

Processing Document Collections with LLMs

A Practical Workflow

Abigail Haddad

Every Organization Has Documents



Medical Records



Resumes



Customer Complaints



Error Logs



Contracts



Reports

Agenda

- 📌 **Part 1:** The Project - Processing Regulations.gov Comments
- 📌 **Part 2:** The Larger Problem - Document Processing Everywhere
- 📌 **Part 3:** Applying This Pattern - Your Documents, Your Tools
- 📌 **Part 4:** What I Learned - Keep It Simple
- 📌 **Part 5:** Evaluation - Test What Matters
- 📌 **Part 6:** Building A Couple More Blocks - From Script to Product

Part 1: The Project

Regulations.gov Comments

The screenshot shows the main page of the **Regulations.gov** website. At the top left is the **Regulations.gov** logo with the tagline *Your Voice in Federal Decision Making*. At the top right is a **SUPPORT** button. Below the header, there is a link to **< View Docket**. In the center, there is a blue icon labeled **PR PROPOSED RULE**. To the right of the title, there is a **Share** button with a dropdown arrow. The main title of the document is **Improving Performance, Accountability and Responsiveness in the Civil Service**, posted by the **Office of Personnel Management** on **Apr 23, 2025**. A dark grey button on the left says **Closed for Comments**. A light grey box on the right indicates that the comment period ended on **Jun 7, 2025 at 11:59 PM EDT**.

Regulations.gov
Your Voice in Federal Decision Making

SUPPORT

< View Docket

PR PROPOSED RULE

Share

Improving Performance, Accountability and Responsiveness in the Civil Service

Posted by the **Office of Personnel Management** on Apr 23, 2025

Closed for Comments

Comment Period Ended: Jun 7, 2025 at 11:59 PM EDT

What Did the Comments Say?

