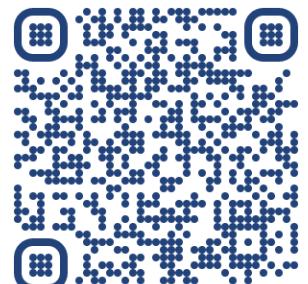


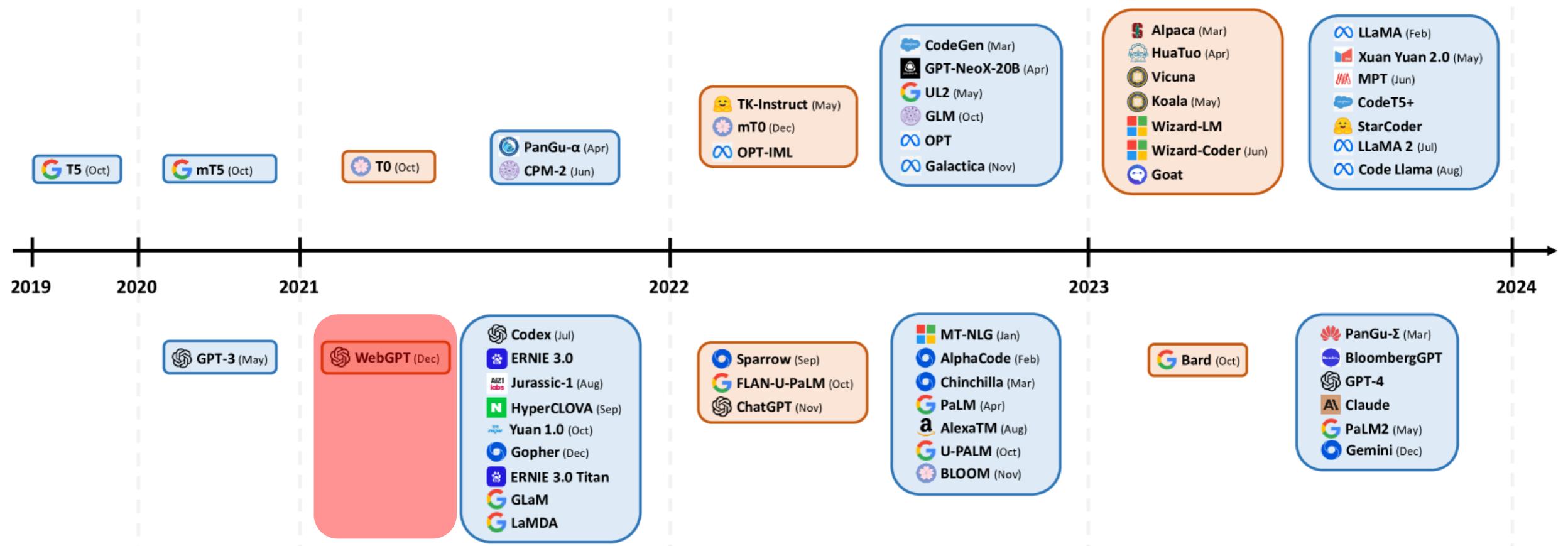
Generatieve AI

*“Hoe kan het gebruik van
Gen-AI valide en veilig
worden toegepast?”*

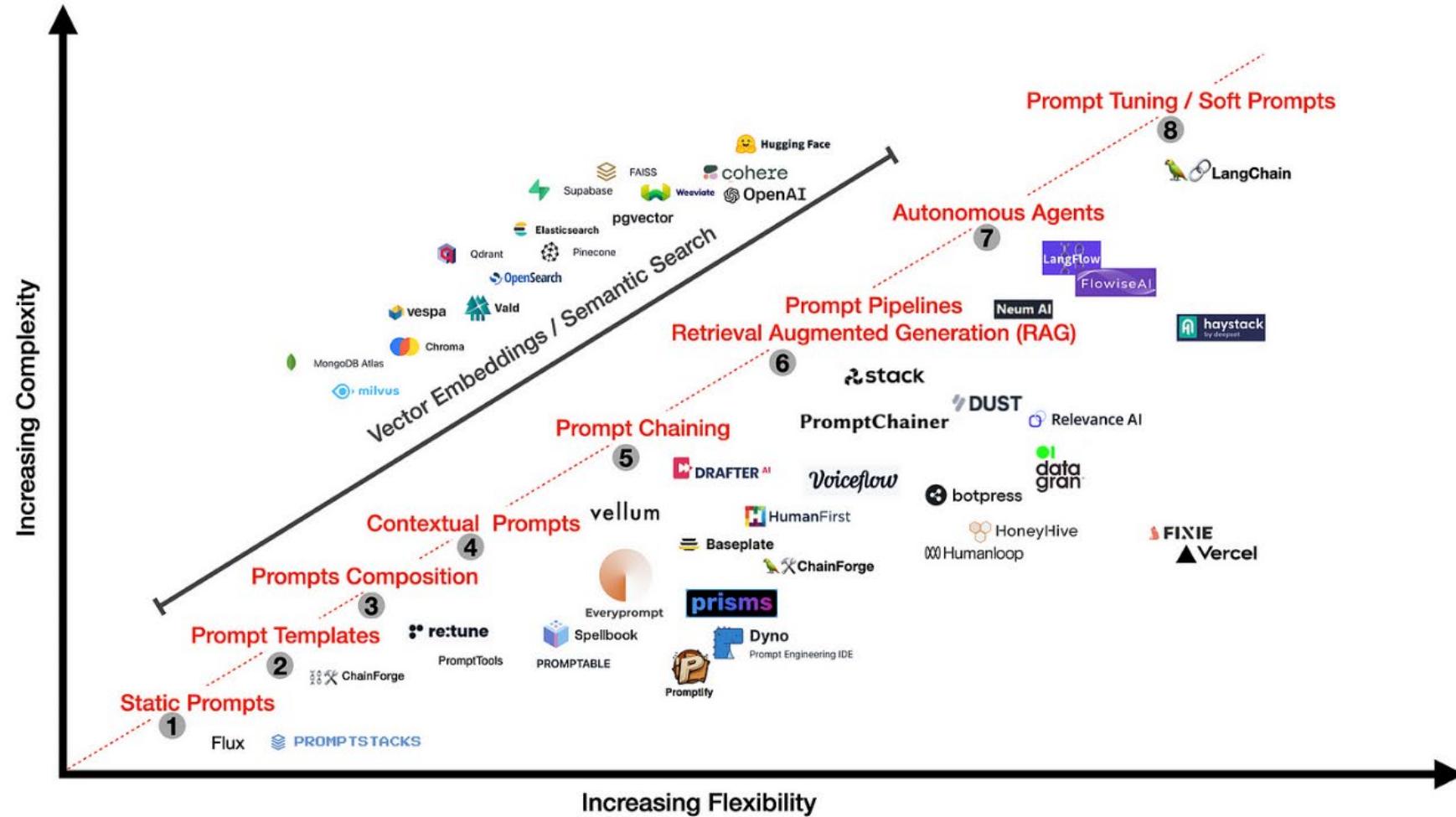


hr.nl/ai

Onstaansgeschiedenis + evolutie van grote taal modellen {LLM}

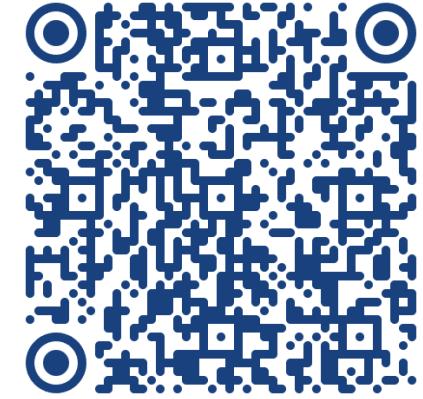


LLM implementations



<https://blogs.novita.ai/exploring-architectural-structures-and-functional-capacities-of-langs/>

Misvattingen die publieke organisaties ervan weerhouden grote taal modellen {LLM} te gebruiken om generatieve AI optimaal te benutten



Misvatting 1:

Mijn organisatie beschikt niet over de juiste tools en platforms om betrouwbare AI te ontwikkelen

Misvatting 2:

AI is te gecompliceerd en veel te kostbaar voor mijn organisatie om vanaf nul op te bouwen

Misvatting 3:

Mijn organisatie beschikt niet over de juiste vaardigheden en expertise om AI te ontwikkelen

Misvatting 4:

De AVG staat het toepassen van AI in het onderwijs in de weg

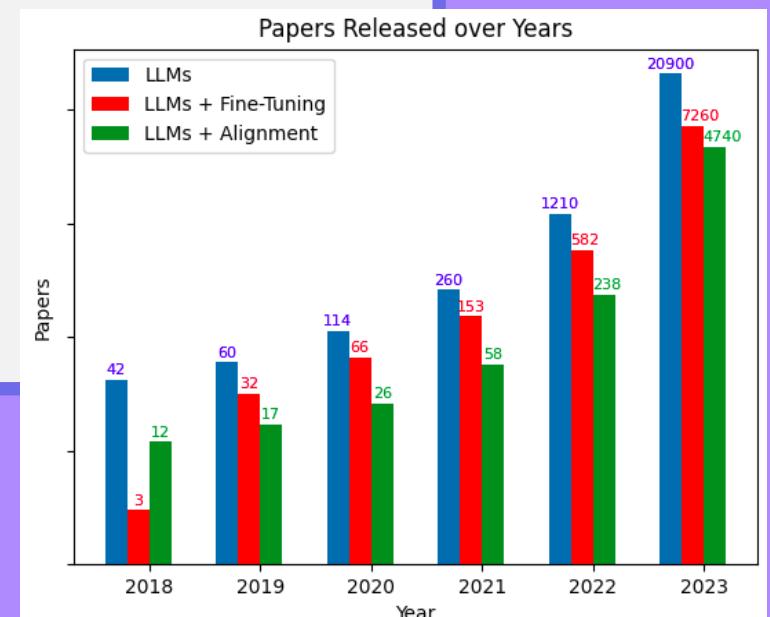


HOGESCHOOL
ROTTERDAM

CONTEXT:

Wat zijn grote taal modellen eigenlijk?

Waarom heeft Generatieve AI zo'n enorme impact op onderwijs & onderzoek



Defining AI is
nonsensical

Shaping Europe's digital future

| Home | Policies | Activities | News | Library | Funding | Calendar | Consultations | AI Office |

[Home](#) > [Library](#) > A definition of Artificial Intelligence: main capabilities and scientific disciplines

REPORT / STUDY | Publication 18 December 2018

A definition of Artificial Intelligence: main capabilities and scientific disciplines

This document expands the definition of Artificial Intelligence (AI) as defined in the Commission Communication on AI. It clarifies certain aspects of AI as a scientific discipline and as a technology, with the aim to avoid misunderstanding, to achieve a shared common knowledge of AI that can be fruitfully used also by non - AI experts, and to provide useful details that can be used in the discussion on both the AI ethics guidelines and the AI policies recommendations.

Downloads



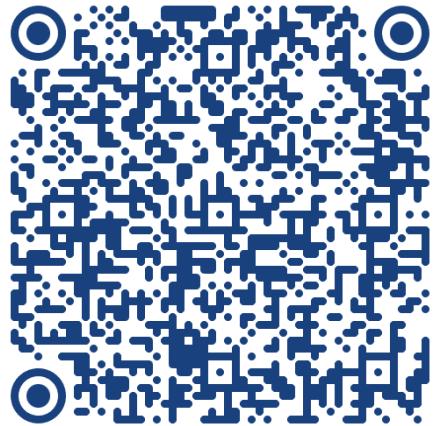
AI Definition.pdf

Download

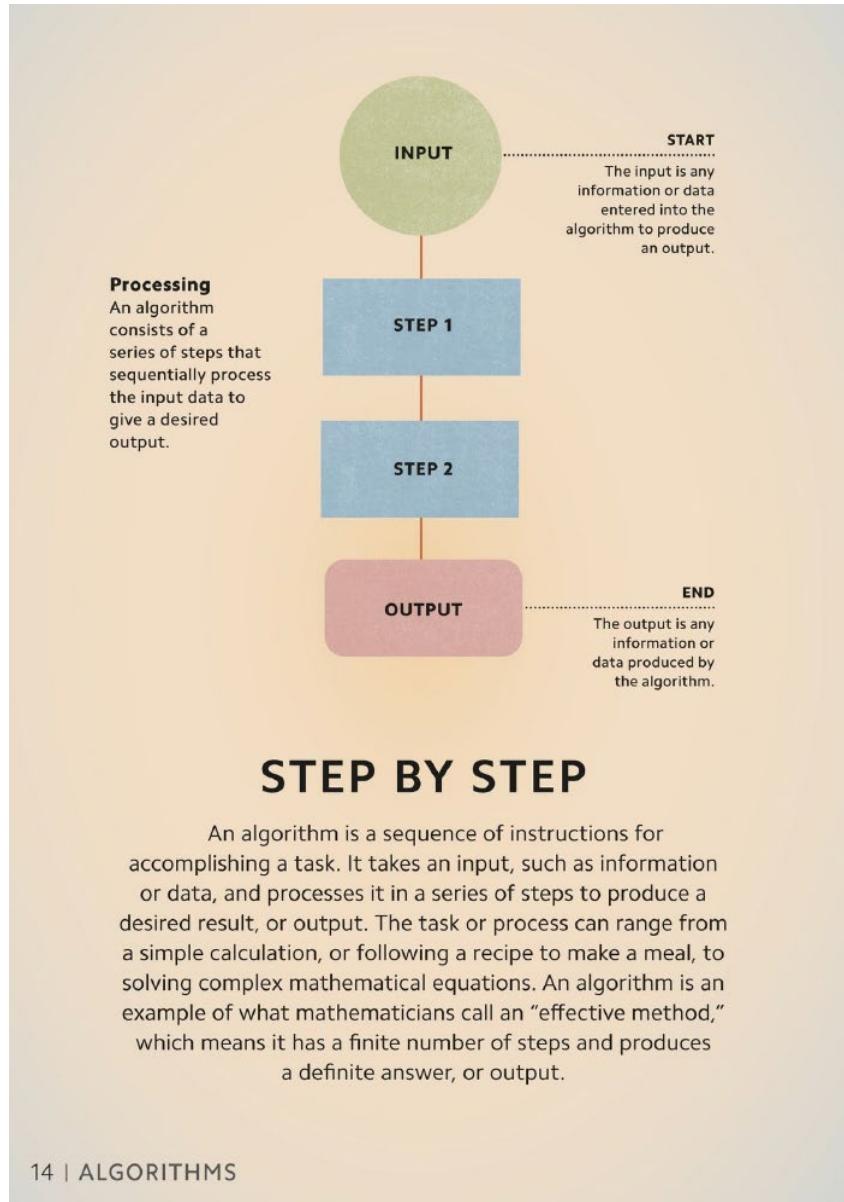
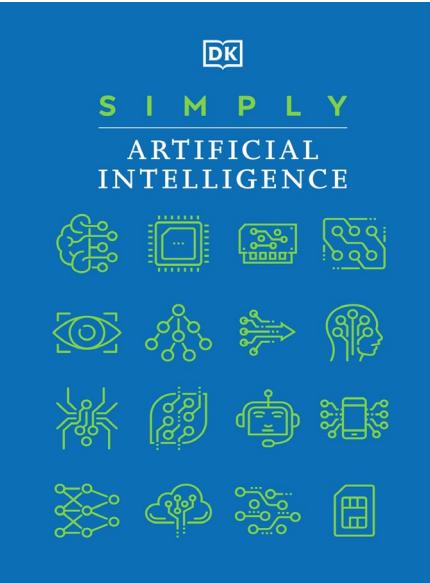
Related topics

[Artificial intelligence](#)

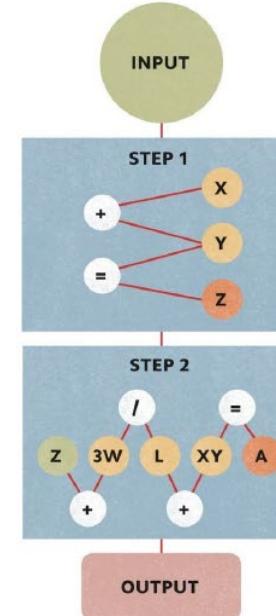
[Advanced Digital Technologies](#)



HOGESCHOOL
ROTTERDAM



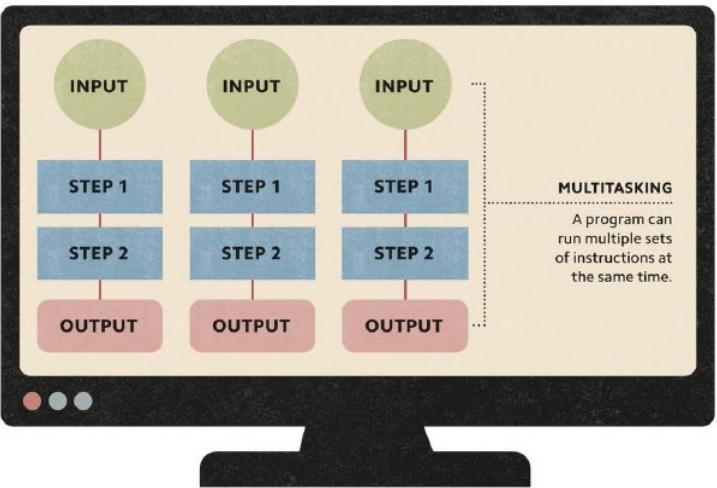
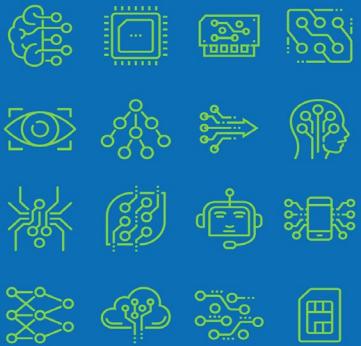
Components of calculation
Computations have an input and an output, and multiple steps. They can vary from simple sums to complex equations.



ALGORITHMS IN ACTION

A computation is a calculation that follows the steps of an algorithm (see opposite). The most straightforward example of computation is arithmetic calculation. For example, if you add together a pair of three-digit numbers in your head, you follow a series of steps, or an algorithm, to achieve this calculation. Computations use symbols to represent numbers, but symbols can represent almost anything else (see p.36). With the right symbols and the right algorithms, immensely complex computation becomes possible.

