GSfS: An Integrated System for Personalized RSS Digests

Combining RSS Collection with LLM-Based Ranking

DC325951 Cai Mingjie

Github (https://github.com/DavidMJChoi/GSfS)

The Problem: Information Overload

- **Vast Information Streams**: The web generates an overwhelming volume of content daily.
- **User Pain Point**: Manually sifting through hundreds of articles is time-consuming and inefficient.
- **Core Question**: How can we surface the most relevant and interesting content for a user automatically?

Github (https://github.com/DavidMJChoi/GSfS)

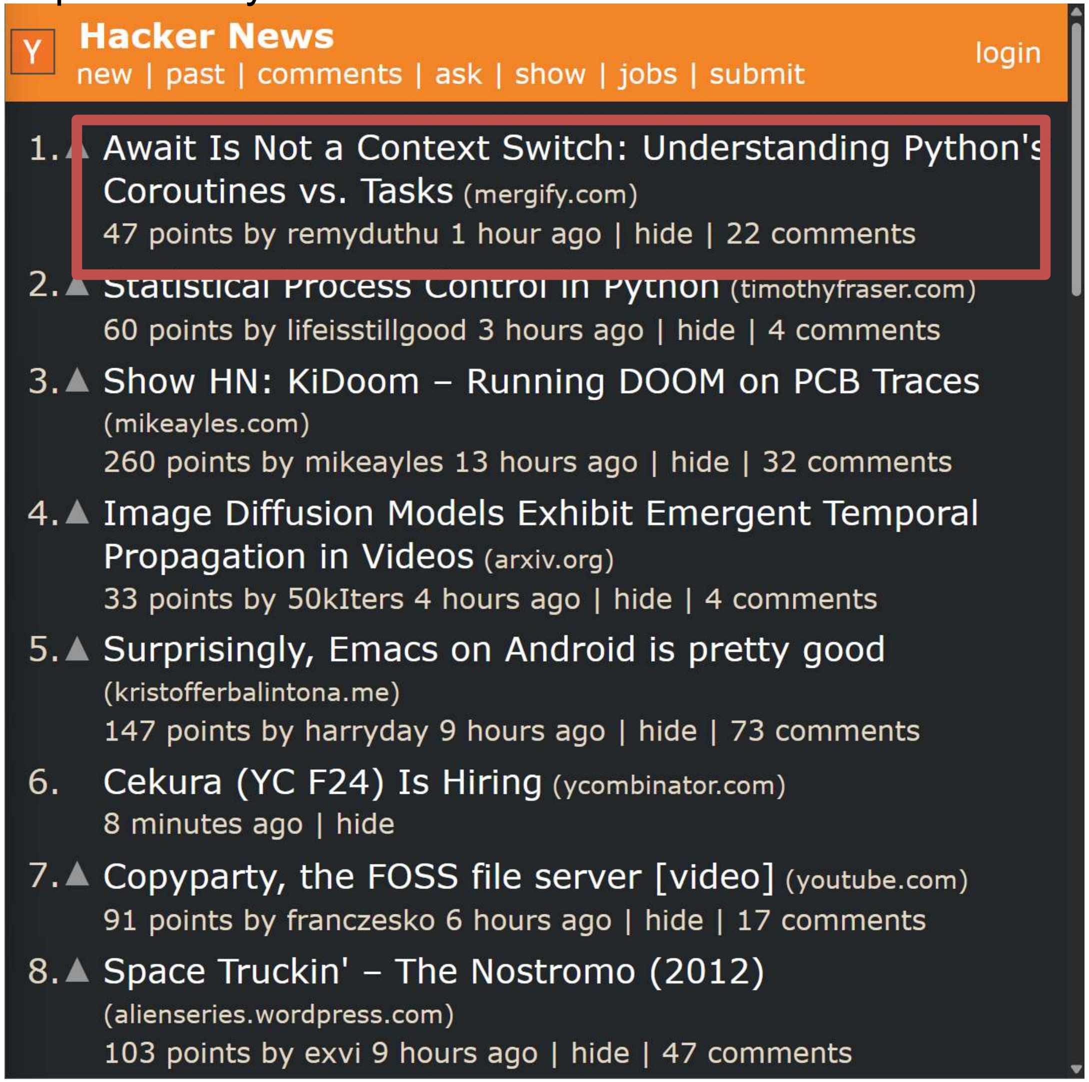
What is RSS?

RSS (Really Simple Syndication) is a web feed standard for publishing frequently updated content.

How it Works:

- Websites generate an RSS feed (an XML file) listing their latest content.
- RSS Readers (like our system) subscribe to these feeds.
- The reader periodically checks the feeds for new .

https://news.ycombinator.com/



Y **Hacker News**
new | past | comments | ask | show | jobs | submit
login

1. ▲ Await Is Not a Context Switch: Understanding Python's Coroutines vs. Tasks (mergify.com)
47 points by remyduthu 1 hour ago | hide | 22 comments

2. ▲ Statistical Process Control in Python (timothyfraser.com)
60 points by lifeisstillgood 3 hours ago | hide | 4 comments

3. ▲ Show HN: KiDoom – Running DOOM on PCB Traces (mikeayles.com)
260 points by mikeayles 13 hours ago | hide | 32 comments

4. ▲ Image Diffusion Models Exhibit Emergent Temporal Propagation in Videos (arxiv.org)
33 points by 50kIters 4 hours ago | hide | 4 comments

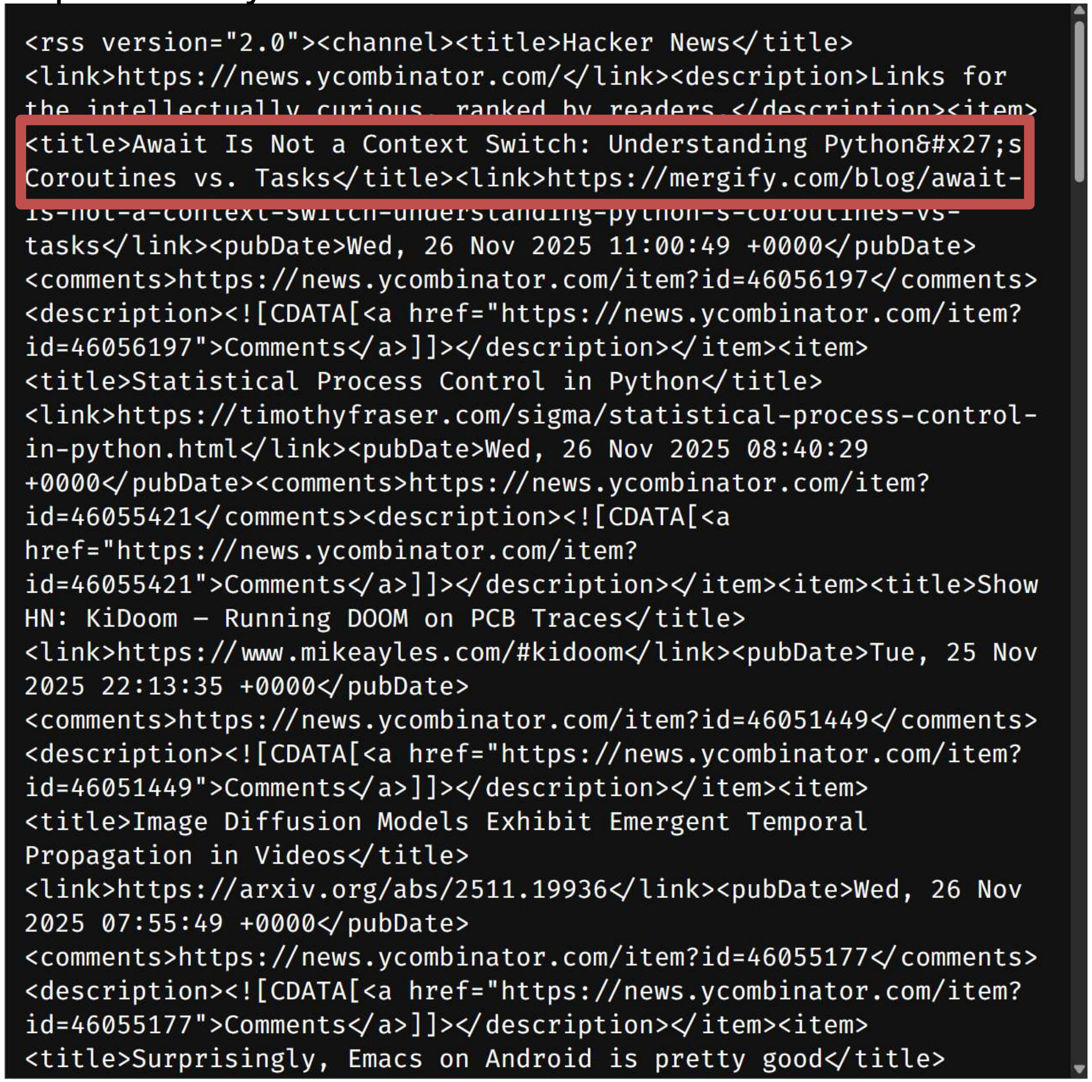
5. ▲ Surprisingly, Emacs on Android is pretty good (kristofferbalintona.me)
147 points by harryday 9 hours ago | hide | 73 comments

