Deploying AI

Introduction to AI Systems

```
$ echo "Data Science Institute"
```

Introduction

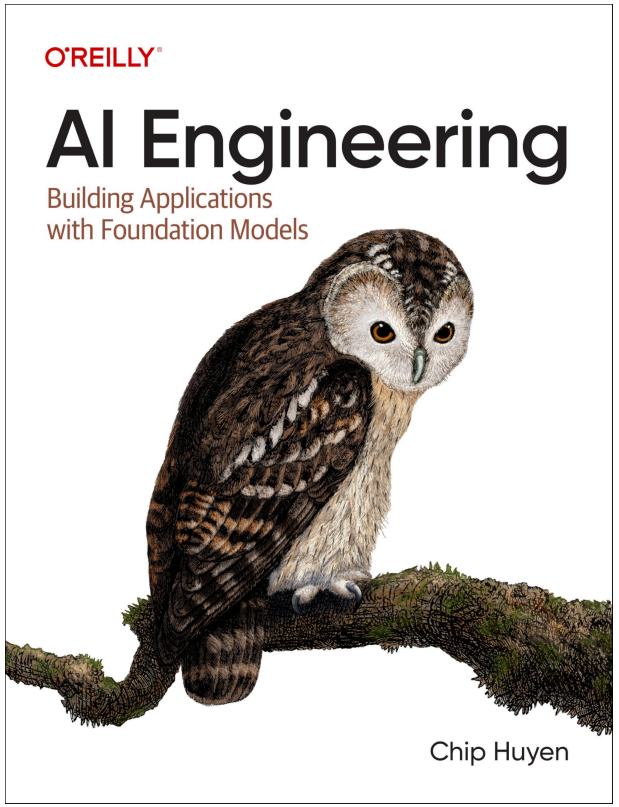
Agenda

Agenda

- What is an AI System?
- Use cases and planning an AI application
- The AI engineering Stack

AI Engineering

We will be covering Chapter 1 of AI Engineering, by Chip Huyen.



Main Points

What is an AI System?

What is an AI System?

- Foundation models
  - Language models
  - Self-supervision
  - From language models to foundation models
- From foundation models to AI engineering

What is an AI System?

- It is a system based on a large-scale machine learning model.
- Many principles of productionizing AI applications are similar to those applied in machine learning engineering.
- However, the availability of large-scale, readily available models affords new possibilities, and also carries risks and challenges.

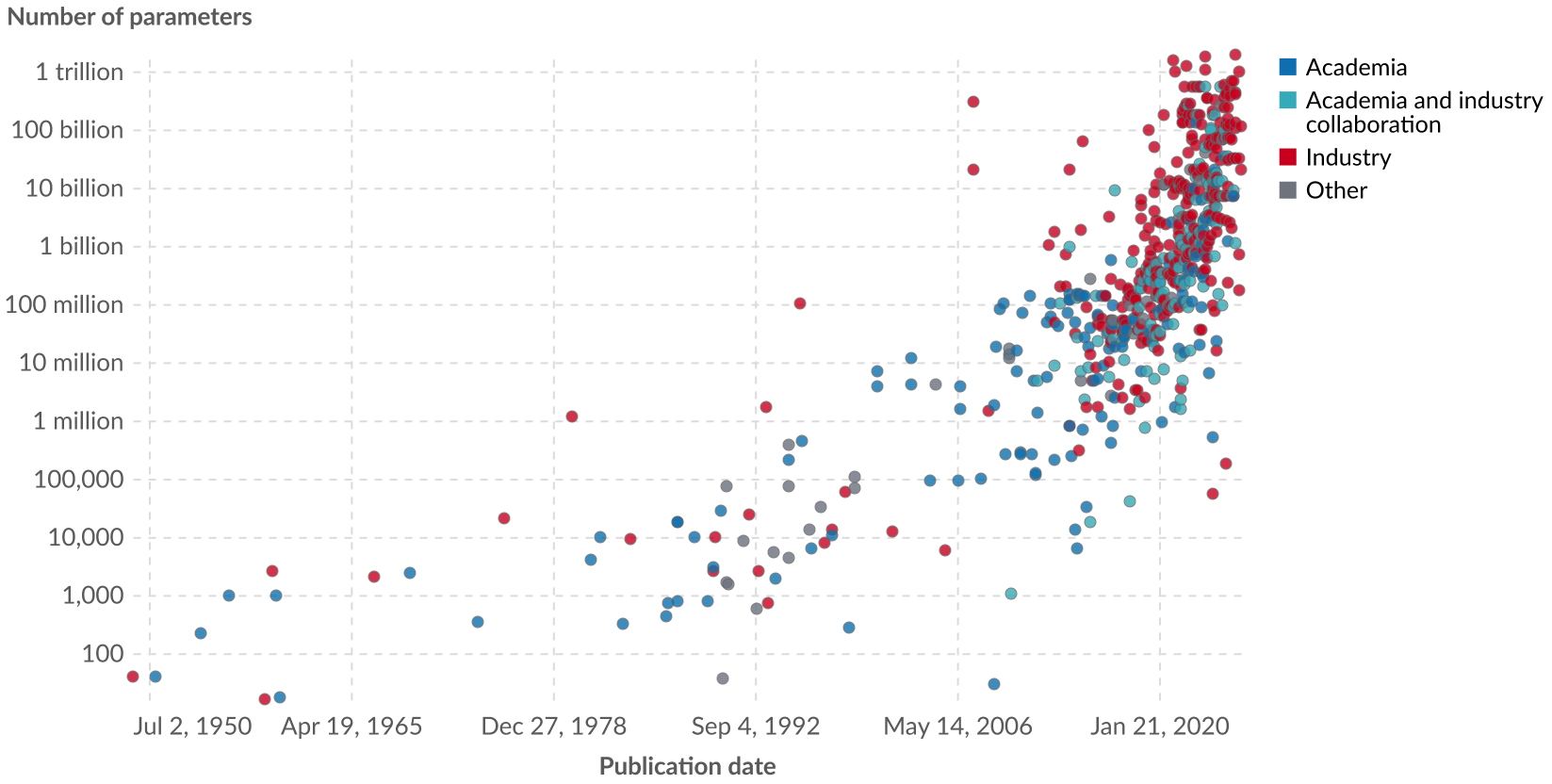
What Makes AI Different?

