From Deep Learning to ChatGPT: l’evoluzione dell’Intelligenza Artificiale, dai modelli profondi al linguaggio naturale

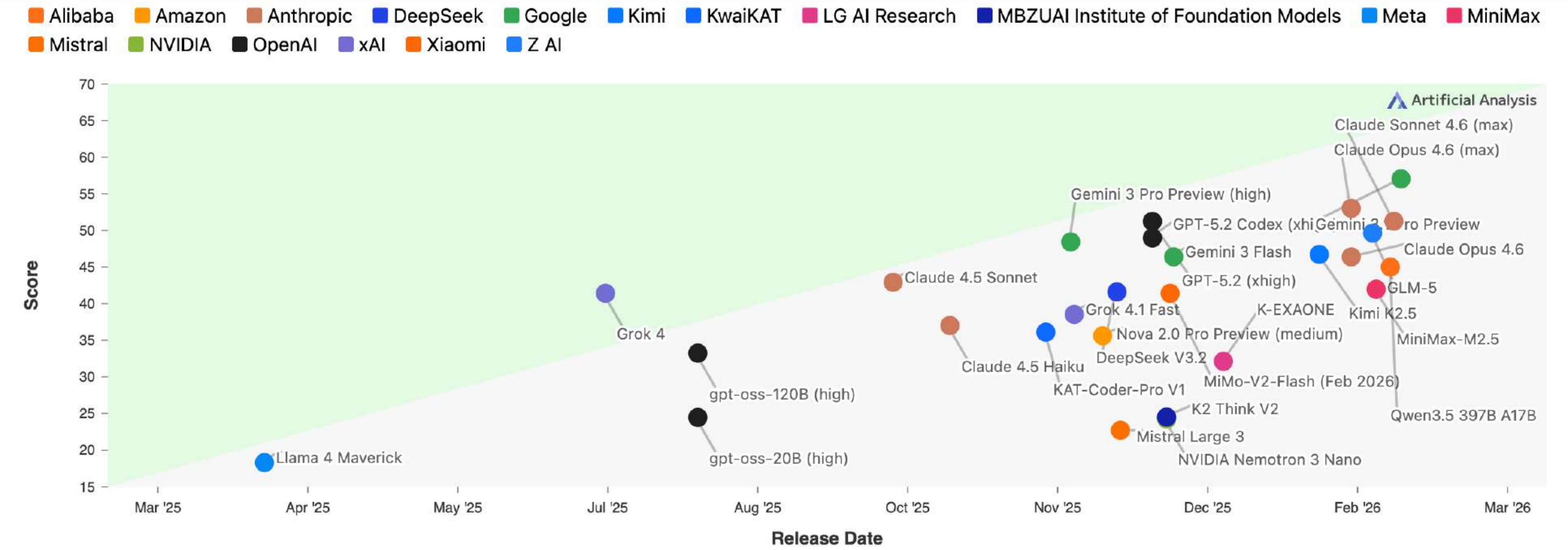
Seconda parte

Prospettive Informatiche

Prof. Misael Mongiovì

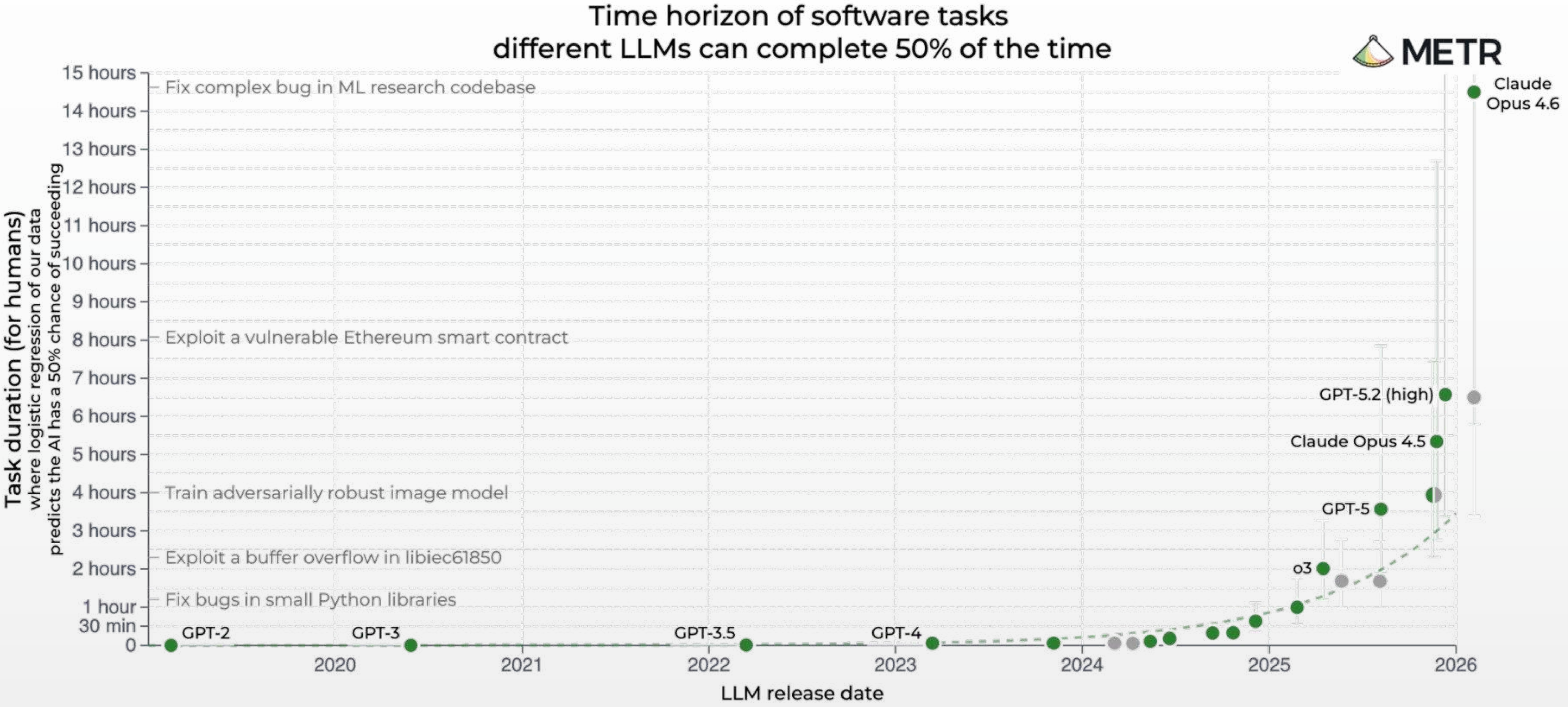
misael.mongiovi@unict.it

LLMs evolution in the last year



https://artificialanalysis.ai/evaluations/artificial-analysis-intelligence-index

