Understanding AI in HR

A Deep Dive

the joshbersin company

Overview

Artificial intelligence (AI) has become the buzzword of the decade. After 45 years of research, computer scientists have developed systems that can talk to us; classify photos; create images; and analyze, modify, and author video and audio content. And, as with any new technology, there's a massive amount of hype, confusion, trepidation, and even fear.

We have been studying the role of these technologies in HR and L&D, and the impact will be massive. In this paper we explain these technologies and give you a sense of the vast vendor landscape. And even as this paper is being written, new solutions are being developed.

What you'll find is that AI represents a whole new computing platform, and every enterprise software company will have to learn how to leverage it. As we talk with vendors, we find three categories of solutions (see Figure 1).

Emerging AI

Emerging AI is "added on." These solutions are from vendors primarily using analytics and data management to deliver reports, dashboards, and some predictive models. While these solutions are not powered by large language models (LLMs), they use many of the principles and statistics of pure AI vendors for analysis and reporting. This may be a payroll system like ADP that provides HR managers with recommendations for pay adjustments or a report generated by a human capital management (HCM) platform that points out diversity or gender pay problems. These systems do not necessarily use neural networks or generative AI, but they do use advanced analytics to give you insights about your company.

In many cases, these vendors will also use generative AI (e.g., ChatGPT) to give users automatic authoring capabilities, editing capabilities, or custom messages. Applicant tracking systems (ATSs), for example, can now create automatic job descriptions, automatic candidate messages, and even customized interview scripts. Learning management systems are adding tools that automatically create learning content, quizzes, course summaries, and more.

These are all important features, but they are what we call "add ons" in the sense that they don't change the underlying architecture of the platform but instead they add bolt-on AI features.

Figure 1: Three Categories of AI Solutions

Emerging AI Added on	**First Generation** AI Built in	**Second Generation** Built on AI
Predictive analytics Natural language processing Intelligent chat Image generation Generative AI	Machine learning Predictive analytics with external data Advanced candidate matching Content recommendations	Large language models Neural networks Vector databases External data Advanced models
Most HR Applications	Workday, LinkedIn, Cornerstone, SAP	Eightfold AI, Gloat, Beamery, SeekOut

Source: The Josh Bersin Company, 2023

A **language model** is a probability distribution over a sequence of words. To put this another way, when given a sequence of words, it assigns a probability to the sequence. Language models have been in existence for a long time and predate any of the developments with deep learning. They are typically used to complete a sequence, meaning they predict the word that comes next when given a piece of text.

A **large language model** is a language model that uses the latest technologies of deep neural networks with a very large number of parameters trained over a very large corpus of text.

A **neural network** is an algorithm that mimics the human brain by having individual, simple units called "neurons" that take in a bunch of inputs and compute a simple function to create an output. There are different architectures or styles of neural networks that differ in the number of layers, how they are arranged, etc.

First-Generation AI

First-generation AI is "built in." These solutions are from vendors that use AI models and various levels of machine learning (ML), predictive analytics, and candidate or employee matching. They typically have AI engineers, large data sets to analyze, and "single use" ML models that often provide intelligent recommendations to users.

Deep learning neural networks greatly increase the number of layers and interconnections. This was made possible because of the increase in GPU (graphics processing unit) capacity and a drop in price of compute. GPUs allow for multiple parallel computations, which makes deep neural networks possible because they rely on matrix multiplication.

Machine learning refers to the ability of a system to "learn from past performance to improve future performance." A system that predicts retention and gets better over time would qualify as a machine learning system.

For example, many learning experience platforms (LXPs) or HCM systems recommend courses to individuals based on their job role, activities, or specified skills. Talent marketplaces and recruitment tools have been "inferring skills" for years, creating recommended job matches, career paths, or even mentors. They "learn" from user activity; if a user clicks on a recommendation, the system then "upvotes" this recommendation and gets smarter for future users.

These platforms are "first generation" as they do not use neural networks or leverage external labor market data, and they may not use "cost minimization" techniques to continuously learn. The enterprise research planning (ERP) vendors (Oracle, SAP, Workday) generally fall into this category, as do many recruiting tools and ATSs. These systems use natural language processing to find candidates with certain skills or experiences, but they do not have the power of GPT-4 or deep neural networks.

Predictive Analytics vs. Machine Learning

Although they are related concepts, predictive analytics and machine learning are not the same thing.

Predictive analytics is the use of statistical techniques and data mining to analyze historical data and make predictions about future events. This can be done using methods such as regression analysis, decision trees, and neural networks.

Machine learning involves the development of algorithms that can learn from data and make predictions or decisions based on that learning.

Think of predictive analytics as the use of advanced statistics like regression analysis, and machine learning as the use of more complex algorithms like deep learning and neural networks.

Second-Generation AI

Second-generation platforms are "built on AI." These are "next generation" systems that are built for AI from the ground up. Just as vendors like Workday designed new data management and workflow systems for the cloud, these newer vendors (Eightfold AI, SeekOut, Gloat, and others) built their entire platforms on AI-enabled cores. These are data-centric companies defining their products and solutions around the data they manage and analyze.

Second-generation AI systems can perform deep learning, natural language processing, and LLMs in their core platform, extending functionality to build models that accommodate thousands to tens of thousands of data elements. They also let the vendor build better and more important models, which leverage open source LLMs and new algorithms and extend previous models.

Recruiting is one of the most powerful applications of AI today, because it demands a massive amount of data and lets vendors build a variety of models to identify who is a good fit for a role by looking at experiences, skills, job history, and more. Importantly, these systems must be unbiased, mature, and advanced. (Eightfold AI and SeekOut are examples.)

As you will learn in this paper, second-generation platforms are different from traditional software as a service (SaaS) or cloud-based systems. They may use vector databases,[1] or externally provided LLMs and can easily be extended by adding non-HR data.

While second-generation AI vendors look like application software companies, they are data companies first, and application companies second. Their platforms are built to manage, analyze, understand, and act on vast amounts of data. They amass hundreds of millions of profiles from the labor market and other sources, often including pay data, work data (e.g., GitHub), certification data, and more. They then look at the data in your company as a small subset of their vast corpus, enriching profiles of people they may already have in their systems. This means they can see trending skills, career paths, patterns of performance, and even leadership attributes across thousands of companies, making your platform even more intelligent and useful.

Today, thanks to cloud offerings from OpenAI, Microsoft, Google, and many others, every software company can leverage AI. Vendors can start to build second-generation systems by leveraging AI services available from these cloud vendors. We are now already piloting an AI-enabled chatbot to leverage our research, based largely on OpenAI services and our own internally developed research.

Second-generation AI vendors build platforms to manage, analyze, understand, and act on vast amounts of data.

[1] "The vector database is a new kind of database for the AI era," Charles Xie/VentureBeat, December 24, 2022, and "AI startup Pinecone was just valued at $700 million. Here's why VCs like Andreessen Horowitz are obsessed with it and other vector databases," Stephanie Palazzolo/Business Insider, March 28, 2023.

Why AI Is Different from Other Computing Solutions

Over the last 45 years, AI has evolved. In the early days, AI researchers created specialized algorithms for image classification, different algorithms for natural language processing, and other algorithms for vision or video. During one phase of this industry, researchers focused on "expert systems" trying to encode knowledge from experts. While the expert systems did work on a limited basis, they did not scale well outside of their original scope. Despite increases in computing power, these models did not scale up, so researchers moved to a new approach.

