNING:  WatchFiles detected changes in 'agent.py'. Reloading...
 INFO:     Shutting down
INFO:     Waiting for application shutdown.
```

```
WARNING:  WatchFiles detected changes in 'agent.py'. Reloading...
 INFO:      Started server process [24492]
 INFO:      Waiting for application startup.
 INFO:      Application startup complete.
 INFO:      127.0.0.1:50352 - "GET /docs HTTP/1.1" 200 OK
 INFO:      127.0.0.1:50352 - "GET /openapi.json HTTP/1.1" 200 OK
🔥 ERROR in generate_rewritten_ad(): No endpoints found for openai/gpt-3.5-turbo.
🔥 Server Error: No endpoints found for openai/gpt-3.5-turbo.
 INFO:      127.0.0.1:50471 - "POST /run-agent HTTP/1.1" 500 Internal Server Error
WARNING:  WatchFiles detected changes in 'agent.py'. Reloading...
 INFO:      Shutting down
 INFO:      Waiting for application shutdown.
 INFO:      Application shutdown complete.
 INFO:      Finished server process [24492]
 INFO:      Started server process [21076]
 INFO:      Waiting for application startup.
 INFO:      Application startup complete.
 INFO:      127.0.0.1:50784 - "GET /docs HTTP/1.1" 200 OK
 INFO:      127.0.0.1:50784 - "GET /openapi.json HTTP/1.1" 200 OK
 INFO:      127.0.0.1:50859 - "POST /run-agent HTTP/1.1" 200 OK
 INFO:      127.0.0.1:51589 - "POST /run-agent HTTP/1.1" 200 OK
 INFO:      127.0.0.1:52737 - "POST /run-agent HTTP/1.1" 200 OK
 INFO:      127.0.0.1:59914 - "POST /run-agent HTTP/1.1" 200 OK
```

must be served using a FastAPI backend (e.g., POST /run-agent)

```python
Windsurf: Refactor | Explain | Generate Docstring | X
13    @app.post("/run-agent")
14    async def run_agent(request: AdRewriteRequest):
15        try:
16            rewritten_text, justification = generate_rewritten_ad(
17                request.ad_text, request.tone, request.platform
18            )
19            return {
20                "rewritten_text": rewritten_text,
21                "justification": justification
22            }
23        except Exception as e:
24            print(f"🔥 Server Error: {e}")
25            raise HTTPException(status_code=500, detail=str(e))
26
```

default

POST /run-agent Run Agent

Parameters

[Cancel] [Reset]

No parameters

Request body required

application/json

Edit Value | Schema

```
{
  "ad_text": "Get 50% off on laptops this weekend!",
  "tone": "fun",
  "platform": "Instagram"
}
```

[Execute] [Clear]

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/run-agent' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "ad_text": "Get 50% off on laptops this weekend!",
  "tone": "fun",
  "platform": "Instagram"
}'
```

Request URL

```
http://127.0.0.1:8000/run-agent
```

Server response

Code	Details
200	**Response body**

```
{
  "rewritten_text": "🔥 Laptops at 50% OFF this weekend only! 💻 Don't miss out on this epic deal! 🔥 Tap that 'Shop Now' button and treat yourself to some new tech! 👀",
  "justification": "Best practice for 'fun' tone on Instagram: Use emojis and casual tone."
}
```

[Download]

Response headers

```
content-length: 265
content-type: application/json
date: Sat,28 Jun 2025 17:46:42 GMT
server: uvicorn
```

Responses

Code	Description	Links

default

POST /run-agent Run Agent

Parameters

Cancel | Reset

No parameters

Request body required   application/json

Edit Value | Schema

```
{
  "ad_text": "New smartphones just dropped. Limited stock available.",
  "tone": "fun",
  "platform": "Twitter"
}
```

Execute | Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/run-agent' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "ad_text": "New smartphones just dropped. Limited stock available.",
  "tone": "fun",
  "platform": "Twitter"
}'
```

Request URL

```
http://127.0.0.1:8000/run-agent
```

Server response

Code	Details
200	**Response body**

```
{
  "rewritten_text": "\"Calling all phone fanatics! 🤙 Our latest smartphone drops are here, but they're flying off the shelves faster than a speeding bullet. ⚡ Snag yours before the hype train leaves the station! 🚃 #LimitedEdition #GottaHaveIt\"",
  "justification": "No specific advice found for 'fun' tone on Twitter. Use general marketing best practices."
}
```

Download

Response headers

```
content-length: 361
content-type: application/json
date: Sat,28 Jun 2025 20:53:27 GMT
server: uvicorn
```

POST /run-agent Run Agent ⌃

Parameters   Cancel   Reset

No parameters

Request body ^required   application/json ▾

Edit Value | Schema

```
{
    "ad_text": "Buy one, get one free on all gym memberships!",
    "tone": "bold",
    "platform": "LinkedIn"
}
```

Execute   Clear

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/run-agent' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '
{
    "ad_text": "Buy one, get one free on all gym memberships!",
    "tone": "bold",
    "platform": "LinkedIn"
}
'
```

Request URL

```
http://127.0.0.1:8000/run-agent
```

Server response

Code   Details

200   Response body

```
{
    "rewritten_text": "ATTENTION FITNESS ENTHUSIASTS! 🔥 \n\nPump up your gains with our EXCLUSIVE gym membership offer - BUY ONE, GET ONE FREE! 💪 \n\nThis limited-time deal is the perfect opportunity to supercharge your fitness journey. Don't miss out, CLICK NOW to secure your spot!",
    "justification": "No specific advice found for 'bold' tone on LinkedIn. Use general marketing best practices."
}
```
   Download