LLMs Ability to Complete Long Tasks Doubles every 7 Months



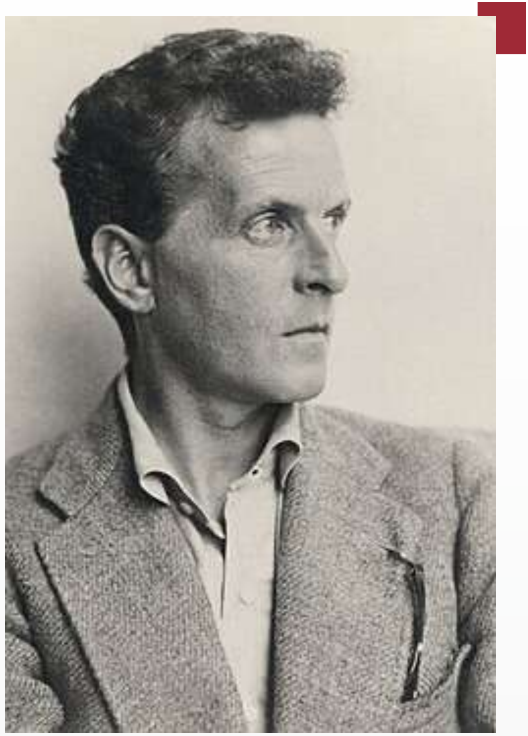
Outline

- Distributional hypothesis and word representation
- Language Modelling
- Large Language Models
- LLMs today and their future

Distributional hypothesis

"The meaning of a word is its use in the language”

Ludwig Wittgenstein



"You shall know a word by the company it keeps”

John Rupert Firth

What does recent English borrowing “*ongchoi*” mean?