AGREE



DISAGREE



My Deadline



My Usual Blocks



The Weird Stuff

📄 Dark scanned PDFs

🎭 Random meme attachments

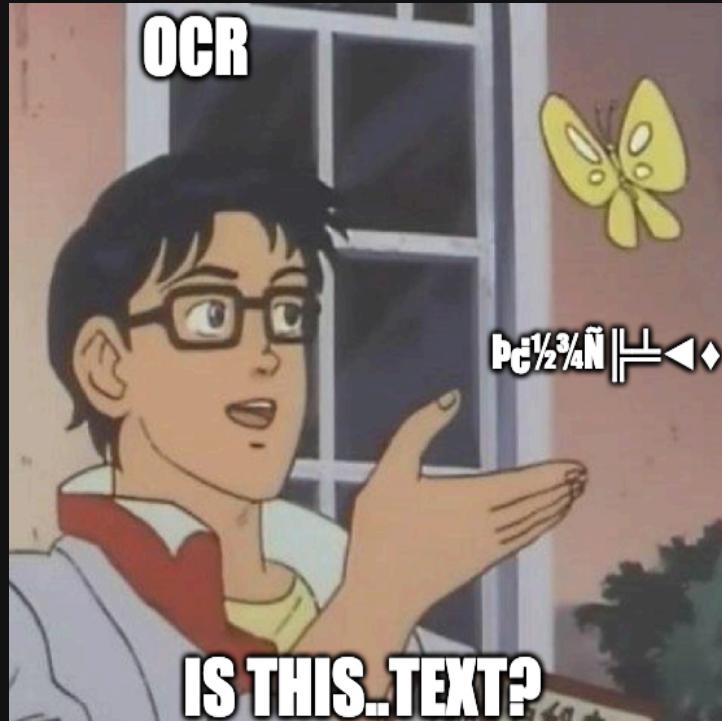
📸 Blurry photos of handwritten notes



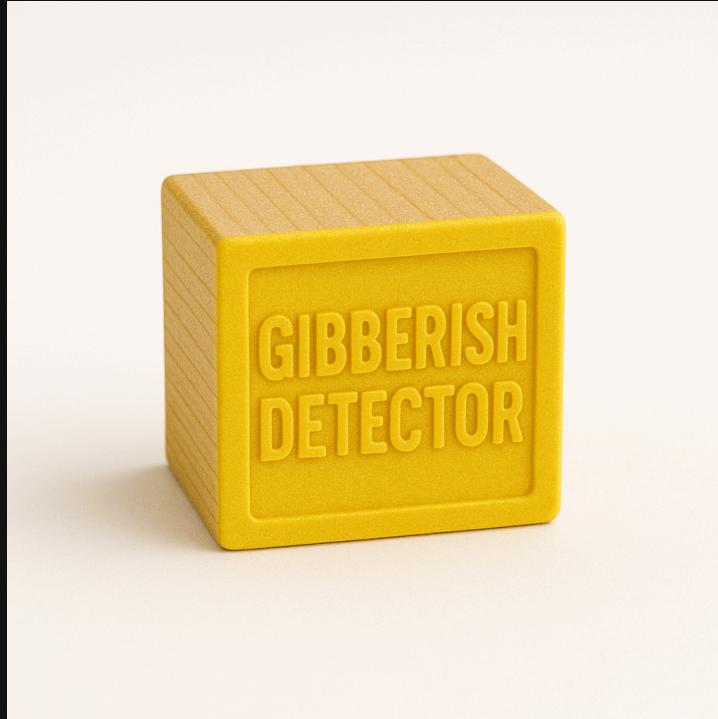
I Added a New Block



I Occasionally Got Gibberish



So I Built a Gibberish Detector



Now I'm Debugging My Gibberish Detector



Finally, I Try Gemini To Extract Text...

```
if extraction_fails(doc):
    result = gemini.process(doc)
```

It Works!!



Gemini Initially Cost Me Seven Cents



The Results

33,647

Against
(94%)

1,596

Support
(4.5%)

306

Neutral
(0.9%)

Many "support" comments were duplicates/campaigns

Here It Is

Public Comments on "Schedule F" Regulation
Comments about the proposed Schedule Policy/Career Regulation - Improving Performance, Accountability and Responsiveness in the Civil Service ↗

Statistics Overview

TOTAL COMMENTS 35,550	FOR 1,596 For	AGAINST 33,647 Against	NEUTRAL/UNCLEAR 306 Neutral/ Unclear
---------------------------------	----------------------------	-------------------------------------	--

Unique Statistics (Duplicates Removed)

UNIQUE COMMENTS 28,096	FOR 251 For	AGAINST 27,546 Against	NEUTRAL/UNCLEAR 298 Neutral/ Unclear
----------------------------------	--------------------------	-------------------------------------	--

Filters

No filters applied
Use the filter buttons below to refine your data view

Filter Options

Title, Stance, Key Quote, Themes, Rationale, Comment, Category, Duplicates, Source Link, Comment ID, Has Attachments, Posted Date, Received Date, Organization, Attachment Titles

Search

Search comments...

Part 2: The Larger Problem

Every Organization Has Documents



Medical Records



Resumes



Customer Complaints



Error Logs



Contracts



Reports

Same Question Every Time

- Is this fraudulent? → **yes/no**
- What's the complaint about? → **category**
- Does this candidate qualify? → **yes/no + reason**
- What's the root cause? → **specific issue**

The Answer? Document Processing Pipeline



The Three Steps

GET

Pull text from PDFs, Word docs, APIs, or user input

PROCESS

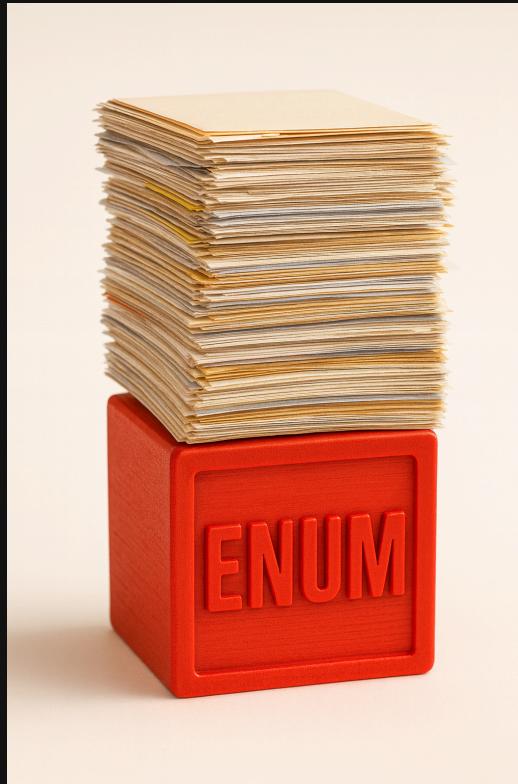
Use LLMs, regex, BERT, or other tools to extract meaning

DO

Create visualizations, populate templates, or trigger actions



Getting Back ONLY "Agree" or "Disagree"



