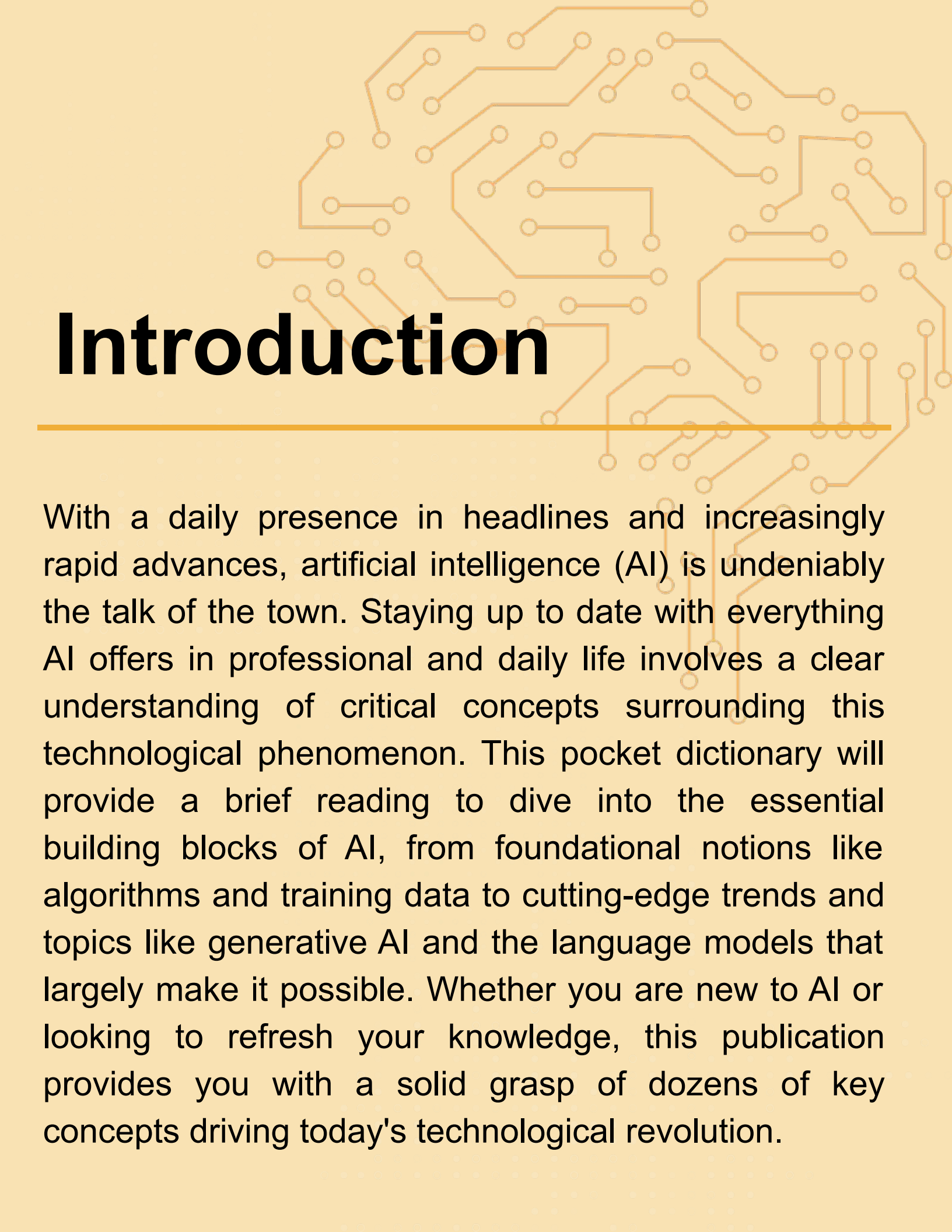


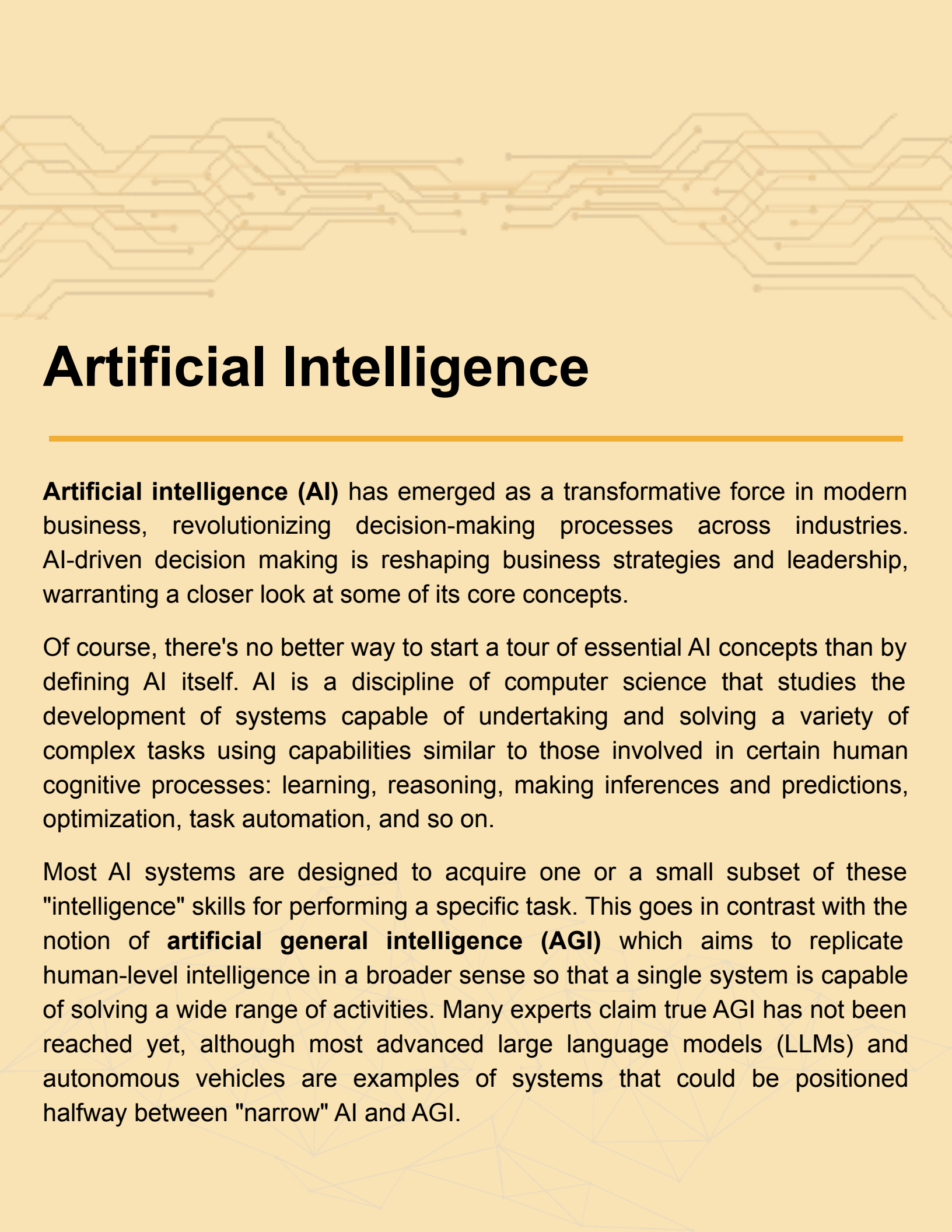
ARTIFICIAL INTELLIGENCE POCKET DICTIONARY





Introduction

With a daily presence in headlines and increasingly rapid advances, artificial intelligence (AI) is undeniably the talk of the town. Staying up to date with everything AI offers in professional and daily life involves a clear understanding of critical concepts surrounding this technological phenomenon. This pocket dictionary will provide a brief reading to dive into the essential building blocks of AI, from foundational notions like algorithms and training data to cutting-edge trends and topics like generative AI and the language models that largely make it possible. Whether you are new to AI or looking to refresh your knowledge, this publication provides you with a solid grasp of dozens of key concepts driving today's technological revolution.

