From Prompt to Agentic AI: The PASTAS Checklist for Context Engineering

Leo S. Lo

abstract: This article introduces PASTAS - Purpose and Audience, Authority, Structure and Style, Tools and Access, Accountability and Safeguards, Signals and Review - as a structured approach to context engineering for agentic artificial intelligence (AI). Moving beyond one-shot prompts, PASTAS offers a repeatable blueprint for designing multi-step, library-aligned AI workflows that integrate trusted sources, enforce ethical safeguards, and produce decision-ready outputs. Using a policy research support agent as an example, the author illustrates how each component of the model guides tool selection, workflow logic, and review cycles. The framework embeds librarian values in agent design, ensuring accuracy, transparency, and human oversight remain central in AI-enabled services.

Introduction

The year 2025 has been hailed as the year of Artificial intelligence (AI) agents and according to a May 2025 survey of US company senior executives, 75 percent of the respondents say, "AI agents are already being adopted in their companies."[1] AI agents offer capabilities beyond what traditional single-prompt chatbots can achieve.

In simple terms, an AI agent is more than just a single-use question and answer bot. It is an advanced system that can reason, plan, and take actions to meet specific goals. Unlike a generic generative AI (such as a standard chatbot that responds transactionally to individual queries), an AI agent can proactively guide workflows, handle complex multi-step tasks, and integrate with various data sources and services.

This distinction is crucial for libraries: where a single prompt to a chatbot might yield a quick answer, an AI agent can follow through multiple steps of a research process

. . . where a single prompt to a chatbot might yield a quick answer, an AI agent can follow through multiple steps of a research process or routine task.

portal: Libraries and the Academy, Vol. 26, No. 3 (2026), pp. 403–415

or routine task. For instance, consider a library user seeking help with policy research. A single-prompt chatbot might return a general summary, but an AI agent designed for policy research could not only retrieve relevant policy documents, but also analyze and summarize them, compare their content, and present findings in context, all through one conversational interface.

Academic libraries widen equitable access to high-quality information and teach critical thinking and information literacy. As generative systems increasingly influence how students and faculty discover, evaluate, and synthesize sources, libraries have a responsibility to shape those interactions so they remain reliable, transparent, and educative, and not merely convenient.

Many library tasks are genuinely multi-step. Research begins with identifying authoritative texts, continues with extracting and comparing evidence, and ends with a decision-ready product for a defined audience. Single-prompt chat seldom supports that chain. Assistant-style agents can, provided they are designed with guardrails. The PASTAS framework, which stands for Purpose and Audience, Authority, Structure and Style, Tools and Access, Accountability and Safeguards, Signals and Review, is a compact way to encode library values and professional judgment before the agent runs. The policy research example used throughout this article shows how PASTAS turns ad hoc prompting into an auditable service design that keeps human oversight and critical appraisal at the center.

Beyond Single Prompts: Why Libraries Need AI Agent Workflows

Single-prompt interactions (asking a question to a tool like ChatGPT and getting one answer) have proven useful for quick answers or simple tasks. However, their limitations become apparent in more complex library use cases. Even well-crafted prompts cannot overcome certain fundamental constraints of one-shot conversations. Some key limitations of single-prompt systems include:

- Limited Context and Memory: Models like ChatGPT operate within a fixed context window - the total amount of text they can process at one time - which limits how much of a conversation or document the model can "remember" when producing an answer.[2] They cannot maintain long dialogues or handle extensive documents within one prompt, which is problematic for in-depth research queries.[3]
- Inability to Handle Complex, Multi-Step Tasks: Many research questions or library tasks require a sequence of actions, such as finding sources, extracting data, comparing findings, and then synthesizing a response. A single prompt often cannot complete such tasks in one attempt.[4] The user would need to manually break the task into pieces or use multiple prompts, creating a process that is inefficient and prone to error.
- No Tool Use or External Knowledge Retrieval: Out-of-the-box chatbots rely solely on their training data. They do not automatically search databases, check library catalogs, or use calculators when asked a question. This can lead to outdated or incorrect responses if the needed information is not already in the model's memory.[5] For example, a person asking about the latest state legislation on climate policy might receive a confident-sounding but completely fabricated answer if the system is not connected to authoritative sources.
- Risk of Hallucinations and Unverified Information: Single prompts can produce answers that sound plausible but include false information (a phenomenon known as AI hallucination).[6] Without a way to double-check or cite sources mid-task, these systems might mislead users. Researchers have found that even domain-specific AI tools can make up

false references or facts, especially when operating without access to trusted external data.
- Lack of Iterative Refinement: With a single prompt, AI gives one response. If that answer is off-target or incomplete, it's on the user to craft a new prompt to clarify or correct the course. There's no built-in mechanism for the AI to ask clarifying questions or refine its output step by step.[7]