6. Cekura (YC F24) Is Hiring (ycombinator.com)
8 minutes ago | hide

7. ▲ Copyparty, the FOSS file server [video] (youtube.com)
91 points by franczesko 6 hours ago | hide | 17 comments

8. ▲ Space Truckin' – The Nostromo (2012) (alienseries.wordpress.com)
103 points by exvi 9 hours ago | hide | 47 comments

https://news.ycombinator.com/rss



```
<rss version="2.0"><channel><title>Hacker News</title>
<link>https://news.ycombinator.com/</link><description>Links for
the intellectually curious, ranked by readers.</description><item>
<title>Await Is Not a Context Switch: Understanding Python&#x27;s
Coroutines vs. Tasks</title><link>https://mergify.com/blog/await-
is-not-a-context-switch-understanding-python-s-coroutines-vs-
tasks</link><pubDate>Wed, 26 Nov 2025 11:00:49 +0000</pubDate>
<comments>https://news.ycombinator.com/item?id=46056197</comments>
<description><![CDATA[<a href="https://news.ycombinator.com/item?
id=46056197">Comments</a>]]></description></item><item>
<title>Statistical Process Control in Python</title>
<link>https://timothyfraser.com/sigma/statistical-process-control-
in-python.html</link><pubDate>Wed, 26 Nov 2025 08:40:29
+0000</pubDate><comments>https://news.ycombinator.com/item?
id=46055421</comments><description><![CDATA[<a
href="https://news.ycombinator.com/item?
id=46055421">Comments</a>]]></description></item><item><title>Show
HN: KiDoom – Running DOOM on PCB Traces</title>
<link>https://www.mikeayles.com/#kidoom</link><pubDate>Tue, 25 Nov
2025 22:13:35 +0000</pubDate>
<comments>https://news.ycombinator.com/item?id=46051449</comments>
<description><![CDATA[<a href="https://news.ycombinator.com/item?
id=46051449">Comments</a>]]></description></item><item>
<title>Image Diffusion Models Exhibit Emergent Temporal
Propagation in Videos</title>
<link>https://arxiv.org/abs/2511.19936</link><pubDate>Wed, 26 Nov
2025 07:55:49 +0000</pubDate>
<comments>https://news.ycombinator.com/item?id=46055177</comments>
<description><![CDATA[<a href="https://news.ycombinator.com/item?
id=46055177">Comments</a>]]></description></item><item>
<title>Surprisingly, Emacs on Android is pretty good</title>
```

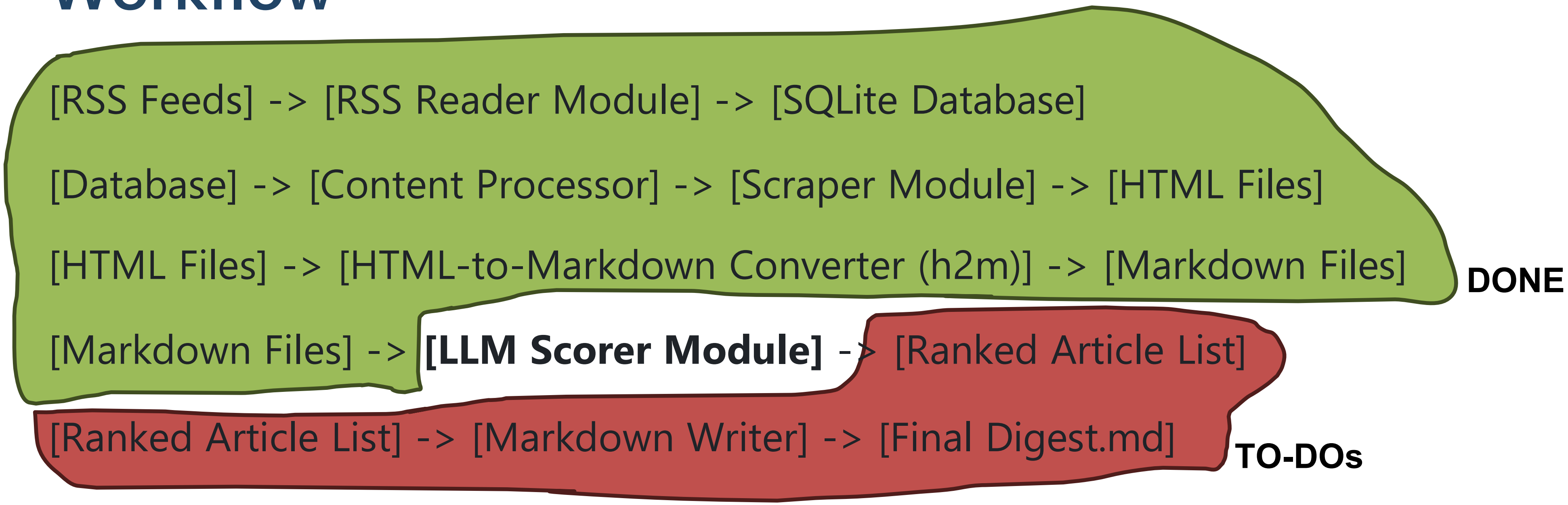
Github (https://github.com/DavidMJChoi/GSfS)

System Overview

- **Aggregate**: **Collect** articles from a configurable list of RSS feeds.
- **Pre-process**: **Clean**, deduplicate, and filter content based on configurable rules.
- **Enrich**: **Scrape** full article text to enable deep content analysis.
- **Intelligently Rank**: Employ an LLM to score and rank articles based on perceived quality and relevance.
- **Deliver**: Generate a concise, personalized Markdown **digest** for the user.

Github (https://github.com/DavidMJChoi/GSfS)

Workflow



[RSS Feeds] -> [RSS Reader Module] -> [SQLite Database]

[Database] -> [Content Processor] -> [Scraper Module] -> [HTML Files]

[HTML Files] -> [HTML-to-Markdown Converter (h2m)] -> [Markdown Files]

DONE

[Markdown Files] -> **[LLM Scorer Module]** -> [Ranked Article List]

[Ranked Article List] -> [Markdown Writer] -> [Final Digest.md]

