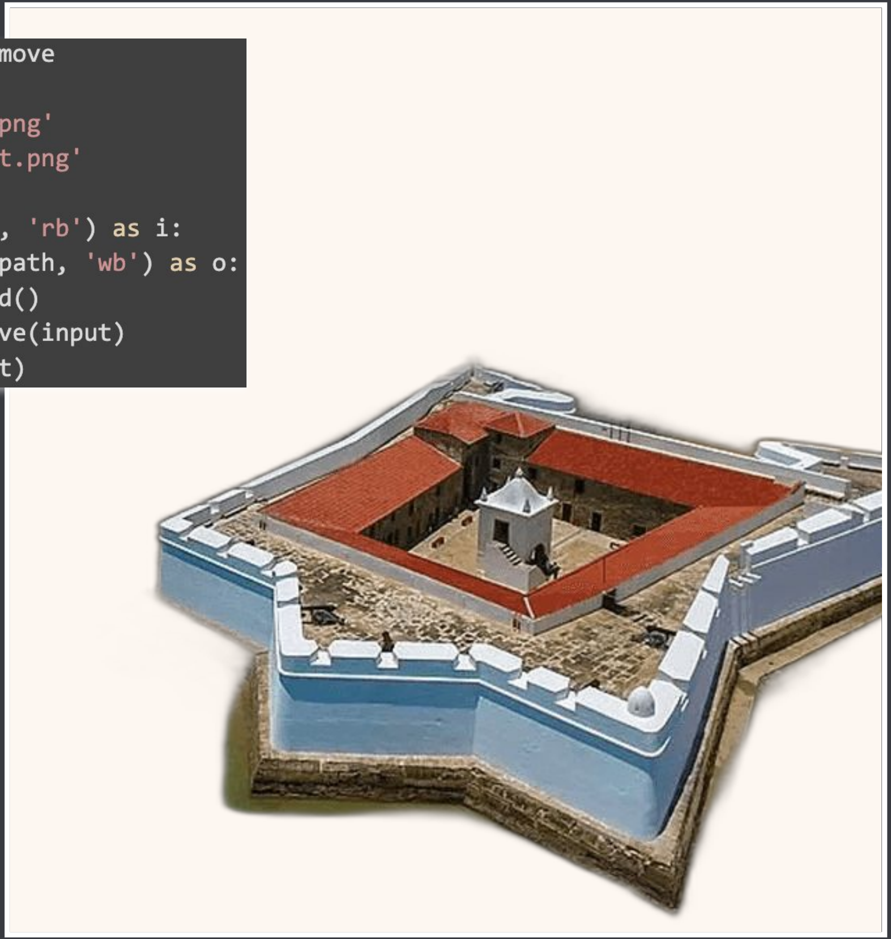
PPGEEC2318

Machine Learning

Fundamentals and First Steps

Ivanovitch Silva

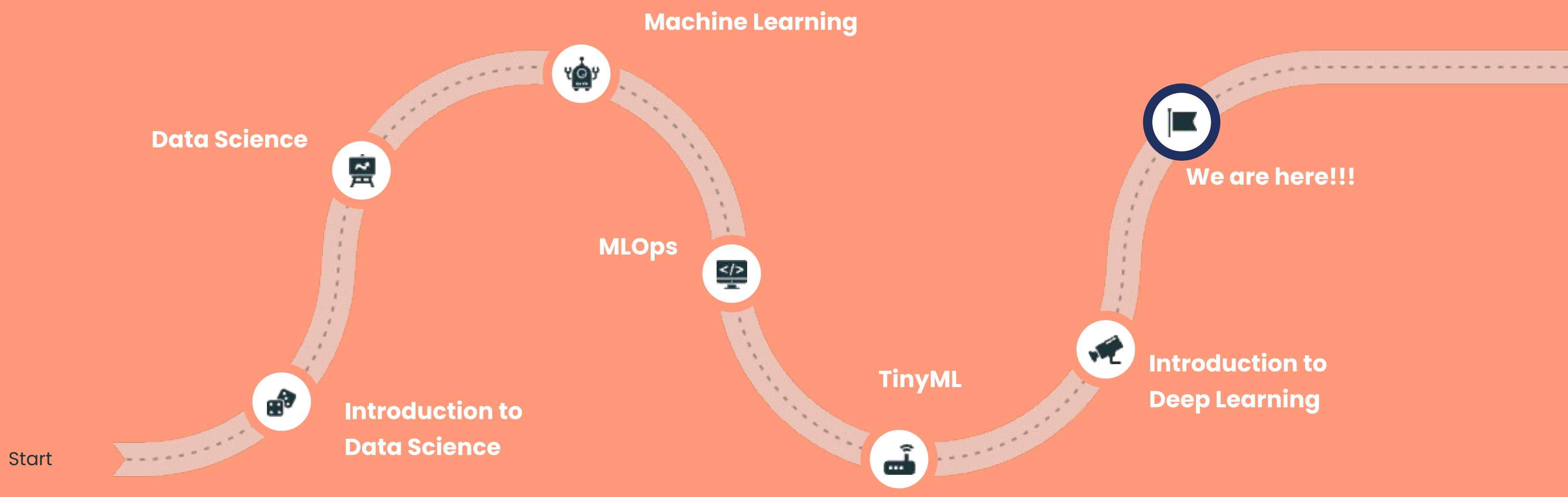


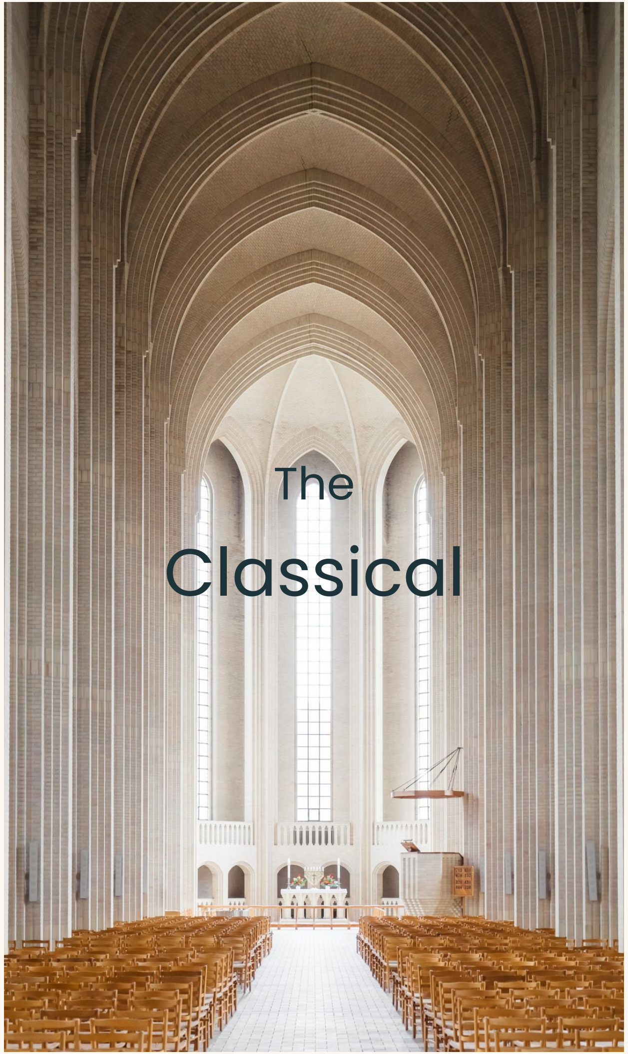
```python
from rembg import remove

input_path = 'input.png'
output_path = 'output.png'

with open(input_path, 'rb') as i:
   with open(output_path, 'wb') as o:
       input = i.read()
       output = remove(input)
       o.write(output)
```