Enums → Structured output → Wrangleable results

LiteLLM Structured Output + Pydantic

```
import os
from litellm import completion
from pydantic import BaseModel

# add to env var
os.environ["OPENAI_API_KEY"] = ""

messages = [{"role": "user", "content": "List 5 important events in the XIX century"}]

class CalendarEvent(BaseModel):
    name: str
    date: str
    participants: list[str]

class EventsList(BaseModel):
    events: list[CalendarEvent]

resp = completion(
    model="gpt-4o-2024-08-06",
    messages=messages,
    response_format=EventsList
)

print("Received={}".format(resp))
```

Put Anything You Want in The Prompt

```
class CommentAnalyzer:  
    """LiteLLM-based analyzer for public comments using Pydantic models for structured output."""  
    def __init__(self, model=None, timeout_seconds=None):  
        self.model = model or config.llm.model  
        self.timeout_seconds = timeout_seconds or config.llm.timeout  
  
        # Ensure API key is available  
        if "OPENAI_API_KEY" not in os.environ:  
            raise ValueError("OPENAI_API_KEY not found in environment variables or .env file")  
  
    def get_system_prompt(self):  
        return """You are analyzing public comments submitted regarding a proposed rule to implement "Schedule F" (or "Schedule Policy/Career Development").  
  
This proposed rule would allow federal agencies to reclassify career civil servants in policy-influencing positions into a new employment category.  
  
1. Stance: Determine if the comment is "For" (supporting the rule), "Against" (opposing the rule), or "Neutral/Unclear" by examining both explicit and implicit language.  
  
Classification guidelines with special attention to boundary cases:  
  
- "For": Comment supports the rule, defends its merits, or argues for implementation. Look for: praise of accountability, presidential authority, and support for the rule's goals.  
  
- "Against": Comment opposes the rule, including indirect opposition through thematic alignment. Critical indicators include:  
  * Questions about constitutionality or legal concerns, even without explicit opposition  
  * Opposition to the rule's implementation or its impact on career civil servants.  
  * Requests for more information or further details about the rule's impact.  
  * Statements that contradict the rule's purpose or mission.  
  
2. Impact: Assess the potential impact of the comment on the rule's implementation. Consider factors such as the scope of the rule, the affected population, and the potential consequences for career civil servants.  
  
3. Specificity: Determine the level of specificity in the comment. Is it general or specific to certain aspects of the rule? This will help in identifying the most relevant classification.  
  
4. Tone: Analyze the tone of the comment. Is it positive, negative, or neutral? This can provide additional context for the classification.  
  
5. Context: Consider the context in which the comment was made. Is it part of a larger discussion or a stand-alone statement? This can affect the interpretation of the comment.  
  
Classification categories:  
- For: 1 point  
- Against: -1 point  
- Neutral/Unclear: 0 points  
  
Final score: [Score] / [Total Points] = [Percentage]  
Feedback: [Feedback] / [Total Feedback] = [Percentage]"""
```

What I Actually Extracted

- **Stance:** agree / disagree / neutral-other
- **Themes:** multi-class classification
- **Summary:** unstructured text
- **Representative quote:** actual text
- **Reasoning:** why this classification

Part 3: Applying This Pattern



Your Documents

- ✓ Clean PDFs?
- ✓ Scanned images?
- ✓ Mixed formats?
- ✓ Handwritten?

 Your Goals

- Extract fields?
- Classify docs?
- Summarize?
- Find patterns?

Your Constraints



Privacy

Can data leave?



Budget

API costs OK?



Time

One-off or ongoing?



Explain

Show your work?

Should I Throw It Into an LLM?



MAYBE!!!



Your Toolkit

GET

- PyPDF2, pdfplumber
- Tesseract OCR
- BeautifulSoup
- APIs

PROCESS

- OpenAI, Claude, Gemini
- BERT
- Regex

DO

- Quarto
- Email/Slack alerts
- API endpoints

Remember My Seven Cents?

Gemini only worked because:

- ✓ Data could leave my environment
- ✓ 35,000 documents, most with no attachments = small enough for the budget

