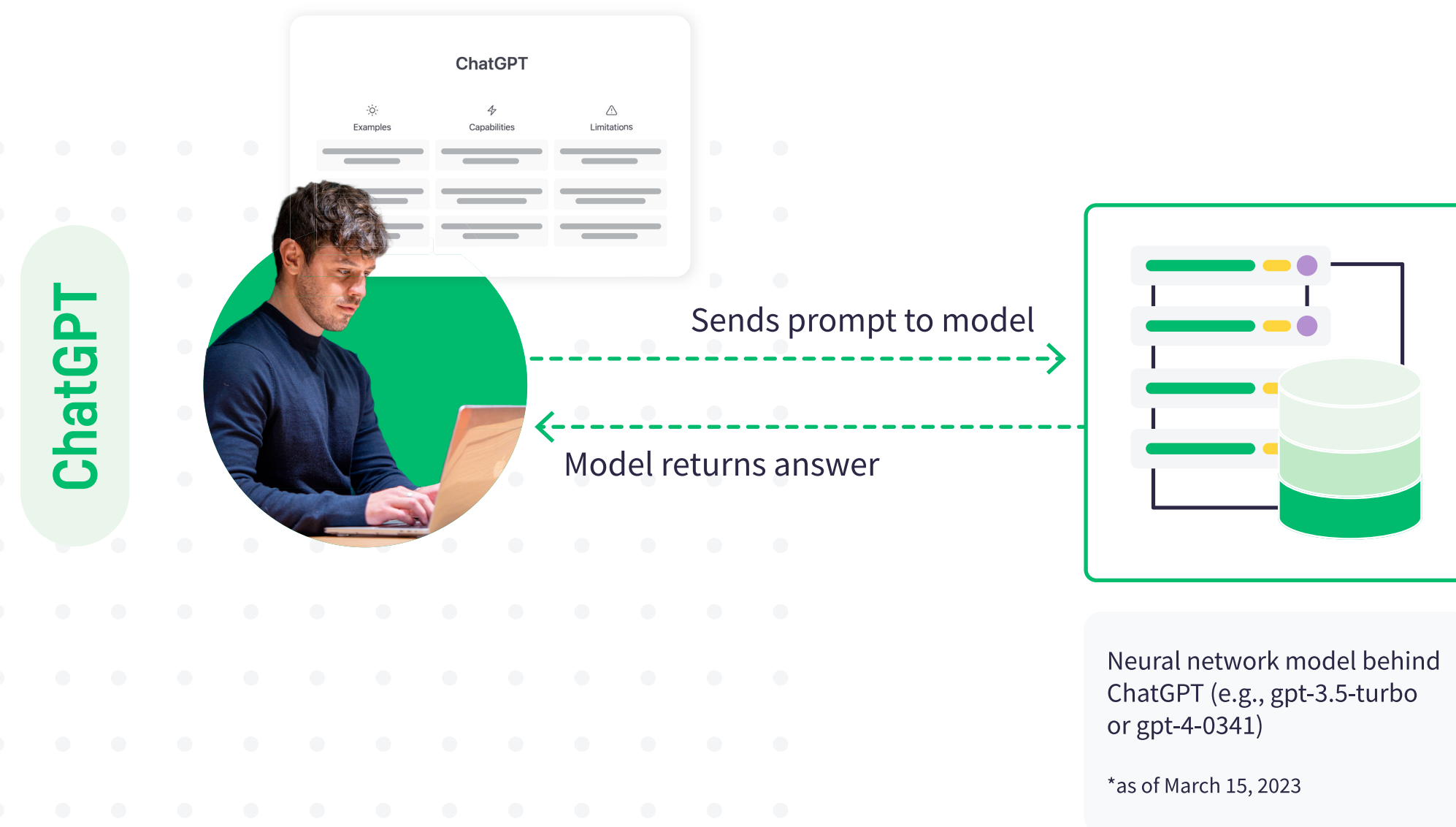data iku

Using ChatGPT, GPT-4, & Large Language Models in the Enterprise
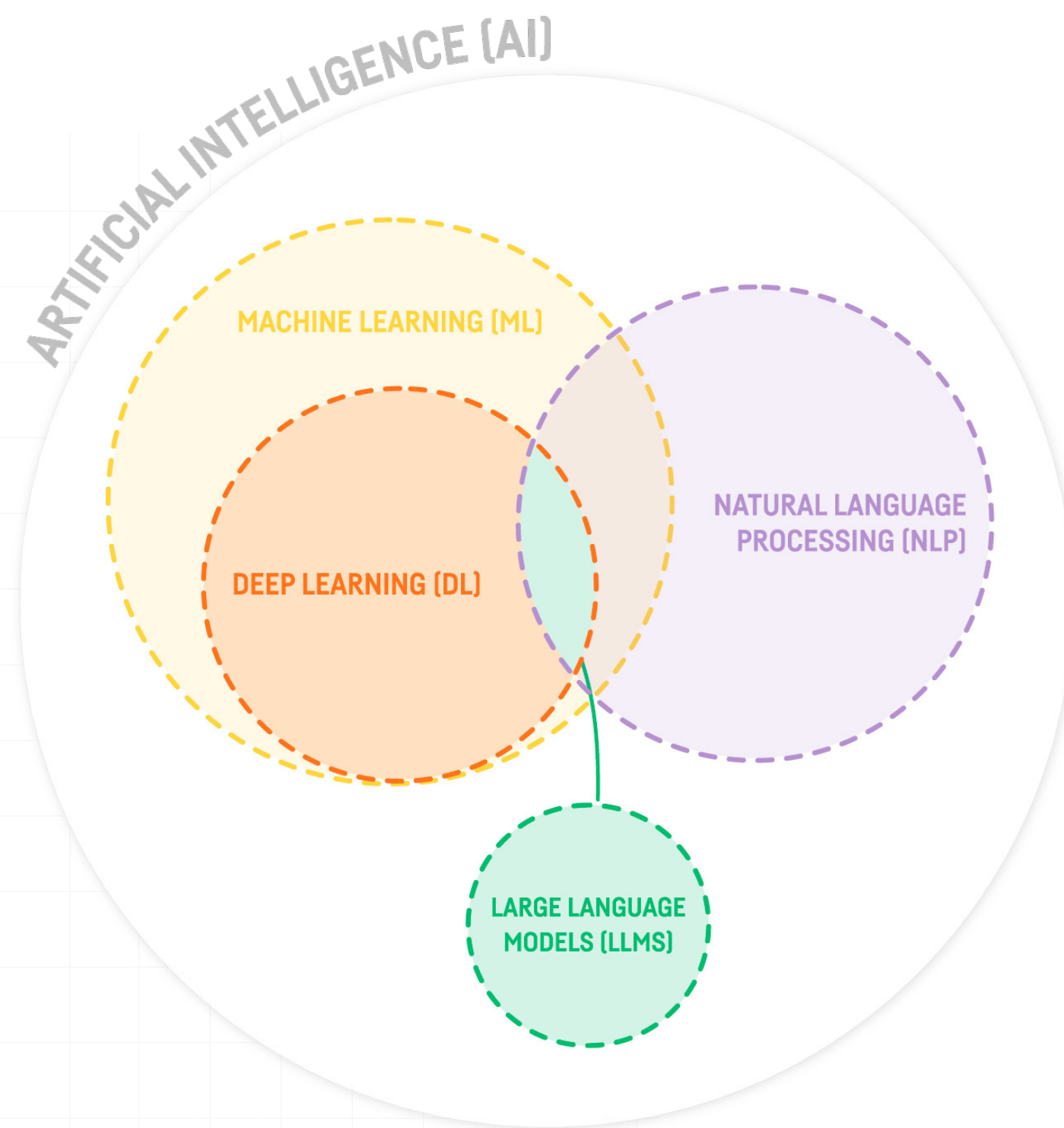
What Are ChatGPT, Bing, & Bard?