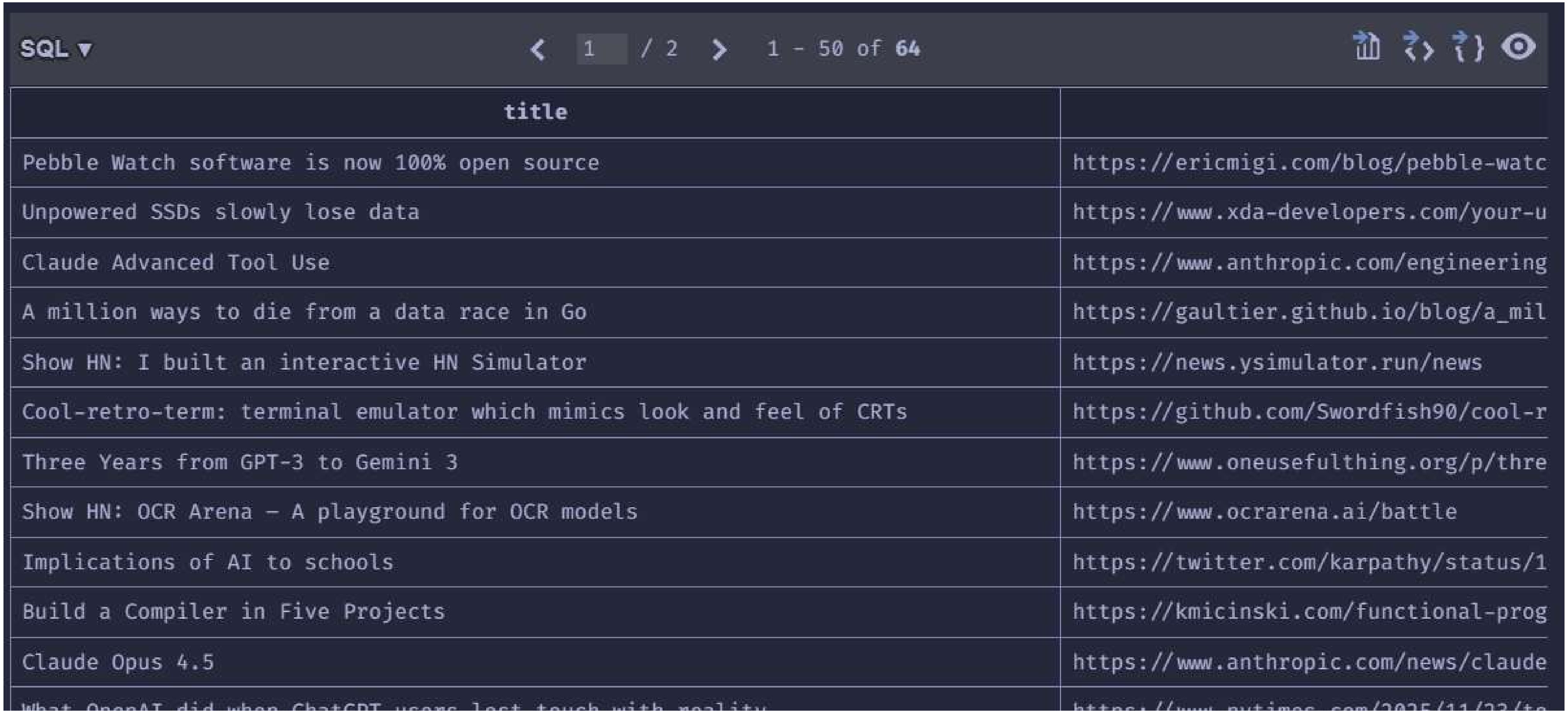
TO-DOs

Github (https://github.com/DavidMJChoi/GSfS)

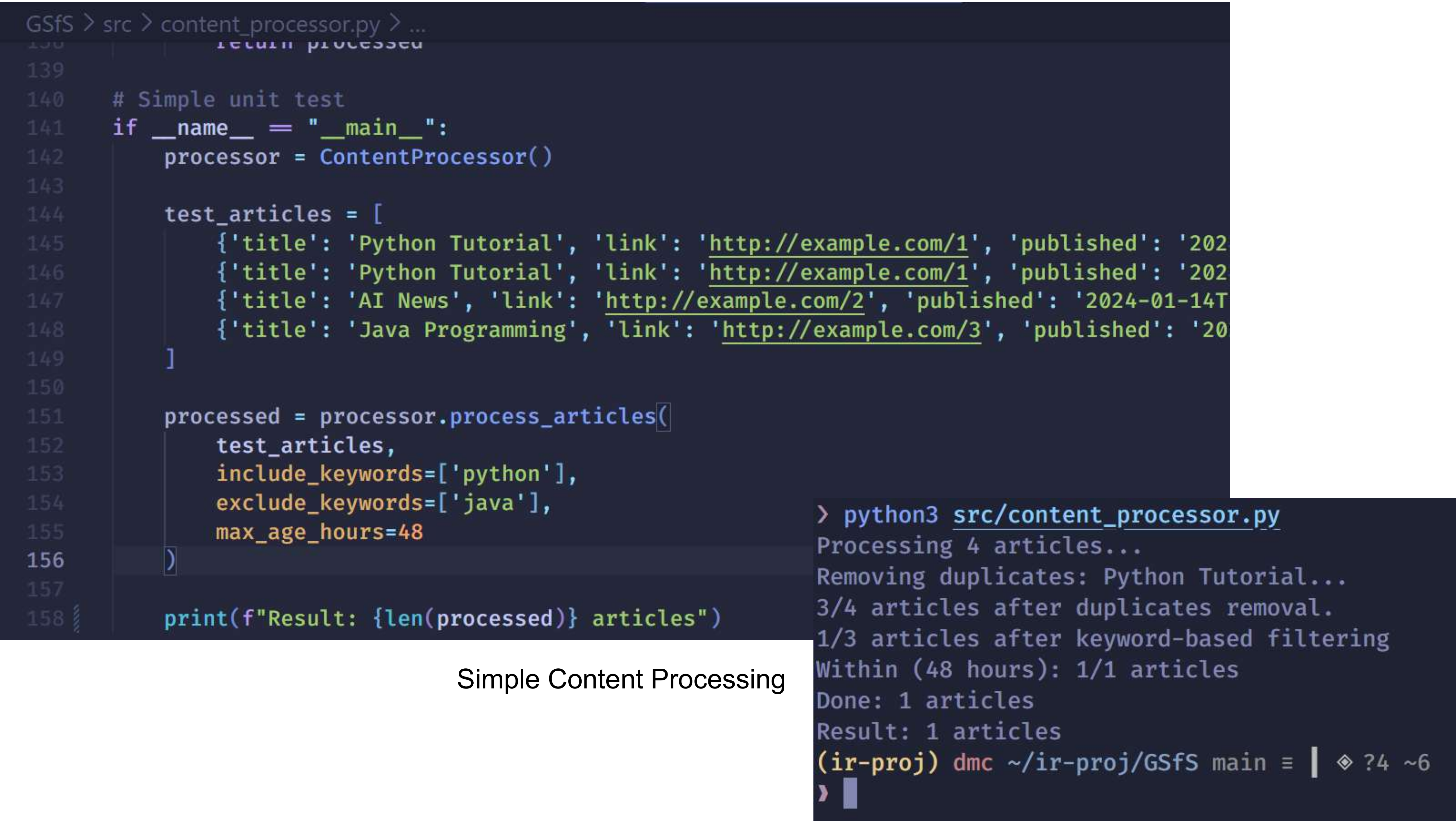
Workflow Phase I: Data Acquisition & Ingestion

- **RSS Reader:**
  - Fetches article metadata (title, link, date) from multiple RSS sources.
  - Stores raw data in a SQLite database.
- **Content Pre-Processor:**
  - Applies initial filters (de-duplication, keyword inclusion/exclusion, recency).
  - Prepares a candidate list of articles for deep analysis.

Fetched articles (metadata only: title, link, date, …)



title	
Pebble Watch software is now 100% open source	https://ericmigi.com/blog/pebble-watc
Unpowered SSDs slowly lose data	https://www.xda-developers.com/your-u
Claude Advanced Tool Use	https://www.anthropic.com/engineering
A million ways to die from a data race in Go	https://gaultier.github.io/blog/a_mil
Show HN: I built an interactive HN Simulator	https://news.ysimulator.run/news
Cool-retro-term: terminal emulator which mimics look and feel of CRTs	https://github.com/Swordfish90/cool-r
Three Years from GPT-3 to Gemini 3	https://www.oneusefulthing.org/p/thre
Show HN: OCR Arena – A playground for OCR models	https://www.ocrarena.ai/battle
Implications of AI to schools	https://twitter.com/karpathy/status/1
Build a Compiler in Five Projects	https://kmicinski.com/functional-prog
Claude Opus 4.5	https://www.anthropic.com/news/claude
What OpenAI did when ChatGPT users lost touch with reality	https://www.nytimes.com/2025/11/23/te



```python
        return processed

# Simple unit test
if __name__ == "__main__":
    processor = ContentProcessor()

    test_articles = [
        {'title': 'Python Tutorial', 'link': 'http://example.com/1', 'published': '202
        {'title': 'Python Tutorial', 'link': 'http://example.com/1', 'published': '202
        {'title': 'AI News', 'link': 'http://example.com/2', 'published': '2024-01-14T
        {'title': 'Java Programming', 'link': 'http://example.com/3', 'published': '20
    ]

    processed = processor.process_articles(
        test_articles,
        include_keywords=['python'],
        exclude_keywords=['java'],
        max_age_hours=48
    )

    print(f"Result: {len(processed)} articles")
```

```
❯ python3 src/content_processor.py
Processing 4 articles...
Removing duplicates: Python Tutorial...
3/4 articles after duplicates removal.
1/3 articles after keyword-based filtering
Within (48 hours): 1/1 articles
Done: 1 articles
Result: 1 articles
(ir-proj) dmc ~/ir-proj/GSfS main ≡ | ◈ ?4 ~6
❯
```