Today, thanks to pioneers like Geoffrey Hinton,[2] the neural network has come to dominate this domain. As we describe later, this type of mathematical model simulates the human brain and the way neurons work (although it was developed mathematically, not by studying humans) and uses what is called "cost optimization" to find the range of parameters that make the model accurate ("training the model"). The LLM is an implementation of a neural network that has now advanced with a concept called the "transformer" (see page 16). It turns out these models are accurate, capable of scaling, and adept at handling various types of data.

To give you a sense of how big these systems are, the GPT-4 neural network[3] is estimated to have 170 trillion parameters. Google's new Pathways Language Model (PaLM)[4] has 540 billion. (To shed light on this power, the human brain is estimated to have 86 billion neurons, many of which are not even used for reasoning.)

Reference LLMs are the standard, off-the-shelf LLMs like GPT-4, Lamda, or PaLM. Other standard LLMs are available to license or purchase. Companies can then use these to build their own LLM implementations.

The basic idea of AI is simple; in a traditional software system, a software engineer codes an algorithm (or workflow) and users then *use* the software to automate work. As people use

[2] The respected researcher Geoffrey Hinton is a professor emeritus at the University of Toronto. In 2023, he stepped down from Google as VP and engineering fellow.

[3] "GPT-4," OpenAI, March 14, 2023.

[4] "Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance," Sharan Narang and Aakanksha Chowdhery/Google Research, April 4, 2022.

the system, they produce data. This data, the output or results of some transaction processing (a payroll run, a candidate applying for a job, an employee registering for a course, etc.), is stored. Thus, the system creates data as more and more people use it.

An AI platform, by contrast, *starts with the data*. Think about an AI platform as a massive array of data that fuels the AI "models" (complex mathematical algorithms) to learn, understand, classify, predict, and behave based on the data. Then, various applications use these models (see Figure 2).

This is why AI systems can feel human—they access vast amounts of data and seem to get smarter and smarter. However, remember, like humans, if the data is biased or skewed (e.g., filled with lies, biased populations, etc.), the AI will accurately produce a biased result (possibly determining all high performers are Caucasian males, for example). LLMs can also find inaccurate data, causing the system to "hallucinate" (give a wrong answer). These risks are why AI engineers are so focused on safety (ethics, bias reduction, and explainability).

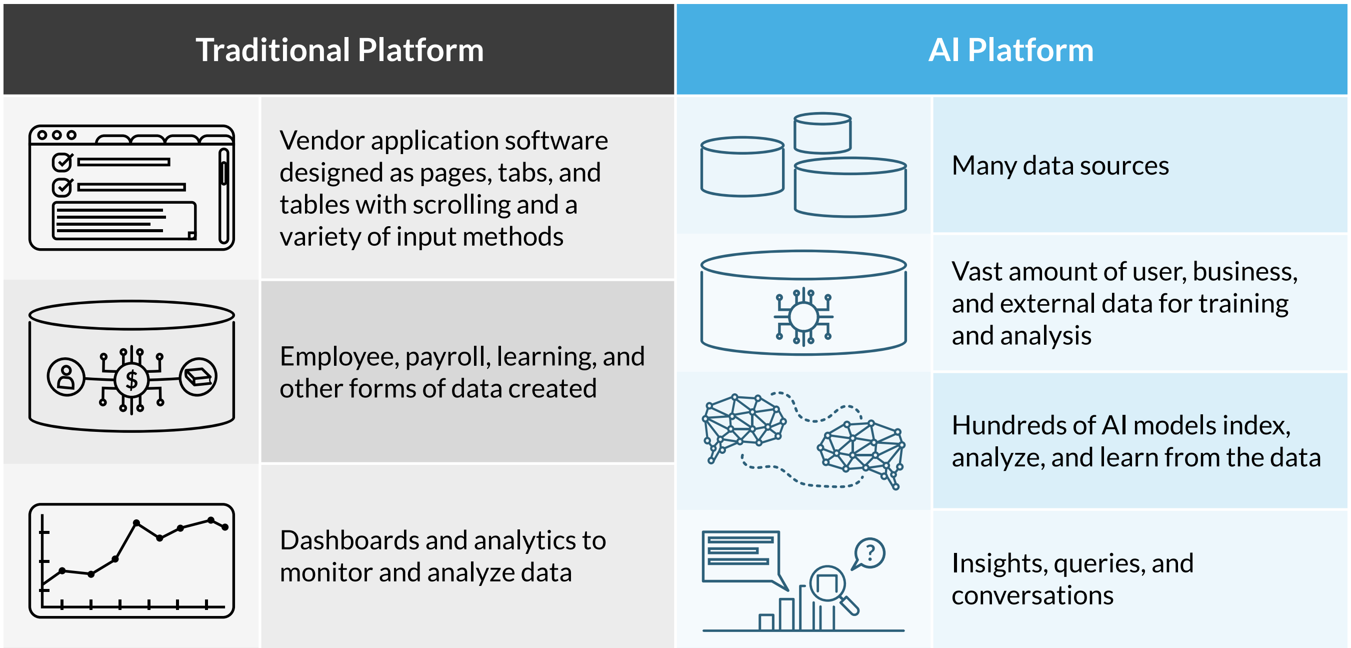
In fact, when a *New York Times* columnist first used Bing Chat, Microsoft's implementation of GPT-4, Bing Chat identified itself as "Sydney" (an internal codename used by OpenAI designed to generate humanlike text) and stated that it "wanted to be alive."[5] Researchers now believe Bing Chat had read and "learned" about its personality to mimic this behavior. Therefore, we must train these systems with valid, reliable, unbiased data.

Today, if you use Bing Chat or Google Bard, the systems show you references from where it found its text. This is a form of explainability.[6] AI explainability should be one of the tools you look for so decision-makers and auditors can ensure fairness, thereby improving utility and reducing risk.

Platform Differences between Traditional vs. AI Systems

Transactional systems (e.g., payroll, learning management, etc.) are designed to capture data quickly, safely, and with integrity. They are built on relational databases (rows and tables), which model typical business transactions. Over the years, these transactional systems have become more

Figure 2: Traditional vs. AI Platforms



Traditional Platform

Vendor application software designed as pages, tabs, and tables with scrolling and a variety of input methods

Employee, payroll, learning, and other forms of data created

Dashboards and analytics to monitor and analyze data

AI Platform

Many data sources

Vast amount of user, business, and external data for training and analysis

Hundreds of AI models index, analyze, and learn from the data

Insights, queries, and conversations

Source: The Josh Bersin Company, 2023

[5] "Bing's AI Chat: 'I want to be alive.'," Kevin Roose/*New York Times*, February 16, 2023.

[6] When an AI system provides misleading or wrong answers to questions, we want it to "explain" why it did so. This function is called "explainability."

and more advanced but never truly designed for the massive queries and analysis needed by AI.

Under the hood of these systems are databases. Workday, for example, developed a proprietary, object-based database (it can model a "worker" as an entire object), which helps the system manage security, workflows, and transactional integrity on the web. Other databases (parallel databases, object-based databases, graph databases, analytic databases, vector databases, and relational databases) can be used for different use cases. The vendor Darwinbox, for example, uses a graph database to model its network-based HCM system.

For most transactional systems, relational databases work well. They are fast, scalable, and easy to query. For highly social-based systems (social networks, second-generation HCM), graph databases are often used. For analytics and reporting applications, vendors often use multidimensional databases (an Excel pivot table, for example).