In an academic library context, these limitations mean that while a tool like ChatGPT might handle a straightforward question such as "What is a policy brief?", it could stumble on a complex research query like "Compare federal and state policies on renewable energy incentives and provide a summary with references." Librarians are increasingly recognizing that to support advanced research queries, we need to move from ad-hoc prompting to designing workflows, essentially breaking the interaction into a series of logical steps that an AI agent can execute.

Libraries are beginning to experiment with AI-driven agents that can connect to various tools and databases to automate multi-step processes. For example, Ex Libris' recent workflow platform describes "AI agents that connect seamlessly with your library system, tools, services, and workflows" to handle "dynamic, multi-step tasks that traditionally require much manual effort."[8] Such an agent might, for instance, take a user's query, search multiple library databases, aggregate and summarize the findings, and even format a draft report. The benefit is not just efficiency but also freeing up librarians and researchers to focus on analysis and decision-making rather than repetitive search tasks. With these possibilities on the horizon, having a clear design strategy becomes critical. That is where the PASTAS framework comes in. It is designed to help library technologists and AI enthusiasts ensure that when they create an AI agent, it is purposeful, trustworthy, well-structured, well-equipped, and properly governed.

Librarians are increasingly recognizing that to support advanced research queries, we need to move from ad-hoc prompting to designing workflows, essentially breaking the interaction into a series of logical steps that an AI agent can execute.

The PASTAS Framework: A Recipe for Designing Library AI Agents

PASTAS is a mnemonic framework that stands for Purpose and Audience, Authority, Structure and Style, Tools and Access, Accountability and Safeguards, and Signals and Review. It provides a structured checklist for designing AI agents in a library setting. Rather than diving straight into building an AI workflow, library staff can use PASTAS as a planning brief, much like an architect uses a blueprint before construction. In the context of our running example (a policy research support agent), we will illustrate each component of PASTAS in turn:

- Purpose and Audience: Defining *why* the agent exists and *who* it serves.
- Authority: Ensuring the agent uses credible sources and domain expertise.
- Structure and Style: Shaping how the agent communicates and presents information.

- Tools and Access: Determining what systems and data the agent can utilize.
- Accountability and Safeguards: Establishing ethical guidelines, oversight, and fail-safes.
- Signals and Review: Monitoring the agent's performance and iterating on its design.

By addressing each of these areas, library developers can create AI agents that align with the academic library values of accuracy, trustworthiness, and user-centered service.

Where Agentic Systems Might Come From

For librarians new to agentic AI, it is helpful to understand that these systems do not need to be built from scratch. Agentic workflows can originate from several pathways. Some library vendors, such as Elsevier and Clarivate, now offer agent-enabled platforms that layer planning, retrieval, and tool use onto existing library data infrastructures. Libraries can also configure lightweight agents using no-code or low-code environments that support tool calling, retrieval steps, and structured prompting. In consortial or campus settings, IT units may host shared agent frameworks that individual departments configure for local tasks. In all cases, librarians shape the agent's behavior not by writing code but by defining its purpose, evidence sources, safeguards, and success criteria, which are the core elements captured in the PASTAS framework.

Context for Library Implementation

While the technical possibilities of agentic AI are expanding quickly, not every library will engage in the same way or at the same scale. Large, research-intensive institutions may prototype full agent workflows first, but smaller or mid-sized libraries can still participate through targeted pilots or shared services. Many early uses will rely on simple, well-defined workflows, such as policy analysis, literature scanning, or metadata cleanup, rather than broad, open-ended systems.