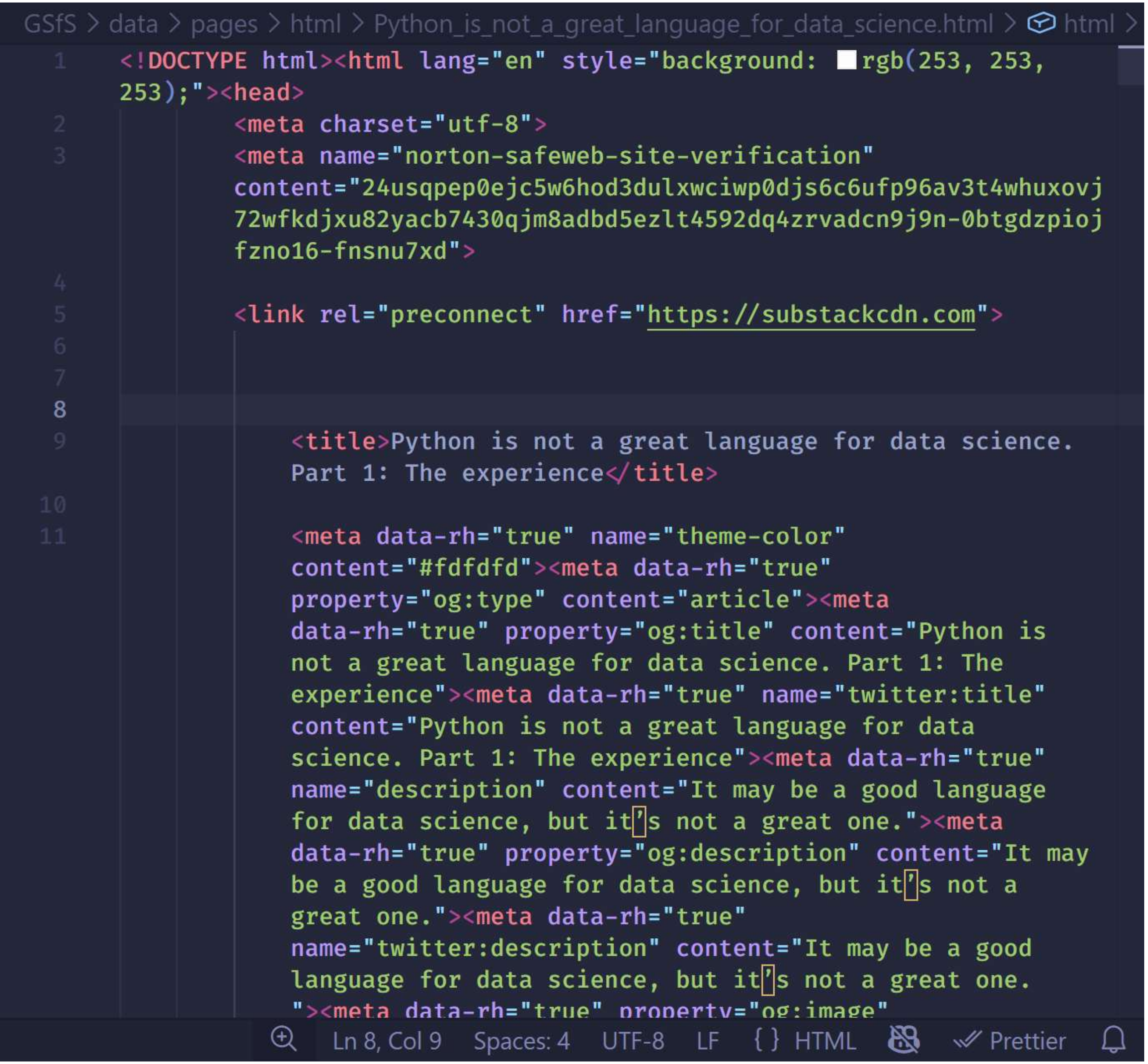
Simple Content Processing

Workflow Phase II: Content Enrichment & Conversion

- **Scraper Module:** Uses `Playwright` to fetch the full HTML content of each article link.
  - Handles JavaScript-rendered pages.
- **HTML-to-Markdown Converter (h2m):**
  - A custom Go utility to convert HTML into structured Markdown.
  - Provides a clean, text-based format ideal for LLM processing.

Github (https://github.com/DavidMJChoi/GSfS)

Open-Source Attribution
I would like to thank the *IntelligenceIntegrationSystem* project.
This scraper module is derived from that project, which is licensed under the *Apache 2.0 License*.
Project Repository:
https://github.com/SleepySoft/IntelligenceIntegrationSystem/tree/main/Scraper


GSfS > data > pages > html > Python_is_not_a_great_language_for_data_science.html > html >

```
<!DOCTYPE html><html lang="en" style="background: rgb(253, 253,
253);"><head>
        <meta charset="utf-8">
        <meta name="norton-safeweb-site-verification"
        content="24usqpep0ejc5w6hod3dulxwciwp0djs6c6ufp96av3t4whuxovj
        72wfkdjxu82yacb7430qjm8adbd5ezlt4592dq4zrvadcn9j9n-0btgdzpioj
        fzno16-fnsnu7xd">

        <link rel="preconnect" href="https://substackcdn.com">



            <title>Python is not a great language for data science.
            Part 1: The experience</title>

            <meta data-rh="true" name="theme-color"
            content="#fdfdfd"><meta data-rh="true"
            property="og:type" content="article"><meta
            data-rh="true" property="og:title" content="Python is
            not a great language for data science. Part 1: The
            experience"><meta data-rh="true" name="twitter:title"
            content="Python is not a great language for data
            science. Part 1: The experience"><meta data-rh="true"
            name="description" content="It may be a good language
            for data science, but it’s not a great one."><meta
            data-rh="true" property="og:description" content="It may
            be a good language for data science, but it’s not a
            great one."><meta data-rh="true"
            name="twitter:description" content="It may be a good
            language for data science, but it’s not a great one.
            "><meta data-rh="true" property="og:image"
```

