LEARNING MADE EASY

Snowflake Special Edition

Generative AI and LLMs

for dummies®

A Wiley Brand

Create new business insights with gen AI

Laying the gen AI base with a cloud data platform

Boost productivity with LLMs

Brought to you by

snowflake®

David Baum

About Snowflake

Snowflake enables every organization to mobilize their data with Snowflake's Data Cloud. Customers use the Data Cloud to unite siloed data, discover and securely share data, power data applications, and execute diverse AI/ML and analytic workloads. Wherever data or users live, Snowflake delivers a single data experience that spans multiple clouds and geographies. Thousands of customers across many industries, including 647 of the 2023 Forbes Global 2000 (G2K) as of October 31, 2023, use Snowflake Data Cloud to power their businesses. Learn more at snowflake.com.

Generative AI and LLMs

Snowflake Special Edition

by David Baum

for dummies®
A Wiley Brand

Generative AI and LLMs For Dummies®, Snowflake Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-394-23842-2 (pbk); ISBN 978-1-394-23843-9 (ebk)

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Development Editor: Nicole Sholly
Project Manager: Jennifer Bingham
Acquisitions Editor: Traci Martin
Editorial Manager: Rev Mengle
Sales Manager: Molly Daugherty
Content Refinement Specialist: Saikarthick Kumarasamy

Table of Contents

Introduction

Generative AI (gen AI) and large language models (LLMs) are revolutionizing our personal and professional lives. From supercharged digital assistants that manage our email to seemingly omniscient chatbots that can communicate with enterprise data across industries, languages, and specialties, these technologies are driving a new era of convenience, productivity, and connectivity.

In the business world, gen AI automates a huge variety of menial tasks, saving time and improving efficiency. It generates code, aids in data analysis, and automates content creation, freeing knowledge workers to focus on critical and creative tasks. It also enhances personal experiences by tailoring content to your preferences, delivering personalized recommendations for playlists, movies, and news feeds that enrich our daily lives.

Traditional AI uses predictive models to classify data, recognize patterns, and predict outcomes within a specific context or domain, such as analyzing medical images to detect irregularities. Gen AI models generate entirely new outputs rather than simply making predictions based on prior experience. This shift from prediction to creation opens up new realms of innovation. For example, while a traditional predictive model can spot a suspicious lesion in an MRI of lung tissue, a gen AI app can also determine the likelihood that a patient will develop pneumonia or some other type of lung disease and offer treatment recommendations based on best practices gleaned from thousands of similar cases.

Both in the public sphere of the Internet and within the realm of private enterprise, the transformative potential of this rapidly evolving field is reshaping the way people live, work, and interact.

About This Book

This book provides an introductory overview to LLMs and gen AI applications, along with techniques for training, tuning, and deploying machine learning (ML) models. The objective is to provide a technical foundation without "getting into the weeds," and to help bridge the gap between AI experts and their counterparts in marketing, sales, finance, product, and more.

In the pages that follow, you learn about the importance of gen AI applications that are secure, resilient, easy to manage, and that can integrate with your existing technology ecosystem. You also discover the importance of standardizing on a modern data platform to unlock the full potential of your data. Prepare to embark on a transformative journey that will shape the way your business operates.

Icons Used in This Book

Throughout this book, the following icons highlight tips, important points to remember, and more:

TIP

Tips guide you to easier ways to perform a task or better ways to use gen AI in your organization.

REMEMBER

This icon highlights concepts worth remembering as you immerse yourself in the understanding and application of gen AI and LLM principles.

TECHNICAL STUFF

The jargon beneath the jargon, explained.

Beyond the Book

If you like what you read in this book and want to know more, visit `www.snowflake.com`, where you can learn about the company and what it offers, try Snowflake for free, obtain details about different plans and pricing, view webinars, access news releases, get the scoop on upcoming events, access documentation, and get in touch with them — they would love to hear from you!

Disclaimer: Snowflake's AI features and capabilities that are referenced or described in this book may not be generally available, be different than described, or no longer exist at the time of reading.

IN THIS CHAPTER

- Reviewing the history of AI
- Emphasizing the role of data in gen AI projects
- Discussing the importance of gen AI to the enterprise
- Using a cloud data platform to manage gen AI initiatives

Chapter 1
Introducing Gen AI and the Role of Data

Traditional AI, often referred to as machine learning (ML), has primarily focused on analytic tasks like classification and prediction. *Generative AI* (gen AI) goes a step further with its ability to create new, original content. This creative breakthrough has the potential to transform nearly every industry, enhancing human creativity and pushing the boundaries of what machines can accomplish. This chapter puts gen AI in a historical context, defines key terms, and introduces the data foundation that organizations need to succeed with gen AI initiatives.

The Historical Context of Gen AI

Gen AI is a type of artificial intelligence that uses neural networks and deep learning algorithms to identify patterns within existing data as a basis for generating original content. By learning patterns from large volumes of data, gen AI algorithms synthesize knowledge to create original text, images, audio, video, and other forms of output. To understand the transformative nature

of these unique technologies, it is helpful to place them in their historical context.

AI has a rich history marked by decades of steady progress, occasional setbacks, and periodic breakthroughs. Although certain foundational ideas in AI can be traced back to the early 20th century, classical (or traditional) AI, which focused on rule-based systems, had its inception in the 1950s and came into prominence in the ensuing decades. ML, which involves training computer algorithms to learn patterns and make predictions based on data, emerged in the 1980s. At about this same time, neural networks gained popularity, inspired by the structure and functioning of the human brain. These software systems use interconnected nodes (neurons) to process information.

During the first two decades of the 21st century, deep learning revolutionized the AI landscape with its capability to handle large amounts of data and execute complex tasks. As a type of neural network, *deep learning* employs multiple layers of interconnected neurons, allowing for more sophisticated learning and representation of data. This breakthrough led to significant advancements in computer vision, speech recognition, and natural language processing (NLP), launching the era of general-purpose AI bots such as Siri and Alexa. *Convolutional neural networks* (CNNs) proved themselves to be particularly successful at computer vision tasks, while *recurrent neural networks* (RNNs) excelled in sequential data processing, such as language modeling. These technologies laid the foundation for gen AI.

Introducing LLMs and foundation models

Large language models (LLMs) are advanced AI systems designed to understand the intricacies of human language and to generate intelligent, creative responses when queried. Successful LLMs are trained on enormous data sets typically measured in petabytes (a million gigabytes). Training data has often been sourced from books, articles, websites, and other text-based sources, mostly in the public domain. Using deep learning techniques, these models excel at understanding and generating text similar to human-produced content. Today's LLMs power many modern applications, including content creation tools, language translation apps, customer service chatbots, financial analysis sites, scientific research repositories, and advanced Internet search tools.

REMEMBER

In the field of AI, language models are powerful software systems designed to understand, generate, and manipulate human language. Some models handle images and other media along with text. These are often referred to as *multimodal language models.*

Transforming the AI landscape

AI systems with humanlike reasoning capabilities have been around since the 1950s, but only with the advent of LLMs have they gained widespread adoption. According to a recent *Forbes* article called "Transformers Revolutionized AI. What Will Replace Them?" a key breakthrough came in 2017 when the Google Brain team introduced the *transformer architecture,* a deep learning model that replaced traditional recurrent and convolutional structures with a new type of architecture that's particularly effective at understanding and contextualizing language, as well as generating text, images, audio, and computer code.

LLMs based on the transformer architecture have enabled new realms of AI capabilities. Perhaps the best-known example is OpenAI's ChatGPT, which stands for chatbot generative pre-trained transformer. A CNN article, "Microsoft confirms it's investing billions in the creator of ChatGPT," shows support for the development of progressively larger LLMs, some of which may incorporate hundreds of billions of parameters to generate coherent and contextually relevant responses.

Accelerating AI functions

Another important factor in the evolution of AI is the advent of accelerated hardware systems known as graphics processing units (GPUs). Although central processing units (CPUs) are designed for general-purpose computing tasks, GPUs, initially developed for graphics rendering, are specialized processors that have proven to be adept at ML tasks due to their unique architecture.

TECHNICAL STUFF

GPUs have a large number of cores that can process multiple tasks simultaneously. Transformers use GPUs to process multiple threads of information, leading to faster training of AI models that effectively handle not just text but also images, audio, and video content. This parallel processing capability is crucial for the computationally intensive calculations involved in ML, such as

matrix operations. GPUs can perform these computations much faster than CPUs, accelerating training and inference times and enhancing the overall performance of ML algorithms. Refer to *Cloud Data Science For Dummies* (Wiley) by David Baum for additional information on these concepts. Figure 1-1 summarizes AI progress.