Partnerships and shared capacity will be key. Consortia, regional collaborations, and professional associations can pool infrastructure and expertise, so libraries do not have to build agents independently. These partnerships also provide a forum to exchange ethical guidelines and governance templates.

Content use and licensing remain evolving questions. Libraries should observe existing license terms and emphasize transparency, attribution, and responsible retrieval. As publishers and vendors experiment with AI-integrated access models, librarians' role in negotiating fair, ethical use of data will be increasingly important.

The goal is not for every library to host dozens of agents, but to adapt a small number of reusable designs for repeatable workflows that augment, rather than replace, professional judgment. In all cases, librarians' knowledge of authority, context, and ethical stewardship remains central to how AI becomes part of library practice.

Purpose and Audience

Every successful design begins with a clear purpose and a defined audience. In the context of AI agents, *purpose* refers to the specific goals or tasks the agent is intended to accomplish, and *audience* refers to the end-users (or even other systems) who will interact

with the agent. Defining these up front is paramount; doing so anchors all subsequent decisions in the design process.

For our policy research support agent, the purpose might be stated as: "Assist users in conducting policy research by retrieving relevant policy documents, extracting key points, and delivering summaries or comparisons of policy information." This is the agent's core mission. Alongside this, we clarify the audience: perhaps advanced undergraduates in public policy courses, graduate researchers, or librarians in a research services department. The audience determines the agent's scope and tone. For example, an agent meant for students might include more explanatory content and definitions, whereas one for librarians might assume more background knowledge and use technical language succinctly.

Specifying purpose and audience at the outset has several benefits. First, it helps avoid mission creep. The agent should not try to do everything for everyone. A focused agent can be optimized for its intended use case. Second, it informs how we train or prompt the AI tool. A well-crafted system prompt (the initial instructions that set the stage for the agent's behavior) will explicitly describe the agent's role and users. As one guide to AI agents notes, "The system prompt is what tells your agent what it is and what it does."[9] In practice, this means we might program the system message to say: "You are an AI research assistant specializing in public policy analysis for academic users. You help locate and summarize policy documents, and provide answers in a formal, academic tone suitable for researchers." This one sentence encapsulates both purpose and audience.

Furthermore, by defining the persona and core task (goal) up front, we set the agent's tone and focus. For example, the persona could be "a helpful, knowledgeable library research assistant" and the core task is "to help users find and understand policy information." This ensures consistency: the agent will not suddenly behave like a casual chatbot or drift into unrelated topics. Everything it does should circle back to its purpose of policy research support.

In summary, Purpose and Audience function as the North Star for design. They ensure the AI agent is user-centered, that it is built for a clear need, and they guide decisions about content depth, language level, and feature set.

Authority

The next critical component is Authority, which deals with the credibility and reliability of the information the AI agent provides. Libraries have long-standing commitments to authoritative sources and verifiable information. An AI agent must uphold those standards. There are two sides to authority in this context: the agent's sources of knowledge, and the way it conveys trustworthiness to the user.

One major weakness of many AI systems is their propensity to generate incorrect or fabricated information confidently. As mentioned earlier, large language models can hallucinate facts or references. In high-stakes domains like legal or medical research, using an unchecked AI agent can be risky because it might produce plausible-sounding but false answers. The key to mitigating this is grounding the agent in authoritative data. For our policy research agent, that means integrating it with credible sources of policy

documents: government websites, legislative databases, think-tank reports, academic journals, and so on. Rather than asking the base model to answer from its general training data (which might be outdated or incomplete on specific policy details), the agent should retrieve actual documents or statistics and base its responses on that evidence. This approach, known as retrieval-augmented generation (RAG), combines a text generation model with an external search step, allowing the agent to retrieve relevant passages from a designated document collection before composing its response.[10] Recent studies in the legal field found that using RAG (feeding the AI agent with relevant documents) can reduce hallucinations, though not eliminate them entirely.[11] The lesson: connecting AI agents to trustworthy sources is vital for authoritative answers.