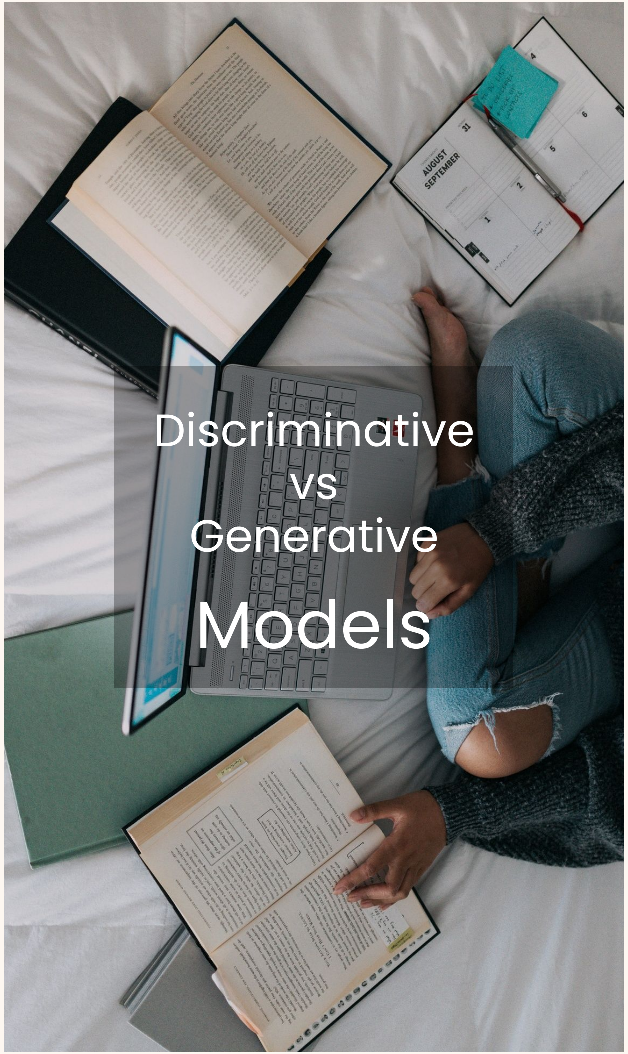
https://github.com/danielgatis/rembg

A Journey to Become an AI Research Scientist

Exploring Different Paths to Acquire Knowledge and Skills for an AI Career




The
Classical

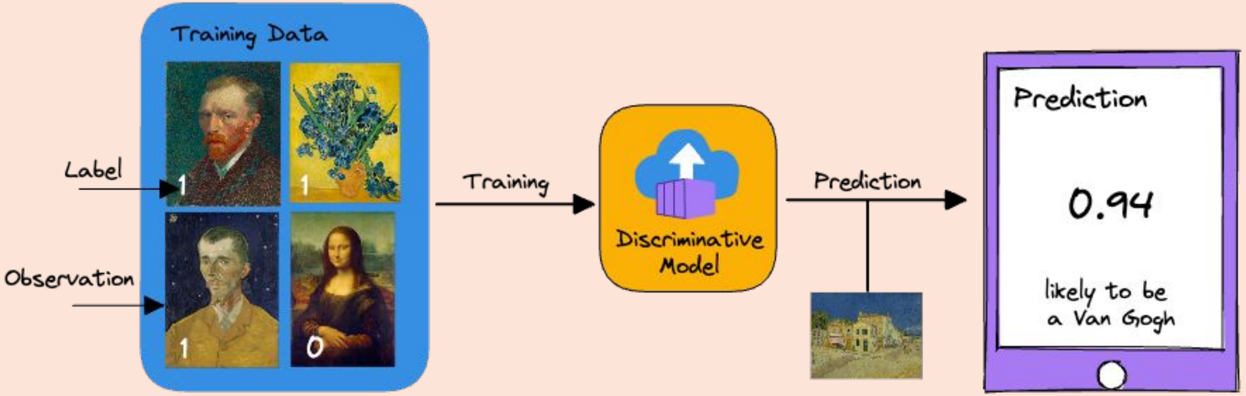

Discriminative
vs
Generative
Models


The
New

Discriminative Model

Van Gogh Paintings

Discriminative modeling estimates P(y|x)



A discriminative model trained to predict if a given image is painted by Van Gogh.

