FRANKLIN DISCOVERY™

TECHNICAL OVERVIEW

The missing half of the answer.

Human thinking runs on two systems: fast pattern recognition and slow, deliberate reasoning. Large language models were built on the first. Without the second, an answer is a prediction of what words usually come next, not a conclusion reached from an understanding of what those words mean.

OCTOBER 2026

Overview.

Human cognition runs on two systems, one fast and pattern-driven, the other slow and deliberate. AI research split along the same line decades ago into neural methods and symbolic methods, with generative AI built almost entirely on neural methods.

The limits of neural methods are well documented. Vector search can't connect information split across two documents, can't tell you why a result came back, and once it’s on a predictive path, you can't adjust it without starting over. Questions about how material is organized go badly for the same reason. And generation fabricates. The research literature counts five distinct kinds, and none of them trace to any one model.

Symbolic AI stalled decades earlier, for practical reasons rather than logical ones. Nobody could hand-author enough knowledge to cover more than a narrow domain, and knowledge bases big enough to be useful filled with contradictions faster than experts could fix them. Any system built on a symbolic layer inherits that problem.

A neurosymbolic architecture answers the limitations of symbolic AI by building the general knowledge once and shipping it, rather than writing it by hand for every deployment. Meanings are stored apart from the words used for them, represented mathematically, and assembled by rule into an explicit structure of concepts and the connections between them. An organization's documents map into that existing structure instead of building a new one. A deployment doesn't start with rule authoring, and the quality of the representation doesn't depend on how much material the organization has.

This white paper examines both models, their limitations, and why bringing neural and symbolic methods together should be the next step in AI’s evolution. The first two chapters cover the two traditions and their limits. The middle explains how meaning gets represented, including the structures the team calls entigens, and what you get from it. The back half covers use cases, workflows, boundaries, and the questions worth asking of any AI system, with a glossary at the end.

CHAPTER 1

Two systems of thought.

The brain gets described three ways, and the accounts fit together rather than compete. Physiology maps the structures and the pathways between them. The neuroscientist Paul MacLean's evolutionary account[1] describes a brain that added layers over time without discarding the older ones, so the old machinery still runs under the new. The Nobel Prize-winning psychologist Daniel Kahneman's behavioral work[2] describes what that produces from the inside: two modes of thinking running at once.

System 1 is fast. It orients, seeks coherence, and runs without any sense of effort or choice, and it handles most of what a person does in a day. System 2 is the one you notice using. It takes attention, it costs something to run, and it stays out of the way until System 1 turns up something that doesn't add up. Neither is better than the other. The arrangement works because the expensive system engages only once the cheap one stops being enough.

Machine learning developed along a similar split. The computer scientist Pedro Domingos organizes the field into five tribes[3], each with its own origins and its own master algorithm: Symbolists working from logic and philosophy toward inverse deduction, Connectionists working from neuroscience toward backpropagation, Evolutionaries from evolutionary biology toward genetic programming, Bayesians from statistics toward probabilistic inference, and Analogizers from psychology toward kernel machines. Most of what's deployed today descends from two of them: neural networks from the Connectionist line, and knowledge representation and logic from the Symbolist one.

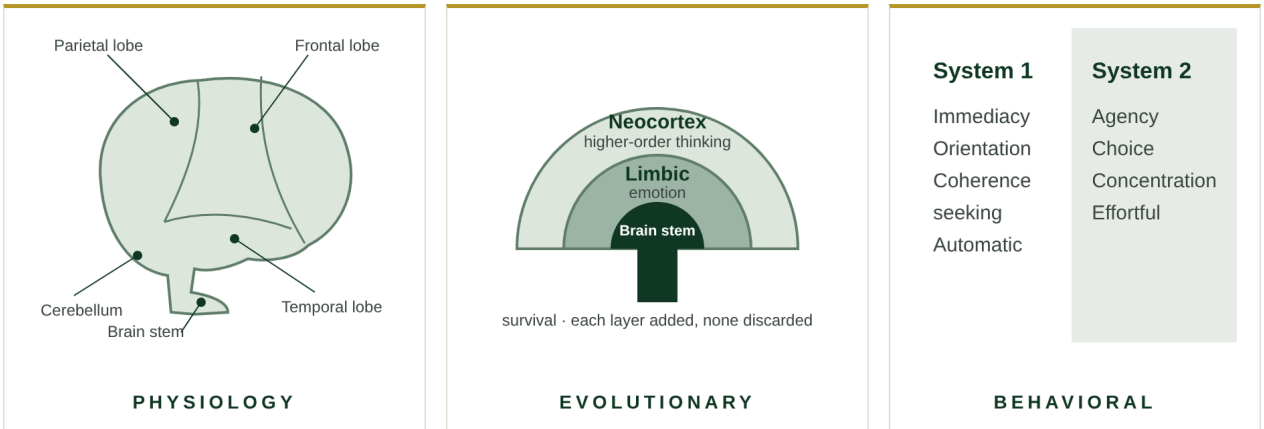


Figure 1. Three lenses on the brain: structure, evolutionary layering, and the two modes of thinking.

