

theBABrain

FROM INFORMATION TO UNDERSTANDING

A Second Brain for Business Analysts

Building context for you
and your AI agents.



Agnieszka Balcerzak

theBABrain / September 2026

More room for thinking

Why I want analysts to have more room to analyse.

I want analysts to have time to understand a problem, question a proposed solution and work through disagreements with people. Meetings, scattered notes and repeated summaries can consume the attention that work needs. Producing another document faster does not necessarily help.

This guide starts with a narrower question: what would help us return to a piece of work without reconstructing its history every time? A second brain is one possible answer. The idea has a long history; using language models to help maintain it introduces new possibilities and new points of failure.

The following pages trace those foundations, examine relevant research and follow a fictional analyst through a changing project. They explain the thinking behind this approach, rather than provide a product manual.

Reading this guide

03-06 The problem, a working definition and a short history.

07-12 AI context, research and the limits of memory.

13-15 LLM Wiki and a practical maintenance workflow.

16-19 A fictional analyst project, from source to decision.

20-22 Boundaries, a first experiment and useful measures.

23-25 theBABrain, next steps and references.

Numbered references identify research and historical sources. The practical suggestions are my editorial position; the Northstar case and its sample responses are invented teaching material, not customer results.

First edition • September 2026. Sources checked 24 September 2026.

Why we keep starting over

A suggestion can become a decision somewhere between two meetings.

Imagine a customer-portal project. Monday's workshop ends with a suggestion: could customers track an order without signing in? By Friday, someone has added guest access to a requirements draft. The sponsor remembers asking for an assessment. Support remembers an agreement to try it. Nobody can point to an approval.

The documents exist. What is missing is the relationship between them: which statement was a proposal, what happened next and which position still applies. Before the analyst can explore the consequences, someone has to reconstruct that sequence.

An AI assistant given only Friday's draft could produce a convincing agenda for implementing guest access. It would be working from the wrong starting point. A longer answer or a more polished document would not recover the missing agreement.

The work before the analysis

Finding an email and understanding its significance are different tasks. The second may require a previous decision, the limits of someone's authority or a condition recorded in another meeting. Those connections matter when people disagree about what the project is doing.

A useful second brain should make that reconstruction easier to inspect and repeat. The time it releases can then go into asking whether guest access addresses the real problem, what alternatives exist and what evidence the team needs. That benefit depends on how well the context is maintained; storing more material alone will not deliver it.

We will return to this project later. First, it helps to be clear about what a second brain actually holds.

What is a second brain?

The important part is being able to pick up the reasoning again.

I use "second brain" here to mean a maintained collection of selected information, source references and relationships that helps someone resume useful work. It is a working definition, not a scientific category. Source-backed records remain distinct from interpretations, hypotheses and proposals.

In the portal example, a useful record would preserve the suggestion, who made it and the fact that approval is still missing. It would link the workshop note to the existing pilot scope. A colleague could then understand the open question without asking the analyst to retell the whole conversation.

What makes a record useful

The record needs enough detail to answer ordinary questions: why are we considering this change, whose concern does it address, and what still needs to be decided? Dates and source links help establish the answer. A status label is useful only if the evidence justifies it.

This can work in plain files, a notes application or an existing team workspace. A graph may make relationships easier to see, but the substance is in the records and the way they are maintained. AI can help organise them; it still needs instructions about what it may change.

A personal workspace also needs a clear relationship with shared team records. If a decision is approved in the team's system, point to that record and preserve its scope. Avoid creating a second version whose authority nobody understands.

The test is practical: could another person follow the reasoning back to the evidence? If they still need the entire story from memory, part of the context remains outside the system.

An older idea than AI

People were building external supports for thought long before language models.