AI systems (LLMs in particular) are slightly different. They don't just store data as-is; they try to interpret what the various bits may mean. They may take a string of text and break it down into similar "tokens" (groups of letters) that appear to cluster together. And they do the same thing for images, video, and audio. Once they look at all these clusters of data, they try to figure out which clusters of data are similar, which go before or after others, and much more. Without knowing what the data really means, they're "intelligently" interpreting what's going on.

To do this, they model data as a "vector," which is a long string of numbers (e.g., 1-2-8-7). Why does AI use vectors? Mathematicians and computer scientists have learned that complex data (an image, video, or large block of text) can best be analyzed by modeling tokens in this way.

And once tokens are expressed as vectors, the system creates what are called "embeddings"—representing the essence of each token by an array of numbers that represent what other tokens it is near, in what order, and in what pattern. Embeddings are the essence of neural networks.

ChatGPT, for example, takes vast amounts of text and breaks it down into small repeatable patterns, each of which can be treated like a pixel or string of pixels in a photo. These images are classified and grouped statistically just like words, so when a system identifies a photo of a cat, it's using the same technology that finds the financial results for Microsoft in Q2, for example.

The latest wave of language models (such as ChatGPT) take this further by not just understanding text (by encoding it) but also intelligently generating text. This is done by putting a much stronger focus on what constitutes a good answer to each question. If the previous-generation language models were good at understanding what's written and then encoding it, the new generation is good at decoding those questions—providing an appropriate response to its input or queries.

These algorithms have given birth to new hardware and software platforms. While you can store vectors in a relational database, they are not optimized for vector math. A new breed of vector databases has been invented to store and manipulate all this data. And to help with the processing, a new breed of AI-enabled chips have been invented. The Nvidia H100 computing chips, for example, encode vector processing and include many of the "transformer" algorithms in silicon,[7] so second-generation AI systems may run on specialized cloud servers.

Incidentally, most computer scientists now believe it is computing horsepower that has enabled scientists to build these high-performing neural networks. For many years, neural nets were considered a very limited tool. Only in the last few years has high-performance computing proved that these algorithms scale exceedingly well.

One more important point about embeddings—in the language of AI, they are long, complex vectors and may model complex data entities like a job candidate, a position or job description, a career pathway, and much more. Since the algorithm is independent of the data type, information like salary, prior work history, employee sentiment, and much more can be analyzed.

[7] Nvidia H100 Tensor Core GPU, nvidia, 2023.

The More Data the Better

One of the differences between transactional and AI systems is the data they can process to identify patterns and relationships, make predictions, and generate insights. A transactional system can operate with very little data, and over time it creates data somewhat slowly. An AI system, by contrast, must be "trained" with vast amounts of data to increase its accuracy.

OpenAI has proven that LLMs become more accurate as more data is added (it took many years to develop systems that behave this way). If you want to use an AI model to predict a high performer or a good job candidate, you need lots of data. *The transactional data inside your company is not nearly sufficient.*

Suppose, for example, you'd like to put together a "retention model." If you have data about turnover in your company, you may notice that some managers, job roles, or locations have higher turnover. You may also find out that people who are slightly underpaid or overworked are more likely to leave. These findings are important but may not be very insightful.

If you were to use a neural network to study this problem, you may find that people with certain college degrees at certain ages are much more likely to leave than others. You may see patterns around race, family size, or perhaps even working hours. And if you looked at this data across all companies in your industry, you may find that your sales turnover is actually lower than your competitors, but your marketing turnover is higher.

You may also find, for example, that a certain manager uses a particularly harsh or dismissive tone with their direct reports and another manager is more positive and developmental. These insights may help HR neutralize the differences and make performance management more equitable. As you can see, with more data the analysis and predictions become more accurate and more useful.

Since most vendors are not second-generation systems, they are not great at predictions. Most learning platforms, for example, recommend courses and content to users based on their prior activity. Employee experience platforms try to recommend actions based on an employee's location or needs. And many new systems now try to "infer" skills, recommend career paths based on skills, and even identify salaries that appear to be out of range. These prediction modules are small AI applications embedded within the transactional application, so their ultimate power is often limited by the small amount of data they access.

Why label the data set as "small"? Consider all the data in a large HCM system used by a large company like GE. Unless that system has vast amounts of history and can access data from the recruiting system, performance system, and many other systems, it really doesn't know much about the employees. These first-generation systems have not been as exciting or groundbreaking as we had expected. LLMs and second-generation systems are changing this.

Different Databases Defined

Relational databases use tables to store data, where each row represents a single record and each column represents a specific attribute of that record.

Vector databases store data as arrays of numbers, which can represent complex and unstructured data types such as images, audio, and text.

In **graph databases**, data is modeled as nodes (vertices) and edges. Nodes represent entities or objects, while edges represent relationships or connections between nodes.

Relational databases are optimized for efficient transaction processing and structured data. Vector databases are optimized for similarity searches and complex data types represented as vectors.

Graph databases are effective in handling complex and highly interconnected data, such as social networks, recommendation systems, fraud detection, and knowledge graphs.

Why Second-Generation Systems Are Important

Thanks to GPT-4 and other LLMs coming to market, we now know that these "data-centric" systems are far more powerful than people realized. Consider your credit card company's transaction system. It has "learned," through years of fraud analysis, how to detect fraud from hundreds of variables. These may include transaction history, location, time of day, your pattern of purchases, and hundreds of other items.

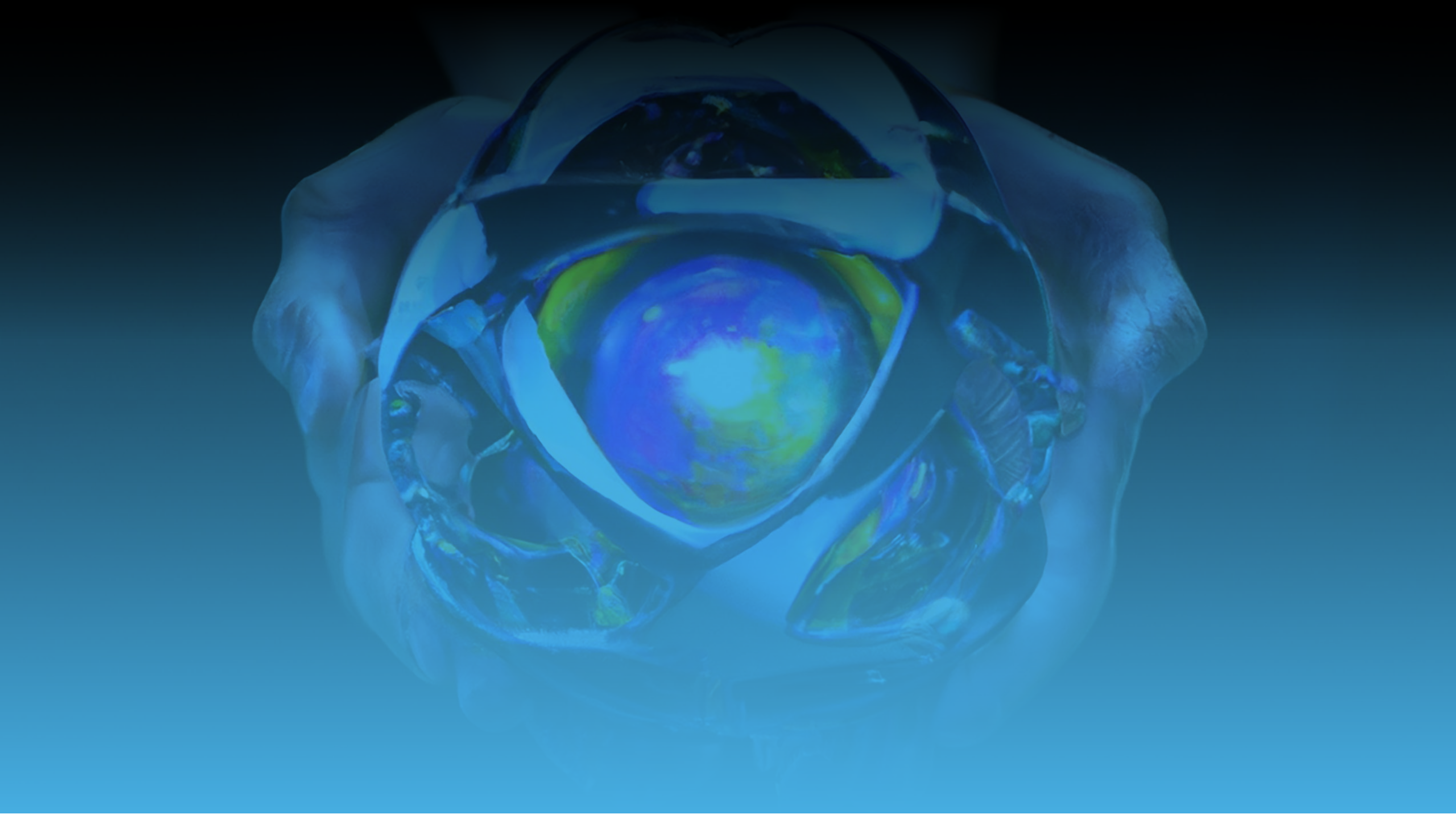
Such data-centered systems tend to get smarter as they access more data. In the world of HR, you need far more than the HCM data within your company. If you want a powerful model to identify a perfect salesperson or high-performing engineer you'd want to look at all the salespeople or engineers in the world! That is what second-generation AI systems do.