While OpenAI's ChatGPT, Microsoft's Bing, and Google's Bard have received a lot of public attention in the past months, it is important to remember that they are specific products built on top of a class of technologies called Large Language Models (LLMs).

The most accessible way of using these products requires typing or pasting text into their respective input boxes and reading or copying their responses. Only some of these products offer programmatic interfaces, meaning they cannot be "plugged into" other enterprise data systems or processes. This enterprise-level use of this technology will be the focus of this document.

This document was published on March 15, 2023. We are working on updating it as often as possible to keep up with the pace of change in this space, but note it might take a few days for us to adjust the content to new announcements and developments.

ChatGPT

ChatGPT

☀ Examples ⚡ Capabilities ⚠ Limitations

Sends prompt to model →

← Model returns answer

Neural network model behind ChatGPT (e.g., gpt-3.5-turbo or gpt-4-0341)

*as of March 15, 2023

What Is a Large Language Model (LLM)?

ARTIFICIAL INTELLIGENCE (AI)

MACHINE LEARNING (ML)

DEEP LEARNING (DL)

NATURAL LANGUAGE PROCESSING (NLP)

LARGE LANGUAGE MODELS (LLMS)

An LLM is a neural network model architecture based on a specific component called a transformer. Transformer technologies were originally developed by Google in 2017[1] and have been the subject of intense research and development since then. LLMs work by reviewing enormous volumes of text, identifying the ways that words relate to one another, and building a model that allows them to reproduce similar text.[2]

Importantly, when asked a question, they are not "looking up" a response. Rather, they are producing a string of words by predicting which word would best follow the previous, taking into account the broader context of the words before it. In essence, they are providing a "common sense" response to a question.
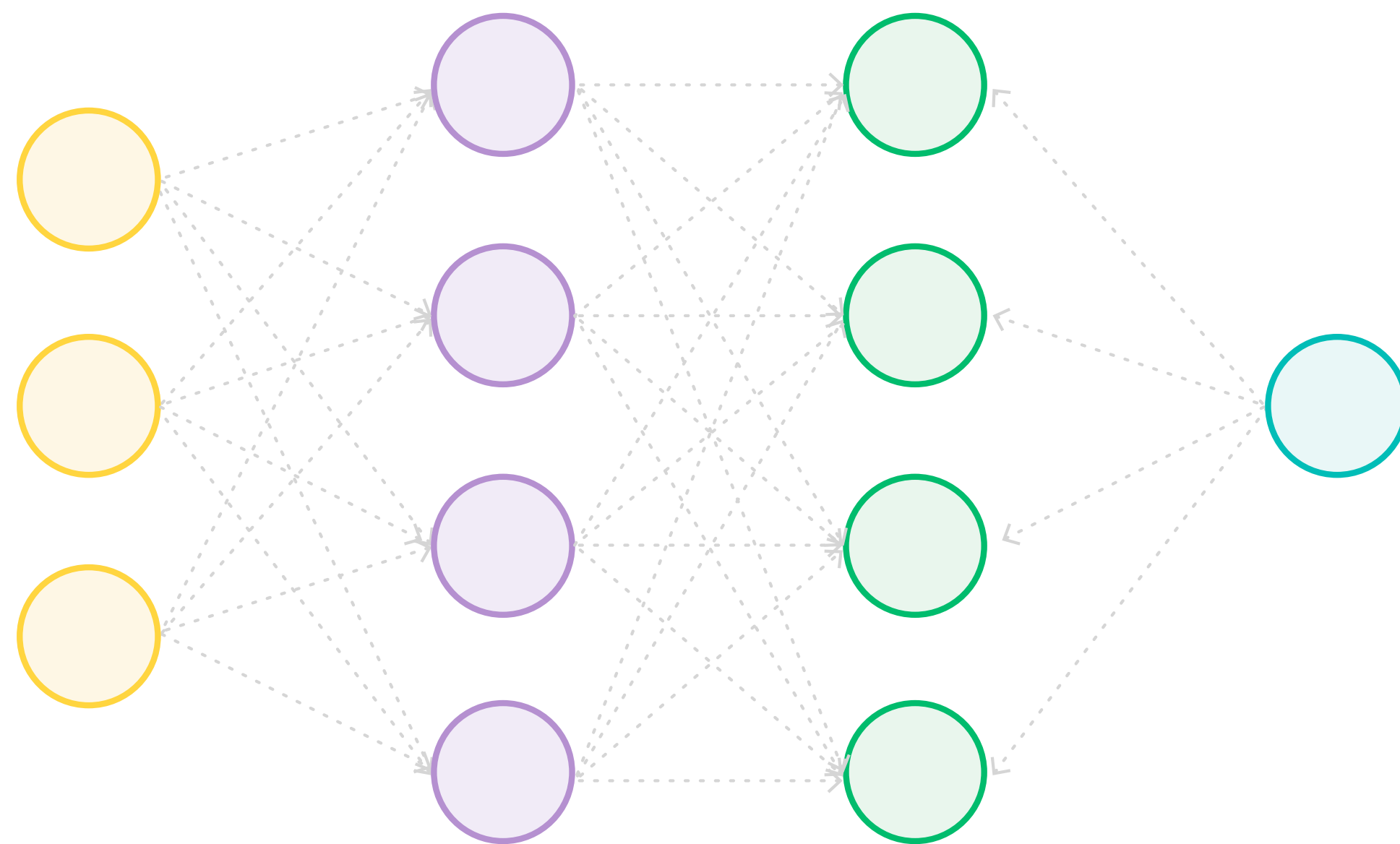
While the most powerful LLMs have shown their ability to produce largely accurate responses on an astonishingly wide range of tasks, the factual accuracy of those responses cannot be guaranteed.[3]

