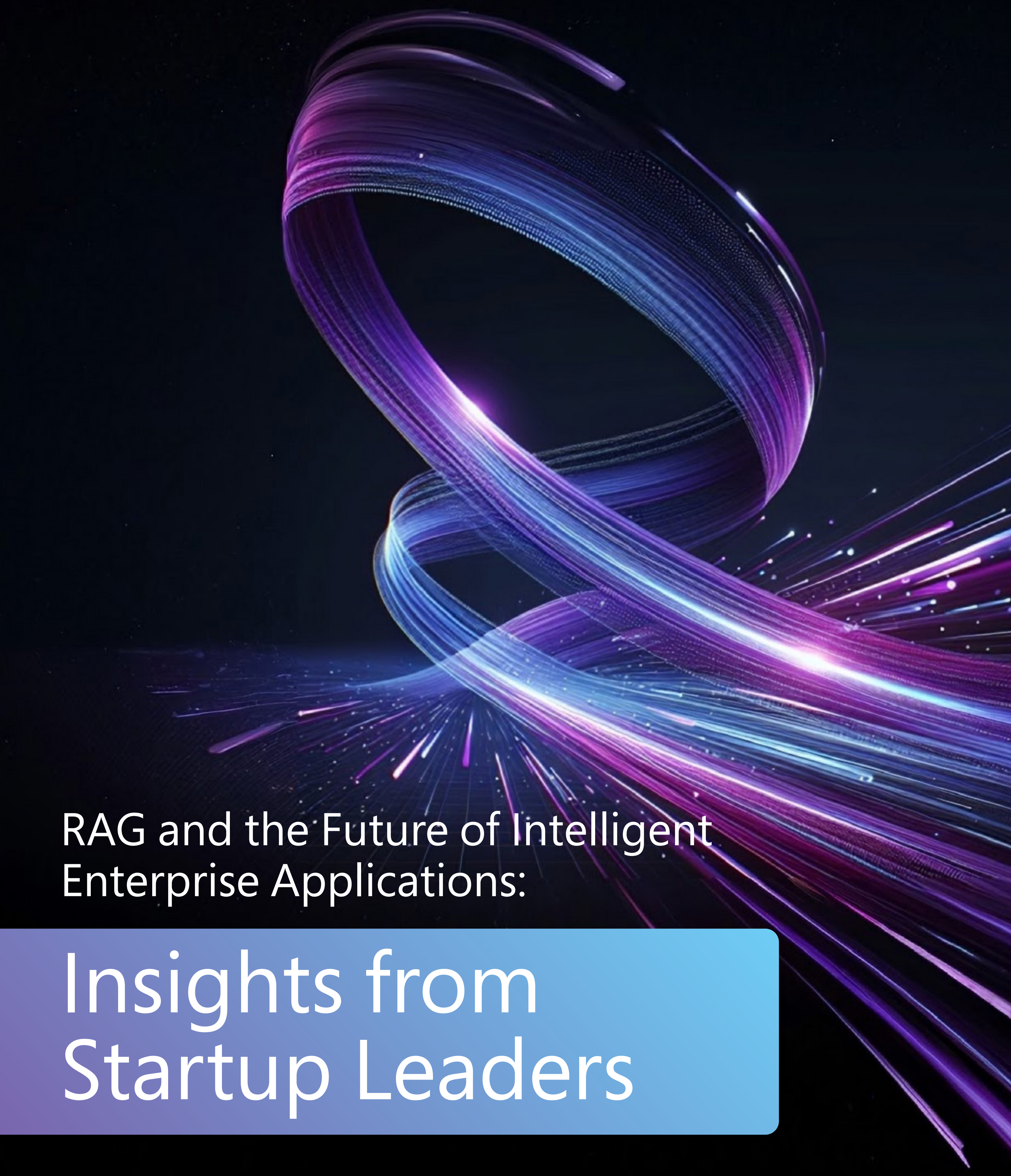
Microsoft | Microsoft for Startups



RAG and the Future of Intelligent Enterprise Applications:

Insights from Startup Leaders

What This Paper Is About, Who It's For, and What Reading It Will Do for You

AI has never been so powerful, but scaling generative AI (GenAI) applications to enterprise-grade production is hard. This white paper is intended to be a field guide for navigating the numerous hurdles specific to building robust GenAI applications. Through dozens of interviews with buyers, AI practitioners, and founders, authors Rob Ferguson and Nick Giometti have assembled a set of beliefs and best practices for building production-ready AI systems.

While there's no "one-size-fits-all approach" to building GenAI applications, this collection of practical advice centers on the assertion that Retrieval-Augmented Generation (RAG) is the best methodology for marrying enterprise-specific context to the emergent capabilities of language models. By the end of this paper, the authors want readers to understand three key areas:

- **Outcomes:**
  What is RAG, and how does it drive value in GenAI applications?
- **Challenges:**
  What are the biggest pains preventing GenAI systems from reaching production?
- **Solutions:**
  How are today's leading startups solving these problems, and how can enterprises partner with them to build toward the future of intelligent applications?

This paper consists of **three chapters and a conclusion.**

In the first chapter, author Rob Ferguson establishes the **current state of enterprise GenAI adoption** and provides a **brief primer on RAG.**

Next, author Nick Giometti posits that scaling pains can be mapped to **three common domains,** each representing significant investment opportunities for startups to overcome: **mastering context, building trust, and incorporating feedback.**

In the third chapter, eight startup leaders each **share a lesson** from their hard-earned perspectives building in GenAI, and demonstrate how their solutions **empower enterprise buyers to create their own intelligent systems.**

Lastly, the conclusion highlights the components to prioritize for successful intelligent applications and closes with predictions for how enterprises can build **future-proof AI scaffolding** to match the ever-evolving capabilities of language models.

8 Lessons from Founders

1 **Don't Let Perfect Be the Enemy of Production:** Start with simpler, cheaper embedding models and basic retrieval before jumping to complex language models or architectures.

2 **Data Quality is the Largest Hurdle:** Your existing business data likely needs significant preprocessing and restructuring to work effectively with AI systems.

3 **AI Techniques are Adaptable:** Modern AI systems can handle multiple types of data (text, images, audio) simultaneously, expanding their potential business applications.

4 **Using Humans to Build Minimum Viable Preference:** Human feedback is crucial for establishing quality benchmarks before exposing AI systems to customers.

5 **Evolving Through Production-driven Development:** Real-world usage data and feedback should drive your AI system's evolution rather than theoretical improvements.

6 **Performance Depends on the Whole System:** Success requires integrating AI with your existing technology stack, not just implementing standalone AI models.

7 **RAG Isn't a Single Technology:** RAG should be viewed as a complete system of integrated components rather than a single plug-and-play solution.

8 **Enforce Governance by Building for the Right Trade-offs:** Balance the power of AI search capabilities with appropriate access controls and data governance from the start.

Ready to explore advanced AI solutions for your startup?

Learn more about Microsoft for Startups and kickstart your journey. →

Index

Chapter 1

Chapter 2

Chapter 3

Conclusion

About Microsoft for Startups

Chapter 1:
Getting to GenAI-Native Applications

Co-Author: Rob Ferguson
Head of AI for Microsoft for Startups

Introduction

This white paper started with a simple mission. My co-author Nick Giometti and I wanted to discover which AI startups are delivering meaningful technology for the enterprise by using the latest generative AI (GenAI) models. We spoke with dozens of startups and were amazed by the creativity and insights from those on the cutting edge. What you're reading is the result of months of learning about the essential components of building successful AI applications.

We delve into the experiences of leading AI startups to give enterprises a clear understanding of Retrieval-Augmented Generation (RAG). By exploring real-world applications, challenges, and best practices, you'll gain actionable knowledge to navigate the complexities of RAG integration, make informed decisions about AI adoption, and strategically position your organization for success in building intelligent enterprise applications.

The adoption of GenAI in enterprise environments is moving fast but is still in its early days. While many enterprises have experimented with AI technologies, most implementations remain pilots or proofs of concept (POCs) rather than full-scale deployments. In this chapter, we'll explore the current landscape of GenAI adoption and the emergence of custom AI copilots, and introduce a framework for understanding the horizons of GenAI technology integration.

Adopting Copilots

According to the Microsoft Trend Index, 59% of employees are bringing their own AI tools into work. While this raises concerns for business leaders about safely and responsibly integrating AI technology, it highlights the eagerness of employees to leverage AI in their workflows. While 42% of enterprises reported using AI in some capacity, the majority of these implementations aren't yet fully integrated into enterprise operations.

So far, most people experience GenAI through various copilots, like ChatGPT, Bing Search, and GitHub Copilot. However, startups building custom copilots have seen incredible interest and traction. Early leaders like Harvey, Sierra, and Glean have each raised over $100 million, achieving "unicorn" status.

Custom copilots created by startups are an amazing way to discover what these GenAI models can do. They excel when adapting to existing workflows or integrating to solve long-standing problems. Imagine being a lawyer who had to summarize 10,000 legal documents before GenAI came along. That's a game-changer.

But as incredible as these custom copilots are, they can still hallucinate facts, they can be expensive and integrating them into complex enterprise environments often comes with challenges. The most common criticism is simple: a bad copilot promises it can do everything, but it doesn't do any of those things well.

