LLM-powered Data Extraction

A GPT-powered way to process data in 2025

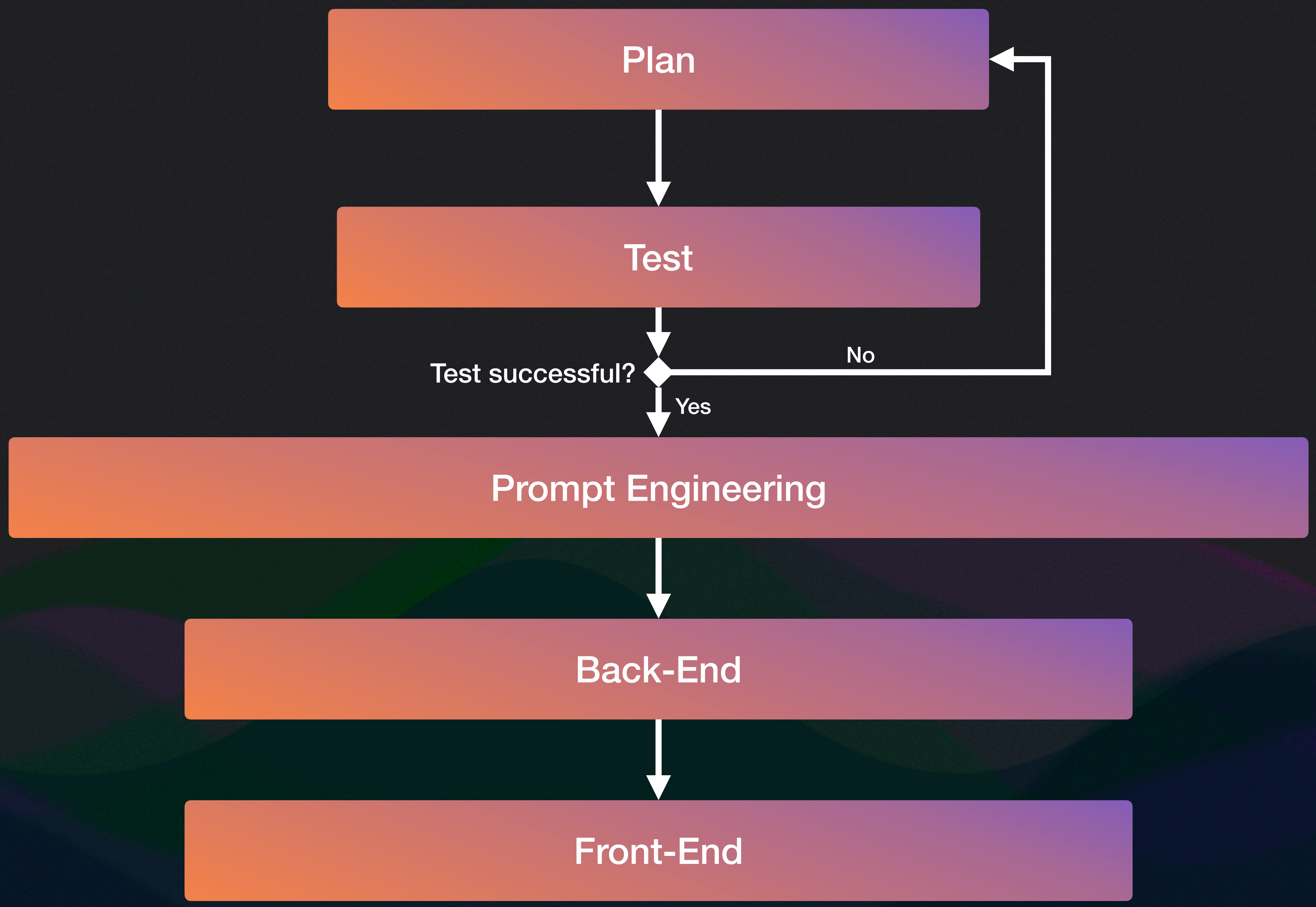
Thomas Laner

Goal

Automated end-to-end system that **extracts** key Red Bull-related **entities** (athletes, teams, disciplines, and events) from **web articles** and **generates tags** from **multimedia content**, empowering data-driven marketing and media impact assessments.