- AI is different because of scale.
- Large Language Models (LLMs) and other Foundation Models (FMs) follow a maximalist approach to creating models: more complex models are trained on more data as more compute and storage become available.
- FMs are becoming capable of more tasks and therefore they are deployed in more applications and more teams leverage their capabilities.
- FMs require more data, compute resources, and specialized talent.

Parameters in notable artificial intelligence systems

Our World in Data

Parameters are variables in an AI system whose values are adjusted during training to establish how input data gets transformed into the desired output; for example, the connection weights in an artificial neural network.



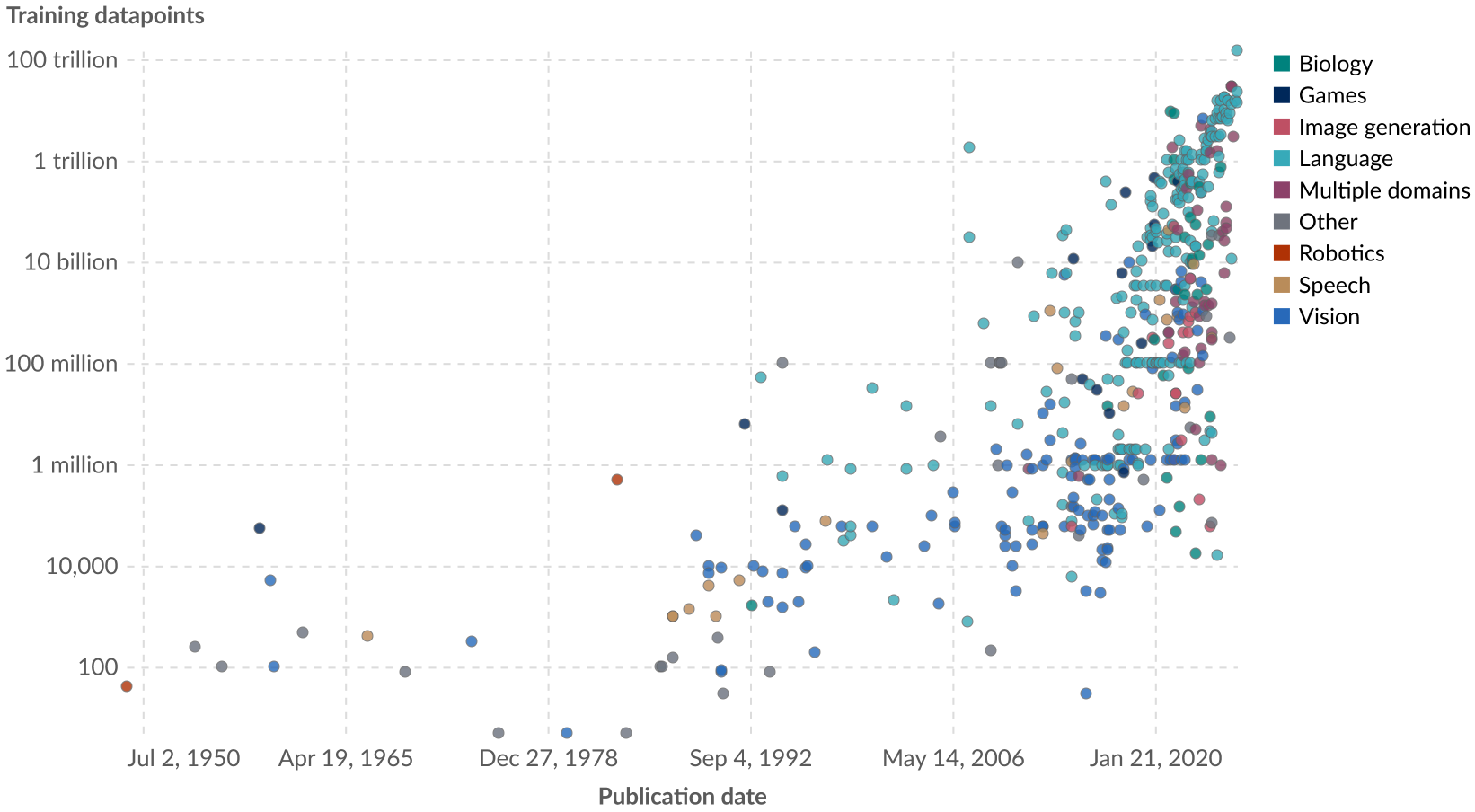
Data source: Epoch (2025)