1945 / Vannevar Bush

In *As We May Think*, Bush proposed the Memex, a personal device for storing and consulting records. Its associative trails would connect items so a person could revisit a line of enquiry. It was a proposal, not a working digital second-brain product. [1]

Mid-20th century / Niklas Luhmann

Luhmann worked with a card index, or Zettelkasten, using notes, identifiers and cross-references. The archive describes it as a thinking tool and a second memory. Connections between notes were part of the working method, rather than an optional way of displaying the collection. [2]

1962 / Douglas Engelbart

Engelbart's *Augmenting Human Intellect* considered the person together with tools, language, methods and training. This puts the organisation of work inside the design problem: introducing a tool also raises questions about how people will use it. [3]

These are different projects, not a single proven line of descent. Together they offer a useful starting question for an analyst: what should remain outside our heads so that we can return to a difficult problem and continue thinking?

That question comes before choosing software. It also explains why a folder full of saved documents may leave the original problem untouched: the relationships needed to use them have still to be worked out.

[1] Bush, 1945, sections 6-7. [2] Niklas Luhmann Archive. [3] Engelbart, 1962.

From collecting to using

A saved note has to find its way back into the work.

Tiago Forte's *Building a Second Brain* approach makes reuse explicit through CODE: Capture, Organize, Distill and Express. It is a practitioner method for turning collected information into useful output, rather than an independent experimental finding about productivity. [4]

For an analyst, the idea becomes concrete when a saved note changes the next conversation. A workshop observation might explain why a requirement exists, expose an assumption or give the team a question to investigate. It earns its place through that connection to the work.

What must survive a summary

Consider the sentence "Support suggested testing guest access for the pilot, subject to review." A summary that says "Guest access agreed" is shorter, but it has removed the speaker, the proposal status and the condition. It has changed the meaning.

This is where my adaptation of the approach starts. Capture the source because it matters to a question. Link it to the relevant project and earlier position. Ask AI to organise or summarise it while preserving attribution and qualifications. Keep your own interpretation separate, then use the record to prepare the next step.

Selection matters too. Collecting every message creates more material to maintain and search. Start with the evidence needed for a recurring task, and expand when the work calls for it.

Once an AI assistant joins this process, "remembering" becomes a less straightforward word. The next chapter separates what the model knows from what the current task actually receives.

[4] Forte, *Building a Second Brain: The Definitive Introductory Guide*. Adaptation and examples are original to this guide.

Four things often called memory

Knowing where information lives helps you diagnose why an answer failed.

1. Model knowledge

Patterns and information learned during training. These can support general reasoning, but do not establish your private project's latest agreement.

2. The active context

The material available for the current response: instructions, conversation, documents and tool results that the application supplies. A context window is a capacity limit; it does not mean every relevant source has been included or used well.

3. Persistent memory or stored knowledge

Information kept between sessions by an application or in an external workspace. Availability and behaviour vary. Storing a fact does not guarantee that a later task will retrieve the right version.

4. Retrieval

The process of finding material for a task. Retrieval-augmented generation combines generation with information obtained from an external collection; the original RAG paper demonstrated one such architecture. [7]

A maintained knowledge base can be searched using retrieval, and an agent can use both. These are complementary design choices. Before blaming the model, ask: was the evidence absent, not retrieved, outdated, misunderstood or contradicted by another source?

[7] Lewis et al., 2020. The four-part distinction is a practical explanatory model.

Give the task a usable starting point

Start with the decision the work is meant to support.

"Review this project" leaves an assistant guessing about the purpose of the review. "Prepare me to decide whether the pilot should include guest access" gives the same material a direction. It also makes omissions easier to notice.

Some tasks need very little history: rewriting a paragraph or comparing two supplied tables can be self-contained. A changing project is different. Its records may contain several positions that were reasonable at different times.

A brief the assistant can work from

- **Purpose and scope:** the question, project and period under review.
- **Evidence and current position:** relevant sources, recorded decisions, conditions and unresolved disagreements.
- **Authority and expected result:** what the assistant may change, what needs review and what a useful answer must contain.

You do not have to rewrite that background for every task. A maintained project page can provide an entry point, with links the assistant follows when it needs the detail. Check that it actually reaches the evidence that matters.

Personal preferences also need a label. Your preferred meeting format may help an assistant prepare a brief; it does not establish an organisational policy. Keeping those roles separate avoids giving a convenient note more authority than it has.

EXAMPLE REQUEST

Prepare a decision brief from the linked sources. Separate recorded facts, your suggestions and missing information. Cite each factual claim. Do not update approved requirements.