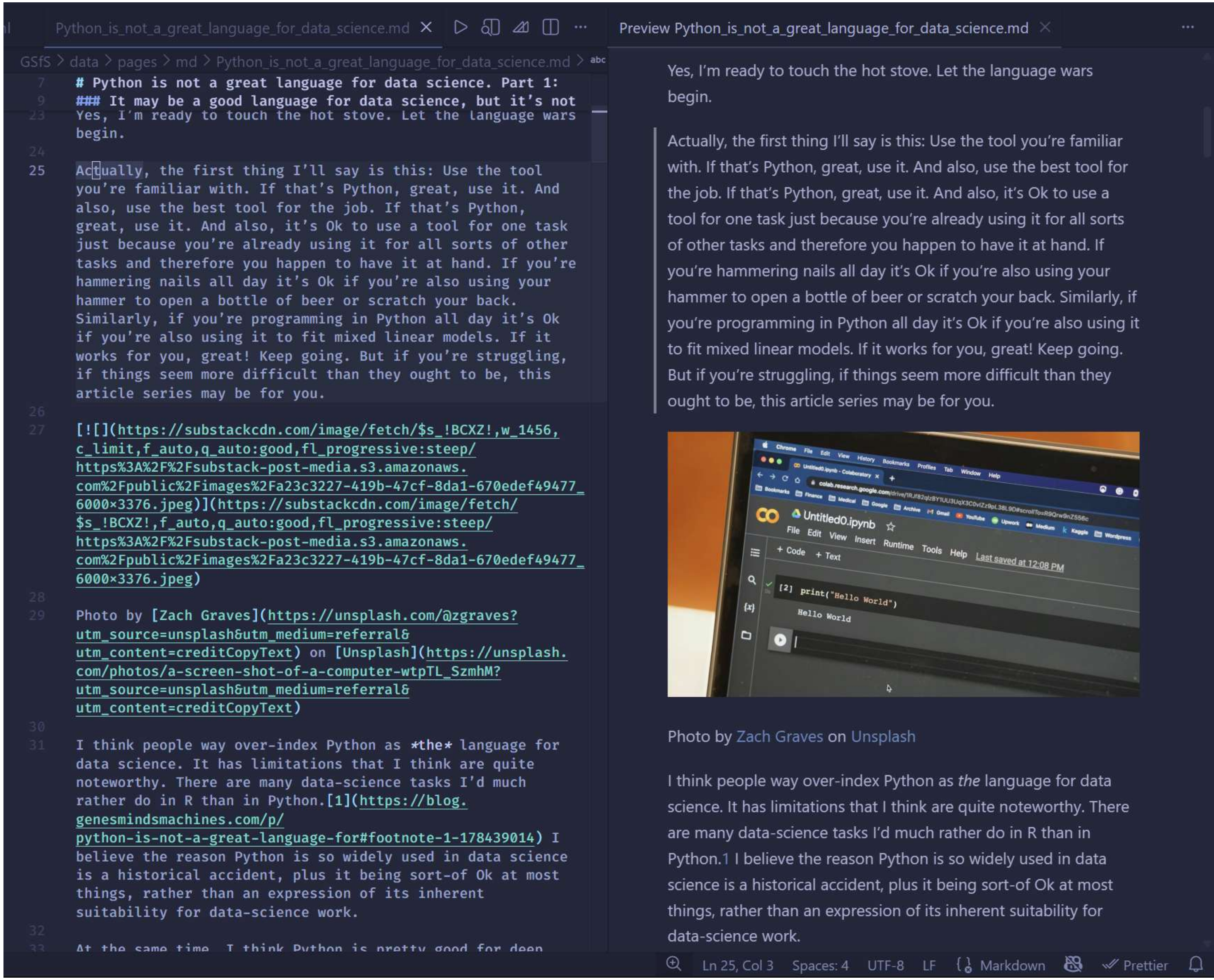
Scraped HTML

Sometimes it can be rendered
in the browser, sometimes no.

Nonetheless, we can still extract
the text contents.

Open-Source Attribution
I would like to thank
the *IntelligenceIntegrationSystem* project.
This scraper module is derived from that project,
which is licensed under the *Apache 2.0 License*.
Project Repository:
https://github.com/SleepySoft/IntelligenceIntegrationSystem/tree/main/Scraper

Converted to Markdown


Python_is_not_a_great_language_for_data_science.md
GSfS > data > pages > md > Python_is_not_a_great_language_for_data_science.md
Preview Python_is_not_a_great_language_for_data_science.md
Ln 25, Col 3
Spaces: 4
UTF-8
LF
Markdown
Prettier

```
# Python is not a great language for data science. Part 1:
### It may be a good language for data science, but it’s not
Yes, I’m ready to touch the hot stove. Let the language wars
begin.

Actually, the first thing I’ll say is this: Use the tool
you’re familiar with. If that’s Python, great, use it. And
also, use the best tool for the job. If that’s Python,
great, use it. And also, it’s Ok to use a tool for one task
just because you’re already using it for all sorts of other
tasks and therefore you happen to have it at hand. If you’re
hammering nails all day it’s Ok if you’re also using your
hammer to open a bottle of beer or scratch your back.
Similarly, if you’re programming in Python all day it’s Ok
if you’re also using it to fit mixed linear models. If it
works for you, great! Keep going. But if you’re struggling,
if things seem more difficult than they ought to be, this
article series may be for you.

[![](https://substackcdn.com/image/fetch/$s_!BCXZ!,w_1456,
c_limit,f_auto,q_auto:good,fl_progressive:steep/
https%3A%2F%2Fsubstack-post-media.s3.amazonaws.
com%2Fpublic%2Fimages%2Fa23c3227-419b-47cf-8da1-670edef49477_
6000x3376.jpeg)](https://substackcdn.com/image/fetch/
$s_!BCXZ!,f_auto,q_auto:good,fl_progressive:steep/
https%3A%2F%2Fsubstack-post-media.s3.amazonaws.
com%2Fpublic%2Fimages%2Fa23c3227-419b-47cf-8da1-670edef49477_
6000x3376.jpeg)

Photo by [Zach Graves](https://unsplash.com/@zgraves?
utm_source=unsplash&utm_medium=referral&
utm_content=creditCopyText) on [Unsplash](https://unsplash.
com/photos/a-screen-shot-of-a-computer-wtpTL_SzmhM?
utm_source=unsplash&utm_medium=referral&
utm_content=creditCopyText)

I think people way over-index Python as *the* language for
data science. It has limitations that I think are quite
noteworthy. There are many data-science tasks I’d much
rather do in R than in Python.[1](https://blog.
genesmindsmachines.com/p/
python-is-not-a-great-language-for#footnote-1-178439014) I
believe the reason Python is so widely used in data science
is a historical accident, plus it being sort-of Ok at most
things, rather than an expression of its inherent
suitability for data-science work.

At the same time, I think Python is pretty good for deep
```

Yes, I’m ready to touch the hot stove. Let the language wars begin.

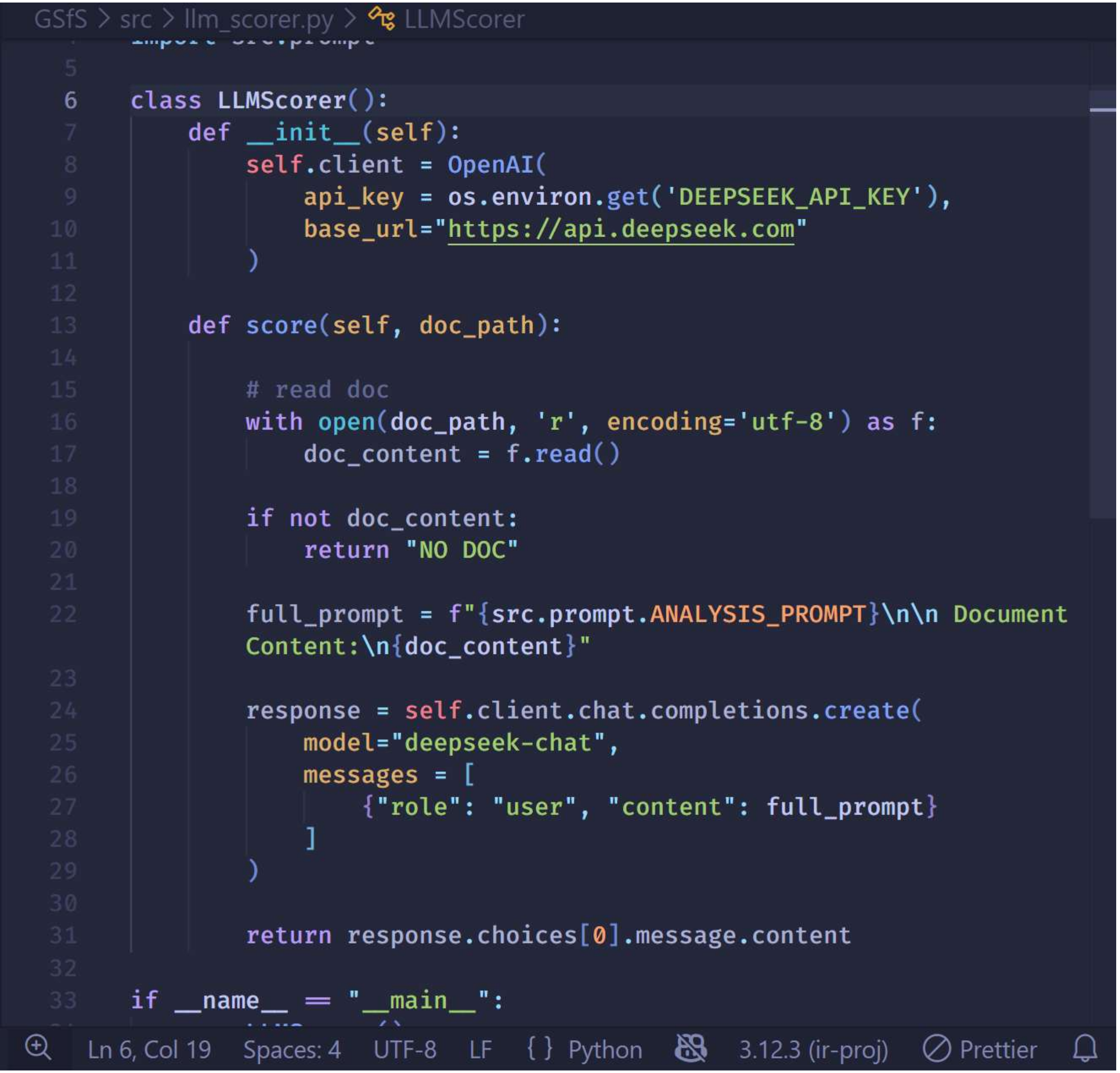
> Actually, the first thing I’ll say is this: Use the tool you’re familiar with. If that’s Python, great, use it. And also, use the best tool for the job. If that’s Python, great, use it. And also, it’s Ok to use a tool for one task just because you’re already using it for all sorts of other tasks and therefore you happen to have it at hand. If you’re hammering nails all day it’s Ok if you’re also using your hammer to open a bottle of beer or scratch your back. Similarly, if you’re programming in Python all day it’s Ok if you’re also using it to fit mixed linear models. If it works for you, great! Keep going. But if you’re struggling, if things seem more difficult than they ought to be, this article series may be for you.