OurWorldinData.org/artificial-intelligence | CC BY

Note: Parameters are estimated based on published results in the AI literature and come with some uncertainty. The authors expect the estimates to be correct within a factor of 10.

Datapoints used to train notable artificial intelligence systems

Each domain has a specific data point unit; for example, for vision it is images, for language it is words, and for games it is timesteps. This means systems can only be compared directly within the same domain.



Data source: Epoch (2025)

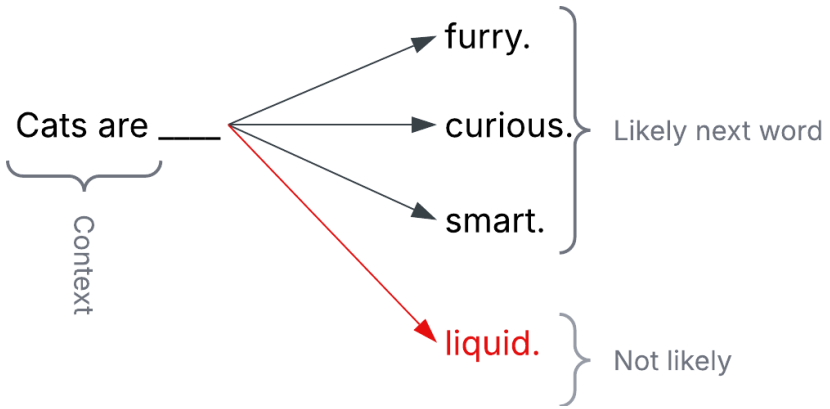
OurWorldinData.org/artificial-intelligence | CC BY

What Makes AI Engineering Different?

- FMs are costly to create, develop, deploy, and maintain. Only a few organizations have the capabilities to do so and typical applications are built upon Models-as-a-Service.
- AI Engineering is the process of building applications on top of readily available models.