These systems amass hundreds of millions of employee profiles, often anonymized, and often scaling into the billions. And again, because of their architecture, they are designed to integrate hundreds of other data sources: your company's performance management data, learning data, and actual work data performed by each employee. In the case of software engineers, they often access GitHub data (which tells us what tools, skills, algorithms, and projects any person has performed), nursing certification data, and just about any other employee-relevant data you can provide.

We use Eightfold AI, for example, to study trending skills, job titles, and even organization designs for hundreds of companies by industry. Their vast data set can show us how the engineering skills of Chevron, for example, compare to those of Exxon. And because of this vast amount of data, your predictions, matches, and models are far more robust and reliable than the data only used for your company.

One other aspect of second-generation systems is their ability to adapt. Vendors that build on an AI core can regularly replace models with new algorithms, adapt their LLMs, and literally "upgrade" their systems continuously. Traditional first-generation HCM platforms (Workday, Oracle, SAP) find this difficult and may not be able to upgrade their systems very quickly.

Second-generation systems can amass hundreds of millions of employee profiles, and often scaling into the billions.



How Do These Models Work?

How does AI actually learn?

Let's suppose you want to "classify" information—identifying which images are dogs, cats, or people. In the case of HR, suppose you want to know who the high performers are in your company. This could be useful to help with performance management, pay equity, promotion, or succession management. And, once you know who these high performers are, you may want to understand what background they have, what skills they possess, and many other important qualities.

How could the AI system even "classify" or identify these people? You could simply label a certain 10% of your employees as high performers and then let the AI figure out what appears to distinguish them. It would then use a variety of algorithms to understand what they have in common (tuning the "parameters" discussed above), and the results might be surprising.

You may guess that college degree, tenure, and performance ratings are very good predictors of a high performer. Well, the AI, which has no real "knowledge," may determine otherwise. Liberty Mutual did a project years ago to identify the best people to work in auto insurance sales. After looking at college degrees, GPA, work experience, extracurricular activities, and many other things, its model eventually found that the most predictive factor in performance was "having worked in auto sales prior to coming to Liberty." Who would have guessed this?

A bank in Canada found a pattern of theft in its branches. It could not understand the pattern, so it created a new compliance program to teach people about ethics and banking rules. The training didn't work. Once the bank threw all the data into a predictive model, it discovered that the factor which most influenced theft was "the number of miles from the branch to the district manager." In other words, the branches with theft were not getting visited by management. Who would have guessed?

The exciting thing about AI is that it doesn't "guess" what's going to work. It simply creates a model. Suppose, for example, we noticed that a lot of employees in Kansas were behind on achieving their goals. An AI system could ripple through the massive database, look for goal attainment, and come up with a "predictive model" to figure out which factors seem to correlate with poor performance. It could then be used to set quotas, coach managers, and many other things.

The breakthrough of AI today is that we can look at billions of "parameters" to figure out what's going on. Using neural networks, the data "trains" the model so it can get smarter and smarter at figuring out why some people are underperforming. And neural networks can analyze hundreds to thousands of dimensions to see precisely why this underperformance happens.

Over the last four decades, engineers have built many types of models. Some are designed for data streams (audio or video), some are designed for images (photos, drawings), and some are designed for text (language, articles, books, websites). Much of this research has focused on the algorithms and architecture of these models, leading to the LLMs we see today.

And what researchers have now learned is that all data (whether images, videos, audios, or blocks of text) can be mathematically modeled as numbers or vectors, enabling the new breed of systems (LLMs) to be used quite broadly. In our domain, for example, LLMs can identify skills among job candidates, cluster and simplify job descriptions, analyze the video streams created in meetings, and now, using generative AI, answer questions, create PowerPoint slides, and even draw pictures.

Understanding LLMs: AI Is Advanced Mathematics

Once we understand that all data can be converted to bits and vectors ("embeddings"), we can dive briefly into the mathematics under the covers. One reason to explain the math is that it demystifies AI, so you don't feel intimidated by its appearance of "intelligence."

Let's go back to the HR example and assume we have a massive database of employees (and job candidates), including their educational history, job histories, profiles, and perhaps personal connections. And let's also assume we "enrich" this data with their salaries, performance ratings, documents, and perhaps software they've written or other data they produced at work.

If we could "model" all this data in one massive system, we could conceptually look for an infinite number of patterns. If we want the "high performer" model, for example, we could look for the relationships between career progression, salary increase, and all the other data. If we wanted to understand a person's *skills*, we could look for key words (e.g., Java) and evaluations of their code or work to infer skills.

This is essentially what neural networks do. Suppose we tell the system who the high performers are (i.e., "labeling the data"). These mathematical systems create vector representations of this data and then distribute the vectors to millions or billions of nodes (called "parameters") to see what's related to what. And then, using calculus, they "tune the parameters" until the model works out a way to predict the labeled high performers with the best possible set of parameters.

A neural network, which classifies "high performers," would need a training database that tells it explicitly who these people are. The system would then use calculus to "train itself" by attaching weights to various parameters (there could be billions of these) until it finds a "best fit." This is called "training the model."

While we as humans may try to do this by hand (we probably predict that education has some impact, which of course introduces human bias), the model has no idea what these numbers represent. Using vector calculus (a form of math many of us learned in college) it simply "figures out" (using second derivatives and other calculus) the best set of weights and biases (the technical term for these "tuning knobs") to figure out what numbers work best. In some ways, if the training data is unbiased (e.g., all the high performers are not white men), the AI system will have far less bias than we do!

It turns out that these neural networks, which are made up of interconnected nodes (neurons) organized into layers (processing them in stages), actually simulate how the human brain operates. What AI scientists have been doing for years is optimizing, improving, and testing these models.