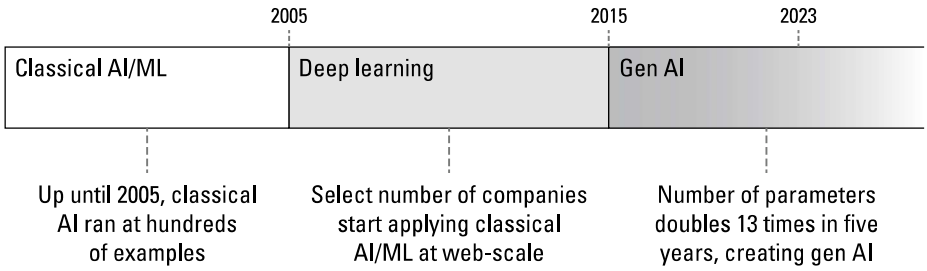

FIGURE 1-1: Gen AI builds on traditional AI concepts while vastly expanding applicability, scaling potential — and with web-scale processing demands.

The Role of Data in AI Projects

As impressive as they are at language generation, reasoning, and translation, gen AI applications that have been built on public data can't realize their full potential in the enterprise until they're coupled with enterprise data stores. Most organizations store massive amounts of data, both on-premises and in the cloud. Many of these businesses have data science practices that leverage structured data for traditional analytics, such as forecasting. To maximize the value of gen AI, these companies need to open up to the vast world of unstructured and semistructured data as well. According to a February 2021 report from MIT titled "Tapping the Power of Unstructured Data," 80 to 90 percent of data is unstructured — locked away in text, audio, social media, and other sources. For enterprises that figure out how to use this data, it can provide a competitive advantage, especially in the era of gen AI.

To amass a complete data set, consider not only your internal first-party data, but also second-party data from partners and suppliers, and third-party data from a service provider or data marketplace. See the nearby sidebar for more information.

CAST A WIDE DATA NET

To maximize the potential of your gen AI endeavors, cast a wide net to utilize the three basic types of data sources:

- **First-party data** is internal data produced via everyday business interactions with customers and prospects.
- **Second-party data** is produced by or in collaboration with trusted partners, such as product inventory data shared with an e-commerce or retail sales channel.
- **Third-party data** can be acquired from external sources to enrich internal data sets. Common examples include manufacturing supply chain data and financial market data.

Explaining the Importance of Generative AI to the Enterprise

Today's LLMs have paved the way for an immense array of advanced applications centered around content generation, logical reasoning, language translation, text retrieval, code generation, content summarization, and search:

- **LLMs for content generation:** Gen AI can streamline content creation by generating various types of media, including text, sound, and images. For instance, a marketing department can utilize gen AI to generate the first drafts of blogs, press releases, posts on X (formerly Twitter), and product descriptions, including producing custom images for promotional campaigns.

  One popular use of this technology in the enterprise is to develop chatbots that engage in conversational interactions with business users, helping them obtain accurate answers to their questions. By harnessing private data such as customer transaction histories and customer service records, these systems can even deliver personalized content to target audiences while maintaining data security. LLMs are also adept at analyzing documents, summarizing unstructured text, and converting unstructured text into structured table formats.

- **LLMs as logical reasoning engines:** Within the field of AI, *natural language understanding* (NLU) focuses on comprehending the intricate meaning in human communication. LLMs can unravel the underlying meaning in textual data, such as product reviews, social media posts, and customer surveys. This makes them valuable for sentiment analysis and other complex reasoning tasks that involve extracting meaningful insights from text and providing a deeper understanding of human language.
- **LLMs as translation engines:** LLMs have transformed text translation between languages, making it easier for people to communicate across linguistic barriers. By leveraging this understanding, LLMs can accurately convert text from one language to another, ensuring effective and reliable translation. This breakthrough in language processing has greatly enhanced accessibility and global communication, allowing individuals and businesses to connect, collaborate, and understand each other more easily, regardless of language differences.
- **LLMs for text retrieval, summarization, and search:** LLMs are pretrained on vast amounts of text data, allowing them to grasp the nuances of language and comprehend the meaning of text. They can search through large databases or the Internet in general to locate relevant information based on user-defined queries. LLMs can also generate concise summaries while maintaining the essence of the original information. For example, a tech company might use an LLM to optimize content for search engines by suggesting relevant keywords, giving visibility into common search queries associated with the topic, and ensuring crawlability.

REMEMBER

Gen AI models, and hence the decisions made from those models, are only as good as the data that supports them. The more data these models ingest and the more situations they encounter, the smarter and more comprehensive they become.

Pretrained models

There's a rapidly growing market for creating and customizing gen AI foundation models in many different industries and domains. This has given rise to a surge of LLMs that have been pretrained on data sets with millions or even billions of records,

allowing them to accomplish specific tasks. For example, as explained by SiliconAngle's "Nvidia debuts new AI tools for biomolecular research and text processing," MegaMolBART (part of the NVIDIA BioNeMo service and framework) can understand the language of chemistry and learn the relationships between atoms in real-world molecules, giving researchers a powerful tool for faster drug discovery. Pharmaceutical companies can fine-tune these foundation models using their own proprietary data. Training these commercial foundation models is an immense effort that costs tens of millions of dollars. Fortunately, businesses that use them don't have to repeat that massive process to adapt an LLM to their needs; they can adapt an existing foundation model for a fraction of that amount.

Large technology companies are constantly inventing new model architectures, even as they expand the capabilities of their existing LLMs (for more on this, see Chapter 2). Thousands of open-source models are available on public sites such as GitHub and Hugging Face. Developers can use the pretrained AI models as a foundation for creating custom AI apps.

Security versus ease of use

All logical reasoning engines need data to function. Although many of today's LLMs have been trained on vast amounts of Internet data, they become even more powerful and relevant when they're trained with enterprise data. Because much of this data is protected in private databases and resides behind network firewalls, the challenge facing today's enterprises involves augmenting LLMs with this corporate data in a secure and governed manner.

Gen AI systems learn from data; the more data they can access, the more capable they become. But how do you ensure that your business users, software developers, and data scientists can easily access a secure, consistent, governed data set — without adding onerous constraints that inhibit innovation? Enterprises need to be able to leverage gen AI technology in an easy, straightforward manner. They also need to uphold essential data security, governance, and regulatory issues — not only for their data but also for the models that learn from the data and extract information from it.

How can you achieve this without squelching innovation? You start by unifying data in a comprehensive repository that multiple

workgroups can access easily and securely. This allows you to centralize data governance and democratize access to gen AI initiatives across your organization while minimizing complexity and optimizing costs.

Managing Gen AI Projects with a Cloud Data Platform

A cloud data platform is a specialized cloud service optimized for storing, analyzing, and sharing large and diverse volumes of data. It unifies data security and data governance activities by ensuring that all users leverage a single copy of data. It fosters collaboration and ensures that the organization has a scalable data environment for new foundation models and related analytic endeavors. A cloud data platform extends your AI horizons by allowing you to store your first-party data and leverage a network of data from second- and third-party data providers as well. It provides a protected ecosystem where you can easily and securely share models and data sets, internally and with partners and customers.

By utilizing a cloud data platform, you can seamlessly leverage existing infrastructure to support gen AI initiatives with minimal hassle. As a fully managed service, the platform eliminates the need to deal with the complexities and technical overhead of building and managing infrastructure. You can easily provision and effortlessly scale compute resources for each type of data, such as GPUs for model training, fine-tuning, and inference activities. Finally, by using the same core data foundation for all your data-driven initiatives, you can ensure consistency and reliability in managing your gen AI, data science, and analytics projects.

TIP

Data is your core differentiator in the age of gen AI. The best way to harness and protect enterprise data for gen AI initiatives is to consolidate disparate sources into a cloud data platform that provides strong security and governance for data and the models customized with that data. Data can be structured tabular data; semistructured data from IoT devices, weblogs, and other sources; or unstructured data, such as image files and PDF documents.

IN THIS CHAPTER

- Categorizing and classifying LLMs
- Reviewing the technologies that power LLMs
- Understanding the role of vector databases
- Identifying LLM terms, concepts, and frameworks
- Reiterating the importance of data governance

Chapter 2
Understanding Large Language Models

Large language models (LLMs) are widely known for their ability to generate written text, computer code, and other content, as well as for their astonishing ability to respond to queries in humanlike ways. However, the utility of these AI systems extends beyond explaining concepts and summarizing text. Today's LLMs have the potential to revolutionize how enterprises acquire, handle, and analyze information, opening up new avenues for exploration and inquiry. This chapter defines the various types of LLMs and discusses their applicability to the enterprise.