SIMPLY

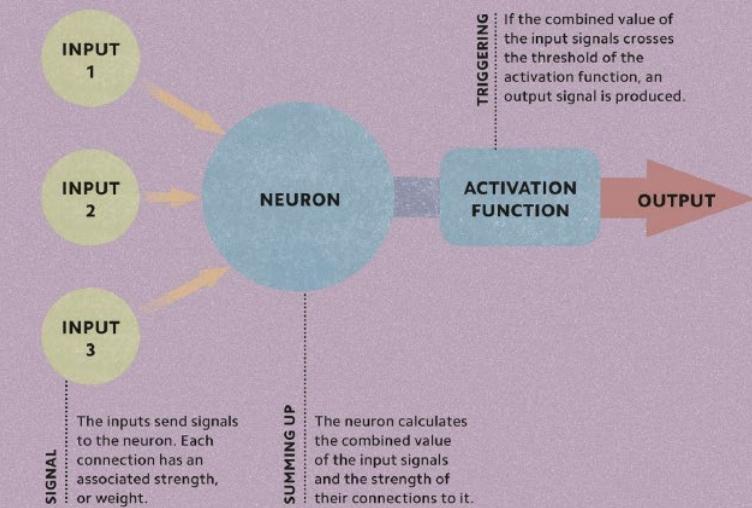
ARTIFICIAL
INTELLIGENCE

INSTRUCTING COMPUTERS

A program is a sequence of instructions written in code that enables a computer to perform one or more tasks. Charles Babbage (see opposite) imagined the first program. He was inspired by the design of a certain silk loom, which had parts that moved up or down in response to a pattern of holes punched into a card. Babbage recognized that these holes could store instructions to operate the cogs and levers of a machine he was designing: the "Analytical Engine". Modern computers work on the same principle, following sequences of instructions, which are usually written in binary code (see p.13).

ARTIFICIAL NEURONS

Each of the 86 billion neurons in the human brain is effectively a tiny processor, receiving electrical signals (inputs) from other neurons and sending out signals of its own (outputs). McCulloch and Pitts (see opposite) realized that neurons can act as logic gates—devices that can switch on and off (see p.13), depending on the input. The scientists described an imaginary neuron called a "threshold logic unit". This neuron works by first adding the values of its inputs (signals from other neurons) and then multiplying that value by a variable called a "weight" (see p.78)—this is the strength of a connection between neurons. If the input signals exceed a certain value (see p.79), the neuron is triggered to send an output signal. This triggering is called the "activation function".



**Defining what
characterizes AI**
gives Insight

{Human-in-the-Loop}

$$\mathbf{AI} = \mathbf{ML} + \mathbf{TD} + \mathbf{HITL}$$

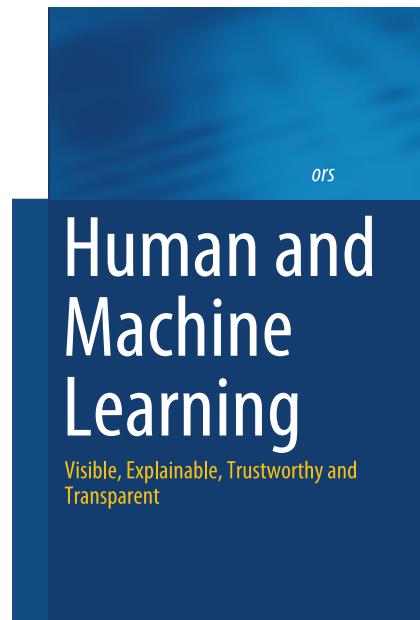


Artificial Intelligence:
in contrast to natural intelligence, it is *the ability of computer systems to perform tasks or actions that would normally require a human*

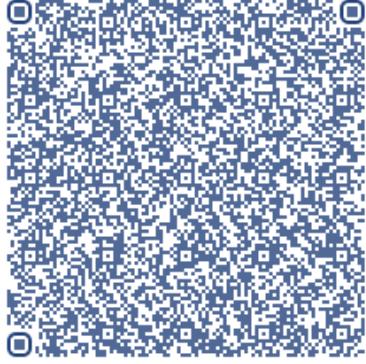
Machine Learning:
the ability of computer systems to use algorithms and statistical models to perform tasks without explicit instruction, through patterns and inferences

Training Data:
the data used to train a machine learning algorithm to perform a task in supervised machine learning

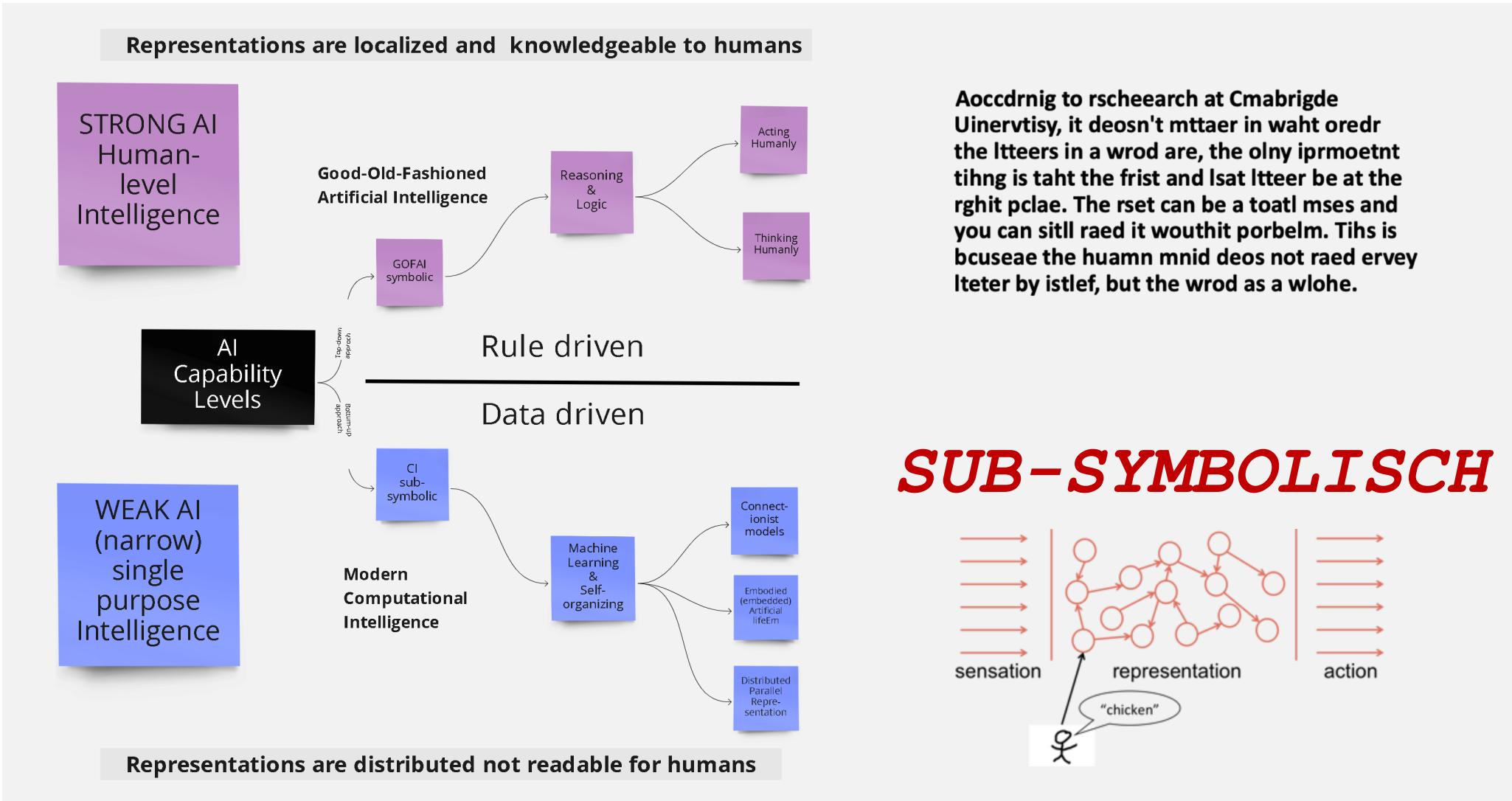
Human in the Loop:
the involvement of a human in training a machine learning algorithm



AI-taxonomie is complex

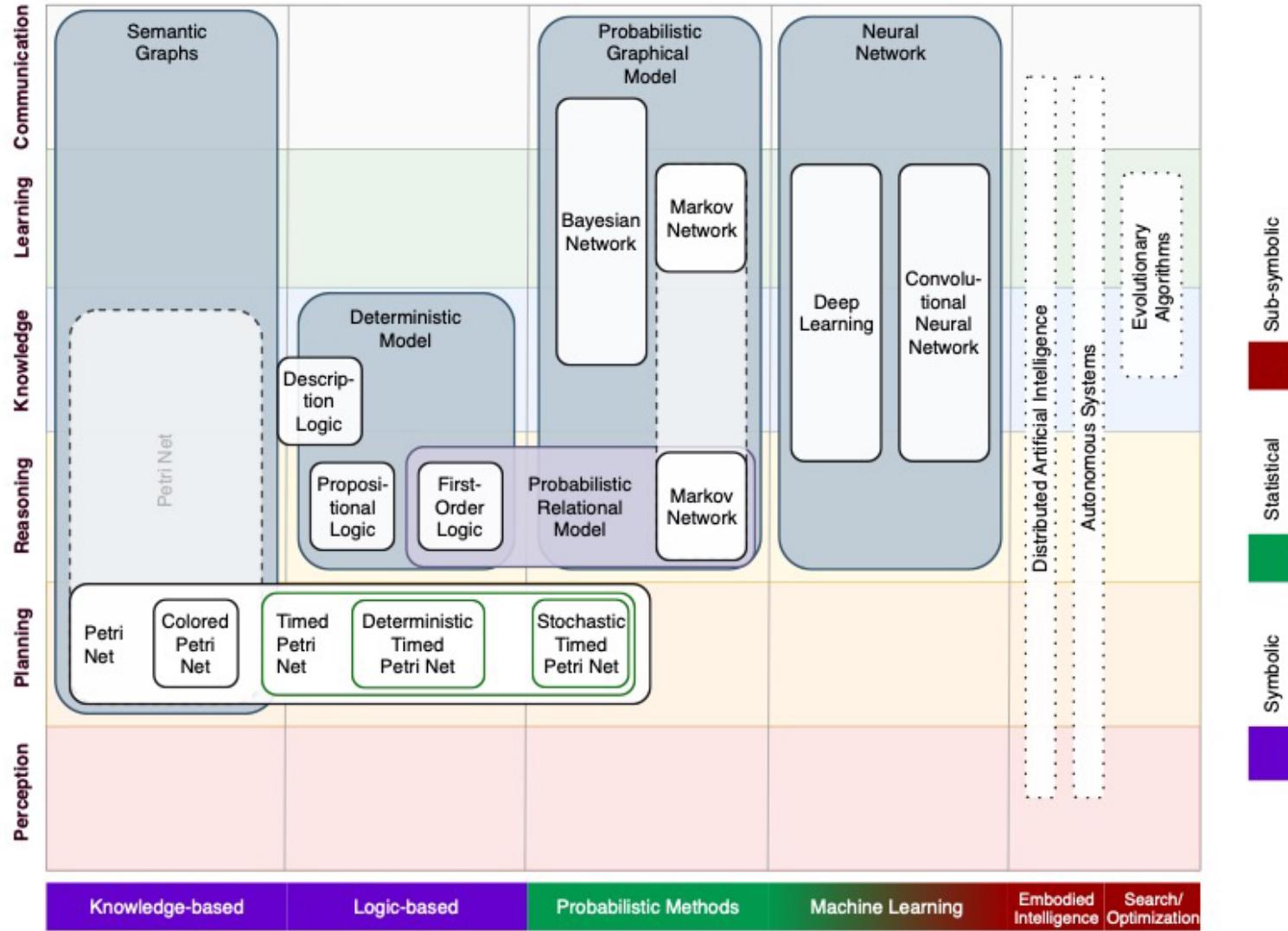
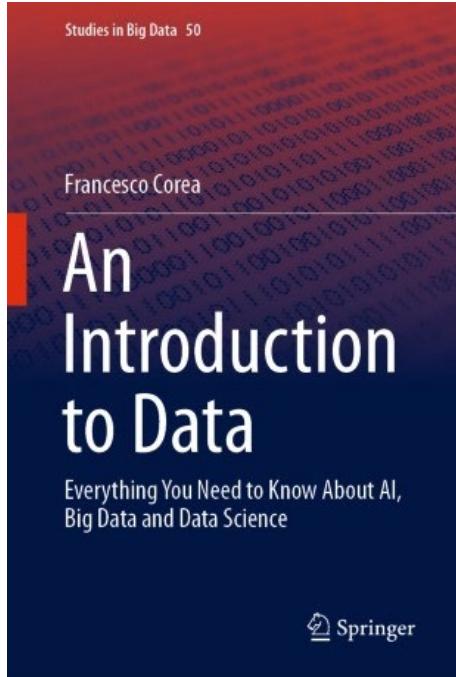


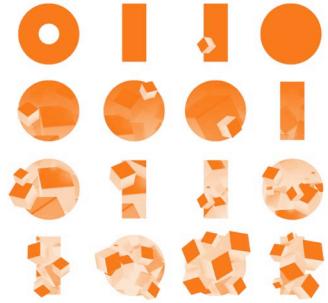
https://www.researchgate.net/publication/359424818_Designing_Neural_Networks_Through_Sensory_Ecology_Biology_to_the_rescue_of_AI_Produced_by_Living-Lab_AIRA_Hub_voor_Data_Responsible_AI_Hogeschool_Rotterdam_Lunch-Lezing_Creating-010_FEB_2022



{01}

Fundamentals





Artificial Intelligence:
How knowledge is created,
transferred, and used



Prof. Enrico Motta,
Professor of Knowledge
Technologies, The Open
University, United Kingdom

Trends in China, Europe,
and the United States

Foreword

Defining AI: new approaches help with AI ontologies



Prof. Enrico Motta,
Professor of Knowledge
Technologies, The Open
University, United Kingdom

"Disciplines do not exist per se. They emerge because of a collective construction process, whereby a community of researchers comes together, formulating and sharing common objectives, methods, and conceptualizations. Hence, disciplines are essentially about research communities. As these evolve, so do the associated disciplines. Thus, attempts at characterizing disciplines are in my view more successful if they follow a bottom-up approach, focusing less on top-down definitions than on identifying the relevant body of work."

Given this premise, I am very happy to endorse this report produced by the Elsevier team, which provides an operational characterization of the field of AI, in terms of 600,000 documents and over 700 field-specific keywords. This is an impressive piece of work that, to my knowledge, provides the most comprehensive characterization of AI outputs produced so far. Crucially, in contrast with manually developed taxonomies of research areas, which inevitably end up reflecting the specific viewpoints of the experts involved in the process, this characterization is data-driven, using machine learning and text mining techniques to classify documents and identify the relevant keywords. Thus, in my view, the report enjoys greater validity, providing a more objective reflection of the variety of existing contributions to the AI field.

In addition to its scientific value, there is also no doubt that this report will be a very valuable practical resource for people who wish to explore this space. For example, it will be very interesting to use this comprehensive characterization of the AI field to get a better understanding of key trends and topics, especially when the relevant body of work may be spread across different

research communities, as it is the case, for instance, with work in the highly important of AI ethics.

On a personal level, this work is also very exciting for me because it provides the basis for interesting new research. One of my main research areas concerns the use of A technologies to develop innovative solutions can help people to make sense of the dynamics of scientific research. Within this broad context my team has developed an original approach to the automatic generation of taxonomic research areas and, for example, it would be extremely interesting to investigate to what extent these different methods can cover a research space and to what extent they can be combined to improve accuracy. This is just one example of the many interesting possibilities further research opened up by this work.

In sum, this is not just an excellent piece of work, but also the start of a very interesting research. I congratulate the Elsevier team on their tremendous work and I look forward to further developments in this space."

2.2 Seven AI research clusters

We aimed to provide more depth to our subsequent analysis by structuring AI into research areas, using an unsupervised clustering technique.⁵⁸ This approach maps the keywords of all co-occurrence within the documents. Co-occurrence indicates that those clusters do not stand alone, but strongly relate to each other, e.g., neural networks in a co-occurrence document. The resulting graph illustrates the subfields of AI (Figure 2.2) and the connections through co-occurrence in scholarly publications. The resulting graph is interactive.⁵⁹ The graph is interactive, allowing users to browse individual connections and clusters, by region and over time.

As shown in Figure 2.2, AI seems to cluster around the areas of Search and Optimization, Fuzzy Systems, Natural Language Processing and Knowledge Representation, Computer Vision, Machine Learning and Probabilistic Reasoning, Planning and Decision Making, and Neural Networks. These seven fields, such as self-driving cars and robotics, are embedded into Planning and Decision Making as they have fewer underlying publications. The clusters seem to focus on statistics-based AI. Knowledge-based capabilities, such as "Ontologies or Semantics," do not form a cluster on their own, but are embedded in other clusters, predominantly in Natural Language Processing and Knowledge Representation.⁶⁰ Future research might investigate the sensitivity of this approach to the number of keywords and related publications in terms of normalized proportions over time. The strong growth of publications in recent years within the learning system field might outweigh knowledge-based approaches from more than 5 years ago.

Figure 2.2 illustrates the breadth of industry keywords (green), especially in the areas of "Fuzzy Systems" and "Computer Vision," whereas specific research keywords appear in "Neural Networks," teaching keywords in "Search and Optimization," and media keywords in fields such as "Planning and Decision Making" and "Natural Language Processing and Knowledge Representation."⁶¹ The relatively low proportion of media-driven keywords could indicate that these are not key AI research fields, or that they are still in their research infancy, representing only a fraction of AI documents.