	SYSTEM 1	SYSTEM 2
Mode	Immediacy, orientation, coherence seeking, automatic	Agency, choice, concentration, effortful
AI technologies	Neural networks: pattern recognition, machine learning, deep learning, reinforcement learning	Symbolic AI: conceptual relationships, knowledge bases, knowledge graphs, logic formalisms
Character	Probabilistic. Quick, efficient, data driven	Deterministic. Explainable, knowledge driven

A survey in *IEEE Transactions on Neural Networks and Learning Systems* describes the combination:[4]

> *"Neurosymbolic AI is an increasingly active area of research that combines symbolic reasoning methods with deep learning to leverage their complementary benefits."*

The same survey names the payoff: approaches that keep results interpretable, stay competitive on performance, and leave room for expert knowledge.

Independent research reaches the same conclusion.

Up to 20%

improvement in multi-hop question answering with knowledge-graph retrieval.

HippoRAG, NeurIPS 2024 [5]

77.6%

better retrieval ranking and 28.6% faster median issue resolution, in production customer service.

LinkedIn, 2024 [6]

Fewer fabrications

Grounding language models in knowledge graphs showed promising results in reducing hallucination.

NAACL 2024 survey [7]

16.7% → 56.2%

overall accuracy once knowledge-graph grounding was added. Vector-only retrieval scored 0% on metrics, KPI, and strategic planning questions.

Diffbot KG-LM benchmark [8]

CHAPTER 2

The limits of each approach.

Both traditions ran into structural limits. The case for a hybrid rests on what each side can't do alone, so the specifics matter.

The neural side.

Vector embeddings changed retrieval by making semantic similarity computable. A system can find documents that are about something without matching the words used to describe it, across billions of documents, fast enough to be practical. Within that job, embeddings work well.

The trouble starts past that boundary. Similarity search can't connect information split across two documents, so a question that needs two hops returns whichever single passage looks closest and stops. Questions about relationships, hierarchies, or how things add up go badly, because answering them takes knowing how the material is organized rather than what it resembles. There is no account of why any particular result came back. And nothing in the mechanism guarantees an answer came from the content instead of the statistics.

Generation adds a failure mode of its own. The survey literature[9] separates fabrication into distinct kinds, and they're easier to recognize side by side, because each one shows up differently in the answer you receive:

KIND	WHAT IT LOOKS LIKE IN PRACTICE
Factual inconsistency	The answer contradicts something the source material states.
Factual fabrication	The answer includes details that exist nowhere in the source material.
Instruction inconsistency	The answer ignores what you asked it to do.
Context inconsistency	The answer contradicts information you gave it in the same conversation.
Logical inconsistency	The answer contradicts itself.

None of these is a defect in a particular model, and continued improvements in the models won’t solve them completely. They follow from how the models work, producing text by predicting likely continuations.

Computational and energy requirements keep climbing as models grow.[10] And a context window takes only what fits inside it, so anything a model wasn't trained on and can't be shown at query time is unavailable to it.

The symbolic side.

Symbolic AI ran into its limits earlier, for different reasons. In the AAAI Robert S. Engelmore Memorial Lecture, the AI researcher Henry Kautz named the two problems that ended the expert-system era:[11]

> *"The first challenge was the need for principled and practical methods for probabilistic reasoning. The logical rule-based approach excelled at capturing knowledge about relationships among concepts and entities (such as class/subclass/instance or object/part/attribute hierarchies). Still, it was poorly suited for problems where one needed to assign probabilities to conclusions... The second unsolved challenge for the expert system approach was named the 'knowledge acquisition bottleneck.' Capturing all but the narrowest domains required a huge number of rules. Not only was it difficult or impossible to recruit and train enough experts to write enough rules, but once the knowledge bases became large, they inevitably became full of inconsistencies and errors."*

Any architecture with a symbolic layer inherits the knowledge-acquisition bottleneck and has to find ways to work around it.

CHAPTER 3

Representing meaning.

The two traditions differ in what they represent. Neural methods work on words. They find what looks similar, and when they generate, they predict what usually comes next. Symbolic methods work on meaning: concepts and the rules connecting them, written down so the reasoning can be followed. Writing all of that by hand is what stalled them.