Three Horizons of GenAI Adoption

As the Head of AI for Microsoft for Startups, I, together with my team, not only help the world's best startups use the best of Big Tech but also help Big Tech use the best of startups. Through this experience, we've developed a framework for understanding how GenAI technology is being adopted. We call it the "Three Horizons of GenAI Adoption."

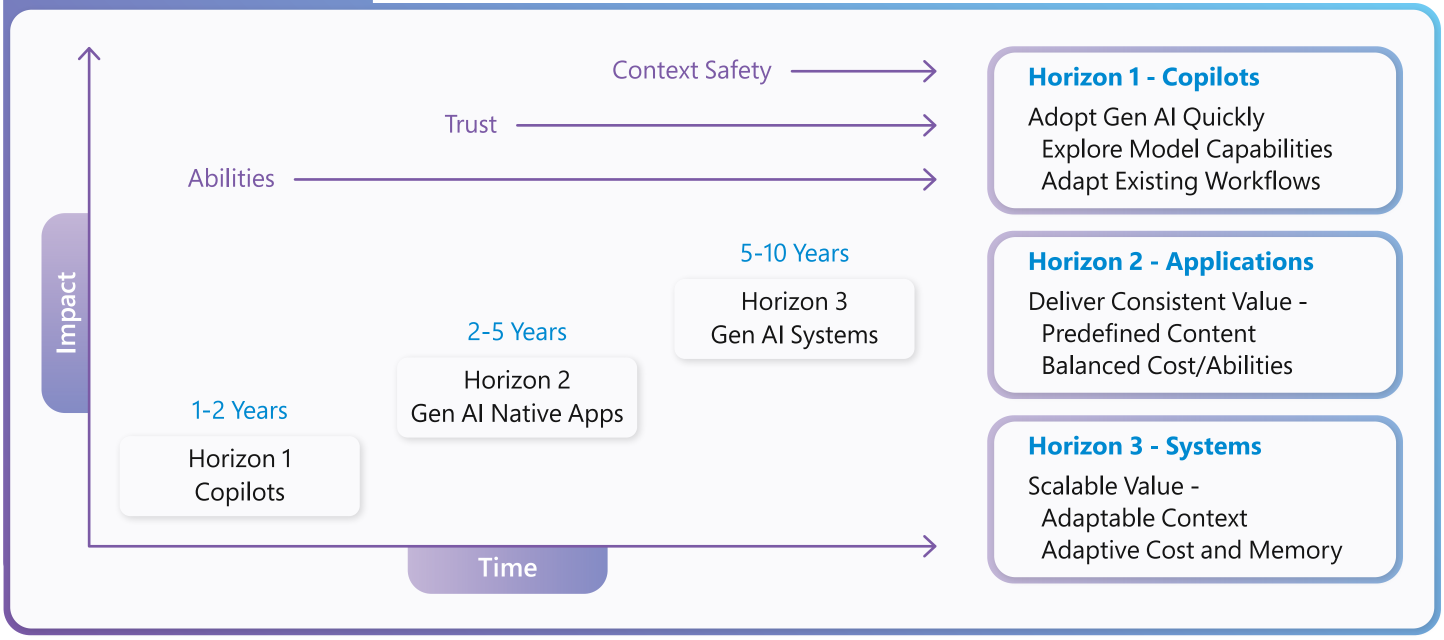
What we've noticed is that the best GenAI companies are really good at controlling the context that GenAI is exposed to during a workflow. They understand the data a user is working with and how they're working with it. They then perform an incredible balancing act of matching the capabilities of GenAI models within the user's specific context.

We consider these standout applications "GenAI-native applications" when they're predictably successful while balancing costs. (I've seen many cool demos where you wouldn't believe the cost to run at scale.) The key is that GenAI-native applications limit context to avoid over-promising and hallucinating information that isn't within the models' priors.

Open-ended copilots are an incredible technology, but they're best at augmenting an individual's success. GenAI-native applications scale across workflows.

Looking toward the final horizon, we see a tremendous future for agents that work together across application boundaries. These aren't just autonomous agents that can deliver a result on their own. In this stage, which we call "GenAI systems," the value of GenAI scales between enterprises by planning operations while safely sharing memory, encapsulating the cost of tasks, working within a secure plugin or API architecture, and explicitly addressing trust and safety concerns. At the final horizon, many experts (myself included) wonder if the value of GenAI could scale to match the entire software industry.

Gen AI Horizons



Context Safety

Trust

Abilities

Impact

5-10 Years

Horizon 3
Gen AI Systems

2-5 Years

Horizon 2
Gen AI Native Apps

1-2 Years

Horizon 1
Copilots

Time

Horizon 1 - Copilots
Adopt Gen AI Quickly
Explore Model Capabilities
Adapt Existing Workflows

Horizon 2 - Applications
Deliver Consistent Value -
Predefined Content
Balanced Cost/Abilities

Horizon 3 - Systems
Scalable Value -
Adaptable Context
Adaptive Cost and Memory

Understanding RAG: A Primer

Whenever you see a copilot cite its sources, chances are that RAG techniques are at work. Many people, after studying RAG, might think, "Is this just fancy search?"

It's easy to get caught up in the latest innovations. However, the real value of any technology lies not in its novelty but in its ability to solve real-world problems and improve business operations. RAG represents only a fraction of what gets built in intelligent enterprise applications, but it plays a crucial role in enabling businesses to understand how the information is retrieved and integrated.

Understanding RAG isn't about chasing the latest tech trends. It's about recognizing a powerful tool that can deliver tangible improvements to workflows and decision-making processes. By augmenting large language models (LLMs) with the ability to dynamically retrieve and incorporate relevant information, RAG addresses a fundamental challenge in enterprise AI: combining the broad capabilities of AI with the specific, up-to-date knowledge that businesses rely on.

What Is RAG?

RAG enhances the capabilities of LLMs by dynamically incorporating external information during the generation process. Think of it as adding search (or retrieval) capabilities to your LLM. This approach bridges the gap between the vast knowledge embedded in LLMs and the specific, current information needed for accurate and contextual responses.

At its core, RAG is a process. When presented with a query, the system first retrieves a curated knowledge base and then uses this information to augment the LLM's response. This process can be broken down into three main steps.

RAG at a Glance

- **Retrieval:** Collect up-to-date data.
- **Augmentation:** Combine real-time data with GenAI models.
- **Generation:** Produce accurate, context-aware results.

Why Enterprises Should Adopt RAG

RAG addresses several critical limitations when building applications with LLMs.

Increased Accuracy and Relevance

Leveraging real-time data to reduce misinformation and outdated knowledge.

Domain-Specific Customization

Tailoring AI solutions to address unique industry challenges.

Scalability without Retraining

Enabling efficient updates to knowledge bases without the need for frequent model retraining.

Transparency and Explainability

Promoting trust in AI by clearly showing data sources, enhancing decision-making processes.

Conclusion

For now, if you still think of RAG as "fancy search" that helps a GenAI model cite its sources, that's perfectly fine. We'll dive deeper into the specifics in Chapter 3. For now, I'll hand it off to Nick to explain why he invests in infrastructure startups that build with RAG technologies in mind.

Chapter 2:
B Capital's Bet on the Future of Enterprise Intelligence

Co-Author: Nick Giometti
Senior Principal at B Capital

Key Themes

- At B Capital, we're betting on enterprise incentives to create a vast distribution of specialized intelligences rather than rely on a singular artificial general intelligence (AGI). **The future of intelligence is contextual, not general.**
- **Scaling intelligent applications is hard.** The model layer is only a single component in a larger foundation of opinionated application and infrastructure design choices. Defining best practices in GenAIOps means wrangling interdisciplinary concepts across all of DevOps, MLOps, as well as LLMOps.
- The **largest investment** opportunities exist for companies building the essential abstractions that empower enterprises to build intelligent systems, which:

- Master enterprise context.
- Build and maintain trust.
- Drive continuous improvement.

No Singular Intelligence

What's the enterprise incentive for artificial generative intelligence (AGI)? While AGI captivates the imagination, it's hard to reconcile our current economic constructs with a world where every job has been displaced by the "one model to rule them all." What happens to corporate competitive advantage when every enterprise is hiring and selling services generated by the same universally capable AI worker?

At B Capital, we envision a future shaped not by a monolithic general intelligence, but by a diverse ecosystem of highly specialized intelligences. We believe enterprises will thrive by leveraging their proprietary data and domain expertise to build artificial contextual intelligence, rather than relying on generalized models.

By tailoring models, knowledge bases, and retrieval systems to their specific domains, enterprises can:

- Maximize the value of their unique expertise
- Drive superior returns on technology investments
- Maintain and enhance their competitive advantage

This approach aligns AI development with business objectives, ensuring that advancements in artificial intelligence augment rather than replace human capabilities, fostering sustainable growth and innovation.

Mastering Context

In the world of intelligent enterprise applications, **context is king.** Not only does context **enhance and filter** the performance of general models to **achieve specialized outcomes,** but it also serves as a guiding design principle: when building for tomorrow, **context provides a means of backwards induction.** The delta between the state of today's data, processes, and infrastructure and the desired future outcomes **creates a map for filling production gaps.**

For AI systems to augment or automate productivity in a manner that approximates and eventually surpasses an enterprise's human workforce, they need to fully capture the specifics of its unique domain. This contextual mesh includes its language, workflows, regulatory environment, and unique value propositions.