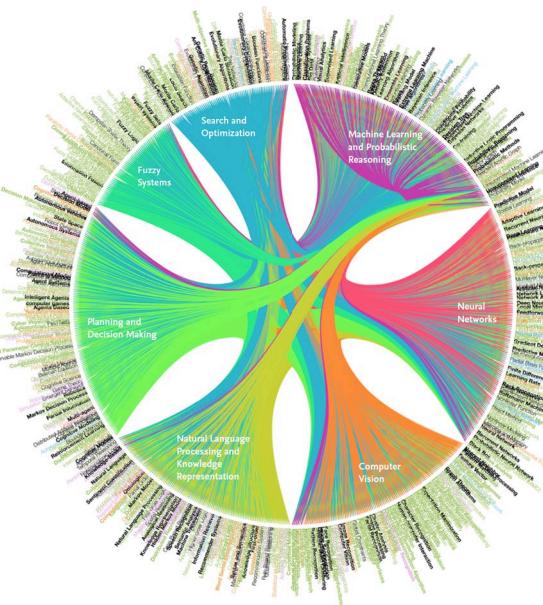
58 Latent clustering:
https://en.wikipedia.org/w/index.php?title=Latent_clustering&oldid=9100000
 "The Laton method is a simple, efficient and easy-to-implement method to analyze large networks. It is based on the assumption that a network is a sparse projection of a latent space. The original idea for the method is due to Domenic Lefebvre and Jean-Loup Guillaume, published in the journal *Journal of Statistical Mechanics: Theory and Experiment* in March 2009. The method was first published in "Fast unfolding of communities in large networks," Vincent D Blondel, Jean-Loup Guillaume, François Lambiotte, and Etienne Lefebvre, *J. Stat. Mech.: Theory Exp.*, P09008 (2008). DOI: 10.1088/1742-5468/2008/09/P09008. arXiv: <http://arxiv.org/abs/0803.0881>

59 <https://www.semantics.ai/connected/ai-resource-center/>
<https://www.semantics.ai/connected/ai-resource-center/>
 Fuzzy Systems: <https://www.semantics.ai/connected/ai-resource-center/#fuzzy-systems>
 Computer Vision: <https://www.semantics.ai/connected/ai-resource-center/#computer-vision>
 Machine Learning: <https://www.semantics.ai/connected/ai-resource-center/#machine-learning>
 Natural Language Processing: <https://www.semantics.ai/connected/ai-resource-center/#natural-language-processing>
 Neural Networks: <https://www.semantics.ai/connected/ai-resource-center/#neural-networks>

60 <https://www.semantics.ai/connected/ai-resource-center/>

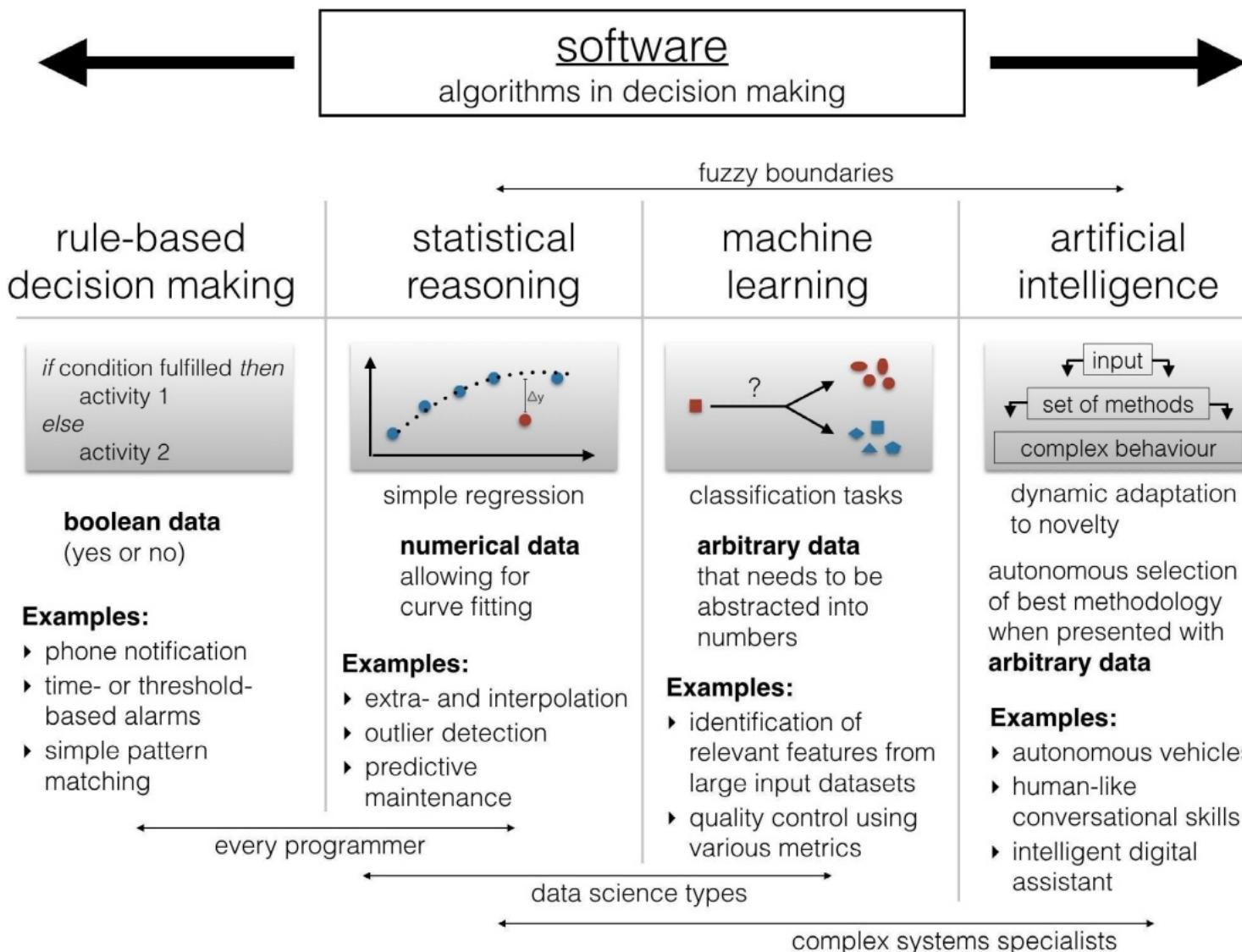
61 <https://www.semantics.ai/connected/ai-resource-center/>

In summary, our co-occurrence analysis reveals a substructure of the AI field, determined by its document corpus and layered relation. Those might influence the weighting and share of subfields, such as knowledge-based fields. Further research might explore normalized approaches to compare subfields per year and investigate the evolution of keywords to the structure of the field. In chapter 3 we will use these clusters to prevent global and regional trends in AI.

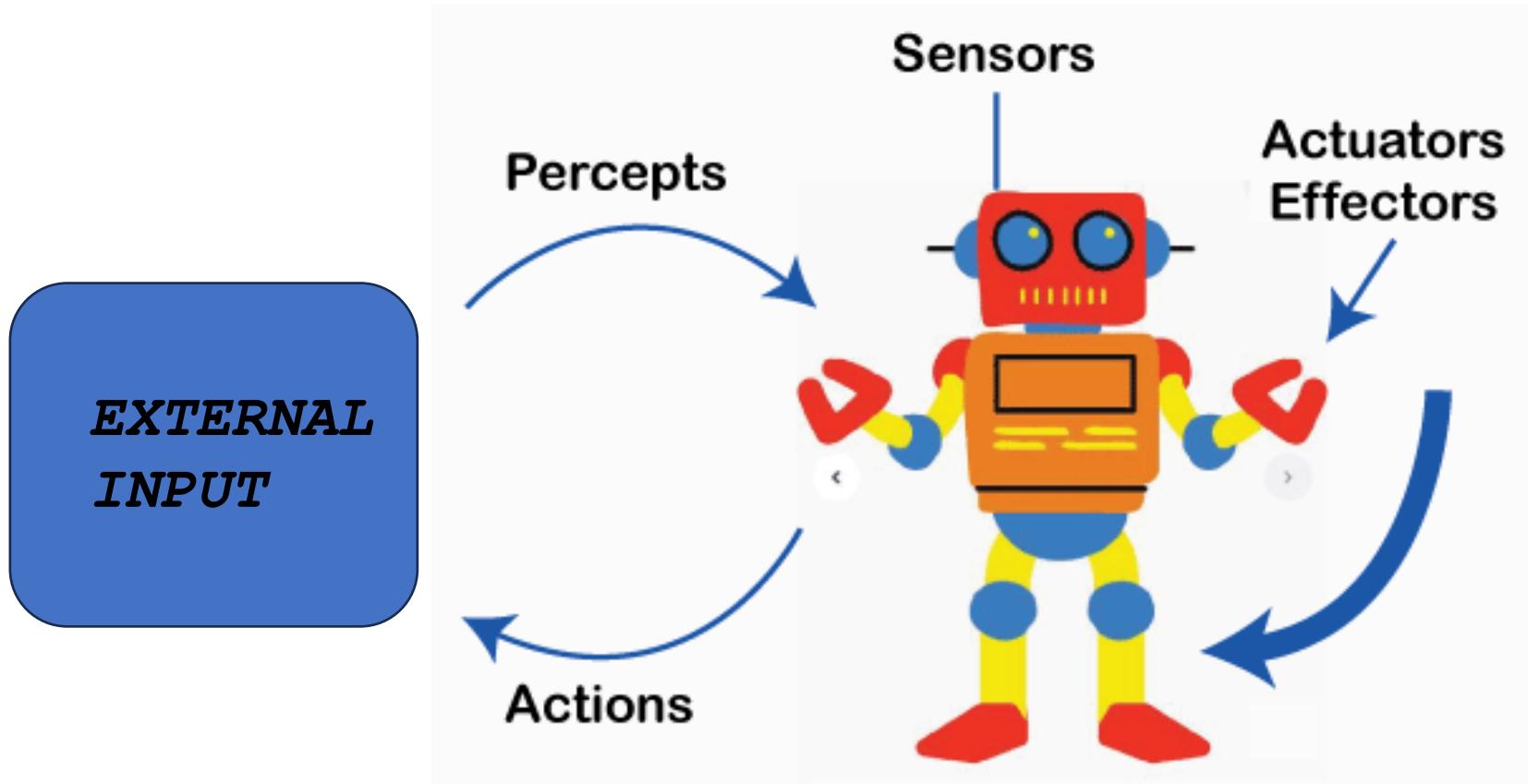


The AI research field clusters around seven main research areas.

FIGURE 2.2
 Keyword clusters and co-occurrences in the AI field, 2022;
 the color of the keyword represents its originating perspective:
 Teaching: orange, Research: blue, Industry: green, Media: pink,
 Multiple perspectives: black. Source: Scopus.



multimodal agent

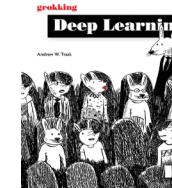


Generatieve AI according to Google

*Neurale netwerk modellen die machinaal hebben geleerd op basis van bestaande **multimodale content (trainen van het model)** nieuwe inhoud te creëren, zoals tekst, afbeeldingen, muziek en code.*

Generatieve AI zijn geen informatiedatabases of deterministische informatiezoeksysteem, omdat het voorspellingsystemen zijn.

What is machine learning?



“ A field of study that gives computers the ability to learn without being explicitly programmed.

—Attributed to Arthur Samuel

Given that deep learning is a subset of machine learning, what is machine learning? Most generally, it is what its name implies. Machine learning is a subfield of computer science wherein *machines learn* to perform tasks for which they were *not explicitly programmed*. In short, machines observe a pattern and attempt to imitate it in some way that can be either direct or indirect.

Machine learning \approx Monkey see, monkey do



Supervised machine learning

Supervised learning transforms datasets.

Supervised learning is a method for transforming one dataset into another. For example, if you had a dataset called Monday Stock Prices that recorded the price of every stock on every Monday for the past 10 years, and a second dataset called Tuesday Stock Prices recorded over the same time period, a supervised learning algorithm might try to use one to predict the other.



If you successfully trained the supervised machine learning algorithm on 10 years of Mondays and Tuesdays, then you could predict the stock price on any Tuesday in the future given the stock price on the immediately preceding Monday. I encourage you to stop and consider this for a moment.

Supervised machine learning is the bread and butter of applied artificial intelligence (also known as narrow AI). It's useful for taking *what you know* as input and quickly transforming it into *what you want to know*. This allows supervised machine learning algorithms to extend human intelligence and capabilities in a seemingly endless number of ways.

The majority of work using machine learning results in the training of a supervised classifier of some kind. Even unsupervised machine learning (which you'll learn more about in a moment) is typically done to aid in the development of an accurate supervised machine learning algorithm.



For the rest of this book, you'll be creating algorithms that can take input data that is observable, recordable, and, by extension, *knowable* and transform it into valuable output data that requires logical analysis. This is the power of supervised machine learning.

Unsupervised machine learning

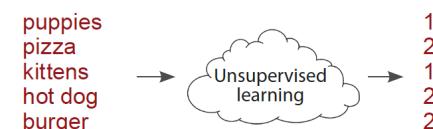
Unsupervised learning groups your data.

Unsupervised learning shares a property in common with supervised learning: it transforms one dataset into another. But the dataset that it transforms into is *not previously known or understood*. Unlike supervised learning, there is no “right answer” that you're trying to get the model to duplicate. You just tell an unsupervised algorithm to “find patterns in this data and tell me about them.”

For example, *clustering a dataset into groups* is a type of unsupervised learning. Clustering transforms a sequence of *datapoints* into a sequence of *cluster labels*. If it learns 10 clusters, it's common for these labels to be the numbers 1–10. Each datapoint will be assigned to a number based on which cluster it's in. Thus, the dataset turns from a bunch of datapoints into a bunch of labels. Why are the labels numbers? The algorithm doesn't tell you what the clusters are. How could it know? It just says, “Hey scientist! I found some structure. It looks like there are groups in your data. Here they are!”



I have good news! This idea of clustering is something you can reliably hold onto in your mind as the definition of unsupervised learning. Even though there are many forms of unsupervised learning, *all forms of unsupervised learning can be viewed as a form of clustering*. You'll discover more on this later in the book.

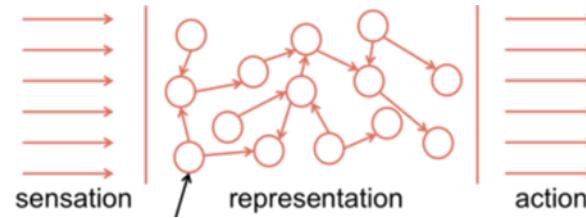


Check out this example. Even though the algorithm didn't tell what the clusters are named, can you figure out how it clustered the words? (Answer: 1 == cute and 2 == delicious.) Later, we'll unpack how other forms of unsupervised learning are also just a form of clustering and why these clusters are useful for supervised learning.

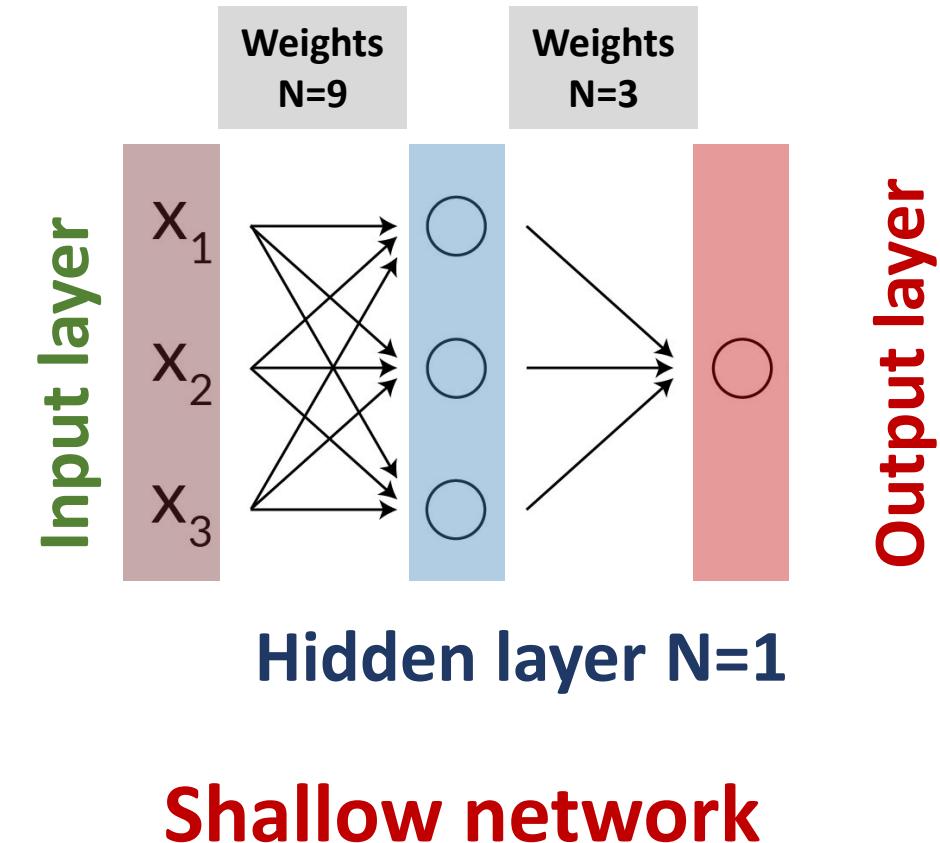
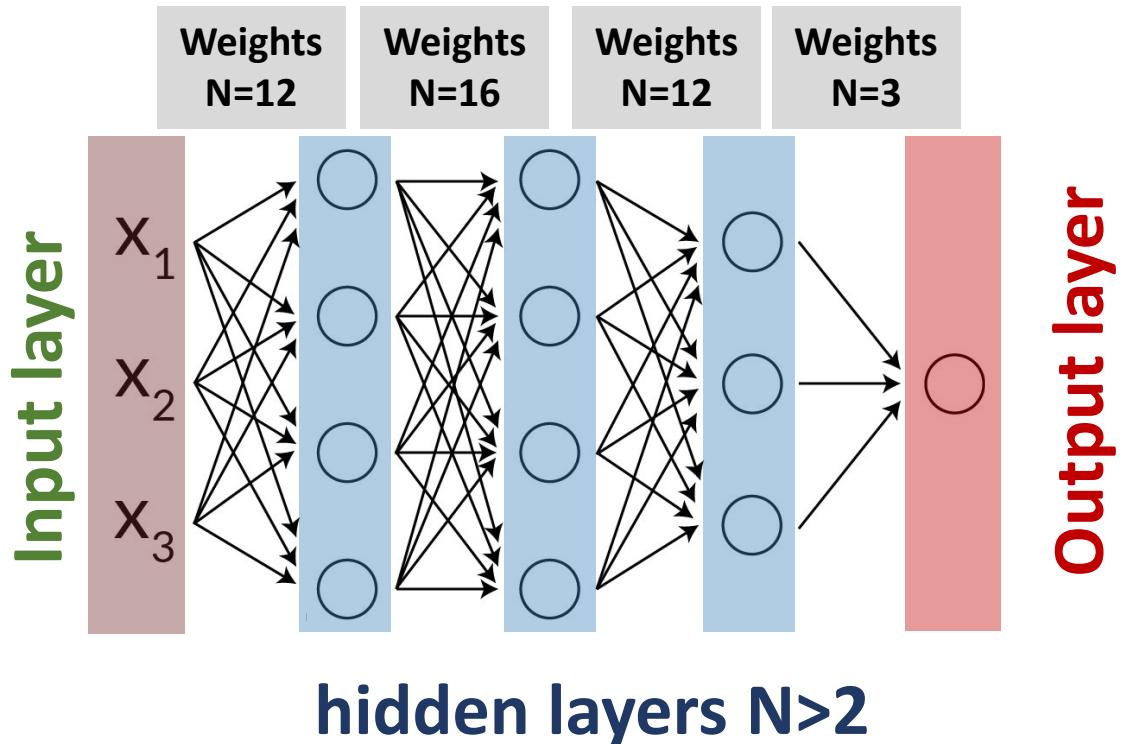