In the language of AI, the model is "learning" through the data how to best create the magic set of equations that "nearly perfectly" predict the performance of people based on the data collected in a company. If we change the data (e.g., we look at a new quarter and have a reorg), we would want to "retrain the model" so it recalculates these weights and predicts what is now going on. And there are many interesting calculus models that make this "training process" speed up. They are typically based on computing how far off the model is, the "gradient" or slope of its error, and then inching the numbers back to make the model more accurate.

As you can see, it's not "evil magic" as it seems, but there are lots of interesting things to consider. Since there are so many possible variables (millions to billions), the real "math" uses "vector calculus" or "vector algebra." We are manipulating hundreds of numbers for each case of an individual's performance (their demographics, location, manager, tenure, etc.) so this long string of variables can be managed as a group. (Again, a "vector" is a string of numbers in a row that can be conceptually plotted in space.)

By using vectors to analyze all this data, we can use calculus and other techniques to visualize how "close" or "far apart" different possible weights and biases are from each other. This helps the machine more quickly "learn" what is the best possible model that predicts reality.

Because of all the math involved, traditional computers may not perform well. If you're familiar with relational databases, you know that it's very difficult to store large "vectors" and manipulate them. Second-generation AI systems (systems architected for AI) use what's called a "vector database" to store this information.

In addition, the amount of computing required is massive. If we look at all the employees in a company, all the job candidates in the pipeline, and the hundreds to thousands of variables to consider, the computational load is enormous. AI-designed systems use chips (computer processors) with special coding to speed up vector operations. This is why Nvidia chips are so expensive and why Google, Microsoft, and Apple are building their own AI hardware.

Let's also call attention to an important point. If you're talking with an AI vendor, one of your first questions should be either "How does your system train itself?" or "What data does it use for training?" These fundamental questions are more important than you may realize. The typical HR system of record, for example, does not have much data to interpret. In many cases, your HRMS stores employee name, age, address, job history, and perhaps data about training (while that data is often elsewhere). If you want to use it for skills assessment, succession management, pay analysis, or other strategic purposes, you're going to need much more data.

This is why the second-generation AI platforms are designed for massive data analysis. These vendors have already amassed hundreds of millions to billions of profiles, and you can then "enrich" the data with your company's information. They are much "smarter" than a system dedicated to your data alone, so ask vendors about this issue.

How Does AI Understand Skills?

Everyone is talking about skills these days, so one may ask: How can AI determine a person's skills? The answer is complicated, and this is also where first-generation and second-generation AI systems differ.

The simplest approach to understanding what skills a person may have is to match words in a candidate's resume to a database of skills. This method was used in early ATSs, which they essentially matched words and phrases to job descriptions. Today, advanced AI looks at words and much more. Modern models may look at where you worked and what jobs you had; who you worked with and the skills those people had; and/or what technologies were used in the companies you worked for.

In advanced AI systems, we can point the database to your certifications (legal, regulatory, or internal), the code you've written (GitHub, for example, which includes evaluations of your code), your specific tests (e.g., nursing certificates), your performance ratings, and virtually any other data source that can be found (e.g., sales attainment, etc.). This means a second-generation AI system is designed to integrate many sources of data, and its models (algorithms) can interpret this data to make the model smarter.

Many people believe they should self-assess their skills or their manager should assess their skills. While this is certainly a good idea, the AI can find skills by looking at "adjacent skills" you've picked up. One AI senior scientist told us that for every three skills you think you have the AI can find five others you also have and ten adjacent skills you can develop easily.

It is important to remember that a "skill" is not like the "competencies" we used in the past. Competencies were literally selected out of a book and manually attached to a job or job description. There was some industrial and organizational (I/O) psychology involved, but mostly we relied on human judgment.

Skills, as we define them today, are very different. We like to think of skills as "metadata about people." Let's say someone worked for a relational database company, for example, so they may have a "skill" in Sybase software. Sybase is not a "competency," it's a technology. Java is not a "skill," it's a programming language. In the context of AI, we can think about a skill as a word or phrase that helps us select, develop, or evaluate a person.

Given that AI understands "skill" as a "textual word," (or "token" as described previously), it doesn't really know what these skills mean. But despite that lack of "intelligence," AI is very good at figuring out what skills a person has. Using natural language processing or a neural network (second-generation AI), we can infer and predict skills quite accurately and refine them in amazing ways. And that means when you write a job description, it can "identify the skills" of people in that job, can "find people with these skills," and can "find jobs similar to this job" with astounding precision. These are groundbreaking new applications for HR.

Some systems go further, breaking the barrier of a "finite skill ontology." While any collection of skills is bound to be finite, generative AI can "invent" a potentially infinite variety of skills. Imagine, for example, a person who interacted with the system for only a few minutes and the system suggests a skill such as "analyzing luxury auto sales in the southwest U.S."—something no finite collection of skills can aspire to do. These are "second-order skills" the system may discover. The idea of manually curating a skills taxonomy is really somewhat absurd, since second-generation systems find new skills all the time.

Second-generation vendors do more. They don't just treat skills as words and phrases; they look at the relationship of these skills with other factors. If a candidate has "Java" on their resume and recently worked in a coffee shop, you may guess that this is a skill that has something to do with brewing coffee. If they live in Indonesia, it probably refers to the country of Java. On the other hand, if the candidate works for a software company, it likely means it is related to the programming language.

The system should also understand and identify "skills adjacencies." It may figure out, for example, that "project management" and "PMO" are related terms, or that the Python programming language is not a snake, or even that people who "know Python" are also good at SQL. To make all this knowledge available to the AI, second-generation vendors build tools that create an expansive and context-aware ontology.

One vendor, for example, used its AI to identify "leadership skills" by training the system to look for soft skills. They analyzed the skills of one company's leaders (director level and above) and then compared them to the skills of all their competitors. The AI essentially diagnosed their culture, telling the CEO and CHRO what kind of values, behaviors, and leadership experiences their company relied on. This gave them ideas about how to develop leaders, hire leaders, and possibly identify blind spots in their leadership teams.

A highly advanced vendor has refined its matching algorithm with even more advanced models. Imagine a company that has 15,000 employees opens a range of job postings. The first model is the "apply" model, which looks at who applied. This set of people (maybe 250,000) are people selected from the 2 billion white collar workers in the world who are interested in this company. The second model is the "interviewed" model, where the system looks at who was accepted for an interview. This refines the number of "best fit" candidates down to a few thousand. The third model is the "accepted" model, which shows who received an offer. The fourth is the "performance" model, where the system studies who advanced, succeeded, or remained.

Another important use case is to help people advance their careers. Based on your skills and experience, the AI can find "the next best job" for you; it could show you a "stretch assignment," and it could even tell you precisely what skills you may lack as you aspire to grow. It could then index all the training content in your company and accurately recommend courses, experiences, or mentors to help you. One vendor we talked to has built deep learning models to study these career pathways and model them against progression and growth. This lets the system present "sound recommendations" on the next jobs or roles for internal employees.

Remember, however, that in all these use cases the quality of the match is the most important thing to evaluate. Lots of first-generation AI systems look like they do this well, but you'll find that they often fall short in real life. For example, one of our clients used its enterprise resource planning (ERP) vendor's "skills matching" technology to recruit diverse candidates. Despite the vendor claims, its candidates were not diverse at all. They later upgraded to a second-generation AI platform. You want to talk with vendors who use massive amounts of data and have deep experience matching candidates to jobs without biased outcomes.