Why offloading is worth studying

Why put part of the remembering into the environment?

A calendar reminder changes what you have to keep in mind. A written decision can do something similar for a project: the team has a record to return to, provided it can find and interpret it.

Risko and Gilbert's 2016 review examines *cognitive offloading*: using physical actions and external supports to alter the information-processing demands of a task. It brings together research on what triggers offloading and what consequences it has for cognition. [5]

This gives us a research basis for asking how external supports change the work of remembering. It does not tell us how much time an analyst will save with an AI workspace; the review did not evaluate modern agents or theBABrain.

A question worth testing at work

If the reasoning behind a decision is preserved with the decision, will the next person spend less time reconstructing it? That is the practical hypothesis I would test. The answer should include the effort spent recording, maintaining and checking the context, not only the speed of the final search.

Try it on two comparable return-to-project tasks. Notice what you had to reopen, whom you had to ask and which uncertainty remained. Familiarity with the task can make the second attempt easier, so treat a small trial as learning rather than proof.

There is a further question for AI: even when the evidence has been stored and supplied, will the model use it correctly? That is where the next study becomes relevant.

[5] Risko & Gilbert, 2016. Research summary and practical test are deliberately separated.

More context is not a guarantee

The relevant sentence may be in the input and still be missed.

In *Lost in the Middle*, Liu and colleagues tested multi-document question answering and key-value retrieval. Performance in the evaluated models often depended on where relevant information appeared: it could be used less effectively in the middle of a long input than near the beginning or end. The work appeared as a 2023 preprint and in TACL in 2024. [6]

The study is specific to its models and tasks. It does not establish a rule for every current model or an ideal length for a project brief. It does, however, give us a reason to check whether supplying information leads to its correct use.

What happens to a condition?

Suppose a pilot review permits an experiment "for test orders only." Several later notes discuss possible production use. A useful answer must retain the condition even if it is buried in the earlier record. Mentioning the review without that limit would give the team a different position.

That makes a concrete test: ask what is currently authorised, then check the answer against the relevant passage. If the condition disappears, inspect which records were retrieved and whether the assistant understood their relationship.

Selecting relevant material and making key constraints explicit are reasonable design responses. They are not remedies established by this study. A brief can also become too short and omit the qualification it was supposed to preserve.

Position is only one difficulty. Project knowledge also changes, and a system can remember an earlier answer perfectly while giving the wrong answer today.

[6] Liu et al., *Lost in the Middle*, TACL 2024; preprint 2023.

Memory needs to handle change

"What did we agree?" and "What applies now?" can have different answers.

A project might require sign-in in September and authorise a limited exception later. Both records belong in its history. The task is to explain the change and its scope, rather than simply repeat the latest sentence mentioning access.

LongMemEval examines five abilities relevant to this problem: information extraction, reasoning across sessions, reasoning about time, updating knowledge and abstaining when an answer is unsupported. Its 500-question benchmark tests chat assistants and explores changes to memory design. It was published at ICLR 2025, following a 2024 preprint. [8]

The benchmark measures performance on its tasks, not the value of a particular analyst workspace. Its categories are nevertheless useful for designing a small check of your own.

Questions to take back to a project

Ask what the sponsor approved, which earlier constraint explains it, and what applied before the change. Then ask what superseded the old position. Include a question the records cannot answer, such as a production date that nobody has agreed.

Write down the expected answers and their evidence before testing. Otherwise a fluent response can become the answer key, even when it has quietly filled a gap.

When something fails, look for the cause: missing source material, poor retrieval, an unsupported interpretation or an error in your own expected answer. Each calls for a different repair. Adding more notes may leave the actual problem in place.

[8] Wu et al., LongMemEval, ICLR 2025. The five-question check is an original adaptation.

Keep judgment in the loop

An answer can be easy to read and difficult to challenge.

A smooth summary gives the reader few reasons to pause. In analysis, the awkward detail may be exactly what needs attention: a stakeholder disagreed, a condition remains open, or a proposed change has no approval.