- Suppose you see these sentences:
  - Ongchoi is delicious **sautéed with garlic**.
  - Ongchoi is superb **over rice**
  - Ongchoi **leaves** with salty sauces
- And you've also seen these:
  - …spinach **sautéed with garlic over rice**
  - Chard stems and **leaves** are **delicious**
  - Collard greens and other **salty** leafy greens
- Conclusion:
  - Ongchoi is a leafy green like spinach, chard, or collard greens
    - We could conclude this based on words like "leaves" and "delicious" and "sauteed"

Ongchoi: *Ipomoea aquatica "Water Spinach"*

空心菜

kangkong

rau muống

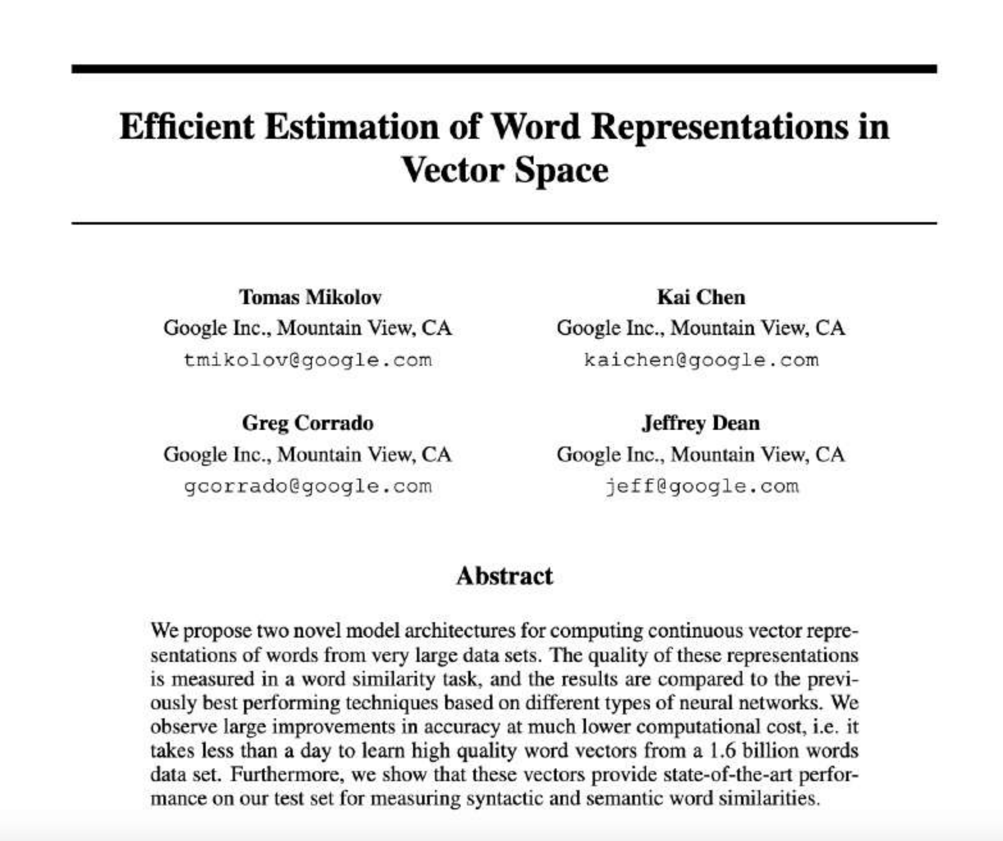
…

Yamaguchi, Wikimedia Commons, public domain

Idea 1: Defining meaning by linguistic distribution

Idea 2: Meaning as a vector in a multidimensional space (2013)