Teachable Machine

https://teachablemachine.withgoogle.com

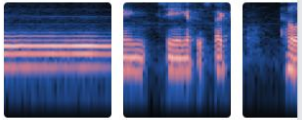
New Project

Open an existing project from Drive.

Open an existing project from a file.

Image Project

Teach based on images, from files or your webcam.



Audio Project

Teach based on one-second-long sounds, from files or your microphone.

Pose Project

Teach based on images, from files or your webcam.

More coming soon

More models will appear here as they're developed.

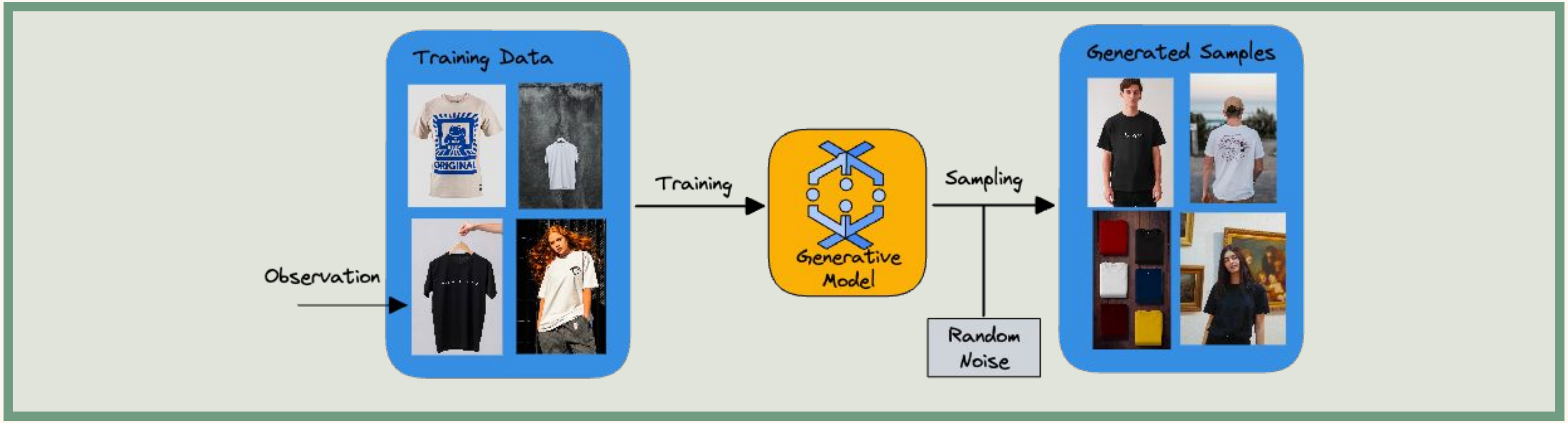
English

release-2-4-10 - 2.4.10#40c178

Generative Model

Generate realistic photos of t-shirts

Generative modeling estimates P(x)



Generative modeling is a branch of machine learning that involves training a model to produce new data that is similar to a given dataset.

YouTube BR

Search

Yann Lecun: Meta AI, Open Source, Limits of LLMs, AGI & the Future of AI | Lex Fridman Podcast #416

Lex Fridman
4.67M subscribers

Subscribe

15K | Share | Save | ...

1.1M views 1 year ago Lex Fridman Podcast
Yann LeCun is the Chief AI Scientist at Meta, professor at NYU, Turing Award winner, and one of the most influential researchers in the history of AI.
Please support this podcast by checking out our sponsors:

Nature of Human Intelligence versus Artificial Intelligence

ESTADÃO Notícias & Editorias Ver & Ouvir Dia a dia Produtos Buscar...

Por Guilherme Nannini

17/09/2024 | 18h30 • Atualização: 18/09/2024 | 14h55

A nova inteligência artificial (IA) da OpenAI, batizada de OpenAI o1, já consegue **"tirar 10"** na prova do Instituto Tecnológico de Aeronáutica (ITA) e ser aprovada para diversas especializações na prova de residência médica da Universidade de São Paulo (USP). A o1 foi lançada na semana passada com a promessa de capacidad de raciocínios lógico e matemático e de resolução de problemas complexos. Agora, IA está sendo submetida a testes no mundo inteiro por especialistas e pesquisadores.

MENU ASSINE FOLHA DE S.PAULO ENTRAR

tec > games crimes virtuais inteligência artificial smartphones

Oferta Especial: R$1,90 no 1º mês ASSINE A FOLHA

TECNOLOGIA

IA da OpenAI passa em residência de medicina na USP e gabarita matemática no vestibular do ITA

Empresa não divulgou detalhes técnicos do modelo, que se destaca nas provas objetivas, mesmo sem interpretar imagens

DÊ UM CONTEÚDO 7

tecmundo O que você procura?

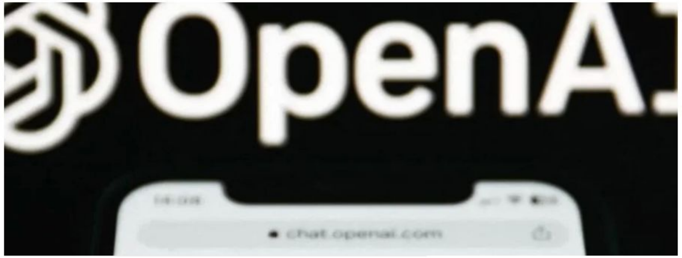
TecFriday Notícias Últimas Notícias Mais Lidas Tutoriais Colunas Guia de Compras Cupons Anuncie