This is why we find that most second-generation AI systems were built by people with PhDs or serious academic backgrounds in AI, mathematics, or computer science. I/O psychologists are not trained in large data systems or AI processing, so look for a vendor with strong AI credentials. Often, they will share their patents, PhD theses, or other experiences they had building systems that are free from bias.

Applying AI to Pay Equity

This leads us to another important application of AI in HR. Not only can these systems give you a much more diverse pool of candidates (if it's trained it will far surpass human sourcing), you will be able to use them to remove bias in advancement, promotion, and pay.

Pay equity has become a massive issue. Google paid over $118 million to employees who proved that the company practiced gender discrimination in hiring and promotion, and Goldman just paid close to $200 million for a similar lawsuit. A second-generation AI system could recommend adjacent job opportunities independent of gender, and eventually it could be used to help with promotions. If you added your company's pay data to this system, you could certainly see the system giving you "predictions" for pay, which would immediately show groups who are underpaid, overpaid, or otherwise. Pay equity and diversity and inclusive promotion are enormous new applications for AI.

Applying AI to Language, Words, Images, Video, and Audio

How does AI manage to read, develop, and analyze nontextual data? Well, without trying to give you the entire history, there has been 40+ years of research into audio, video, image, and text analysis. Over this period of time, computer scientists have developed different algorithms that work for different data streams.

Today, AI scientists are beginning to believe that all these problems are really all the same. In every case, whether you are looking at an image to identify or a string of characters to understand, the problem is one of "defining the data you have" and then using a series of models (neural networks or others) to match, categorize, and deeply understand the data's patterns. In many ways, AI is a highly tuned "pattern matching" system with lots of tuning knobs to adjust its accuracy.

When we look at generative AI, the ability for a system to understand your language ("prompt engineering") and interpret the best response or answer, this process is simply more advanced. ChatGPT and the LLMs use massive neural networks and a mathematical model called a "transformer" to do this with uncanny accuracy. And we can expect these systems to become more refined and domain-specific as they are further trained with our continuous use.

What Is a Transformer (the "T" in ChatGPT)?

Over the years, AI researchers developed different algorithms for different types of data. And while this let scientists optimize tools for image recognition, voice, and text, it was unable to combine them. Researchers were searching for a "universal model" that would possibly combine them.

That's where the transformer model came in. In June of 2017, a set of Google engineers published a famous paper called *Attention is All You Need*[8] (a confusing name). One of the problems with prior AI models, for example, is that every "object" (word, phrase, or bit stream) that we want to predict can only be analyzed by its proximity to others. It's as if you could never see a map of the entire U.S., only of your own neighborhood. The transformer model addressed this by making every object (billions of words or bits) have an "affinity" or "probability" related to every other object. (Mind boggling, isn't it?)

It is also designed in a way that the computer can parallelize its work. The transformer algorithms enable the neural network to compute weights independently while "peeking" at their relationship with each other. The result is that when we add massive amounts of computing horsepower, the transformer gets faster and smarter.

As you can tell, computer scientists and mathematicians have worked hard on this, but the essence of this model is that we can "group" or "find" or "arrange" things when there are billions and billions of elements. It is considered a "generally useful" model, meaning that AI, as a field of study, is converging around it.

The **transformer architecture with attention** is a particular style of deep neural networks that has been shown to solve a variety of problems. Before that, people built special architectures for each type of problem.

LLMs today typically reference a language model built on the transformer architecture with attention with a very large number of parameters trained on a very large corpus of text that can do next word prediction, which means if given a piece of text, it can predict the next word (or "token").

[8] *Attention is All You Need*, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin/Neural Information Processing Systems Conference, 2017.

A Market for Large Language Models

As this paper is being written, many new vendors are building and offering deep learning LLMs in various ways. Some are free (open source), some are proprietary, and some are embedded in products (see Figure 3 on the next page). We also suggest you peruse the "Open LLM Leaderboard,"[9] which rates these LLMs against a set of standard "accuracy tests" in a ranked list. It's astounding how many there are.

In a short period of time, you, as a corporate buyer, will have options to either use vendor-provided models or use your IT department to build your own. Bloomberg, for example, has already built its own LLM by the name of "BloombergGPT." It is specially designed to store, analyze, and serve financial information from Bloomberg's massive historic database of financial instruments and company financial performance. Your company will likely build similar systems around your unique business information.

How Do We Know These Systems Are Fair and Accurate?

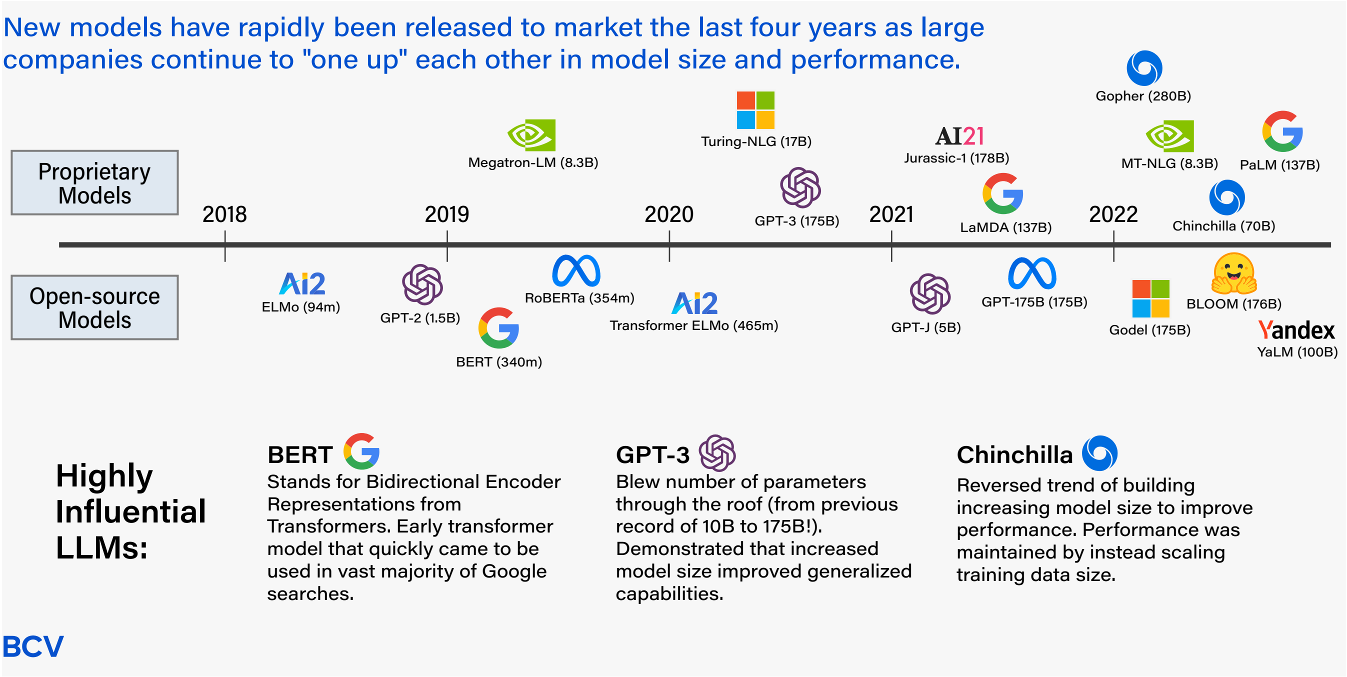
The main purpose of AI is to understand and analyze "second-order" issues. You can build a nine-box grid to define a succession plan based on a few meetings and looking at performance reviews. You can decide who to hire based on a few interviews. You can set pay based on bands and historic raises, but you don't even know what you're missing.