{NN Layers}

SUB-SYMBOLISCH



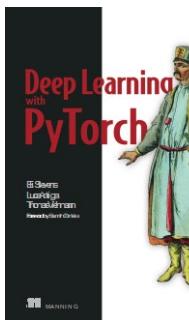
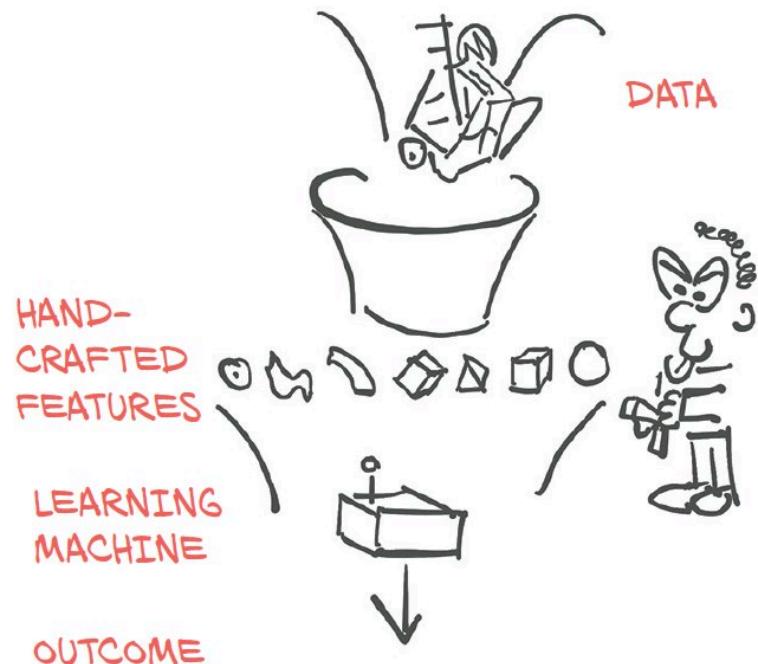
Neural Network {NN} Layer Architecture



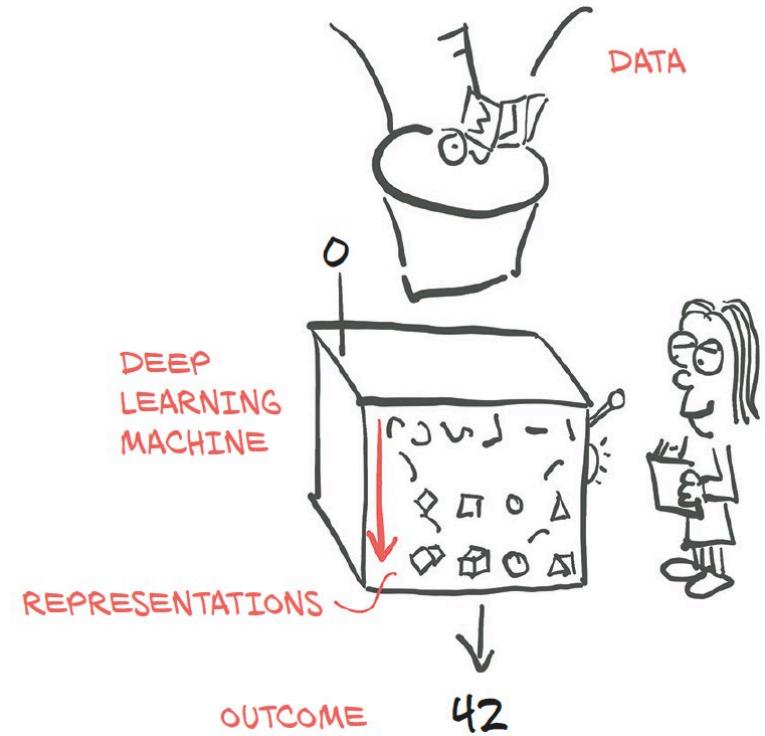
{AI Paradigm-shift}

More data, parameters & computing power | Less human-in-the-loop

Machine Learning Paradigm {ML}



Deep Learning Paradigm {DL}

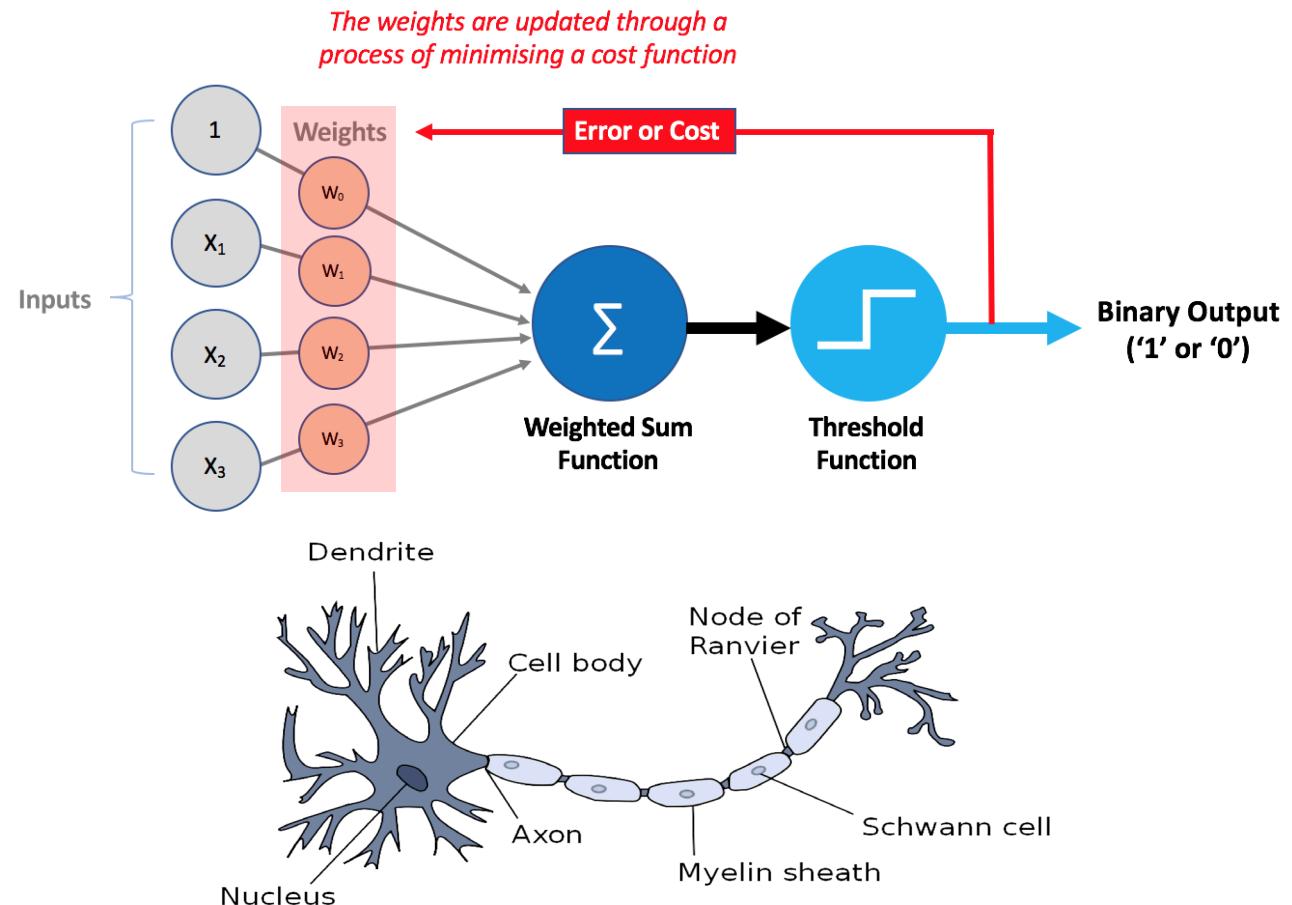


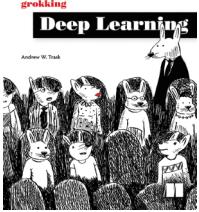
{Artificial Neurons}

Deep Neural Nets {DNNs} harbor vast amounts of
“artificial neurons” →smallest computational unit←

**Names for
 Artificial Neurons**

{unit}
{cell}
{node}
{perceptron}





Backpropagation in code

You can learn the amount that each weight contributes to the final error.

At the end of the previous chapter, I made an assertion that it would be important to memorize the two-layer neural network code so you could quickly and easily recall it when I reference more-advanced concepts. This is when that memorization matters.

The following listing is the new learning code, and it's essential that you recognize and understand the parts addressed in the previous chapters. If you get lost, go to chapter 5, memorize the code, and then come back. It will make a big difference someday.

```

import numpy as np
np.random.seed(1)

def relu(x):
    return (x > 0) * x
    >Returns x if x > 0;
    returns 0 otherwise

def relu2deriv(output):
    return output>0
    Returns 1 for input > 0;
    returns 0 otherwise

alpha = 0.2
hidden_size = 4

weights_0_1 = 2*np.random.random((3,hidden_size)) - 1
weights_1_2 = 2*np.random.random((hidden_size,1)) - 1

for iteration in range(60):
    layer_2_error = 0
    for i in range(len(streetlights)):
        layer_0 = streetlights[i:i+1]
        layer_1 = relu(np.dot(layer_0,weights_0_1))
        layer_2 = np.dot(layer_1,weights_1_2)

        layer_2_error += np.sum((layer_2 - walk_vs_stop[i:i+1]) ** 2)

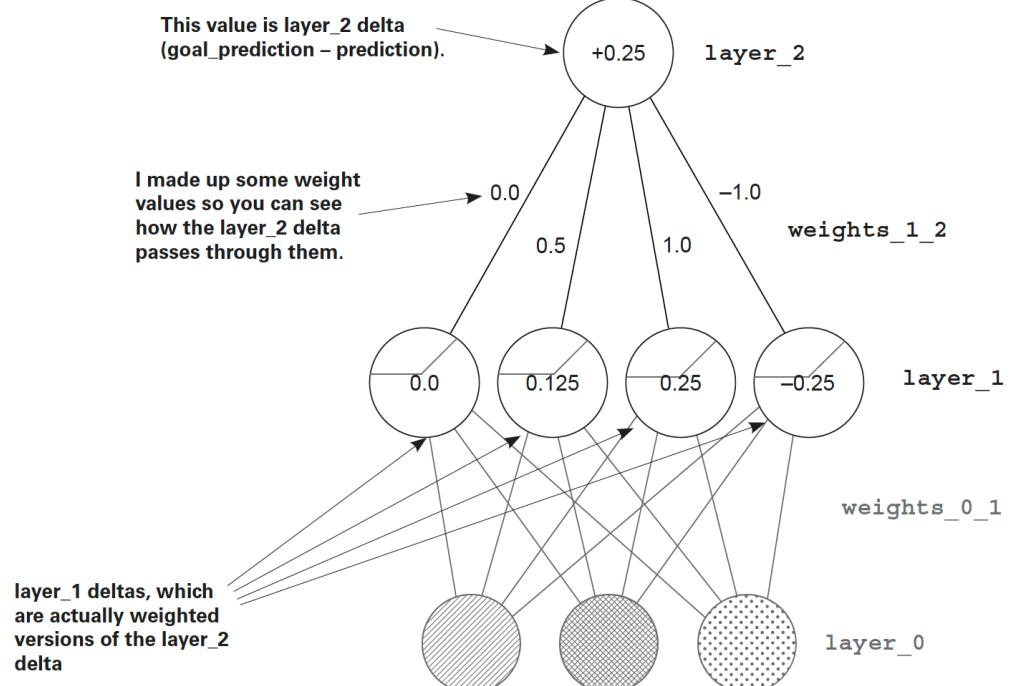
        layer_2_delta = (walk_vs_stop[i:i+1] - layer_2)
        This line computes the
        delta at layer_1 given
        the delta at layer_2
        by taking the layer_2_
        delta and multiplying
        it by its connecting
        weights_1_2.
        layer_1_delta=layer_2_delta.dot(weights_1_2.T)*relu2deriv(layer_1)

        weights_1_2 += alpha * layer_1.T.dot(layer_2_delta)
        weights_0_1 += alpha * layer_0.T.dot(layer_1_delta)

    if(iteration % 10 == 9):
        print("Error:" + str(layer_2_error))

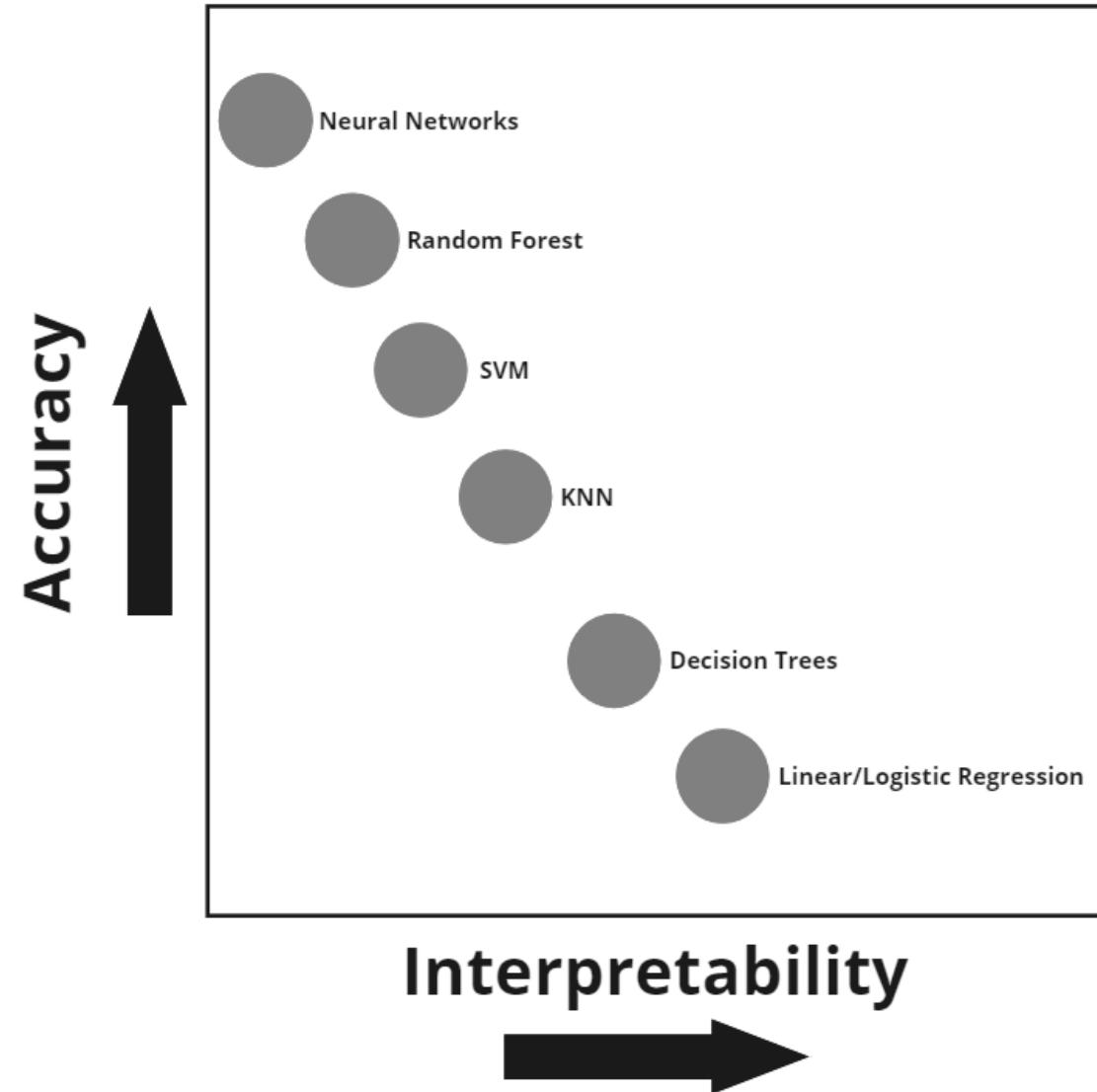
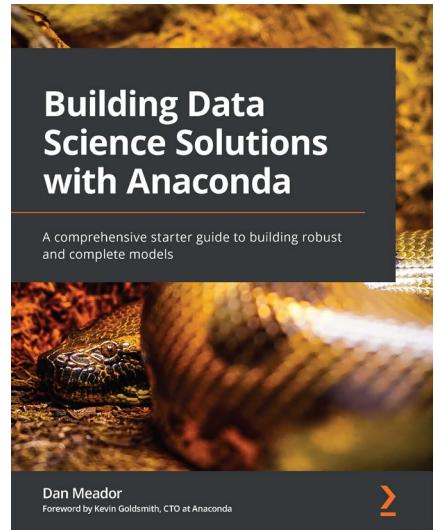
```

Believe it or not, the only truly new code is in bold. Everything else is fundamentally the same as in previous pages. The `relu2deriv` function returns 1 when `output > 0`; otherwise, it returns 0. This is the *slope* (the *derivative*) of the `relu` function. It serves an important purpose, as you'll see in a moment.