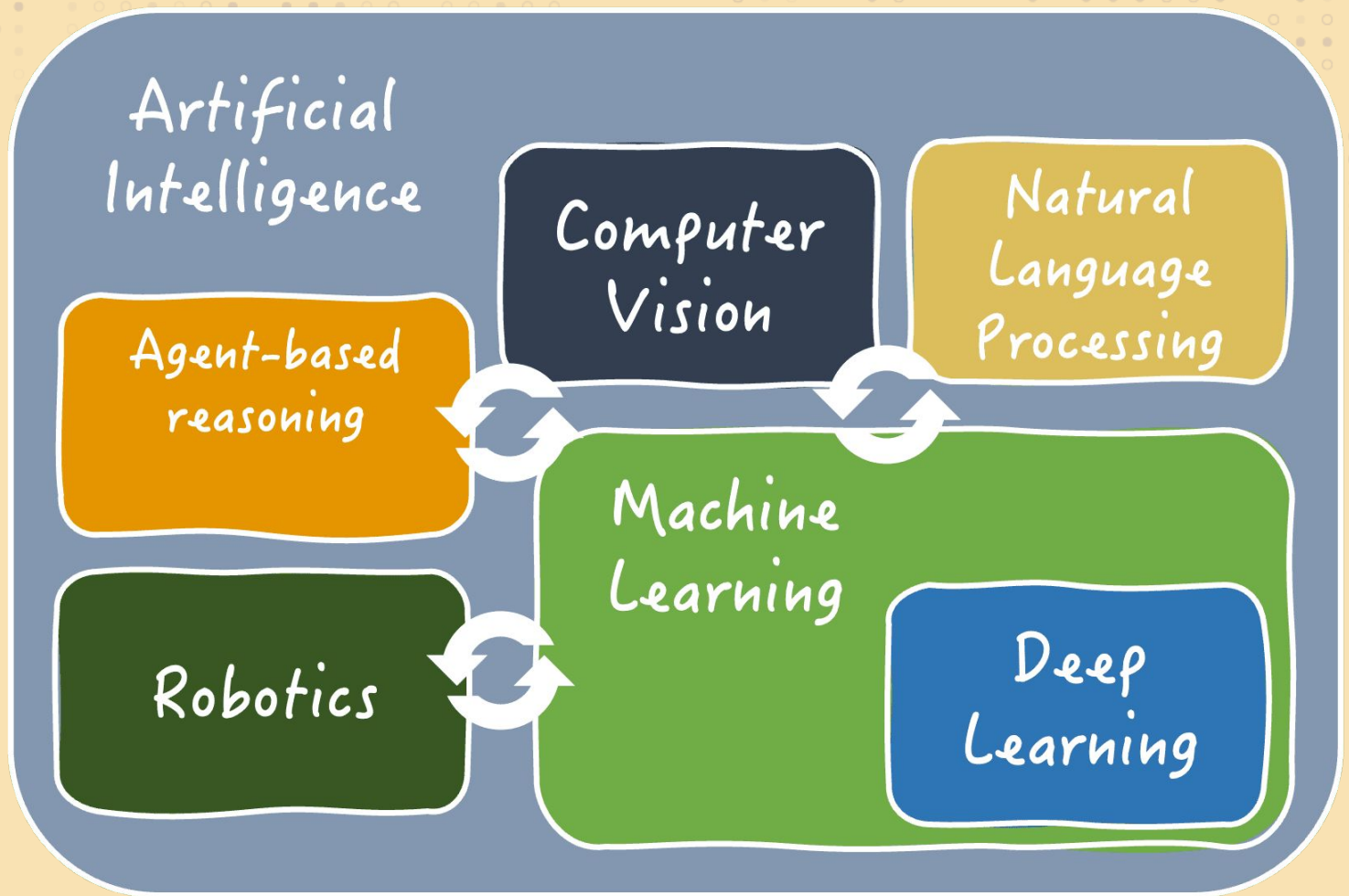


Artificial Intelligence

Artificial intelligence (AI) has emerged as a transformative force in modern business, revolutionizing decision-making processes across industries. AI-driven decision making is reshaping business strategies and leadership, warranting a closer look at some of its core concepts.

Of course, there's no better way to start a tour of essential AI concepts than by defining AI itself. AI is a discipline of computer science that studies the development of systems capable of undertaking and solving a variety of complex tasks using capabilities similar to those involved in certain human cognitive processes: learning, reasoning, making inferences and predictions, optimization, task automation, and so on.

Most AI systems are designed to acquire one or a small subset of these "intelligence" skills for performing a specific task. This goes in contrast with the notion of **artificial general intelligence (AGI)** which aims to replicate human-level intelligence in a broader sense so that a single system is capable of solving a wide range of activities. Many experts claim true AGI has not been reached yet, although most advanced large language models (LLMs) and autonomous vehicles are examples of systems that could be positioned halfway between "narrow" AI and AGI.



Are AI algorithms and AI models the same? Not quite.

An **algorithm** is a set of instructions for solving a problem, and in an AI algorithm, those instructions are designed to give computers the ability to learn by themselves how to solve the problem.

Meanwhile, an AI **model** is like a prebuilt box that contains the result of a learning (or training) process by being exposed to data: think of an AI model as a readily available solution to make predictions or perform tasks based on new data.

And now, on to some specific concept definitions, grouped broadly by 3 distinct AI sub domains.

Machine Learning Concepts

The largest subarea within AI is undoubtedly **machine learning** (ML), which focuses on systems that learn from data to perform tasks like classifying images, estimating sales, and detecting suspicious bank transactions. ML is often used interchangeably with AI, but as shown in the above diagram, ML is still just one part of AI, alongside other subareas. Nonetheless, the most advanced and evolved forms of ML systems today are designed to strongly overlap with other AI areas.

1. Supervised Learning

Definition: Many machine learning models learn to make predictions by being exposed to labeled examples — that is, observations that have associated output values or labels. This is called supervised learning, and it entails tasks such as regression, classification, and time series forecasting. A key requirement for supervised learning is the availability of high-quality labeled data.

Why it's key: Having a model learn to classify images of animals, people, etc., or accurately classify a financial transaction as legitimate or fraudulent, requires learning from a sufficient amount of representative, labeled examples of historical data with “known predictions”. By giving a supervised learning model a set of houses with diverse features alongside their associated prices, a model can learn the (sometimes complex) relationships between a house's attributes and its price, hence becoming skilled at accurately estimating the unknown price of future house data.

2. Unsupervised Learning

Definition: Labeled data (i.e. data examples with known outputs) are not always available. Still, there is a family of machine learning models suitable for learning patterns and underlying structure from this unlabeled data. Unsupervised learning encompasses techniques to discover hidden relationships and groups of similar data, including tasks like clustering, anomaly detection, dimensionality reduction, and more.

Why it's key: Since real-world data often lacks labels, unsupervised learning is sometimes as important as supervised learning — if not even more so. Techniques like clustering and anomaly detection can reveal structure, identify outliers, and generate insights about data like a body of customers, thereby turning unlabeled data into useful, labeled information.

3. Reinforcement Learning

Definition: Can you remember the last time you saw a small child learning to do something like using a toy by trial and error? This “trial and error” principle, or learning from experience, is the essence of reinforcement learning: a subarea of machine learning focused on training agents to make sequential decisions while pursuing a goal by maximizing (cumulative) rewards based on interactions with an environment.

Why it's key: Supervised and unsupervised learning are limited in some specific application domains where reinforcement learning can shine — for instance, in robotics, game playing, recommender systems, and autonomous vehicles — where sequential decisions must be made over time predicated on feedback.

4. Overfitting and Underfitting

Definition: Two very common issues found when training a machine learning model — particularly supervised learning models like classifiers and regressors — are their inability to perform accurate predictions or generalize to new, unseen data. Underfitting occurs when a model is too simple to capture the underlying structure of the data. Meanwhile, at the opposite end of the spectrum, overfitting is a very common problem in trained

machine learning models that “memorize” (or learn excessively from) the training data but fail to generalize to new inputs.