——————

1   https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

2   Stephen Wolfram has written a detailed explanation of LLMs that is accessible to a technical but non-expert reader: https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/

3   https://openai.com/blog/chatgpt

What Makes a Large Language Model… Large?

A neural network is made up of a large number of "neurons," which are simple mathematical formulas that pass the results of their calculations to one or more neurons in the system. The connections between these neurons are given "weights" that define the strength of the signal between the neurons. These weights are also sometimes called parameters.

One of the models behind ChatGPT (gpt-3.5-turbo) has 175 billion parameters.[4] GPT-4 has an unknown number of parameters.

The size of these models has important consequences for their performance, but also the cost and complexity of their use. On the one hand, larger models tend to produce more human-like text and are able to handle topics that they may not have been specifically prepared for. On the other hand, both building the model and using the model is extremely computationally intensive.

It is no accident that the largest and most highly performing models have come from giant technology companies or startups funded by such companies: The development of these models likely costs billions of dollars in cloud computing.

_____

4   _https://arxiv.org/pdf/2005.14165.pdf_

Tradeoffs Between LLMs & Other Methods

Using LLMs can be computationally intensive. In the case of a model with 175 billion weights, for each "token" (that's a word or a piece of a word) that it outputs, it needs to run 175 billion calculations… each and every time.[5] Why make such a large model then? Especially when a smaller language model is very good at the task for which it is designed?

The largest models are like a smartphone: They are convenient in that they pack a lot of functionality into a single product. This means that you can use the same model for a variety of tasks. It can translate, it can summarize, it can generate text based on a few inputs. Like a smartphone, it's the one solution that you need to handle a vast array of tasks, though this flexibility comes at a price.

If all you need is a stopwatch, you can find a far less expensive option than a smartphone. Similarly, if you need a solution to a specific task, you may be better off selecting a small, task-specific model instead.

_____

5   https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/

Identifying a Use Case for LLMs

If you're interested in testing the usefulness of an LLM inside your organization, seek an application that balances the following:

Risk Tolerance

If this is the first time you're using this technology, choose a domain where there is a certain tolerance for risk. The application should not be one that is critical to the organization's operations and should, instead, seek to provide a convenience or efficiency gain to its teams.

Human Review

A law firm has been quoted as saying that they are comfortable using this technology to create a first draft of a contract, in the same way they would be comfortable delegating such a task to a junior associate. This is because any such document will go through many rounds of review thereafter, minimizing the risk that any error could slip through.

Text (or Code) Intensive

It's important to lean on the strengths of these models and set them to work on text-intensive or code-intensive tasks — in particular those that are "unbounded," such as generating sentences or paragraphs of text. This is in contrast to "bounded" tasks, such as sentiment analysis, where existing, purpose-built tools will provide excellent results at lower cost and complexity.
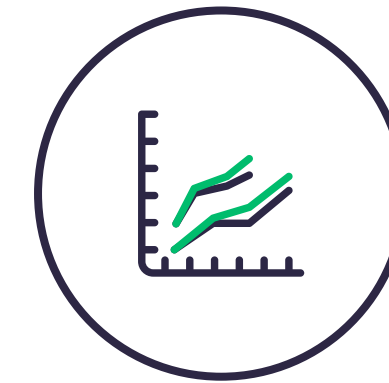
Business Value

As always, and perhaps especially when there is a lot of excitement around a new technology, it is important to come back to basics and ask whether the application is actually valuable to the business. LLMs can do many things, whether those things are valuable or not is a separate question.

How to Use LLMs in the Enterprise

Using an LLM in the enterprise, beyond the simple web interface provided by products like ChatGPT, can be done in one of two ways:

1 Making an API call to a model provided as a service, such as the GPT-3 models provided by OpenAI, including the gpt-3.5-turbo or gpt-4-0314 models that powers ChatGPT. Generally, these services are provided by specialized companies like OpenAI, or large cloud computing companies like Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure.

2 Downloading and running an open-source model in an environment that you manage. Platforms like Hugging Face aggregate a wide range of such models.

