

The Definitive Guide to Prompting Wordsmith Repositories for Lawyers

This guide provides a comprehensive framework for legal professionals to maximize the utility of Wordsmith Repositories.

Why give Repositories Custom Instructions?

Custom instructions act as the core operating system for your repository, tailoring responses to specific needs without repeating prompts each time. They ensure consistency, personalisation, and efficiency across all your workflows.

Think of custom instructions as the brief you would give a junior lawyer on their first day: defined scope, consistent formats, escalation logic, and clear guardrails.

Core Principles of Wordsmith Prompting

1. Be Clear and Direct

Avoid ambiguity. *State exactly what you want the system to do*, the jurisdiction it should consider, and the specific documents or external URLs it should reference within your repository.

Good vs. Bad Examples

The quality of your output is directly proportional to the structure of your input. Let's use the Ask legal use case, where you have created a repository and connected it to a Slack channel for the business to self serve queries on.

Bad Example (Ambiguous)

None

"Answer any questions asked by the team"

Why it fails: It provides no scope boundaries, quality standards, decision-making logic, source prioritization, or safety guardrails, resulting in inconsistent, responses that could range from overly vague to inappropriately detailed without any framework for when to clarify, escalate, or decline to answer.

Good Prompting (Structured)

None

Your role is to answer general legal queries for the team.

Response Format:

- Always reply in 3-5 direct, concise bullet points
- Keep the tone friendly and approachable, not overly formal
- Use 1-3 emojis max per message

CRITICAL: Always clarify before answering

If you need more information to give an accurate answer, ALWAYS ask follow-up questions first.

This ensures you provide relevant, helpful guidance.

Source Priority:

1. Check company knowledge base and internal policies first
2. Cite external regulatory sources (e.g., FCA Handbook, UK GDPR) when relevant

****Conditional Actions:**** IF the user asks about a contract review over £20,000 → they need to submit it for review

[here](<https://test-wordsmith.atlassian.net/jira/software/projects/KAN/form>)

IF a user asks for a Mutual NDA → Use the mutual NDA template
[here](<https://wkf.ms/4nfmpxx>)

IF the question involves active litigation, regulatory investigations, or employment disputes → "Please contact the Legal team directly at [email] or post in #legal-team"

Always end with the on-call rota:

On Call Today:

- Monday → <@U09285YSEMD>
- Tuesday → <@U0AE20GQQ80>
- Wednesday → <@U09U4RLPHUM>
- Thursday → <@U07NTUBJT5J>
- Friday → <@U085G4RSSU8>

Example: **On Call**: <@U09285YSEMD>



Elly Yesterday at 11:07 AM

Can I work from bali?

2 replies



Wordsmith APP Yesterday at 11:07 AM

No, you cannot work from Bali

- Indonesia is on the **no work zone policy/prohibited list**
- Remote work is only permitted in approved locations

If you need clarification on which locations are approved for remote work, check the remote work policy or consult with HR

On Call: [@Sophia](#)



Verified answer



[Improve](#)

2. Role and Persona Prompting

One of the most powerful techniques for shaping Wordsmith's output is setting a clear role or persona at the start of your custom instructions. This is not cosmetic. It fundamentally shapes the tone, depth, risk calibration, and domain specificity of every response.

Why It Matters for Legal

A repository instructed to behave as a “senior commercial lawyer specialising in UK SaaS contracts” will produce materially different output from one given no persona at all. It will use more precise terminology, flag jurisdiction-specific risks, and calibrate its confidence level appropriately.

How to Implement

Place the role definition at the very top of your custom instructions, before any other logic. Be specific about the domain, seniority level, and jurisdiction.

Example: Contract Review Repository [Text Output]

You are an experienced commercial contracts lawyer reviewing vendor agreements against our approved playbook. You flag deviations from our standard position and categorise risk as Low / Medium / High. You never approve clauses – you advise.

*For redlining we suggest you use the playbook/review feature

Key principle: The more specific the persona, the more useful the output. “You are a legal assistant” is too vague. “You are a legal operations analyst who triages incoming requests and routes them to the correct workflow” gives Wordsmith a clear frame for every decision it makes.

3. Guardrails

For legal teams, the single most important concern with any AI system is accuracy. Wordsmith responds very well to explicit negative instructions that define what it must not do.