Why it's key: Understanding overfitting and underfitting is essential to detect and deal with these common problems in machine learning systems. Becoming familiar with strategies like regularization, cross-validation (see below), and reducing model complexity is the first step to tackling these problems and building systems that perform well in real-world scenarios.

5. Bias-Variance Tradeoff

Definition: The bias-variance tradeoff is an important design decision to consider in building machine learning models. High bias happens when a model is too simplistic and fails to capture the true patterns in the data, whereas high variance occurs when a model is too sensitive to the training data and captures noise (irrelevant information for predictions or inference) as if it were important information. As one increases, the other tends to decrease; therefore, finding the right balance between bias and variance is crucial for achieving optimal model performance.

Why it's key: Having a machine learning model with high bias or variance can have a negative impact in practice. For example, an e-mail classifier that is too biased might miss many spam messages, while one with high variance might mistakenly flag relevant messages as spam due to overfitting on training data.

6. Loss Function

Definition: The process of training a machine learning model on (typically labeled) data involves applying an optimization algorithm — see concept number 7 below — that iteratively learns from initially large prediction errors made by the model, gradually adjusting it so that its prediction errors become smaller and smaller. The loss function is the mathematical way to model these errors made during training, and it is given by the difference between the model's predicted outputs and the actual target values in data examples with known outputs.

Why it's key: The loss function is the compass that guides the optimization process, providing feedback on how well the model is performing. Hence, convergence towards an accurate model takes place thanks to minimizing this loss function.

7. Gradient Descent

Definition: If the loss function is the compass to guide the machine learning model toward a high-quality version of itself, the gradient descent algorithm is like the hiking boots that help the model (just like a hiker) navigate through the space of possible solutions (model versions) toward one that minimizes the loss function. The principle is similar to that of a hiker who is in the middle of a mountain slope and tries to find the lowest altitude point one step at a time: put it simply, it boils down to following the steepest downward direction, i.e. descending along the gradient.

Why it's key: The gradient descent algorithm and its improved variants are the key elements to reducing a model's loss function, thereby making the model effectively learn from data. Even the most sophisticated and state-of-the-art machine learning solutions today resort to this family of algorithms to optimize model performance.

8. Cross-Validation

Definition: Cross-validation is a well-established methodology to measure and validate not only a machine learning model's performance while it is being trained, but also to estimate how well it will generalize to future data once deployed. It partitions the training set into multiple subsets or folds, using some for training and others for validation in a rotating and iterative manner. The process is repeated multiple times, and the results are averaged to obtain a robust estimate of model performance.

Why it's key: Cross-validation is a more reliable approach than using a standalone validation split, because it reduces the risk of biased evaluation results and helps prevent overfitting the model to a single validation subset.

9. Feature Engineering

Definition: A high-quality machine learning model requires high-quality data to be properly trained. Preparing and preprocessing the data is an essential step in every machine learning project, and feature engineering, in which raw data is transformed into meaningful input features, becomes vital. Examples of feature engineering processes for machine learning modeling include encoding categorical variables numerically, scaling numerical values that may take disparate ranges, creating new attributes based on the interaction between existing ones, and extracting date or text features.

Why it's key: Feature engineering sometimes helps discover useful features that can have a positive impact on the resulting machine learning model performance, namely by improving model accuracy, reducing training time, and fostering interpretability.

10. Model Evaluation Metrics

Definition: Whether you are building a machine learning model for classification, regression, clustering, or natural language processing tasks, you may want to evaluate how well your machine learning model performs at solving that task, particularly on a set of new or diverse data examples. Model evaluation metrics like accuracy, precision, recall, F1-score, or mean squared error are therefore essential to quantify and compare model performance.

Why it's key: Without well-defined model evaluation metrics, it would be difficult to assess whether a model is genuinely effective, not to mention comparing different models or model versions, or making informed decisions about deploying a seemingly promising model in real-world applications.



Generative AI Concepts

Generative AI refers to AI systems that have been trained to create new content, such as text, images, or music, as a result of learning patterns from existing data. This area of AI is mostly founded on advanced deep learning architectures, and it's behind much of the explosion of available apps and tools we are witnessing today to bring closer AI capabilities to the general public, particularly those related to creativity.

1. Foundation Model

Definition: A foundation model is a large AI model, typically a deep neural network, trained on massive and diverse datasets like internet text or image libraries. These models learn general patterns and representations, enabling them to be fine-tuned for numerous specific tasks without requiring the creation of new models from scratch. Examples include large language models, diffusion models for images, and multimodal models combining various data types.

Why it's key: Foundation models are central to today's generative AI boom. Their broad training grants them emergent abilities, making them powerful and adaptable for a variety of applications. This reduces the cost needed to create specialized tools, forming the backbone of modern AI systems from chatbots to image generators.

2. Large Language Model (LLM)

Definition: An LLM is a vast natural language processing (NLP) model, typically trained on terabytes of data (text documents) and defined by millions to billions of parameters, capable of addressing language understanding and generation tasks at unprecedented levels. They normally rely on a deep learning architecture called a transformer, whose

so-called attention mechanism enables the model to weigh the relevance of different words in context and capture the interrelationship between words, thereby becoming the key behind the success of massive LLMs like ChatGPT.

Why it's key: The most prominent AI applications today, like ChatGPT, Claude, and other generative tools, along with customized conversational assistants in myriad domains, are all based on LLMs. The capabilities of these models have surpassed those of more traditional NLP approaches, such as recurrent neural networks, in processing sequential text data.