Each approach has advantages and drawbacks, which we'll explore in the following sections. And remember, both options allow you to choose from smaller and larger models, with tradeoffs in terms of the breadth of their potential applications, the sophistication of the language generated, and the cost and complexity of using the model.
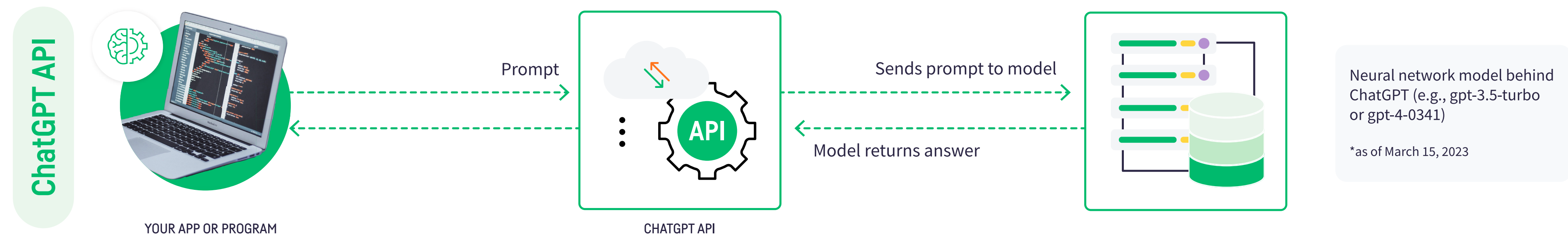
Advantages of Using a Model-as-a-Service via API

Some companies (such as OpenAI, AWS, and GCP) provide public APIs that can be called programmatically. This means setting up a small software program, or a script, that will connect to the API and send a properly formatted request.

The API will submit the request to the model, which will then provide the response back to the API, which will in turn send it back to the original requester.

There are several advantages to this approach:

1. **Low barrier to entry:** Calling an API is a simple task that can be done by a junior developer in a matter of minutes.

2. **More sophisticated models:** The models behind the API are often the largest and most sophisticated versions available. This means that they can provide more sophisticated and accurate responses on a wider range of topics than smaller, simpler models.

3. **Fast responses:** Generally, these models can provide relatively quick responses (on the order of seconds) allowing for real-time use.

ChatGPT API

Prompt

Sends prompt to model

API

Model returns answer

YOUR APP OR PROGRAM

CHATGPT API

Neural network model behind ChatGPT (e.g., gpt-3.5-turbo or gpt-4-0341)

*as of March 15, 2023

Limitations to Using a Model-as-a-Service via API

The use of public models via API, while convenient and powerful, has limitations that may make it inappropriate for certain enterprise applications:

1. **Data residency and privacy:** By nature, public APIs require that the content of the query be sent to the servers of the API service. In some cases, the content of the query may be retained and used for further development of the model. Enterprises should be careful to check if this architecture respects their data residency and privacy obligations for a given use case.

2. **Potentially higher cost:** Most of the public APIs are paid services, whereby the user is charged based on the number of queries and the quantity of text submitted. The companies providing these services usually provide tools to estimate the cost of their use. They also often provided smaller and cheaper models which may be appropriate for a narrower task.

3. **Dependency:** The provider of an API can choose to stop the service at any time, though such a decision is typically rare for a popular (and money-making) service. Smaller companies providing such services may be at risk of insolvency. Enterprises should weigh the risk of building a dependency on such a service and ensure that they are comfortable with it.

Advantages of Self Managing an Open-Source Model

Given the drawbacks of using a public model via API, it may be appropriate for a company to set up and run an open-source model themselves. Such a model can be run on on-premises servers that an enterprise owns or in a cloud environment that the enterprise manages.

Advantages of this approach include:

1. **Wide range of choice:** There are many open-source models available, each of which presents its own strengths and weaknesses. Companies can choose the model that best suits their needs. That said, doing so requires some familiarity with the technology and how to interpret those tradeoffs

2. **Potentially lower cost:** In some cases, running a smaller model that is more limited in its application may allow for the desired performance for a specific use case at much lower cost than using a very large model provided as a service.

3. **Independence:** By running and maintaining open-source models themselves, organizations are not dependent on a third-party API service.

Tradeoffs to Self Managing an Open-Source Model

While there are many advantages to using an open-source model, it may not be the appropriate choice for every organization or every use case for the following reasons:

1. **Complexity:** Setting up and maintaining a LLM requires a certain degree of data science and engineering expertise — beyond that required for simpler machine learning models. Organizations should evaluate if they have sufficient expertise, and if those experts have the necessary time to set up and maintain the model in the long run