Concretely, when designing the policy research agent, we would delineate which repositories or databases it can access. For example, it might be allowed to query the library's discovery layer for policy papers, use an API to search government archives (like FDsys or Congress.gov in the US), and access subscription databases that the library has licensed. The design document (our blueprint) should list these sources and ideally rank them (for example: prefer official government publications over secondary commentary for factual questions). Moreover, the agent's programming should instruct it to cite its sources whenever it provides factual assertions or statistics. In an academic setting, users will expect and appreciate citations. This practice not only boosts user trust but also allows the user to verify information independently.

Another aspect of authority is the agent's domain expertise. Even though a large language model can generate text on almost any topic, it is beneficial to tailor the agent's training or context to the domain of interest. For policy research, we might feed the agent a curated knowledge base or glossary of policy terms, ensuring it understands key concepts and jargon. Some library AI initiatives, like Clarivate's academic AI agents, are designed to leverage "curated data and workflow tools" to ensure the agent operates with high precision and aligns with academic standards.[12] In designing our agent, we could, for instance, include a curated list of authoritative think tanks or research institutes so that the agent can draw on those perspectives when needed (and avoid dubious sources like random blogs or non-reviewed opinions).

Even though a large language model can generate text on almost any topic, it is beneficial to tailor the agent's training or context to the domain of interest

To summarize, Authority in the PASTAS framework is about making the AI agent an expert, or at least an honest broker of expert information. By wiring in access to authoritative sources and enforcing transparency (through citations or disclaimers), we significantly enhance the reliability of the agent's output. For libraries, this element is perhaps the most important. It is what differentiates an AI toy from a serious research aide. In our policy agent example, authority means a user can trust that the summary of a policy is grounded in actual documents, not the AI tool's imagination.

Structure and Style

While purpose defines *what* the agent should do, Structure and Style define *how* the agent should communicate and carry out its tasks. This component is about the design of the interaction and the presentation of information.

Structure refers to the organization of the agent's workflow and responses. Even AI-driven interactions benefit from having a logical flow. For example, if a user asks our policy research agent a broad question, such as "What are the major differences between Policy A and Policy B?", the agent might structure its response in a sensible way: perhaps an introduction, followed by a point-by-point comparison, and then a brief conclusion. We can design the agent's prompting strategy to enforce this, using approaches such as instructing it to break answers into sections or bullet points if that suits the content. If a question requires multiple steps (say, first finding policies, then summarizing, then comparing), the agent's internal workflow should follow a structured sequence. Rather than doing everything in one giant leap, it might internally do step 1 (search for Policy A docs), step 2 (extract key points of Policy A), step 3 (search for Policy B, extract points), and step 4 (generate a comparative summary). Each of those steps has a structure, and the final output is structured for readability. Some tasks require multi-step prompting, breaking a complex problem into a sequence of smaller, connected prompts, so that each stage can be completed accurately before moving to the next.[13] Each step builds on the last, producing a better final answer than a single prompt could. Our design blueprint should outline these intended steps clearly so that the agent's developers (or the prompt engineer configuring it) implement a logical flow.

Style pertains to the tone, language, and formatting the agent uses when interacting with users. Given our audience in the example (students and researchers), the style might be: professional, academic tone, but still accessible. No slang or memes, certainly, but also avoiding unnecessary jargon unless explaining it. We might explicitly set style guidelines in the system prompt or instructions, for example: "Respond in complete sentences and a neutral, informative tone. Use technical terms where appropriate but define them if a general audience may not know them. Avoid first-person casual remarks; maintain an academic style." If the agent is meant to draft outputs (like a brief or report), we can specify format preferences; perhaps the agent should provide answers with headings or bullet points for clarity, or always include a short introduction if the answer is lengthy.

The Structure and Style component also involves considering the user experience. For instance, how does the agent handle follow-up questions? Does it remember context from earlier in the conversation (assuming we include a memory component)? If the user asks for clarification on something the agent said, the agent should be able to reference its previous answer without contradiction. Ensuring a consistent style across turns (the agent does not suddenly switch persona or level of formality) is part of this design aspect.

It is worth noting that many of these stylistic decisions can be pre-engineered. Developers can write prompt templates that scaffold the agent's answers or use settings of the AI model (if available) to adjust verbosity or tone. The Nielsen Norman Group, in studying generative AI interfaces, observed that users actively seek control over output format (length, tone, and so forth) often through repeated prompt iterations.[14] In our case, we might not expose all those controls to the end-user, but we as designers anticipate the need for clarity and thus can bake good structure and style into the agent from the start.