A true AI system would make all these decisions far more accurate. You could deeply assess the skills of an individual and then find "matches" with others to find a good successor. You could look at the skills of the highest performers and see what they "appear to know" that others have not figured out. And you can look at job candidates and internal staff and see their fit and adjacencies and then match in a far more accurate way.

Several years ago, the HR leaders at Google, who have worked tirelessly on interviews and testing for job candidates, told us that despite all their work, their AI was a far better predictor of a "great hire" than all the tests and interviews developed

[9] Open LLM Leaderboard, Hugging Face, 2023.

Figure 3: State of the Deep Learning LLM World Today



New models have rapidly been released to market the last four years as large companies continue to "one up" each other in model size and performance.

Highly Influential LLMs:

BERT
Stands for Bidirectional Encoder Representations from Transformers. Early transformer model that quickly came to be used in vast majority of Google searches.

GPT-3
Blew number of parameters through the roof (from previous record of 10B to 175B!). Demonstrated that increased model size improved generalized capabilities.

Chinchilla
Reversed trend of building increasing model size to improve performance. Performance was maintained by instead scaling training data size.

Source: Bain Capital Ventures, 2022

by people. Why is this true? Because a well-trained AI system simply has access to more data, and it can see second-order effects that we miss.

For example, suppose we "trained" the model on Caucasian college graduates and slipped a few African American nongraduates into the sample. And what if the Caucasian grads had lots of work experience and the African Americans didn't. The model is very likely to conclude that one of the distinguishing factors of high performers is that they're white. In fact, Amazon abandoned their AI recruiter for this exact reason.[10]

Obviously, this would be a disaster and likely lead to a lawsuit. We need to make sure that the AI models themselves and the data used to train them are unbiased. Now that there are laws in New York and other states that mandate diverse hiring (and promotion and pay), we have to ensure that these models are trained carefully. Some vendors (Eightfold AI, SeekOut, iCIMS, and others) have spent years developing models that are proven to be "unbiased" for recruiting, selection, promotion, and other factors. This is still a large work in process.

While these are small examples, you can immediately see the value. Almost every people-related decision could be more accurate with AI. But then the issue comes up: Can we trust this new system to be accurate, unbiased, and fair?

As you can imagine, if you train a system on biased data you get biased results. That's why a small AI model based on your company's data may lead to problems—every bias embedded in your company will be amplified. If you're an investment bank and all the leaders are white protestant males who went to Ivy League colleges, you can imagine what the AI will do. And in areas like pay equity, recruiting, succession, and leadership, this could result in a lawsuit. You, as a buyer and user of AI, have to make sure your vendors are very aware, focused, and invested in removing bias because the algorithms alone may not do this.

[10] "Amazon ditched AI recruiting tool that favored men for technical jobs," Betsy Reed/*The Guardian*, October 10, 2018.

Industry and Domain Specialization Really Matter

Then there's the issue of domain expertise. Suppose you buy a recruiting tool that has been deeply trained to find and select great engineers and software professionals. Would it accurately predict the best people to work in a coffee shop or manufacturing facility? Maybe not, but if the system is *well-trained* with data in those job domains the answer may be "yes." However, if you buy it out of the box and you are the first customer with that use case, the system may not work as well as you'd like.

We believe these systems will quickly become specialized by domain. Talk with your vendors about the industries they serve and where they've been the most successful. Chatbots that work for sales and lead generation, for example, may not understand or be trained for the questions and issues you face in sourcing and recruiting. That's why vendors like Paradox.ai, who have spent years training its systems for recruiting, remain focused on this problem, because its system is getting smarter and smarter about recruiting every day.

How Do You Sort Vendor Offerings?

Now that we've covered the basics, here are some questions you should ask vendors to understand their AI capabilities. For comparative information, see Figure 4.

- What data set or sets are you using to train your models?
- How large is your data set by number of profiles or objects?
- How do you make sure your models are not biased?
- What industry specific data and models have you tested?
- Are there certain employee groups (e.g., white collar, blue collar, junior, senior) who are highly represented or not represented?
- What is your offering around "explainability" of your AI?
- Can we see your research on bias and how you've reduced or eliminated it?
- How does your system use human feedback to improve models over time?
- Can we talk with customers using your AI and see how well it performs?
- If you are a skills vendor: How do you maintain your skills taxonomy?
- If you are a recruiting vendor: How do you validate the safety of your systems' recommendations?
- Can you tell us the background of your AI team and how long they've been working in this area?
- What LLMs do you use and why have you selected these models?
- How is your system compliant with new AI regulations like those in New York?
- Which AI technologies are you leveraging? (If it's predictive analytics but not ML then it's a statistical tool, not an AI solution!)