Tomáš Mikolov



Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

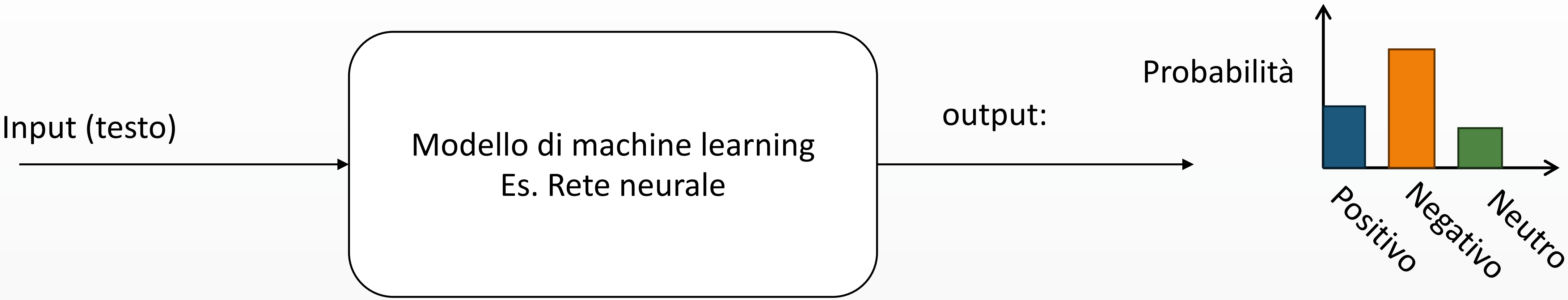
We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

We define the meaning of a word as a vector

- Called an "embedding" because it's embedded into a space
- The standard way to represent meaning in NLP
- **Every modern NLP algorithm uses embeddings as the representation of word meaning**
- Fine-grained model of meaning for similarity

Cosa ci facciamo con in word embeddings?

Esempio di utilizzo dei word embeddings: Classificazione di testo



- Distribuzione di probabilità: per ogni classe possibile il modello fornisce una stima di probabilità che quella sia la classe corretta
  - Ogni probabilità è un valore compreso tra 0 e 1
  - La somma delle probabilità assegnate a tutte le classi è 1

Probabilistic Language Modeling

- Goal: compute the probability of an upcoming word given previous words:

  $P(w_5|w_1,w_2,w_3,w_4)$

- Related task: compute the probability of a sentence or a sequence of words :

  $P(W) = P(w_1,w_2,w_3,w_4,w_5\ldots w_n)$

- A model that computes either of these:

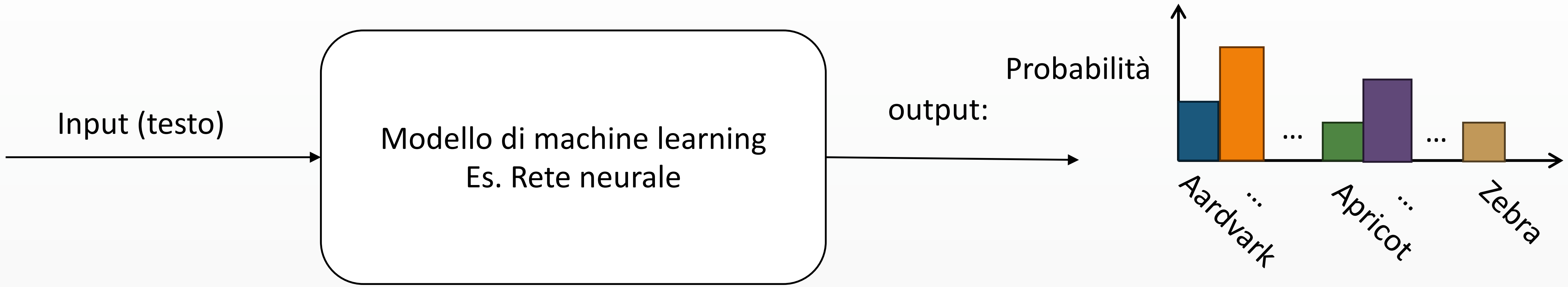
  $P(W)$ or $P(w_n|w_1,w_2\ldots w_{n-1})$ is called a **Language Model (LM)**