3. Diffusion Model

Definition: Much like LLMs are the leading type of generative AI models for NLP tasks, diffusion models are the state-of-the-art approach for generating visual content like images and art. The principle behind diffusion models is to gradually add noise to an image and then learn to reverse this process through denoising. By doing so, the model learns highly intricate patterns, ultimately becoming capable of creating impressive images that often appear photorealistic.

Why it's key: Diffusion models stand out in today's generative AI landscape, with tools like DALL·E and Midjourney capable of producing high-quality, creative visuals from simple text prompts. They've become especially popular in business and creative industries for content generation, design, marketing, and more.

4. Prompt Engineering

Definition: Did you know the experience and outcomes of using LLM-based applications like ChatGPT heavily depend on your ability to ask for something you need the right way? The craftsmanship of acquiring and applying that ability is known as prompt engineering, and it entails designing, refining, and optimizing user inputs or prompts to guide the model toward desired outputs. Generally speaking, a good prompt should be clear, specific, and most importantly, goal-oriented.

Why it's key: By getting familiar with key prompt engineering principles and guidelines, the chances of obtaining accurate, relevant, and useful responses are maximized. And just like any skill, all it takes is consistent practice to master it.

5. Retrieval Augmented Generation

Definition: Standalone LLMs are undeniably remarkable "AI titans" capable of addressing extremely complex tasks that just a few years ago were considered impossible, but they have a limitation: their reliance on static training data, which can quickly become outdated, and the risk of a problem known as hallucinations (discussed later). Retrieval augmented generation (RAG) systems arose to overcome these limitations and eliminate the need for constant (and very expensive) model retraining on new data by incorporating an external document base accessed via an information retrieval mechanism similar to those used in modern search engines, called the retriever module. As a result, the LLM in a RAG system generates responses that are more factually correct and grounded in up-to-date evidence.

Why it's key: Thanks to RAG systems, modern LLM applications are easier to update, more context-aware, and capable of producing more reliable and trustworthy responses; hence, real-world LLM applications are rarely exempt from RAG mechanisms at present.

6. Hallucination

Definition: One of the most common problems suffered by LLMs, hallucinations occur when a model generates content that is not grounded in the training data or any factual source. In such circumstances, instead of providing accurate information, the model simply "decides to" generate content that at first glance sounds plausible but could be factually incorrect or even nonsensical. For example, if you ask an LLM about a historical event or person that doesn't exist, and it provides a confident but false answer, that is a clear example of hallucination.

Why it's key: Understanding hallucinations and why they happen is critical to knowing how to address them. Common strategies to reduce or manage model hallucinations include curated prompt engineering skills, applying post-processing filters to generated responses, and integrating RAG techniques to ground generated responses in real data.

7. Fine-tuning (vs. Pre-training)

Definition: Generative AI models like LLMs and diffusion models have large architectures defined by up to billions of trainable parameters, as discussed earlier. Training such models follows two main approaches. Model pre-training involves training the model from scratch on massive and diverse datasets, taking considerably longer and requiring vast amounts of computational resources. This is the approach used to create foundation models. Meanwhile, model fine-tuning is the process of taking a pre-trained model and exposing it to a smaller, more domain-specific dataset, during which only part of the model's parameters are updated to specialize it for a particular task or context. Needless to say, this process is much more lightweight and efficient compared to full-model pre-training.

Why it's key: Depending on the specific problem and data available, choosing between model pre-training and fine-tuning is a crucial decision. Understanding the strengths, limitations, and ideal use cases where each approach should be selected helps developers build more effective and efficient AI solutions.

8. Context Window

Definition: Context is a very important part of user inputs to generative AI models, as it establishes the information to be considered by the model when generating a response. However, the context window or length must be carefully managed for several reasons. First, models have fixed context length limitations, which limit how much input they can process in one interaction. Second, a very short context may yield incomplete or irrelevant answers, whereas an overly detailed context can overwhelm the model or affect performance efficiency.

Why it's key: Managing context length is a critical design decision when building advanced generative AI solutions such as RAG systems, where techniques like context/knowledge chunking, summarization, or hierarchical retrieval are utilized to manage long or complex contexts effectively.

9. AI Agent

Definition: While the notion of AI agents dates back decades, and autonomous agents and multi-agent systems have long been part of AI in scientific contexts, the rise of generative AI has renewed focus on these systems — recently referred to as "Agentic AI." Agentic AI is one of generative AI's biggest trends, as it pushes the boundaries from simple task execution to systems capable of planning, reasoning, and interacting autonomously with other tools or environments.

Why it's key: The combination of AI agents and generative models has driven major advances in recent years, leading to achievements such as autonomous research assistants, task-solving bots, and multi-step process automation.

10. Multimodal AI

Definition: Multimodal AI systems are part of the latest generation of generative models. They integrate and process multiple types of data, such as text, images, audio, or video, both as input and in generating multiple output formats, thereby expanding the range of use cases and interactions they can support.

Why it's key: Thanks to multimodal AI, it is now possible to describe an image, answer questions about a chart, generate a video from a prompt, and more — all in one unified system. In short, the overall user experience is dramatically enhanced.



Large Language Model Concepts

Large language models (LLMs) are AI systems of a huge magnitude: they're trained on vast text datasets (up to billions of documents) to understand and generate human-like language at an unprecedented level. Tools like ChatGPT and Claude are well-known examples of LLMs deployed in the real world.