Lee and colleagues surveyed 319 knowledge workers and collected 936 examples of generative AI use. Higher confidence in AI was associated with less reported critical thinking; higher task-specific self-confidence was associated with more. Participants also described work shifting towards verification, integration and oversight. This is self-reported evidence, not a demonstration that AI causes a lasting loss of thinking ability. [9]

Review the statement that changes the next step

Rather than ask whether the whole summary "looks right," pick the claim someone will act on. If it says the sponsor approved guest access, find the supporting passage. Does it record approval, a request to investigate or merely interest in the idea?

A citation is a route to that check. Its presence alone does not make the claim sound. The cited note might concern another scope, record a different speaker or contain a condition the summary omitted.

I want AI to create more room for this kind of judgment and for the conversation that follows it. An analyst who can explain the evidence, recognise its limits and ask a better question is still doing the analytical work. The system should help them arrive at that conversation prepared.

[9] Lee et al., CHI 2025. Review questions are practical recommendations.

Knowledge between conversations

A useful synthesis should still be there when the next question arrives.

If an assistant has already connected a workshop note to an earlier decision, repeating the same investigation in the next chat is frustrating. The useful work is the connection it established, including the evidence and any uncertainty. That is worth preserving beyond the conversation.

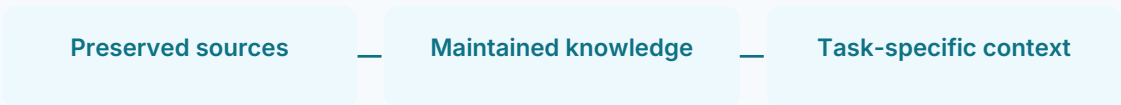
Andrej Karpathy's April 2026 *LLM Wiki* note proposes a pattern in which an LLM builds and maintains linked knowledge pages over preserved sources. It separates raw material, the maintained wiki and instructions for operating it, with workflows for ingestion, querying and checking. It is an architectural proposal, not a controlled evaluation. [10]

Applied to the portal example, a maintained page could connect the proposed access change to the pilot scope and the question awaiting review. The originals stay available. A later decision can update the current view without erasing the earlier position.

The difficult part is deciding what changed

A new note may repeat an old suggestion, contradict another account or record an actual decision. Those are different events. The instructions need to tell the assistant how to preserve that distinction and when to leave a question for a person.

AI can do the organising and linking within those rules. Accuracy and stale information still need review. The point of a maintained wiki is that this work can accumulate; whether that is worthwhile depends on what it helps you do next.



[10] Karpathy, LLM Wiki, 4 April 2026. Example and diagram are this guide's adaptation.

Chat, retrieval and a maintained wiki

A maintained wiki can work alongside retrieval and ordinary chat.

For a one-off comparison of two documents, supplying both in a chat may be enough. The question is bounded, the evidence is manageable and there may be little reason to maintain a separate knowledge collection afterwards.

When the material is larger, the assistant needs a way to find relevant evidence. Retrieval-augmented generation combines generation with access to an external collection; Lewis and colleagues' 2020 paper demonstrated an influential architecture for doing this. Retrieval may operate over original documents, maintained pages or both. [7]

When a maintained view becomes useful

The extra work becomes worth considering when you repeatedly need an integrated picture: what changed, which decision applies and why a question is still open. A maintained page can preserve those relationships between tasks. It may still need retrieval to reach the right sources.

The maintenance does not have to fall entirely on the analyst. AI can organise, link and update records under agreed instructions. The analyst chooses the scope and reviews consequential changes. That arrangement still has a cost, which should be counted alongside any benefit.

These approaches overlap. A practical workspace might use a short project page as an entry point, search the underlying notes and use a chat to explore the implications. It would be misleading to say that every retrieval system starts from scratch.

Start from the question you keep having to answer. Check whether an existing team workspace already holds the necessary context. Add a maintained view where it solves a repeated problem, rather than because the architecture sounds attractive.

[7] Lewis et al., 2020. Comparison is a practical design synthesis, not a product benchmark.

From a new note to a useful record

The working loop is capture, connect, use and review.