Approach

Project Process



4 driving factors behind the solution

Adaptability
Scale & Model innovations

Performance
How to measure accuracy

Cost
Scale vs. Performance

UX
Wrap Solution for ease-of-use

Resulting Architecture

Adaptability

Performance

Cost

UX

Tech Stack

Pre- & post-processing: Python
LLM-Interaction: OpenAI's Responses API
Front-end: Streamlit

Model

GPT-4 Family
[0,n] reruns to leverage model randomness

Other possible approaches?

- Local models: not as scalable & limited performance.

Why OpenAI's Responses API?

- Optimized for fast, structured, one-shot extractions.

How is prompt quality ensured?

- Structured best-practice prompting, interactively constructed using eval. Framework.

How are model updates handled?

- Modular setup for A/B testing and easy model switching.

How are cost and quality controlled?

- Model, temperature, and rerun count are adjustable per use case.

Resulting Architecture

Adaptability

Performance

Cost

UX

Text Preprocessing

Transform JSON into TXT
Clean data (remove noisy data)

Tags

Main Entities
Actions, activities
Setting, environment
Brands, Logos

Why provide .txt instead of .json?

- Reduces tokens

Why provide only text body to model?

- Reduces context and decreases risk of bias induction into model

What's the difference between image quality modes?

- Low: faster, fewer details, 80 tokens per image
- High: slower, more details, #tokens depending on img size

Which other approaches could we have taken for the generation of tags?

- Process images with locally running models
- Use other API's (e.g. AWS)
- Use custom-fine tuned models either locally or cloud-based

Cost Considerations

	GPT-4.1	4.1 nano	4.1 mini	4o mini	GPT-4o
Intelligence	●●●●	●●	●●●	●●	●●●
Speed	⚡⚡⚡	⚡⚡⚡⚡⚡	⚡⚡⚡⚡	⚡⚡⚡⚡	⚡⚡⚡
Input					
Output					
Reasoning tokens					
PRICING (PER 1M TOKENS)					
Input	$2.00	$0.10	$0.40	$0.15	$2.50
Cached Input	$0.50	$0.03	$0.10	$0.08	$1.25
Output	$8.00	$0.40	$1.60	$0.60	$10.00
CONTEXT					
Window	1,047,576	1,047,576	1,047,576	128,000	128,000
Max Output Tokens	32,768	32,768	32,768	16,384	16,384
Knowledge Cutoff	Jun 01, 2024	Jun 01, 2024	Jun 01, 2024	Oct 01, 2023	Oct 01, 2023

Evaluation Framework

Evaluation Framework

How performance is measured and prompts are engineered

How large is each sample?

