Current Active Tools:

| Tool | Purpose | Usage |

|----------------------------|----------------------------|----------------------------|

| extractor\_gpt5\_oe\_final.py | Single document extraction | --single "paper\_name" |

| extractor\_gpt5\_batch.py | Parallel batch processing | --workers 3 --batch-size 5 |

| evidence\_inspector\_app.py | View extracted data | streamlit run |

| quick\_api\_test.py | Test GPT-5 connection | Direct run |

The system is now clean, well-documented, and ready for production use with OpenEvidence-grade medical evidence extraction.

> My computer is about to do a forced restart. What will happen to pending outputs

● During a forced restart, here's what will happen:

⚠️ What You'll Lose:

1. Any currently running extractions - GPT-5 API calls in progress will be terminated

2. Unsaved terminal output - Console logs not yet written to files

3. The raw response file (last\_oe\_final\_output.json) - Gets overwritten with each run

✅ What's Safe:

1. All completed extractions in data/oe\_final\_outputs/ - These are saved immediately after each document finishes

2. Batch summaries in data/oe\_batch\_outputs/ - Saved every 5 batches

3. Your source files - Input JSONs and PDFs remain untouched

🔄 Recovery After Restart:

For Batch Processing:

# The batch processor auto-detects completed files!

python tools/extractor\_gpt5\_batch.py

# It will show:

# Auto-skip: Found 37 already processed files

# Will process 13 new files out of 50 total

For Single File:

# Check what's already done

ls data/oe\_final\_outputs/\*.json

# Re-run will skip completed files automatically

💡 Before Restart (if you have 30 seconds):

# 1. Hit Ctrl+C to gracefully stop current extraction

# 2. Check what's been completed

ls -la data/oe\_final\_outputs/\*.json | wc -l

# 3. Note the last file being processed (for reference)

The system is designed to be restart-resilient - you won't lose completed work and can easily resume where you left off!

> CRTL C doesnt seem to work to stop it

● If Ctrl+C isn't working, here are your options:

🛑 Force Stop Methods:

1. Close the Terminal/Command Window

- Just click the X button to close the window entirely

- This immediately kills the Python process

2. Task Manager (Windows)

# Or press Ctrl+Shift+Esc

# Find "python.exe" or "python3.exe"

# Click "End Task"

3. New Terminal Window (if you can open one):

# Windows PowerShell

Get-Process python | Stop-Process -Force

# Or find the process ID

tasklist | findstr python

taskkill /PID [process\_id] /F

📝 What Gets Saved:

Good news: The extractor saves each document immediately after completion, so:

- ✅ Any files showing "✓ Complete" in the terminal are already saved

- ✅ Even if you force-kill mid-extraction, all previous documents are safe

- ❌ Only the currently processing document (showing "Processing...") will be lost

🔄 After Restart:

The batch processor will automatically skip all completed files:

python tools/extractor\_gpt5\_batch.py

# Will show: "Auto-skip: Found X already processed files"

Batch Processing Command for Remaining Files

You have 242 files remaining to process (312 total - 70 completed).

Overnight Batch Processing Command:

Python tools/extractor\_gpt5\_batch.py --batch

This will:

- Automatically skip the 70 already-processed files

- Process the remaining 242 files

- Save outputs to data/oe\_final\_outputs/

- Create a batch summary with timestamps

Alternative: Process with Progress Monitoring

nohup python3 tools/extractor\_gpt5\_batch.py --batch > batch\_process.log 2>&1 &

Then monitor with:

tail -f batch\_process.log

Estimated Processing Time

- ~242 files × 30-45 seconds per file = 2-3 hours

- Cost estimate: ~$12-24 (depending on document lengths)

The script automatically handles:

- Skipping already-processed files

- Rate limiting and retries

- Error recovery

- Progress tracking with [X/242] counter

python tools/extractor\_gpt5\_batch.py --batch