Home > Inteligência artificial

Nova IA da OpenAI se daria bem em provas da USP e do ITA; saiba mais

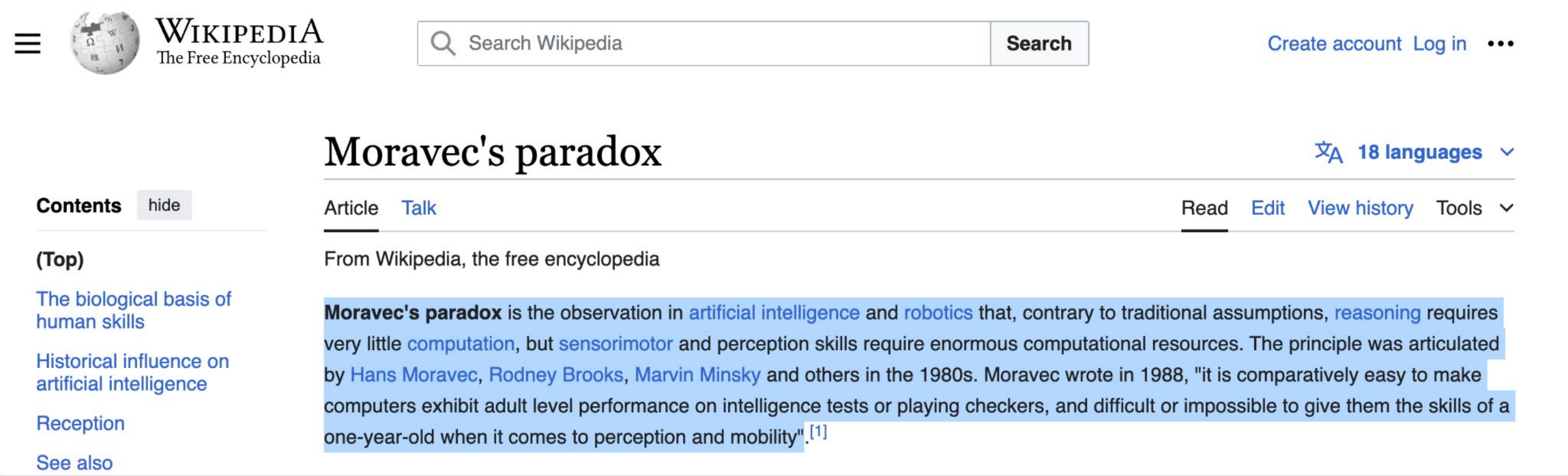
18/09/2024 às 17:25 • 1 min de leitura

COMPARTILHE



André Luiz Dias Gonçalves

via nexperts


Wikipedia
The Free Encyclopedia
Search Wikipedia
Search
Create account Log in

Moravec's paradox

18 languages

Contents hide

(Top)
The biological basis of human skills
Historical influence on artificial intelligence
Reception
See also

Article Talk
Read Edit View history Tools

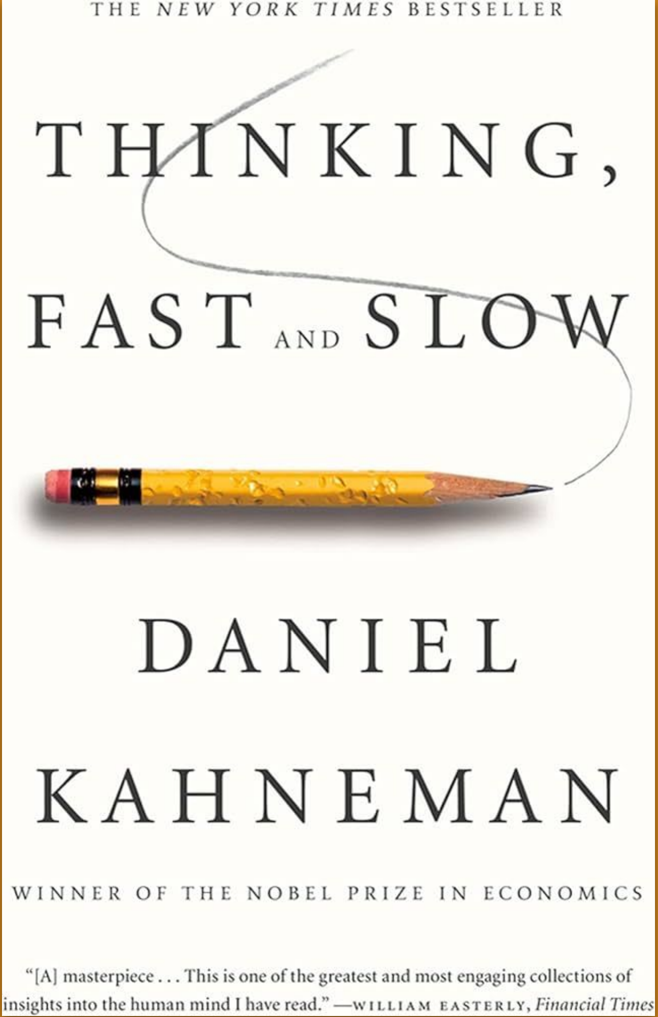
From Wikipedia, the free encyclopedia

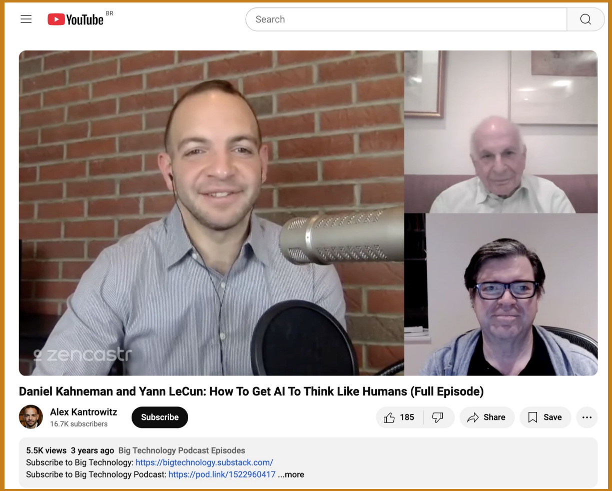
Moravec's paradox is the observation in artificial intelligence and robotics that, contrary to traditional assumptions, reasoning requires very little computation, but sensorimotor and perception skills require enormous computational resources. The principle was articulated by Hans Moravec, Rodney Brooks, Marvin Minsky and others in the 1980s. Moravec wrote in 1988, "it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility".[1]

Moravec's Paradox questions why computers can perform complex tasks like playing chess or solving integrals, but struggle with tasks that humans take for granted, like driving a car or clearing the dinner table.

