### Legal Document Review Workflow (Expanded with Direct Quote Extraction)

#### 0. Create Experts from Document

\* \*\*Input\*\*: Source document

\* \*\*Objective\*\*: Generate a list of experts relevant to the document content

\* \*\*Steps\*\*:

\* Parse the document to identify key themes, terms, and concepts (e.g., contract clauses, obligations, technical specifications).

\* Match each theme/topic to domain-specific experts (e.g., legal, compliance, engineering, finance).

\* For each match, define an expert profile including:

\* Domain

\* Expertise area

\* Relevance to document sections

\* Link each expert profile to the specific document segments they should review.

\* \*\*Output\*\*: List of expert profiles with assigned document segments

---

#### 1. Iterate Through Experts

\* \*\*Input\*\*: Legal document + expert list

\* \*\*Objective\*\*: Collect domain-specific reviews tied to direct evidence

\* \*\*Steps\*\*:

\* For each expert:

\* Provide the relevant document sections (not full document unless required)

\* Explicitly request:

\* Detailed observations

\* At least one \*\*direct quote\*\* from the document to support each observation

\* Store each review with the following structure:

```json

{

"expert": "name",

"domain": "legal/technical/etc.",

"review": [

{

"observation": "Issue or point raised",

"quote": "Exact quote from document",

"location": "Section or page number"

},

...

]

}

```

\* \*\*Output\*\*: Structured list of expert reviews, each containing direct quotes

---

#### 2. Detect the Experts' Responses Language

\* \*\*Input\*\*: Expert reviews

\* \*\*Objective\*\*: Determine the primary language for subsequent output

\* \*\*Steps\*\*:

\* Analyze language of each expert response

\* Identify the dominant/common language across reviews

\* Normalize and ensure consistency for the next phases

\* \*\*Output\*\*: Detected language (string)

---

#### 3. Analyze the Reviews

\* \*\*Input\*\*: Structured expert reviews

\* \*\*Objective\*\*: Synthesize critiques and insights from direct quotes

\* \*\*Steps\*\*:

\* Aggregate all observations

\* For each issue identified:

\* Reference the corresponding \*\*direct quote\*\*

\* Group common or recurring concerns

\* Tag each point with metadata: domain, quote, page

\* Identify contradictions or missing elements in the document

\* Detect clauses or language that may raise compliance or legal risk

\* \*\*Output\*\*: Structured, quote-supported analysis string

---

#### 4. Generate Suggestions

\* \*\*Input\*\*: Analysis output

\* \*\*Objective\*\*: Translate insights into actionable, targeted improvements

\* \*\*Steps\*\*:

\* For each issue or observation:

\* Generate a clear suggestion addressing the problem

\* Link it to the \*\*quote and location\*\* it derives from

\* Ensure recommendations are pragmatic and domain-specific

\* Express suggestions in the detected language

\* \*\*Output\*\*: List of actionable suggestions (with optional reference metadata)

---

#### 5. Build the Final Response

\* \*\*Input\*\*: Language, analysis, and suggestions

\* \*\*Objective\*\*: Return a complete structured output with guidance

\* \*\*Output JSON format\*\*:

```json

{

"language": "<detected document language>",

"analysis": "<synthesized analysis string with quote references>",

"suggestions": [

{

"suggestion": "<concise recommendation>",

"quote": "<direct quote from document>",

"location": "<section/page number>"

},

...

],

"message": "Markdown message addressing the user. If user info is present, personalize guidance; otherwise instruct them to review the suggestions carefully."

}

```

---