Because enterprise context is constantly evolving, mapping this complex web is no simple task. Businesses must integrate vast, multimodal, unstructured, and dynamic context into their operational use cases to achieve production-ready outcomes. Given the critical yet challenging nature of capturing this context, we see significant value creation opportunities for startups that simplify these complexities.

Investment Opportunities in Context Management

- Ingesting, structuring, and refining domain-specific data
- Developing custom knowledge graphs and fine-tuned embeddings
- Optimizing retrieval strategies to anticipate user-specific and use case–specific context

Startups that navigate, curate, and adapt each enterprise's unique context form the basis of a **portable, Retrieval-Augmented Generation (RAG)–based AI scaffolding.** While RAG is the fastest way to bring enterprise data into a model's context window, the complexity arises from transforming data into a functioning knowledge base. While knowledge is heterogenous, we believe critical RAG infrastructure will need to answer a core set of questions, like:

- Where's my data, and who's responsible for maintaining it?
- How do I make my data machine-interpretable without losing context?
- How do I make my system better at answering my users' most important questions?

RAG is an immediate and lasting value driver because, regardless of what new model enhancements emerge, maintaining a dynamic collection of each enterprise's context allows businesses to rapidly experiment toward production. Isolating context as a constant and varying the model allows businesses to maximize AI-driven revenue by identifying more valuable use cases and outcomes, and to minimize costs by optimizing for the cheapest model without sacrificing performance.

While building intelligent applications feels like both a marathon and a sprint, the best advice for both distances is to **race your own race, and control what you can control.** For most enterprises, what new models can do is a function of frontier research labs. Instead, mastering the context through which AI is applied becomes the highest-impact behavior. At this stage of GenAI adoption, **the startups that abstract the context-control plane excite us the most.**

Building Trust through Transparency and Reliability

In any enterprise environment, AI adoption depends foremost on **trust.** If employees and customers don't trust an AI's decisions, they will hesitate to use it, wasting developers' time and resources.

Trust is built on two pillars: **transparency**—how the AI arrives at its conclusions—and **reliability**—consistent, unbiased, secure performance. Enterprises need assurance that AI systems provide correct information, but they must also be able to prove they do so ethically, legally, and securely.

Building trust is no easy task: increased expectations brought on by impressive demos burden builders as they attempt to scale these evolving capabilities to production. A developer who has built the safest possible system still has to contend with the user's attention span of 5,000 milliseconds or less. And because users won't trust any application that fails four or more times in rapid succession, the trust window between alpha and production is incredibly small. Old problems intersecting with new models require a new standard for trust.

Building trust into the next generation of intelligent applications doesn't just have to overcome classical problems related to safety and robustness—role-based access control (RBAC), managing personally identifiable information (PII), and latency. It also faces the added challenge of trying to wrangle powerful nondeterministic engines whose emergent behaviors aren't well understood.

It can take months of iterating from experimentation to production to gain a user's trust, but only a moment to lose it. Unfortunately, while the path to earning trust is narrow, the branches leading to betrayal are wide. GenAI's emergent capabilities present a paradox: we love solutions that offer creative ways to solve problems, but we fear what we can't control.

Earning trust means adhering to a strict contract of expected behaviors; and depending on the use case or industry, those expectations vary widely. That said, we believe there are **core dependencies for building trustworthy intelligent applications.** The end goal of building effective guardrails is to ensure that the right data is surfaced to **the right user at the right time.**

Investment Opportunities in Reliability and Transparency

- Explainability and interpretability tools
- Governance and compliance frameworks
- Data privacy and security solutions

Large language models (LLMs) are probabilistic, making it challenging for an AI assistant to explain how it arrived at its answer. Even with chain-of-thought prompting, where the model walks through its 'reasoning,' it is still generating the most probable next token in sequence. Rather than revealing the underlying weights and biases encoded in a neural network of billions of parameters, the model demonstrates how it 'thinks' a similar prompt might be answered by another model—without offering a direct window into its internal mechanics.

If we can't control the "why," then building trustworthy outputs becomes a function of **controlling the "what" and the "how."** Without fine-tuning a base model to align it with some guidelines for moderation, governance shifts to critical application and infrastructure choices.

How does shifting transparency and reliability to an approach driven by "how" and "what" guardrails manifest in application design? These decisions materialize through questions like:

- What data sources should a RAG application have access to and should RBAC be handled at the user or source level?
- What's the threshold for flagging harmful prompts or escalating them to security teams?
- When can cached responses be used to save on cost and generate faster answers?

Choices to answer these questions come with **trade-offs.**

- Should guardrails be handled at each node in an application or just before an answer is generated?
- Should intelligent applications be accessed through a single gateway or individual walled gardens?

We're excited to support startups that simplify these complexities, enabling enterprises to quickly establish their own standards for trust, reliability, and transparency.

Feedback-Driven Systems

Lastly, AI systems, like the businesses they serve, need to evolve continuously to stay relevant and effective. The real world is dynamic: markets shift, customer preferences change, and regulations evolve. This is where **feedback-driven systems** come in. These systems are designed to learn from their users, their environments, and their outcomes. Feedback loops allow AI to adapt, improving its performance over time and aligning more closely with the enterprise's evolving needs.

Because GenAI is still in its nascent phase of human-machine interactions, we believe that today's collaboration with artificial intelligence will appear primitive to our future selves. As we're only just beginning to understand what these models are capable of, developers face the added challenges of creativity and preference: how do we know what we want if we've never seen it before?

Steve Jobs famously said, "You can't just ask customers what they want and then try to give that to them. By the time you get it built, they'll want something new." In a recent Tweet, AI researcher Andrej Karpathy evolved on this sentiment by describing the future of intelligent applications as "Input Optional Products." He states:

> *Don't ask your users for input. Coming up with input is hard, and a barrier to use. Think of users as wanting to play. We have AI – predict the input! Design product into autonomous environments. Allow users to play by steering a bit.*

The key to building intelligent applications that humans readily accept with little to no intervention lies in the iterative capture of preference data.

Input may be optional, but feedback is essential.

Startups that focus on incorporating preference into the model and application infrastructure will accrue value, as they enable enterprises not just to scale their intelligent systems to production, but also to evolve over time.

Investment Opportunities in Feedback-Driven Systems

- Implicit and explicit feedback collection tools
- Active learning frameworks
- AI performance monitoring and evaluation platforms

Abstracting feedback systems is a multifaceted challenge: answering 'Is this a good answer?' is entirely different from answering 'Is this a good experience?' Additionally, feedback systems must accommodate increasingly nuanced preferences. Early GenAI applications relied on simple thumbs-up or thumbs-down buttons, but today's intelligent systems, integrated into browsers or IDEs, enable users to edit outputs directly in-line. The difference between the original machine-generated output and the human-edited version can now be evaluated across dimensions like tone, complexity, and syntax.

As intelligent applications advance, so too must the mechanisms for capturing and applying feedback. If we are defined by what we measure, these startups will play a pivotal role in shaping what intelligent systems can ultimately become.

Scaling to the Future of Intelligence

Through our interviews, we discovered that these three themes—**mastering context, building trust,** and **enabling feedback-driven evolution**—are the cornerstones of scaling AI in enterprise environments. The leaders we spoke with each create software abstractions to simplify these complex challenges and are well-positioned to become essential partners for companies looking to implement intelligent enterprise applications. As AI moves from demo environments to mission-critical business operations, mastering these dimensions will be key to success, both for enterprises and for the startups that help them get there.

All information is as of 3.17.2025 and subject to change. Certain statements reflected herein reflect the subjective opinions and views of B Capital personnel. Such statements cannot be independently verified and are subject to change. Reference to third-party firms or businesses does not imply affiliation with or endorsement by such firms or businesses. It should not be assumed that any investments or companies identified and discussed herein were or will be profitable. Past performance is not indicative of future results. The information herein does not constitute or form part of an offer to issue or sell, or a solicitation of an offer to subscribe or buy, any securities or other financial instruments, nor does it constitute a financial promotion, investment advice or an inducement or incitement to participate in any product, offering or investment. Much of the relevant information is derived directly from various sources which B Capital believes to be reliable, but without independent verification. This information is provided for reference only and the companies described herein may not be representative of all relevant companies or B Capital investments. You should not rely upon this information to form the definitive basis for any decision, contract, commitment or action.

Chapter 3: Lessons from the Innovators

In our quest to understand the real-world applications and challenges of implementing intelligent systems in enterprise environments, we went on an extensive research journey. We had the privilege of interviewing dozens of startups at the forefront of AI and data-augmented applications. Through these conversations, we gained invaluable insights into the practical considerations, innovative approaches, and emerging best practices in this rapidly evolving field.

From the wealth of information we gathered, we've distilled key lessons from eight standout startups. Each of these companies offers a unique perspective on how to effectively leverage technologies like Retrieval-Augmented Generation (RAG) and other AI-driven solutions to address specific business challenges. Their experiences and insights provide a ground-level view of what it takes to build and deploy intelligent applications that deliver real value in enterprise settings.

Lesson 1: Don't Let Perfect Be the Enemy of Production

Enterprises prioritize maximizing profit and return on investment, which often means optimizing existing operations. But when it comes to generative AI (GenAI), even the most advanced companies face the same challenge: you can't optimize what isn't in production. The sooner businesses deploy their GenAI applications and gather real customer feedback, the sooner they can decide whether to refine and scale the product or pivot to something new.