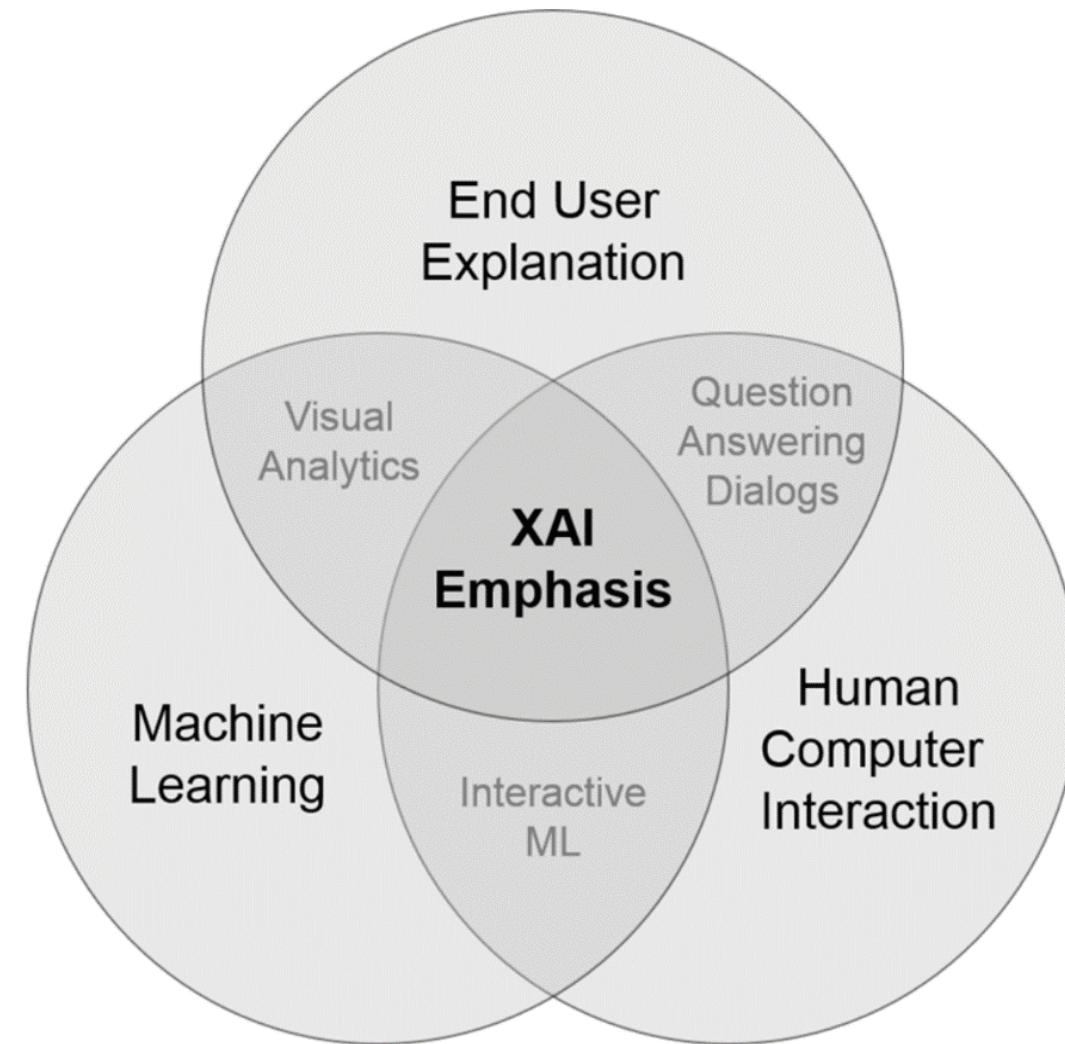
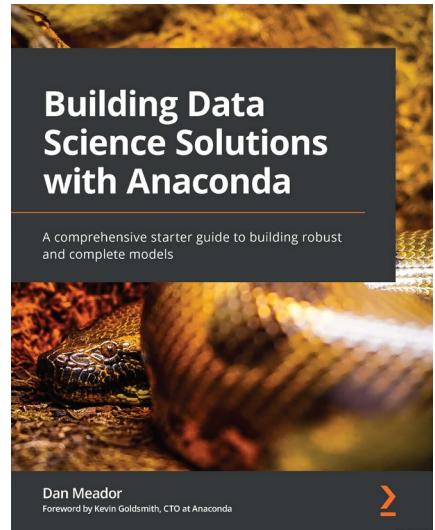


HOGESCHOOL
ROTTERDAM



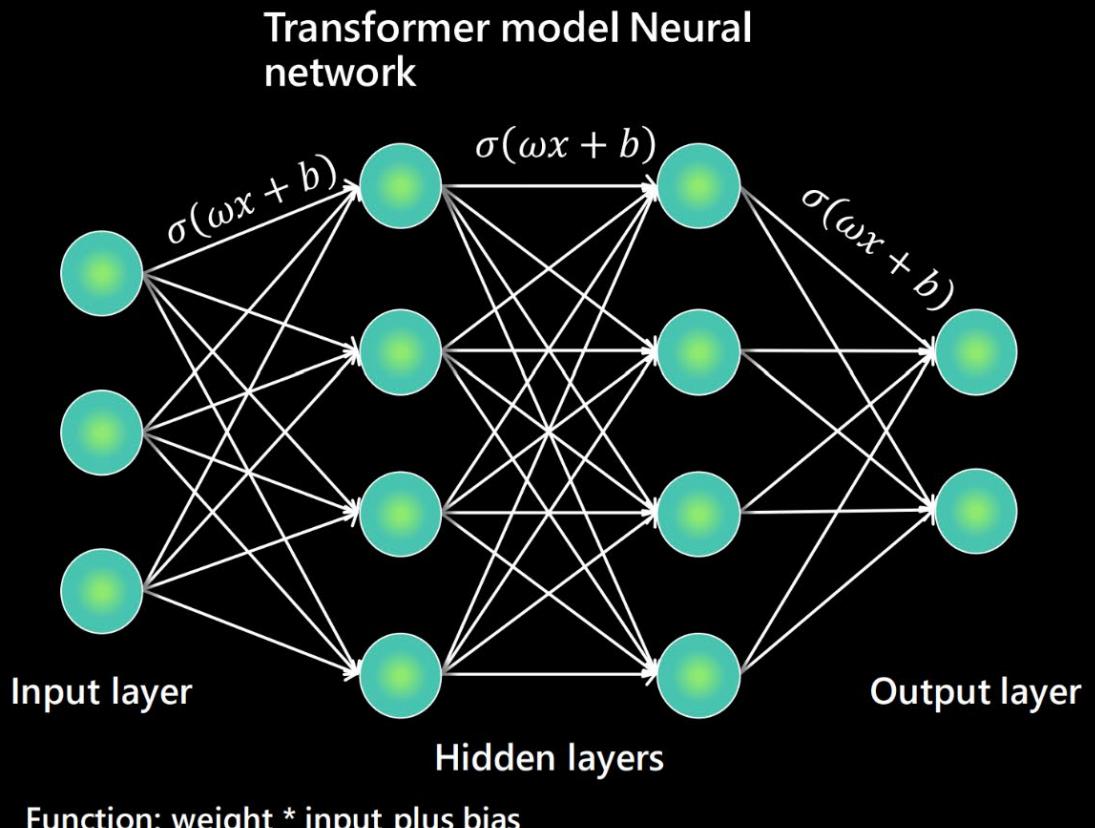


HOGESCHOOL
ROTTERDAM



Deep Neural Networks build itself

How large are they?



BERT Large - 2018

345M

GPT2 - 2019

1.5B

GPT3 - 2020

175B

Turing Megatron NLG
2021

530B

GPT4 – 2023

1.4T (estimated)

{Big-data}

Big-data is needed to avoid hand-crafted feature extraction

A Unified Approach to Interpreting Model Predictions

Scott M. Lundberg
 Paul G. Allen School of Computer Science
 University of Washington
 Seattle, WA 98105
 slundb@cs.washington.edu

Su-In Lee
 Paul G. Allen School of Computer Science
 Department of Genome Sciences
 University of Washington
 Seattle, WA 98105
 suinlee@cs.washington.edu

Abstract

Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy or efficiency. Transparency for large datasets is often achieved by complex models that even experts have trouble to interpret, such as ensemble or deep learning models, creating a tension between *accuracy* and *interpretability*. In response, various methods have recently been proposed to help users interpret the predictions of complex models, but it is often unclear how these methods are related and when one method is preferable over another. To address this problem, we present a unified framework for interpreting predictions, SHAP (SHapley Additive exPlanations). SHAP assigns each feature an importance value for a particular prediction. Its novel components include: (1) the identification of a new class of additive feature importance measures, and (2) theoretical results showing there is a unique solution in this class with a set of desirable properties. The new class unifies six existing methods, notable because several recent methods in the class lack the proposed desirable properties. Based on insights from this unification, we present new methods that show improved computational performance and/or better consistency with human intuition than previous approaches.

1 Introduction

The ability to correctly interpret a prediction model's output is extremely important. It engenders appropriate user trust, provides insight into how a model may be improved, and supports understanding of the process being modeled. In some applications, simple models (e.g., linear models) are often preferred for their ease of interpretation, even if they may be less accurate than complex ones. However, the growing availability of big data has increased the benefits of using complex models, so bringing to the forefront the trade-off between accuracy and interpretability of a model's output. A wide variety of different methods have been recently proposed to address this issue [5, 8, 9, 3, 4, 1]. But an understanding of how these methods relate and when one method is preferable to another is still lacking.

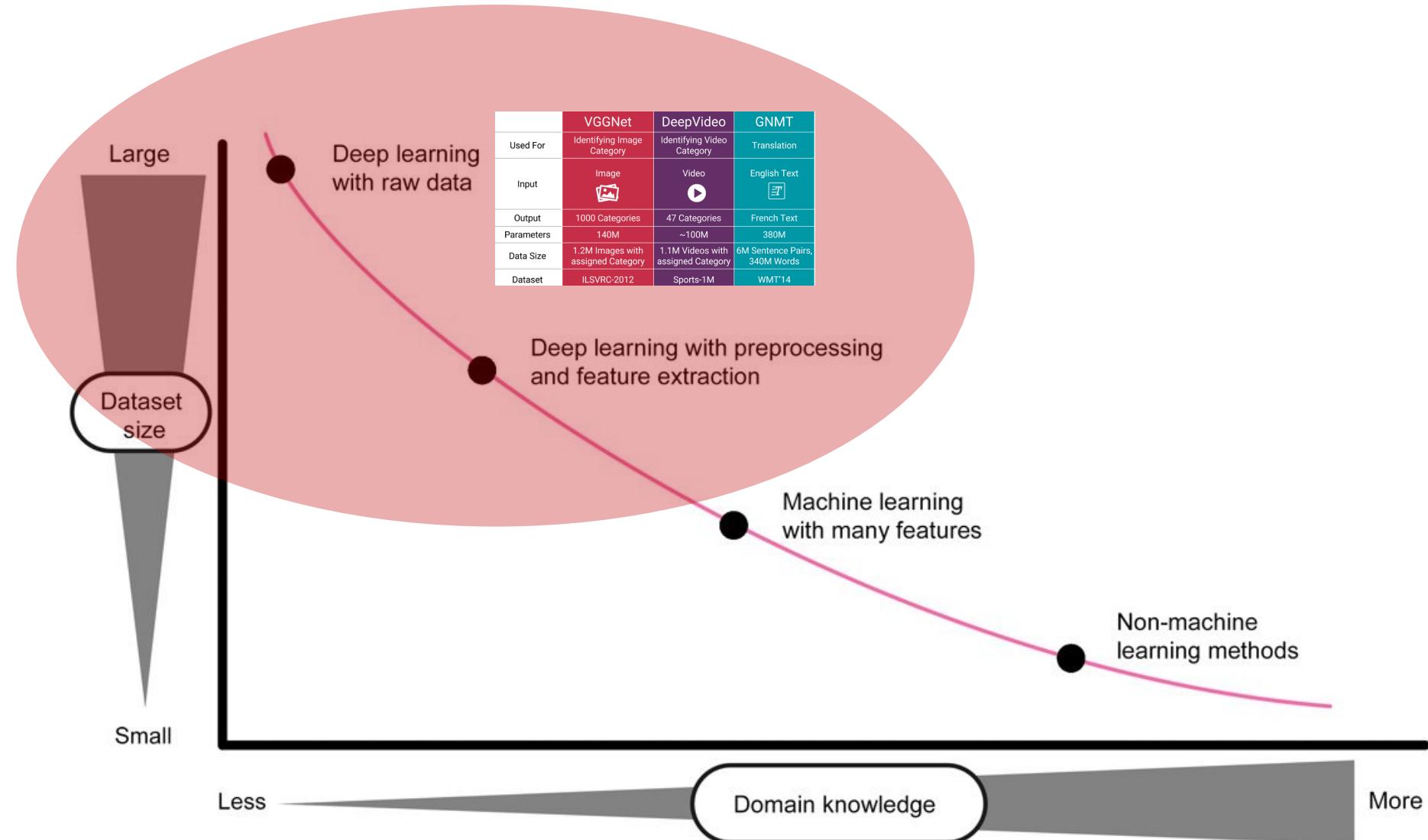
Here, we present a novel unified approach to interpreting model predictions.¹ Our approach leads to three potentially surprising results that bring clarity to the growing space of methods:

1. We introduce the perspective of viewing *any* explanation of a model's prediction as a model itself, which we term the *explanation model*. This lets us define the class of *additive feature attribution methods* (Section 2), which unifies six current methods.

¹<https://github.com/slundberg/shap>

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

<https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf>



{Top-down}

Top-down Encoding Capacity increases by adding hidden layers

What are the limits of deep learning?

The much-hyped artificial intelligence approach boasts impressive feats but still falls short of human brainpower. Researchers are determined to figure out what's missing.

M. Mitchell Waldrop, Science Writer

There's no mistaking the image: It's a banana—a big, ripe, bright-yellow banana. Yet the artificial intelligence (AI) identifies it as a toaster, even though it was trained with the same powerful and oft-publicized deep-learning techniques that have produced a white-hot revolution in driverless cars, speech understanding, and a multitude of other AI applications. That means the AI was shown several thousand photos of bananas, slugs, snails, and similar-looking objects, like so many flash cards, and then drilled on the answers until it had the classification down cold. And yet this advanced system was quite easily confused—all it took was a little day-glow sticker, digitally pasted in one corner of the image.

This example of what deep-learning researchers call an "adversarial attack," discovered by the Google Brain team in Mountain View, CA (1), highlights just how far AI still has to go before it remotely approaches human capabilities. "I initially thought that adversarial examples were just an annoyance," says Geoffrey Hinton, a computer scientist at the University of Toronto and one of the pioneers of deep learning. "But I now think they're probably quite profound. They tell us that we're doing something wrong."

That's a widely shared sentiment among AI practitioners, any of whom can easily rattle off a long list of deep learning's drawbacks. In addition to its vulnerability

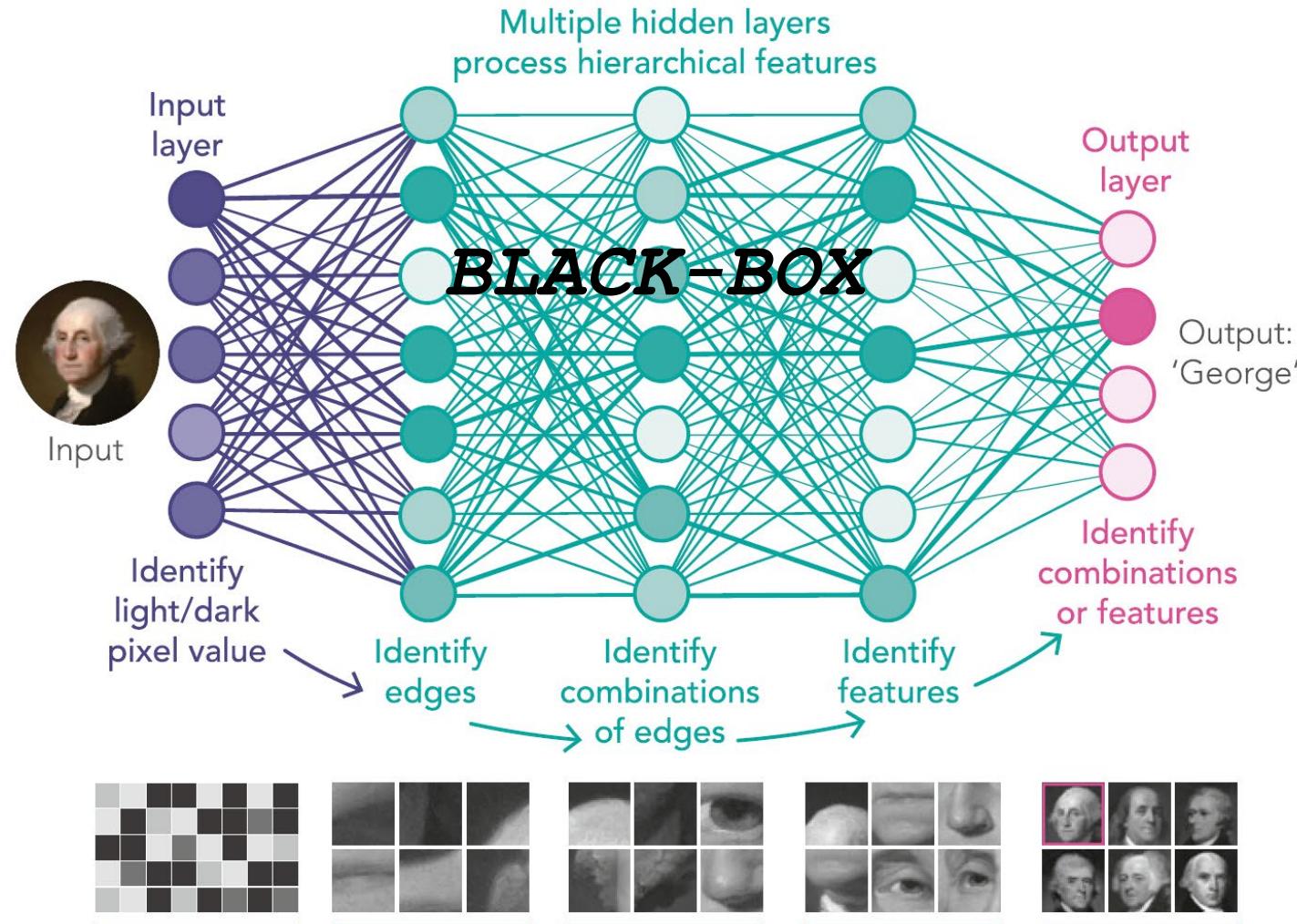


Apparent shortcomings in deep-learning approaches have raised concerns among researchers and the general public as technologies such as driverless cars, which use deep-learning techniques to navigate, get involved in well-publicized mishaps. Image credit: Shutterstock.com/MONOPOLY919.

Published under the PNAS license.