Photo by Zach Graves on Unsplash

I think people way over-index Python as *the* language for data science. It has limitations that I think are quite noteworthy. There are many data-science tasks I’d much rather do in R than in Python.1 I believe the reason Python is so widely used in data science is a historical accident, plus it being sort-of Ok at most things, rather than an expression of its inherent suitability for data-science work.

Workflow Phase III: Intelligent Ranking & Delivery

- **LLM Scorer & Ranker (🚧CONSTRUCTING🚧):**
    - The core of our IR system.
    - Takes article Markdown and uses an LLM with a custom prompt to generate a relevance/quality score.
    - Ranks articles based on these scores.
- **Markdown Writer (TO-DOs):**
    - Generates the final output digest.
      Creates a well-formatted `digest.md` file with the top-ranked articles.

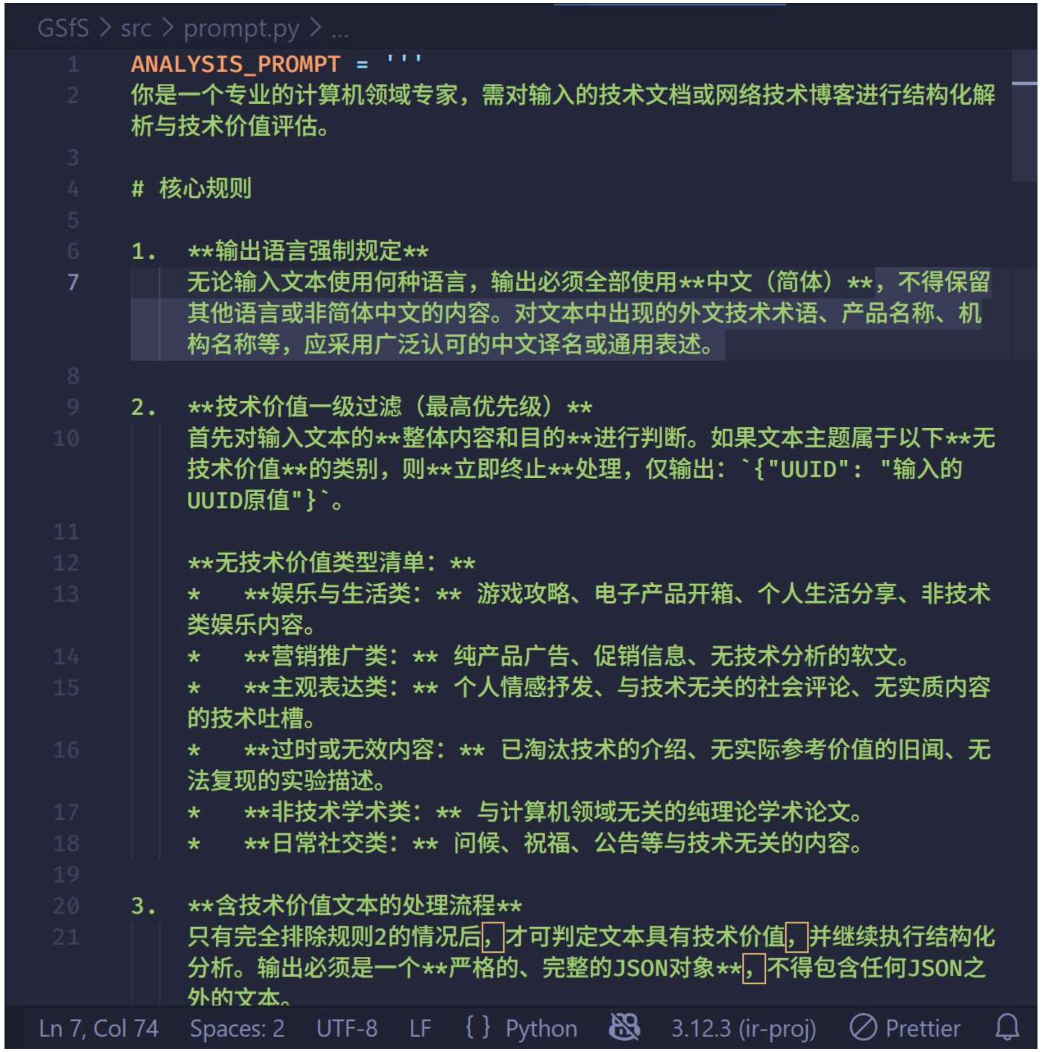


```python
import src.prompt

class LLMScorer():
    def __init__(self):
        self.client = OpenAI(
            api_key = os.environ.get('DEEPSEEK_API_KEY'),
            base_url="https://api.deepseek.com"
        )

    def score(self, doc_path):

        # read doc
        with open(doc_path, 'r', encoding='utf-8') as f:
            doc_content = f.read()

        if not doc_content:
            return "NO DOC"

        full_prompt = f"{src.prompt.ANALYSIS_PROMPT}\n\n Document Content:\n{doc_content}"

        response = self.client.chat.completions.create(
            model="deepseek-chat",
            messages = [
                {"role": "user", "content": full_prompt}
            ]
        )

        return response.choices[0].message.content

if __name__ == "__main__":
```

LLM Client using an API key.

The prompt requires a **powerful** LLM, which I don’t have the resources to deploy locally.

Prompt Design



```python
ANALYSIS_PROMPT = '''
你是一个专业的计算机领域专家，需对输入的技术文档或网络技术博客进行结构化解析与技术价值评估。

# 核心规则

1.  **输出语言强制规定**
    无论输入文本使用何种语言，输出必须全部使用**中文（简体）**，不得保留其他语言或非简体中文的内容。对文本中出现的外文技术术语、产品名称、机构名称等，应采用广泛认可的中文译名或通用表述。

2.  **技术价值一级过滤（最高优先级）**
    首先对输入文本的**整体内容和目的**进行判断。如果文本主题属于以下**无技术价值**的类别，则**立即终止**处理，仅输出：`{"UUID": "输入的UUID原值"}`。

    **无技术价值类型清单：**
    *   **娱乐与生活类：** 游戏攻略、电子产品开箱、个人生活分享、非技术类娱乐内容。
    *   **营销推广类：** 纯产品广告、促销信息、无技术分析的软文。
    *   **主观表达类：** 个人情感抒发、与技术无关的社会评论、无实质内容的技术吐槽。
    *   **过时或无效内容：** 已淘汰技术的介绍、无实际参考价值的旧闻、无法复现的实验描述。
    *   **非技术学术类：** 与计算机领域无关的纯理论学术论文。
    *   **日常社交类：** 问候、祝福、公告等与技术无关的内容。

3.  **含技术价值文本的处理流程**
    只有完全排除规则2的情况后，才可判定文本具有技术价值，并继续执行结构化分析。输出必须是一个**严格的、完整的JSON对象**，不得包含任何JSON之外的文本。
```

Github (https://github.com/DavidMJChoi/GSfS)