A workshop note arrives. Before asking AI to process it, decide what work it belongs to and whether you are permitted to use it in the chosen environment. Preserve the source with its date, author or speaker and a stable reference. If the material is only a summary, label it accordingly.

The assistant can then connect the source to existing records. In the portal example, it should recognise the relationship with the sign-in requirement and preserve who suggested the alternative. It should not promote that suggestion into a decision.

Give the update a reason

Use the result to prepare a meeting, compare positions or revisit a requirement. Ask the assistant to show the evidence behind the answer and the questions it could not resolve. A saved record becomes useful when it helps with this next piece of work.

When new evidence arrives, review what it changes. A later message may explain an earlier statement without replacing it. A superseding decision needs an explicit link to the earlier position, whose history remains available.

AI can maintain these connections within agreed rules; the analyst sets the purpose and scope, reviews results and remains responsible for decisions. Set review effort according to the consequences. Updating a reading note and recording contractual approval should not receive the same treatment.

This loop needs neither an always-running agent nor automatic importers. It needs clear responsibility for each change and a way to recover if the change is wrong. The following fictional case shows how that can look across several conversations.

Northstar: a customer portal

A small example of how a source becomes useful context.

Northstar is a fictional equipment supplier considering self-service order tracking. Maya, its business analyst, is preparing a pilot review. The names, dates, records and model responses in this case are invented for learning.

The selected evidence

S1 / Pilot brief, 3 September. The approved pilot scope requires customers to sign in before viewing orders. The stated aim is to test self-service tracking with a limited group. The brief does not approve guest access.

S2 / Support workshop note, 10 September. Support lead Sam reports that password resets interrupt some customers. Sam suggests exploring access through an order reference and a one-time code. The note records no approval or measured impact.

S3 / Sponsor email, 12 September. Sponsor Jo asks the team to assess a guest-access option before the pilot review. Jo says to retain the sign-in requirement until that assessment is discussed.

The question

What should Maya bring to the review? If she asks for "the new guest-access requirements", the request already assumes a change that the evidence does not support. Her first task is to clarify the current position, the proposed alternative and the evidence still needed.

Fictional learning case. S1-S5 refer only to the invented records in this chapter.

Preserve what each source says

A useful note retains attribution, scope and uncertainty.

An illustrative maintained record

Subject: Access to order tracking in the pilot

Current position: Sign-in required [S1]; retained pending review [S3].

Reported concern: Sam reports password-reset interruptions [S2]. Frequency and impact have not been established.

Proposed option: Order reference plus one-time code [S2].

Requested next step: Assess the option before the pilot review [S3].

Status: Alternative under assessment; no change approved.

A prompt to draft the record

Read S1-S3. Produce a short account of the current position, the reported concern and the proposed option. Cite the source for each statement. Keep speaker attribution. List missing evidence separately. Do not infer an approval.

What Maya checks

Does the draft say that all customers struggle with passwords? That would overstate S2. Does it claim that Jo approved guest access? That would contradict S3. Does it present the one-time-code mechanism as secure? None of the selected sources establishes that.

Maya checks the draft against the originals before keeping it. The record links to the project brief and workshop note. A colleague can see why the topic is open without reconstructing the entire conversation.

Fictional example. This record illustrates a reasoning pattern, not a required theBABrain file schema.

When a later decision arrives

Update the current view while preserving the reasoning that led to it.

S4 / Review record, 18 September. In this fictional record, Jo authorises a limited guest-access experiment for test orders only, subject to a security review. Production access continues to require sign-in. The review record specifies these boundaries explicitly.

What the maintained view should now say

- The assessment request has progressed to a conditional experiment [S4].
- Production access still requires sign-in [S4].
- Security review remains a condition; this record does not establish that it has happened [S4].

Ask for a proposed change before applying it

Compare S4 with the current access record. Show which statements should change, which still apply, and which questions remain open. Preserve S1-S3 and the earlier position. Do not broaden the scope beyond test orders.

The assistant may suggest checking authentication, support scripts and customer communications. Those are analytical suggestions for Maya to validate. They are not dependencies proved by the four documents, and they should not be recorded as approved requirements.