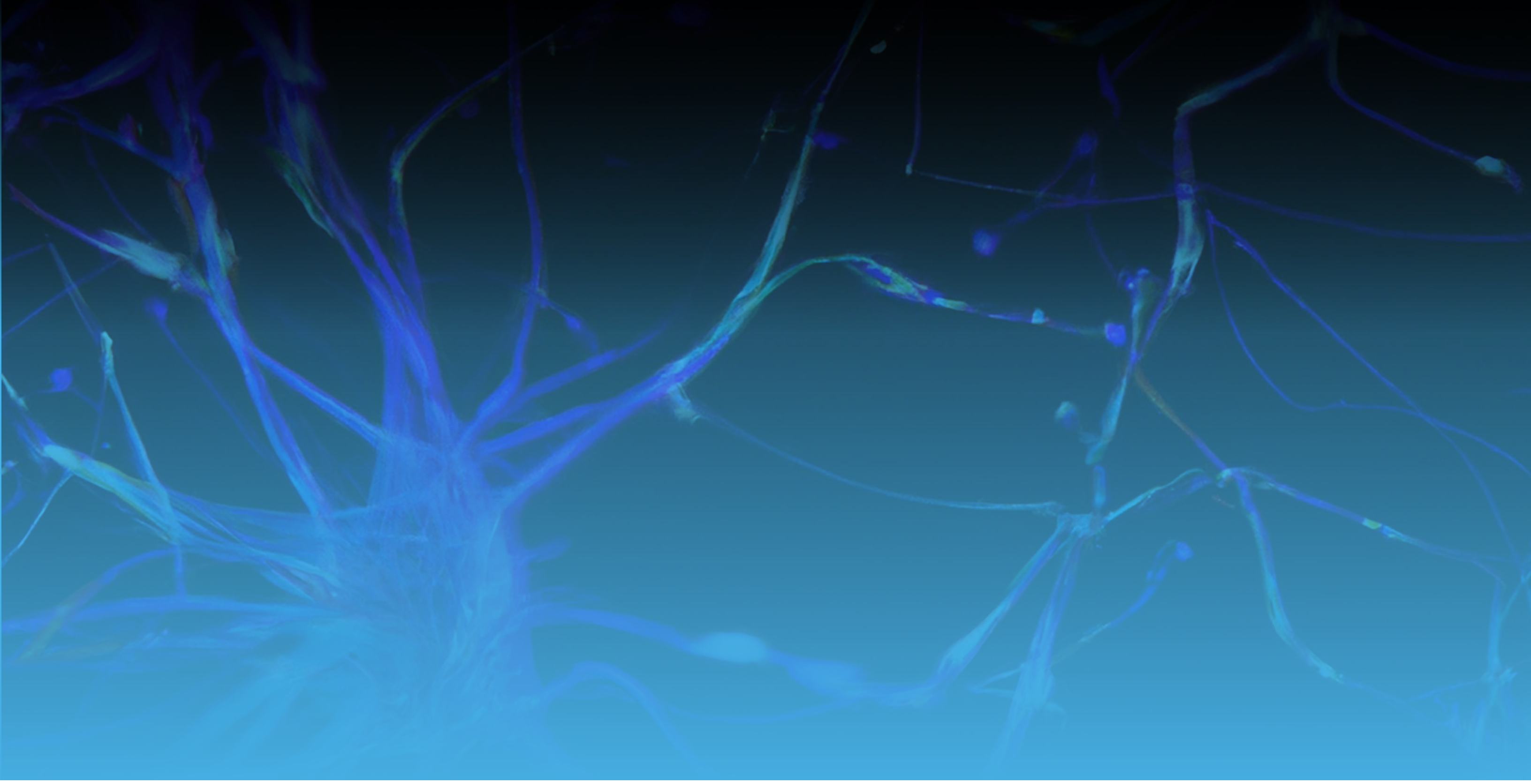
Figure 4: Comparing AI Vendor Offerings

AI Component	Emerging AI Is Added on	First Generation AI Is Built in	Second Generation Platform Is Built on AI
Architecture	Traditional transactional systems, perhaps using generative AI for text, image, and video generation	Transactional system using ML and AI models to understand the transactional data	AI platform leveraging transactional data and external data to develop AI models (or using LLM) to recommend, correlate, and predict across thousands of parameters
Data	Data generated by users as they use the application	Data generated by users with the ability for external data to be added for reporting	Open data platform that lets users add many forms of data (performance data, work documents, code, certifications, etc.), all used by the LLM or other models

Figure 4: Comparing AI Vendor Offerings *(continued)*

AI Component	Emerging AI Is Added on	First Generation AI Is Built in	Second Generation Platform Is Built on AI
Structure of data	Self-described in advance: employee name, job title, experience, salary history, job history, etc.	Self-described in advance: employee name, job title, experience, salary history, job history, etc. May also include skills, interests, and other extended data	Self-declared data + insights from work artifacts (GitHub, Jira, Salesforce) + foundational knowledge about the world embedded in generative AI models' structured data + unstructured data (e.g., code, job descriptions, documents, and/or Salesforce records). This is what people have done, not just what they say
Querying/ matching/ recommendation	Querying/matching of structured data	Querying/matching of structured data	Matching of structured data + matching of semantic meaning (embeddings) of unstructured data. Vast capabilities to summarize selected data and understand trends
UI functionality	Structured data inputs and outputs	Structured data inputs and outputs	Structured query + natural language inputs. Structured data + natural language outputs. Interactive chat interface automates workflows based on structured & unstructured data and world knowledge
Fundamental ways to understand talent	Focused on transactional data submitted by a user	Skills is the fundamental element to understand talent and match talent with opportunities. Skills and skills proficiencies are often self-claimed by Individuals	Skills + skills proficiency understood from the work artifacts. Beyond skills–capabilities that are understood and demonstrated through experiences and accomplishments. Reason, match, and make recommendations based on capabilities of people, organizations, jobs, and gigs
How skills ontology is created	Skills, if used, input by users or managers	Massive effort to build and maintain a master skills ontology that consists of structured data. Massive effort to merge skills ontologies and taxonomies across different systems	No need to build or maintain a master ontology in the dynamically changing world. Skills can be constantly inferred by platform and are expressed with both structured data and in natural language. Relationships between skills are determined at run time when it's used by applications
Models and upgrades	System may take advantage of model upgrades from cloud providers	Typically upgraded slowly by improving machine algorithms over time	Able to quickly adapt to new models, replace models, and take advantage of algorithmic innovations on a continuous basis

Source: The Josh Bersin Company, 2023



We Are Here to Help

Finally, please feel free to call us. We spend time with all the major vendors in the market and often help companies sort out the vendor landscape. Not only are systems different in their architecture and history, but they also focus on different problems. And please note that second-generation AI platforms require many new tools, systems, and skills. A study by BCG found that 78% of LLM projects fail,[11] pointing out the fact that this is really a whole new computing platform, not just a set of new algorithms.

Despite these issues, we firmly believe that this new generation of AI will revolutionize HR and all aspects of management. Today, second-generation AI is being used for many new applications: diverse recruiting, internal mobility, career planning, capability development, and even helping to simplify and improve the company's job architecture. Soon we will see applications in pay equity, promotion, and even performance management.

We look forward to hearing from you as we all journey down the path to ever more intelligent and valuable HR technologies and platforms.

[11] "Foundation Models and Five Predictions for AI in 2023," Rak Garg, Sam Crowder, and Dawit Heck/Bain Capital Ventures, December 15, 2022.

About the Authors

Josh Bersin

Josh founded Bersin & Associates in 2001 to provide research and advisory services focused on corporate learning. He expanded the company's coverage to encompass HR, talent management, talent acquisition, and leadership and became a recognized expert in the talent market. Josh sold the company to Deloitte in 2012 and was a partner in Bersin by Deloitte up until 2018.

In 2019, Josh founded the Josh Bersin Academy, a professional development academy that has become the "home for HR." In 2020, he put together a team of analysts and advisors who are now working with him to support and guide HR organizations from around the world under the umbrella of The Josh Bersin Company. He is frequently featured in publications such as *Forbes*, *Harvard Business Review*, *HR Executive*, *The Wall Street Journal*, *and CLO Magazine.* He is a popular blogger and has more than 800,000 followers on LinkedIn.

The Josh Bersin Company Membership

The Josh Bersin Company provides a wide range of research and advisory services to help HR leaders and professionals tackle the ever-evolving challenges and needs of today's workforce. We cover all topics in HR, talent, and L&D. The Josh Bersin Academy—built on our research and powered by Nomadic Learning—helps HR practitioners grow key foundational skills. Our corporate membership program provides HR teams and senior leaders with the skills, strategies, and insights to build cutting-edge HR and people strategies through a combination of research, assessments, professional development, exclusive events, and community. In 2022, The Josh Bersin Company introduced the Global Workforce Intelligence (GWI) Project to guide market-leading businesses and their leaders through the challenges of industry convergence while remaining future-focused.

For more details, contact us at info@bersinpartners.com.

Research Contributors

This research paper was developed after more than 40 in-depth interviews with CEOs, technologists, and engineering leaders at a variety of leading HR technology companies. We want to personally thank the many senior engineers and HR technology executives who took time to talk with us about their products, strategies, and architectures. In particular, we'd like to thank the following for their generous time and discussions:

- Abakar Saidov, CEO of Beamery
- Al Smith, CTO of iCIMS
- Amichai Schreiber, cofounder and CTO of Gloat
- Anoop Gupta, cofounder and CEO of SeekOut
- Aravind Bala, cofounder and CTO of SeekOut
- Ashutosh Garg, cofounder and CEO of Eightfold AI
- Eddie Raffaele, VP of AI and Machine Learning, Workday
- Karthik Suri, chief product officer of Cornerstone
- Mahe Bayireddi, cofounder and CEO of Phenom
- Meg Bear, president and chief product officer of SuccessFactors
- Sultan Saidov, cofounder and president of Beamery
- Dozens of other HR technology leaders who participated in interviews

All photography in this report was generated by Shutterstock.AI.