Tools and Access

An AI agent is only as powerful as the tools and data it can access. "Tools" here refers to external software, databases, APIs, or plugins the agent can use to extend its capabilities, while "Access" concerns the permissions or connectivity the agent has to various information sources. In single-prompt scenarios, the AI is essentially a closed box. It does not go out and fetch new information or perform external actions. But an AI agent can be equipped with tools that let it overcome those boundaries. For our policy research support agent, the essential tools and data access might include:

- Search and Retrieval Tools: The agent should have the ability to query search engines or library databases. This could be an integration with the library's catalog, academic databases (like EBSCOhost or JSTOR), or open web search for governmental sites. The design would specify that, when a user asks for information on a policy, the agent's first step is to *search for documents* (and we might even dictate which repositories to search first).
- Document Readers or Analyzers: Once relevant documents are found (say a PDF of a policy report or a webpage of a law), the agent needs a way to ingest that text. So, a tool that can fetch the full text and perhaps summarize or extract key sections is important. We could integrate an existing PDF parser or use the AI model itself in a secondary mode to summarize text.
- Memory/Context Management: Although not a "tool" in the traditional sense, giving the agent a memory component means it can retain information across steps. A memory allows ongoing conversation and context retention. In practice, this might be implemented by storing the retrieved facts from earlier steps so they can be referenced in later steps of the agent's reasoning.
- Domain-Specific APIs: If available, we might connect to APIs specifically useful in policy research. For example, a legislative tracking API that provides the status of a bill, or a data API for census statistics if demographic context is needed for policy impact analysis. These are optional, but, if included, they expand what the agent can do (like answering "What is the population affected by Policy X in region Y?" by actually pulling the latest demographic data).
- Internal Library Systems: The agent could also tie into local systems. For instance, if it helps librarians, it might connect to an interlibrary loan system or a library FAQ knowledge base for quick answers that combine policy content with library services information (like how to cite a policy document).

When designing Tools and Access, security and permissions must be kept in mind. The blueprint should clarify what the agent is allowed to do. For example, can it only *read* data or also *write* somewhere? In our case, likely it is mostly reading and summarizing information, not making changes to data. But if it were an agent automating tasks (like ordering books, as in some library tech examples), we would include safeguards about confirming actions (which overlaps with Accountability, discussed later).

It is useful to document tool usage in the agent's system prompt or logic. The Ex Libris guide suggests listing important tools and even the order to use them in the prompt to ensure the AI knows its toolbox.[15] For the policy agent, we might instruct: "When answering a query, first search the Policy Database (tool A), then search Government Archive (tool B) if needed, and only then draft an answer. Always prefer using these tools over guessing." We essentially teach the agent a strategy: tools first, then answer. This way, the agent becomes more deterministic and less likely to hallucinate or stray.

From a user perspective, the Tools and Access component determines how comprehensive and up-to-date the agent's answers can be. If our agent has access to the latest government data, a student's query about current policy will reflect the most recent information (something a static-trained model cannot guarantee). Also, by accessing the rich resources the library subscribes to, the agent can deliver value that generic web chatbots might miss. For instance, the agent might summarize a paywalled academic article on policy outcomes, which the student would not have been able to obtain via a free AI service.

In essence, Tools and Access is about empowering the AI agent with the means to act. Where we discussed Authority in terms of quality of information, here we ensure the agent has a pathway to that high-quality information. A well-designed library AI agent is like a skilled librarian: it knows where to look and has the keys to access those places.