January 22, 2019 | vol. 116 | no. 4

www.pnas.org/cgi/doi/10.1073/pnas.1821594116

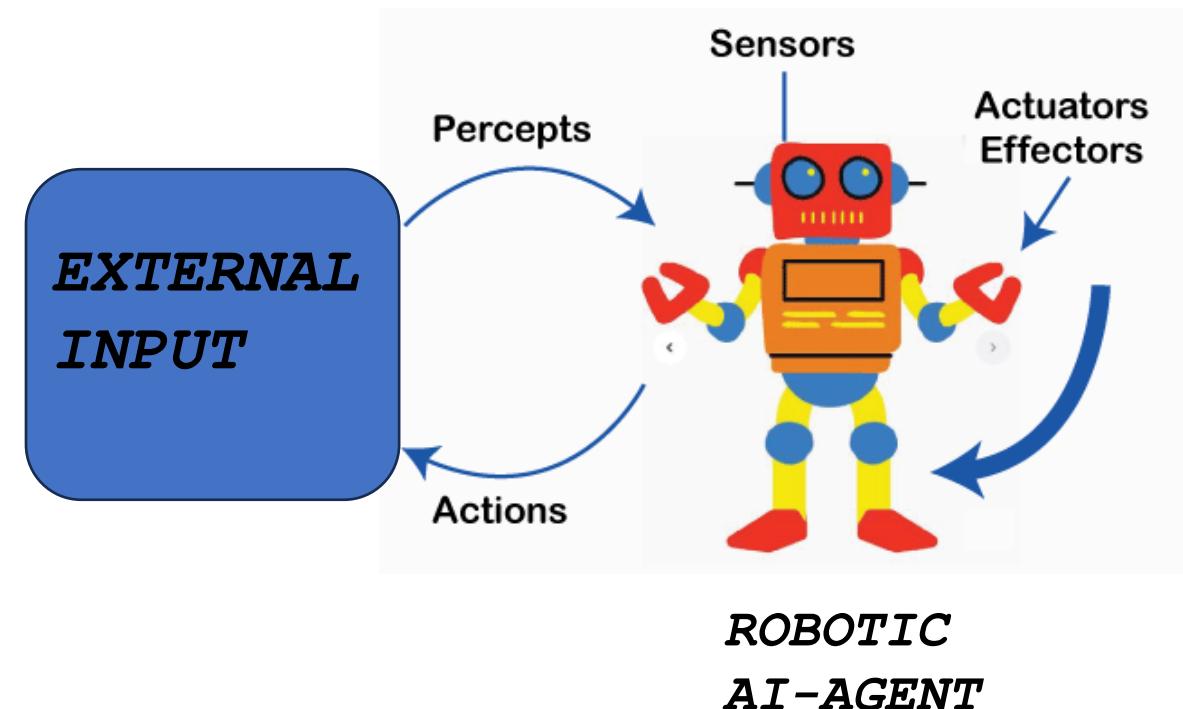
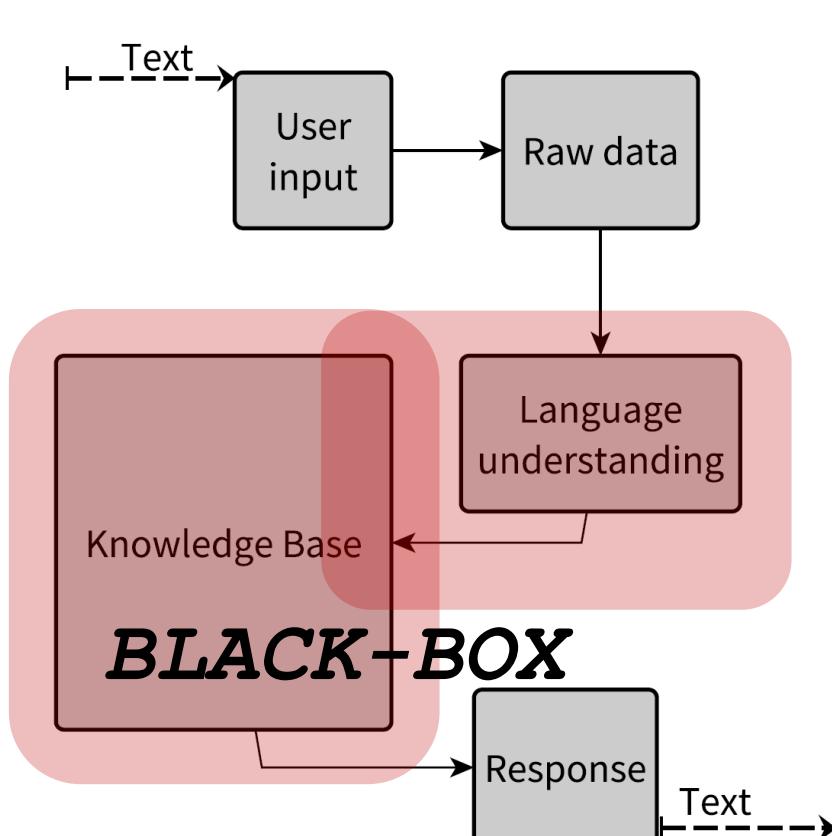


MACHINE LEARNING data-flow

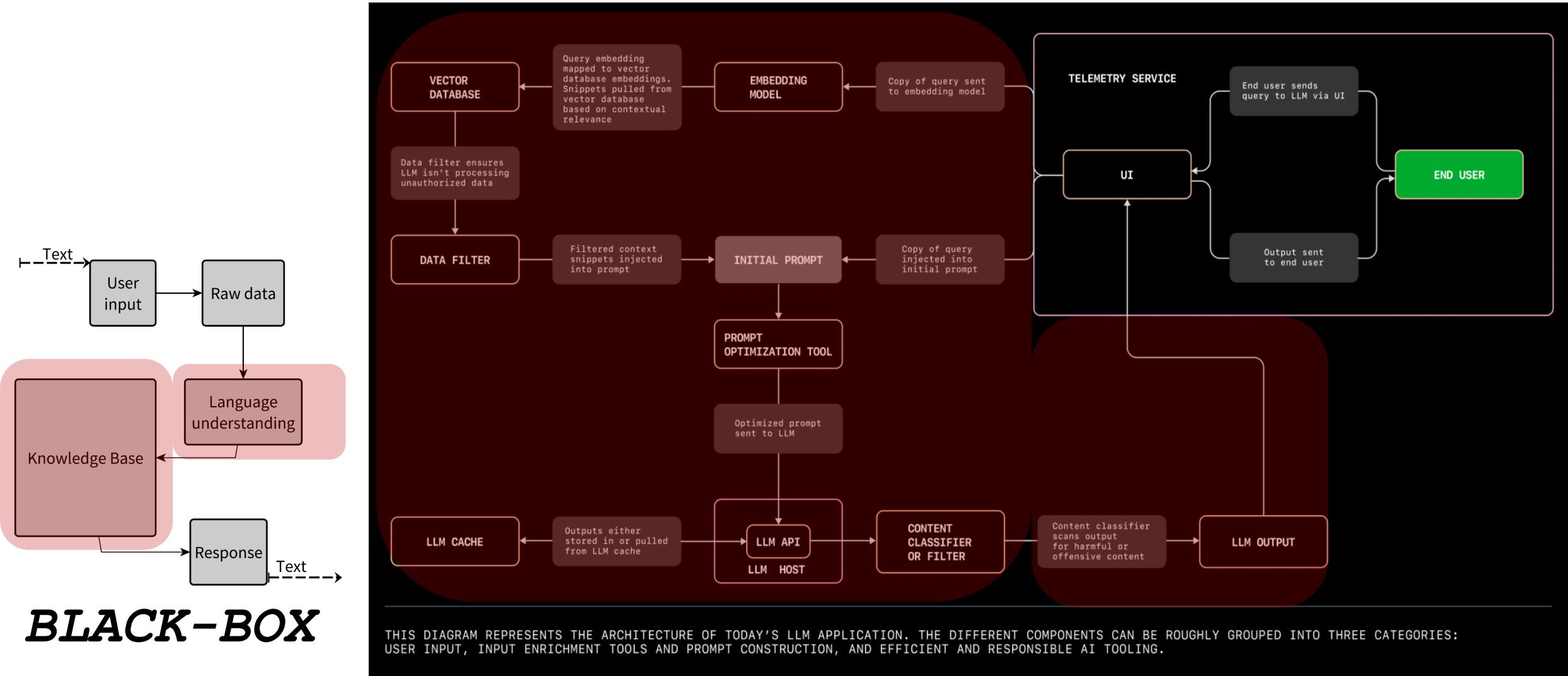


DEEP LEARNING data-flow

ChatGPT is een Conversationele tekst-in/tekst-uit ChatBot



ChatGPT == 99% BLACKBOX + 1% user-interface



GPT {LLM} versus ChatGPT

GPT vs. ChatGPT: The Difference



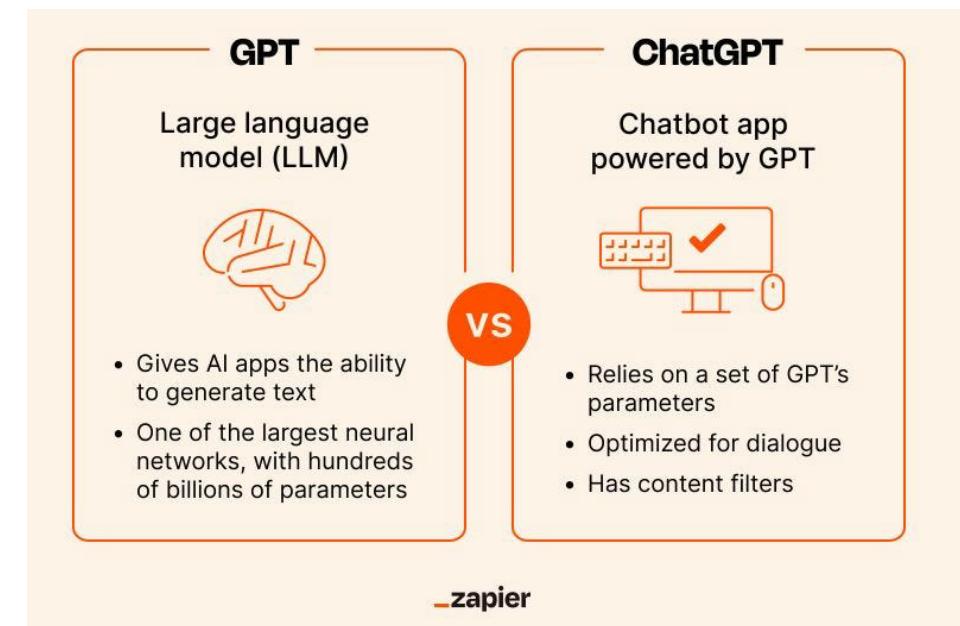
GPT*

- A series of large language models (LLM)
- Generates content for various applications

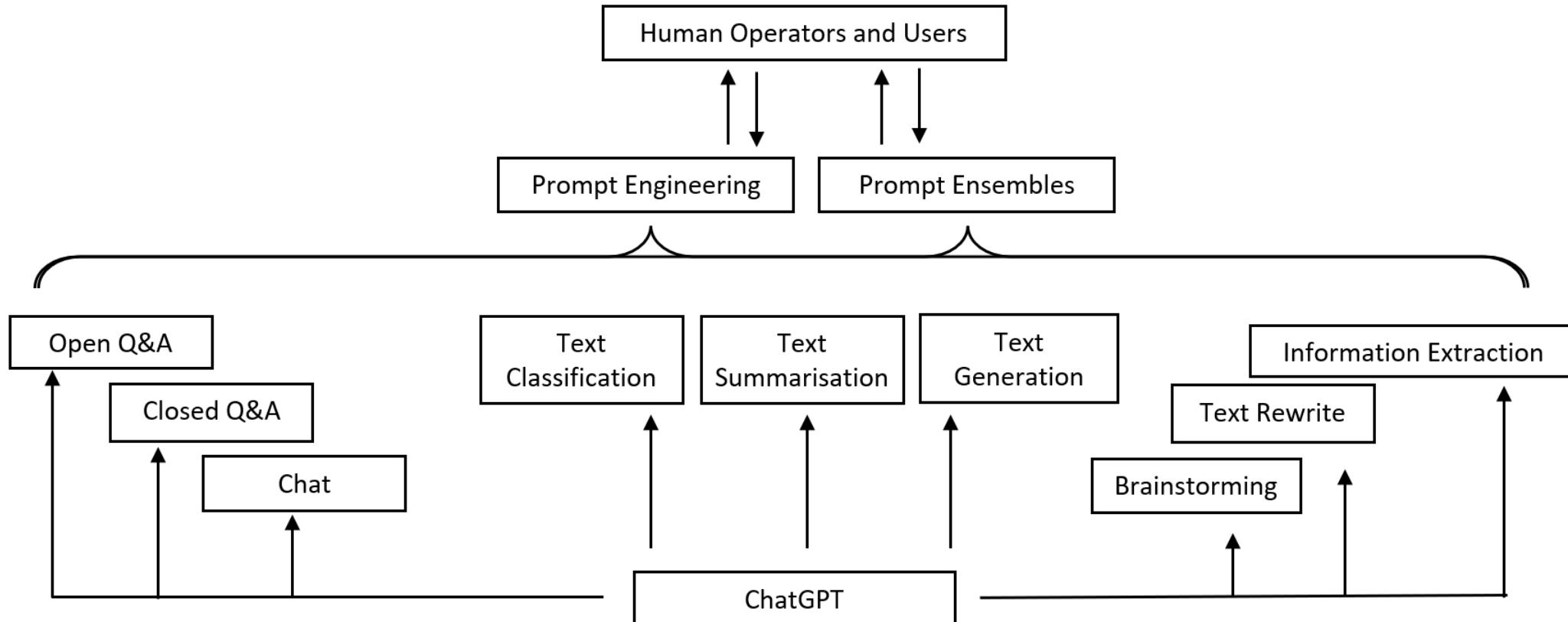
*GPT = Generative Pre-trained Transformers

ChatGPT

- An AI-powered chatbot application
- Utilizes GPT language models for human-like interactions



ChatBot Use-Cases



Conferences > 2023 IEEE International Confe... ⓘ

ChatGPT and Generative AI Guidelines for Addressing Academic Integrity and Augmenting Pre-Existing Chatbots

Publisher: IEEE

[Cite This](#)



Daswin De Silva ; Nishan Mills ; Mona El-Ayoubi ; Milos Manic ; Damminda Alahakoon [All Authors](#)

635
Full
Text Views



generieke train-dataset
machinaal leren
Black-Box

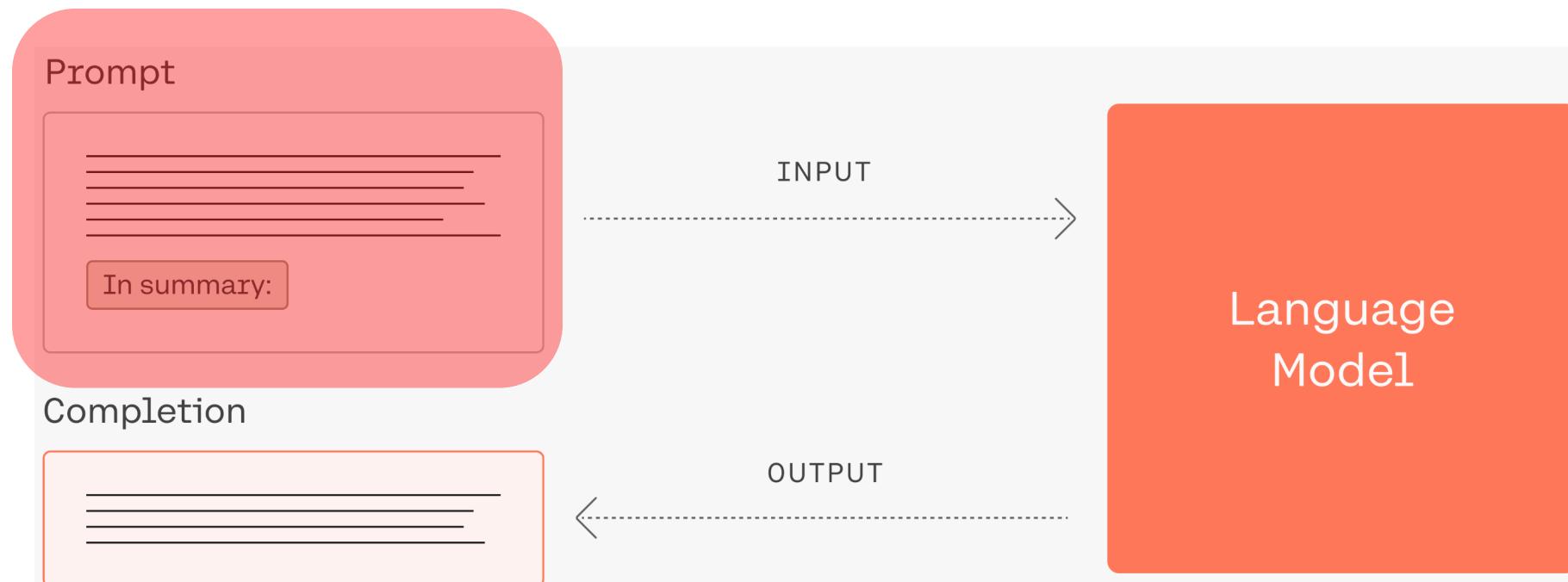
GEEN Human-in-the-loop
multimodaliteit
Commerciële belangen

creëert risico's
veiligheid
Privacy
betrouwbaarheid &
reproduceerbaarheid

*Hoe kun je een chatbot
Dialoog effectief &
unbiased sturen*

*Met behulp van
prompt engineering
kun je de output van LLMs
Betroubaarder maken*

*Conversationele AI-agenten worden aangestuurd via “**prompts**”*



<https://docs.cohere.com/docs/prompt-engineering>



<https://docs.cohere.com/docs/introduction-to-large-language-models>

PROMPT-ENGINEERING

Het creatieve proces van het schrijven van een effectief ***prompt-recept*** wordt in het Engels "***prompt engineering***" genoemd.