Categorizing LLMs

General-purpose LLMs handle a wide range of tasks and understand a broad spectrum of languages — both natural languages and computer languages. They are trained by scraping massive amounts of data from the Internet, as well as by ingesting data

from private data sources that are relevant to the purpose of the model. This allows LLMs to generate contextually related feedback on just about any topic.

Foundation models are a class of generative AI (gen AI) models that are well suited for a wide range of use cases. To increase their usefulness at a specific task, these models can be specialized, trained, or modified for specific applications. Common foundation models include the following:

- **Task-specific LLMs** such as Meta's Code Llama specialize in unique, highly targeted tasks like generating software code.
- **Domain-specific LLMs** apply gen AI technology to specific subjects and industries. For example, NVIDIA's BioBERT, which has been trained on biomedical text, helps researchers understand scientific literature and extract information from medical documents.

Domain-specific and task-specific models are fine-tuned using data specific to the domain they're built for, such as law, medicine, cybersecurity, art, and countless other fields. They aren't limited to language. Some of them can also generate music, pictures, video, and other types of multimodal content.

REMEMBER

An LLM is a general-purpose model primarily useful for tasks related to unstructured text data. A *foundation model* serves as the basis for developing specialized applications adapted to specific industries, business problems, and use cases. A foundation model can often be *multimodal*, meaning it handles both text and other media such as images.

Defining general-purpose LLMs

GPT-3, a general-purpose LLM, was developed by OpenAI based on the Generative Pre-trained Transformer (GPT) series of machine learning (ML) models. ChatGPT isn't a language model per se, but rather a user interface tailored around a particular language model such as GPT-3, GPT-3.5, or GPT-4, all of which have been optimized for conversational interactions within the ChatGPT LLM platform. Other popular LLM options are described in the nearby sidebar.

SIZING UP THE CONTENDERS

As the software industry steps up research and development into LLMs, several prominent offerings have emerged in this highly competitive sector:

- **OpenAI's GPT** family is based on the GPT series of models. These LLMs are renowned for their impressive language-generation capabilities and capability to perform well across various language tasks, including search, text generation, reasoning, and multimedia content delivery.
- **Bidirectional Encoder Representations from Transformers (BERT),** developed by Google, employs a masked language model to learn contextual representations, enabling it to better comprehend the meaning of sentences. This model powers Google Bard's conversational AI chat service.
- **Llama (**Large Language Model Meta AI) is a family of LLMs introduced by Meta that excels at language translation, text generation, and question-answering. Llama 2 is an open-source model that is available for research and development purposes.
- **Code Llama,** also developed by Meta AI, is a language model tailored to understand and generate code snippets and programming instructions. It has been trained to assist in coding tasks, code completion, and suggesting efficient coding techniques.
- **Snowflake Copilot,** an LLM fine-tuned by Snowflake, generates SQL from natural language and refines queries through conversation, improving user productivity.
- **XLNet,** developed by Carnegie Mellon University and Google, focuses on generating high-quality text in multiple languages, making it useful for language translation and content creation.

Those listed in the sidebar and other language models are becoming progressively more relevant to the business world. According to a June 1, 2023, press release from Bloomberg Intelligence, the gen AI market is poised to explode, growing to $1.3 trillion over the next 10 years from a market size of just $40 billion in 2022 — a compound annual growth of 42 percent.

Using task-specific and domain-specific LLMs

Bing and Bard are examples of applications developed utilizing their respective foundation LLMs. These applications have a user interface and have undergone additional specialized training, enhancing their capabilities for specific tasks. For example, Bard offers chatbot access to Google's full suite of products — including YouTube, Google Drive, Google Flights, and others — to assist users in a wide variety of tasks. Google users can link their personal Gmail, Google Docs, and other account data to allow Bard to analyze and manage their personal information. For example, you can ask Bard to plan an upcoming trip based on suggestions from a recent email string, complete with flight options. You can also ask Bard to summarize meeting notes you have logged in the files and folders of your Google Drive hierarchy.

Domain-specific LLMs focus on a specific subject area or industry. For example, BioBERT is trained on biomedical text, making it an excellent resource for understanding scientific literature and extracting information from medical documents. CodeBERT is a cybersecurity solution that has been trained to assist with IT security concerns such as vulnerability detection, code review, and software security analysis. These specialized LLMs can be further trained and fine-tuned using data specific to targeted areas of interest, and can incorporate additional sources of data to build proficiency on designated subjects. To gain adoption and drive value from models, AI teams must build user interfaces that allow users to interact with these LLMs in designated ways.

Reviewing the Technology Behind LLMs

Neural networks are a key component of AI systems. As discussed in the previous chapter, most neural networks use a combination of complex recurrent or CNN structures to do their jobs. However, today's gen AI models also have an attention mechanism that helps the *encoder* (the part that understands the input) and the *decoder* (the part that generates the output) work together more effectively (see Figure 2-1).

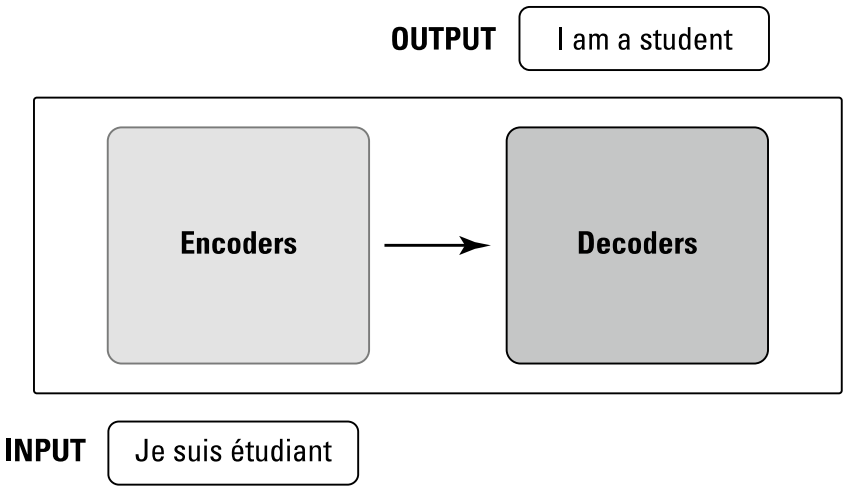

FIGURE 2-1: A gen AI's encoder and decoder work together to produce accurate outputs.

It all springs from the advent of the transformer architecture (introduced in Chapter 1), which presented a new and simpler way to understand language using attention mechanisms. Unlike the previous models, transformer models eliminate dependencies on other processing steps, giving them the added benefit of being more parallelizable, which accelerates training.

The power of the transformer architecture lies in its capability to learn the relevance of each word to all the other words in a statement or document through self-attention mechanisms, in conjunction with training on large data sets.

Introducing key terms and concepts

Today's LLMs are easy to use. Rather than requiring formal code to communicate with software libraries and application programming interfaces (APIs), they can understand natural language or human instructions and perform tasks similar to a human. The text you provide to a language model is called a *prompt*. The prompt is given to the model, which then generates an answer. The result produced by the model is known as a *completion*, and the process of using the model to generate text is called *inference*.

As you will see in Chapters 3 and 4, users can influence the learning process by providing well-crafted prompts or by using techniques such as *reinforcement learning with human feedback (RLHF)* to guide the model's output. You'll also learn about *prompt engineering* and *in-context learning (ICL)*, which allow you to guide the model at inference time, as well as how to

fine-tune input parameters as you instruct the LLM to generate relevant outputs specific to your private data.

Explaining the importance of vector embeddings

LLMs deal with large and complex data sets. By representing these data sets in vector form, where words are represented by numbers in a multidimensional space, it becomes easier for the models to compare and analyze information. Vector databases store data as mathematical representations that can be easily parsed by ML models. These vector "embeddings" are the most efficient way to process and store quantitative representations of natural language. Data can be identified based on similarity metrics, such as distance between two vectors, instead of exact matches, which saves a tremendous amount of processing time (see Figure 2-2).

