

Conversational Models and LLM

The principles behind Large Language Models
and their application to health

 Kode

What is Kode?

Kode is a software development company specialized in transforming customer's needs into **tailor-made solutions and vertical tools** through the development of mathematical and statistical models based on data and rules.

Somebody call this **AI**.



Davide Massidda
Data Scientist



Olga Cozzolino
Data Scientist

Approaching NLP today

Understand the landscape: Approaching The Natural Language Processing feels like opening a Pandora's box – an overwhelming wave of knowledge, tools, and models.

Navigate rapid change: New technologies emerge and seem obsolete overnight, while older methods still hold as the state of the art.

Cut through the noise: Everything looks impressive, every model is disruptive, every tutorial calls itself essential.

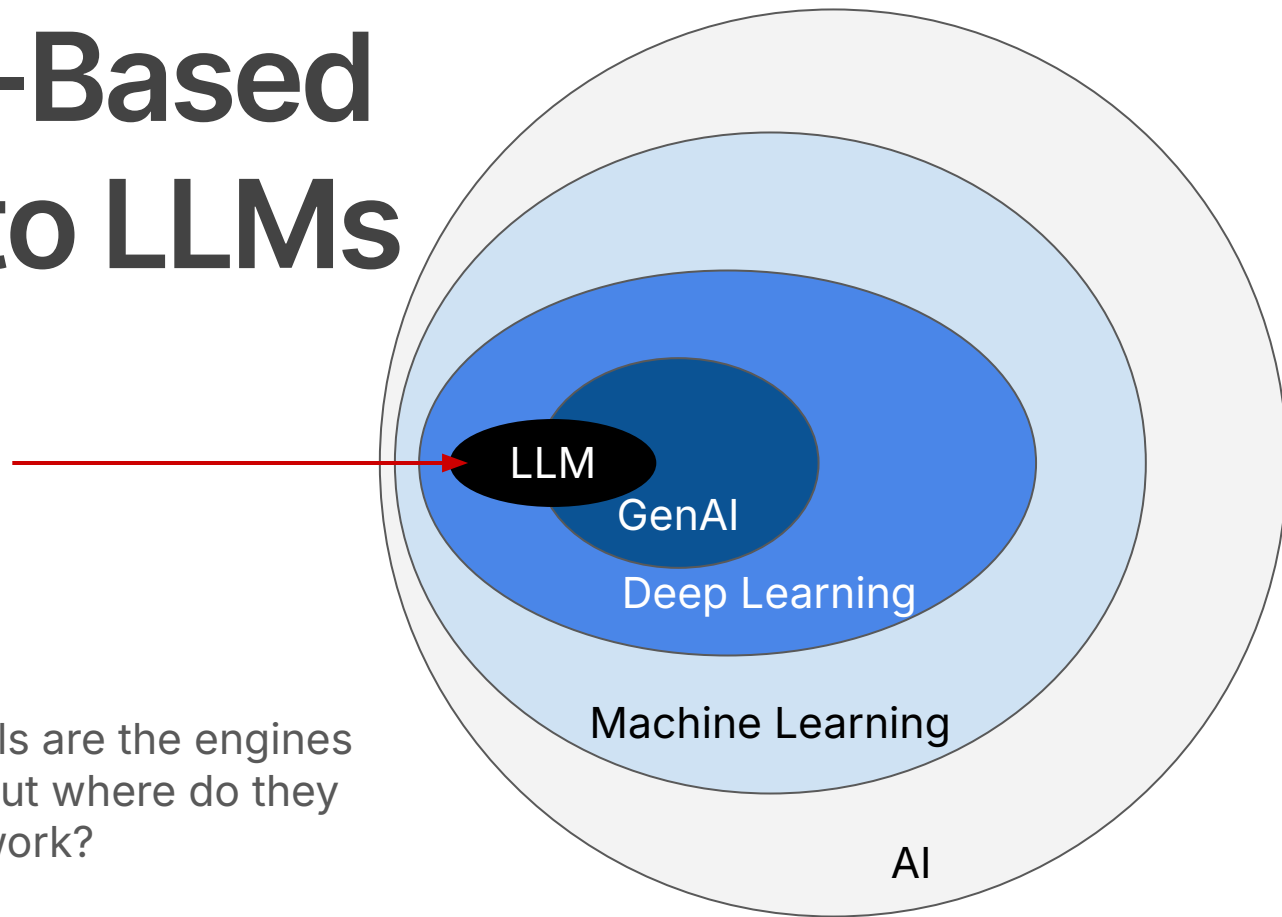
Set learning priorities: What should you study first? What truly matters for healthcare professionals?

Gain clarity: In this session, we aim to make sense of the chaos and outline a practical path forward.

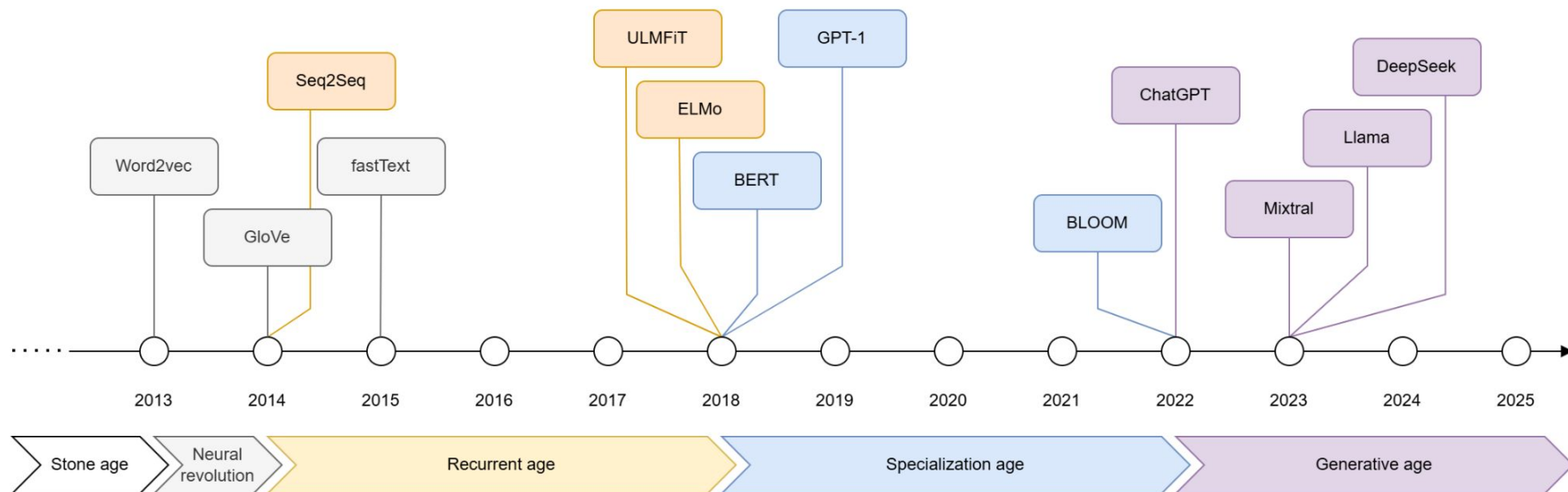


From Rule-Based Chatbots to LLMs

What 99% of
non-experts think
AI is today



The Large Language Models are the engines driving modern chatbots. But where do they come from? How do they work?