In this section we'll cover what it means to build RAG-powered applications toward minimum viable production.

A Brief Aside on Vector DBs

We didn't interview a single "vector database (DB) company" while writing this white paper.

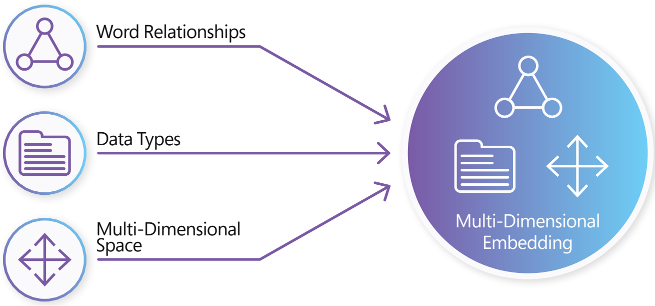
Vector databases are undeniably important. However, while developers often start by selecting a vector DB and identifying the ideal reference architecture for their intelligent applications, this approach may overlook key considerations. It doesn't necessarily help developers grasp the trade-offs involved in building GenAI-native applications.

We believe the first step should instead focus on understanding the underlying embeddings that define your domain space and the mechanics of retrieving them effectively.

A Primer on Embeddings: Are Vectors Just Vibes?

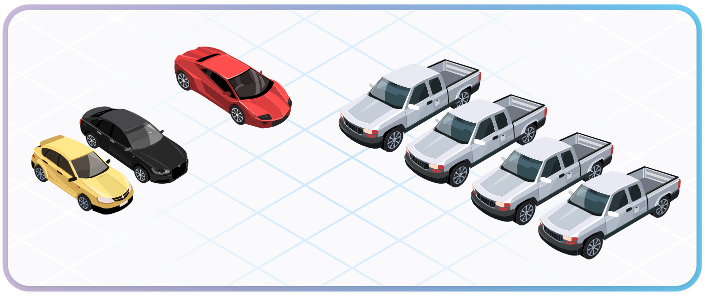
At the heart of building modern AI are unique digital fingerprints, known as vectors, that turn data like words into unique positions in a multidimensional space. There are lots of ways to capture these sorts of relationships, but any kind of data—including entire documents or even images—can be embedded into a multidimensional vector space. In this space, each entity has a relationship with every other one, no matter how small.

Multi-dimensional Embedding



Imagine you walk into a theoretical "multidimensional" showroom. From your initial vantage point, you notice:

- A Honda and a Toyota are parked close together in one area.
- A Porsche is placed slightly apart.
- A row of pickup trucks is in another section.



Right away, you can tell the cars have been logically grouped together.

How to arrange cars in a multi-dimensional showroom?

Group by Brand
Cars of the same brand are parked close together.

Group by Price
Cars of similar price ranges are parked together.

Group by Type
Cars of similar types (e.g., sedans, SUVs) are parked together.

As you walk around to other vantage points, you notice additional groupings that might not have been immediately obvious:

- By fuel efficiency
- By country of origin

Even if you can't discern all these dimensions at once, you understand that these cars relate to each other in multiple ways. The closer they're placed to each other, the more related they are in a particular aspect. The relationship might not be perfectly clear, but you can imagine how a Toyota and a Honda "vibe" together in a way that separates them from a Porsche, even if you're not exactly sure why.

Understanding Car Brand Relationships

Toyota and Honda
Represent a close relationship in reliability and affordability

Porsche
Stands apart due to its luxury and performance focus

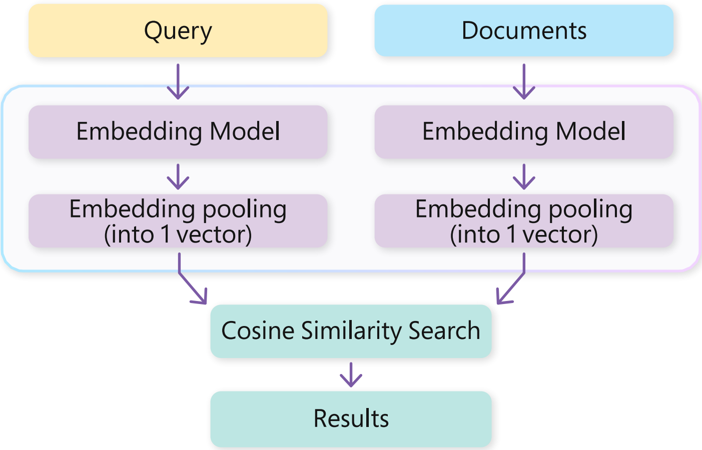
That's the power of embeddings: they capture intricate relationships in a format that computers can easily work with.

Minimum Viable RAG

So how do we "transform" our data so it becomes machine-readable? This is where embedding models come into play. A lightweight embedding model is a specialized tool designed to convert text into these multidimensional representations. In fact, in the simplest possible version of RAG, we can skip the large language model (LLM) altogether and just use embeddings to generate search results.

Defining the Bi-Encoder Approach

1. One process encodes the search words in our query into embeddings.
2. Another process converts our candidate documents into embeddings.
3. We then compare these embeddings by using a measure called "cosine similarity."



The **vector search** result is the document with the closest match across the dimensions; therefore, it's considered the most relevant to the given prompt. In other words, the query searched and retrieved the documents whose "vibes" most closely matched the question's. Depending on how you want to 'augment' the search results, there might be no need for a complex LLM.

Founder Spotlight: LlamaIndex's Approach to Production-Ready RAG

Jerry Liu,
CEO of LlamaIndex

Jerry Liu, CEO of **LlamaIndex**, has empowered thousands of developers to build production-ready GenAI applications through his company's flexible RAG framework. Jerry believes that, while optimization is a complex, iterative process, reaching a workable production state is relatively simple. As he sees it, you don't need the most powerful LLM to start seeing significant results, but "can get reasonably far just as a base layer using relatively cheap embedding models" paired with "a query or rewriting layer."

LlamaIndex is an immensely popular open-source toolkit for connecting LLMs to external data sources, with a growing community of contributors to ensure constant improvement and innovation. According to Jerry, "The toolkit is intentionally unopinionated because we want to capture all the best practices and give developers optionality."

Along with vector search, LlamaIndex's framework provides powerful tools for ingesting and indexing various types of data, from structured databases to unstructured text documents. This flexibility allows developers to connect their LLMs to any data source and optimize for speed and accuracy in retrieval.

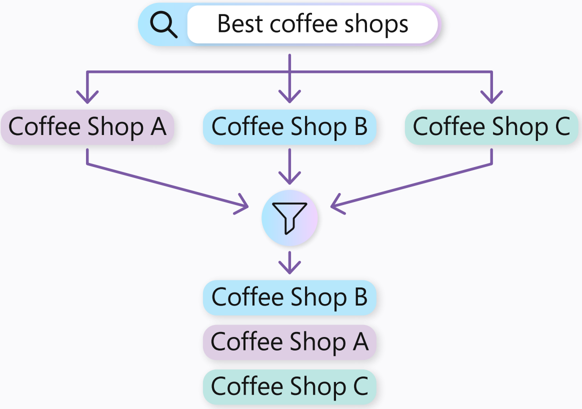
But Jerry sees an important distinction in isolating the performance optimizations that result from tinkering with the retrieval system and those derived from fine-tuning the underlying generative language model.

There are pain points of retrieval and then there are pain points around synthesis. It actually helps isolate these two because a lot of times users have bad retrieval. And when you have bad retrieval, this isn't really an LLM problem anymore. It's just a recommendation systems problem.

Jerry believes that *"you can and should try to optimize your retrieval system" to make sure "you have a good search interface" before putting any "retrieved information into an LLM."*

How can builders optimize retrieval? Jerry highlights several best practices that are available to experiment with in LlamaIndex, both at the prompt level (How can you force the system to ask better questions?) and at the search level (What algorithms are used to find the most contextually relevant information?). Regarding prompting, Jerry has this specific advice: "Take the question and break it down into sub-queries." What was originally a complex question becomes a series of more simplistic searches that are easier to execute.

When it comes to "the retrieval setting," builders should try both "hybrid search," which combines keywords (exact words and phrases) with semantic meaning (contextually similar concepts), as well as "reranking," an approach that refines the order of retrieved documents based on their relevance to a query. Only after improving the performance of the retrieval system should builders focus on gains from tuning language model synthesis.



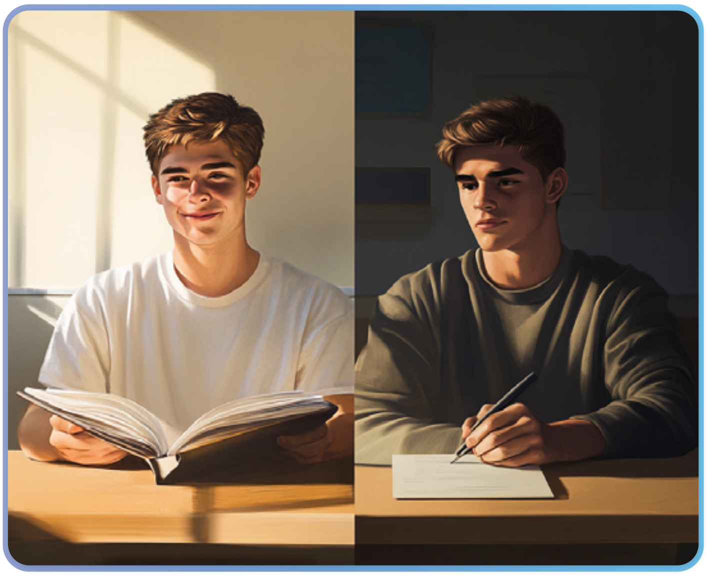
Generation

The “Generation” part of RAG is where the GenAI model comes in.