Language Models

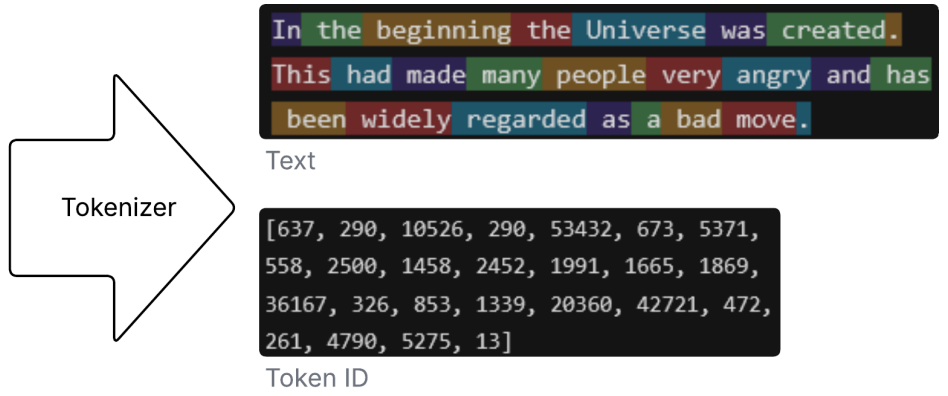
- FMs emerged from LLMs which developed from language models.
- Language models are not new, but have recently developed greatly through *self-supervision*.
- A language model encodes statistical information about one or more languages. Intuitively, we can use this information to know how likely a word is to appear in a given context.



Tokenization

- The basic unit of a language model is a token.
- Tokens can be a character, a word, or a part of a word, depending on the model.
- Tokenization: the process of converting text to tokens.
- The set of all tokens is called *vocabulary*.

In the beginning the Universe was created. This had made many people very angry and has been widely regarded as a bad move.

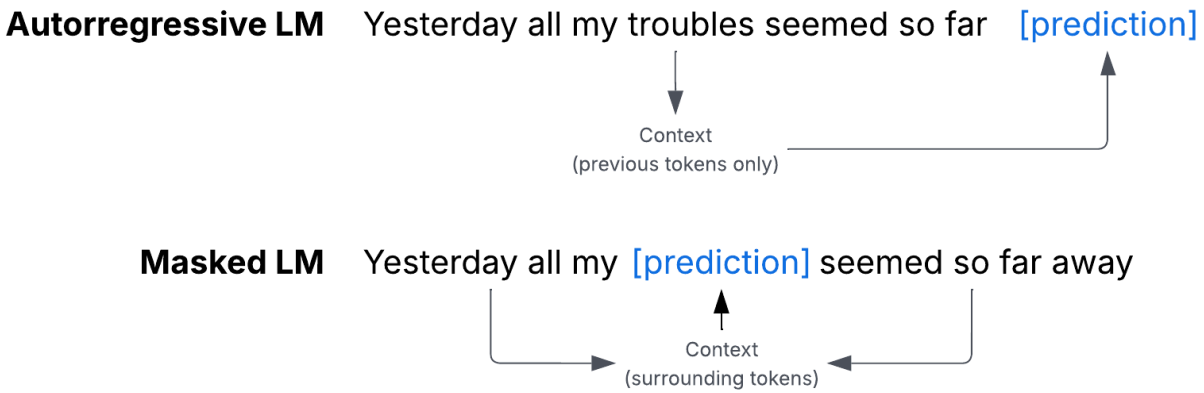


Why use tokens?

1. Tokens allow the model to break words into meaningful components: "walking" can be broken into "walk" and "ing"
2. There are fewer unique tokens than unique words, therefore the vocabulary size is reduced
3. Tokens help the model process unknown words: "chatgpting" can be broken down to "chatgpt" and "ing"

Types of Language Models

There are two types of Language Models (LM): Autorregressive LM and Masked LM.



Masked Language Models

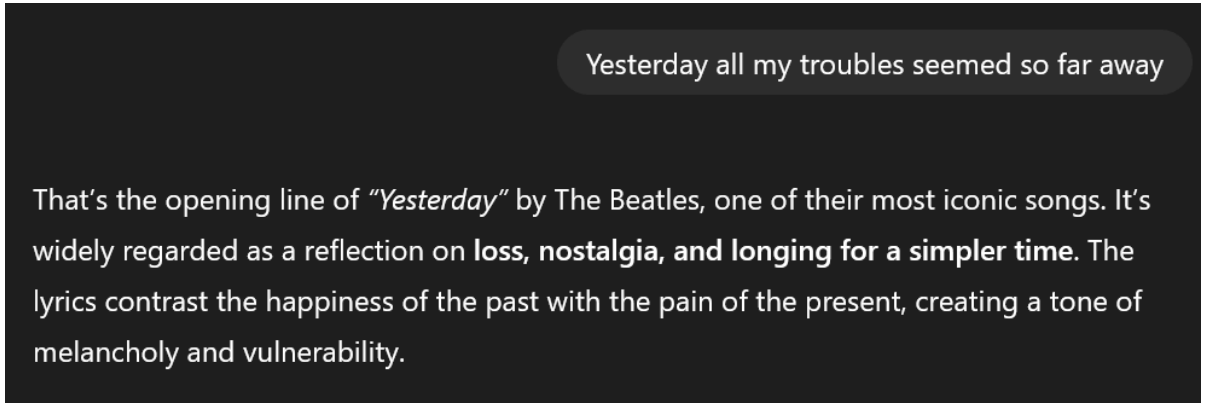
- Masked language model: predicts missing tokens anywhere in a sequence using only the preceding tokens.
- Commonly used for non-generative tasks such as steniment analysis, text classification, and tasks that require an understanding of the general context (before and after the prediction), such as code debugging.
- Example, BERT (Devlin et al., 2018).