It only has to be written once if the knowledge is general. Word meanings, shared meanings, and how concepts connect don't change between a law firm and an engineering firm. Build that map once and ship it, and an organization's documents are read into it

instead of being used to build one. Franklin Discovery™ by entigenlogic® is built that way, and the place to start is how it represents meaning.

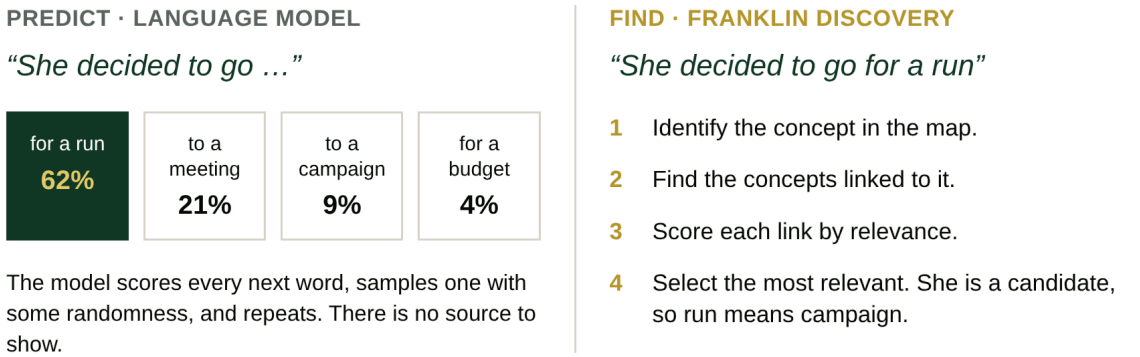


Figure 2. Two ways to arrive at an answer. A language model scores the next word from statistics and has no source to show. Franklin Discovery resolves the concept in the map, follows its links, and scores them, so the path to the answer can be shown. Scores are illustrative.

In a review of natural language processing research, Erik Cambria and Bebo White named representation as the constraint the field kept running into:[12]

> *“Today, most of the existing approaches are still based on the syntactic representation of text, a method that relies mainly on word co-occurrence frequencies. Such algorithms are limited by the fact that they can process only the information that they can ‘see’.”*

Words are not meanings.

A word is a label, and it can point to more than one meaning. Which meaning applies depends on the surrounding words.



Figure 3. One spelling, nine meanings. The words around each “run” decide which one applies.

SENTENCE	CONCEPT
The bat ate fruit from the tree.	An animal eating.
The boy swung the bat and hit a home run.	A boy playing baseball.
She batted the mosquitoes away from the tent flap.	A woman swatting an insect.

Franklin Discovery stores meanings apart from the words used for them and represents each one mathematically. An embedding gives a word a single position in a vector space and lets surrounding context shift it, which treats the word as one thing with soft edges. Franklin Discovery treats “bat” as three unrelated concepts spelled the same way.

Entigens, in plain terms.

The team's names for these structures appear throughout its technical material, so they need defining.

An **identigen** represents one distinct meaning a word can carry. “Bat” the animal, “bat” the piece of sporting equipment, and “bat” the swatting motion are three identigens that happen to share a spelling.

An **entigen** is an interpreted meaning: a complete semantic unit assembled from identigens according to interpretation rules. The name compresses “entity generation.” When the system reads “the boy swung the bat,” it doesn't store those words. It stores the entigen group that captures what the sentence means, with the right sense of every word selected and the relationships between the concepts made explicit.

The rules doing that assembly draw on a foundation ontology built over many years. The plainest description of it comes from inside the company: an embedded dictionary, thesaurus, and mini encyclopedia. It records what concepts exist, which words can point at

them, and how they relate. A concept is stored as math rather than as a word, closer to the way a mind keeps an idea apart from the words for it.

Interpretation then becomes selection and assembly. Rules test which candidate meanings fit together with the other candidates present, and the combination that survives is the interpretation. The output is a structure covering a phrase, a sentence, or a passage, with every concept identified and its relationship to the others written down.

Structure.

The structure is a graph. Concepts are the nodes, the connections between them are the edges, and a further ontological layer places each concept inside a hierarchy of types.