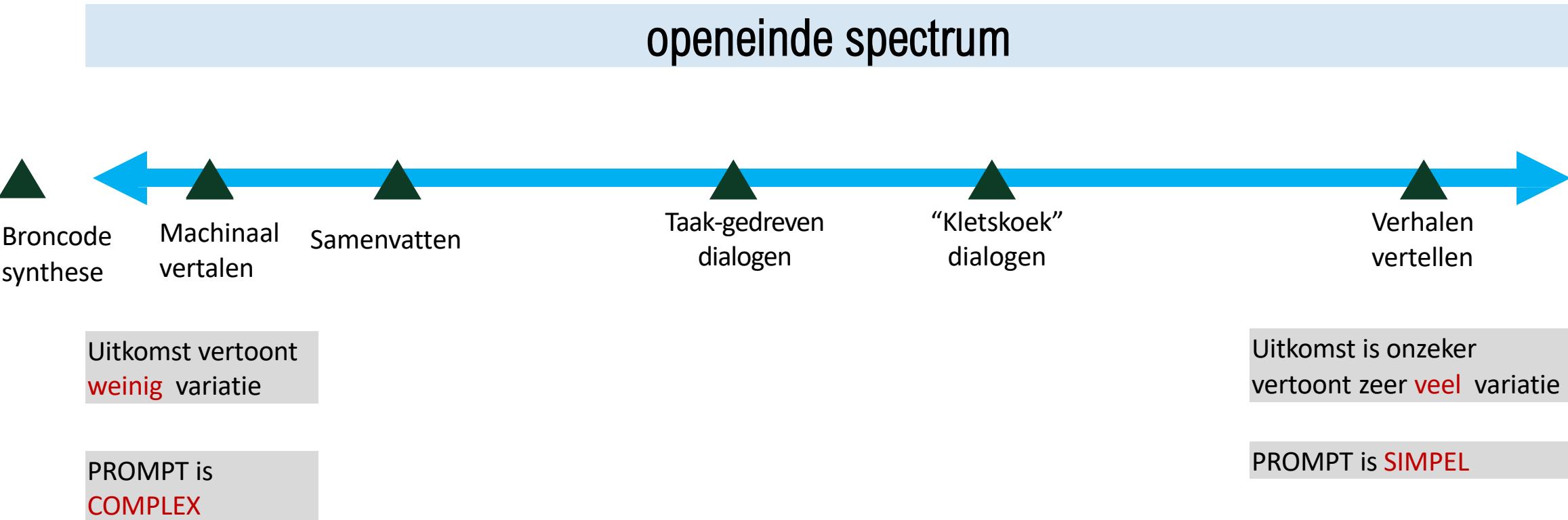
Het schrijven van prompt-recepten

---pseudo-Code---

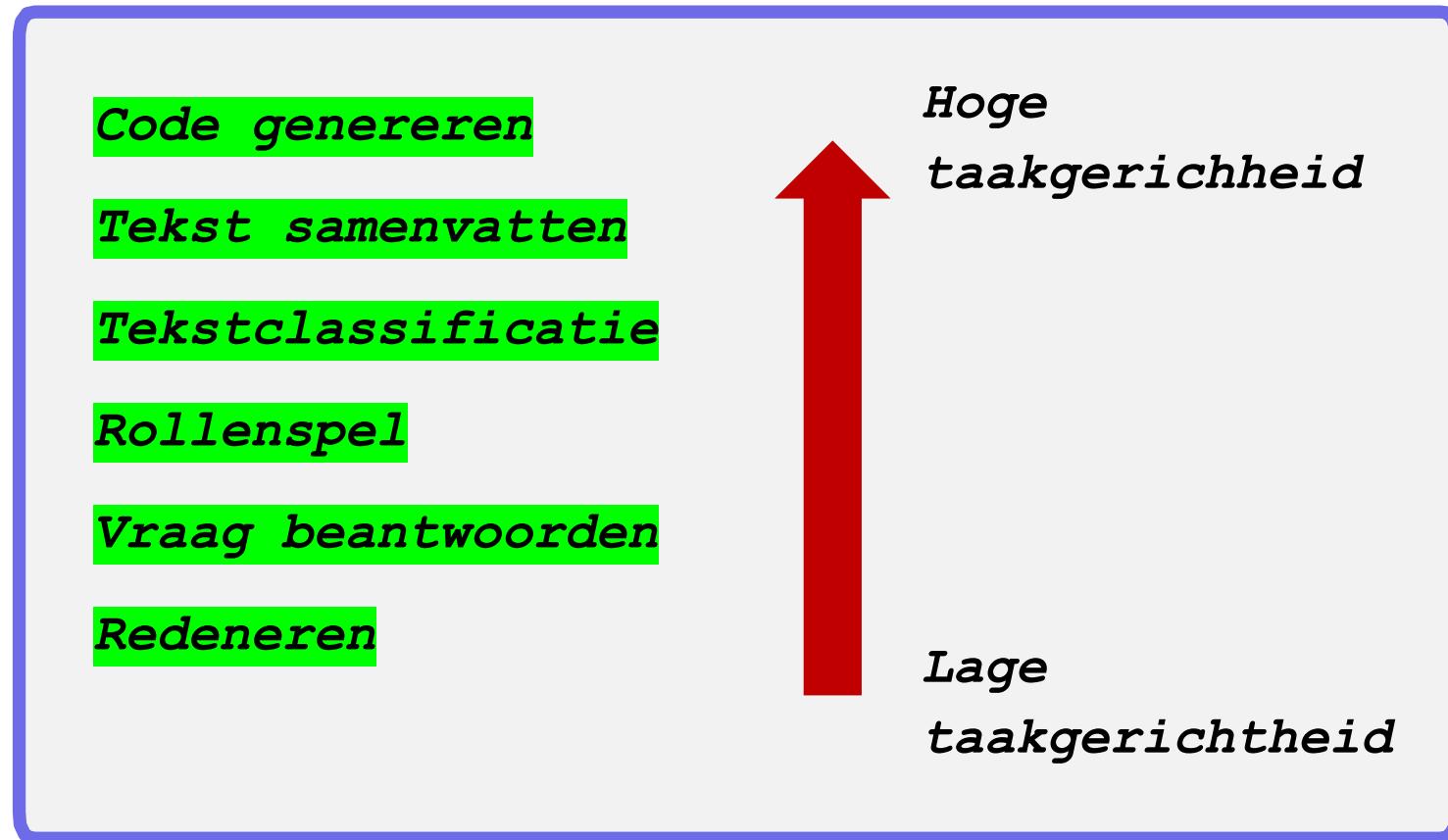
is een talige manier van het programmeren van "bevroren" voorgetraind taalmodellen.



Prompt Taxonomie



Taakgerichtheid van prompts



Wat is het belang van Prompt Recepten Schrijven?

*Sturen van de mate van taakgerichtheid door
reduceren van variatie in het antwoord zodat de
kans groter wordt dat de uitkomst correct is.*

Prompt Recept Structuur

Een prompt is opgebouwd uit de volgende elementen:

Instructie(s)

Context

Invoergegevens

Uitvoer-indicator

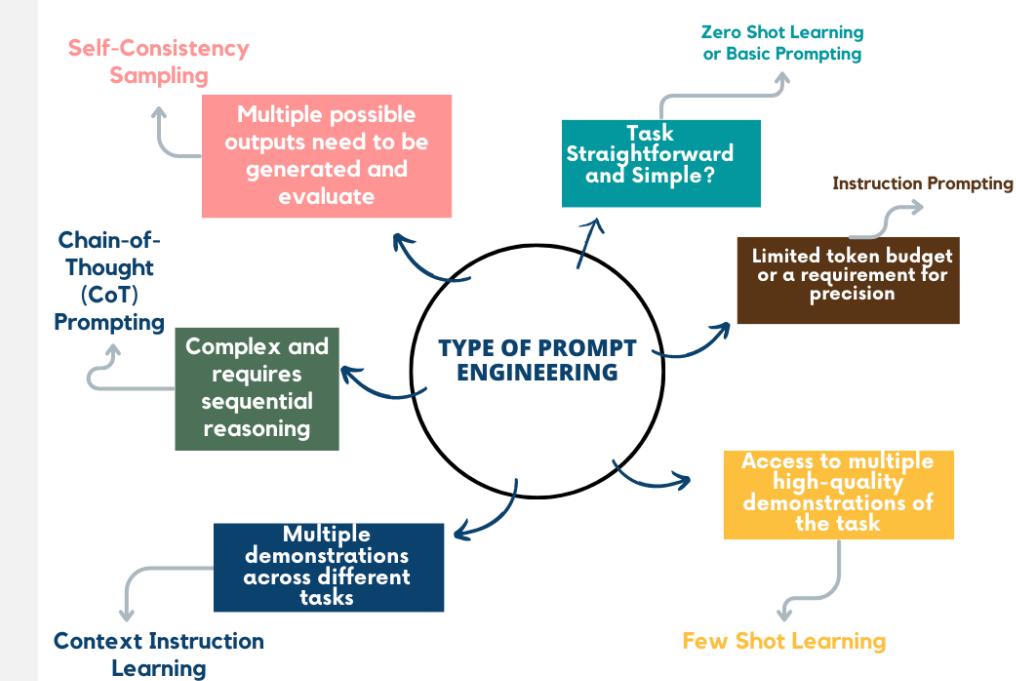
Classificeer de onderstaande tekst als neuraal, negatief of positief

Text: Ik vond het eten wel zoso.

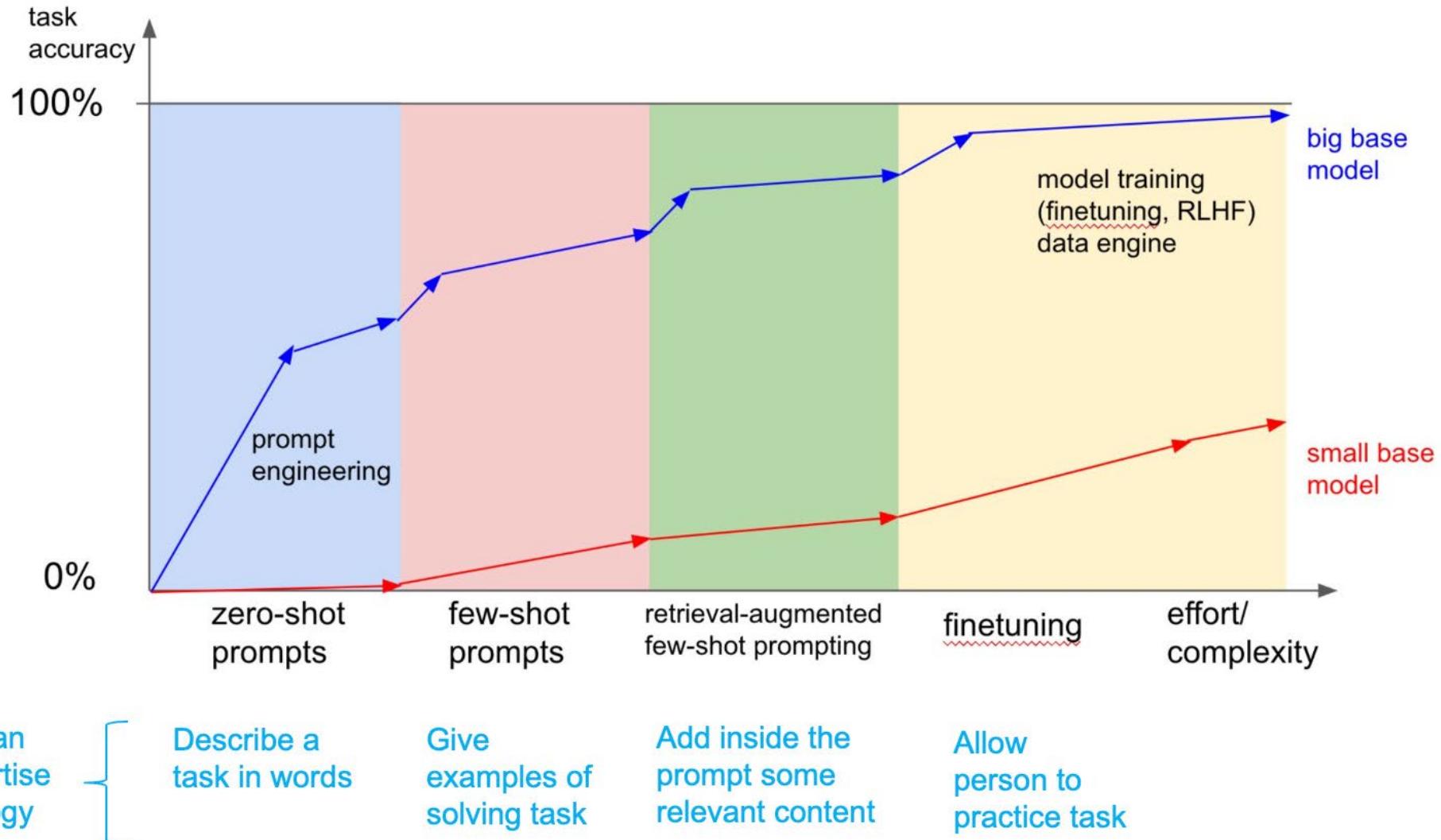
Sentiment:

Prompt recept ontwerptechnieken gebaseerd op fine-tuning van het onderliggende taal-model

Few-shot prompts (**In Context Leren**)
Chain-of-thought (**CoT**) reasoning
Self-Consistency Sampling
Knowledge Generation Prompting
ReAct



Building a PoC dialogue



*Hoe bouw
en test je veilig
**Talige Generatieve
AI-technologie?***

Azure OpenAI service

OVERVIEW

Build intelligent apps with AI models

Cutting edge models ▾

Quickly develop generative AI experiences with a diverse set of prebuilt and curated models from OpenAI, Meta and beyond.

[Try the Azure AI Studio](#)

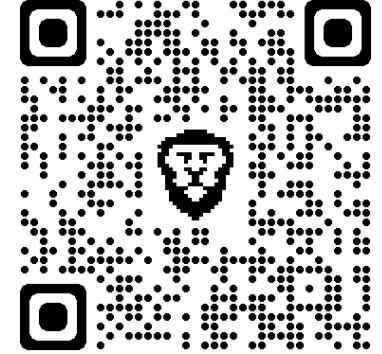
Data grounding ▾

Trust and transparency ▾

Data, privacy and security ▾

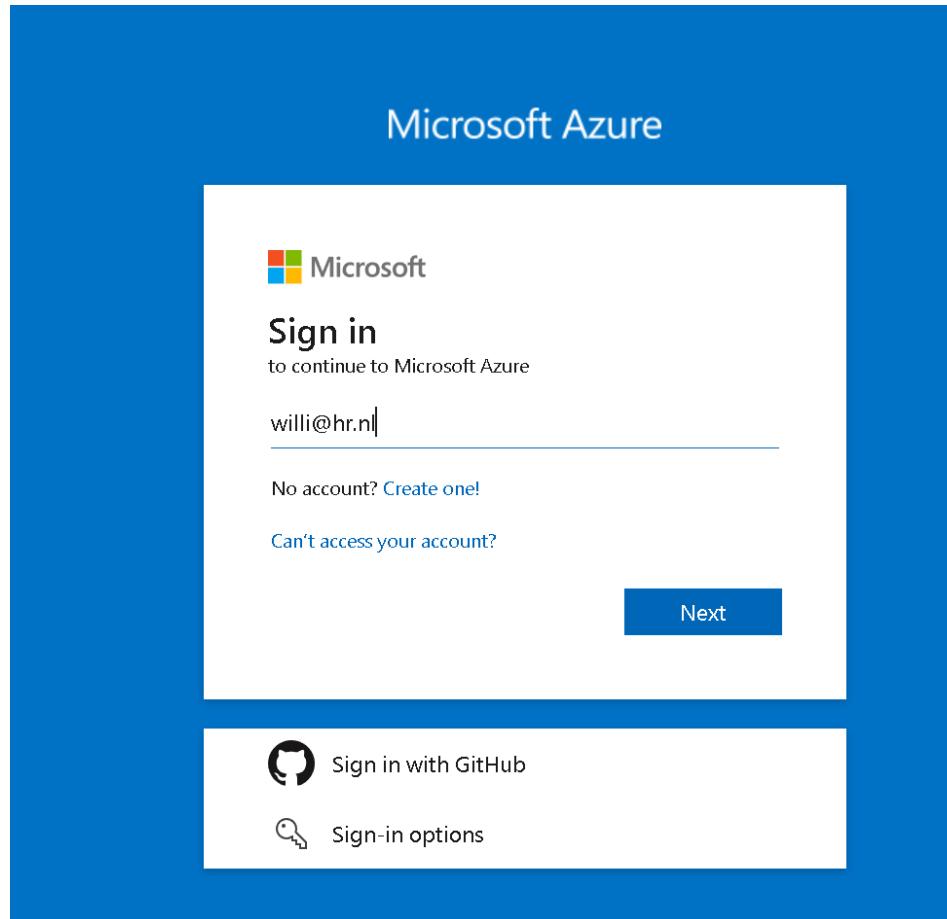
USE CASES

Apply generative AI to a variety of use cases



<https://azure.microsoft.com/en-us/products/ai-services/openai-service>

portal.azure.com



Log in

Student number or personnel code

Password

Login



HOGESCHOOL
ROTTERDAM

Azure services



Resources

Recent Favorite

Name	Type	Last Viewed
GPT4-SWEDEN-GROUP	Azure OpenAI	6 days ago
Azure for Students	Subscription	3 weeks ago
CHATBOT02	Azure OpenAI	3 weeks ago
DefaultResourceGroup-westeurope	Resource group	2 months ago
AV07	Speech service	7 months ago
NLP	Resource group	7 months ago
LLM01	Language understanding	7 months ago
Visual Studio Professional Subscription	Subscription	7 months ago
LLM01-Authoring	Language understanding	7 months ago
WILLI107	Resource group	7 months ago
cursusa1-900	Azure Machine Learning workspace	8 months ago
cursusa9006361709869	Key vault	8 months ago

See all

Navigate



Tools



Useful links

Azure mobile app

Create Azure OpenAI ...

Azure services

[Create a resource](#)[Azure OpenAI](#)[Subscriptions](#)[Resource groups](#)[Education](#)[Cost Management](#)[All resources](#)[Azure AI services](#)[App Services](#)[More services](#)

Resources

[Recent](#) [Favorite](#)

Name	Type	Last Viewed
GPT4-SWEDEN-GROUP	Azure OpenAI	8 minutes ago
cursusai9006361709869	Key vault	9 minutes ago
Azure for Students	Subscription	9 minutes ago
Visual Studio Professional Subscription	Subscription	10 minutes ago
NLP	Resource group	11 minutes ago
CHATBOT02	Azure OpenAI	3 weeks ago
DefaultResourceGroup-westeurope	Resource group	2 months ago
AV07	Speech service	7 months ago
LLM01	Language understanding	7 months ago
LLM01-Authoring	Language understanding	7 months ago
WILLI107	Resource group	7 months ago
cursusai-900	Azure Machine Learning workspace	8 months ago

[See all](#)[Review the Azure OpenAI code of conduct](#)

Create and deploy an Azure OpenAI Service resource

Article • 09/06/2023 • 4 contributors

Feedback

Choose your preferred resource creation method

Portal **CLI** PowerShell

In this article

[Prerequisites](#)[Create a resource](#)[Deploy a model](#)[Next steps](#)

This article describes how to get started with Azure OpenAI Service and provides step-by-step instructions to create a resource and deploy a model. You can create resources in Azure in several different ways:

- The [Azure portal](#)
- The REST APIs, the Azure CLI, PowerShell, or client libraries
- Azure Resource Manager (ARM) templates

In this article, you review examples for creating and deploying resources in the Azure portal and with the Azure CLI.