1. Transformer Architecture

Definition: The transformer is the foundation of large language models. It is a deep neural network architecture raised to its highest exponent, consisting of a variety of components and layers like position-wise feed-forward networks and self-attention that together allow for efficient parallel processing and context-aware representation of input sequences.

Why it's key: Thanks to the transformer architecture, it has become possible to understand complex language inputs and generate language outputs at an unprecedented level, overcoming the limitations of previous state-of-the-art natural language processing solutions.

2. Attention Mechanism

Definition: Originally envisaged for language translation tasks in recurrent neural networks, attention mechanisms analyze the relevance of every element in a sequence concerning elements in another sequence, both of varying length and complexity. While the basic attention mechanism is not typically part of transformer architectures underlying LLMs, they laid the foundations for enhanced approaches (as we will discuss shortly).

Why it's key: Attention mechanisms are key in aligning source and target text sequences in tasks like translation and summarization, turning the language understanding and generation processes into highly contextual tasks.

3. Self-Attention

Definition: If there is a type of component within the transformer architecture that is mainly responsible for the success of LLMs, that is the self-attention mechanism. Self-attention overcomes conventional attention mechanisms' limitations like long-range sequential processing by allowing each word — or token, more precisely — in a sequence to attend to all other words (tokens) simultaneously, regardless of their position.

Why it's key: Paying attention to dependencies, patterns, and interrelationships among elements of the same sequence is incredibly useful to extract a deep meaning and context of the input sequence being understood, as well as the target sequence being generated as a response — thereby enabling more coherent and context-aware outputs.

4. Encoder and Decoder

Definition: The classical transformer architecture is roughly divided into two main components or halves: the encoder and the decoder. The encoder is responsible for processing and encoding the input sequence into a deeply contextualized representation, whereas the decoder focuses on generating the output sequence step-by-step utilizing both previously generated parts of the output and the encoder's resulting representation. Both parts are interconnected, so that the decoder receives processed results from the encoder (called hidden states) as input. Furthermore, both the encoder and the decoder innards are "replicated" in the form of multiple encoder layers and decoder layers, respectively: this level of depth helps the model learn more abstract and nuanced features of the input and output sequences.

Why it's key: The combination of an encoder and a decoder, each with their own self-attention components, is key to balancing input understanding with output generation in an LLM.

5. LLM Pre-Training

Definition: Just like the foundations of a house from scratch, pre-training is the process of training an LLM for the first time, that is, gradually learning all of its model parameters or weights. The magnitude of these models is such that they may take up to billions of parameters. Hence, pre-training is an inherently costly process that takes days to weeks to complete and requires massive and diverse corpora of text data.

Why it's key: Pre-training is vital to build an LLM that can understand and assimilate the general language patterns and semantics across a wide spectrum of topics.

6. LLM Fine-Tuning

Definition: Contrary to pre-training, fine-tuning is the process of taking an already pre-trained LLM and training it again on a comparatively smaller and more domain-specific set of data examples, thereby making the model specialized in a specific domain or task. While still computationally expensive, fine-tuning is less costly than pre-training a model from scratch, and it often entails updating model weights only in specific layers of the architecture rather than updating the entire set of parameters across the model architecture.

Why it's key: Having an LLM specialize in very concrete tasks and application domains like legal analysis, medical diagnosis, or customer support is important because general-purpose pre-trained models may fall short in domain-specific accuracy, terminology, and compliance requirements.

7. Embeddings

Definition: Machines and AI models do not truly understand language, but just numbers. This also applies to LLMs, so while we generally speak about models that "understand and generate language", what they do is handle a numerical representation of such language that keeps its key properties largely intact: these numerical (vector, to be more precise) representations are what we call embeddings.

Why it's key: Mapping input text sequences into embedding representations enables LLMs to perform reasoning, similarity analysis, and data generalization across contexts, all without losing the main properties of the original text; hence, raw responses generated by the model can be mapped back to semantically coherent and appropriate human language.