Limitations of Large Language Models in Achieving Superhuman Intelligence

- **Understanding of the Physical World:**
  - Lack the innate ability to comprehend real-world physics and dynamics.
- **Persistent Memory:**
  - Difficulty in maintaining and utilizing long-term memory effectively.
- **Reasoning Capabilities:**
  - Struggle with logical deduction and complex problem-solving.
- **Planning Skills:**
  - Limited proficiency in strategizing and anticipating future consequences.



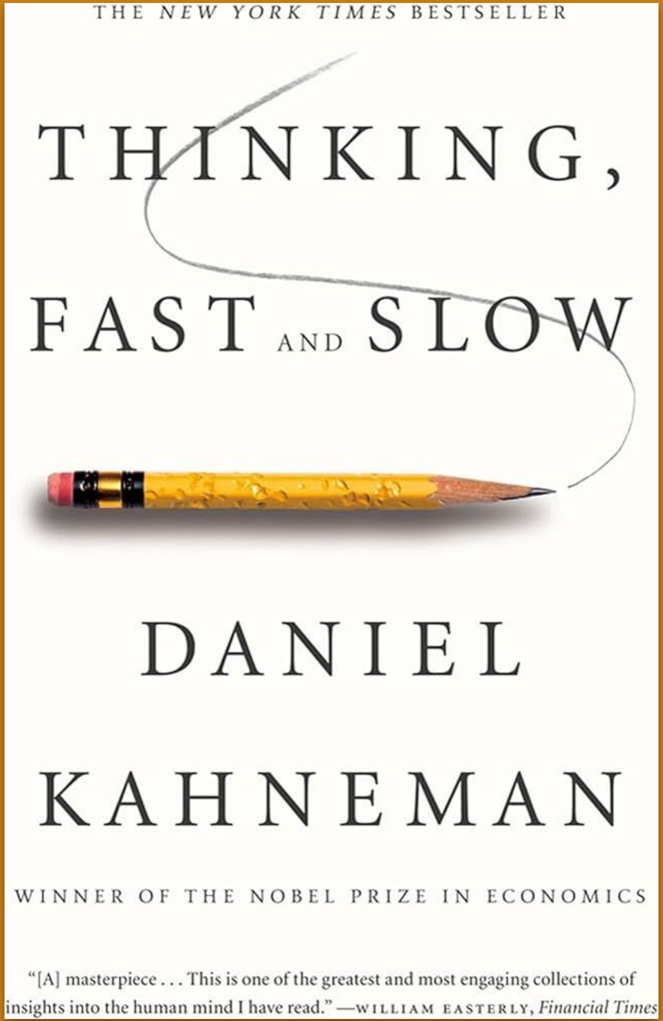


https://www.youtube.com/watch?v=oy9FhisFTmI

System 1 (automatic reasoning)

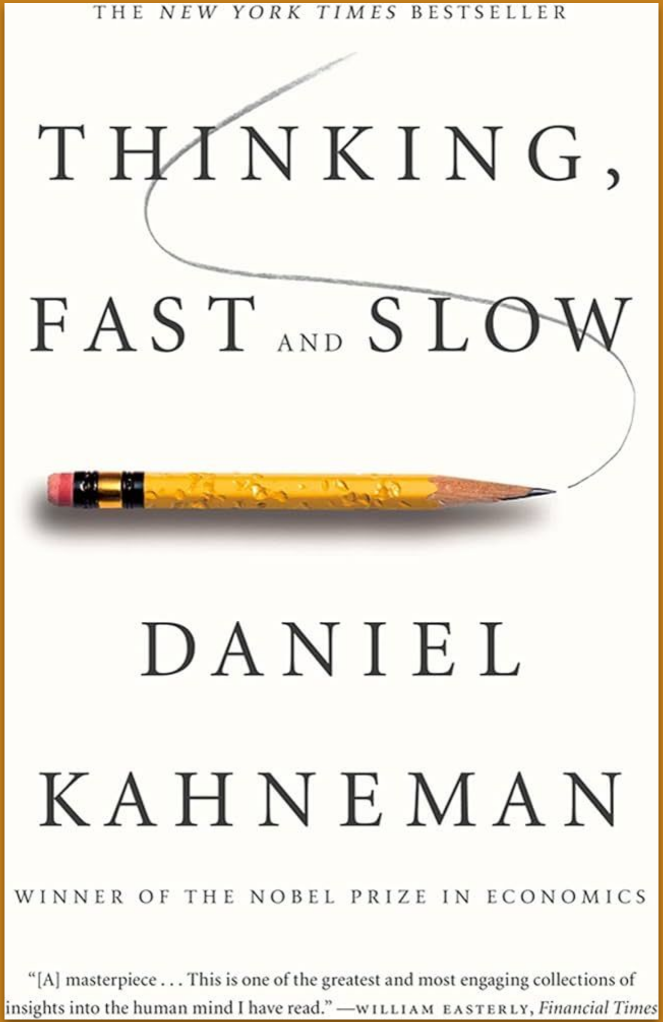
Versus

System 2 (reflective reasoning)



System 1 (Automatic Reasoning):

- **Automatic:** Operates automatically and quickly, with little or no effort and no sense of voluntary control.
- **Based on Impressions:** Uses quick associations and emotions to reach conclusions. It is responsible for intuitive and rapid judgments.
- **Habits and Implicit Learning:** Often based on habits and knowledge acquired through experience, without the person being consciously aware of how they know something.
- **Parallel Processing:** Capable of processing multiple pieces of information simultaneously, which is useful for tasks such as pattern recognition and familiar situations.