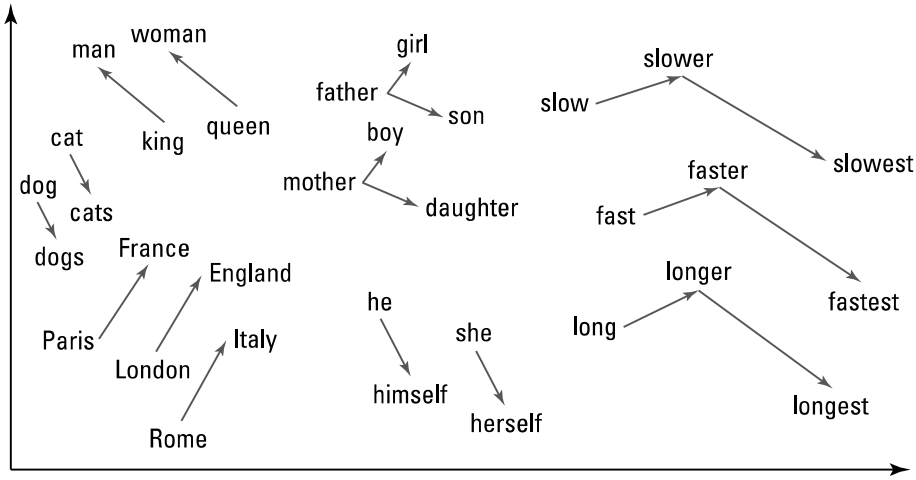

FIGURE 2-2: A word can be identified by the group of words that are used with it frequently. This is the basis for creating vector embeddings.

Vector databases enable gen AI systems to quickly retrieve relevant data during the generation and inference processes. They are particularly useful for ML models because of their capability to power key language applications such as search, recommendations, and text generation.

TIP

Make sure that your data platform includes vector search functionality that can handle essential gen AI tasks that help you contextualize LLMs with your data, such as retrieval-augmented generation (RAG), in-context learning (ICL), and vector similarity search (VSS). For more information on these concepts, see Chapter 3.

Identifying developer tools and frameworks

The capability of LLMs to process and interpret vast amounts of text, audio, video, and other forms of content have made them an indispensable part of many data science workflows. Although nontechnical users can interact with LLMs with little or no training, data science teams use established software frameworks to interact with LLMs and create gen AI applications. Popular frameworks that assist with text classification, sentiment analysis, text generation, and other gen AI tasks include the following:

- **OpenAI GPT Playground:** OpenAI provides pretrained language models, such as GPT-3, which can be accessed via APIs for various language understanding and text-generation tasks. The GPT Playground allows users to experiment with these models and fine-tune prompts interactively.
- **Snowflake Cortex:** An intelligent, fully managed service that offers access to industry-leading AI models, LLMs, and vector search functionality on secure and governed enterprise data.
- **Hugging Face Transformer Library:** An open-source library that offers a high-level API for working with pretrained language models like BERT and GPT. The library includes a user-friendly interface and prebuilt functionality for fine-tuning LLMs, with straightforward APIs and command-line tools to simplify the process.

Enforcing data governance and security

Models need abundant data relating to the problems they're attempting to solve. But how do you put the right data in the hands of LLM users without exposing sensitive information, compromising data privacy, or putting your brand at risk?

A comprehensive cloud data platform allows you to work with LLMs within a protected environment. Rather than "taking your data to the processing engine," it allows you to "bring your processing to the data," where you can control user access to corporate data sources and enforce security and governance policies. If you can run that model as a service within your cloud data platform, you can ensure that data, prompts, and completions are not shared with unauthorized users.

TIP

All gen AI project stakeholders should take care to protect sensitive data as it is accessed, shared, and exchanged with LLMs and gen AI applications. When you conduct these projects within a cloud data platform, you can uphold data security, data privacy, and data governance without imposing intrusive restrictions on the workforce.

Extending governance for all data types

A modern cloud data platform extends governance to all types of data — structured, semistructured, and unstructured. This is a unique capability not necessarily offered by cloud service object stores such as Amazon S3, Microsoft Azure Blob Storage, and Google Cloud Storage. These cloud services make it relatively easy to store unstructured data, such as PDF files, audio, and video, as binary large objects (blobs). However, granular access controls such as row-level permissions aren't always available at the blob level. These services may broaden your access to complex data types but not without increasing risk. Even if you choose to store your data in a cloud object store, be sure that your cloud data platform can provide governance on top of that data.

REMEMBER

Loading all data into a centralized repository with a cohesive layer of data governance services allows you to enforce universal policies that broaden access while reducing risk. Applying these controls in a modern cloud data platform that supports structured data, semistructured data, and unstructured data is easier and less risky.

IN THIS CHAPTER

- Defining the scope of the project
- Selecting an appropriate LLM
- Adapting LLMs to particular tasks
- Exploring prompt engineering, fine-tuning LLM models, and more
- Exposing data and LLMs as applications

Chapter 3
LLM App Project Lifecycle

The generative AI (gen AI) project lifecycle guides you through the process of selecting, adapting, and implementing large language models (LLMs) as you create AI applications. This chapter describes the major steps including defining the use case, selecting an LLM, and guiding the use and customization of the model for the project. It also covers the key considerations in model customization steps.

Defining the Use Case and Scope

The first step in the gen AI project lifecycle (see Figure 3-1) is to identify the business problem or use case. For example, you might want to use gen AI to create personalized product descriptions, summarize transcripts, extract answers from documents, create compelling characters for a video game, or to train a computer vision system to recognize particular objects.

Next, determine what proprietary data you will use to customize or contextualize the model effectively. Many foundation LLMs contain massive amounts of information learned from the Internet, which gives the models their knowledge of language as well as many aspects of the world around us. More than likely, you have a project in mind that requires specific domain knowledge or access to internal information. For example, to create product descriptions for an e-commerce system, you might begin with public data that describes the general types of products, supplemented by internal data that identifies the unique aspects of your products.

TIP

Defining the use case sets the foundation for clarifying the business problem and or the goal to be achieved. It will help define data and user experience requirements.

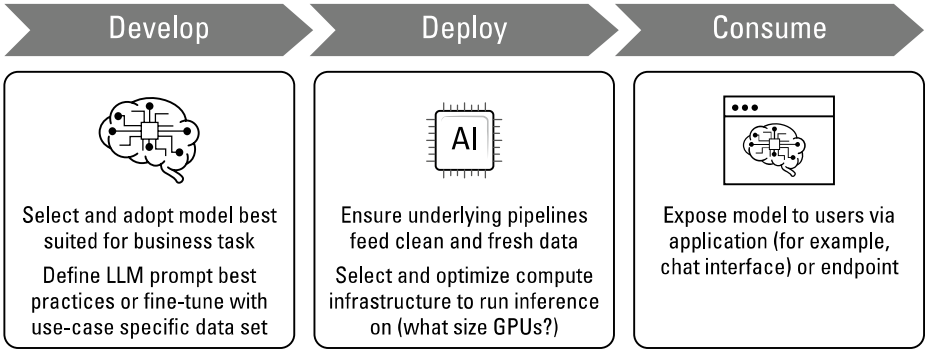

FIGURE 3-1: The gen AI project lifecycle and its players at a glance.

Selecting the right LLM

Many different language models are available for public use (for more on this, see Chapter 2). Hosted LLMs such as ChatGPT and Bard are provided as a service that anybody can access, via a user interface or via APIs. This makes interacting with the LLMs very easy because there's no overhead to host and scale the infrastructure where it runs. Open-source LLMs like Llama are freely available for download and modification, and you can deploy them in your own environment. Although this gives you more control, it also requires you to set up and maintain the underlying infrastructure. Not sending data to an external environment and having more control over the model may be of high importance for sensitive data, but the additional control puts the compute infrastructure management in your hands.

Comparing small and large language models

The *parameters* in a language model refer to the trainable variables. More parameters mean more knowledge is part of the model out of the box, but bigger isn't always better. Smaller LLMs have fewer parameters and thus consume less compute resources and are faster to fine-tune and deploy. They're well suited for running very specific tasks in a more cost-effective way. LLMs have a higher number of parameters (typically 10 billion or more) and can learn more nuanced language patterns, and provide more accurate and contextually relevant outputs for a wider range of scenarios. However, they require more resources to train and adapt to your needs.

TECHNICAL STUFF

For example, with only 117 million parameters, GPT-2 is a good choice for a narrow set of tasks such as language completion and summarization. With 175 billion parameters, GPT-3 is better for complex tasks, such as translating text and generating dialogue.

REMEMBER

The choice between small and large language models is a cost-performance tradeoff, it all depends on the set of use cases that need to be supported by a single model.

SELECTION CRITERIA FOR LLMS

When selecting the right LLM for a gen AI project, consider:

- **Task alignment:** Choose an LLM that aligns to the task, such as GPT for conversational applications or BioBERT for biomedical research.
- **Training data:** Evaluate whether the LLM has been trained on data that matches the domain or context of the project.
- **Model size and complexity:** Models with tens of billions of parameters provide higher-quality outputs but require more computational resources.
- **Adapting and tuning:** Determine if the chosen LLM can be effectively contextualized with prompts or fine-tuning.
- **Ecosystem and support:** Investigate the availability of resources, tools, and community support surrounding the LLM.