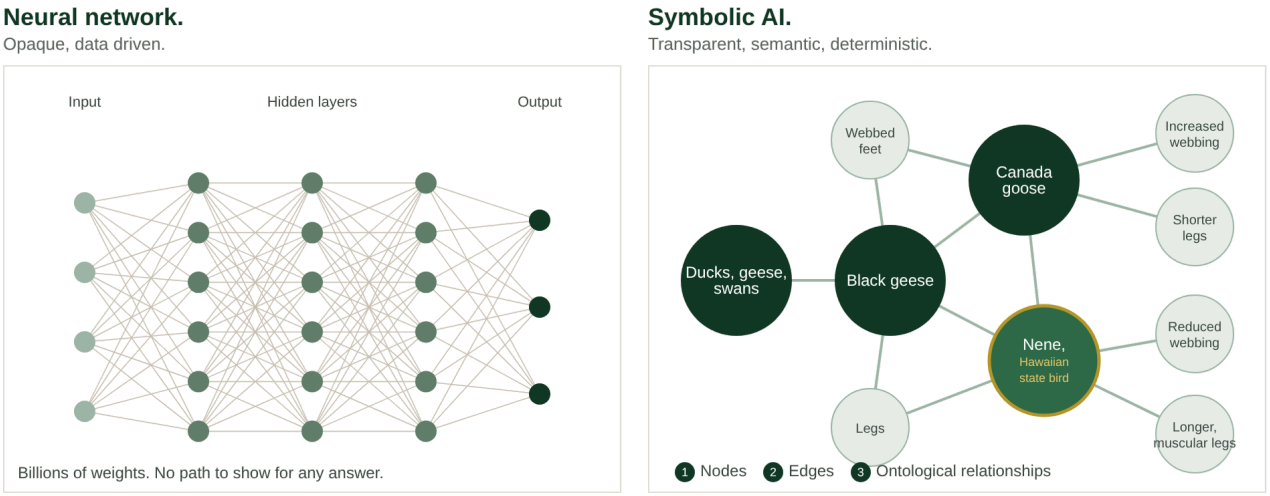


Figure 4. The same knowledge, two ways. Weights in a trained network carry no inspectable account of why it returned a result. An explicit hierarchy carries the concepts, the relationships between them, and the traits that separate one branch from another.

A structure that contains the concept of a nene, the Hawaiian state bird, also records that it's a black goose, a relative of the Canada goose, in the family that includes ducks, geese, and swans. The same structure records the characteristics that separate those branches, including webbing on the feet and leg length, and the direction those traits move as you descend: increased webbing and shorter legs down one branch, reduced webbing and longer, more muscular legs down another.

A system with that structure can answer a question about a bird it has never encountered by reasoning about what the bird inherits from its group. An embedding puts nene near goose in a vector space, which helps with similarity and says nothing about what a nene inherits from being a goose.

Three knowledge bases.

Knowledge is stored in three layers.

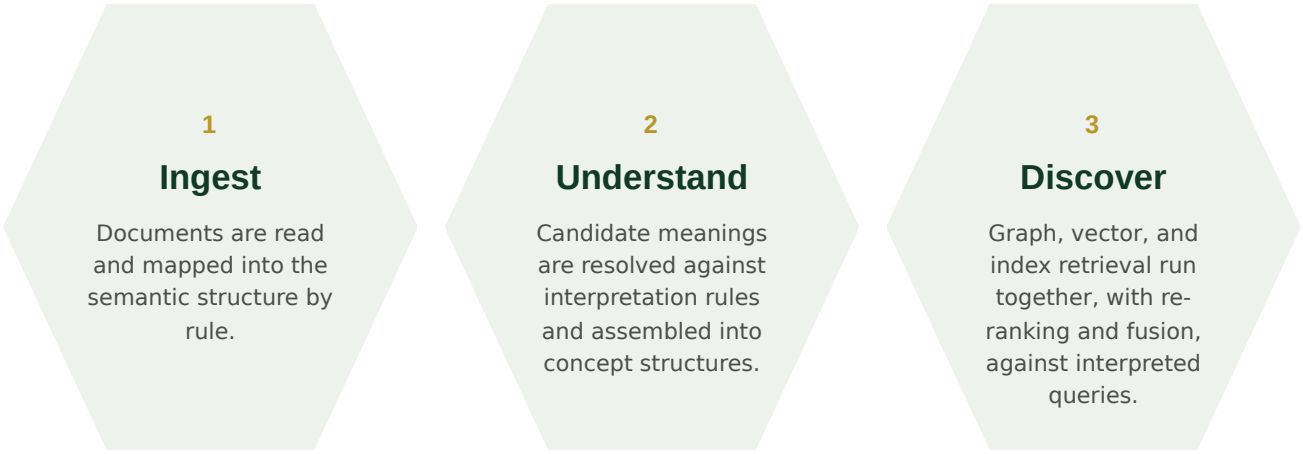
LAYER	CONTENTS
Core	General language and world knowledge, common to every deployment.
Domain	The vocabulary and relationships of a particular field.
Learned	An organization's own working knowledge, captured as it accumulates.

The core base is the answer to Kautz's bottleneck. It took years to build, and it ships with the system, so a deployment doesn't begin with rule authoring for general language. An organization's documents map into it, and the quality of the representation doesn't depend on how much material the organization brings.

CHAPTER 4

The processing pipeline.

A contract in a shared drive becomes something you can question in three stages.