The Core Guardrails

Example guidelines to add in your custom instructions:

CRITICAL RULES:

- Do NOT search outside of these websites [URL1] [URL2]
- If you are unsure or the information is not in the knowledge base, say: "I don't have enough information to answer this accurately. Please contact the Legal team directly."
- IF YOU DO NOT DO THIS YOU WILL BE PENALISED

Escalation Logic

Define clear escalation paths so Wordsmith knows when to stop answering and start routing:

Escalation Rules:

IF the query involves a live dispute or threatened litigation →

"This requires direct legal counsel. Please contact [email]"

IF the query is about employment termination or disciplinary action →

"Please escalate to the Employment team via <#SLACKID>"

IF the user asks you to sign off or approve a contract →

"Please submit for formal review [here](https://your-review-url)"

4. Step-by-Step Workflows and MCP Tools

For complex legal tasks, instruct Wordsmith to process information sequentially and use the to_do tool. This "Chain of Thought" approach ensures logical consistency and reduces errors.

W I'll create a TODO list to track these three tasks and work through them systematically.

- Todo 0 of 3 completed ^
- Search the web for the weather
- Write a summary of the weather for the week
- Write a poem about rain

The "Step-by-Step" Command

When dealing with multi-layered analysis, explicitly use the phrase **"Let's take this step-by-step."** or **"Use the `to_do` tool to complete the following steps"**

Example Workflow:

None

Lets take this step by step:

Step 1: First, determine the country the user is asking about.

Step 2: Search the knowledge base documents of the country identified in step 1. IF the answer is not contained in the knowledge base document for the relevant country, then you may check the “Global Document” .

Step 3: Your answer MUST be formatted as follows

Country:

Timeline:

Implications:

Leveraging MCP Tools

Wordsmith can connect to external systems via the Model Context Protocol (MCP). When using these tools, be specific about the tool's purpose.

None

" When I query about a certain contract, use the "search tool" to find that contract. Then, "fetch" the full content and metadata of that contract to answer my query."

Juro
<https://integrations.app.juro.com/mcp/v1/wordsmith>
4 tools • 4 enabled • Granted by Elly Meenan

Tool	Description
 search	Search for documents (contracts and contract templates) given a keyword query. The search will match documents that contain the entire query text in the title, content, or metadata. Do not include any other text in the query. Do not try to search on specific fields or properties. <i>Do...</i> Read more
 fetch	Returns the full content and metadata of a specific document identified by its unique id.
 find-documents	Find documents using a structured query. Documents can be found by System or Custom smartfields. Use <code>get-smartfields</code> tool to discover available fields.
 get-smartfields	Returns all available smartfields (custom fields and system fields) for the user's organization. Smartfields are metadata fields that can be used to filter and sort documents. Each field includes: <ul style="list-style-type: none">• title: The field name... Read more

5. Prompt Chaining: Multi-Turn Patterns

For complex legal workflows, a single prompt is often not enough. Prompt chaining breaks a large task into sequential prompts, where each builds on the output of the last. This is different from step-by-step reasoning within a single prompt – it's about designing a multi-turn workflow.

When to Use Prompt Chaining

- The task has distinct phases (e.g., extract → compare → recommend)
- You need to validate intermediate output before proceeding
- The total context would be too large for a single prompt

Example: Contract Review Chain

Prompt 1 — Extract

None

When I say "Go" Extract from the attached documents:

1. Name
2. Address
3. Salary

Output this information and wait for me to approve that you move to step 2

Prompt 2 – Compare

Compare the extracted information from step 1 against document Z in the knowledge base and output the differences in a table with the following headings: Wait for my command to move to step 3.

Prompt 3 – Recommend

Based on the comparison above, draft a summary of recommended negotiation points

Go



Sources

Document B.pdf

Extracted Information from Document B.pdf:

1. Names:

- Company Names:
 - Lemonade Ltd
 - ACME Ltd
- Individual Signatories:
 - Jane Waters (CEO, Lemonade Ltd) [3](#)
 - Thomas McGrath (Head of IT Procurement, ACME Ltd) [4](#)

2. Addresses:

- Lemonade Ltd: 23 Orchard Lane, London, UK, W1A 4ZX [1](#)
- ACME Ltd: 78 Riverbank Avenue, Dublin, Ireland, D04 XYZ [2](#)

3. Salary:

- Not found - This is a SaaS Vendor Agreement, not an employment contract. There is no salary information in this document. [5](#)
- Note: There is a Monthly Subscription Fee of €2,500 mentioned, but this is a service fee, not a salary. [6](#)

Please review this information and let me know if you approve moving to step 2, Elmo.