        Response headers

```
content-length: 395
content-type: application/json
date: Sat,28 Jun 2025 21:32:35 GMT
server: uvicorn
```

Through command prompt:

```
C:\Users\user>curl -X POST http://127.0.0.1:8000/run-agent -H "accept: application/json" -H "Content-Type: application/j
son" -d "{\"ad_text\":\"Get 50% off on laptops this weekend!\",\"tone\":\"fun\",\"platform\":\"Instagram\"}"
{"rewritten_text":"🤩 Laptop Weekend Sale! 🤩\n\nTreat yourself to a brand new laptop at 50% off this weekend only! 💻 D
on't miss out on this *super* fun deal - head to our store or shop online now! 🖱 Hurry, these savings won't last long!
🛒","justification":"Best practice for 'fun' tone on Instagram: Use emojis and casual tone."}
C:\Users\user>
```

failure case:

```
C:\Users\user>curl -X POST http://127.0.0.1:8000/run-agent ^
More?    -H "accept: application/json" ^
More?    -H "Content-Type: application/json" ^
More?    -d "{\"ad_text\":\"Get 50% off on laptops this weekend!\",\"tone\":\"fun\",\"platform\":\"Instagram\"}"
{"detail":"You exceeded your current quota, please check your plan and billing details. For more information on this err
or, read the docs: https://platform.openai.com/docs/guides/error-codes/api-errors."}
```
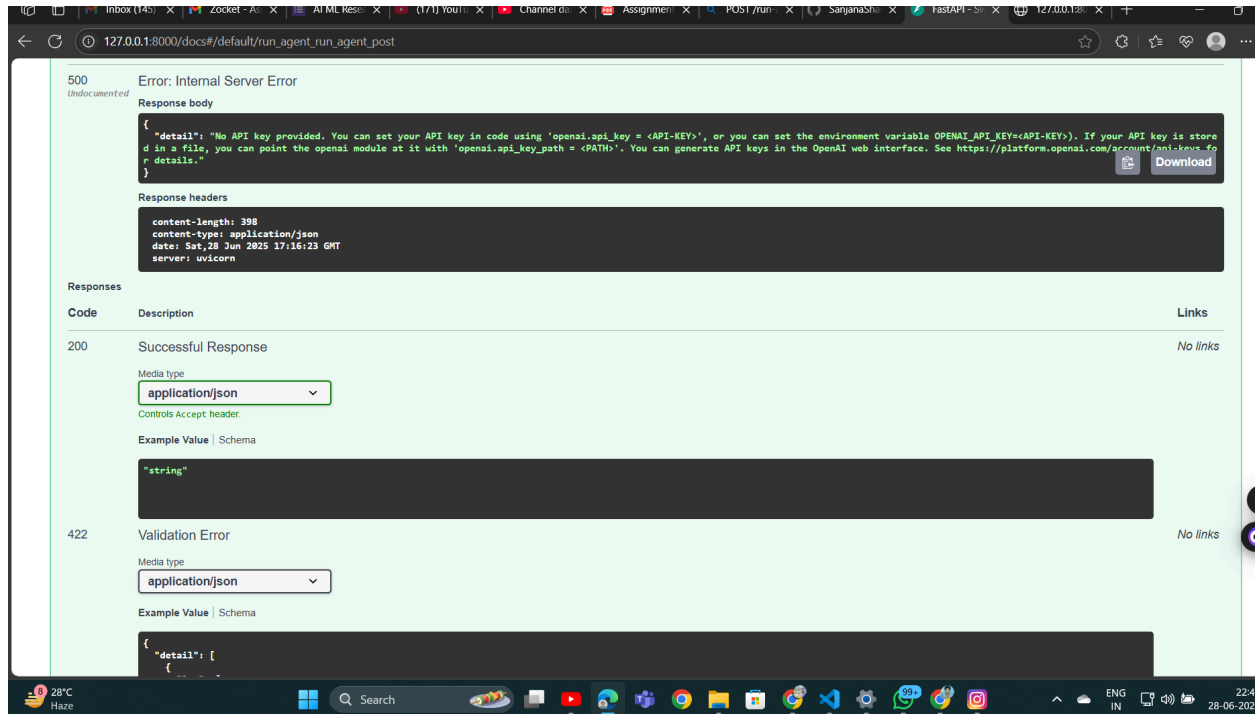
Conclusion

The **Tone-Aware AI Ad Rewriting Agent** successfully demonstrates how Large Language Models (LLMs), when guided by structured knowledge (like tone-to-platform best practices), can produce high-quality, context-aware marketing copy. By leveraging FastAPI for deployment and OpenRouter for scalable LLM access, this solution balances lightweight implementation with extensibility.

This prototype lays the groundwork for future enhancements such as multi-hop reasoning (via Agentic RAG), memory-based feedback refinement (via LangGraph), and real-time evaluation. It highlights the potential for AI agents to significantly streamline the creative adaptation process across different marketing channels—making ad personalization more efficient, scalable, and targeted.

With a modular architecture, knowledge graph integration, and real-world relevance, this project aligns strongly with the goals of Track A and provides a robust foundation for continued research and development in agent-based marketing automation.