Notice: “GenAI” model, not “large language model” (LLM). The reality is, there’s no particular need at this stage to use a very large model. Nearly all of the top startups we talked to use multiple models, and they understand where to use them. Saying “GenAI model” will help your business recognize that it’s not always appropriate to use the “large” model.

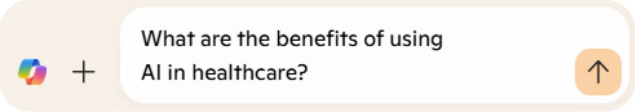
The Generation stage leverages the power of GenAI models to synthesize coherent and contextually relevant responses based on the information retrieved in the earlier stage. Let’s see how retrieval and the GenAI model might fit together with a technique called “in-context learning.”

The Open-Book Exam



1. User query example

Prompt:



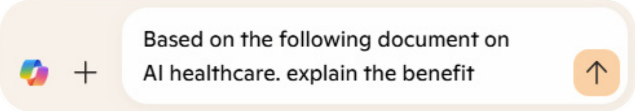
What are the benefits of using AI in healthcare?

2. Document retrieval

Process: The retriever searches for a candidate document that identifies a document (or section) discussing AI applications in healthcare.

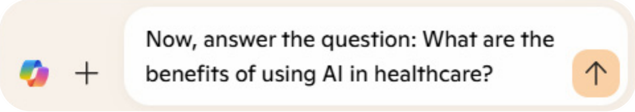
3. Prompt construction

Augmented prompt:



Based on the following document on AI healthcare. explain the benefit

{{insert relevant excerpt or summary}}.



Now, answer the question: What are the benefits of using AI in healthcare?

In this simple example, we include all of the relevant text we found in the retrieval step and squeeze it into one long description (a prompt) of what we want the GenAI model to answer.

When the search process successfully retrieves relevant information, the GenAI model can craft a well-informed answer. Researchers describe this as using information retrieval to augment answer generation—or, more simply, as an “open-book exam.” This analogy is particularly fitting: much like a student consulting their textbook during an exam, the RAG system draws upon external sources to bolster its responses. By contrast, traditional prompting relies solely on the model’s internal knowledge—the information “frozen” within its parameters during training: a “closed-book exam.”

Context Windows

When using RAG in-context, such as in the example above, there’s an upper limit to how much information you can squeeze in at one time. This limit is called the “context window length.” As you can imagine, there’s only so much studying you can catch up with during an open-book exam. If it’s too much, you’ll have to find another way prior to the exam itself.

“We were in the business of inventing a bunch of these techniques in the beginning,” Jerry Liu says of building LlamaIndex, “and as a result, you have common core solutions and different techniques. You have to sift through and figure out the best ones.” In fact, one of LlamaIndex’s major innovations was handling long contexts by summarizing and retrieving the most relevant information from large datasets. This allowed LLMs to work with much larger amounts of data than they could otherwise process in a single session.

In this example, we didn’t use a vector DB at all. If you have a small number of documents with little governance rules, maybe you could skip it?

Lesson 2: Data Quality Is the Largest Hurdle

Even if we could cram all the right books into our "open-book exam," having the right information doesn't always mean we generate the right result.

Common RAG Problems

- Knowledge updates
- Data attribution
- External knowledge
- Data preparation
- Mapping data surface area (what's relevant for in-context)
- Compute resources
- Latency requirements
- Hallucinations

When helping companies navigate these challenges, Jerry Liu says, "Basically, RAG is this sequence of different components, where each component has tunable parameters. And to really optimize the entire system, you have to jointly tune all the parameters at once, which is why there's so many choices in RAG." Jerry describes this as a "combinatorial explosion" that complicates standards from forming: "The downside of having so many techniques is that it becomes very hard for developers to figure out best practices." While the state-space of RAG-related challenges is vast, one area we believe startups are adding the most value in today is reimagining and refining data quality to become compatible with LLMs.

Unstructured Data Is a Major Challenge

RAG data is a really specific kind of data. Depending on how it's parsed and ingested, when data is brought into a context window, the way a human interprets a document might be completely different than the way a language model sees it.

Any AI system is only as good as the data that powers it, and the most common complaint we hear from builders struggling to reach production is the difficulty of processing unstructured data into machine-readable context. For your enterprise, there's a good chance that tables in your documents aren't going to be interpreted correctly by an LLM.

It might be surprising, but the data on the left below is more likely to work with many GenAI models if it's structured in the format on the right.

Retrieved Document

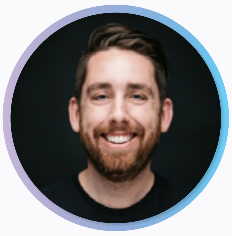
Table 1: Sales by Region (2023)

Region	**Sales ($M)**
North	1,250
South	980
East	1,100
West	1,450
Total	**4,780**

Potential Format in the GenAI Model's Priors

```
<table>
  <caption>Table 1: Sales by Region (2023)</caption>
  <thead>
    <tr> <th>Region</th> <th>Sales ($M)</th> </tr>
  </thead>
  <tbody>
    <tr> <td>North</td> <td>1,250</td> </tr>
    <tr> <td>South</td> <td>980</td> </tr>
    <tr> <td>East</td> <td>1,100</td> </tr>
    <tr> <td>West</td> <td>1,450</td> </tr>
  </tbody>
  <tfoot>
    <tr> <td>Total</td> <td>4,780</td> </tr>
  </tfoot>
</table>
```

Founder Spotlight: Building Next-Gen Data Pipelines with Unstructured



Brian Raymond, CEO of Unstructured

Data preparation has long been a critical challenge for enterprises, but the rise of generative AI (GenAI) applications has introduced new complexities. While Microsoft offers advanced tools like Azure Form Recognizer and Azure AI Document Intelligence for extracting data from documents, these were developed for specific, structured ingestion pipelines. Similarly, intelligent document processing platforms like HyperScience and Instabase excel at handling uniform datasets, such as millions of identical file types with consistent layouts.

However, most GenAI applications are designed to work with highly diverse file types and sources. As Brian Raymond, CEO of **Unstructured**, explains, "They were built for a totally different use case. And the use case was: I have a million documents, but they're all an identical file type and have identical layouts." Today's GenAI-native use cases demand a new generation of data preparation techniques tailored to manage this heterogeneity effectively.

Unstructured provides a powerful platform designed to help companies preprocess and transform their data, regardless of the format (such as PDFs, Word documents, images, and more) or source (data lakes, external applications, or local drives), into formats that are easily digestible by LLMs.

As Brian sees it, chances are that your developers have "an Azure blob that's full of who knows what. Tons of different file types, an infinite number of document layouts, and I just need to get that to a Vector Database."

Addressing data ingestion pipelines for GenAI-native applications presents a significant cold-start challenge, often described as "death by a thousand cuts." The complexity lies in overcoming countless engineering hurdles associated with processing diverse file types and document layouts. To tackle these issues, Unstructured has taken an innovative approach by integrating 500 ingestion libraries into its platform. This comprehensive solution is designed to handle the heterogeneity of file formats and layouts, ensuring seamless data preparation for GenAI applications.

For the enterprise, perhaps the most common reality is a table that has been translated in the wrong format. Brian continues, "We're simplifying the world, and in our world that means we are spending lots of energy on tables and forms."

Table Detection with Vision Models

In practice, Unstructured developed custom technology, leveraging vision models, to better interpret the role of tables in documents. As Brian explains, "Only the data owners truly understand the extent of information loss when restructuring data for an LLM. The goal is to minimize this loss when transitioning data from raw to 'RAG-ready.' You want to preserve as much valuable signal as possible, while also filtering out unnecessary noise."

Unstructured developed Chipper, a vision transformer model that performs both object detection and optical character recognition (OCR) in a single step, as well as Hi-Res, which Brian describes as a "more traditional object detection model for tables, forms, and other types of data." Measuring performance optimization within unstructured data pipelines involves a wide range of metrics:

"Plain concatenated text, percentage of words missed, word order, accuracy on columns, accuracy on rows, content accuracy, as well as self-predicted accuracy, all those sorts of things."

Structured vs. Unstructured ETL

Is an entirely new extraction, transformation, and loading (ETL) pipeline needed for enterprises that have existing tools like Fivetran or dbt? When comparing these to structured data pipeline tools, Brian calls out two key distinctions: "They're terrific at moving data" and "terrific at transforming structured data." However, he asserts, "They will do nothing to help get it into a RAG-ready state." Again, Brian calls out a mismatch in previous "modern data stack" and current GenAI use cases:

The language of the modern data stack is SQL, right? And it's designed to feed data warehouses to feed BI applications. Our world is primarily feeding vector stores and the 'BI applications' are really like chatbot UXs powered by foundation models.