Example

- Given the text “Tart with apricot…”, we want to predict the next word (token):

P(“jam” | “Tart”, “with”, “apricot”)	we want it to be high
P(“sousage” | “Tart”, “with”, “apricot”)	we want it to be very low
P(“satellite” | “Tart”, “with”, “apricot”)	we want it to be very low

Implementazione di un language model



Input (testo)

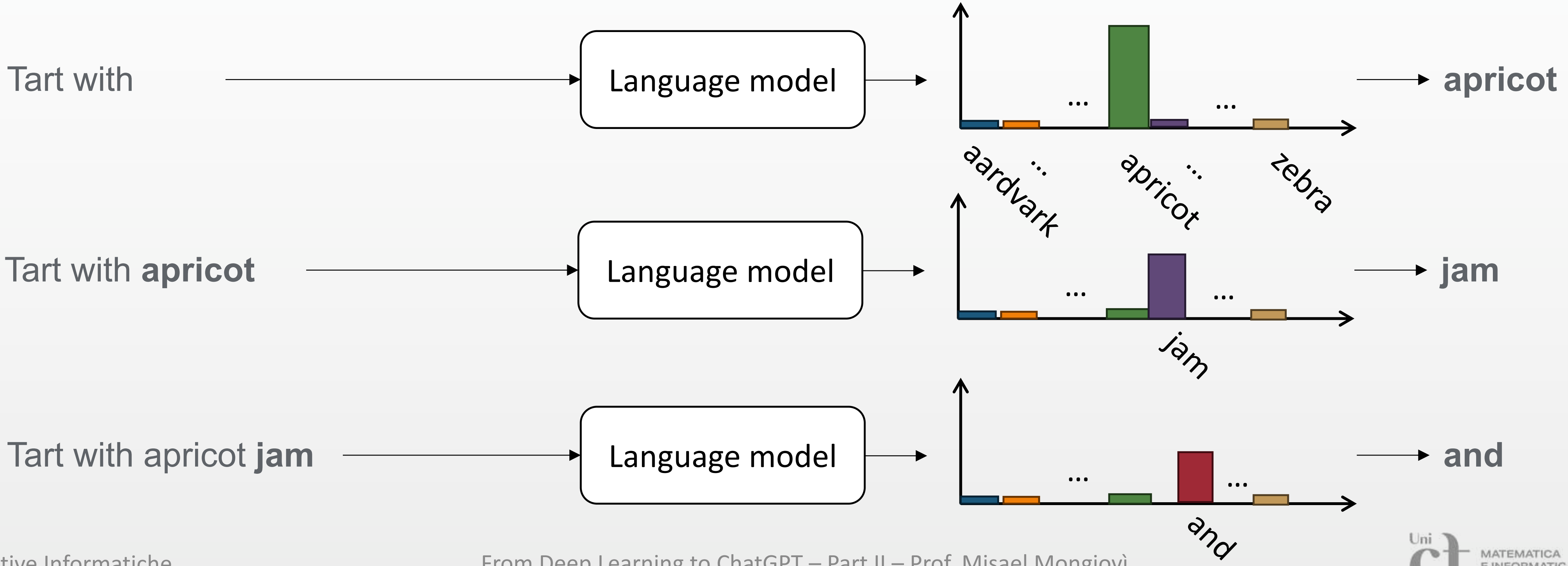
Modello di machine learning
Es. Rete neurale

output:

- Dato del testo in input, il modello stima la *distribuzione di probabilità* della parola successiva su tutto il vocabolario
- **Training**: dato un grande corpus di testo, considero ciascuna parola come una parola da predire (output) date tutte le parole precedenti (input)