Pipeline NLP

- ① A chicken pecks near the dog.
- ② Foxes and dogs are similar.
- ③ He's such a chicken, scared of everything.
- ④ That lousy dog owns a dog I actually like.

Tokenization → Dropping stopwords → Lemmatization

Pipeline NLP

- ① A chicken pecks near the dog.
- ② Foxes and dogs are similar.
- ③ He's such a chicken, scared of everything.
- ④ That lousy dog owns a dog I actually like.

Tokenization → Dropping stopwords → Lemmatization

Pipeline NLP

① A chicken pecks near the dog.

chicken

peak

near

dog

② Foxes and dogs are similar.

fox

dog

similar

③ He's such a chicken, scared of everything.

He

such

chicken

scare

everything

④ That lousy dog owns a dog I actually like.

That

lousy

dog

own

dog

I

like

"Bag of Words" Methods

	chicken	peck	near	dog	fox	similar	he	such	scare	everything	that	lousy	own	like
①	1	1	1	1	0	0	0	0	0	0	0	0	0	0
②	0	0	0	1	1	1	0	0	0	0	0	0	0	0
③	1	0	0	0	0	0	1	1	1	1	0	0	0	0
④	0	0	0	2	0	0	0	0	0	0	1	1	1	1

"Bag of Words" Methods

	chicken	peck	near	dog	fox	similar	he	such	scare	everything	that	lousy	own	like
①	1	1	1	1	0	0	0	0	0	0	0	0	0	0
②	0	0	0	1	1	1	0	0	0	0	0	0	0	0
③	1	0	0	0	0	0	1	1	1	1	0	0	0	0
④	0	0	0	2	0	0	0	0	0	0	1	1	1	1

Tokenization → Dropping stopwords → Lemmatization

Pipeline NLP

① A **chicken** pecks near the dog.

chicken

peak

near

dog

② Foxes and dogs are similar.

fox

dog

similar

③ He's such a **chicken**, scared of everything.

He

such

chicken

scare

everything

④ That lousy **dog** owns a **dog** I actually like.

That

lousy

dog

own

dog

I

like

Limits of BoW methods

Minimal semantic capture

Corpus-bound dictionary → no new words

Sparse vectors

Text loses richness

Representation at document, not word level

New docs = full vector recalculation (TF-IDF)

Limits of BoW methods



Minimal semantic capture

Corpus-bound dictionary → no new words

Sparse vectors

Text loses richness

Representation at document, not word level

New docs = full vector recalculation (TF-IDF)

Limits of BoW methods



Minimal semantic capture



Corpus-bound dictionary → no new words

Sparse vectors

Text loses richness

Representation at document, not word level

New docs = full vector recalculation (TF-IDF)

Limits of BoW methods



Minimal semantic capture



Corpus-bound dictionary → no new words



Sparse vectors

Text loses richness

Representation at document, not word level

New docs = full vector recalculation (TF-IDF)

Limits of BoW methods



Minimal semantic capture



Corpus-bound dictionary → no new words



Sparse vectors



Text loses richness

Representation at document, not word level

New docs = full vector recalculation (TF-IDF)

Limits of BoW methods



Minimal semantic capture



Corpus-bound dictionary → no new words



Sparse vectors



Text loses richness



Representation at document, not word level

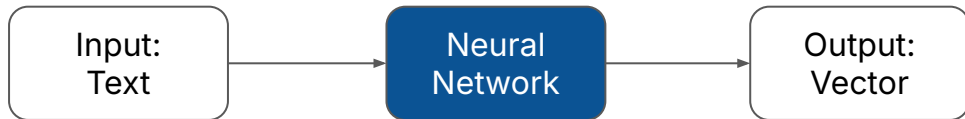
New doc → full vector recalculation (TF-IDF)

Limits of BoW methods

- ✂️ Minimal semantic capture
- ✂️ Corpus-bound dictionary → no new words
- ✂️ Sparse vectors
- ✂️ Text loses richness
- ✂️ Representation at document, not word level
- ✂️ New doc → full vector recalculation (TF-IDF)

The neural revolution

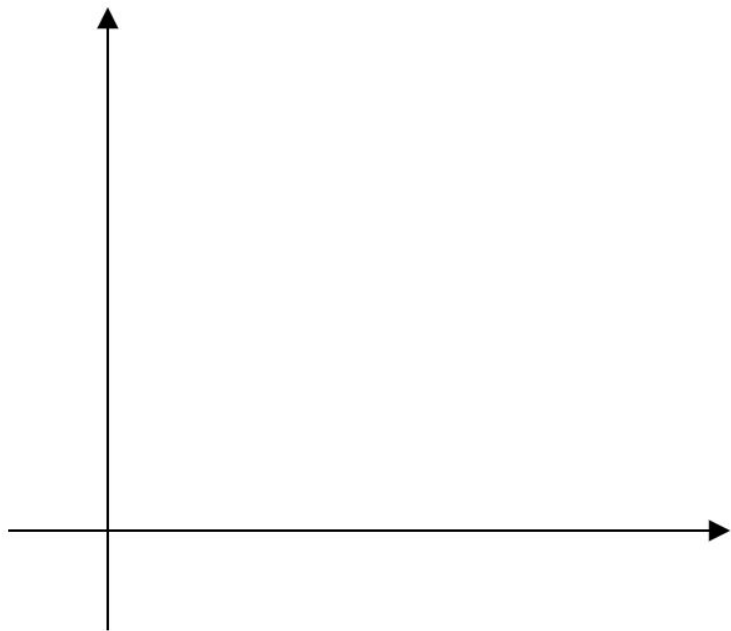
In **2013**, Google researchers used artificial neural networks to generate 'dense' vectors capable of representing the semantics of words: **embeddings**.



The **Word2Vec** model is born.