Brian sees these use cases as existing in "parallel universes." These problem spaces are both so massive that Brian believes each has a right to exist and drive meaningful enterprise value. He further elaborates:

There's enough competition to figure out how to clean and normalize structured data. If you're talking image and natural language data... from doing scheduling and how you're architecting the connectors... you need such a complex suite of tooling to do that.

And providing that complex suite of tooling is exactly where Unstructured sees its place in any enterprise RAG application.

Defining RAG-Ready Data (and Metadata)

At the core of Unstructured's value proposition is the notion of helping businesses refine their unique context from crude representations to "RAG-ready data." Brian qualifies this end state as "Chunked, Vectorized, Summarized JSON." But context contains not only the source data itself but also key characteristics that surround a file. According to Brian, this hidden context, known as metadata, exists in three buckets:

1. **File-Level:**
   "For example, role-based access controls, versioning, and file path."

2. **Pipeline Generated:**
   "Document elements, like is this a title, subtitle, header, footer, hierarchy within a document, XY coordinates, page number, language detection."

3. **Classifiers**:
   "Organizations may want to nest these in this pipeline to generate net new metadata or to populate a knowledge graph with tuples to support their use cases."

This processed data and metadata work in tandem with retrieval systems to reduce hallucination, improve filtering and reranking, and provide more contextually relevant responses.

Moving at the Speed of Data

Another confounding factor for unstructured data quality is that context needs to be regularly refreshed; the value of static embeddings rapidly decays in a dynamic world.

How can Unstructured help enterprises keep their context current? Brian describes the world they're building for as "one where you're continuously hydrating long-term memory" while "continuously feeding your architecture human-generated data." That way, RAG systems "counter hallucinations that are out of data" with "new context from your organization."

Build Next-Gen Data Pipelines with Microsoft for Startups →

Lesson 3: AI Techniques Are Adaptable

Up until this point, we've largely focused on data ingestion, quality, and retrieval strategies for text and image-based context. What happens when you introduce multimodal data such as audio and video into RAG systems?

The Evolution of Content Management and Retrieval

Data types might be heterogenous, but context is nearly always dynamic. Cody Coleman, CEO of Coactive AI, explains, "When you think about an enterprise, they have very specific, things that they care about. Whether it be for their brand, their characters, or their specific IP, a model might not have a notion about it. "Cody took us through an example of finding all of a brand's logos in a video. "It is a faster, more scalable, cheaper to be able to handle that last mile of getting to the custom taxonomy that matters for that business. As well as, you know, deal with the dynamic nature of the world that we live in - where there's new things and new trends coming up all the time."

At Coactive AI, they conceptualize this as dynamic tags found in the underlying data. "Dynamic tags allow us to do scalable multimodal prompts. We can take in content, vectorize text prompts, vectorize image prompts, and we can take these classifiers and vectorize that as well." Experts might call this "efficient active learning."

"You can label an entire catalog with a dynamic tag in a matter of seconds or minutes from scratch by providing as few as five examples or a single word."

For multimodal AI, this seems extremely important. "Visual concepts can't be described easily in words," explains Cody. "You need to do a step to define what it's like visually. You know, the Barbie Movie?" Cody showed us an image of the look and feel of "Barbie-core":

"Barbie-Core"

"Like you can't describe it in words or anything like that. You have to describe it through examples and through this kind of process."

RAG vs. Retrieval Augmented Classification (RAC)

Although we tend to think of these models for their generative capabilities, using the techniques Cody mentioned creates an entirely new concept, which we could call "discriminative AI. (diagram) "Cody called this process "Retrieval Augmented Classification" (RAC)." Fix: Although we tend to think of these models for their generative capabilities, using the techniques Cody mentioned creates an entirely new concept, which we could call "discriminative AI." (diagram) Cody called this process "Retrieval Augmented Classification" (RAC).

- **Generative AI** (GenAI) creates or generates new content (text, images, and so on).
- **Discriminative AI** classifies or makes decisions about existing data.

RAC could be used to rapidly create or refine classifiers or decision-making models for specific concepts. Just as RAG improved GenAI by grounding it in retrieved information, RAC could potentially improve discriminative AI by providing it with more relevant context for its decisions. Cody explains, "What we've seen from augmenting GenAI with retrieval could also be applied to discriminative AI tasks, potentially leading to more accurate and efficient classification and decision-making processes, especially for complex or specialized domains."

Founder Spotlight: Coactive AI Takes RAG Multimodal

Cody Coleman, CEO of **Coactive AI**, is an expert at distributed retrieval systems.

Cody Coleman, CEO of Coactive AI

Coactive AI helps businesses process and derive insights from vast amounts of unstructured image and video data. Their innovative platform streamlines the traditionally manual process of tagging and searching visual content, using advanced AI techniques to make this data searchable without the need for metadata or annotations.

According to Cody, most organizations use a simple "tag-load-search" algorithm to retrieve and enrich content:

1. **Tag:** Content is manually or automatically tagged with metadata like keywords and categories.

2. **Load:** Context is loaded into a database and associated with the metadata.

3. **Search:** The system retrieves results by matching the search query against the stored tags.

Cody suggests we might benefit by shifting from "tag-load-search" to "load-search-tag," especially with enterprise-scale multimodal AI. "With multimodal AI," says Cody, "which is what we do at Coactive, we can flip the process on its head with a load-search-tag approach... Here we can load and index the raw images and videos...and then make them searchable by understanding the pixels or in the audio directly." In other words, to scale massively, we can use AI to tag our content much more efficiently.

Lesson 4: Using Humans to Build Minimum Viable Preference

We've reached the point of the white paper where it's time for our readers to put their GenAI applications in front of real users. While we've covered best practices for mastering internal context (building high-quality, multimodal data ingestion and retrieval pipelines), we've yet to discuss the external goals and preferences of our users. Where context is concerned, "it takes two to tango." Similar to the issue we raised in Lesson 1, if we can only optimize what's in production, how do we establish a ground truth? What even is a good answer, anyway?

Founder Spotlight: Getting to Ground Truth with Labelbox

Manu Sharma, CEO of Labelbox

Manu Sharma, CEO of **Labelbox**, has seen how critical golden datasets are to building next-gen applications. Manu shared that even the most advanced enterprises struggle with getting their intelligent applications to answer questions in a manner that's consistent with their users' expectations. He states, "At a sufficiently large scale, models are not really behaving as how they would like to. It's usually evident in user engagement or response quality and feedback."

Labelbox is a data-centric AI platform that specializes in helping enterprises efficiently manage, label, and optimize large datasets for machine learning (ML). The platform is built to streamline the process of creating high-quality training data through its robust annotation tools, allowing teams to annotate, organize, and iterate on datasets across various media types, including images, text, and video, and more.

What Is a Golden Dataset?

In the context of RAG applications, a golden dataset is a carefully curated collection of data, typically taking the format of:

- **Input:** What was the question?
- **Output:** What was the answer?
- **Score:** How well did the output answer the input's prompt according to a judge?

The input and output data can be any combination of modalities, but the ultimate goal is to serve as a benchmark for training, fine-tuning, and evaluating RAG applications.

The use of a golden dataset is crucial because it ensures that internal data mastery translates into a seamless and relevant user experience. By leveraging such a dataset, developers can fine-tune their RAG applications to align with real-world use cases, ensuring that the system delivers outputs that meet user expectations and external objectives.

The Role of Human-in-the-Loop Data in RAG Systems

Manu believes that because these applications are doing things that have never been done before, it's hard to simulate human feedback. When it comes to frontier behaviors, there's no substitute for human experts' preference data:

> *The primary way to mitigate and to improve the performance of the system is to figure out...where the system failed to produce the right answer...then figure out a human-in-the-loop process to produce a reference experience or example.*

Labelbox's network of expert labelers and its annotation platform solve the cold-start problem of seeding ground truth with a mix of human intelligence augmented with software. Before putting your application in front of paying customers and risking losing them to suboptimal experiences, you can pay to have Labelbox's experts generate the first batch of production-grade feedback in a shielded environment. Battle-testing with paid experts can offer a lower-risk, high-reward hedge toward establishing a foundation of ground truth to improve upon.

Building in the Unknown

Intelligent applications are nondeterministic. Their outputs or behaviors can vary even with the same inputs. For Labelbox, that sometimes means helping enterprises understand really complex activities. "To give you a sense," Manu says, "some of the best models that are pushing the boundaries of coding capabilities, you really need very advanced software engineers to work for many hours to produce the right example of data or to do an eval on which quote is particularly good and why that might be the case."

Human-in-the-loop data is integral in all kinds of AI operation activities. The most common human-in-the-loop processes include:

- Classification.
- Reward data.
- Production data.

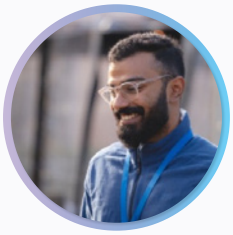
According to Manu, "Evals are very important. And even today, it continues to be the case that human reviewed ground truth is the gold standard." Some of the metrics for assessing RAG quality include "plain concatenated text, percentage of words missed, word order, accuracy on columns, accuracy on rows, content accuracy, as well as self-predicted accuracy - all those sorts of things."