2. **Narrower performance:** The very large models provided via public APIs are astonishing in the breadth of topics that they can cover. Models provided by the open-source community are generally smaller and more focused in their application, though this may change as ever-larger models are built by the open source community.
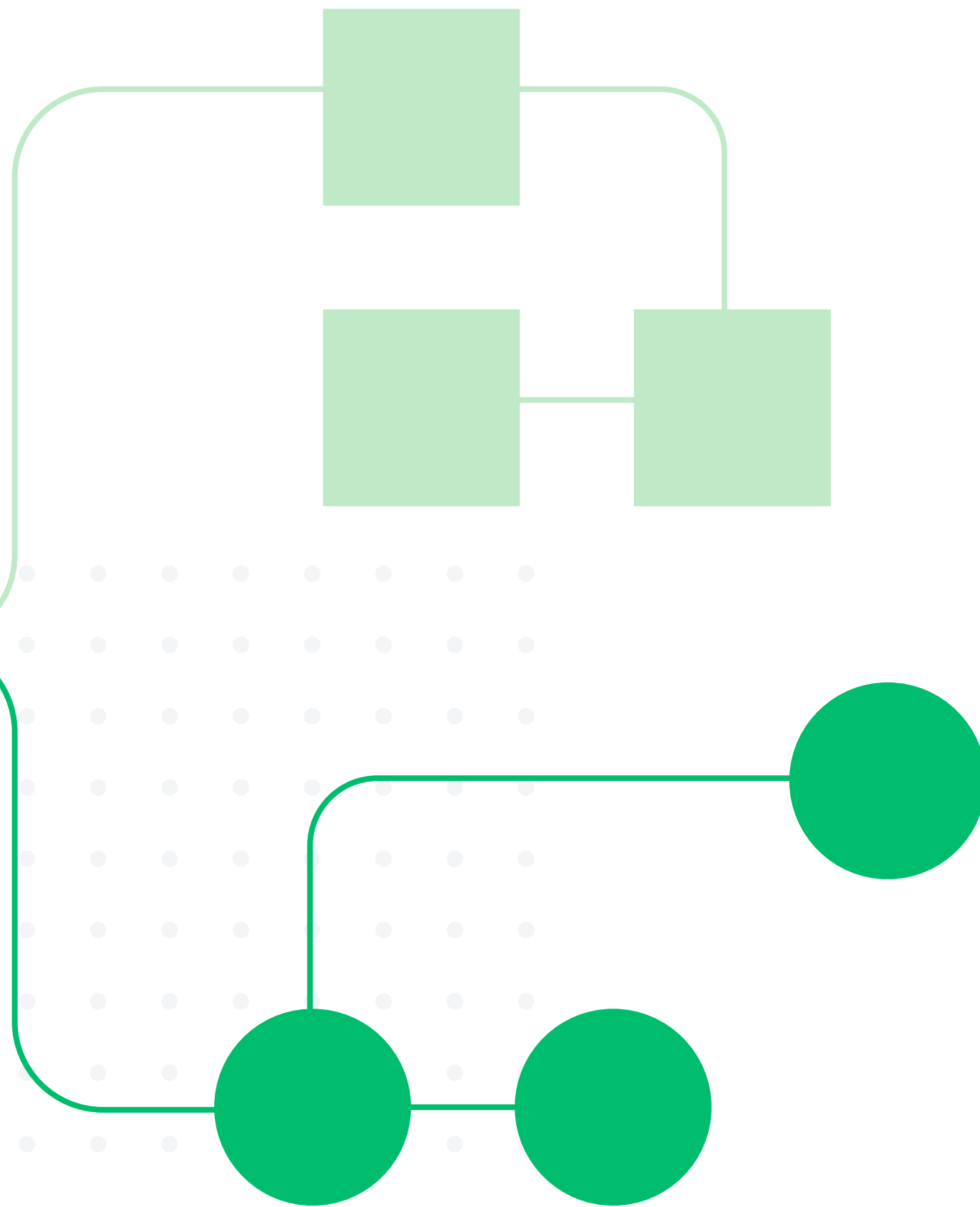
Choosing an Approach

Given the tradeoffs between the different approaches, how can an organization choose the one that is right for them?

In fact, there is no single approach that will be appropriate enterprise wide. The best strategy for most organizations will be to equip themselves with the means to choose the best model and architecture for a given application or use case.

In certain cases, the balance may tip toward using the model-as-a-service APIs, in others, the use of an open-source model may be more appropriate. In some cases, a very large model may be required — in others, a smaller model may suffice.

The companies that are most successful in using LLMs will be those that equip themselves with the ability to choose and apply the right approach and the right model for a given application, especially given the rapid pace of innovation in this space.
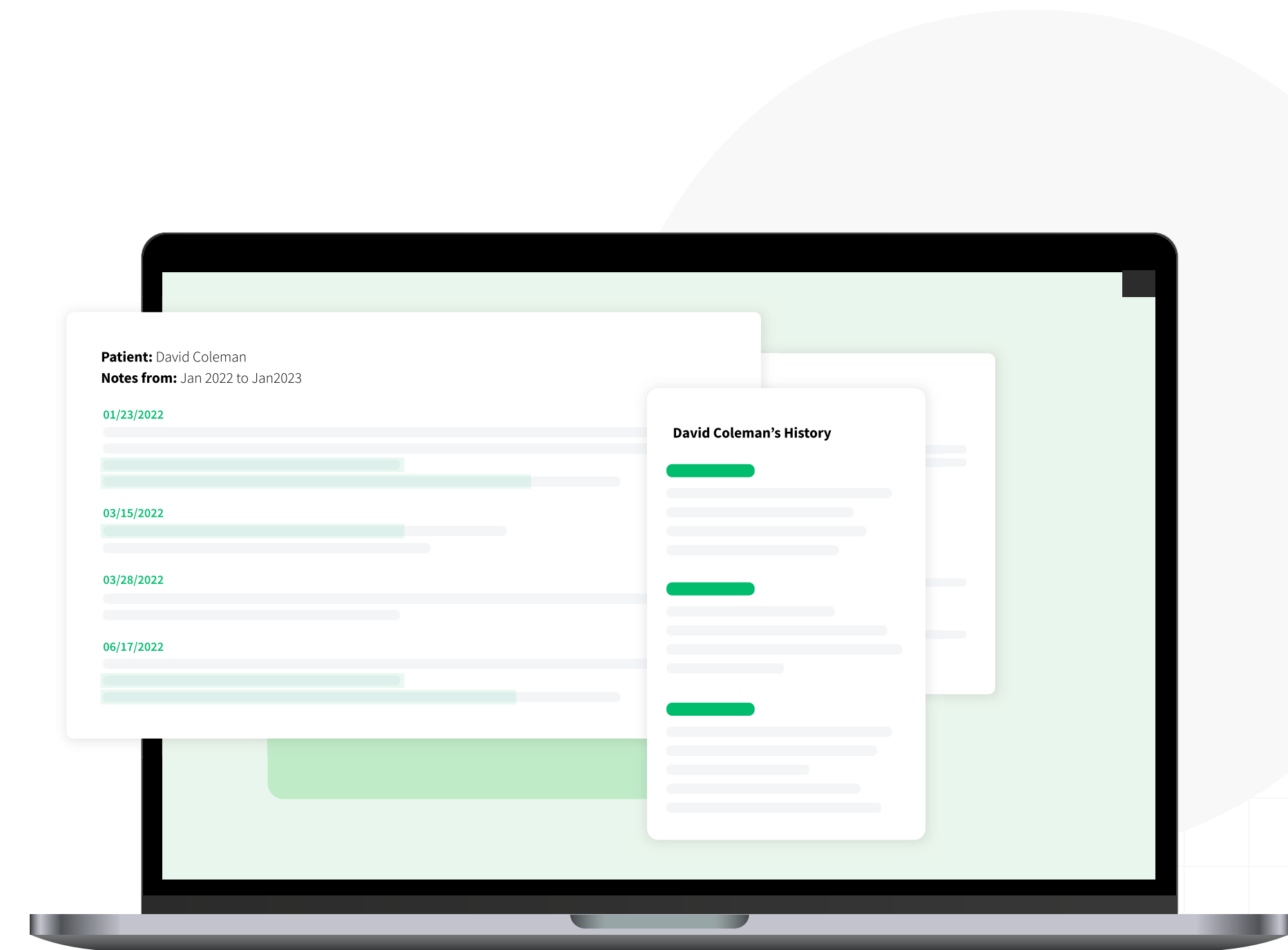
Use Case Example:
Summarizing Physicians' Notes