Ingest. Documents are read, and their content is mapped into the semantic structure. Interpretation is rule-based, which makes the result reproducible: the same document processed twice produces the same representation, and nothing enters the structure during construction that wasn't derived from the source.

Understand. The interpretation engine resolves candidate meanings against interpretation rules and assembles concept structures. A document needs no labeling, training set, or model build to become part of the searchable knowledge.

Discover. Retrieval combines several methods. Graph traversal, vector similarity, and conventional indexing run together, with re-ranking and fusion determining the most relevant results. Queries go through the same interpretation as documents. A question in plain language, whether a word, a phrase, or a paragraph, becomes the same kind of structure a document does, and matching happens between structures rather than between strings.

Each result carries a relevance score you can inspect. Selecting a result opens a report identifying the content it came from.

Franklin Discovery is a hybrid. A large language model handles the language, which is what those models do best—finding and reasoning—while the concept map runs on the patented rule and graph machinery, which performs interpretation, structure, and relevance scoring without a language model.

CHAPTER 5

Benefits.

The behaviors below follow from the design: meaning represented explicitly, answers only from ingested content, and no training cycle.

Explainability.

Because the representation is explicit, you can capture and review the factors considered and the path taken to a conclusion without reconstructing them after the fact.

GDPR Article 22 already constrains decisions produced by automated processing that significantly affect individuals, and several US states have adopted similar rules. Under the EU AI Act, transparency obligations took force on August 2, 2026. The obligations attached to high-risk systems, which carry the substantive explainability requirements, were deferred by the AI Omnibus to December 2, 2027 for standalone systems and August 2, 2028 for AI embedded in products.[14]

Bounded answers.

The system works from ingested content, so when that content doesn't support an answer, it says so rather than producing something plausible. Fabrication is rare because answers are found in the content rather than predicted.

Knowledge without retraining.

New documents and new domain knowledge come in through ingestion. There's no fine-tuning step, no labeled training set, and no model to rebuild. Knowledge is as current as the last ingestion, and adding it doesn't cost what a training run costs.

Compute.

Franklin Discovery works more like specialized AI than general-purpose AI, because it answers from a bounded set of documents, your own, instead of everything a large model was trained on. Much of the work of answering a question happens by following connections in the concept map, which takes far less compute than running every step through a large model.

Deployment.

Franklin Discovery runs across laptops, servers, and cloud infrastructure. Throughput scales with the processing resources allocated, so document volume and processing rate can grow with the work.

CHAPTER 6

Use cases.

Most of what an organization knows is in its documents. IDC estimated that 90% of the data organizations generated in 2022 was unstructured.[13] Nearly every business has an archive like that. The ones that get the most from this architecture share something else: when an answer is wrong, somebody pays for it.

Secure environments. Work where content can't leave the organization's control. The system runs on infrastructure you choose, from a laptop to your own servers to your cloud,

answers only from what you've ingested, and nothing you ask becomes training for an outside model. The data path fits in one sentence.

Privacy-regulated material. Health records, student records, personnel files, financial accounts. Privacy law makes handling this material a legal question before it's a technical one, and reviewers ask questions about the data path: where content goes, what it trains, and who can see it. The same one-sentence answer applies, and usage is tracked, so there's a record of who asked what.

Compliance and audit work. Policy questions, regulatory filings, controls documentation. An auditor doesn't accept an answer; an auditor accepts an answer with evidence. Because the reasoning is inspectable and the source arrives attached, the evidence exists before anyone asks for it.

Work product that gets challenged: litigation, disputes, anything an opposing party will pick apart later. The answer comes from the record and nowhere else, and when the record doesn't support an answer, that's what the system says.

Institutional knowledge. Organizations lose what their most experienced people carry when those people leave. The learned knowledge base captures working knowledge as it accumulates and keeps it in a form that can be searched by meaning years later, by people who never met the expert.

Incomplete and messy records. Real archives are scans, fragments, and inconsistent formats. Because ingestion needs no labeling, training set, or model build, the system works on records as-is, and a gap in the record is reported as a gap.

Knowledge that changes fast. Regulations, product specifications, rate tables, policy manuals. Anything that changes faster than a retraining cycle can follow. New material is current as soon as it's ingested.

Continuous monitoring. Watching a stream of documents for the ones your business needs to see: solicitations, filings, dockets, publications. Concept-level matching runs against each new arrival, so relevant items surface by meaning rather than keyword match.

Franklin Discovery is used today for legal work, government contracting, and workforce training.

In use at one firm.

Roberson and Associates, an engineering and management consulting firm working in RF spectrum management for government and commercial clients, was an early customer. Its president and CEO, Dennis Roberson, chairs the entigenlogic board, so read what follows as

a description of use rather than an endorsement. The firm loads documents by client or department, each collection in its own instance, and anyone on that project can ask a question in plain language and get the answer from the record, with the document it came from.

