Split and Gather Example: Analyzing Trump Immunity Case

This example demonstrates how to use the Split and Gather operations in DocETL to process and analyze a large legal document, specifically the government's motion for immunity determinations in the case against former President Donald Trump. You can download the dataset we'll be using here. This dataset contains a single document.

Problem Statement

We want to analyze a lengthy legal document to identify all people involved in the Trump vs. United States case regarding presidential immunity. The document is too long to process in a single operation, so we need to split it into manageable chunks and then gather context to ensure each chunk can be analyzed effectively.

Chunking Strategy

When dealing with long documents, it's often necessary to break them down into smaller, manageable pieces. This is where the Split and Gather operations come in handy:

1. **Split Operation**: This divides the document into smaller chunks based on token count or delimiters. For legal documents, using a token count method is often preferable to ensure consistent chunk sizes.
2. **Gather Operation**: After splitting, we use the Gather operation to add context to each chunk. This operation can include content from surrounding chunks, as well as document-level metadata and headers, ensuring that each piece maintains necessary context for accurate analysis.

Pipeline Overview

Our pipeline will follow these steps:

1. Extract metadata from the full document
2. Split the document into chunks
3. Extract headers from each chunk
4. Gather context for each chunk
5. Analyze each chunk to identify people and their involvements in the case
6. Reduce the results to compile a comprehensive list of people and their roles

Example Pipeline and Output

Here's a breakdown of the pipeline defined in trump-immunity_opt.yaml:

1. **Dataset Definition**: We define a dataset (json file) with a single document.
2. **Metadata Extraction**: We define a map operation to extract any document-level metadata that we want to pass to each chunk being processed.
3. **Split Operation**: The document is split into chunks using the following configuration:

   ```
   - name: split_find_people_and_involvements
     type: split
     method: token_count
     method_kwargs:
       num_tokens: 3993
     split_key: extracted_text
   ```

   This operation splits the document into chunks of approximately 3993 tokens each. This size is chosen to balance between maintaining context and staying within model token limits. `split_key` should be the field in the document that contains the text to split.
4. **Header Extraction**: We define a map operation to extract headers from each chunk. Then, when rendering each chunk, we can also render the headers in levels above the headers in the chunk--ensuring that we can maintain hierarchical context, even when the headers are in other chunks.
5. **Gather Operation**: Context is gathered for each chunk using the following configuration:

   ```
   - name: gather_extracted_text_find_people_and_involvements
     type: gather
     content_key: extracted_text_chunk ①
     doc_header_key: headers ②
   ```

```
    doc_id_key: split_find_people_and_involvements_id ③
    order_key: split_find_people_and_involvements_chunk_num ④
    peripheral_chunks:
      next:
        head:
          count: 1
      previous:
        tail:
          count: 1
```

① The field containing the chunk content; the split_key with "_chunk" appended. Automatically exists as a result of the split operation. **This is required.**

② The field containing the extracted headers for each chunk. Only exists if you have a header extraction map operation. **This can be omitted if you don't have headers extracted for each chunk.**

③ The unique identifier for each document; the split operation name with "_id" appended. Automatically exists as a result of the split operation. **This is required.**

④ The field indicating the order of chunks; the split operation name with "_chunk_num" appended. Automatically exists as a result of the split operation. **This is required.**

This operation gathers context for each chunk, including the previous chunk, the current chunk, and the next chunk. We also render the headers populated by the previous operation.

6. **Chunk Analysis**: We define a map operation to analyze each chunk.
7. **Result Reduction**: We define a reduce operation to reduce the results of the map operation (applied to each chunk) to a single list of people and their involvements in the case.