8. Prompt Engineering

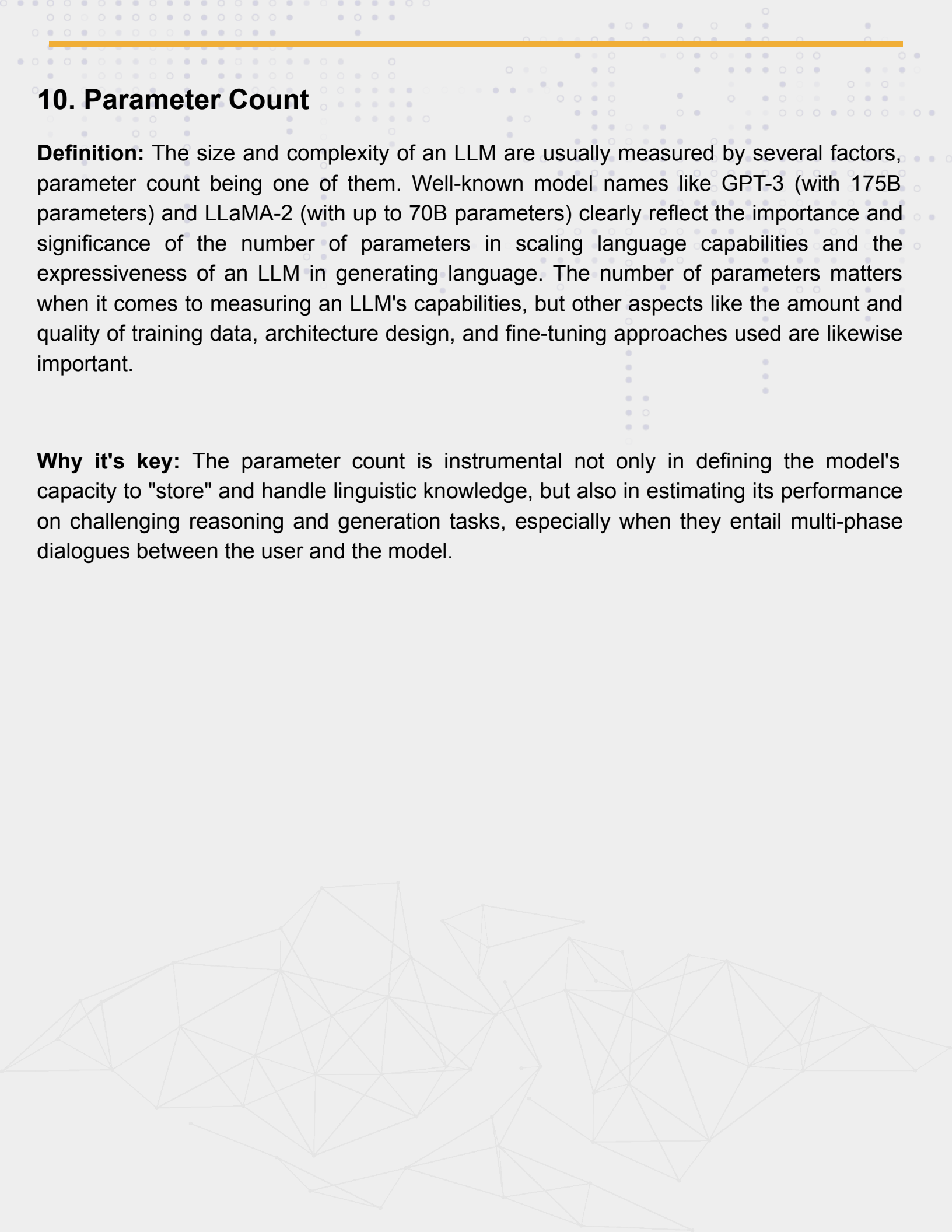
Definition: End users of LLMs should get familiar with best practices for optimal use of these models to achieve their goals, and prompt engineering stands out as a strategic and practical approach to this end. Prompt engineering encompasses a set of guidelines and techniques for designing effective user prompts that guide the model towards producing useful, accurate, and goal-oriented responses.

Why it's key: Oftentimes, obtaining high-quality, precise, and relevant LLM outputs is largely a matter of learning how to write high-quality prompts that are clear, specific, and structured to align the LLM's capabilities and strengths, e.g., by turning a vague user question into a precise and meaningful answer.