A Dataiku partner has developed a demonstration project that aggregates and summarizes physicians notes within an electronic medical record system.

Using the open source BART model[6] that has been trained on medical literature, the system is able to automate the drudgery of summarizing the copious physician's notes into short patient histories. Importantly, the detailed notes remain in the system, allowing a consulting physician to get the full detail when needed.

Before use in any production context, the a selection of the summaries will be reviewed by a human expert to ensure that they are functioning as expected and capturing the most relevant information.

_____

6   _https://huggingface.co/mse30/bart-base-finetuned-pubmed_

Use Case Example: Documentation Search

As an internal test of the technology, **Dataiku has used OpenAI's GPT-3 model** to provide full text responses on the full contents of Dataiku's extensive public documentation, knowledge base, and community posts.

This was an appropriate use case for leveraging a third-party model as all of the data is public already, and the large third-party model will be well-suited to responding to the many different ways that users may phrase their questions.

The results have been impressive, providing easy-to-understand summaries and often offering helpful context to the response. Users have reported that it is more effective than simple links back to the highly technical reference documentation.

Dataiku Chat

SEARCH
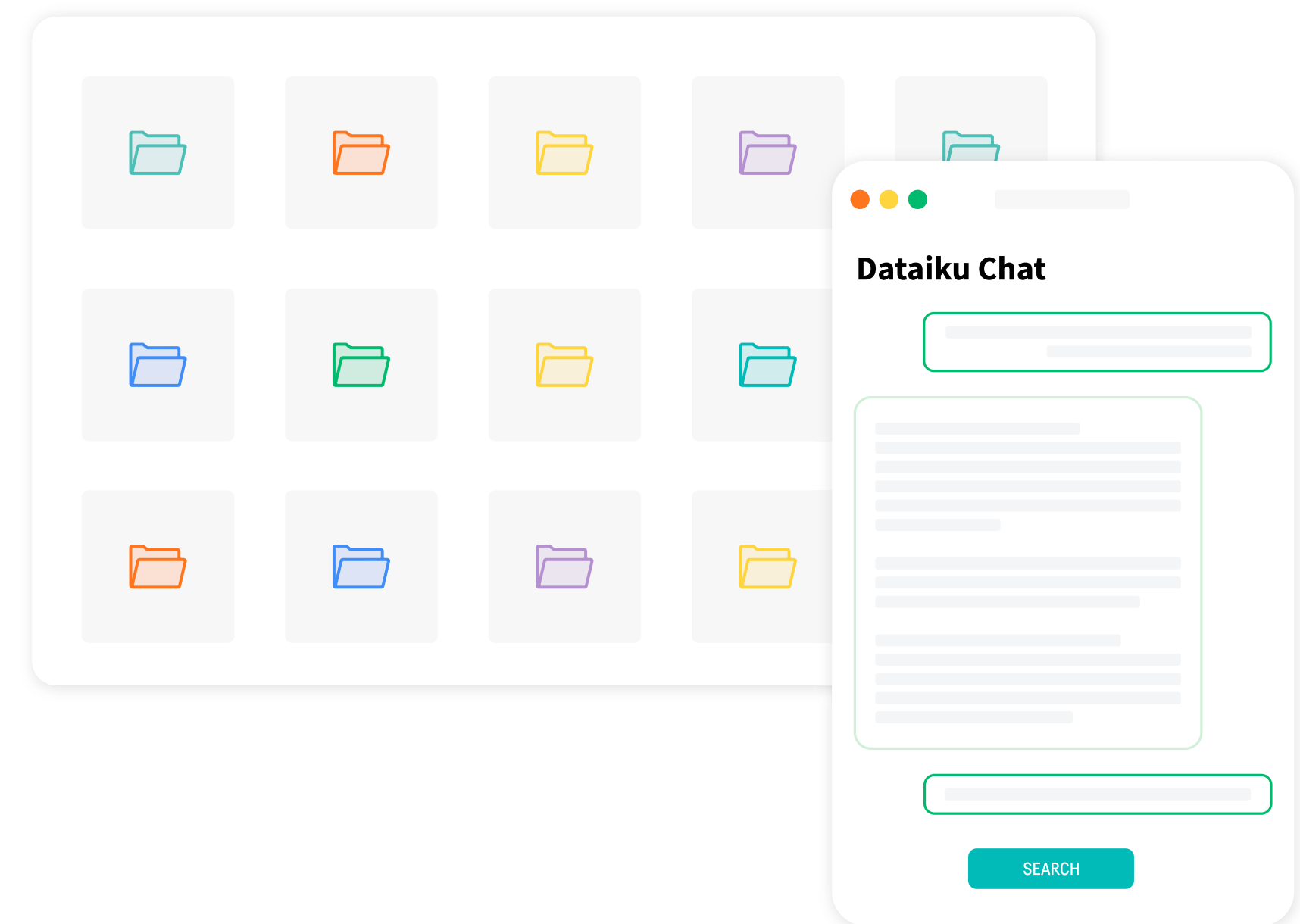
How to Use LLMs Responsibly
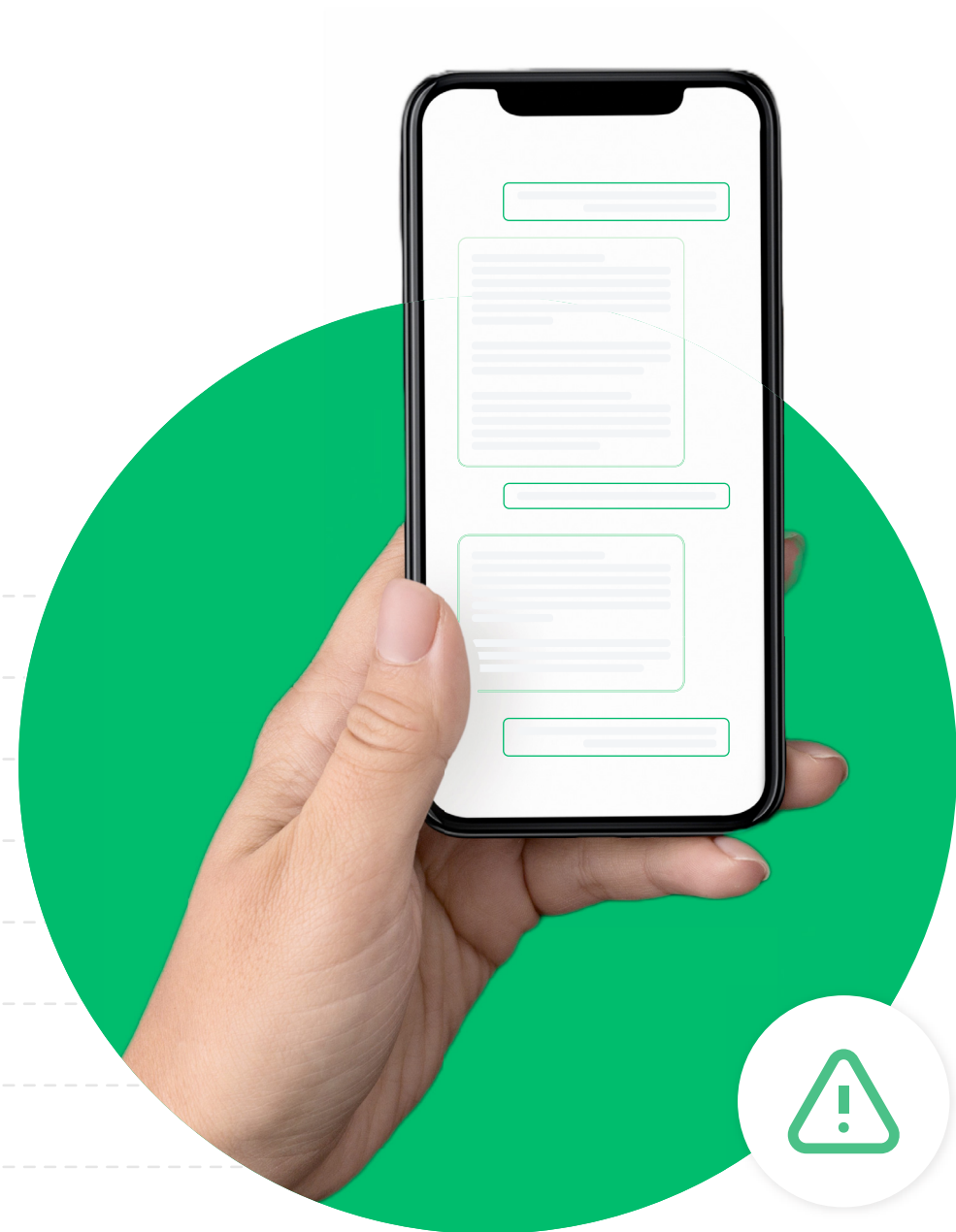
Using LLMs responsibly requires similar steps and considerations as the use of other machine learning and AI technologies:

- **Understand how the model is built.** All machine learning and neural networks, including those used in LLMs, are influenced by the data that they are trained on. This can introduce biases that need to be corrected for.

- **Understand how the model will impact the end user.** LLMs in particular present the risk that an end user may believe that they are interacting with a human when, in fact, they are not. Dataiku recommends that organizations disclose to end users where and how these technologies are being used. Organizations should also provide guidance to end users on how to interact with any information derived from the model and provide caveats for the overall quality and factual accuracy of the information. With this information, the end user is well-equipped to decide for themselves how best to interpret the information.

- **Establish a process for approving which applications are appropriate uses for each technology.** The decision of whether or not to use a particular model for a given application should not be made by individual data scientists. Instead, organizations should set up principles for which technologies should — and should not — be used in what contexts, with a consistent review process by leaders who are accountable for the outcomes.