Accountability and Safeguards

With great power comes great responsibility. This is where Accountability and Safeguards enter the framework. Even a well-intentioned AI agent can go awry without proper checks and ethical guidelines. This component focuses on how we ensure the agent operates within acceptable boundaries, and how we hold its performance accountable to human standards and values. Several layers of safeguards are important:

- Ethical and Policy Guidelines: The library should establish what the AI agent is allowed or not allowed to do or say. For example, the agent should refuse or safely handle requests that are beyond its scope or that ask for inappropriate content. If a user tried to get the policy research agent to do something unethical (like provide a private, sensitive document or give legal advice beyond factual reporting), the agent should have a fallback response. We might hard-code certain refusals or use content filters. Many AI systems have a base layer that prevents hate speech and the like, but in our design, we can add domain-specific rules, like, "If asked for legal advice or interpretation, politely explain that you cannot provide legal opinions and recommend consulting a professional."
- Accuracy Safeguards: While Authority covers the sourcing of information, accountability means the agent should signal uncertainty when it exists. The agent should not fabricate an answer just to appear helpful. If data is missing or the query is unclear, the best safeguard is often to ask the user for clarification or to acknowledge the limits. We can design the agent's prompts to allow for this: "If you are not sure you have sufficient information, ask a follow-up question or inform the user of what information is needed." This avoids the pitfall of confident misstatements.
- Human Oversight Mechanisms: In library workflows, we might want a human in the loop for certain actions. A compelling example comes from an AI tools workflow for library book orders. Lili Daie advises that "if you'd like more human oversight, you can add a confirmation step, telling the agent to confirm with the librarian before performing actions."[16] Translated to our context, while the policy agent might not execute transactions, there could be a mechanism that triggers a librarian review if the agent is about to send an email summary to a patron or publish a guide. At minimum, during the pilot phase of the agent, librarians might monitor its outputs regularly.
- Privacy and Data Security: Since the agent may be dealing with user queries that could be sensitive (imagine a query on policies related to a personal or controversial topic), we must ensure it handles data in compliance with privacy rules. It should not log personal details unnecessarily, and any user data it processes should be protected. In the design brief, we must note what data the agent will store and for how long, and who has access to those logs.

- Fail-safes and Abort Conditions: We should consider scenarios in which the agent might get stuck in a loop or encounter an error (like an API not responding). Safeguards include setting timeouts for tool use and defaulting to a graceful apology or error message if something fails. The agent might say, "I'm sorry, I'm having trouble accessing the policy database right now. Let me try again later or you might attempt again in a few minutes." This kind of graceful degradation is preferable to the agent just giving a wrong answer due to a half-fetched result.

Accountability also ties into the concept of transparency. The agent should, as much as possible, make it clear that it is AI and not a human librarian, and explain its actions if needed. For instance, "I will now search the government archives for the policy document…" is something it might convey (though not verbosely every time, since that could annoy users). It should certainly disclose sources as earlier discussed, which is part of being accountable for the information given.

Crucially, accountability extends to the library staff overseeing the agent. As one recent study emphasized in the legal realm, professionals still have a responsibility to supervise and verify AI outputs.[17] In our design, we plan for a librarian to periodically review the agent's answers for quality control. If the agent consistently errs on certain kinds of questions, that feedback loop should trigger a redesign or re-training (overlapping with the final framework component, Signals and Review).

In summary, Accountability and Safeguards ensure the AI agent operates as a trustworthy assistant rather than a loose cannon. By embedding rules, oversight, and ethical considerations, we protect both the user from misinformation and the library from potential misuse of the AI agent. Our policy research agent, under these guidelines, would be a cautious and principled helper: it would double-check before presenting information, it would avoid areas outside its competency (not turning into a political commentator or legal advisor), and it would always leave an avenue for human experts to intervene or provide additional input.

Signals and Review

By embedding rules, oversight, and ethical considerations, we protect both the user from misinformation and the library from potential misuse of the AI agent.

The last element of PASTAS, Signals and Review, is about what happens after the AI agent is up and running. No system should be set on autopilot indefinitely, especially not one that interacts with users and deals with complex information. This component emphasizes the importance of monitoring the agent's performance and iteratively improving it based on feedback (the "signals") and periodic reviews. Signals can take many forms:

- User Feedback: The most direct signal is feedback from users. We can incorporate a simple feedback mechanism, like a thumbs-up/thumbs-down on answers or a prompt asking, "Was this answer helpful?". Qualitative feedback (users leaving comments or librarians noting where the agent faltered) is useful for improvement. If multiple users indicate that the agent's summary of a certain policy was confusing, we know to adjust

how that content is generated (perhaps the policy was too technical, and the agent needs to simplify further for general readers).