Lesson 5: Evolving through Production-driven Development

Production is a continuously evolving process. Even if an application has shipped to paying customers, the initial version will look and behave very differently than later iterations, whether it's the second, fifth, or hundredth version. Just like our interactions with AI systems, there's no "single shot" when it comes to production: each deployment is an opportunity to learn, iterate, and improve.

Feedback from production users is critical in shaping both the performance and user experience of an intelligent system. This feedback helps identify areas where the interface could be more intuitive, or where the system's responses need refinement to better meet user needs.

Founder Spotlight: Incorporating Real-Time Feedback with RAGAS



Jithin James,
co-founder of RAGAS

As we've outlined extensively up to this point, RAG-powered applications have many levers to pull to improve performance. As such, collecting and incorporating feedback into these intelligent systems requires specialized evaluation frameworks. **Jithin James**, co-founder of **RAGAS**, has open-sourced his beliefs on what such a specialized system would entail.

When building and evaluating an LLM application, there are several key components to consider. You'll have assets, such as the datasets and models, the application itself that performs the desired tasks, metrics or evaluation criteria to measure performance, and tools for logging and visualization.

From our perspective, much of the focus remains on two critical aspects: creating a good test set and determining how to effectively measure performance. Both of these areas are nuanced and require deeper exploration. For instance, there are many layers of complexity even within these two areas alone.

RAGAS is an open-source platform aimed at automating the evaluation of RAG systems. It fills a critical gap by offering metrics to evaluate RAG pipelines without relying on human annotations, making it an important tool for developers working on LLM-based applications. It's a comprehensive framework for assessing these systems across several dimensions, such as faithfulness, precision, and relevance of the retrieved data.

Customizable Failure Responses

"We have handling failures at different levels. At the API level, you can build logic into exactly what you want to do, looking at the output of that, or even at the whole application."

"We call it **Production-driven Development**," says Jithin, "but the whole idea is that, so even when you're building ML applications, you have a test set, a set of metrics, and then what you do is you try to get an objective way to measure what is happening."

Key Components of RAG Evaluation

- Test set creation
- Metrics for assessing RAG performance
- Automated evaluation processes

In the RAGAS implementation, *"There are two parts of it.* ***1.*** *The innovations on the model-assisted eval.* ***2.*** *The innovations on tracing UI and how to visualize it."* It's an iterative process, says Jithin.

> *"You try to figure out why that's happening is the workflow we are advocating for. The community is slowly figuring this out... It won't be just a lot of code, paradigms, (all this stuff will exist), but also census model assisted evaluation and a testing platform."*

In practice, production-driven development yields a faster time-to-production when working with GenAI models. It improves reliability and performance, and results in better alignment with real-world use cases.

Lesson 6: Performance Depends on the Whole System

While many enterprises are successfully integrating GenAI tools into their workflows, from our research, very few enterprises have left the stage of building out GenAI Applications. There's a big gap between a proof of concept and a production-ready application. *"As your business evolves to Day 2: You might be launching in a new country, the way you use LLM flows will inevitably be slightly different,"* says **Jeu George,** CEO of Orkes.

Founder Spotlight: Orchestrating Intelligent Systems with Orkes

Jeu George, CEO of Orkes

Orkes is a platform specializing in **workflow orchestration,** designed to help developers build and scale distributed applications with enhanced observability, security, and durability. Orkes builds upon the open-source Conductor project, which the team initially developed at Netflix. This orchestration tool, now widely adopted across industries, allows businesses to streamline complex workflows, including microservices and event-driven architectures.

Jeu explains that the challenge involves, "building out that LLM model and then going to existing applications where some of the services may be written in Java, some may be in Python, some may be in Golang, and now you are trying to use this LLM model in the right place in the right app." To us, it sounds a lot like **GenAI needs DevOps.**

Model Selection and Routing

Every major AI company we interviewed for this white paper admitted to needing to use multiple models to either balance costs, improve functionality with domain-specific models, or improve system performance in the face of the long latency times experienced with larger models. Although bleeding-edge startups are usually routing between different GenAI models, that might not be the case for enterprises.

As Jeu explains, you have to choose the right model for specific tasks, and dynamic routing is based on performance. "So, when it comes to model routing there isn't one way of figuring out what is the best model to answer it... The best would be based on the performance or the cost or the latency, whatever might be the criteria there."

For many enterprises, this is going to involve hybrid approaches: combining LLMs with traditional ML models. Traditional ML classifiers—decision trees, support vector machines (SVMs), k-nearest neighbors (KNN), and so on—still represent the vast majority of deployed AI. They're often used in tasks where structured data is involved, such as predicting customer churn or classifying images in limited, domain-specific contexts. These models are typically simpler, faster to train, and well-suited to tasks involving smaller, labeled datasets. They also tend to be more interpretable, making them the go-to solution for structured datasets and scenarios where understanding model decisions is critical.

Jeu explains this as "integrating deterministic and probabilistic models":

> *"You could have trained a simple, classifier model on your data, so you don't really need an LLM to do that. Depending upon the question, instead of asking an LLM, I could just route it to my own custom model, which could be very tiny, which would be very inexpensive to run as well."*

Lesson 7:
RAG Isn't a Single Technology

At this point, you likely realize that RAG in practice isn't really one specific technology. In "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," when Lewis et al. formalized and popularized RAG at Facebook Research AI, they were actually introducing a very specific method of building "RAG models." The output of the suggested system is a single "contextualized" generative model. The components work together by using ML across the entire system, not just the language model.

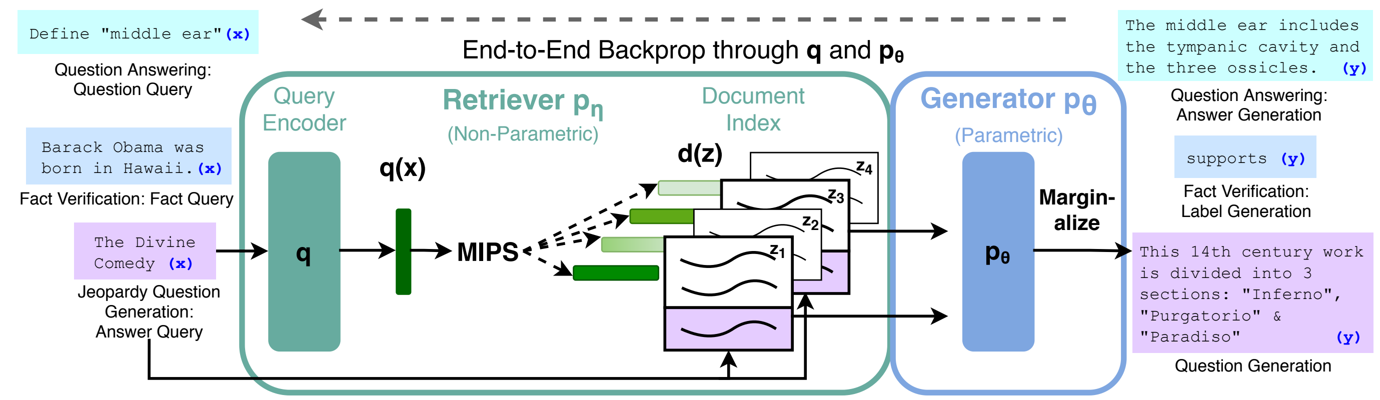

Define "middle ear" (x)
Question Answering: Question Query
Barack Obama was born in Hawaii. (x)
Fact Verification: Fact Query
The Divine Comedy (x)
Jeopardy Question Generation: Answer Query
End-to-End Backprop through q and pθ
Query Encoder
q
q(x)
Retriever pη
(Non-Parametric)
MIPS
Document Index
d(z)
z4
z3
z2
z1
Generator pθ
(Parametric)
pθ
Margin-alize
The middle ear includes the tympanic cavity and the three ossicles. (y)
Question Answering: Answer Generation
supports (y)
Fact Verification: Label Generation
This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso" (y)
Question Generation

Figure 1: Overview of our approach. We combine a pre-trained retriever (Query Encoder + Document Index) with a pre-trained seq2seq model (Generator) and fine-tune end-to-end. For query **x**, we use Maximum Inner Product Search (MIPS) to find the top-K documents **zi**. For final prediction y, we treat **z** as a latent variable and marginalize over seq2seq predictions given different documents.

Large-Scale Systems

"It's not just about language models or Vector Databases," says Douwe Kiela, CEO of Contextual AI. "It's about the entire system... The model is maybe 10-20% of the entire system. And in the end, it's about how all these parts work together." At the point a person realizes that their system might actually be Frankenstein's Monster, they're likely surprised it works at all. It goes to show how powerful this technology is, that it continues to be useful even if you use stale intelligence.

The reality is that these derivations from the optimal system matter mostly when the ingestion, retrieval, or extraction pipelines are complex or at a very large scale. As Douwe says, "Extraction, especially at scale, is much more difficult than most people anticipate."

Harkening back to the original RAG paper, Contextual AI can take any pretrained model and then use end-to-end ML to build what they've termed **RAG 2.0**. By focusing on end-to-end training, they improve performance dramatically for complex systems. In simple terms, they combine all the components of RAG, retrieval, augmentation, and generation by building a model that optimizes across all of the components. Douwe explains, "Our approach is to let them grow up together, learning to work in unison from the beginning. This integrated growth strategy leads to specialization and maximized performance."