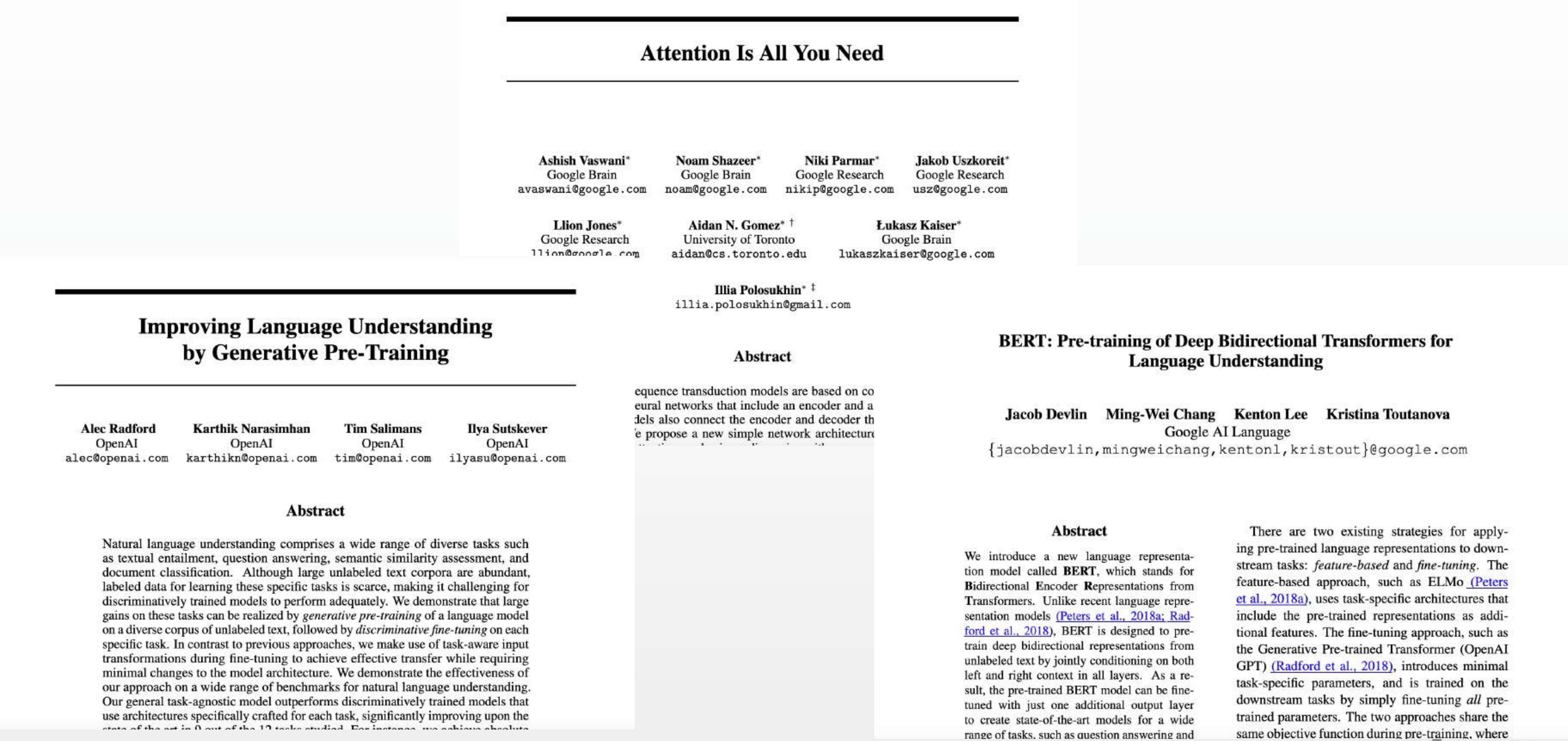
Language model for text generation

- Take as predicted word the highest probable or sample (chose randomly) from the probability distribution distribution
- Add the predicted word to the input and continue predicting the same one



Come implementare un language model efficace?

Come implementare un language model efficace?