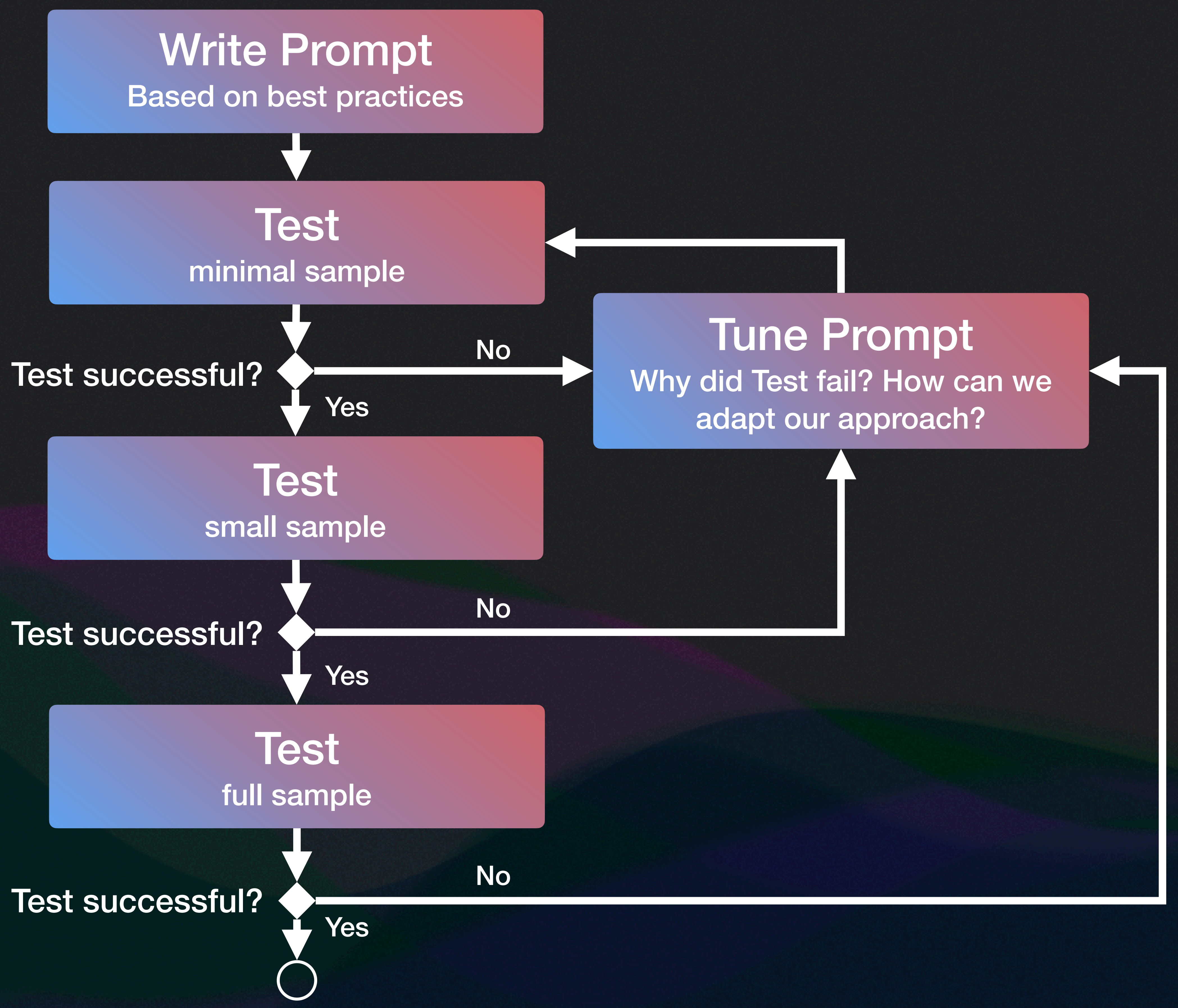
- Minimal sample: 1 file
- Small sample: 5 files
- Full Sample: 500 files

How are models compared?

- Run evaluation for both models and compare results.

When can a test be considered successful?

- **Accuracy > Threshold** based on cost & performance requirements



Evaluation Framework

Test Accuracy

$$\text{Accuracy} = \frac{1}{E \times S} \sum_{e=1}^{E} \sum_{s=1}^{S} \textit{Allocated Points}_{e,s}$$

1 pt: All entities are recognized

0.5 pts: Most important entities are recognized

0 pts: Important entities not recognized

E = # of entities to be tested (3 or 4 depending on tag/entity)

S = Sample size

Prompts

Entity Extraction

Prompt

```
Goal:
Extract from the provided article the following entities:
1. AthletesAndTeams: List all athletes and teams affiliated with Red Bull.
   List any aliases or variations of the team names and correct any spelling
   mistakes. If someone is known by a nickname, use nickname instead of name.
2. Disciplines: Capture every mention of competitive sports & e-sports
   disciplines. Consider both full names and common abbreviations.
3. Events: Identify any formally named tournaments, championships, or events
   (e.g.: "League of Legends World Championship").

Additional Instructions:
– Translate all Discipline- and Event names to English
– Search entire text (including background or historical references) for all
  explicit and implicit references to the above categories.
– Return exactly one JSON object containing the keys "AthletesAndTeams",
  "Disciplines", and "Events". If any of categories not mentioned, provide
  empty array for that key.
– Do only include mentions from the article, not from the instruction.

Output single JSON object with these exact keys, no extra text or different
formatting should be returned:
{
"AthletesAndTeams": [],
"Disciplines": [],
"Events": []
}

Article:
<<<EXTRACTED ARTICLE>>>
```