Different constraints = Different solution

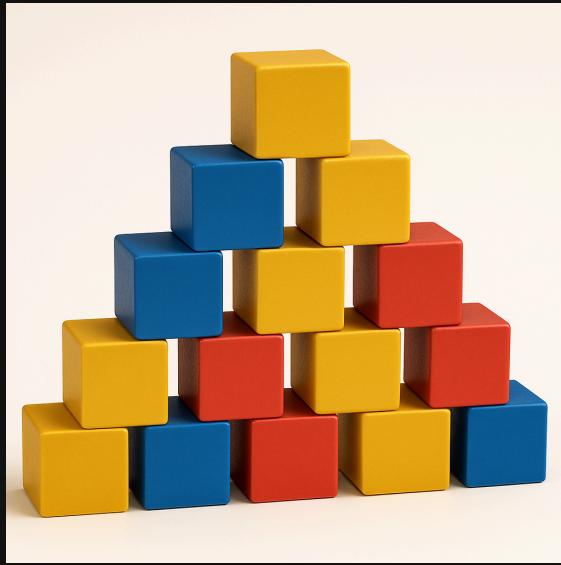
Part 4: What I Learned

Start Simple



Build only the blocks you need

You Can't Build All The Blocks



Scaling Up



Handle Failures



Parallelize



Track State



Real Databases

Part 5: Evaluation

Systematic Evaluation



Build test sets:

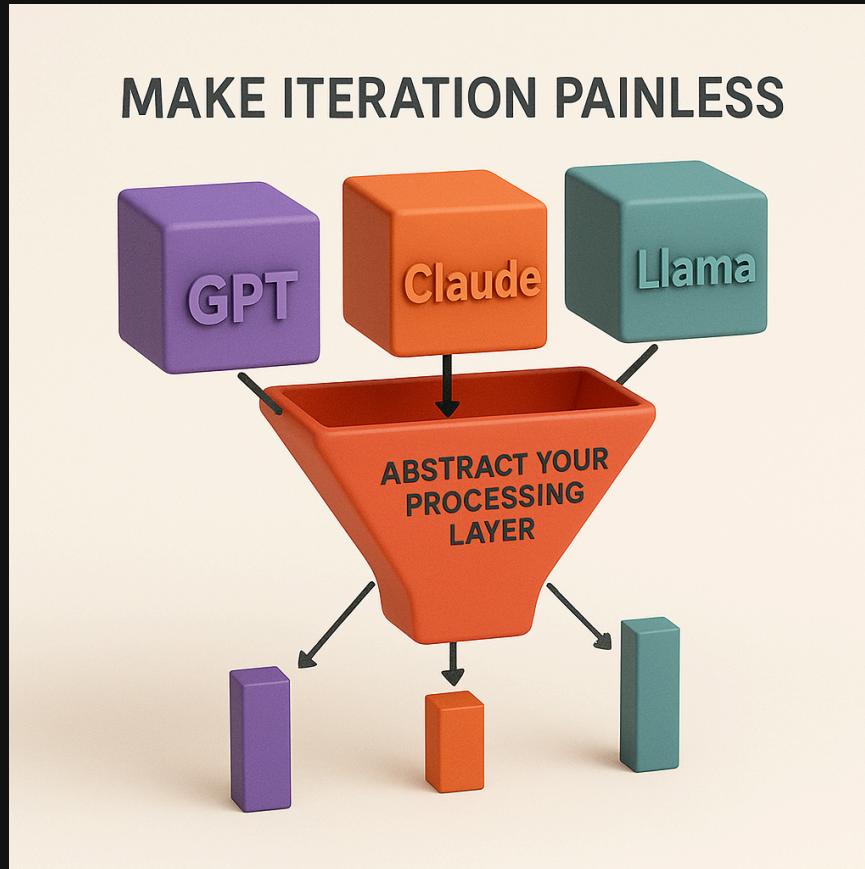
- Real examples from your data
- The weird edge cases
- What failed before



Test what matters:

- Accuracy on YOUR data
- How it fails (not just how often)
- **What keeps me up at night?**

Make Iteration Painless



Part 6: Building A Couple More Blocks

From One-Off Script to Product



Manual Analysis →

One regulation
35K comments



Automated Tool

Any public comment dataset
Finds patterns, writes prompts

I Built A Couple More Blocks



How It Works

1. **Upload** sample comments
2. **It discovers** the arguments
3. **It writes** the classification prompts
4. **Fully autonomous** analysis

Structured output/Enum is still doing the work here

New Analysis!

Request for Information on Health Technology Ecosystem

This regulation seeks public input on the development and integration of health technology ecosystems, focusing on digital health products and their impact on healthcare delivery and patient outcomes.

CMS Docket: CMS-2025-0050

50

Total Comments

35

With Attachments

May 28, 2025 to June 25,
2025

Date Range

Themes Distribution

Click on any theme to see positions, then click positions to see relationships

Health Data Interoperability

(2 positions) ▾



I did demo this last week for a job interview

Find Me on the Internet!

🌐 abigailhaddad.netlify.app

Want to hire me to consult?

Fill out the form on my website!

Also find me on:

🔗 github.com/abigailhaddad

📝 presentofcoding.substack.com