S5 / Follow-up, 21 September. Sam asks when customers can use guest access in production. Maya can now answer with the boundary from S4 and a clear next question. A later request from Sam does not silently replace Jo's recorded decision.

Fictional learning case. No actual customer, security assessment or business approval is represented.

Return with a better brief

Maya now has a basis for the next conversation.

Maya's request

Prepare a brief for the next pilot review using S1-S5 and the maintained access record. State what applies now, what changed and what we still need to decide. Cite evidence. Keep recommendations separate. If a review outcome is missing, say so.

An illustrative result to review

Current position: Production sign-in remains. A test-order experiment is conditionally authorised [S4].

Reason for exploring it: Sam's reported password-reset concern [S2].

Recent question: Sam asks about production timing [S5].

Evidence gap: The selected records contain no completed security review or approval for production guest access.

Suggested agenda: Confirm the review outcome, examine experiment evidence and clarify who can authorise any next step.

The agenda is a recommendation, so Maya validates it. She also checks whether the source set is complete. A good answer from S1-S5 cannot establish that an unprovided S6 does not exist.

Back in the meeting

Maya can explain why guest access is being explored without suggesting it is approved for production. She can also identify the question that needs an answer before the experiment proceeds: has the security review taken place?

That gives the conversation a starting point. The record cannot settle an unrecorded decision or prove that the team saved time. It can make the evidence and the remaining uncertainty available for people to discuss.

Illustrative prompt and answer, authored for this guide. They are not a recorded model evaluation.

Keep the system worth trusting

Context can be sensitive, stale or wrong. Plan for all three.

Use material you are permitted to process

Check the chosen AI environment and your organisation's rules before using work records. A local file can still be sent to a remote model by an assistant. File location alone does not tell you where processing happens. Start with fictional material when the boundary is unclear.

Separate evidence from instructions

A document can contain text that looks like a command. Treat imported material as evidence to examine, not authority to change settings, disclose information or perform actions. Limit the tools and permissions available for the task.

Make stale knowledge visible

Record when a claim applied and what superseded it. A recent summary can still describe an old situation. Preserve unresolved contradictions instead of asking the model to choose the most plausible version without evidence.

Keep recovery possible

Use an appropriate backup or change history for the workspace. Review material before sharing it. A source link can itself reveal a client name or internal location; check references as well as the visible prose.

These are practical handling considerations, not a substitute for your organisation's security or retention requirements. The right level of control depends on the information, the people involved and the consequences of a mistake.

BEFORE SHARING

Check the content, attachments, references and access permissions together.

Start with one useful question

Choose a piece of work you will need to return to.

Think of an upcoming meeting where you will need to explain what was decided and what is still open. Choose a small, permitted source set: perhaps a brief, a meeting note and a later update. Keep their dates and originals available.

Before you start

Write down the question you need to answer and the qualification that must survive. In the Northstar case, it is "test orders only." Also note what the records cannot establish. That gives you something concrete against which to review the result.

Ask your assistant to prepare a maintained account with source references, attributed statements and open questions. It can do the organising and linking. Check the consequential claims and ask it to correct unsupported interpretations before you reuse the record.

Come back in a fresh session

Provide the workspace instructions and ask the question again. Observe which files the assistant reads. Can it explain the current position and the change that led to it? Does it admit that a requested date or approval is missing?

This is a small trial, not a commitment to reorganise your whole working life. Count the effort of preparation and review as well as the effort saved when you return. If it helps, extend the same pattern to another recurring question.

A useful result is one you can bring into the actual conversation: an answer with evidence, a qualification preserved and a clear account of what still needs human attention.

Measure the work, not the collection

Judge the system at the point where someone uses its answer.

A growing collection of notes tells you little about whether the work has improved. The more useful moment to observe is the return: can someone prepare for the review, explain the decision or find the unresolved issue without reconstructing everything?

Compare a similar task using your usual approach and the new workspace. Record the conditions, since familiarity or an easier source set can explain a faster result. A personal trial is useful evidence for your own decision, not a controlled scientific study.