Here is the full pipeline configuration, with the split and gather operations highlighted. Assuming the sample dataset looks like this:

```
[
  {
    "pdf_url":
"https://storage.courtlistener.com/recap/gov.uscourts.dcd.258148/gov.uscourts.
dcd.258148.252.0.pdf"
  }
]
```

Full Pipeline Configuration

```yaml
datasets:
  legal_doc:
    type: file
    path: /path/to/your/dataset.json
    parsing: ①
      - function: azure_di_read
        input_key: pdf_url
        output_key: extracted_text
        function_kwargs:
          use_url: true
          include_line_numbers: true

default_model: gpt-4o-mini

system_prompt:
  dataset_description: the Trump vs. United States case
  persona: a legal analyst

operations:
  - name: extract_metadata_find_people_and_involvements
    type: map
    model: gpt-4o-mini
    prompt: |
      Given the document excerpt: {{ input.extracted_text }}
      Extract all the people mentioned and summarize their involvements
in the case described.
    output:
      schema:
        metadata: str

  - name: split_find_people_and_involvements
    type: split
    method: token_count
    method_kwargs:
      num_tokens: 3993
    split_key: extracted_text

  - name: header_extraction_extracted_text_find_people_and_involvements
    type: map
    model: gpt-4o-mini
    output:
      schema:
        headers: "list[{header: string, level: integer}]"
    prompt: |
      Analyze the following chunk of a document and extract any headers
you see.

      { input.extracted_text_chunk }

      Examples of headers and their levels based on the document
structure:
        - "GOVERNMENT'S MOTION FOR IMMUNITY DETERMINATIONS" (level 1)
        - "Legal Framework" (level 1)
        - "Section I" (level 2)
        - "Section II" (level 2)
        - "Section III" (level 2)
        - "A. Formation of the Conspiracies" (level 3)
      - "B. The Defendant Knew that His Claims of Outcome-Determinative
Fraud Were False" (level 3)
      - "1. Arizona" (level 4)
      - "2. Georgia" (level 4)

  - name: gather_extracted_text_find_people_and_involvements
    type: gather
    content_key: extracted_text_chunk
    doc_header_key: headers
    doc_id_key: split_find_people_and_involvements_id
    order_key: split_find_people_and_involvements_chunk_num
    peripheral_chunks:
      next:
        head:
          count: 1
      previous:
        tail:
          count: 1

  - name: submap_find_people_and_involvements
    type: map
    model: gpt-4o-mini
    output:
      schema:
        people_and_involvements: list[str]
    prompt: |
      Given the document excerpt: {{ input.extracted_text_chunk_rendered
}}
      Extract all the people mentioned and summarize their involvements
in the case described. Only process the main chunk.

  - name: subreduce_find_people_and_involvements
    type: reduce
    model: gpt-4o-mini
    associative: true
    pass_through: true
    synthesize_resolve: false
    output:
      schema:
        people_and_involvements: list[str]
    reduce_key:
      - split_find_people_and_involvements_id
    prompt: |
      Given the following extracted information about individuals
involved in the case, compile a comprehensive list of people and their
specific involvements in the case:

      {% for chunk in inputs %}
      {% for involvement in chunk.people_and_involvements %}
      - {{ involvement }}
      {% endfor %}
      {% endfor %}

      Make sure to include all the people and their involvements. If a
person has multiple involvements, group them together.

pipeline:
  steps:
    - name: analyze_document
      input: legal_doc
      operations:
        - extract_metadata_find_people_and_involvements
        - split_find_people_and_involvements
        - header_extraction_extracted_text_find_people_and_involvements
        - gather_extracted_text_find_people_and_involvements
        - submap_find_people_and_involvements
        - subreduce_find_people_and_involvements

  output:
    type: file
    path: /path/to/your/output/people_and_involvements.json
    intermediate_dir: /path/to/your/intermediates
```

① This is an example parsing function, as explained in the Parsing docs. You can define your own parsing function to extract the text you want to split, or just have the text be directly in the json file.

Running the pipeline with `docetl run pipeline.yaml` will execute the pipeline and save the output to the path specified in the output section. It cost $0.05 and took 23.8 seconds with gpt-4o-mini.