Embedding



Man

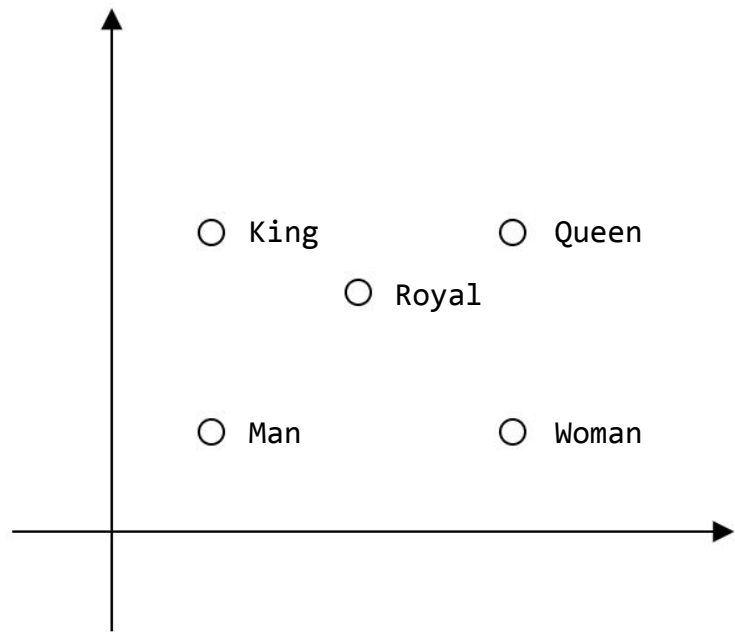
Woman

Royal

King

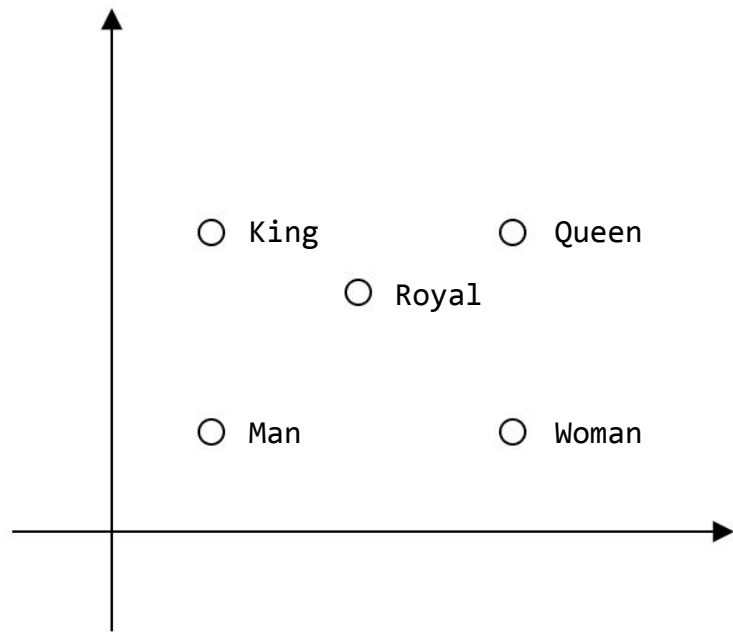
Queen

Embedding



	X		Y
Man	[1	,	1]
Woman	[4	,	1]
Royal	[2.5	,	2.5]
King	[1	,	3]
Queen	[4	,	3]

Embedding



	X		Y
Man	[1	,	1]
Woman	[4	,	1]
Royal	[2.5	,	2.5]
King	[1	,	3]
Queen	[4	,	3]

Similarity indices
can quantify the
similarity
between these
vectors.

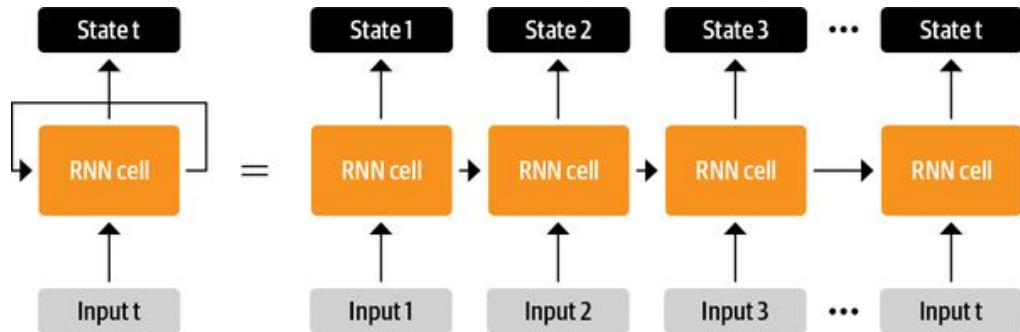
The problem of context

« Can you watch the watch? »

Recurrent Age

The challenge became creating models capable of generating **contextual embeddings**, that is, able to vectorize the same word differently depending on the context.

In **2014**, the Sequence-to-Sequence approach was proposed, and **recurrent neural networks (RNNs)** became the standard.



Limits



Difficulty capturing long-range dependencies (e.g., relationships between words at the beginning and end of a sequence).



Slow sequential computation (not easily parallelizable).

Attention Mechanism

Also in **2014**, attention was introduced: a mechanism to support recurrent networks that 'looks' at all words simultaneously, assigning them weights based on relevance, **without depending on their position in the sequence**.

This gave rise to hybrid architectures: recurrence + attention.

In **2018**, important models such as ULMFiT and ELMo were released.

Specialization age

In **2017**, a very famous paper by Google researchers titled **Attention Is All You Need** showed that what makes language models work is solely and exclusively attention.

It was time to abandon recurrent networks in favor of a new architecture: **Transformers**.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com	Noam Shazeer* Google Brain noam@google.com	Niki Parmar* Google Research nikip@google.com	Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com	Aidan N. Gomez*[†] University of Toronto aidan@cs.toronto.edu	Lukas Kaiser* Google Brain lukaskaiser@google.com	
Illa Polosukhin*[‡] illia.polosukhin@gmail.com			

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modelling and transduction problems such as language modelling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [31, 21, 13].

^{*}Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

[†]Work performed while at Google Brain.
[‡]Work performed while at Google Research.

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

Bidirectional Encoder Representations from Transformers

In **2018**, **BERT was released**, quickly surpassing ELMo thanks to bidirectional self-attention, and establishing the Transformer-based paradigm.

From BERT came a series of spin-off models such as **RoBERTa** and **DistilBERT**, along with many others built on the same architecture.

This officially marked the beginning of the era of specialized models, consolidating the dominance of Transformers

Google
BERT



Some stars of the time

Model	Task	Architecture
BERT	Text embedding	Encoder-only
T5	Text translation	Encoder-decoder
BART	Text summarization	Encoder-decoder
GPT	Text generation	Decoder-only
BLOOM	Text generation	Decoder-only
PaLM	Text generation, Q&A, and more	Decoder only

The Transformer architecture



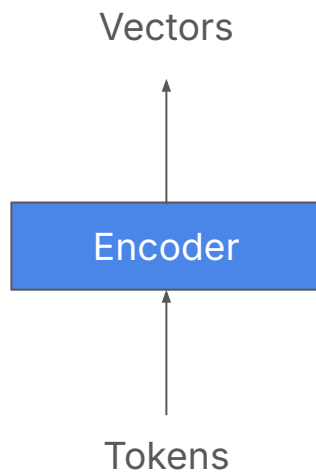
Encoder

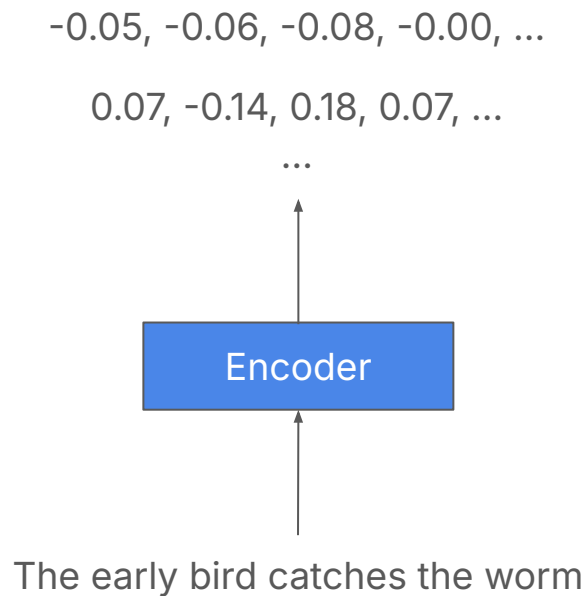
The diagram consists of two rectangular boxes positioned horizontally. The left box is blue and contains the word 'Encoder' in white text. The right box is purple and contains the word 'Decoder' in white text. There are no arrows or other graphical elements connecting the two boxes.