This list highlights various techniques that you can use to tailor the LLM to meet your specific needs:

- **Prompt engineering** is the process of optimizing text prompts to guide the LLM to generate pertinent responses.
- **In-context learning (ICL)** allows the model to dynamically update its understanding during a conversation, resulting in more contextually relevant responses.
- **Retrieval-augmented generation (RAG)** combines retrieval and generation models to surface new relevant data as part of a prompt.
- **Fine-tuning** entails customizing a pretrained LLM to enhance its performance for specific domains or tasks.
- **Reinforcement learning from human feedback (RLHF)** is the ongoing approach to fine-tuning in near real time by providing the model with feedback from human evaluators who guide and improve its responses.

Each of these techniques is described in further detail in the following sections.

Engineering prompts

LLMs are sophisticated predictive models that anticipate the next word in a sequence based on the context provided to them as part of a process referred to as a *completion*. Carefully constructed prompts help these models deliver tailored content, yielding better completions. LLM performance is influenced not only by the training data but also by the context provided by these user inputs — the prompts.

Prompt engineering is the practice of crafting inputs to shape the output of a language model and achieve a desired result. For instance, if you're using the LLM to generate a summary of a 20-page research paper, you can engineer the prompt by specifying which sections of the document are most important, and what should be the word count of the output. By carefully crafting the prompt, you can obtain more targeted and relevant responses. Zero-shot, one-shot, and few-shot prompting are all techniques used in prompt engineering.

- *Zero-shot prompting* is the default; you simply issue a question and rely on the LLM's pretrained information to answer it.
- With *one-shot prompting,* you include an example of the desired output to help the model understand the desired output. For instance, assume you're writing a travel brochure and you want the AI model to describe a vibrant public market in an exotic city. Even with limited exposure to this particular scenario, a model can generate a creative description that matches the tone of voice, vibrant use of adjectives, and structure that has already proven successful in your marketing content.
- *Few-shot prompting* takes it a step further by providing multiple examples to more clearly teach the LLM the desired output structure and language.

REMEMBER

Prompt engineering involves carefully crafting prompts to coax the language model toward a desired outcome based on explicit instructions and context. A carefully constructed prompt will help ensure clarity, provide insight to influence model performance.

Learning from context

Although prompt engineering involves carefully designing and refining instructions to help the LLM formulate a useful and accurate completion, *in-context learning* (ICL) involves training a language model with a data set that aligns with the desired context or domain. By exposing the model to contextual information, such as relevant documents or proprietary content, it becomes better equipped to generate accurate and coherent responses within that context. For example, you could use ICL to train a customer support chatbot on your company-specific documents, emails, and technical support tickets, helping it to respond more effectively to questions about your organization's products and services.

REMEMBER

ICL allows users to give context to the model using private data, enhancing its performance in specific domains. This is a simple way to help language models understand and generate text that's contextually relevant to specific tasks, scenarios, and domains.

Augmenting text retrieval

RAG leverages a pretrained language model in conjunction with a large-scale vector search system to enhance the content-generation process. It addresses some of the common problems

associated with gen AI systems, such as limited knowledge of arcane subjects, a failure to recognize facts and events that took place after the model was trained, and lack of insight into any proprietary data.

RAG accesses up-to-date information by retrieving relevant data stored as *vectors* (numerical representation of the data for fast retrieval), such as current news, to bring the model up to date with recent events or domain-specific content from a particular industry or market. You can augment LLMs by allowing them to access your data and documents, including private wikis and knowledge bases. By retrieving this additional information, models can produce more accurate and contextually appropriate responses. For example, you might use RAG to generate purchase recommendations in a chat by allowing it to retrieve information on a customer's stated preferences and purchase history, enabling more personalized interactions.

Fine-tuning language models

Fine-tuning enables you to adapt an LLM to particular tasks by updating the parameters of a pretrained model. These techniques empower users to shape LLMs according to their preferences and achieve better results in various applications.

ADJUSTING MODEL PARAMETERS

Fine-tuning allows you to adjust a model's parameters to achieve better results. There are three basic steps to the process:

1. Select the pretrained LLM that is most germane to the use case.
2. Identify data sets related to the use case to refine the LLM.

   Teach the model how to respond based on a training data set that includes examples of prompts as well as the data the model needs to answer or complete the prompt. In this process, the model weights are adjusted to get better at generating responses to the new set of prompts.
3. Evaluate the fine-tuned LLM to verify results meet requirements.

   You can adjust the learning rate, batch size, and other factors to improve outcomes.

REMEMBER

By applying fine-tuning techniques, you can adapt a model to specific needs and use cases. You can also rapidly build customized solutions by building on top of an existing foundation model rather than training a new model from scratch.

Reinforcement learning

Reinforcement learning from human feedback (RLHF) is a form of fine-tuning that you can use to guide the learning process and further enhance the performance and behavior of your model, with the goal of improving its responses over time. Many creators of large language systems use this technique to teach their chatbots to carry on realistic conversations — such as to engage in dialogue rather than just provide one-off responses.

You can use RLHF to train your models to better understand human prompts and generate more humanlike responses, as well as to ensure that the model aligns with your preferences. RLHF can also help you minimize the risk of harmful content by training the model to avoid toxic language or to avoid sensitive topics.

Tuning models to learn individual preferences opens the door to exciting new applications of gen AI technology. For example, in the business world, you can create personalized assistants. In the field of education, you can develop tailored learning plans that meet the unique needs of each student. In healthcare, you can leverage patient data and clinical expertise to create personalized treatment plans, resulting in more effective and precise medical interventions. In the entertainment industry, you can use RLHF to generate personalized recommendations for users, enhancing their viewing or listening experiences.

TIP

RLHF can improve a model's performance over the original, pretrained version. It can also help you to avoid potentially harmful or inaccurate language that results when models are trained on data from the Internet, where such language is common.

Using a vector database

LLMs use vector embeddings to represent textual elements, with each word mapped to a token ID and transformed into a vector (for more on this, see Chapter 2). These mathematical representations enable efficient storage and searching of data, as well as identification of semantically related text.

A vector database is important because it enables efficient storage, retrieval, and manipulation of vector embeddings. By assigning a unique key to each vector, the database allows for quick and direct access to the content at a discrete level. This capability is particularly valuable in applications like RAG, where rapid retrieval and matching of vectors allow the model to discover semantically related text, such as a product that's similar to one that a customer searched for previously.

Implementing LLM Applications

Selecting and adapting an LLM is an iterative process. Once you have a model that's well aligned with your needs, you can deploy it to continuously run inference as a stand-alone service or as part of an application user interface.

Deploying apps into containers

Many DevOps teams use *containerization software,* such as Docker, to package their LLM applications. Containers can be consistently deployed across many types of computing environments. This is useful for sophisticated AI models, which may have special processing needs and require access to massive amounts of proprietary data.

Unfortunately, the complexity of creating, developing, and running AI workloads at scale forces developers and data scientists to spend their time managing containers rather than developing new applications.

One solution is to standardize on a cloud data platform that enables you to deploy, manage, and scale LLMs and other containerized workloads via an infrastructure *that is fully managed by the data platform itself*. This allows you to run LLM jobs within a governed environment, and to take advantage of configurable hardware options, such as graphics processing units (GPUs).

A comprehensive cloud data platform includes a scalable pool of compute resources, which insulates your team from the complexities of managing and maintaining infrastructure. Moving governed data outside of the data platform (thereby exposing it to additional security risks) isn't necessary to use it within your AI models and applications.

Some data platforms also allow you to use data, applications, and models from third-party providers, instantly available as native apps within an associated marketplace.

REMEMBER

Developers and data scientists in large enterprises and other organizations must deal with massive amounts of proprietary data to run, tune, and deploy their models. Unfortunately, many of these scarce technology professionals must spend their time managing compute and storage resources for these applications rather than focusing on the business problems they're trying to solve. Select a cloud data platform that offers serverless access to LLMs as well as container services for running gen AI apps in a fully governed, easy-to-provision, managed services environment.

Allocating specialized hardware

Having the right hardware is essential for training, tuning, and running LLM models. GPUs can accelerate training and inference processes. Ideally, your cloud data platform should automatically provision these hardware resources in the background.

To maximize deployment flexibility, select a cloud data platform that is *cloud agnostic,* which means it can work with the major cloud service providers (CSP). That way, you can abstract where the model runs and easily move it to the cloud that makes the most sense, whether it's because of where the data is or where the end-user application is hosted.

Integrating apps and data

As discussed throughout this chapter, LLMs and other AI models often leverage unique, individualized data sets to improve the relevance and accuracy of their outputs. Capturing, processing, storing, and synchronizing data among projects can be very complex, sometimes exposing your organization to data privacy violations and compliance risks.