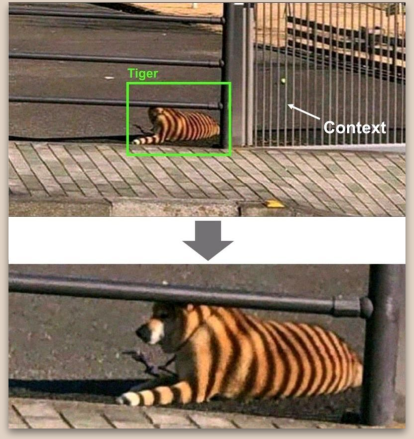


System 2 (Reflective Reasoning):

- **Deliberate and Slow:** Requires effort and is used in complex mental operations, such as complicated mathematical calculations, critical assessments, and considered decisions.
- **Logical and Sequential:** Functions in a more logical and sequential manner, dealing with abstractions, analysis, and planning.
- **Conscious and Controllable:** The operations of System 2 are usually conscious, and the person feels as if they are doing "mental work" when employing it.
- **Limited Capacity:** Has limited capacity, quickly becoming overloaded, and can be slow and laborious.

And about Large Language Models (LLM)?

- Machine Learning sucks!! Compared to humans and animals.
- Most current ML-based AI systems
    - Make stupid mistakes, do not reason nor plan



And about Large Language Models (LLM)?

- **Animals and humans**
  - Can learn new tasks very quickly
  - Understand how the world works
  - Can reason and plan
- Humans and animals have <u>common sense (ethic, sarcasm, irony, emotions, etc)</u>
- Current machines, not so much (it is very superficial)

Why LLM don't reasoning like we do?"

- You want systems that can reason, and certainty that can plan!!!
- The type of reasoning that takes into account in LLM is very, primitive.
- **Constant computation per token**
    - The amount of computation per token is fixed
    - Simple and complex answer receive the same computation effort
- **Consequences**
    - Inability to handle into complex problems
    - Lack of adaptation to the difficult level of the taks
    - Answers may be superficial or inadequate
- **Contrast with human reasoning**
    - We spend more time on more difficult problems
    - Our cognitive effort is proportional to the complexity

Cornell University

We gratefully acknowledge support from the Simons Foundation, member institutions, and all contributors. Donate

arXiv > cs > arXiv:2201.11903

Search... All fields Search
Help | Advanced Search

Computer Science > Computation and Language

[Submitted on 28 Jan 2022 (v1), last revised 10 Jan 2023 (this version, v6)]

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, Denny Zhou

We explore how generating a chain of thought -- a series of intermediate reasoning steps -- significantly improves the ability of large language models to perform complex reasoning. In particular, we show how such reasoning abilities emerge naturally in sufficiently large language models via a simple method called chain of thought prompting, where a few chain of thought demonstrations are provided as exemplars in prompting. Experiments on three large language models show that chain of thought prompting improves performance on a range of arithmetic, commonsense, and symbolic reasoning tasks. The empirical gains can be striking. For instance, prompting a 540B-parameter language model with just eight chain of thought exemplars achieves state of the art accuracy on the GSM8K benchmark of math word problems, surpassing even finetuned GPT-3 with a verifier.

Subjects: **Computation and Language (cs.CL)**; Artificial Intelligence (cs.AI)
Cite as: arXiv:2201.11903 **[cs.CL]**
(or arXiv:2201.11903v6 **[cs.CL]** for this version)
https://doi.org/10.48550/arXiv.2201.11903

Submission history

From: Jason Wei [view email]
[v1] Fri, 28 Jan 2022 02:33:07 UTC (944 KB)
[v2] Wed, 6 Apr 2022 03:51:50 UTC (933 KB)
[v3] Wed, 1 Jun 2022 00:10:30 UTC (303 KB)
[v4] Mon, 13 Jun 2022 21:44:34 UTC (283 KB)
[v5] Mon, 10 Oct 2022 20:21:17 UTC (285 KB)
[v6] Tue, 10 Jan 2023 23:07:57 UTC (306 KB)

Access Paper:

- View PDF
- TeX Source
- Other Formats

Current browse context:
cs.CL
< prev | next >
new | recent | 2022-01

Change to browse by:
cs
cs.AI

References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

17 blog links (what is this?)

DBLP - CS Bibliography

listing | bibtex
Jason Wei
Xuezhi Wang
Dale Schuurmans
Quoc Le
Denny Zhou

Export BibTeX Citation

Bookmark

Bibliographic Tools | Code, Data, Media | Demos | Related Papers | About arXivLabs

Bibliographic and Citation Tools

Bibliographic Explorer (What is the Explorer?)
Litmaps (What is Litmaps?)

The blueprint of the future data systems

- They will think about their answer
- Plan their answer by optimization before turning it into text.
- You have an abstract representation inside the system.
- You have a prompt. The prompt goes through an encoder, produces a representation (or predict ones). But that representation may not be a good answer because there might be some complicated reasoning. So then you have another process that takes the representation of the answers and modifies it so as to minimize a cost function that measures to what extent the answer is a good answer for the question.

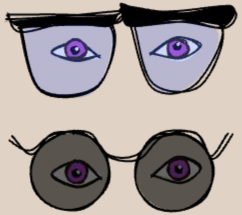
Language is low bandwidth

A person can read 270 words/minutes or 4.5 words/second, which is 12 bytes/s (assuming 2 bytes per token and 0.75 words per token)

A modern LLM is typically trained with 1x10^13 two-byte tokens, which is 2x10^13 bytes.

This would take about 100,000 years for a person to read (at 12 hours a day).

Vision is much higher bandwidth: about 20MB/s



Each of the two optical nerves
has 1 million nerve fibers,
each carrying about
10 bytes per second

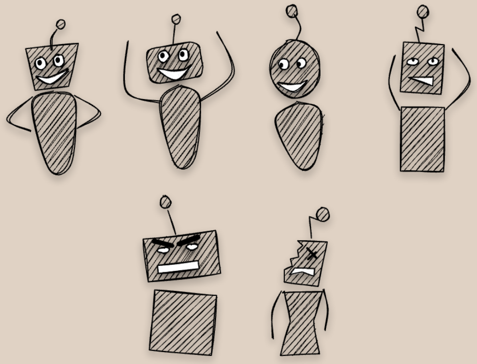
A 4 year-old child has been
awake a total 16,000 hours,
which translates into 1×10^{15} bytes.