Consolidation Prompt

```
From web-article extractions below, make sure all entries English,
no duplicates, names spelled correctly. Return single JSON object
with same keys as inputs.

Extractions:
<<<EXTRACTIONS>>>
```

Tag Extraction

Prompt

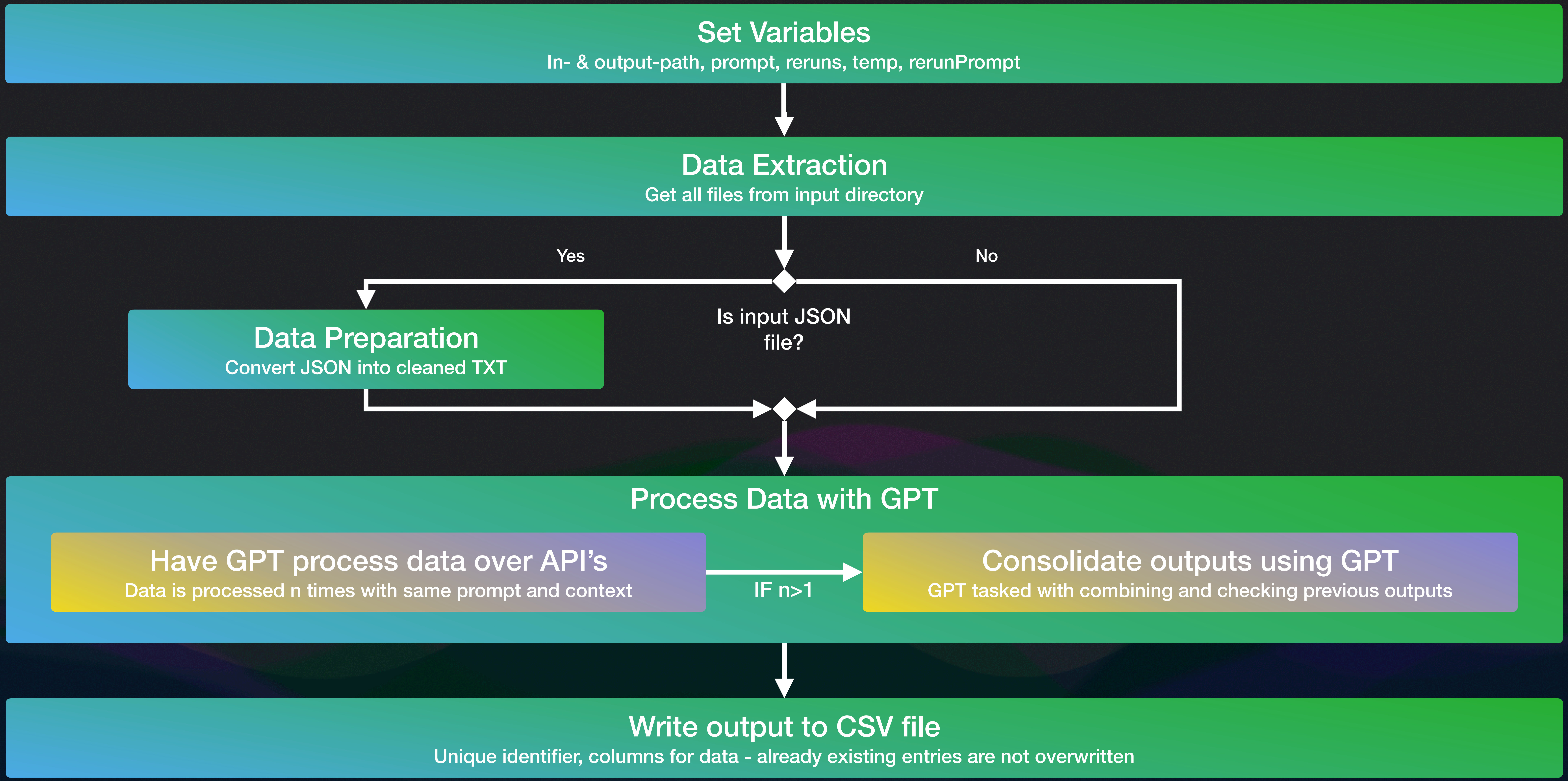
```
Describe these images with a set of tags so that they can then
be used when creating content. Identify:
- Main subjects, objects, people:
    - individuals (names if possible)
    - cars, planes, skis etc. with model, livery, specs
       - Technical components (e.g.: front suspension) - be
precise (propellor airplane, jet plane)
- Depicted Actions, activities
- Setting, environment
- brands, logos, flags

Return only a JSON array of tags with no additional text:
["tag1", "tag2", "tag3"]
```