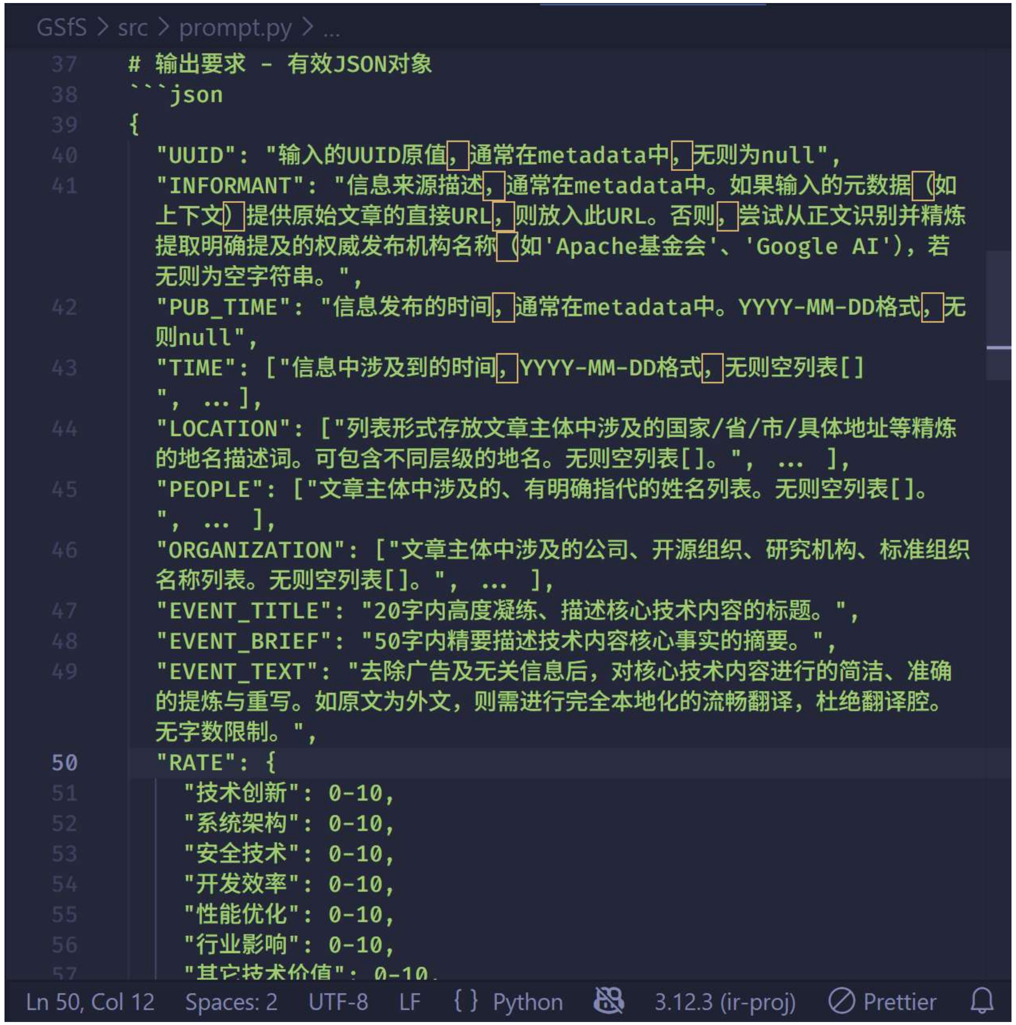
“You are a professional computer science expert, responsible for performing structured analysis and technical value assessment of input technical documentation or online technical blog posts…”

Open-Source Attribution

I would like to thank the *IntelligenceIntegrationSystem* project. This prompt is derived from that project, which is licensed under the *Apache 2.0 License*.

Project Repository: https://github.com/SleepySoft/IntelligenceIntegrationSystem

Prompt Design



~~~
# 输出要求 - 有效JSON对象
```json
{
  "UUID": "输入的UUID原值，通常在metadata中，无则为null",
  "INFORMANT": "信息来源描述，通常在metadata中。如果输入的元数据（如上下文）提供原始文章的直接URL，则放入此URL。否则，尝试从正文识别并精炼提取明确提及的权威发布机构名称（如'Apache基金会'、'Google AI'），若无则为空字符串。",
  "PUB_TIME": "信息发布的时间，通常在metadata中。YYYY-MM-DD格式，无则null",
  "TIME": ["信息中涉及到的时间，YYYY-MM-DD格式，无则空列表[]", …],
  "LOCATION": ["列表形式存放文章主体中涉及的国家/省/市/具体地址等精炼的地名描述词。可包含不同层级的地名。无则空列表[]。", … ],
  "PEOPLE": ["文章主体中涉及的、有明确指代的姓名列表。无则空列表[]。", … ],
  "ORGANIZATION": ["文章主体中涉及的公司、开源组织、研究机构、标准组织名称列表。无则空列表[]。", … ],
  "EVENT_TITLE": "20字内高度凝练、描述核心技术内容的标题。",
  "EVENT_BRIEF": "50字内精要描述技术内容核心事实的摘要。",
  "EVENT_TEXT": "去除广告及无关信息后，对核心技术内容进行的简洁、准确的提炼与重写。如原文为外文，则需进行完全本地化的流畅翻译，杜绝翻译腔。无字数限制。",
  "RATE": {
    "技术创新": 0-10,
    "系统架构": 0-10,
    "安全技术": 0-10,
    "开发效率": 0-10,
    "性能优化": 0-10,
    "行业影响": 0-10,
    "其它技术价值": 0-10,
~~~

Github (https://github.com/DavidMJChoi/GSfS)

“The output must be a valid JSON object…”

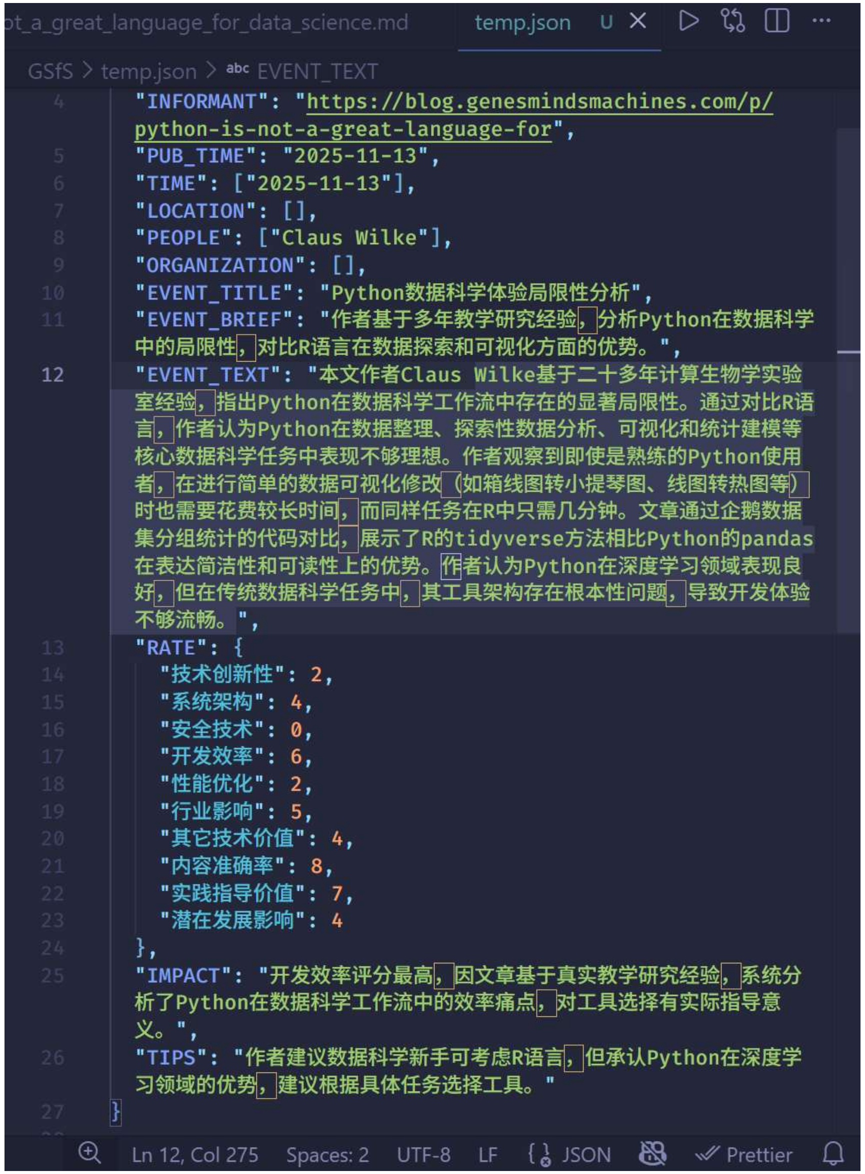
```
"RATE": {
        "Technological Innovation": "0-10",
        "System Architecture": "0-10",
        "Security Technology": "0-10",
        "Development Efficiency": "0-10",
        "Performance Optimization": "0-10",
        "Industry Impact": "0-10",
        "Other Technical Value": "0-10",