By leveraging a centralized cloud data platform that provides near-unlimited data storage and compute power, gen AI stakeholders can acquire the data they need to use, customize, and contextualize new applications quickly, either using open-source models as-is or models fine-tuned to specific data sets. The platform should empower any user to incorporate LLMs into analytical processes quickly, and developers can create AI-powered apps in minutes, whether it uses an LLM or a more traditional machine

learning (ML) model. And in the same platform, teams can collaborate in executing all related tasks such as data preparation, data engineering, analytics, and other important workloads (see Figure 3-2).

Leading cloud data platform vendors offer tools to build gen AI applications such as chatbots and search engines using the necessary building blocks for LLM app development that come natively integrated. They may also offer tools for ingesting and querying documents, such as loading legal contracts, invoices, rental agreements, and many other types of content, then use the reasoning engine within the LLM to instantly extract meaning from them.

If the data platform includes a marketplace, you can avail yourself of gen AI apps from other data platform users, as well as package your own AI models and distribute them as applications to other marketplace participants.

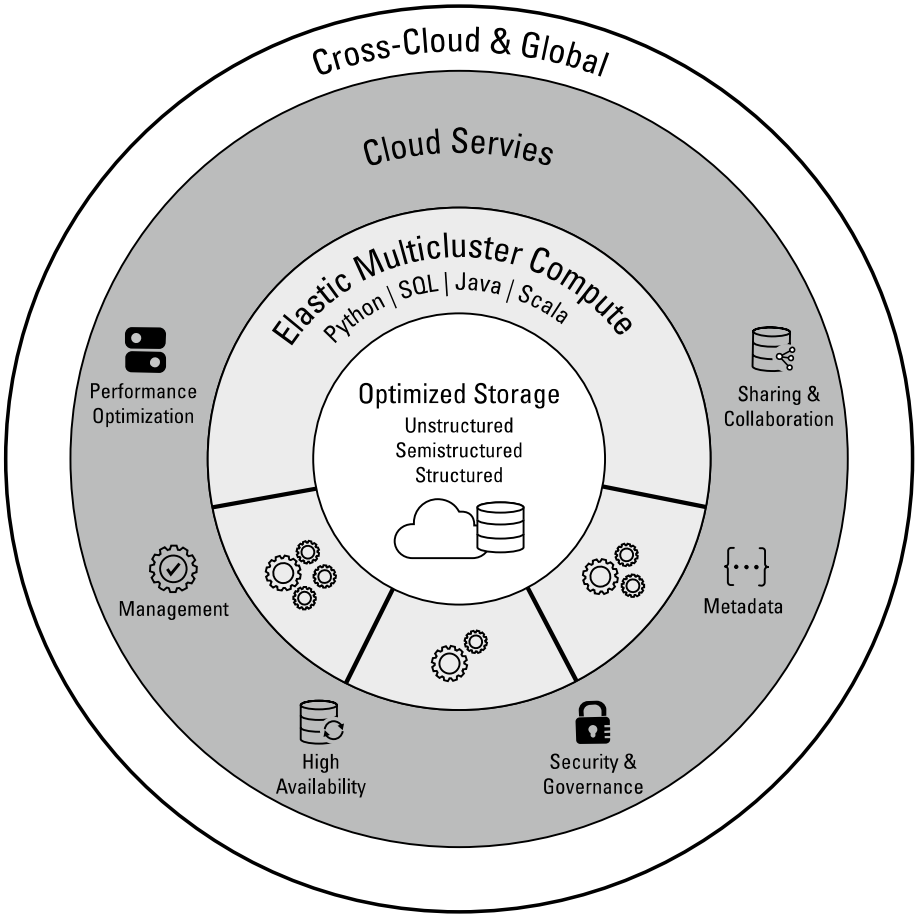

FIGURE 3-2: A cloud data platform that supports many data types and spans public clouds can unite the work of data analysts, data scientists, and data engineers.

IN THIS CHAPTER

- Understanding semantic caching, feature injection, and context retrieval
- Exploring processing for inference
- Developing interactive applications with user-friendly interfaces
- Orchestrating AI agents
- Splitting and chaining prompts

Chapter 4
Bringing LLM Apps into Production

As you progress beyond using a model out of the box without any customization and begin incorporating custom data into your language models, you will most likely need to assess your data processing needs. This chapter discusses the challenges of bringing large language model (LLM) apps into production, including building data pipelines, improving model accuracy, calculating costs, orchestrating external data sources, developing user interfaces, and calling external functions.

Adapting Data Pipelines

Data pipelines play a critical role in gen AI initiatives by facilitating the smooth flow of data, including efficient data ingestion, preprocessing, training, and inference. By establishing robust data pipelines, data engineers can ensure a continuous and reliable supply of high-quality data, which is vital for training accurate and effective models.

By integrating your gen AI initiatives into your existing data infrastructure, you can often avoid building data pipelines and other foundational services (for more on this, see Chapter 1). A cloud data platform provides scalable GPU-infrastructure and vector search functionality as well as flexibility to reuse and adapt existing data pipelines developed for other downstream processes — such as business intelligence, analytics, and machine learning (ML) — without causing bottlenecks.

In addition to the traditional data pipelines where data is cleaned, curated, and governed, you need to pay attention to semantic caching, feature injection, and context retrieval. These are integral components of data pipelines that feed the LLMs because they enhance the performance, personalization, and accuracy of AI models.

Semantic caching

Semantic caching involves temporarily storing semantic representations or embeddings of data. By employing semantic caching techniques, AI systems can provide more precise, meaningful, and efficient responses. For example, let's say that you have a chatbot that needs to generate responses to user queries. Before the bot goes live, it can precompute the semantic representations of a large set of possible user queries and store them in a cache. These representations capture the underlying meaning or intent of the queries. When a user interacts with the chatbot, instead of processing the query from scratch, the system can retrieve the precomputed semantic representation from the cache. This significantly reduces the computational overhead and speeds up the response generation process.


REMEMBER

Semantic caching allows AI systems to generate responses more quickly and efficiently by having relevant data easily accessible for computations. It can be particularly useful in scenarios where real-time or near real-time responses are required, such as in chatbots, virtual assistants, or recommendation systems.

Feature injection

Feature injection refers to the process of incorporating additional information or features into the AI model. Although *feature engineering* is a general practice in data science, feature injection is a specialized technique used to enhance the performance and

reasoning capabilities of AI models. Features can improve the model's ability to handle specific tasks. By injecting relevant features, the model can capture and leverage important patterns in the data, leading to improved performance.

TIP

By introducing features that are relevant to a specific prompt, LLMs can gain a deeper understanding of the data and can better capture complex patterns and relationships.

Context retrieval

Context retrieval involves retrieving relevant contextual information to enhance the understanding of AI models. By considering the surrounding context, such as previous interactions or user history, AI systems can generate more accurate and personalized completions. For example, a customer support system might use context retrieval to provide personal assistance. If a customer has previously interacted with the support system and mentioned a specific issue or order number, the system can retrieve that context to better understand the customer's current inquiry or concern. Retrieval-augmented generation (RAG), a type of in-context learning (ICL), is also important in this context. See Chapter 3 for additional information.

Processing for Inference

Processing for inference involves running the necessary computations to apply a trained model to new data and generate predictions or outcomes. For example, if you're creating a generative AI (gen AI) app to quickly analyze hundreds of contracts to identify areas of business risk, you need to help the model learn what types of language or clauses to spend more time on. The app that sits on top of the model becomes the interface for the legal and risk teams to either adjust the contracts or find ways to mitigate risks on existing agreements. There are three primary considerations:

- **Infrastructure:** This involves selecting suitable hardware infrastructure. GPUs are specifically designed to handle parallel computations, making them well suited for running LLMs efficiently. The choice of GPU depends on factors such as the model's size, memory requirements, and latency constraints.

- **Access:** The model can be hosted either on private servers or cloud platforms. Hosting in the cloud offers benefits such as scalability, easy deployment, and maintenance. However, this choice raises data security and governance concerns, emphasizing the need for proper measures to protect sensitive data and ensure compliance with relevant regulations. A cloud data platform can alleviate these concerns.
- **Consumption:** LLMs can be accessed via APIs and software functions, enabling programmatic integration. They can also be accessed through application-specific interfaces, making them accessible to a broader range of users. The choice of access method depends on the specific use case, target audience, and the level of flexibility desired.