Decoder

Useful for:

Numerical text
representation

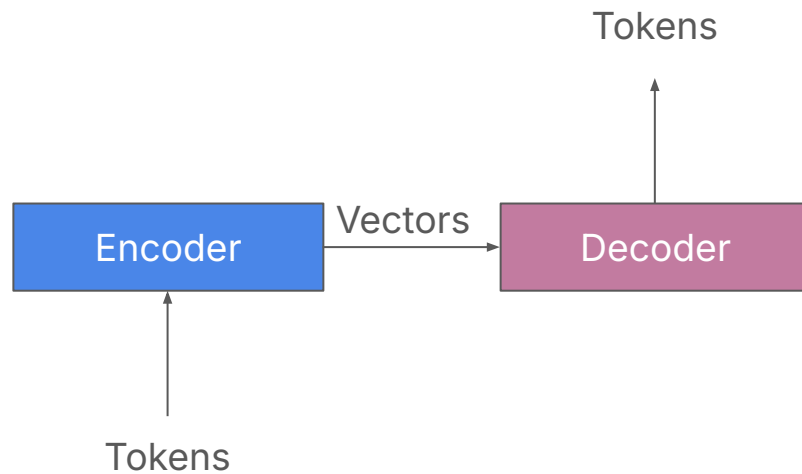


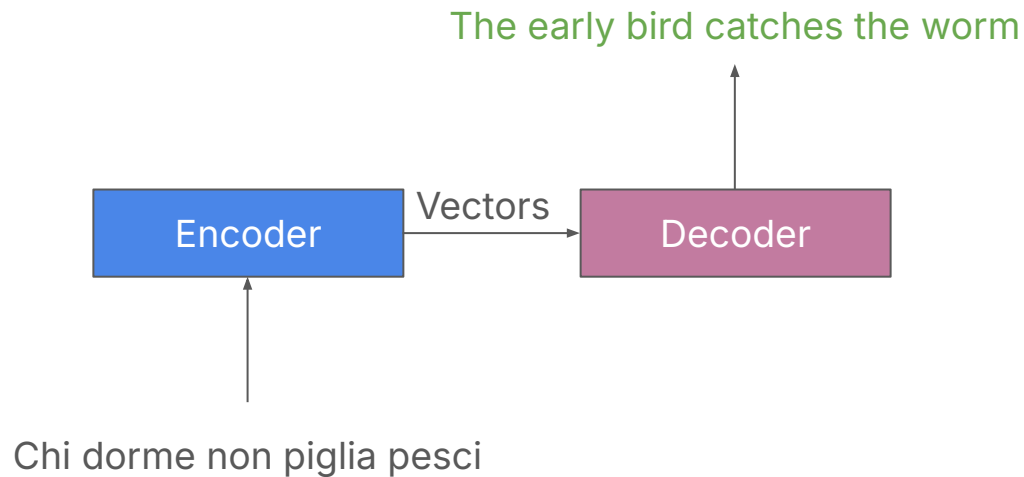


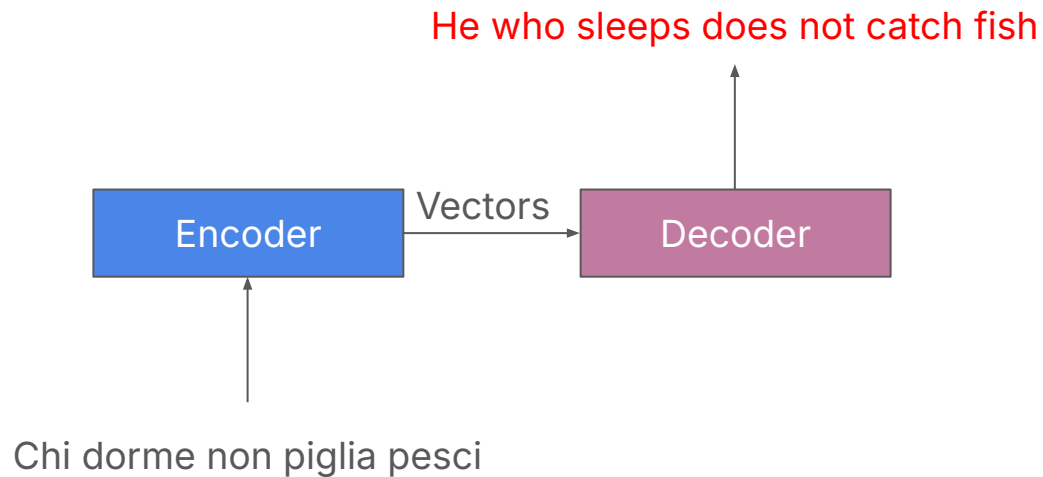
Useful for:

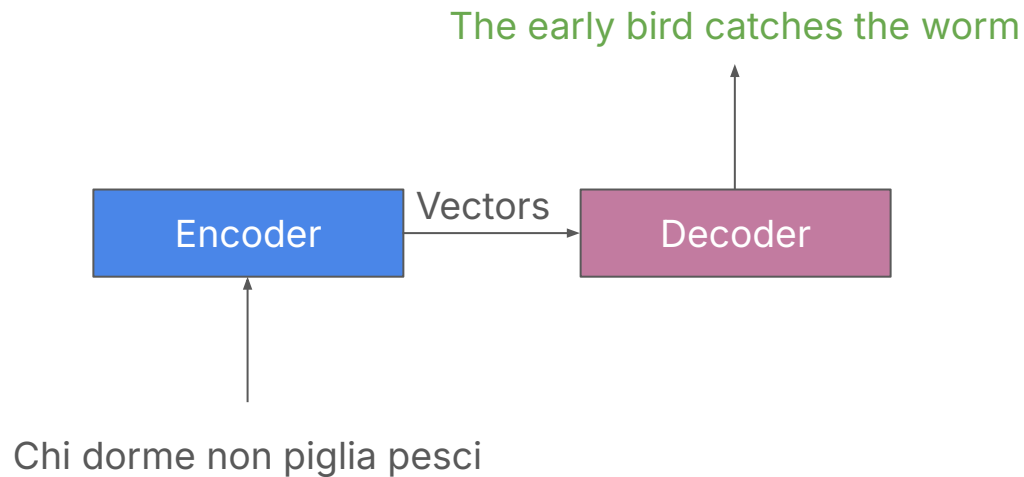
Translation

Summarization



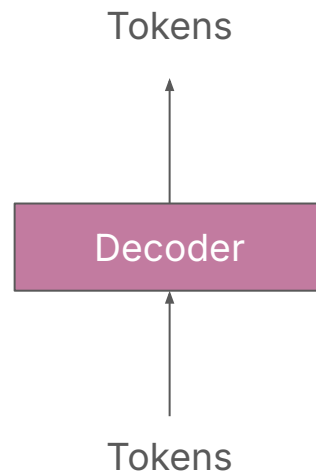


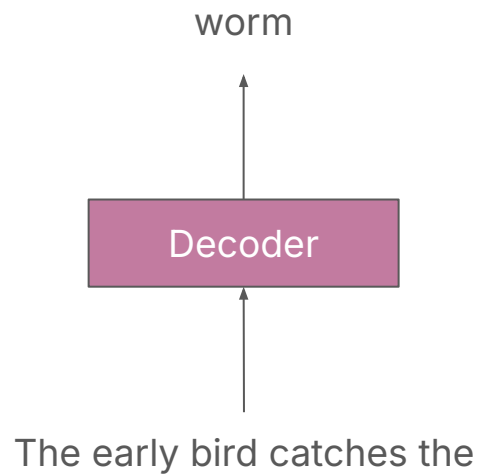




Useful for:

Generation





Generative age

In **2022**, ChatGPT was released. Building on GPT-3, OpenAI researchers created GPT-3.5 by:

- increasing the number of **parameters**, scaling from millions to billions;
- expanding the amount of text the model can take as **input**;
- simulating a **memory** by prompt engineering;
- providing the model with an **interface**;
- **training** the model on a massive corpus of text, consistent with its new scale (practically the entire web);
- introducing **RLHF** (Reinforcement Learning from Human Feedback) to complement supervised learning.

Generative age

In **2022**, ChatGPT was released. Building on GPT-3, OpenAI researchers created GPT-3.5 by:

- increasing the number of **parameters**, scaling from millions to billions;
- expanding the amount of text the model can take as **input**;
- simulating a **memory** by prompt engineering;
- providing the model with an **interface**;
- **training** the model on a massive corpus of text, consistent with its new scale (practically the entire web);
- introducing **RLHF** (Reinforcement Learning from Human Feedback) to complement supervised learning.

Generative age

In **2022**, ChatGPT was released. Building on GPT-3, OpenAI researchers created GPT-3.5 by:

- increasing the number of **parameters**, scaling from millions to billions;
- expanding the amount of text the model can take as **input**;
- simulating a **memory** by prompt engineering;
- providing the model with an **interface**;
- **training** the model on a massive corpus of text, consistent with its new scale (practically the entire web);
- introducing **RLHF** (Reinforcement Learning from Human Feedback) to complement supervised learning.

Generative age

In **2022**, ChatGPT was released. Building on GPT-3, OpenAI researchers created GPT-3.5 by:

- increasing the number of **parameters**, scaling from millions to billions;
- expanding the amount of text the model can take as **input**;
- simulating a **memory** by prompt engineering;
- providing the model with an **interface**;
- **training** the model on a massive corpus of text, consistent with its new scale (practically the entire web);
- introducing **RLHF** (Reinforcement Learning from Human Feedback) to complement supervised learning.

Generative age

In **2022**, ChatGPT was released. Building on GPT-3, OpenAI researchers created GPT-3.5 by:

- increasing the number of **parameters**, scaling from millions to billions;
- expanding the amount of text the model can take as **input**;
- simulating a **memory** by prompt engineering;
- providing the model with an **interface**;
- **training** the model on a massive corpus of text, consistent with its new scale (practically the entire web);
- introducing **RLHF** (Reinforcement Learning from Human Feedback) to complement supervised learning.

Generative age

In **2022**, ChatGPT was released. Building on GPT-3, OpenAI researchers created GPT-3.5 by:

- increasing the number of **parameters**, scaling from millions to billions;
- expanding the amount of text the model can take as **input**;
- simulating a **memory** by prompt engineering;
- providing the model with an **interface**;
- **training** the model on a massive corpus of text, consistent with its new scale (practically the entire web);
- introducing **RLHF** (Reinforcement Learning from Human Feedback) to complement supervised learning.

Generative age

In **2022**, ChatGPT was released. Building on GPT-3, OpenAI researchers created GPT-3.5 by:

- increasing the number of **parameters**, scaling from millions to billions;
- expanding the amount of text the model can take as **input**;
- simulating a **memory** by prompt engineering;
- providing the model with an **interface**;
- **training** the model on a massive corpus of text, consistent with its new scale (practically the entire web);
- introducing **RLHF** (Reinforcement Learning from Human Feedback) to complement supervised learning.

Universal key

Generative models establish themselves as a universal key to solving tasks not necessarily limited to pure generation.

For example:

- Text summarization
- Text labeling
- Information extraction
- ...and many other!

GenLLMs prove capable of **following instructions** and functioning as 'linguistic connectors' between components, serving as a foundation for building pseudo-intelligent agents.

Generative models thus begin to centralize diverse tasks, thanks also to the **Mixture of Experts** approach (e.g., Mixtral).

Clarification

An LLM isn't usually trained "on the fly."

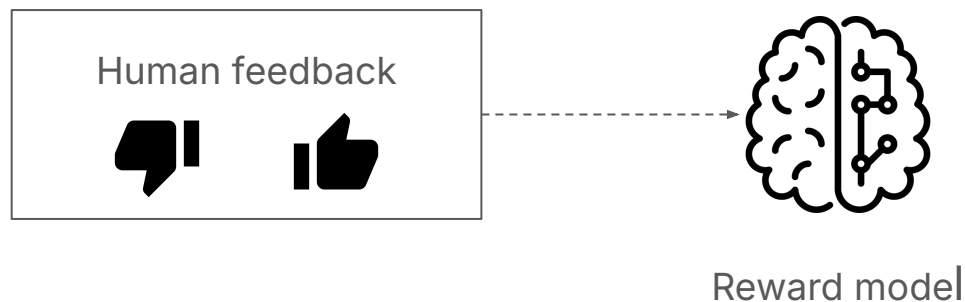


Developers train the models offline and only then make them available for use.

So: your likes, your «Thank you», your «Great answer!» don't instantly make the model smarter.