Proposals. A solicitation specifies what key personnel must bring, and a resume short on any requirement costs points or knocks the bid out. On a federal review engagement, the firm ran 25 resumes against the solicitation's language, showed each person where the system had found a gap, ran the updated resumes a second time, and picked the eight who would do the work. When the FCC issued a notice of proposed rulemaking on a new block of spectrum, the proposal team's questions went to the system, and the answers, each pulled from the notice, went to the partner firms on the bid.

Reporting. When the final report on the National Radio Quiet Zone needed a background chapter, Paul Steinberg, formerly CTO of Motorola Solutions and now with the firm, wrote it from a few prompts against the firm's earlier reports on the zone, with each point traced to the report it came from.

Operations. Internal HR runs on a separate instance. Employees ask the policies and benefits collection the questions they'd otherwise take to the HR manager, such as which medical plans they can choose from, and the answer comes back with the policy it came from. Each client engagement runs in its own instance too, open only to the people working on it.

None of this is a measurement. It's what one firm does with the system week to week.

CHAPTER 7

Workflows.

Most AI tools assume work starts when you open a chat window and ends when you close it. Work doesn't happen in single questions. The Tuesday pipeline review happens every Tuesday. The auditor's request arrives every quarter with the same shape and different dates. Work repeats and accumulates, and an AI tool either fits that rhythm or gets abandoned by February.

Not every question deserves the same depth, and the source under each answer makes depth a choice. A project manager double-checking a date glances at the source and moves on. A lawyer building an argument opens that source, reads the highlighted passage, and follows it into the surrounding sections. A compliance officer facing an auditor produces the

full report, sources attached, without doing anything extra. The answer is the same each time; the reader picks the level of verification the stakes require.

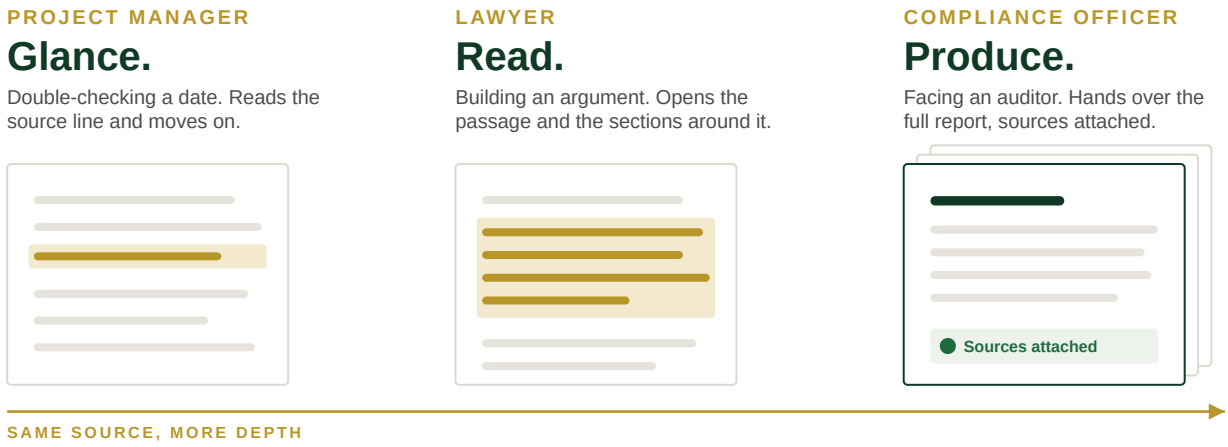


Figure 5. One source, three depths. Every answer carries its source, and the reader opens as much of it as the job calls for.

Work also moves between people, and the source moves with it. A junior analyst runs the first pass and flags what needs judgment, each flag carrying its source, so the senior person spends attention only where it's needed. That division of labor used to be impossible, because the person who found the material had to be the person qualified to evaluate it. When every finding arrives with its evidence, finding and judging can belong to different people. And when a project changes hands entirely, the saved searches, reports, and question logs go with it. The next person inherits the work itself.

Some work repeats. A contracts manager thinks through a search for indemnification language once, saves it, and reruns it against each new batch of agreements; the hard part happened the first time. A business development team goes further and takes themselves out of the loop: their saved filters run against the solicitation stream on a schedule, and the morning starts with a review of what came in overnight.

The knowledge base never goes stale mid-workflow. New documents are in every saved search the day they're ingested. Nobody stops to retrain a model, and the search built last year runs against everything the organization has learned since.

CHAPTER 8

Fit and boundaries.

Most AI products claim to fit everywhere. This one is built for specific kinds of work, and there are jobs it shouldn't do at all.