Prerequisites

- An Azure subscription. [Create one for free](#).
- Access granted to Azure OpenAI in the desired Azure subscription.
- Access permissions to [create Azure OpenAI resources](#) and to [deploy models](#).

Note

Currently, you must submit an application to access Azure OpenAI Service. To apply for access, complete [this form](#). If you need assistance, open an issue on this repository to contact Microsoft.



Request Access to Azure OpenAI Service

* Required

Please read all instructions carefully and complete form as instructed

Thank you for your interest in Azure OpenAI Service. Please submit this form to register for approval to access and use Azure OpenAI's Limited Access text and code and/or DALL-E 2 text to image models (as indicated in the form). All use cases must be registered. Azure OpenAI Service requires registration and is currently only available to approved enterprise customers and partners. Learn more about limited access to Azure OpenAI Service [here](#).

Limited access scenarios: When evaluating which scenarios to onboard, we consider who will directly interact with the application, who will see the output of the application, whether the application will be used in a high-stakes domain (e.g., medical), and the extent to which the application's capabilities are tightly scoped. In general, applications in high stakes domains will require additional mitigations and are more likely to be approved for applications with internal-only users and internal-only audiences. Applications with broad possible uses, including content generation capabilities, are more likely to be approved if 1) the domain is not high stakes and users are authenticated or 2) in the case of high stakes domains, anyone who views or interacts with the content is internal to your company.

Please be sure to visit the [Azure OpenAI Service's transparency note](#), which provides information and guidelines for responsible use of the service as well as system limitations that may be applicable to your scenario.

If you are a current Azure OpenAI customer and would like to add additional use cases, please fill out the [Azure OpenAI Additional Use Case form](#)



https://customervoice.microsoft.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB7en2Ais5pxKtso_Pz4b1_xUOFA5Qk1UWDRBMjg0WFhPMkIzTzhKQ1dWNyQIQCN0PWcu

Azure AI | Azure OpenAI Studio

« Azure AI Studio > Chat playground

Chat playground

Assistant setup

System message Add your data (preview)

Save changes

Specify how the chat should act

Use a template to get started, or just start writing your own system message below. Want some tips? [Learn more](#)

Use a system message template

Select a template

System message ⓘ

You are an AI assistant that helps people find information.

Examples ⓘ

+ Add an example

Sample Code

You can use the following code to start integrating your current prompt and settings into your application

<https://gpt4-sweden-group.openai.azure.com/> python

```
1 #Note: The openai-python library support for Azure OpenAI is in preview.
2 import os
3 import openai
4 openai.api_type = "azure"
5 openai.api_base = "https://gpt4-sweden-group.openai.azure.com/"
6 openai.api_version = "2023-07-01-preview"
7 openai.api_key = os.getenv("OPENAI_API_KEY")
8
9 response = openai.chatcompletion.create(
10     engine="GPT4-32K",
11     messages = [{"role": "system", "content": "You are an AI
assistant that helps people find information."},
12 {"role": "user", "content": "A neutron star is the collapsed core of a
massive supergiant star, which had a total mass of between 10 and 25
solar masses, possibly more if the star was especially metal-rich.
Neutron stars are the smallest and densest stellar objects, excluding
black holes and hypothetical white holes, quark stars, and strange
stars. Neutron stars have a radius on the order of 10 kilometres (6.2
mi) and a mass of about 1.4 solar masses. They result from the
supernova explosion of a massive star, combined with gravitational
collapse, that compresses the core past white dwarf star density to
that of atomic nuclei.\n\nQ: How are neutron stars created?\nA:"}],
13     "role": "assistant", "content": "Neutron stars are created from the
supernova explosion of a massive star, combined with gravitational
collapse, that compresses the core past white dwarf star density to
that of atomic nuclei.\n\nQ: How are neutron stars created?\nA:"}]
```

Endpoint ⓘ

<https://gpt4-sweden-group.openai.azure.com/openai/deployments/GPT4-3...>

Key ⓘ

.....

You should use environment variables or a secret management tool like Azure Key Vault to prevent accidental exposure of your key in applications. [Learn more](#)

Copy Close

Azure chat completions example (preview)

In this example we'll try to go over all operations needed to get chat completions working using the Azure endpoints.

This example focuses on chat completions but also touches on some other operations that are also available using the API. This example is meant to be a quick way of showing simple operations and is not meant as a tutorial.

```
 1 import os
 2 import openai
 3 openai.api_type = "azure"
 4 openai.api_base = "https://gpt4-sweden-group.openai.azure.com/"
 5 openai.api_version = "2023-07-01-preview"
 6 openai.api_key = "ded218c778894f6da4d3c595c6904194"
 7
 8
 9 #!setx AZURE_OPENAI_KEY "ded218c778894f6da4d3c595c6904194"
10 #!setx AZURE_OPENAI_ENDPOINT "https://gpt4-sweden-group.openai.azure.com/openai/deployments/gpt4-32k/chat/completions?api-version=2023-07-01-preview"
11
12 response = openai.chatCompletion.create(
13     engine="GPT4-32K",
14     messages = [
15         {"role": "system", "content": "You are a helpful assistant."},
16         {"role": "user", "content": "Does Azure OpenAI support customer managed keys?"},
17         {"role": "assistant", "content": "Yes, customer managed keys are supported by Azure OpenAI."},
18         {"role": "user", "content": "Do other Azure AI services support this too?"}
19     ],
20     temperature=0.7,
21     max_tokens=800,
22     top_p=0.95,
23     frequency_penalty=0,
24     presence_penalty=0,
25     stop=None)
26
27
28 print(response)
29 print(response['choices'][0]['message']['content'])
30
```

✓ 17.9s

```
{
  "id": "chatcmpl-8Cp65kWmkKF58MB1TtRnUPd60Ak",
  "object": "chat.completion",
  "created": 1698067083,
  "model": "gpt-4-32k",
  "prompt_filter_results": [
    {
      "prompt_index": 0,
      "content_filter_results": {
        "hate": {
          "filtered": false,
          "severity": "safe"
        },
        "self_harm": {
          "filtered": false,
          "severity": "safe"
        },
        "sexual": {
          "filtered": false,
          "severity": "safe"
        },
        "violence": {
          "filtered": false,
          "severity": "safe"
        }
      }
    }
  ]
}
```

PROBLEMS 797 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER CODEWHISPERER REFERENCE LOG

S C:\Users\rob>



HOGESCHOOL
ROTTERDAM

Learn / Azure / AI Services /

Learn how to generate or manipulate text

Article • 08/17/2023 • 2 contributors

In this article

- Design prompts
- Classify text
- Trigger ideas
- Conduct conversations

Show 5 more

Azure OpenAI Service provides a **completion endpoint** that can be used for a wide variety of tasks. The endpoint supplies a simple yet powerful text-in, text-out interface to any [Azure OpenAI model](#). To trigger the completion, you input some text as a prompt. The model generates the completion and attempts to match your context or pattern. Suppose you provide the prompt "As Descartes said, I think, therefore" to the API. For this prompt, Azure OpenAI returns the completion endpoint "I am" with high probability.

The best way to start exploring completions is through the playground in [Azure OpenAI Studio](#). It's a simple text box where you enter a prompt to generate a completion. You can start with a simple prompt like this one:

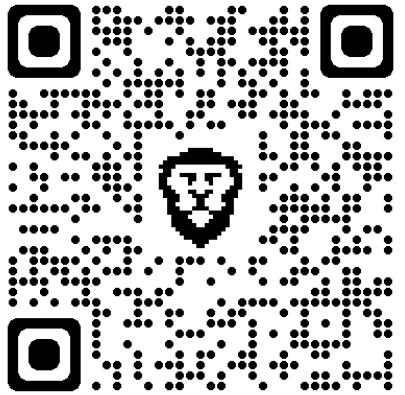
Console Copy

```
write a tagline for an ice cream shop
```

After you enter your prompt, Azure OpenAI displays the completion:

Console Copy

```
we serve up smiles with every scoop!
```



<https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/completions>

GPT-3.5 models

GPT-3.5 Turbo is used with the Chat Completion API. GPT-3.5 Turbo (0301) can also be used with the Completions API. GPT3.5 Turbo (0613) only supports the Chat Completions API.

GPT-3.5 Turbo version 0301 is the first version of the model released. Version 0613 is the second version of the model and adds function calling support.

Model ID	Base model Regions	Fine-Tuning Regions	Max Request (tokens)	Training Data (up to)
<code>gpt-35-turbo¹</code> (0301)	East US, France Central, South Central US, UK South, West Europe	N/A	4,096	Sep 2021
<code>gpt-35-turbo</code> (0613)	Australia East, Canada East, East US, East US 2, France Central, Japan East, North Central US, Sweden Central, Switzerland North, UK South	North Central US, Sweden Central	4,096	Sep 2021
<code>gpt-35-turbo-16k</code> (0613)	Australia East, Canada East, East US, East US 2, France Central, Japan East, North Central US, Sweden Central, Switzerland North, UK South	N/A	16,384	Sep 2021
<code>gpt-35-turbo-instruct</code> (0914)	East US, Sweden Central	N/A	4,097	Sep 2021

¹ Version 0301 of gpt-35-turbo will be retired no earlier than July 5, 2024. See [model updates](#) for model upgrade behavior.



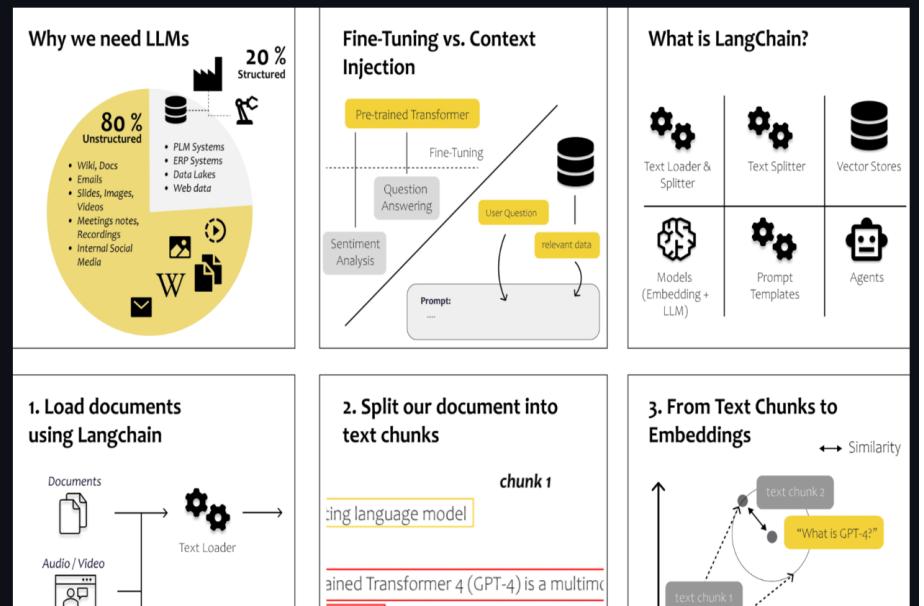
Leer-je-eigen-documenten-bevragen

Context & Doelen



RAG implementatie met Azure + LangChain + OpenAI

1. Begrijpen wat RAG wel en niet kan [\[Wat is RAG\]](#)
2. Veiligheidsmaatregelen nemen
3. LangChain leren gebruiken voor RAG implementatie met
- 4a. [Azure Resource aanmaken nodig voor deployment van een LLM zoals GPT](#)
- 4b. [Azure OpenAI API key + deployment aanmaken voor model: "text-embedding-ada-002"](#)
5. Jupyter Notebook aanmaken in CoLab of Anaconda
6. DEMO [\[DEMO\]](#).



<https://github.com/HR-DATA-FABRIC/Leer-je-eigen-documenten-bevragen-met-generatieve-AI>



demo

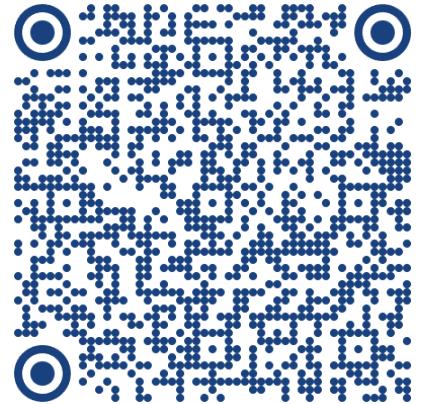
Een voorbeeld van een werkende RAG implementatie met Azure + LangChain + OpenAI is te in te zien via de volgende: [Google Colab Notebook](#): genaamd : LangChain-GPT35_v01.ipynb

The screenshot shows a Jupyter Notebook interface with several code cells. The code is written in Python and performs the following steps:

- Installeert de benodigde packages via Jupyter Notebook.
- Importeert verschillende modules: dotenv, AzureOpenAI, UnstructuredFileLoader, CharacterTextSplitter, AzureOpenAIEmbeddings, Chroma, RetrievalQA.
- Configures environment variables (SHELL, LIBCURLRS_VERSION, INVIDIA_VISIBLE_DEVICES, COLAB_JUPYTER_TRANSPORT, INVA_DEV_VERSION, INVA_PACKAGE_NAME, CGROUP_XMEMORY_EVENTS).
- Laadt het document 'sample.pdf' met de UnstructuredFileLoader.
- Toont de inhoud van het geladen document.
- Splijt het document in kleine stukken met de CharacterTextSplitter.
- Maakt embeddings voor de gesplitste documenten met AzureOpenAIEmbeddings.
- Stelt op voor een RetrievalQA model met de gemaakte embeddings.
- Vraagt de AI naar de auteurs van het document.

De terminal output toont de uitvoer van de code, waaronder de URL van de AI en de resultaten van de zoekopdracht.

<https://colab.research.google.com/drive/1bNbHBXzP1tnDU-xNDRLLeCLFkK3XnI7K#scrollTo=mcw9fPYBOh7b>



{Knowledge Dissemination & Curation}

High quality, insightful Dutch reviews on AI



De (on)mogelijkheden van kunstmatige intelligentie in het onderwijs



In opdracht van:
Ministerie van Onderwijs, Cultuur & Wetenschap

Project:
2018.068

Publicatienummer:
2018.068.1828 v1.0.116

Datum:
Utrecht, 21 januari 2019

Auteurs:
ir. Tommy van der Vorst
ir. Nick Jelicic
mr. Marc de Vries
Julie Albers

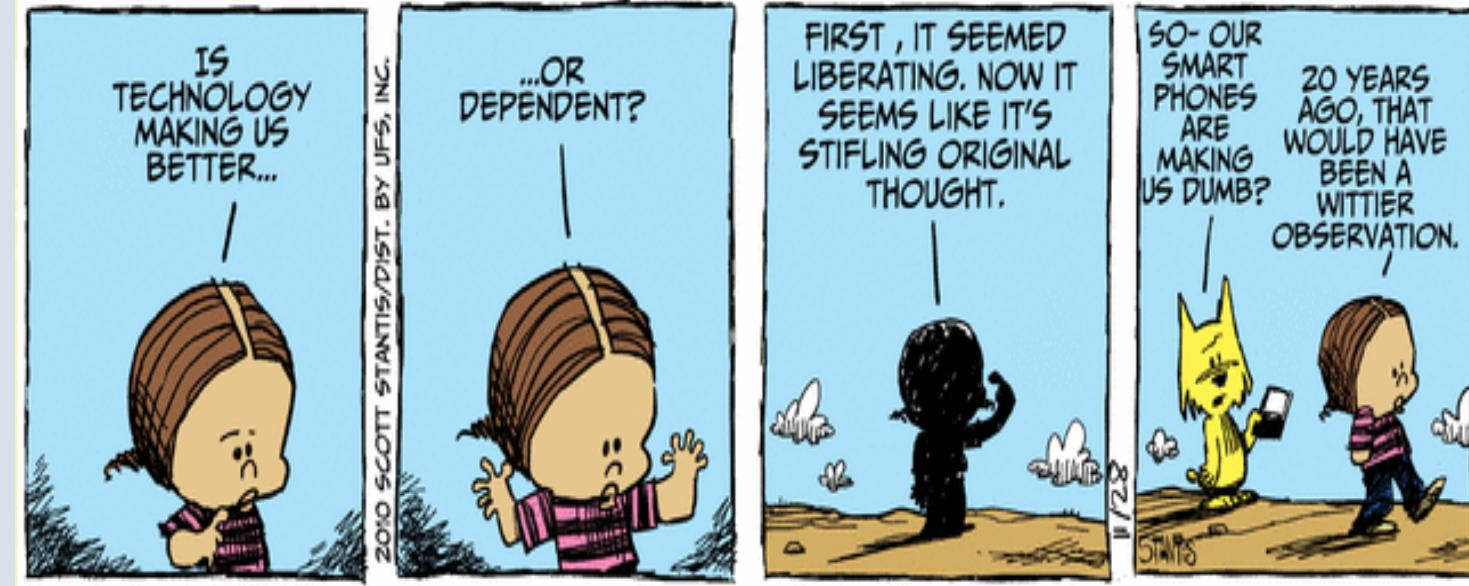
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

These materials are licensed under a Creative Commons Attribution-Share-Alike license.
You can change it, transmit it, show it to other people. Just always give credit to RFvdW.



This seminar was developed by:
Programma AI & Ethisiek
Lead-Tech: Rob van der Willigen

JUNI 2024



Creative Commons License Types		
	Can someone use it commercially?	Can someone create new versions of it?
Attribution	①	②
Share Alike	①②	Yup, AND they must license the new work under a Share Alike license.
No Derivatives	①③	
Non-Commercial	②④	Yup, AND the new work must be non-commercial, but it can be under any non-commercial license.
Non-Commercial Share Alike	①②③④	Yup, AND they must license the new work under a Non-Commercial Share Alike license.
Non-Commercial No Derivatives	①②④	

SOURCE
<http://www.masternewmedia.org/how-to-publish-a-book-under-a-creative-commons-license/>