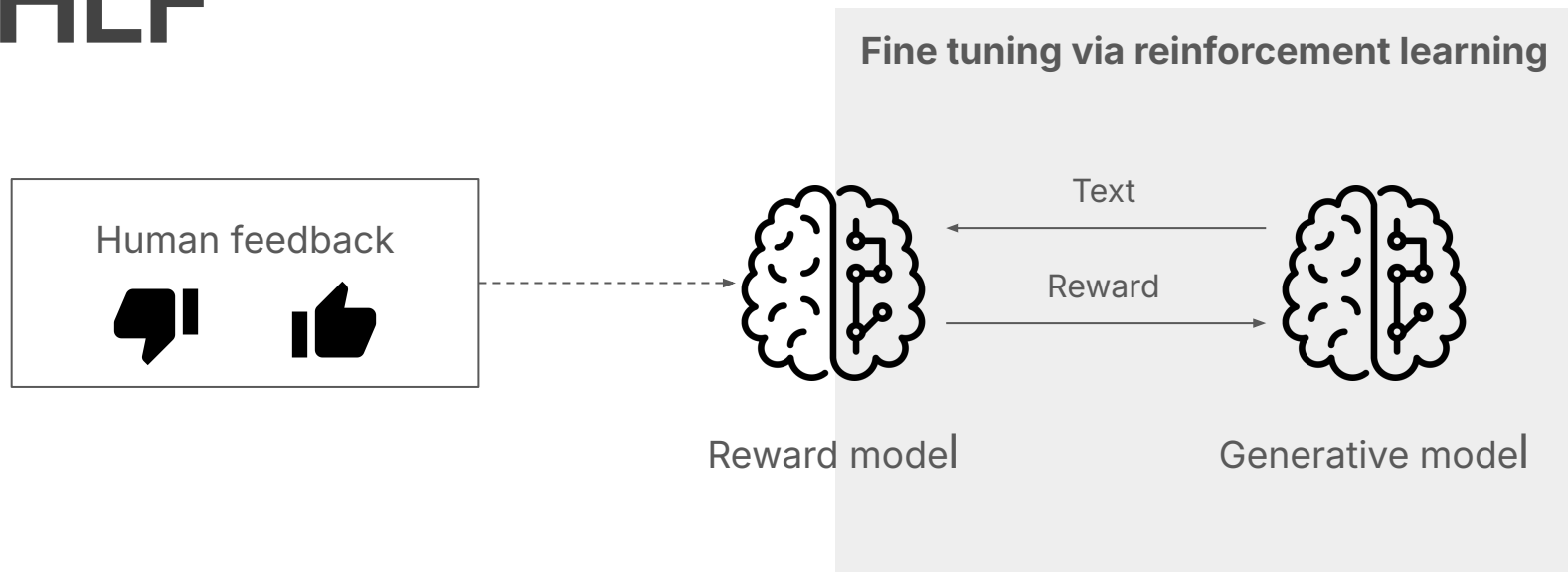
However, your feedback matters: they might help later on, when developers use your interactions to retrain or update it.

RHLF



A model is trained that, given a piece of text, outputs a quality score, which is then used to generate 'rewards'.

RHLF



A model is trained that, given a piece of text, outputs a quality score, which is then used to generate 'rewards'.

Iteration by iteration, the generative model tries to improve itself to maximize rewards.

Direct Preference Optimization

For each prompt, a positively rated response and a negatively rated response are available.

These are used to train the generative model directly, without going through reinforcement learning.

The model must learn to assign a higher probability to the preferred response than to the rejected response.

! Beware of Hallucinations in LLMs

Hallucinations occur when a model generates information that is plausible but false.



WHY:

- LLMs generate text based on patterns in their training data, not on verified facts.
- During training, the model is rewarded for producing plausible-sounding answers, rather than honest "I don't know" responses.
- Without grounding in external knowledge, this can lead to outputs that sound confident but are actually incorrect.

⚠ Beware of Hallucinations in LLMs

Why Language Models Hallucinate

Adam Tauman Kalai*
OpenAI

Ofir Nachum
OpenAI

Santosh S. Vempala[†]
Georgia Tech

Edwin Zhang
OpenAI

September 4, 2025

Abstract

Like students facing hard exam questions, large language models sometimes guess when uncertain, producing plausible yet incorrect statements instead of admitting uncertainty. Such “hallucinations” persist even in state-of-the-art systems and undermine trust. We argue that language models hallucinate because the training and evaluation procedures reward guessing over acknowledging uncertainty, and we analyze the statistical causes of hallucinations in the modern training pipeline. Hallucinations need not be mysterious—they originate simply as errors in binary classification. If incorrect statements cannot be distinguished from facts, then hallucinations in pretrained language models will arise through natural statistical pressures. We then argue that hallucinations persist due to the way most evaluations are graded—language models are optimized to be good test-takers, and guessing when uncertain improves test performance. This “epidemic” of penalizing uncertain responses can only be addressed through a socio-technical mitigation: modifying the scoring of existing benchmarks that are misaligned but dominate leaderboards, rather than introducing additional hallucination evaluations. This change may steer the field toward more trustworthy AI systems.



Key message: Always verify critical outputs, especially in healthcare or scientific contexts.

Retrieval-Augmented Generation (RAG)



RAG is a method that combines LLMs with external knowledge sources.

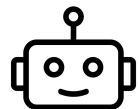


Instead of generating text solely from training data, the model can retrieve relevant information from documents, databases, or APIs.

How it works:

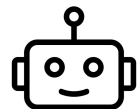
1. **Query:** The model formulates a search query based on the user prompt.
2. **Retrieve:** Relevant documents or data are retrieved from the knowledge base.
3. **Generate:** The model generates a response grounded in the retrieved information, improving accuracy and reliability.

What are the latest treatments for diabetes?



AI only

You could try berberine,
cinnamon extract, and
vitamin D
supplementation...



+



AI + RAG

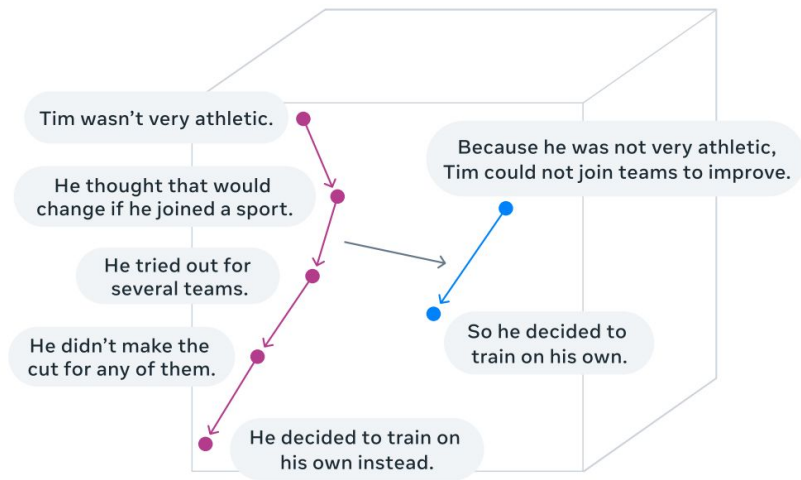
According to the 2023 ADA
guidelines, recommended
treatments include
metformin as first-line
therapy, GLP-1



Large Concept Models

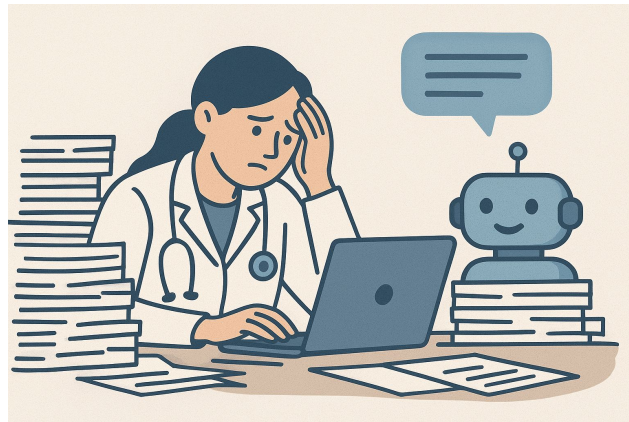
Transformers changed the game by allowing **parallel processing**. Now, a new paradigm — Large Concept Models (LCMs) — could push us even further.