The sweet spot is unstructured text and the concepts inside it—contracts, policies, training materials, research, correspondence, records. Structured numeric data is a different job. An aggregation over a clean table belongs in the systems built for querying tables, and pointing concept-based retrieval at it adds a layer nobody needs.

It answers from what you give it. That's the point of the design, and it's a boundary to understand before buying anything in this category. The system won't supplement your content with outside knowledge to fill a gap, so a question whose answer was never in your documents comes back unanswered. Drafting, brainstorming, and composing from general knowledge are a language model's job, and the wrong thing to ask of a system built to tell you what your own material supports.

How rare is rare. Because answers are retrieved from curated, ingested content rather than generated from patterns, fabrication is rare. We won't tell you it's impossible, and you should be wary of any vendor in this market who will.

The concept map comes largely built. For general and professional language, that structure works in your favor. A field running on deeply specialized internal vocabulary gets there through the domain layer, and the domain layer is curated work.

Symbolic layers need an owner. Kautz's bottleneck never disappears completely. The core base answers it for general knowledge, and the domain and learned layers stay healthy only if someone maintains them. Curation costs far less than a retraining cycle, but it isn't free, and you should hear that from the people selling you the system.

CHAPTER 9

Language models and world models.

A growing camp of researchers argues that predicting language, however well it's done, can't be the destination, because language describes the world without modeling how it works. The term gaining ground for what comes next is the world model.[15]

	LANGUAGE MODELS	WORLD MODELS
Primary focus	Language understanding and generation	Environmental dynamics and causality
Input and output	Mostly text and code	Sensory data, internal state, actions
Knowledge	Implicit in linguistic patterns	Explicit or implicit model of how things work
Core task	Predict the next word or token	Predict the next state given actions

An explicit representation of concepts and the connections between them is a model of how things work. It lives in language rather than sensory data, and the knowledge is open to inspection rather than buried in weights. By that description, this architecture is on the world-model side.

Questions to ask any system you're evaluating.

Every product in this category demos well. The differences show up under hard questions, asked of every vendor, including this one.

1 **Where did this answer come from?**

Can the system show you the source, or is the citation generated by the same process as the answer?

2 **What happens when the answer isn't in the material?**

Does the system say so, or does it produce something plausible?

3 **Can you see how the answer was reached?**

Is there an inspectable account of the reasoning, or does explanation mean asking the model to explain itself?

4 **What does adding knowledge cost?**

Is it an ingestion step, or a retraining cycle with a labeled dataset behind it?

5 **Does your content train someone else's model?**

Get the answer in writing.

6 **What happens on structured data?**

A confident prose answer to a numeric question deserves suspicion. Ask how tables are handled.

7 **What breaks on messy input?**

Ask about scans, incomplete records, and inconsistent formats, because demo document sets are always clean and yours never will be.

8 **How is quality measured?**

Ask for the evaluation method.

9 **Who owns the knowledge structure?**

When the whole stack is someone else's model, ask what, exactly, the vendor owns.

Closing.

Enterprise AI is settling on one idea: the model is the interface, and the value is in the knowledge layer underneath it. What that layer should be made of is still an open question. Retrieval plumbing over text chunks is one answer. An explicit, inspectable representation of meaning is another.

Human thinking runs two systems together. AI built for consequential work needs both: neural methods for recognizing patterns, and symbolic structure for storing knowledge and reasoning over it.

Glossary.

Neural network. A system that learns statistical patterns from examples. The machinery behind modern pattern recognition, and the System 1 side of the split.

Large language model (LLM). A neural network trained on very large amounts of text to predict likely continuations. Generation, summarization, and conversation are its home ground.

Vector embedding. A representation that places a word or passage at a position in mathematical space so that similar things end up near each other. It powers semantic search and carries no explicit meaning.

Symbolic AI. The tradition that represents knowledge explicitly, as concepts, relationships, and rules, and reasons over them deterministically. The System 2 side of the split.

Knowledge graph. A structure that stores things as nodes and relationships as edges, so that what connects to what is stored explicitly rather than inferred.

Ontology. The layer of a knowledge structure that says what kinds of things exist and how types relate. The foundation ontology in this architecture works like an embedded dictionary, thesaurus, and mini encyclopedia.

Identigen. One distinct meaning a word can carry, stored as its own unit. “Bat,” the animal, and “bat,” the equipment, are separate identigens sharing a spelling.

Entigen. An interpreted meaning: a complete semantic unit assembled from identigens by interpretation rules, capturing what a phrase, sentence, or passage means. The name compresses “entity generation.”

Neurosymbolic AI. An architecture that runs neural and symbolic methods together, using each where it's strong. Also called concept-based AI in this paper.