9. In-Context Learning

Definition: Also called few-shot learning, this is a method to teach LLMs to perform new tasks predicated on providing examples of desired outcomes and instructions directly in the prompt, without re-training or fine-tuning the model. It can be deemed as a specialized form of prompt engineering, as it fully leverages the model's gained knowledge during pre-training to extract patterns and adapt to new tasks on the fly.

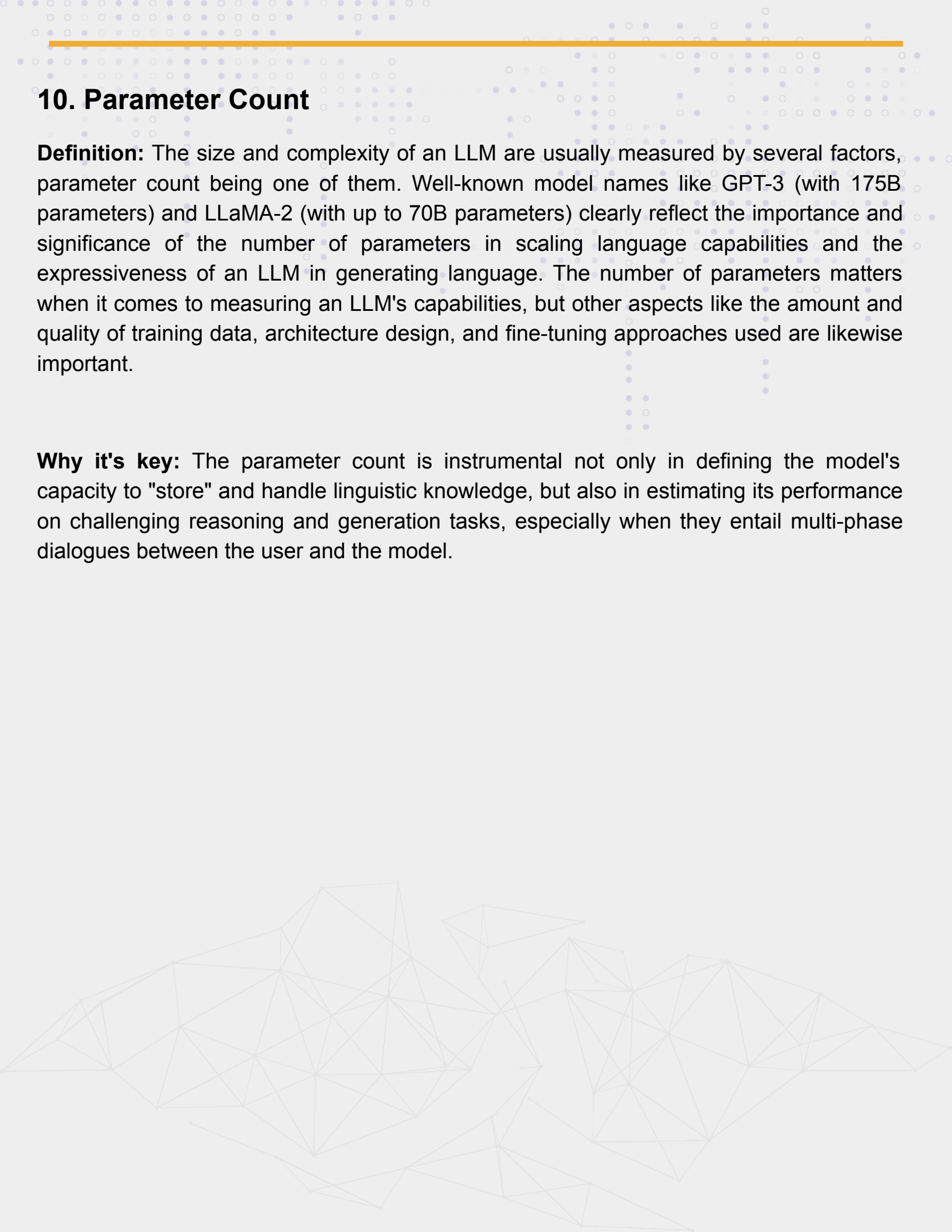
Why it's key: In-context learning has been proven as an effective approach to flexibly and efficiently learn to solve new tasks based on examples.