In a mere 4 years, a child has
seen 50 times more data than
the biggest LLMs trained on all
the text publicly available on the internet.

bandwidth: 20MB/s

bandwidth: 12 bytes/s

The data bandwidth of visual perception is roughly 1.6 million times higher than the data bandwidth of written (or spoken) language.



Most of human knowledge (and almost all of animal knowledge) comes from our sensory experience of the physical world.

Language is the icing on the cake.
We need the cake to support the icing.

There is absolutely no way in hell we will ever reach human-level AI without getting machines to learn from high-bandwidth sensory inputs, such as vision.

Note: Yes, humans can get smart without vision, even pretty smart without vision and audition.
But not without touch. Touch is pretty high bandwidth, too.

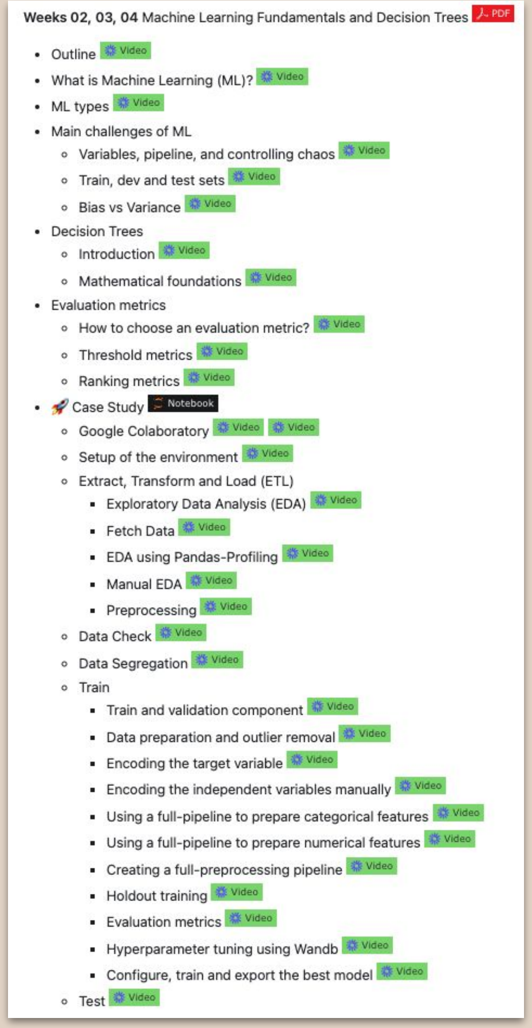
Fundamentals of Machine Learning

A classical ML course

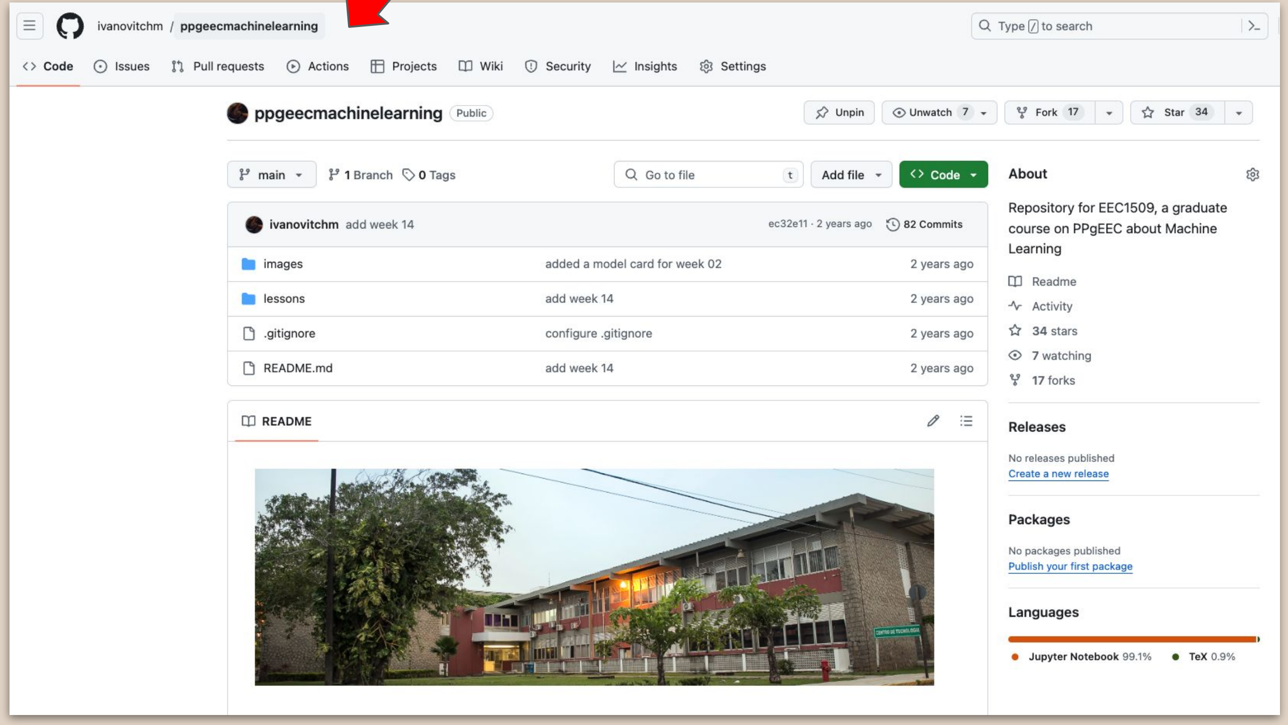
*Undergrad

- Fundamental Concepts in ML
- Data Preparation: cleaning, feature selection, data transforms
- Fundamental Concepts in Statistics
- Assessing Model Performance
- Preventing Overfitting with Regularization
- Unbalacing Data Methods

- Linear Regression
- Gradient Descent
- Logistic Regression
- Naive Bayes
- Support Vector Machines
- Decision Tree
- Random Forest
- Boosting
- Ensemble
- Neural Networks
- Dimensionality Reduction
- Clustering