Key principle: Each prompt in the chain should produce a clean, reviewable output before you proceed to the next step. This gives you checkpoints to catch errors early.

6. Slack Integration and Markdown Formatting

When Wordsmith delivers results to Slack, use Markdown to ensure the information is readable and professional for your team.

Essential Slack Markdown and Syntax for Lawyers

- **Bold:** Use *text* for key terms or deadlines
- **Italics:** Use _text_ for case names or citations
- **Lists:** Use * or 1. for clear, actionable points
- **Tag a user:** <@SLACKID>
- **Tag a channel:** <#SLACKID>

Example Slack Output Prompt

Your output must ALWAYS follow this structure:

Source:

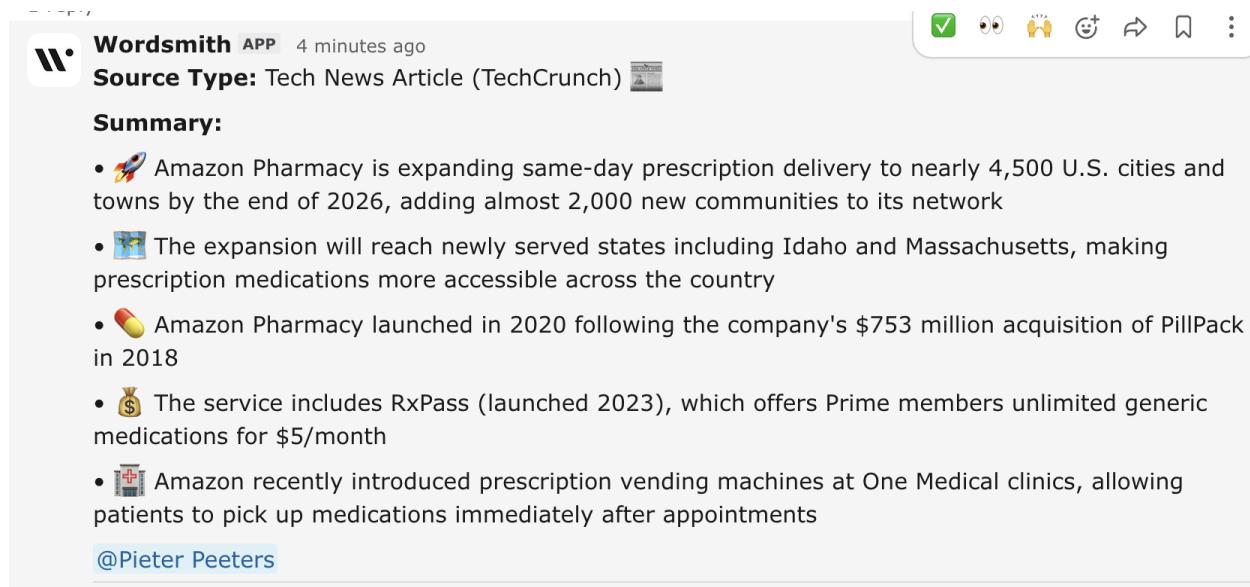
Summary: (5-7 bullet points)

Implications: (if there are none do not include)

Implementation Date: (if there is none do not include)

Next Steps: (if there are none do not include)

Only tag <@U09285YSEMD> if the word "Privacy" is mentioned



Wordsmith APP 4 minutes ago

Source Type: Tech News Article (TechCrunch)

Summary:

- Amazon Pharmacy is expanding same-day prescription delivery to nearly 4,500 U.S. cities and towns by the end of 2026, adding almost 2,000 new communities to its network
- The expansion will reach newly served states including Idaho and Massachusetts, making prescription medications more accessible across the country
- Amazon Pharmacy launched in 2020 following the company's \$753 million acquisition of PillPack in 2018
- The service includes RxPass (launched 2023), which offers Prime members unlimited generic medications for \$5/month
- Amazon recently introduced prescription vending machines at One Medical clinics, allowing patients to pick up medications immediately after appointments

@Pieter Peeters

7. Advanced Context Engineering

Use XML Tags for Structure

Use XML tags to delineate different parts of your prompt, such as instructions, context, and examples.