Reducing latency

Latency refers to the time it takes the LLM to make predictions once it receives input data. This is an important consideration for gen AI projects that require real-time responses. For example, if you're creating a customer support chatbot, low latency is crucial to provide quick and efficient responses to customer inquiries, enabling real-time interactions. As described throughout this book, *keeping the processing close to the data* is a key strategy for reducing the latency of gen AI applications in a production environment. It also allows you to reduce the amount of data that needs to be transferred between the compute resources and the storage layer, improving performance while reducing costs and data security risks.


TIP

Various factors can impact AI performance, including the complexity or size of your model, the amount of input data, and the network latency between the processing and data storage layers. To reduce latency and improve overall performance, consider using smaller models, optimizing models for inference, using efficient hardware and software, and keeping the processing close to the data. This strategy can also improve the scalability of gen AI applications. By distributing the processing across multiple compute resources, you can scale these systems to handle larger volumes of data and more complex workloads, all while reducing the latency of predictions.

Calculating costs

The cost of using a cloud data platform is typically based on three interrelated metrics: data transfer volume, data storage consumption, and compute resources. The best data platforms separate these three services to give administrators complete control over usage. Your data platform should make it easy to track the consumption of all cloud services. This includes built-in resource monitoring and management features that provide transparency into usage and billing, ideally with granular chargeback capabilities to tie usage to individual budgets, departments, and workgroups. Data administrators can set guardrails to make sure no individual or workgroup spends more than expected. For example, they can set time-out periods for each type of workload, along with *auto suspend* and *auto resume* features to automatically start and stop resource accounting when the platform isn't processing data. They may also set limits at a granular level, such as determining how long a training model can run before it is terminated.

TIP

Make sure that the pricing model for your cloud data platform matches the value you obtain from it. Paying for a set amount of storage and computing power, commonly known as *subscription-based pricing,* can cause you to incur significant costs and requires regular management. To ensure that you don't pay for more capacity than you need, your cloud data platform should offer usage-based pricing with billing in per-second increments.

Creating User Interfaces

The front-end user interface (UI) of a gen AI application provides users with a way to input data, receive output from the processing engine, and control the application's behavior. Most UIs are based on some form of the following:

- **Web apps** are the most common type of front end for gen AI apps because they're relatively easy to develop and can be accessed from any device with a web browser.
- **Mobile apps** tailored to specific devices, such as tablets and smartphones, offer a more immersive and engaging experience for users. They can take advantage of the unique aspects of each platform and can cache data for offline use.

- **Chat interfaces** are used in gen AI apps when the app needs to converse with the user, such as to answer questions or assist with certain tasks.
- **Desktop apps** are useful for gen AI apps that require a lot of processing power, or that need to access local resources.
- **Command-line interfaces (CLIs)** are sometimes used for gen AI apps that are accessed by developers and data scientists, such as to empower software engineers to generate code.

Simplifying Development and Deployment

A complete data platform includes the necessary primitives for building and deploying gen AI applications without requiring developers to move data or application code to an external system. This accelerates the process of building web apps, chatbots, and other front-end user interfaces. Look for a platform that offers a user-friendly environment for working with Python and other popular coding languages, characterized by the following essential capabilities:

- A **high-performance** environment for interacting with LLMs and processing large volumes of data
- Innate **scalability** to handle an escalating number of users and manage lots of concurrent requests
- Comprehensive **security** so that gen AI apps can safely process sensitive data according to enterprise policies
- **Ease of use,** making gen AI projects accessible to users with limited programming experience by providing pre-built UIs and ways to interact using natural language

Orchestrating AI Agents

LLMs have brought a wide variety of tools, techniques, and frameworks to the task of building AI-powered applications. New developer tools are emerging under the umbrella of LLMOps, short for LLM operations.

Among LLMOps tools, orchestration frameworks can be used to coordinate AI agents and other components to accomplish specific goals. AI agents are simply individual instances of language models that are responsible for performing specific tasks, such as text summarization, language translation, and sentiment analysis. They're coordinated and managed within an orchestration system to complete complex language processing tasks. This process, known as *orchestration*, involves organizing agents, coordinating the input/output of various models, and managing the flow of data and information among agents. For example, in an e-commerce scenario, a chatbot interacting with a customer might use AI agents to retrieve order details from a database, generate a request for a return label using a shipping partner's API, confirm the customer's information, and initiate the return process by sending a shipping label.

REMEMBER

Connecting LLMs to external applications enables them to engage with the wider world, expanding their usefulness beyond language-related tasks. As demonstrated by the e-commerce example, LLMs can initiate actions by interacting with APIs. LLMs can also establish connections with other programming resources, such as a Python interpreter, which allows them to incorporate precise calculations into their outputs. These integrations broaden the capabilities of LLMs and enable them to interact with various external applications, enhancing their overall functionality and versatility (see Figure 4-1).

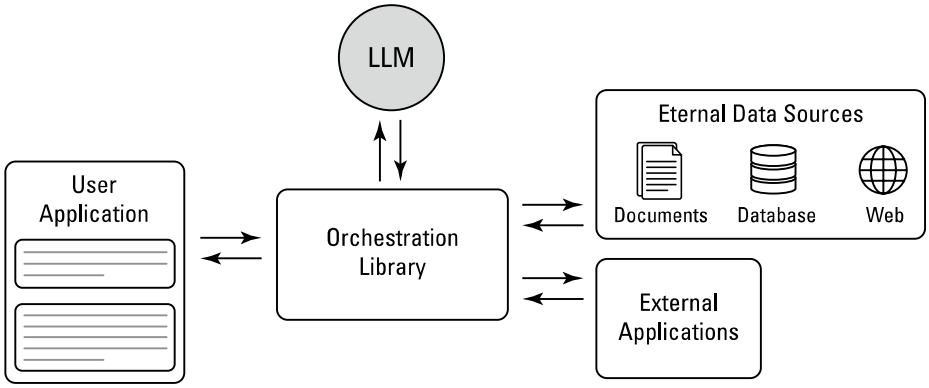

FIGURE 4-1: An orchestration library is used to ensure a seamless flow of data between the user application and the external assets.

All gen AI applications must perform the basic function of passing input to the LLM and returning the results or completions. This is often done through an *orchestration library* that simplifies access to external data sources and connects to APIs within other applications.

Using retrieval-augmented generation (RAG) to connect LLMs to external data sources is an important tactic when you need to update the model with current information, such as breaking news or new research (see Chapter 3 for more on this topic). However, many text sources are too long to fit into the limited context window of the model, which typically holds only a few thousand tokens. Instead, the external data sources are divided into chunks (split), each of which will fit in the context window (and chained together). Packages such as LangChain can handle this work for you.

ORCHESTRATING AI PROJECTS

Orchestration libraries simplify access to external data sources and connect to APIs within other applications in a number of ways, including:

- **Chaining together different prompts:** Allows developers to combine prompts to create more complex applications. This is useful for tasks such as generating creative text formats, answering questions in a comprehensive way, and providing assistance with tasks.
- **Connecting to data sources:** Connect to external data sources, such as databases, APIs, and files. This allows developers to build applications that can access and process information from myriad enterprise sources.
- **Scaling to multiple LLMs:** Use to scale AI applications that have interaction among multiple LLMs. This can become harder to maintain but can also allow developers to use specialized LLMs for different tasks.

IN THIS CHAPTER

- Mitigating data privacy concerns
- Alleviating biases and prejudices
- Fostering ethical training and deployment practices
- Taking responsibility for data privacy and security
- Respecting copyright laws

Chapter 5
Reviewing Security and Ethical Considerations

Although large language models (LLMs) can process text and create content on just about any subject, it is crucial for businesses to consider issues of intellectual property, data privacy, and potential content misuse. Generative AI (gen AI) applications are trained on massive data sets of text and code, which may include sensitive or legally protected information. If this data isn't properly protected, it could be leaked to third parties or accessed by unauthorized individuals.

This chapter discusses the ethical implications of using LLMs and also addresses a few practical concerns.

Reiterating the Importance of Security and Governance

Gen AI is a powerful technology with the potential to revolutionize many enterprise business functions. Enterprises must pay attention to the data privacy risks associated with this technology and take steps to mitigate these risks:

1. **Choose software vendors that have a proven record along with third-party certifications of data privacy and security.**

   Carefully review the vendor's terms of service and privacy policy to understand how your data will be used.

2. **Appoint a data steward — ideally a business owner who understands the data — to take charge of each data set.**

   Establish consistent procedures for data security, data privacy, and data governance to satisfy industry regulations and avoid compliance violations.

3. **During development and production, continually monitor and audit gen AI apps to identify and mitigate any potential risks.**

   This may include monitoring the outputs of these applications for sensitive information and regularly reviewing the training data to ensure that it is relevant and up to date.