Autorregressive Language Models

- Autorregressive language model: trained to predict the next token in a sequence.
- Autorregressive LMs can continually generate one token after another and are the models of choice for text generation.

Completion is a Powerful Task

- The outputs of language models are open-ended.
- Generative model: A model that can generate open-ended outputs.
- An LM is a completion machine: given a text (prompt), it tries to complete the text.

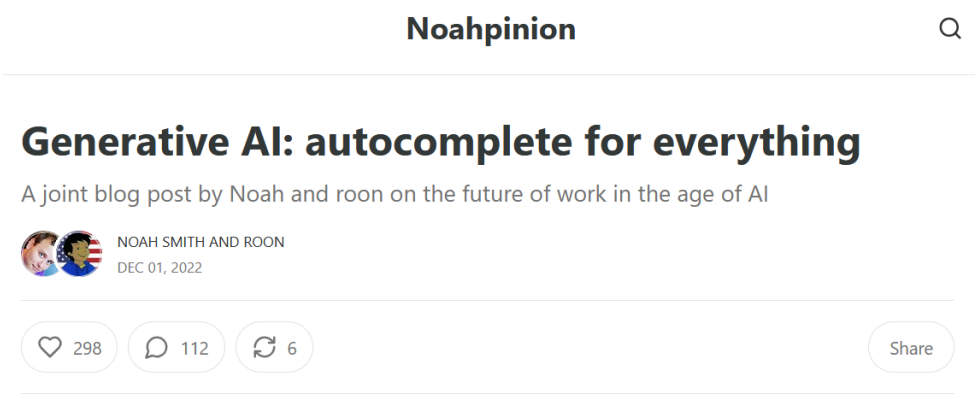


- Completions are predictions, based on probabilities, and not guaranteed to be correct.

Completion Tasks

Many tasks can be thought as completion: translation, summarization, coding, and solving math problems.

> What's common to all of these visions is something we call the "sandwich" workflow. This is a three-step process. First, a human has a creative impulse, and gives the AI a prompt. The AI then generates a menu of options. The human then chooses an option, edits it, and adds any touches they like. (Smith, 2020).



Self-Supervision

- Why language models and not object detection, topic modelling, recommender systems, or any other machine learning task?
- Any machine learning model requires supervision: the process of training a machine learning model using labelled data.
- Supervision requires data labelling, and data labelling is expensive and time-consuming.
- Self-supervision: each input sequence provides both the labels and the contexts the model can use to predict these lables.
- Because text sequences are everywhere, massive training data sets can be constructed, allowing language models to become LLMs.

Self-Supervision: an example

Input	Output (next token)
<BOS>	I
<BOS>, I	love
<BOS>, I, love	street
<BOS>, I, love, street	food
<BOS>, I, love, street, food	.
<BOS>, I, love, street, food, .	<EOS>

From LLM to Foundation Models

- Foundation models: important models which serve as a basis for other applications.
- Multi-modal model: a model that can work with more than one data modality (text, images, videos, protein structures, and so on.)
- Self-supervision works for fourndation models, too. For example, labeled images found on the internet.
- Foundation models transition from task-specific to general-purpose models.

Foundation model use cases

- Coding
- Image and Video Production
- Writing
- Education
- Conversational Bots
- Information Aggregation
- Data Organization
- Workflow Automation

Planning an AI application

- Use Case Evaluation
- Setting Expectations
- Milestone Planning
- Maintenance

The AI engineering Stack

- Three layers of the AI Stacak
- AI Engineering vs ML Enginering
- AI Enginnering vs Full-Stack Engineering

References

References

- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pp. 4171-4186. 2019.
- Huyen, Chip. Designing machine learning systems. O'Reilly Media, Inc., 2022
- Smith, Noah and Roon. Generative AI: autocomplete for everything. Dec. 1, 2022 (URL)