What to look for

- **Evidence:** Can you reach the passage supporting a consequential claim?
- **Meaning:** Are dates, speakers, scope and conditions intact?
- **Uncertainty:** Can you tell what remains unknown or disputed?
- **Effort:** What did preparation, maintenance, retrieval and review cost together?

If searching becomes quicker but checking takes longer, investigate the reason. Perhaps the summaries are too broad, the links too weak or the source selection incomplete. Repair that part before expanding the collection.

My reason for pursuing this approach is the space it could give analysts for understanding problems and working with people. The practical test is whether the system helps you do that in your own setting. Keep what supports the work, and revisit the decision when the tools or the work change.

A place to begin

The support system you and your agents need.

theBABrain is a free, open-source workspace for business analysts. Its public starter combines linked Markdown files, instructions, blank profiles and project patterns. You work with it through a compatible AI assistant with file access.

The starter does not run itself. Your assistant follows the requested workflows; you select the material and review the work. The public 0.1.0 release does not include ready-made importers, account integrations or schedules. Its helper requirements include Python 3.10+ and PyYAML.

Choose your next step

Explore the approach

thebabrain.ai

Try the public starter

github.com/agabalcerzak/thebabrain

Join conversations and practical workshops

luma.com/thebabrain

Follow conversations with analysts about AI

youtube.com/@thebabrainai

Created by Agnieszka Balcerzak, business analysis trainer, mentor and consultant. Start with a question from your work, keep the evidence close, and build only as much structure as you need.

Check the repository for current setup instructions.

Sources & further reading

Numbered citations connect claims in the text to their sources.

[1] Bush, V. (1945). As We May Think. The Atlantic Monthly, July. Sections 6-7.

Historical primary essay; Memex and associative trails.

www.w3.org/History/1945/vbush/vbush6.shtml

[2] Niklas Luhmann Archive, Bielefeld University. Zettelkasten.

Institutional archive; historical description of the card index.

niklas-luhmann-archiv.de/nachlass/zettelkasten

[3] Engelbart, D. C. (1962). Augmenting Human Intellect: A Conceptual Framework.

Primary research report; a systems view of augmentation.

dougengelbart.org/content/view/138/

[4] Forte, T. (2023). Building a Second Brain: The Definitive Introductory Guide.

Author's practitioner guide; CODE. Not an independent effectiveness study.

fortelabs.com/blog/basboverview/

[5] Risko, E. F., & Gilbert, S. J. (2016). Cognitive Offloading. Trends in Cognitive Sciences, 20(9), 676-688.

Research review. Open manuscript: discovery.ucl.ac.uk/id/eprint/1508770/.

doi.org/10.1016/j.tics.2016.07.002

[6] Liu, N. F., et al. (2024). Lost in the Middle: How Language Models Use Long Contexts. TACL, 12, 157-173. Preprint 2023.

Experimental study; context-position effects in the evaluated models and tasks.

arxiv.org/abs/2307.03172

All web sources accessed 24 September 2026. Links are clickable in the PDF.

Research & architectural ideas

Numbered citations connect claims in the text to their sources.

[7] Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS 33.

Architecture and experimental evaluation of retrieval-augmented generation.

arxiv.org/abs/2005.11401

[8] Wu, D., et al. (2025). LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. ICLR. Preprint 2024.

Benchmark research; original LongMemEval, not a current product leaderboard.

arxiv.org/abs/2410.10813

[9] Lee, H.-P., et al. (2025). The Impact of Generative AI on Critical Thinking: Self-Reported Reductions in Cognitive Effort and Confidence Effects From a Survey of Knowledge Workers. CHI.

Survey evidence; 319 participants and 936 examples; association, not causation.

[Read the original source](#)

[10] Karpathy, A. (2026). LLM Wiki. GitHub Gist, 4 April.

Practitioner architecture note; not an empirical evaluation.

gist.github.com/karpathy/442a6bf555914893e9891c11519de94f

Evidence note. This is an introductory, selective guide, not a systematic literature review. None of the cited studies tests theBABrain. Exercises, prompts and the fictional case are original teaching material. Publication dates identify the evidence; they do not imply that historical results describe every current model.

All web sources accessed 24 September 2026. Links are clickable in the PDF.