TIP

Take advantage of a cloud data platform that allows you to break down silos and extend consistent governance and security to disparate sources — internally from your enterprise applications and externally from business partners and third-party data providers. The platform should support popular programming languages, scalable GPU infrastructure, and provide flexibility to use open-source tools to maximize options for your team.

MITIGATING DATA PRIVACY CONCERNS

Pay attention to these data privacy concerns when developing, deploying, and using gen AI applications:

- **Unintentional disclosure of sensitive information:** Gen AI apps can sometimes generate outputs that contain sensitive customer information, even if the prompts or inputs don't explicitly mention this information. For example, a gen AI application used to generate marketing copy could generate text that contains customer names and addresses, even if the prompt only specifies the product or service being advertised.
- **Misuse of generated data:** Gen AI applications can be used to generate synthetic data that's indistinguishable from real data. This synthetic data could then be used for malicious purposes, such as identity theft or fraud. It could also be used to create deep fakes or other forms of disinformation.
- **Compliance violations:** Data privacy regulations such as the Global Data Protection Act (GDPR) and California Consumer Protection Act (CCPA) impose strict requirements on how businesses can collect, use, and store personal data. These same regulations apply to data used in the training of gen AI models.

Centralizing Data Governance

One of the primary reasons to use a cloud data platform for your gen AI initiatives is because it allows you to centralize data privacy and protect sensitive customer information. Enterprises should implement appropriate data privacy and security measures, which may include measures such as data encryption, access control, intrusion detection, and comprehensive data governance.

Data governance entails knowing precisely what data you have, where it resides, who is authorized to access it, and how each type of user is permitted to use it. Instituting comprehensive controls reduces the risk of compliance violations. All data governance strategies should seek to protect sensitive data as it is accessed, shared, and exchanged across the organization and beyond. All of this should apply not only when data is stored but also when processed by a model or surfaced in an application.

Alleviating Biases

One important ethical consideration involves being alert to the inherent model biases that may be present in the training data, which may cause LLMs to generate outputs that are discriminatory or unfair. For example, if a historical data set contains biases against certain demographics, such as race or gender, an LLM trained on this data may inadvertently perpetuate those biases. If a marketing team asks an LLM to generate content for a customer of a specific gender, it's important to keep in mind what kind of bias the model may have as it creates that content, even if there is no explicit intent to discriminate.

TIP

AI enthusiasts commonly cite the three Hs when discussing the responsible deployment of AI: helpfulness, honesty, and harmlessness.

Acknowledging Open-Source Risks

Open source LLMs such as Llama 2, BERT, and Falcon offer tremendous capabilities at little or no cost to users, but they can come with risks that are part of the model's training data set, which is often not publicly accessible.

Other open-source tools that can be used to build LLM apps, such as an orchestration framework, a vector database, and so on, may be vulnerable to risks if not regularly updated and patched. Without proper security measures and consistent maintenance practices, malicious entities can sometimes exploit these vulnerabilities.

In addition to these security concerns, open-source offerings may exhibit inconsistent quality due to the basic nature of their development: They're the byproduct of many community contributions. Consider the cost, performance, and compliance risks of using any LLM. The choice between open source and proprietary LLMs depends on your organization's specific needs, technical resources, and risk tolerance (see Chapter 2 for more).

Contending with Hallucinations

As demonstrated throughout this book, LLMs have an uncanny capability to engage in dialogue, answer questions, provide explanations, generate creative text, and assist with various language-related tasks. However, it's important to note that while LLMs often exhibit impressive capabilities, they may occasionally produce incorrect or nonsensical responses. They are also known to *hallucinate*, meaning that they may generate content that is fictional or erroneous.

TIP

Mitigating hallucinations involves implementing the strategies discussed in Chapters 3 and 4: fine-tuning the model using reliable and accurate data, incorporating human review and oversight, and continuously monitoring and refining gen AI systems to minimize the occurrence of false or misleading information.

ENFORCING ETHICAL PRACTICES

When developing and training gen AI models, follow these three principles:

- **Bias mitigation:** LLMs can reflect and reinforce societal biases present in the data on which they're trained. Ethical considerations involve identifying and mitigating biases to ensure fair and equitable outcomes. Developers and users should actively work to minimize biased results and ensure models that are inclusive and representative.
- **Responsible use:** Establish guidelines and guardrails to prevent misuse or harmful applications of LLMs. This includes setting boundaries and restrictions on the use of LLMs to avoid the spread of misinformation, hate speech, or other forms of harmful content.
- **Societal impact:** LLMs have the potential to influence public discourse and shape attitudes. Ethical considerations involve understanding the broader societal impact of using LLMs and considering the potential consequences for various stakeholders.

Observing Copyright Laws

In September 2023, John Grisham, Jodi Picoult, Jonathan Franzen, George R.R. Martin, and 13 other authors joined the Authors Guild in filing a class action suit against OpenAI, alleging that the company's GPT technology is illegally using the writers' copyrighted works. The complaint called the usage of these works by LLMs a "flagrant and harmful" copyright infringement, claiming that their books were misused in the training of its artificial intelligences.

This suit may have far-reaching consequences for OpenAI and other LLM vendors, depending on how the litigation progresses. Comedians, writers, musicians, movie studios, and many other content creators have filed similar lawsuits alleging that their original works are copyright-protected and may not be freely used to train LLMs without permission. "Authors should have the right to decide when their works are used to train AI," stated Jonathan Franzen in a September 20, 2023, press release issued by the Authors Guild. "If they choose to opt in, they should be appropriately compensated."

REMEMBER

These types of cases highlight the importance of respecting copyrighted content that may have been used to train foundation models. Legal and regulatory frameworks, including litigation outcomes, will help the AI industry establish clear guidelines and reinforce important ethical norms to avoid further legal actions in the future. In the meantime, enterprises should be aware of the implications of the applications they create and the content they use in all gen AI endeavors.

Chapter 6
Five Steps to Generative AI

Following the steps in this chapter will help ensure you reap positive new levels of productivity.

Identify Business Problems

Rank potential projects based on expected business impact, data readiness, and level of executive sponsorship. Research and evaluate pretrained language models, minimize complexity of infrastructure maintenance, and consider solutions that empower large numbers of users to derive value from data.

Select a Data Platform

How do you make sure your data is secured and governed from the time it's used to fine-tune until it is presented through the app UI? How easy is it to allocate and scale GPUs? Standardize on a cloud data platform that offers these benefits:

- Scalable, pay-as-you-go infrastructure to handle the storage and computational requirements
- Near-zero maintenance, there's no need to perform platform updates or other administrative tasks

- Access large language model (LLM) app stack primitives that help teams build custom solutions without integrations of multiple platforms
- Capability for those without AI expertise to bring gen AI to their daily workflows with UI-driven experiences
- Access to structured/semistructured/unstructured data, both internal and from third parties, via a marketplace
- Native support for popular AI frameworks, tools, and programming languages

Build a Data Foundation

Consolidate your data to remove silos, create data pipelines, and make sure that all data is consistently cleansed. Establish consistent procedures for data privacy and data governance to satisfy industry regulations. Extend those procedures to data and apps from third-party providers. Lastly, minimize data exfiltration into compute environments that don't apply consistent security and governance policies of the data.

Create a Culture of Collaboration

How do you enable data scientists, analysts, developers, and business users to access the same data sets simultaneously, without having to copy or move the data? Make sure your data platform empowers all pertinent stakeholders to easily collaborate as they share data, models, and applications. Educate business users on prompt engineering; and other ways to leverage models without customizations that require deeper AI expertise.

Measure, Learn, Celebrate

How do you gauge the success of your gen AI initiatives? Start small, experiment, identify metrics to demonstrate business results, and validate progress with executive sponsors and stakeholders. Share best practices and encourage reusability. Strive to democratize gen AI capabilities throughout your entire organization.

An introductory overview to LLMs and gen AI applications

Generative AI (gen AI) and large language models (LLMs) are revolutionizing personal and professional lives. From supercharged digital assistants that manage email to seemingly omniscient chatbots that can communicate with enterprise data across industries, languages, and specialties, these technologies are driving a new era of convenience, productivity, and connectivity. This book provides an introductory overview to LLMs and gen AI applications, along with techniques for training, tuning, and deploying machine learning (ML) models.

Inside...

- Five steps to get started with generative AI
- Understand important LLM concepts
- Develop applications with user-friendly interfaces
- Select an appropriate LLM
- Recognize the importance of data governance
- See how to orchestrate AI agents

David Baum is is a freelance business writer specializing in science and technology.

Go to Dummies.com™
for videos, step-by-step photos, how-to articles, or to shop!

ISBN: 978-1-394-23842-2
Not For Resale

for dummies®
A Wiley Brand

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.