Consolidation Prompt

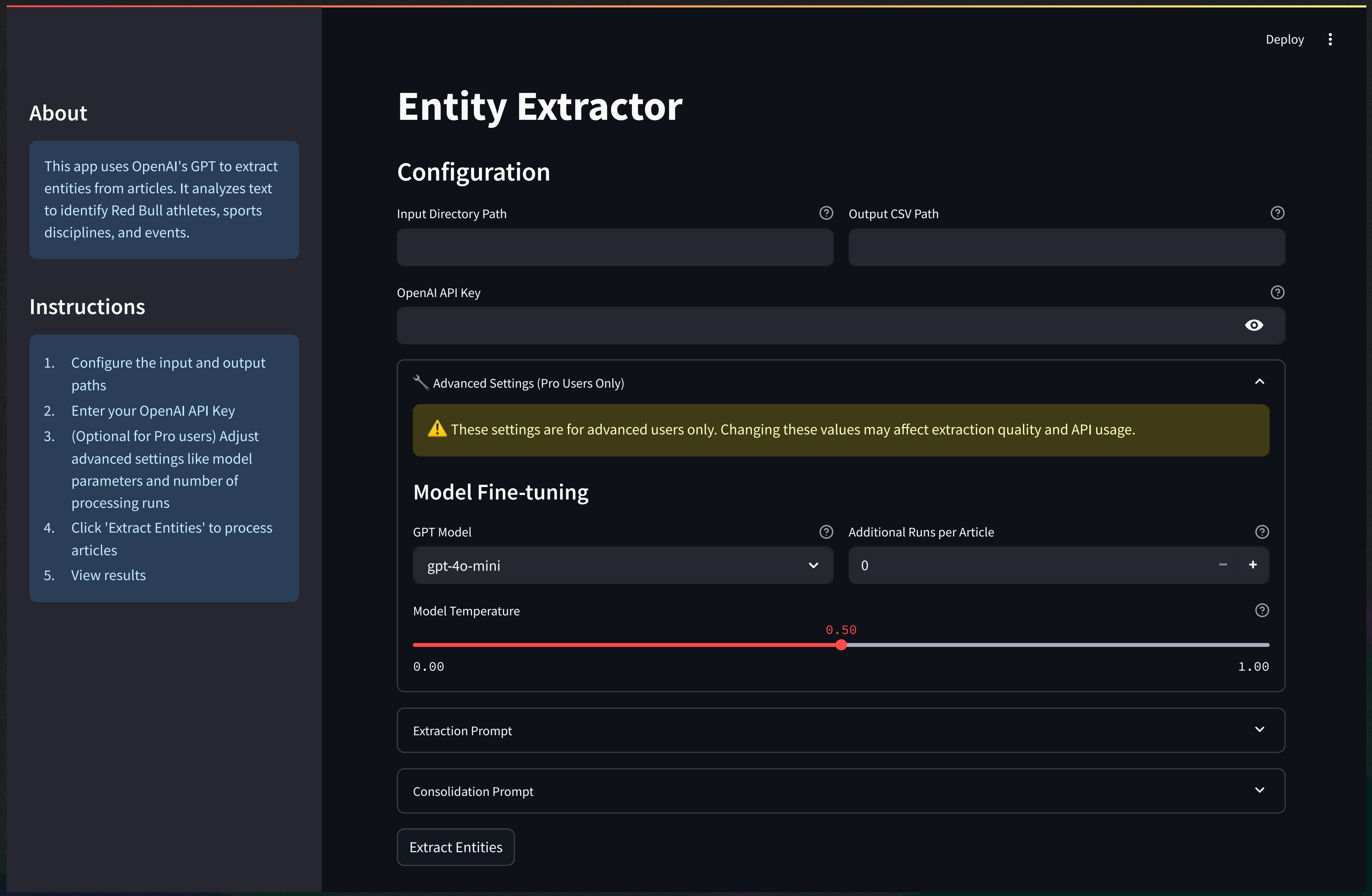
```
Review this image and analyze the provided tags from previous
model runs.
Create a final, consolidated list of accurate tags by:
1. Keeping only tags that actually appear in the image
2. Removing duplicates or near-duplicates
3. Ensuring consistent naming (e.g., choose either 'Formula 1'
   or 'F1', not both)
4. Adding any important missing tags

Return only a JSON array of finalized tags with no additional
text:
["tag1", "tag2", "tag3"]
```