RAG and Fine-Tuning

For ML nerds, Contextual AI's approach is brilliant. By using end-to-end ML, you can back-prop loss through the entire system, not just the language model! Or as Douwe says,

> *"RAG allows us to generalize to new data as it is received, while fine-tuning is used to maximize the performance of the system."*

It's a false dichotomy that you have to make a choice between fine-tuning or RAG. In the RAG 2.0 system, they use an end-to-end pretrained contextual model, which "allows for a fast feedback loop with Knowledge Transfer Optimization (KTO), enabling direct incorporation of feedback." In other words, they can rapidly incorporate new information or corrections into their contextual LMs, without the lengthy retraining processes.

There are many podcasts and experts that will tell you to avoid fine-tuning, but what this advice really represents is that you need to understand where the effort should be spent for your organization. Is it in maximizing the ability of the model to work with a large corpus of data, or is it in integrating business rules and governance?

Founder Spotlight:
Building RAG 2.0 with Contextual AI

Douwe Kiela,
CEO of Contextual AI

Douwe Kiela, CEO of **Contextual AI**, is one of the authors of this seminal paper. Contextual AI creates enterprise-grade LLMs. These models are designed to be highly customizable, secure, and efficient for real-world business applications.

From Douwe's point of view, "A typical RAG system today uses a frozen off-the-shelf model for embeddings, a vector database for retrieval, and a black-box language model for generation, stitched together through prompting or an orchestration framework. This leads to a 'Frankenstein's monster' of GenAI: the individual components technically work, but the whole is far from optimal."

This makes sense with what we learned about embedding models earlier. The embeddings being used in many RAG systems are completely separate from the model they're being used with. Ingestion of the candidate documents, the embedding model, and retrieving are all components that affect each other. As Douwe explains, in the idealized system, "RAG allows us to generalize to new data as it is received, while fine-tuning is used to maximize the performance of the system."

Lesson 8:
Enforce Governance by Building for the Right Trade-Offs

Douwe Kiela explains Contextual AI's approach as: "We bring the model to the data to ensure privacy. Auditability is also a crucial feature. With our model, because it's more integrated, you can trace back exactly where the data came from." Depending on the nature of how data flows through your enterprise and incorporates into your system, this might be the most salient way to build the system.

It helps to step back and think about how enterprise users and applications are using search technology now.

Vector vs. Keyword Search

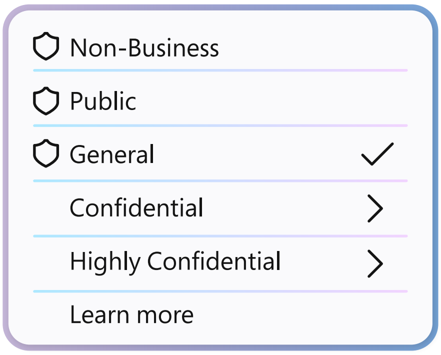
Vector search techniques are often called "semantic search," as opposed to traditional keyword search.

Everyone has used keyword search. You typically try to find something by remembering a unique word in the document or a term that appears frequently. If that exact word doesn't exist in any document, you get no results. Undoubtedly, every enterprise has a system that uses **keyword** search somewhere.

Vector search, on the other hand, always returns results, even if the query and documents aren't closely related. For example, a vector search for "employee onboarding" might return results about "new hire orientation," even if those exact words aren't used. However, depending on how the embeddings were created, it might also return results about "layoffs" if, say, fundraising isn't going well…

Information Access Panic

Immediately after introducing semantic search, the most common "security training" is teaching all the staff to set the confidentiality levels of their documents.



Hallucinations

RAG can easily hallucinate, which is why Vectara even maintains the **Hughes Hallucination Evaluation Model (HHEM) leaderboard** to evaluate how often an LLM introduces hallucinations when summarizing a document.

In practice, RAG techniques are more complex than in these simple examples. For example, Vectara combines vector and keyword search in what they call "Hybrid Retrieval." Amr Awadalla, CEO of Vectara, explains, "it's not just vector search, but vector search augmented with keyword search (lexical search). And Vectara uses specialized techniques to optimize how fast the documents are actually retrieved." Think of the Dewey Decimal system instead of going through documents one by one.

Founder Spotlight: Balancing Great Powers and Greater Responsibilities with Vectara

Amr Awadallah,
CEO of Vectara

Enterprises adopting GenAI tools quickly learn that there's stale, potentially dangerous information with inappropriate access levels, and that many systems have far more visibility than was previously believed. You need "very strict role-based access control," says **Amr Awadallah**, CEO of **Vectara** (he co-founded Cloudera in 2008).

Vectara's mission is to democratize access to powerful, trustworthy AI solutions, especially in search and information retrieval, helping organizations build GenAI applications while avoiding common pitfalls like bias and copyright issues.

For enterprises to be successful at deploying AI tools, understanding retrieval might be the best way to break apart this problem. *"The nice thing about RAG," says Amr, "is because we're not putting the needles inside fine-tuning of the model itself, it is impractical to limit who can see it and who cannot see it."*

Conclusion

Through our conversations with startup leaders, a clear pattern emerged: the evolution of RAG and intelligent systems is moving toward increasingly sophisticated and automated interactions. While current implementations focus primarily on retrieval and generation, the future points toward systems that can not only understand and respond but also take concrete actions based on that understanding. This progression reflects a natural maturation of the technology, as explained by Amr Awadallah's three-phase model.

Getting to the Action Engine

Amr taught us that, "at the beginning of any new technical building block, you always will get people building components as opposed to building the block itself, the solution itself... The IKEA developer market is going to be way bigger than the Home Depot developer market." The technical names are prescriptive vs. descriptive development. "In prescriptive we tell you what to do (ala the recipe from IKEA to assemble a table), while for descriptive we tell you what you can do with the individual pieces (Home Depot raw ingredients) and you have to figure it out how to make the table."

And that's what we learned in the development of this white paper. Some enterprise use cases are going to involve incredibly specific descriptive development for complex workflows, and some are going to involve adopting more generalized, prescriptive components.

Amr breaks the adoption of this technology into three phases:

1. **Search Engines – The Past**
   "You search, you get back a list of documents, but you must go through them one-by-one until you find the answer. That's the old world."
2. **Answer Engines – The Present**
   "You search, and then you get back an answer to a question. You don't have to click on any documents. The answer is going to be right there."
3. **Action Engines – The Future**
   "Action engines take the information provided in an answer and use it to complete an action on behalf of the user. And that's where you hear all these things about agents and AI agents, and that's what the action is about. It's now taking the action on my behalf."

Recap

The landscape of generative AI (GenAI), particularly in Retrieval-Augmented Generation (RAG), is rapidly evolving, paving the way for innovative solutions across various sectors. As organizations transition from basic AI implementations to sophisticated GenAI systems, the key to success lies in a comprehensive understanding of user needs, application context, and data characteristics.

To maximize the potential of these technologies, startups and enterprises should prioritize:

1. **Integration and Flexibility:**
   Move toward more integrated, end-to-end systems like RAG 2.0, which promise greater flexibility and performance.
2. **Planning for Model Orchestration:**
   Tools and platforms that can manage the complexities of model selection, data privacy, and workflow management are essential for enterprise AI deployment.
3. **Multimodal Capabilities:**
   AI systems must seamlessly handle various data types and modalities, from text and images to audio and structured data.
4. **Human-AI Collaboration:**
   Effective AI systems will need to balance automation with human expertise, especially in high-stakes or regulated industries.
5. **Continuous Improvement:**
   Regularly assess and enhance AI capabilities through real-world testing to maintain relevance and effectiveness.
6. **Staying Informed:**
   Stay updated on advancements in AI models and implementation techniques to leverage emerging technologies for a competitive edge.

As we look to the future, it's clear that the success of AI in enterprise settings will depend not just on the power of individual models but on the sophistication of the systems that manage, orchestrate, and optimize these models. The convergence of advanced RAG techniques, intelligent orchestration, and human-centric design promises to unlock new possibilities in AI applications, driving innovation and efficiency across industries.

The road ahead will require continued collaboration between AI researchers, software engineers, and domain experts to build systems that are not only powerful and flexible but also trustworthy and aligned with human values. As these technologies mature, they have the potential to transform how businesses operate, how knowledge is accessed and applied, and how humans and AI collaborate to solve complex problems.

In this evolving landscape, staying informed about the latest developments in RAG, AI orchestration, and related technologies will be crucial for organizations looking to harness the full potential of AI while navigating the challenges of deployment, scalability, and ethical considerations.

Stay ahead of AI advancements – join Microsoft for Startups and gain insights for the future.

Microsoft for Startups empowers early-stage founders with the tools, resources, and mentorship needed for growth and innovation. With access to advanced technologies, expert one-on-one guidance, and a global network, we help startups stay competitive in a rapidly evolving landscape.

Start building your AI with Microsoft for Startups →

About Microsoft for Startups

Microsoft for Startups is a program designed to help founders accelerate their startup journey. The program provides startups with a wealth of resources: access to free Azure credits, pre-built templates to help startups build AI applications quickly and efficiently and go-to-market expertise to sell faster.

Start building your AI with Microsoft for Startups

→

Microsoft | Microsoft for Startups