# LLM-Driven Rule Extraction Pipeline for Knowledge Graph

## Overview

Goal: Convert an unstructured 544-page PDF into a clean, structured JSON list of Rules and Clarifications, ready for linking to concepts and graph population.

## Step 1: Extract Text Chunks from the PDF

Use PyMuPDF to extract overlapping text blocks from the PDF. Avoid trying to parse rule structure here—just chunk raw pages.

Input: Full Rules of Golf PDF

Process: Extract 3–5 pages per chunk with 1-page overlap

Output: List of raw text chunks

## Step 2: Run LLM Extraction on Each Chunk

Feed each chunk to GPT-4/Claude with this prompt:

Prompt:

"Extract all complete golf rules from the text below. Each rule must include:

- rule\_id (e.g. '14.3b(4)')

- title (if available)

- full\_text

Only include fully-contained rules. If a rule seems to start but is incomplete, do not include it.

Return your output in this JSON format: [...]"

Output: Structured JSON list of rules per chunk

## Step 3: Validate and Stitch Extracted Rules

Load all JSON outputs, deduplicate based on rule\_id (keeping longest version), and flag short rules for manual review.

Output: Master rule set JSON file.

## Step 4: Repeat for Clarifications

Use a similar LLM prompt but tailored to Q&A style clarifications, extracting clarification\_id, text, and referenced rules.

## Step 5: Combine and Clean Master Dataset

Merge clean outputs into two files:

- all\_rules.json

- all\_clarifications.json

Optionally run another LLM pass to normalize formatting.

## Step 6 (Optional): Concept Extraction

Feed rule text to LLM to extract concepts:

Output: [{ rule\_id: '16.2', concepts: ['Dangerous Animal Condition', 'Free Relief'] }]

Ready for graph loading.

## Chunking Script (PyMuPDF)

```

import fitz

def extract\_pdf\_chunks(pdf\_path, pages\_per\_chunk=5, overlap=1, output\_dir="./chunks"):

...

```

## Stitching Script

```

import json

def load\_all\_rules(json\_dir="./llm\_outputs"):

...

```