Here's a table with one column listing all the people mentioned in the case and their involvements:

Final Output

People Involved in the Case and Their Involvements
DONALD J. TRUMP: Defendant accused of orchestrating a criminal scheme to overturn the 2020 presidential election results through deceit and collaboration with private co-conspirators; charged with leading conspiracies to overturn the 2020 presidential election; made numerous claims of election fraud and pressured officials to find votes to overturn the election results; incited a crowd to march to the Capitol; communicated with various officials regarding election outcomes; exerted political pressure on Vice President Pence; publicly attacked fellow party members for not supporting his claims; involved in spreading false claims about the election, including through Twitter; pressured state legislatures to take unlawful actions regarding electors; influenced campaign decisions and narrative regarding the election results; called for action to overturn the certified results and demanded compliance from officials; worked with co-conspirators on efforts to promote fraudulent elector plans and led actions that culminated in the Capitol riot.
MICHAEL R. PENCE: Vice President at the time, pressured by Trump to obstruct Congress's certification of the election; informed Trump there was no evidence of significant fraud; encouraged Trump to accept election results; involved in discussions with Trump regarding election challenges and strategies; publicly asserted his constitutional limitations in the face of Trump's pressure; became the target of attacks from Trump and the Capitol rioters; sought to distance himself from Trump's efforts to overturn the election.
CC1: Private attorney who Trump enlisted to falsely claim victory and perpetuate fraud allegations; participated in efforts to influence political actions in targeted states; suggested the defendant declare victory despite ongoing counting; actively involved in making false fraud claims regarding the election; pressured state officials; spread false claims about election irregularities and raised threats against election workers; coordinated fraudulent elector meetings and misrepresented legal bases.
CC2: Mentioned as a private co-conspirator involved in the efforts to invalidate election results; proposed illegal strategies to influence the election certification; urged others to decertify legitimate electors; involved in discussions influencing state officials; pressured Mike Pence to act against certification; experienced disappointment with Pence's rejection of proposed strategies; presented unlawful plans to key figures.
CC3: Another private co-conspirator involved in scheming to undermine legitimate vote counts; promoted false claims during public hearings and made remarks

People Involved in the Case and Their Involvements
inciting fraud allegations; encouraged fraudulent election lawsuits and made claims about voting machines; pressured other officials regarding claims of election fraud.
CC5: Private political operative who collaborated in the conspiracy; worked on coordinating actions related to the fraudulent elector plan; engaged in text discussions regarding the electors and strategized about the fraud claims.
CC6: Private political advisor providing strategic guidance to Trump's re-election efforts; involved in communications with campaign staff regarding the electoral vote processes.
P1: Private political advisor who assisted with Trump's re-election campaign; advocated declaring victory before final counts; maintained a podcast spreading false claims about the election.
P2: Trump's Campaign Manager, providing campaign direction during the election aftermath; informed the defendant regarding false claims related to state actions.
P3: Deputy Campaign Manager, involved in assessing election outcomes; coordinated with team members discussing legal strategies post-election; marked by frequent contact with Trump regarding campaign operations.
P4: Senior Campaign Advisor, part of the team advising Trump on election outcome communication; expressed skepticism about allegations of fraud; contradicted Trump's claims about deceased voters in Georgia.
P5: Campaign operative and co-conspirator, instructed to create chaos during vote counting and incited unrest at polling places; engaged in discussions about the elector plan.
P6: Private citizen campaign advisor who provided early warnings regarding the election outcome; engaged in discussions about the validity of allegations.
P7: White House staffer and campaign volunteer who advised Trump on potential election challenges and outcomes; acted as a conduit between Trump and various officials; communicated political advice relevant to the election.
P8: Staff member of Pence, who communicated about the electoral process and advised against Trump's unlawful plans; was involved in discussions of political strategy surrounding election results.