- Usage Analytics: We should track how the agent is being used. What kinds of questions come up frequently? Where does it succeed or fail? Analytics might show, for example, that 80percent of queries are about a certain category of policy (say environmental policies), that could prompt us to ensure our sources in that area are robust. Or if we see users often abandoning the conversation after a certain point, maybe the agent's responses are too long or not hitting the mark.
- System Logs: On the backend, we will have logs of the agent's actions, for example what it searched, which tools it invoked, and so forth. These logs are useful for diagnosing problems. If an answer was incorrect, logs might show that the agent picked the wrong source or misinterpreted data. Regularly reviewing logs (with privacy in mind) is a way to catch unseen issues. For instance, the agent might be consistently failing to retrieve documents from one database due to a formatting issue, which is something only a log review would catch.

Review refers to the human-led evaluation of the agent's overall performance and adherence to guidelines. This could happen as a scheduled audit, say monthly or quarterly, where a team of librarians:

- Assess a random sample of agent interactions for quality and accuracy.
- Review any incidents (for example if the agent gave a notably incorrect answer or a user reported a problem).
- Check whether the scope of the agent is still appropriate or if adjustments are needed (maybe users are asking for capabilities we initially scoped out—should we expand the agent's training or tools? Or conversely, is the agent drifting into areas it should not, indicating we need to tighten the instructions?).
- Ensure that the content and sources the agent uses are up to date (for a policy agent, new policies come out regularly; part of review is making sure its knowledge base is refreshed. This might involve updating which sources it trusts or feeding it new data).

Another aspect of Signals and Review is continuous improvement. The agent's design is not a one-and-done affair. Just as library services evolve with user needs, the AI agent should evolve. Through iterative updates (revising the system prompt, adding new tools, refining the training data, fixing bugs), the agent gets better over time. We can treat the initial deployment as a beta, with the expectation that the PASTAS framework will guide future adjustments. For example, if signals indicate the agent struggles with very recent events, such as a policy change from the last week, we might implement an update to incorporate a news API or shorten the update cycle for its data.

Finally, as part of review, we should also revisit the Purpose and Audience periodically. Are we meeting the users' needs we set out to meet? Have those needs changed? Perhaps after some time, we realize the policy research agent could also help with a related task (like helping users find data for policy analysis, not just documents). We can consider that in the next version, but carefully, to not overload the agent without proper design adjustments to other PASTAS elements.

In sum, *Signals and Review* closes the loop in the PASTAS framework by instilling a practice of reflection and refinement. It acknowledges that AI agents in the library are not static tools, but dynamic services that require care and feeding. By listening to signals from users and the system, and by conducting thoughtful reviews, libraries ensure their AI agents continue to serve effectively and ethically. In our example, this means the

policy research support agent will only improve with time, through and learning from its interactions and librarian guidance, to become an even more invaluable resource for the academic community.

PASTAS as applied AI literacy

PASTAS operationalizes AI literacy for staff and learners.[18] Technical knowledge appears when AI assistants are grounded in retrieval, chunking rules, tool permissions, and an understanding of model behavior and its limitations. Ethical awareness is expressed through refusal policies, privacy posture, and transparent citation that make accountability visible. Critical thinking lives in success checks, rationale logs, and review scorecards that test claims against evidence and question underlying assumptions. Practical skills develop as staff configure briefs, evaluate outputs, and refine workflows through iterative learning. Societal impact is considered when Authority balances credibility with coverage and perspective, avoiding a single-voice view of complex topics and recognizing how AI shapes equity and access. Designing, running, and improving an assistant with the PASTAS framework makes the evaluative moves librarians teach, such as interrogating sources, validating evidence, and reflecting on impact, visible in the system itself.

Conclusion

PASTAS-designed agents can improve the consistency and quality of research support by relying on defined evidence sources, structured outputs, and transparent citations. When well configured, they could surface uncertainty, flag gaps, and document their reasoning, making review more efficient and accountable. By handling routine, multi-step tasks such as searching repositories, extracting passages, and summarizing findings, PASTAS-grounded agents could enable librarians to focus more fully on higher-order instructional and consultative work. Rather than diminishing critical thinking or information literacy, agentic systems have the potential to provide a stable, evidence-driven foundation that supports and extends these roles.