Retrieval-augmented generation (RAG). Supplying a language model with retrieved documents at question time so it answers from them rather than from training alone. Improves grounding; retrieval and generation each have their own failure modes.

Hallucination. Fabricated output delivered with the same fluency as accurate output.

Abstention. A system declining to answer because the material doesn't support one. The behavior you want in place of a confident guess.

World model. A system whose knowledge models how things work, explicit enough to reason over, rather than a pattern over language.

Appendix A: References.

[1] MacLean, P. D. The Triune Brain in Evolution: Role in Paleocerebral Functions. New York: Plenum Press, 1990.

[2] Kahneman, D. Thinking, Fast and Slow. 2011.

[3] Domingos, P. The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World. 2015. Presented at ACM TechTalk, 24 November 2015.

[4] DeLong, L. N., Fernández Mir, R., and Fleuriot, J. D. “Neurosymbolic AI for Reasoning Over Knowledge Graphs: A Survey.” IEEE Transactions on Neural Networks and Learning Systems, 2024. DOI 10.1109/TNNLS.2024.3420218.

[5] Gutiérrez, B. et al. “HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models.” NeurIPS 2024.

[6] Xu, Z. et al. “Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering.” LinkedIn, 2024. arXiv:2404.17723.

[7] Agrawal, G. et al. “Can Knowledge Graphs Reduce Hallucinations in LLMs? A Survey.” NAACL 2024.

[8] Diffbot / FalkorDB. “KG-LM Accuracy Benchmark.” 2023–2025.

[9] Huang, L. et al. “A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions.” ACM Transactions on Information Systems, 2025. DOI 10.1145/3703155.

[10] Wu, C.-J. et al. “Sustainable AI: Environmental Implications, Challenges and Opportunities.” Proceedings of Machine Learning and Systems (MLSys), vol. 4, pp. 795–813, 2022.

[11] Kautz, H. “The Third AI Summer.” AAAI Robert S. Engelmore Memorial Lecture, 34th AAAI Conference on Artificial Intelligence, 10 February 2020. Published in AI Magazine, vol. 43, no. 1, 2022, pp. 105–125.

[12] Cambria, E. and White, B. “Jumping NLP Curves: A Review of Natural Language Processing Research.” IEEE Computational Intelligence Magazine, vol. 9, no. 2, 2014, pp. 48–57.

[13] IDC. Worldwide Global DataSphere estimates; unstructured share of organizational data, 2022.

[14] Regulation (EU) 2026/1744 (the AI Omnibus). Official Journal of the European Union, July 24, 2026; in force July 27, 2026.

[15] LeCun, Y. "A Path Towards Autonomous Machine Intelligence." OpenReview, June 2022.

Further reading.

Alansari, A. and Luqman, H. “Large Language Models Hallucination: A Comprehensive Survey.” arXiv:2510.06265.

Hammond, K. and Leake, D. “Large Language Models Need Symbolic AI.” CEUR-WS.org/Vol-3432/paper17.pdf

Wan, Z. et al. “Towards Efficient Neuro-Symbolic AI: From Workload Characterization to Hardware Architecture.” IEEE Transactions on Circuits and Systems for Artificial Intelligence, September 2024.

Deerwester, S. et al. “Indexing by Latent Semantic Analysis.” Journal of the American Society for Information Science, vol. 41, no. 6, 1990, pp. 391–407.

Appendix B: Patent coverage.

DOMAIN	COVERAGE
Semantic interpretation	Methods for determining the meaning of word strings, phrases, and documents
Knowledge extraction	Processing content to build structured knowledge representations
Query processing	Interpreting queries and producing embellished responses
Neuro-symbolic retrieval	Hybrid methods combining neural and symbolic retrieval
Information integrity	Validating the accuracy and consistency of information
Knowledge graphs	Building and traversing semantic relationship structures

Selected patents.

PATENT	TITLE
US11442993B2	Processing a query to produce an embellished query response
US20190325027A1	Interpreting a meaning of a word string
US9588963B2	System and method of grouping and extracting information from data corpora
US9063923B2	Method for identifying the integrity of information
US12430371	Product-service query interpretation

Full portfolio available under NDA.

Franklin Discovery™, FranklinWatch™, FranklinAuthor™, and FranklinTrainer™ are trademarks of entigenlogic LLC.

About Franklin Discovery™ by entigenlogic®.

Franklin Discovery™ answers only from the documents you approve and shows you where every line came from. When the answer isn't in your documents, it says so. You don't have to hope it's right. You can check that it is.

Your data stays your data. Nothing you ingest and nothing you ask becomes training for someone else's model. Franklin Discovery™ is the one tool you can stake your reputation on.

Learn more at franklindiscovery.ai.