People Involved in the Case and Their Involvements
P9: White House staffer who became a link between Trump and campaign efforts regarding fraud claims; provided truthful assessments of the situation; facilitated communications during post-election fraud discussions.
P12: Attended non-official legislative hearings; involved in spreading disinformation about election irregularities.
P15: Assistant to the President who overheard Trump's private comments about fighting to remain in power after the 2020 election; involved in discussions about various election-related strategies.
P16: Governor of Arizona; received calls from Trump regarding election fraud claims and the count in Arizona.
P18: Speaker of the Arizona State House contacted as part of efforts to challenge election outcomes; also expressed reservations about Trump's strategies.
P21: Chief of Staff who exchanged communications about the fraudulent allegations; facilitated discussions and logistics during meetings.
P22: Campaign attorney who verified that claims about deceased voters were false; participated in discussions around the integrity of the election results.
P26: Georgia Attorney General contacted regarding fraud claims; openly stated there was no substantive evidence to support fraud allegations; discussed Texas v. Pennsylvania lawsuit with Trump.
P33: Georgia Secretary of State; defended election integrity publicly; stated rumors of election fraud were false; involved in discussions about the impact of fraudulent elector claims in Georgia.
P39: RNC Chairwoman; advised against lobbying with state legislators; coordinated with Trump on fraudulent elector efforts; refused to promote inaccurate reports regarding election fraud.
P47: Philadelphia City Commissioner; stated there was no evidence of widespread fraud; targeted by Trump for criticism after his public statements.

People Involved in the Case and Their Involvements
P52: Attorney General who publicly stated that there was no evidence of fraud that would affect election results; faced pressure from Trump's narrative.
P50: CISA Director; publicly declared the election secure; faced backlash after contradicting Trump's claims about election fraud.
P53: Various Republican U.S. Senators participated in rallies organized by Trump; linked to his campaign efforts regarding the election process.
P54: Campaign staff member involved in strategizing about elector votes; discussed procedures and expectations surrounding election tasks and claims.
P57: Former U.S. Representative who opted out of the fraudulent elector plan in Pennsylvania; cited legal concerns about the actions being proposed.
P58: A staff member of Pence involved in communications directing Pence regarding official duties, managing conversations surrounding election processes.
P59: Community organizers who were engaged in discussions relating to Trump's electoral undertakings.
P60: Individual responses to Trump's directives aimed at influencing ongoing election outcomes and legislative actions.

Optional: Compiling a Pipeline into a Split-Gather Pipeline

You can also compile a pipeline into a split-gather pipeline using the `docetl build` command. Say we had a much simpler pipeline for the same document analysis task as above, with just one map operation to extract people and their involvements.

```
default_model: gpt-4o-mini

datasets:
  legal_doc: ①
    path: /path/to/dataset.json
    type: file
    parsing: ②
      - input_key: pdf_url
        function: azure_di_read
        output_key: extracted_text
        function_kwargs:
          use_url: true
          include_line_numbers: true

operations:
  - name: find_people_and_involvements
    type: map
    optimize: true
    prompt: |
      Given this document, extract all the people and their involvements in
the case described by the document.

      {{ input.extracted_text }}

      Return a list of people and their involvements in the case.
    output:
      schema:
        people_and_involvements: list[str]

pipeline:
  steps:
    - name: analyze_document
      input: legal_doc
      operations:
        - find_people_and_involvements

  output:
    type: file
    path: "/path/to/output/people_and_involvements.json"
```

1. This is an example parsing function, as explained in the Parsing docs. You can define your own parsing function to extract the text you want to split, or just have the text be directly in the json file. If you want the text directly in the json file, you can have your json be a list of objects with a single field "extracted_text".
2. You can remove this parsing section if you don't need to parse the document (i.e., if the text is already in the json file in the "extracted_text" field in the object).

In the pipeline above, we don't have any split or gather operations. Running `docetl build pipeline.yaml [--model=gpt-4o-mini]` will output a new pipeline_opt.yaml file with the split and gather operations highlighted--like we had defined in the previous example. Note that this cost us $20 to compile, since we tried a bunch of different plans...