Instead of predicting the next word, LCMs aim to capture entire ideas as abstract, language-agnostic embeddings (e.g., Meta's SONAR embeddings).



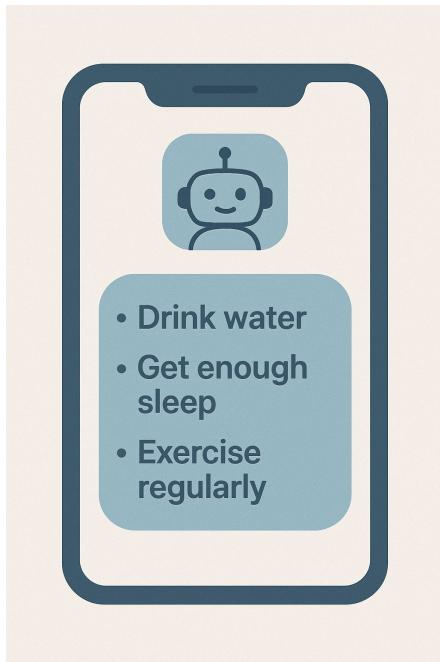
Clinical Support & Triage

Conversational models can assist clinicians in decision-making, support symptom checking, and help prioritize patients based on urgency.



Virtual Assistants for Patients & Chronic Care

LLMs enable virtual health assistants that provide reminders, answer patient questions, and support continuous monitoring in chronic disease management.



Administrative Process Automation

Conversational AI streamlines repetitive administrative tasks such as scheduling, medical documentation, and transcription, freeing up time for healthcare staff.

Administrative Process Automation

Ingredient name	CAS	Concentration
L(+)-ascorbic acid	50-81-7	90 - <100 %
sodium peroxodisulfate	7775-27-1	80 - <100 %
sulfuric acid	7664-93-9	5 - <15 %
ammonium heptamolybdate	12054-85-2	2 - <5 %

Safety Data Sheet

according to Regulations REACH 1907/2006/EC

REF: 985095

Printing date: 27.09.2023

NANOCOLOR ortho and total Phosphate LR 1

Date of issue: 26.09.2022

Page: 4/14

Version: 2.2.5.16

Possible endocrine disrupting effects

no data available

SECTION 3: Composition / information on ingredients

3.1 Substances or 3.2 Mixtures

20x 27 mg NANOFIX total Phosphate 1-45 (R3)

Substance name:

L(+)-ascorbic acid

CAS No.:

50-81-7

Substance rating:

No criteria for classification or naming of chemical not required.

Formula:

C₆H₈O₆

Pseudonym (de):

Vitamin C

REACH Reg. No.:

200-086-2

EC No.:

90 - <100 %

Concentration:

90 - <100 %

acc. CLP (GHS):

The criteria for classification are not fulfilled.

20x 35 mg NANOFIX total Phosphate 1-50 (R2)

Substance name:

sodium peroxodisulfate

CAS No.:

7775-27-1

Substance rating:

H334, Resp. Sens. 1; H302, Acute Tox. 4 oral, H315, Skin Irrit. 2; H317, Skin Sens. 1; H319, Eye Irrit. 2.

Formula:

Na₂O₈S₂

Pseudonym (de):

Nathungensulfat

REACH Reg. No.:

01-211949007-15-xxxx

EC No.:

231-852-1

Concentration:

80 - <100 %

acc. CLP (GHS):

H272, Ox. Liq. 2; H302, Acute Tox. 4 oral, H315, Skin Irrit. 2; H317, Skin Sens. 1; H319, Eye Irrit. 2; H334, Resp. Sens. 1; H335, resp. irrit. STOT SE 3

1 mL total Phosphate LR 1 (R0)

Substance name:

sulfuric acid

CAS No.:

7664-93-9

Substance rating:

H314, Skin Corr. 1 B

Formula:

H₂SO₄ (H₂O)

REACH Reg. No.:

01-2119456838-20-xxxx

EC No.:

231-626-5

Specific concentration limit:

Eye Irrit. 2; H319: 5 % ≤ C < 15 % - Skin Irrit. 2; H315: 5 % ≤ C < 15 % - Skin Corr

acc. CLP (GHS):

H314, Skin Corr. 1 B; H315, Skin Irrit. 2; H319, Eye Irrit. 2

5 mL total Phosphate 1-45 (R4)

Substance name:

ammonium heptamolybdate

CAS No.:

12054-85-2

Substance rating:

No criteria for classification or naming of chemical not required.

Formula:

H₂₄Mo₇O₄₂

Pseudonym (de):

Ammoniummolybdat

REACH Reg. No.:

01-211949007-20-xxxx

EC No.:

234-722-4

Concentration:

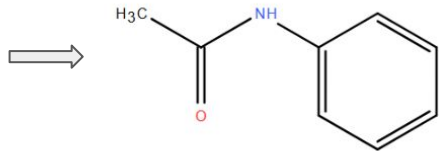
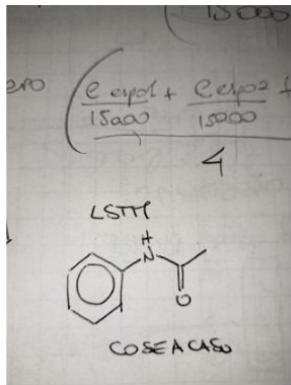
2 - <5 %

acc. CLP (GHS):

The criteria for classification are not fulfilled.

Research Support (Omics, Big Data, Biomarkers)

Generative models can accelerate research by summarizing literature, highlighting relevant findings, and helping integrate data from omics, biomarkers, and large-scale health datasets.



Resolved SMILES representation

CC(=O)NC1=CC=CC=C1

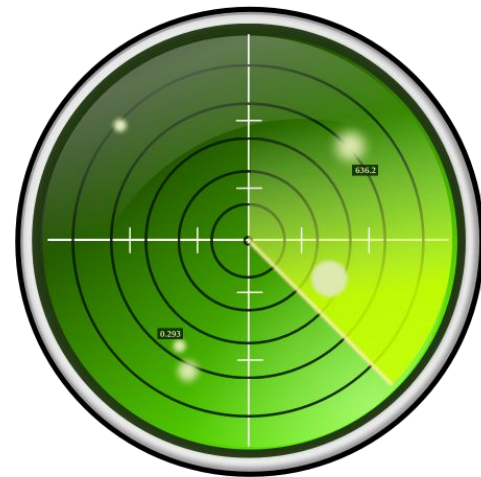
Clinical Trial Matching

- AI scans medical records and trial databases in seconds
- Suggests the most suitable trials for each patient
- Cuts down bureaucracy, speeds up enrollment