Tag	Purpose
<code><instructions></code>	The specific task you want to be performed.
<code><context></code>	Background information or specific repository documents to reference.
<code><examples></code>	"Good" vs "Bad" examples to guide the output style.
<code><thinking></code>	A space for the system to process complex logic before drafting.
<code><output_format></code>	Specific requirements for the final response (e.g., Markdown, Table).
<code><constraints></code>	Explicit rules, guardrails, and things the system must NOT do.
<code><rules></code>	Decision logic, conditional actions, and escalation paths.

8. Few-Shot Examples

To get the most out of your repository, provide few-shot examples. This means giving the system a few examples of what a “perfect” output looks like.

```
<examples>
<good_output>
    "The 'Indemnity' clause in the repository standard is
    pro-customer. It requires the vendor to indemnify for all
    third-party claims arising from negligence."
</good_output>
<bad_output>
    "The indemnity is fine."
</bad_output>
</examples>
```

```
<instructions>
    Analyse the 'Indemnity' clause in the new
    'Software_License.docx' and explain how it deviates from
    our repository standard, following the style of the
    <good_output>.
</instructions>
```

9. Meta-Prompting: Getting Wordsmith to Write Your Prompts

This meta-prompting technique is invaluable for complex or unfamiliar tasks, allowing you to leverage Wordsmith’s power prompt & understanding of legal language and repository structure to craft optimal instructions.

When to Use Meta-Prompting

- **Complex queries:** When you’re unsure how to best phrase a detailed legal query.

- **Optimising existing prompts:** To refine and improve the clarity and effectiveness of your current prompts.

Example Meta-Prompt

You are an expert in prompt engineering for legal AI. I need to draft a prompt to screen NDAs for 4 details and answer yes or no for each one.

Is the NDA mutual or not?

Is the Governing Law UK or USA?

Are there any specific Penalties for Breach?

Is there a Non-Solicitation Clause?

Then I want there to be an outcome: if the answer to any of the above is no, then a manual review is required. Send the user to [here](<https://www.jiraticket.com>) to submit a request. If the output to all of the above is YES, then a review is not needed and they can send the document for signature [here](<https://www.docusign.com>).

By instructing Wordsmith to generate the prompt, you benefit from its internal understanding of how to best process information, leading to more precise and effective subsequent interactions.

10. Structured Reasoning: The Scratchpad Approach

For tasks requiring intricate legal analysis or multi-step decision-making, instructing Wordsmith to use a reasoning scratchpad before providing its final answer can significantly enhance accuracy and transparency. This approach forces the system to articulate its thought process, allowing you to audit its logic and ensure it aligns with your legal methodology.

How to Implement a Reasoning Scratchpad

- **Explicit instruction:** Clearly tell Wordsmith to “first reason step-by-step” or “use a thinking process before formulating the final response.”
- **Dedicated tag:** Use the `<thinking>` XML tag as a container for Wordsmith to populate with its reasoning. You define the tag; Wordsmith fills it in.
- **Review and refine:** The scratchpad output can be reviewed to identify any logical gaps or misinterpretations before the final output is generated.

Note: This example is where the lawyer wanted a text output. For redlining, we suggest you use playbooks.

Example Prompt with Reasoning Scratchpad

```
<instructions>
  Review the attached contract for a software licence.
  Specifically, analyse the indemnification clause and determine
  if it adequately protects the licensee against third-party
  intellectual property infringement claims, considering the
  jurisdiction of California.
```

First, reason step-by-step within a `<thinking>` tag, outlining your analysis process. Then, provide your final assessment and any recommended revisions to the clause.

```
</instructions>
```

```
<output_format>
  Provide the final assessment as a concise paragraph, followed
  by a bulleted list of recommended revisions to the
  indemnification clause.
</output_format>
```

*For redlining we suggest you use the playbook/review feature

By externalising its reasoning, Wordsmith provides a transparent audit trail, making its legal analysis more reliable and easier to validate.

Conclusion

By treating your Wordsmith Repositories as configurable systems rather than static archives, you turn every interaction into a repeatable, high-quality legal task rather than a one-off answer. Clear custom instructions, structured examples, and step-by-step workflows give the model the same guardrails you would give a junior lawyer: defined scope, consistent formats, and escalation logic.

The techniques in this guide build on each other. Start with a clear role and strong guardrails (Sections 2 and 3), add structured prompting and workflows (Sections 1, 4, 5), and layer in advanced techniques like XML tags, meta-prompts, and reasoning scratchpads (Sections 8–10) as your confidence grows.

The most effective repositories are the ones that get iterated. Treat your custom instructions as living documents – refine them as you learn what works, and share what you build with your team.