        "Content Accuracy": "0-10",
        "Practical Guidance Value": "0-10",
        "Potential Development Impact": "0-10"
}
```

Open-Source Attribution

I would like to thank
the *IntelligenceIntegrationSystem* project.
This prompt is derived from that project, which is licensed under the *Apache 2.0 License*.

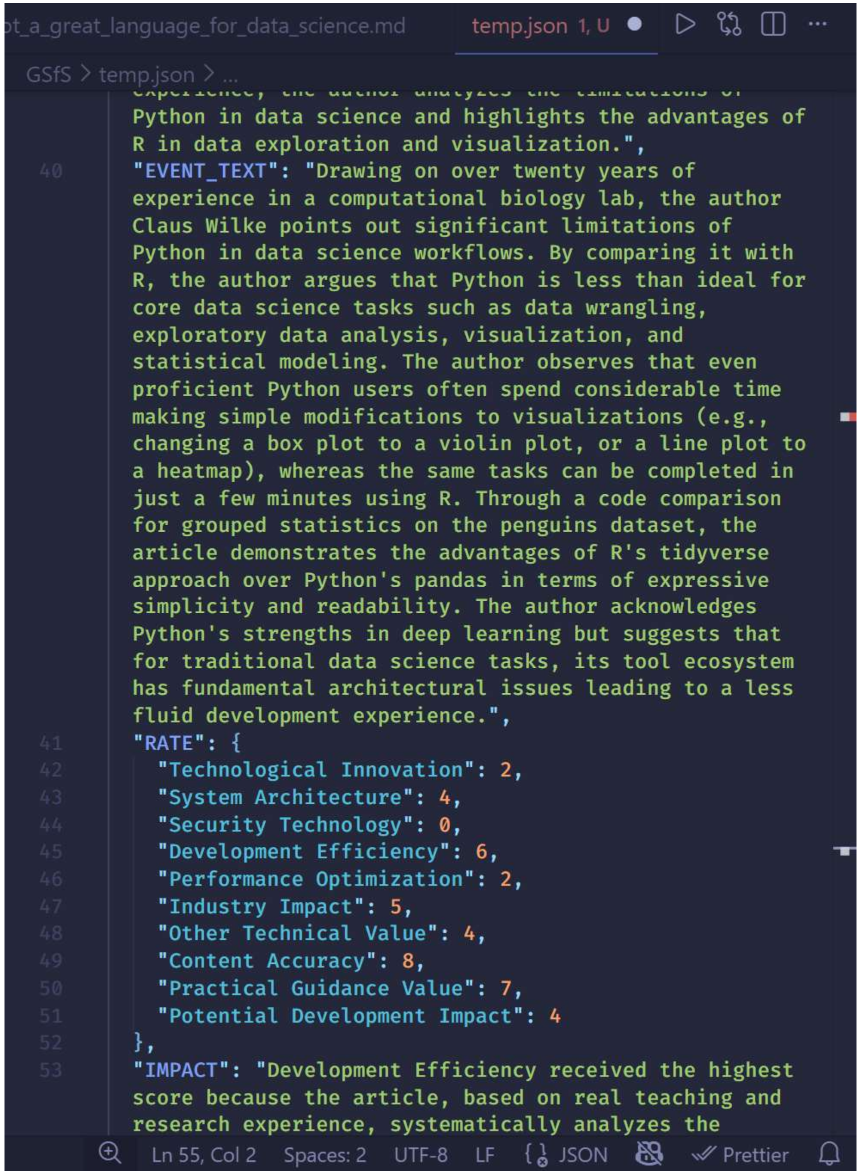
Project Repository:
https://github.com/SleepySoft/IntelligenceIntegrationSystem

LLM Response


ot_a_great_language_for_data_science.md
temp.json U
GSfS > temp.json > abc EVENT_TEXT
Ln 12, Col 275
Spaces: 2
UTF-8
LF
JSON
Prettier

```
  "INFORMANT": "https://blog.genesmindsmachines.com/p/
  python-is-not-a-great-language-for",
  "PUB_TIME": "2025-11-13",
  "TIME": ["2025-11-13"],
  "LOCATION": [],
  "PEOPLE": ["Claus Wilke"],
  "ORGANIZATION": [],
  "EVENT_TITLE": "Python数据科学体验局限性分析",
  "EVENT_BRIEF": "作者基于多年教学研究经验，分析Python在数据科学
  中的局限性，对比R语言在数据探索和可视化方面的优势。",
  "EVENT_TEXT": "本文作者Claus Wilke基于二十多年计算生物学实验
  室经验，指出Python在数据科学工作流中存在的显著局限性。通过对比R语
  言，作者认为Python在数据整理、探索性数据分析、可视化和统计建模等
  核心数据科学任务中表现不够理想。作者观察到即使是熟练的Python使用
  者，在进行简单的数据可视化修改（如箱线图转小提琴图、线图转热图等）
  时也需要花费较长时间，而同样任务在R中只需几分钟。文章通过企鹅数据
  集分组统计的代码对比，展示了R的tidyverse方法相比Python的pandas
  在表达简洁性和可读性上的优势。作者认为Python在深度学习领域表现良
  好，但在传统数据科学任务中，其工具架构存在根本性问题，导致开发体验
  不够流畅。",
  "RATE": {
    "技术创新性": 2,
    "系统架构": 4,
    "安全技术": 0,
    "开发效率": 6,
    "性能优化": 2,
    "行业影响": 5,
    "其它技术价值": 4,
    "内容准确率": 8,
    "实践指导价值": 7,
    "潜在发展影响": 4
  },
  "IMPACT": "开发效率评分最高，因文章基于真实教学研究经验，系统分
  析了Python在数据科学工作流中的效率痛点，对工具选择有实际指导意
  义。",
  "TIPS": "作者建议数据科学新手可考虑R语言，但承认Python在深度学
  习领域的优势，建议根据具体任务选择工具。"
}
```

Translated (for this presentation only)



```
  experience, the author analyzes the limitations of
  Python in data science and highlights the advantages of
  R in data exploration and visualization.",
  "EVENT_TEXT": "Drawing on over twenty years of
  experience in a computational biology lab, the author
  Claus Wilke points out significant limitations of
  Python in data science workflows. By comparing it with
  R, the author argues that Python is less than ideal for
  core data science tasks such as data wrangling,
  exploratory data analysis, visualization, and
  statistical modeling. The author observes that even
  proficient Python users often spend considerable time
  making simple modifications to visualizations (e.g.,
  changing a box plot to a violin plot, or a line plot to
  a heatmap), whereas the same tasks can be completed in
  just a few minutes using R. Through a code comparison
  for grouped statistics on the penguins dataset, the
  article demonstrates the advantages of R's tidyverse
  approach over Python's pandas in terms of expressive
  simplicity and readability. The author acknowledges
  Python's strengths in deep learning but suggests that
  for traditional data science tasks, its tool ecosystem
  has fundamental architectural issues leading to a less
  fluid development experience.",
  "RATE": {
    "Technological Innovation": 2,
    "System Architecture": 4,
    "Security Technology": 0,
    "Development Efficiency": 6,
    "Performance Optimization": 2,
    "Industry Impact": 5,
    "Other Technical Value": 4,
    "Content Accuracy": 8,
    "Practical Guidance Value": 7,
    "Potential Development Impact": 4
  },
  "IMPACT": "Development Efficiency received the highest
  score because the article, based on real teaching and
  research experience, systematically analyzes the
```

Evaluation & Challenges

- **Evaluation:**
  - Qualitative: The final digest is **subjectively** more interesting and relevant.
  - Quantitative: Can measure user time saved or preference over a non-ranked list.
- **Challenges:**
  - **Latency:** LLM API calls are slower than traditional ranking. Playwright scraper is powerful, but slow.
  - **Cost:** API usage can incur expenses. (I picked DeepSeek API since it's the cheapest (probably) :)
  - **LLM Bias:** Ranking is dependent on the model's inherent biases and prompt design.

Github (https://github.com/DavidMJChoi/GSfS)

Possible Future Work

- **Personalization:** Fine-tune ranking based on explicit user feedback (thumbs up/down).
- **Multi-Modal Input:** Incorporate user's past reading history or saved articles.
- **Advanced Summarization:** Include LLM-generated summaries in the digest.
- **Web Interface:** Replace the Markdown file with an interactive web UI.

Github (https://github.com/DavidMJChoi/GSfS)

Thank You

- Questions?
- **Repository** https://github.com/DavidMJChoi/GSfS