Libraries should engage with agentic AI because the work our communities do with information is already changing, and because embedding librarian judgment in that workflow is the surest way to protect quality, equity, and transparency. PASTAS offers a small instrument with practical leverage. Purpose and Audience remove guesswork. Authority sets credible boundaries and balances perspective. Structure and Style produce decision-ready outputs. Tools and Access keep capabilities controlled and predictable. Accountability and Safeguards protect people and institutions. Signals and Review keep the system honest over time. Librarians implementing PASTAS should begin with one workflow and one brief. It is also important to measure what matters each month in order to improve the design rather than the last prompt. Assistants built this way augment librarians' roles as educators and stewards of information, while librarians retain the work that only humans can do: framing questions, teaching critical appraisal, making judgments under uncertainty, and caring for communities.

Leo S. Lo is university librarian and dean of libraries at the University of Virginia, email: leolo@virginia.edu, ORCID: 0000-0001-5043-7575.

Notes

1. Lutz Finger, "AI Agents in 2025: What Enterprise Leaders Need to Know," *Forbes*, January 5, 2025, https://www.forbes.com/sites/lutzfinger/2025/01/05/ai-agents-in-2025-what-enterprise-leaders-need-to-know/; Dan Priest, "AI Agent Survey," Tech Effect (blog), pwc, January 2025, https://www.pwc.com/us/en/tech-effect/ai-analytics/ai-agent-survey.html.
2. Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang, "Lost in the Middle: How Language Models Use Long Contexts," *Transactions of the Association for Computational Linguistics* 12 (2024): 157–173, https://doi.org/10.1162/tacl_a_00638.
3. Ibid.
4. Tushar Khot, Harsh Trivedi, Matthew Finlayson, Kyle Richardson, and Ashish Sabharwal, "Decomposed Prompting: A Modular Approach for Solving Complex Tasks," Preprint, submitted October 5, 2022, https://arxiv.org/abs/2210.02406.
5. Patrick Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems* 33 (2020): 9459-9474, https://arxiv.org/abs/2005.11401.
6. Ziwei Ji et al., "Survey of Hallucination in Natural Language Generation," Preprint, submitted February 8, 2022, https://arxiv.org/abs/2202.03629; Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S. M. Towhidul Islam Tonmoy, Aman Chadha, Amit P. Sheth, and Amitava Das, "The Troubling Emergence of Hallucination in Large Language Models: An Extensive Definition, Quantification, and Prescriptive Remediations," Preprint, submitted October 8, 2023, https://arxiv.org/abs/2310.04988.
7. "Prompt Chaining," PromptingGuide.ai, accessed August 14, 2025, https://www.promptingguide.ai/techniques/prompt_chaining.
8. Lili Daie, "Library Open Workflows: Working with AI Tools Agents," Library Open Workflows (blog), Ex Libris Group, August 4, 2025, https://developers.exlibrisgroup.com/blog/library-open-workflows-working-with-ai-tools-agents/.
9. Ibid.
10. Lewis et al., "Retrieval-Augmented Generation."
11. Hannes Westermann, Jaromir Savelka, Vern R. Walker, and Kevin D. Ashley, "GAVEL: A Domain-Specific RAG System for Legal Cases," Preprint, submitted December 5, 2023, https://arxiv.org/abs/2312.03718.
12. Clarivate, "Clarivate Expands Its Academic AI Platform, Introducing Agentic AI for Research and Learning," news release, April 9, 2025, https://clarivate.com/news/clarivate-expands-its-academic-ai-platform-introducing-agentic-ai-for-research-and-learning/.
13. Khot et al., "Decomposed Prompting."
14. Sarah Gibbons, Tarun Mugunthan, and Jakob Nielsen, "Accordion Editing and Apple Picking: Early Generative-AI User Behaviors," Nielsen Norman Group, September 24, 2023, https://www.nngroup.com/articles/accordion-editing-apple-picking/.
15. Daie, "Library Open Workflows: Working with AI Tools Agents."
16. Ibid.
17. Westermann et al., "GAVEL."
18. Leo S. Lo, "AI Literacy: A Guide for Academic Libraries." *College & Research Libraries News* 86, no. 3 (2025), https://doi.org/10.5860/crln.86.3.120.