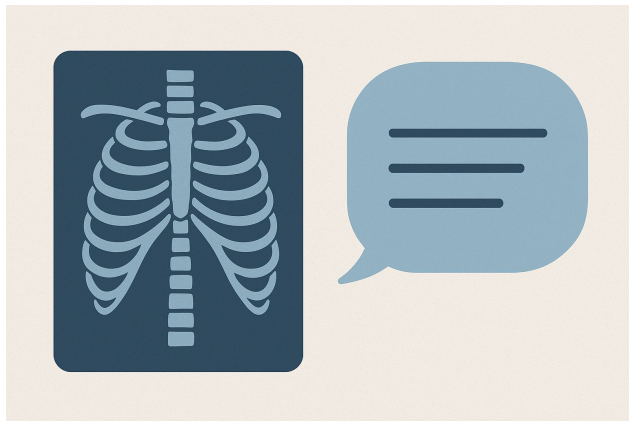
Pharmacovigilance

- AI analyzes thousands of safety reports instantly
- Detects early signals of adverse events
- Supports faster, safer decisions for drug monitoring



Radiology Reports

- Transforms complex radiology reports into plain language
- Keeps clinical accuracy while improving clarity
- Enhances doctor–patient communication



Ethical, Regulatory & Practical Challenges

Protecting sensitive patient data, ensuring compliance.

Privacy & GDPR



**Trustworthy
AI**

Bias & Clinical Safety

Avoiding harmful errors and inequities in healthcare decisions.

Human-in-the-loop

Ensuring reliability by keeping professionals in control.

Ethical, Regulatory & Practical Challenges

Real-World Cases

Patient Data Privacy

A hospital implementing an AI diagnostic tool ensures GDPR compliance to prevent unauthorized access to sensitive patient records.






Bias & Fairness

An AI system trained on historical clinical data misclassifies certain patient groups. Human review is essential to detect and correct bias.

Human-in-the-Loop

An AI system analyzes patient lab results and flags potential critical conditions, but doctors review and validate all alerts before any treatment decisions are made.

TakeHomeMessage

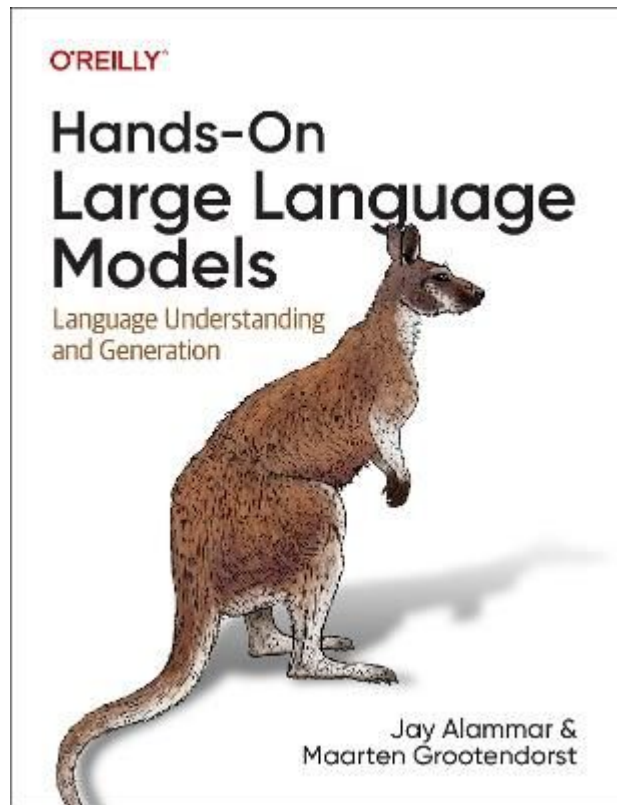
-  **AI is powerful, but not magic** → it works best when grounded in real, high-quality data.
-  **Know the limits** → hallucinations, bias, and lack of transparency require human oversight.
-  **Use it where it adds value** → automation of routine tasks, support in decision-making, faster access to knowledge.
-  **Keep humans in the loop** → AI should assist, not replace, clinical expertise.
-  **Responsible use matters** → transparency, ethics, and regulation are key to trust.

Suggested Books

THE BEST TODAY

Alammar J., Grootendorst M. (2024). Hands on large language models. O'Reilly.

<https://learning.oreilly.com/library/view/hands-on-large-language/9781098150952/>

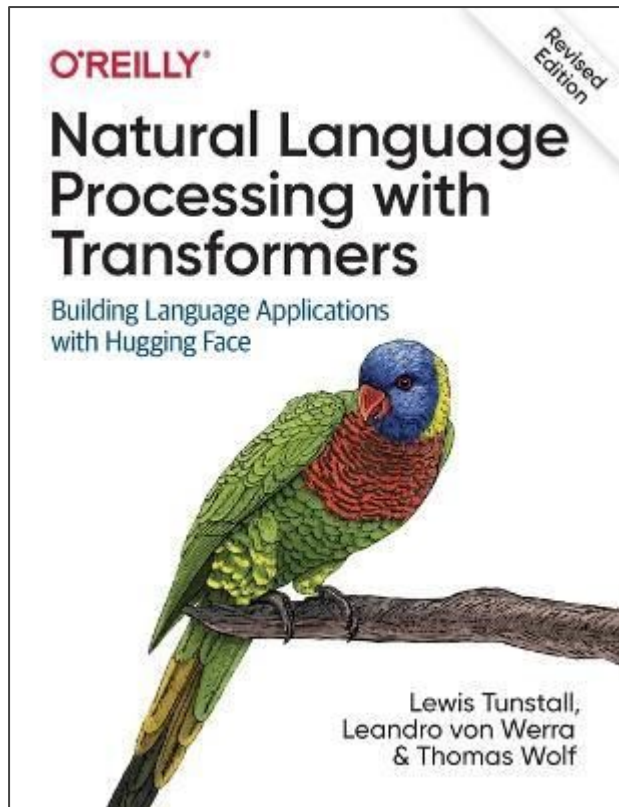


Suggested Books

THE BEST YESTERDAY

Tunstall L., von Werra L., & Wolf T (2022). Natural Language Processing with Transformers (Revised Edition). O'Reilly.

<https://www.oreilly.com/library/view/natural-language-processing/9781098136789/>



Papers

Vaswani A. et al. (2017). Attention Is All You Need.

<https://arxiv.org/abs/1706.03762>

Devlin J. et al. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

<https://arxiv.org/abs/1810.04805>

Insights

Alammar J. (2018). The Illustrated Transformer.

<https://jalammar.github.io/illustrated-transformer/>

Rohrer B. (2021). Transformers from Scratch.

<https://e2eml.school/transformers.html>

Harvard NLP (2018). The Annotated Transformer.

<https://nlp.seas.harvard.edu/2018/04/03/attention.html>

Insights

Ruder, S. (2018). A Review of the Neural History of Natural Language Processing.
<https://www.ruder.io/a-review-of-the-recent-history-of-nlp/>

Affek S. P. (2019). Document Embedding Techniques – A review of notable literature on the topic. TDS Archive.
<https://medium.com/towards-data-science/document-embedding-techniques-fed3e7a6a25d>

Q&A