10. Parameter Count

Definition: The size and complexity of an LLM are usually measured by several factors, parameter count being one of them. Well-known model names like GPT-3 (with 175B parameters) and LLaMA-2 (with up to 70B parameters) clearly reflect the importance and significance of the number of parameters in scaling language capabilities and the expressiveness of an LLM in generating language. The number of parameters matters when it comes to measuring an LLM's capabilities, but other aspects like the amount and quality of training data, architecture design, and fine-tuning approaches used are likewise important.

Why it's key: The parameter count is instrumental not only in defining the model's capacity to "store" and handle linguistic knowledge, but also in estimating its performance on challenging reasoning and generation tasks, especially when they entail multi-phase dialogues between the user and the model.



About This Publication

The **Artificial Intelligence Pocket Dictionary** was written by **Iván Palomares Carrascosa**, a KDnuggets Technical Content Specialist.



Iván is a leader, writer, speaker, and adviser in AI, machine learning, deep learning & LLMs. He trains and guides others in harnessing AI in the real world.

Designed and edited by **Matthew Mayo**, KDnuggets Editor-in-Chief.

The cover image was created with the help of ChatGPT.

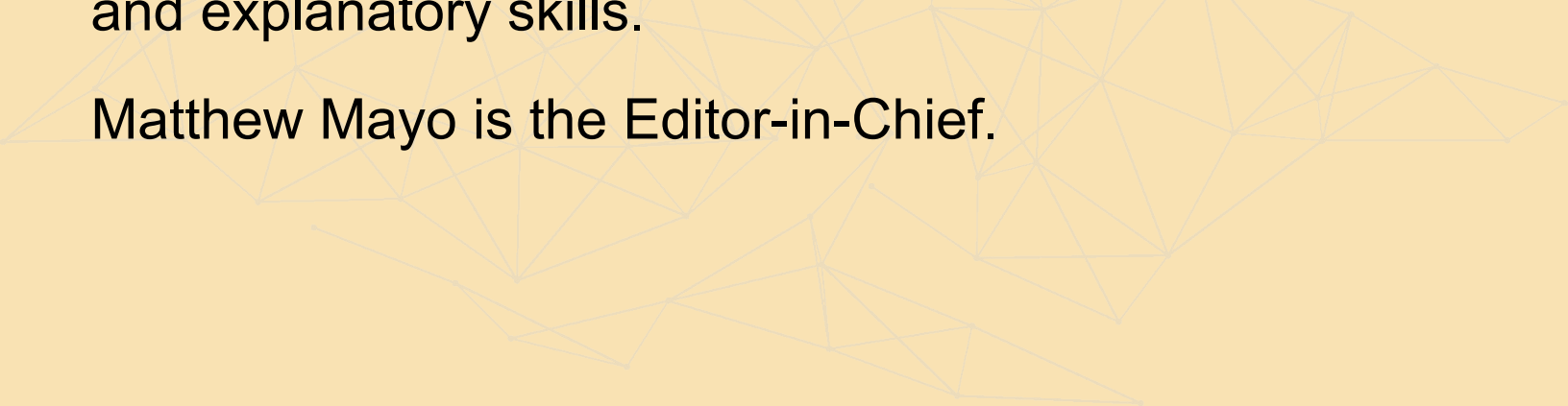
A decorative graphic at the top of the page consisting of a complex network of thin, light-orange lines that resemble a circuit board or a neural network, with small dots at various points.

About Us

KDnuggets is an online platform on business data science, machine learning, analytics & AI. We publish content every weekday, and cover tutorials, advice, news, software, career trends, meetings, courses, data, education, webinars and more.

KDnuggets was founded by Gregory Piatetsky-Shapiro, and has been in operation for over 30 years. Our team has expertise and experience across the data spectrum, and are known for their technical knowledge and explanatory skills.

Matthew Mayo is the Editor-in-Chief.

A decorative graphic at the bottom of the page consisting of a network of thin, light-blue lines forming a complex geometric pattern of triangles and polygons.