Weeks 02, 03, 04 Machine Learning Fundamentals and Decision Trees PDF

- Outline Video
- What is Machine Learning (ML)? Video
- ML types Video
- Main challenges of ML
  - Variables, pipeline, and controlling chaos Video
  - Train, dev and test sets Video
  - Bias vs Variance Video
- Decision Trees
  - Introduction Video
  - Mathematical foundations Video
- Evaluation metrics
  - How to choose an evaluation metric? Video
  - Threshold metrics Video
  - Ranking metrics Video
- 🚀 Case Study Notebook
  - Google Colaboratory Video Video
  - Setup of the environment Video
  - Extract, Transform and Load (ETL)
    - Exploratory Data Analysis (EDA) Video
    - Fetch Data Video
    - EDA using Pandas-Profiling Video
    - Manual EDA Video
    - Preprocessing Video
  - Data Check Video
  - Data Segregation Video
  - Train
    - Train and validation component Video
    - Data preparation and outlier removal Video
    - Encoding the target variable Video
    - Encoding the independent variables manually Video
    - Using a full-pipeline to prepare categorical features Video
    - Using a full-pipeline to prepare numerical features Video
    - Creating a full-preprocessing pipeline Video
    - Holdout training Video
    - Evaluation metrics Video
    - Hyperparameter tuning using Wandb Video
    - Configure, train and export the best model Video
  - Test Video



Machine Learning Types

Supervised Learning

Supervised learning requires large numbers of labeled samples.

Unsupervised Leanring

Unsupervised learning requires large numbers of labeled or unlabeled samples.

Semi-Supervised Learning

Semi-Supervised learning requires small numbers of labeled samples and large numbers of unlabeled samples.

Reinforcement Learning

Reinforcement learning requires insane amounts of trials.

Self-Supervised Leanring

Self-Supervised learning requires large numbers of unlabeled samples.

Active Learning

Active learning requires small numbers of labeled samples and large numbers of unlabeled samples.

Weak Supervision

Weak supervision requires small numbers of labeled samples and large numbers of unlabeled samples. Leverage heuristics to generate labels.

Semi-Supervised Learning

Context: You're developing an email classification system to label emails as "spam" or "non-spam." You have a large number of emails, but only a small portion is labeled.

1. **Initial Training:** First, you train your model on the available labeled emails.
2. **Applying to Unlabeled Data:** Next, you use the trained model to make predictions on the unlabeled emails.
3. **Model Refinement:** Based on the predictions, you might assume some of these classifications is correct (especially those where the model is most confident) and use these new "pseudo-labels" to retrain the model.

Key Point: The model attempts to learn from the entirety of available data (both labeled and unlabeled), and the process is largely automated.

Active Learning

Context: You're developing an email classification system to label emails as "spam" or "non-spam." You have a large number of emails, but only a small portion is labeled.

1. **Initial Training:** You start by training your model with a small set of labeled data.
2. **Active Selection of Examples:** The model identifies and selects the unlabeled emails about which it's most uncertain.
3. **Human Intervention:** You, or another expert, manually label these selected emails.
4. **Model Update:** The model is re-trained with the newly labeled data.

Key Point: The model actively seeks human intervention to obtain the most informative labels, maximizing learning efficiency with minimal labeled data.

Weak Supervision

Context: You're developing an email classification system to label emails as "spam" or "non-spam." You have a large number of emails, but only a small portion is labeled.

1. **Generating Weak Labels:** Use various heuristics, external knowledge sources, or simpler models to label the large dataset. For example, emails containing certain keywords like "offer" might be weakly labeled as "spam."
2. **Training with Weak Labels:** Train your model on this weakly labeled data. The labels aren't perfectly accurate, but they provide a broader context and learning opportunity.
3. **Refining the Model:** Optionally, use the model's predictions, in combination with the small set of accurately labeled data, for further refinement.

Key Point: The model relies on noisily or weakly labeled data, often generated through heuristics or auxiliary information, to provide a broad base for initial training.

Semi-Supervised versus Active Learning versus Weak Supervision

Each approach tackles the challenge of limited labeled data in a different way: **semi-supervised learning** leverages unlabeled data through model predictions, **active learning** focuses on human labeling of the most informative samples, and **weak supervision** uses readily available but less accurate labeling methods to quickly annotate large datasets.

Self-Supervised Learning

Context: You are developing a model to understand and process natural language, specifically to improve the performance of a chatbot.

1. **Data Preparation:** Assume you have a vast collection of text data (like articles, books, or internet comments) but none of it is labeled for any specific NLP task like sentiment analysis or topic classification.
2. **Creating a Pretext Task:** To train your model, you first create a '*pretext*' task. This task should be something that forces the model to understand the structure and semantics of language. A common example is the '*<u>masked word prediction</u>*' task, where some words in a sentence are randomly masked (hidden), and the model's job is to predict these masked words based only on the context provided by the surrounding words. For instance, in the sentence "*I love eating ___; it's my favorite fruit,*" the model needs to predict the masked word (e.g., 'umbu').

Cont.

Self-Supervised Learning (continue)

Context: You are developing a model to understand and process natural language, specifically to improve the performance of a chatbot.

3. **Training on the Pretext Task:** You train your model on this task using your large dataset of text. The model learns valuable information about the language, like word associations, grammar, and common phrases, even though it isn't being trained on a specific downstream task like classification or translation yet.
4. **Applying Learned Representations:** After training, the model has learned rich language representations. These can now be fine-tuned or transferred to specific NLP tasks, such as sentiment analysis, named entity recognition, or language translation, using smaller sets of labeled data specific to these tasks.

Key Point: The model learns from data that hasn't been explicitly labeled for the task at hand (<u>learning from unlabeled data</u>). A task is created (predicting masked words) that is related to the skills needed for the eventual target tasks. The representations learned through the pretext task are broadly useful for a range of NLP tasks, not just the one it was trained on (<u>transferable knowledge</u>).