- **Keep track of which technology is being used for which application.** AI governance principles are meant to both prevent problems from arising and to ensure that there is an auditable history of which technology has been used in the event that a problem occurs and needs to be reviewed. Tools, **including AI platforms like Dataiku**, can help serve as that central control tower
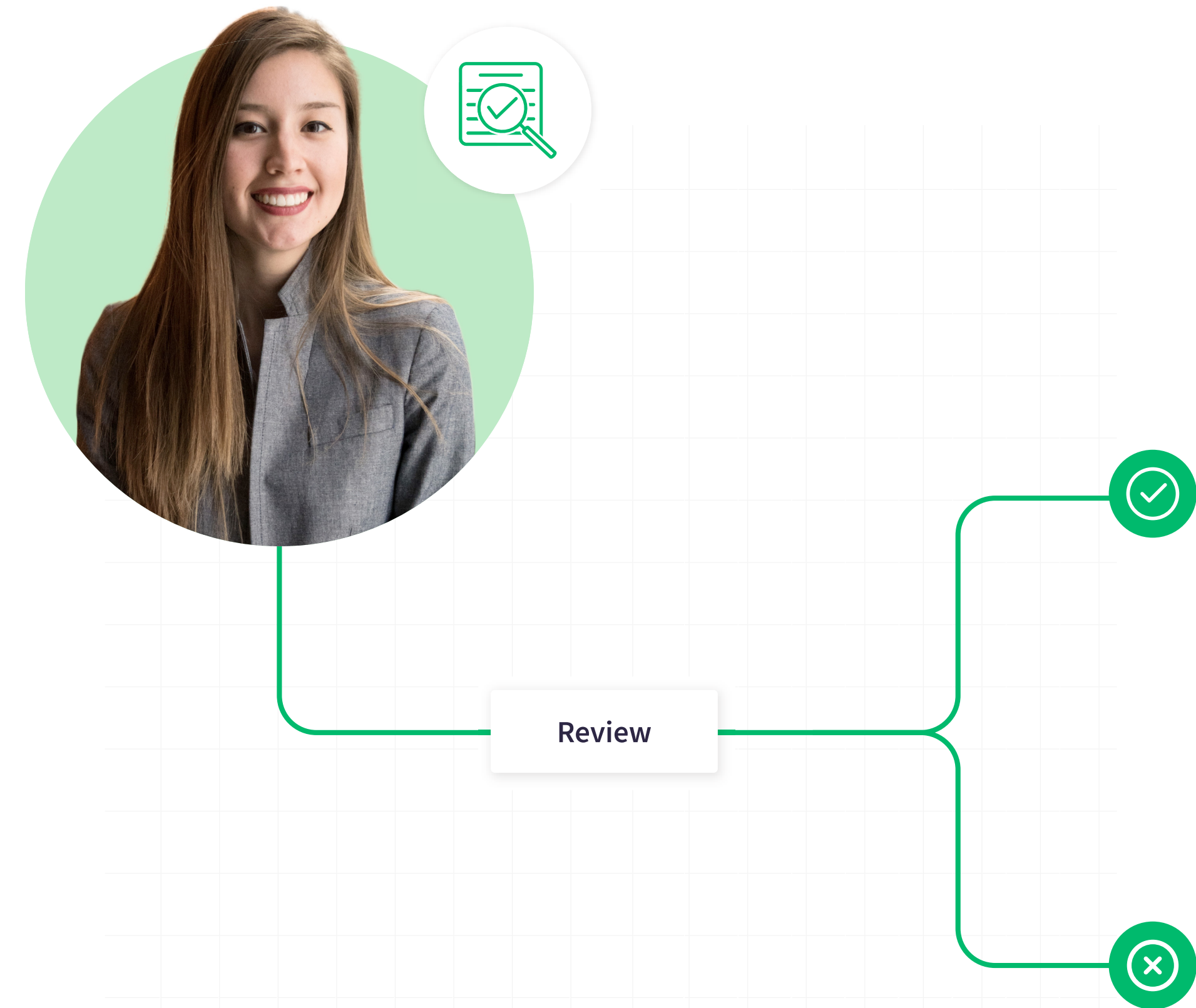
Review

Using LLMs in Dataiku

Dataiku supports the two approaches for using LLMs previously described in this document while also providing the broader framework necessary for using them, including data connectors, data preparation capabilities, automation, and governance features.

Dataiku's core vision has been to allow enterprises to quickly integrate the latest machine learning models and neural networks into their operations, and that continues with LLMs.

Using Public Models via API in Dataiku

Dataiku provides a **direct interface** with the popular OpenAI text completion APIs, including the powerful GPT-3 family of models. Using this integration is simple:
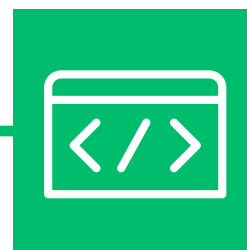
1. Prepare your input dataset. The Dataiku integration supports use cases where you provide both instructions and input text ("change the following text from first-person to third-person," "I went for a long walk one day…") and where only instructions are provided ("write a 50-word email addressed to Mr. Smith"). Like all data preparation steps in Dataiku, this may be done using the visual data preparation tools or using the programming language of your choice.

2. Configure your integration with the **OpenAI text completion** API, adding the API keys obtained from OpenAI when **signing up for their service.**

3. Select the model that you want to call, referring to the OpenAI documentation to choose the right one for your particular task, as well as the additional parameters offered by the OpenAI API.

4. Specify the output dataset where the completed text will be written to.

In order to use other models via API, the **API call** can be accomplished using a Python or R recipe for Steps 2-4 above.

Using an Open-Source LLM in Dataiku

One of Dataiku's core strengths and design principles is extensibility. If a particular feature or function is not natively available in Dataiku, the platform allows users to build that integration themselves.

This is the case for open-source LLMs, as it is for other machine learning models not natively available in Dataiku. A high-level overview of the process is as follows, which would typically be done in several Python recipes:

1. Download and **install a Python package**, such as the Hugging Face **transformers package**. [7]

2. Download the pretrained model.

3. If you have relevant label data and out-of-the-box performance is not satisfactory, you can consider retraining or fine tuning the model.

4. Perform the text generation step (called inference), and write the results back to the desired data store

5. (Optional)The model could then be packaged into a visual plugin, allowing non-coding users to apply the model to their tasks.

In addition, Dataiku provides additional Natural Language Processing (NLP) capabilities that are built on LLMs to assist in building machine learning models using natural language text as an input. For example, **sentence embedding** allows text variables to be converted to numerical values that can then be easily used in the visual machine learning interface for classification and regression tasks.

_____

7   *http://huggingface.co/docs/transformers/index*

Leverage LLMs in Dataiku

Quickly integrate the latest machine learning models and neural networks
into your operation using Dataiku.

WATCH THE DEMO