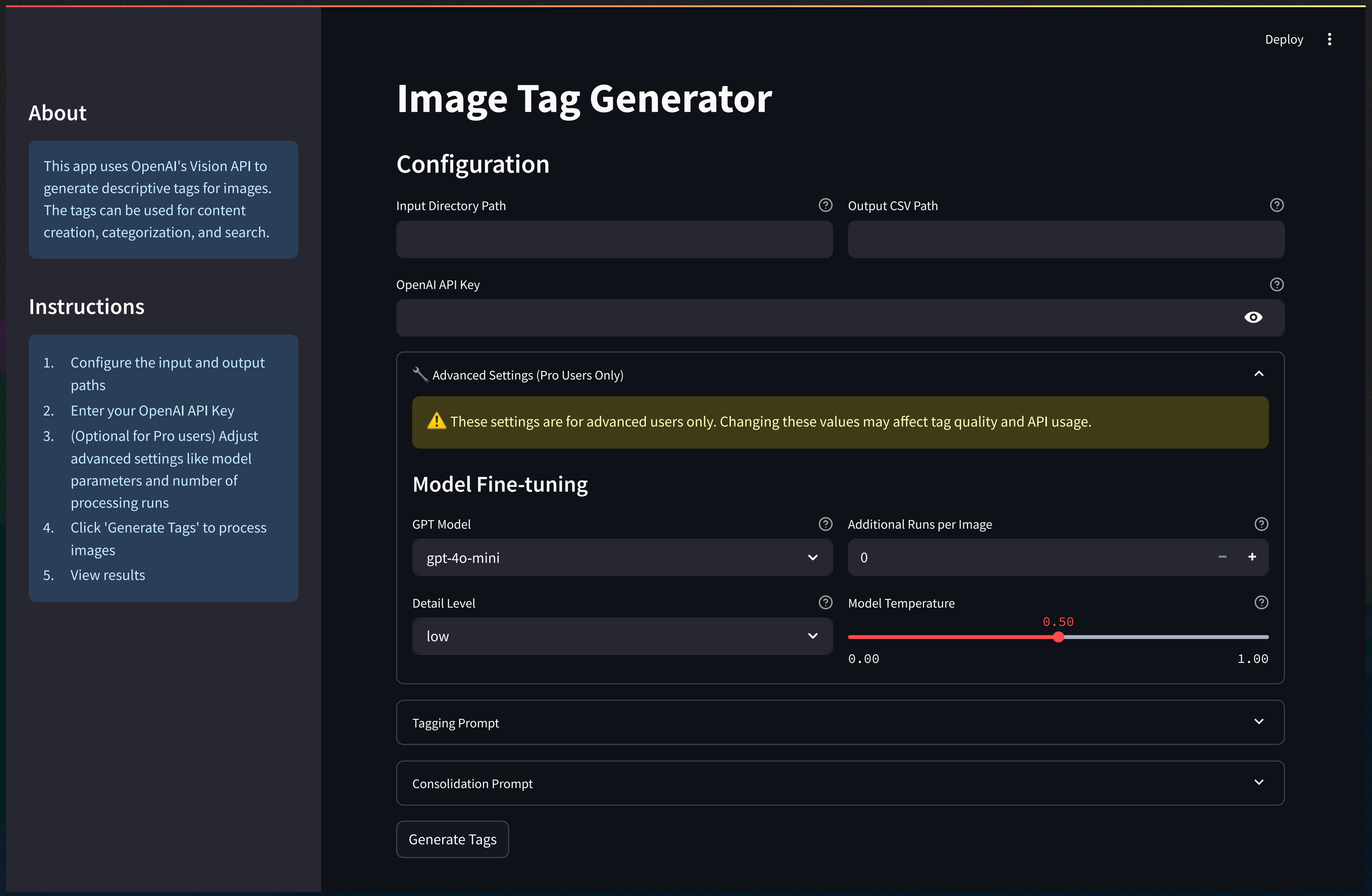
Result

Back-End


Set Variables
In- & output-path, prompt, reruns, temp, rerunPrompt
Data Extraction
Get all files from input directory
Yes
No
Is input JSON file?
Data Preparation
Convert JSON into cleaned TXT
Process Data with GPT
Have GPT process data over API’s
Data is processed n times with same prompt and context
IF n>1
Consolidate outputs using GPT
GPT tasked with combining and checking previous outputs
Write output to CSV file
Unique identifier, columns for data - already existing entries are not overwritten

Front-End


Deploy
About
This app uses OpenAI's GPT to extract entities from articles. It analyzes text to identify Red Bull athletes, sports disciplines, and events.
Instructions
1. Configure the input and output paths
2. Enter your OpenAI API Key
3. (Optional for Pro users) Adjust advanced settings like model parameters and number of processing runs
4. Click 'Extract Entities' to process articles
5. View results
Entity Extractor
Configuration
Input Directory Path
Output CSV Path
OpenAI API Key
Advanced Settings (Pro Users Only)
These settings are for advanced users only. Changing these values may affect extraction quality and API usage.
Model Fine-tuning
GPT Model
gpt-4o-mini
Additional Runs per Article
0
Model Temperature
0.50
0.00
1.00
Extraction Prompt
Consolidation Prompt
Extract Entities


Deploy
About
This app uses OpenAI's Vision API to generate descriptive tags for images. The tags can be used for content creation, categorization, and search.
Instructions
1. Configure the input and output paths
2. Enter your OpenAI API Key
3. (Optional for Pro users) Adjust advanced settings like model parameters and number of processing runs
4. Click 'Generate Tags' to process images
5. View results
Image Tag Generator
Configuration
Input Directory Path
Output CSV Path
OpenAI API Key
Advanced Settings (Pro Users Only)
These settings are for advanced users only. Changing these values may affect tag quality and API usage.
Model Fine-tuning
GPT Model
gpt-4o-mini
Additional Runs per Image
0
Detail Level
low
Model Temperature
0.50
0.00
1.00
Tagging Prompt
Consolidation Prompt
Generate Tags

Further Enhancements

What future versions of the project could incorporate

Confidence Scores

Let model return confidence scores for extracted entities and tags

Semantic Validation

Verify extracted entities & tags against domain-specific knowledge base

Fine-tune

Tune model to align with language & terms commonly used in company to derive better entities & tags

Feedback mechanism

Integrate user feedback as proposals for prompt adjustments