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
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

equence transduction models are based on co
eural networks that include an encoder and a
dels also connect the encoder and decoder th
'e propose a new simple network architecture

Improving Language Understanding by Generative Pre-Training

Alec Radford OpenAI alec@openai.com
Karthik Narasimhan OpenAI karthikn@openai.com
Tim Salimans OpenAI tim@openai.com
Ilya Sutskever OpenAI ilyasu@openai.com

Abstract

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by *generative pre-training* of a language model on a diverse corpus of unlabeled text, followed by *discriminative fine-tuning* on each specific task. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. For instance, we achieve absolute

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pretrained parameters. The two approaches share the same objective function during pre-training, where

Architetture Transformer

- BERT-like
  https://github.com/google-research/bert
  (Auto-encoding transformer models)
- GPT-like
  https://github.com/openai/gpt-2
  (Auto-regressive transformer models)
- BART-like
  https://github.com/pytorch/fairseq/tree/main/examples/bart
  (Seq2Seq transformer models)

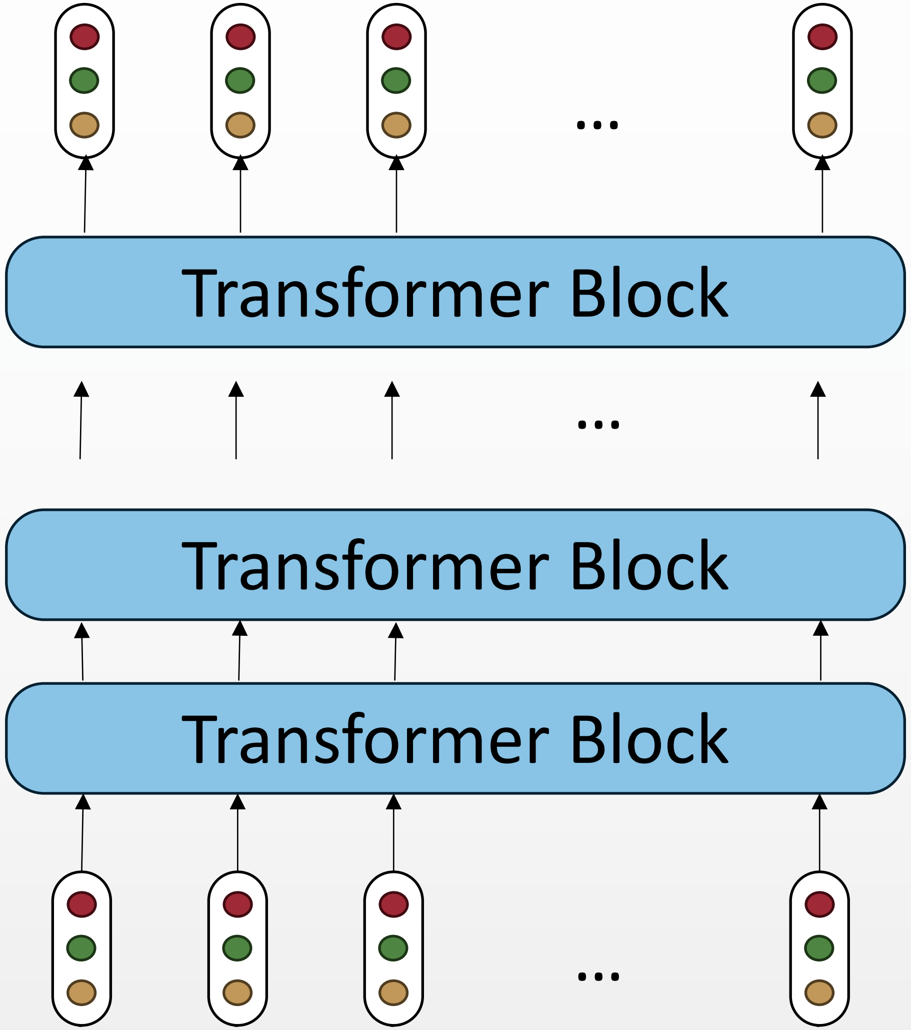
https://github.com/huggingface/transformers

Transformers

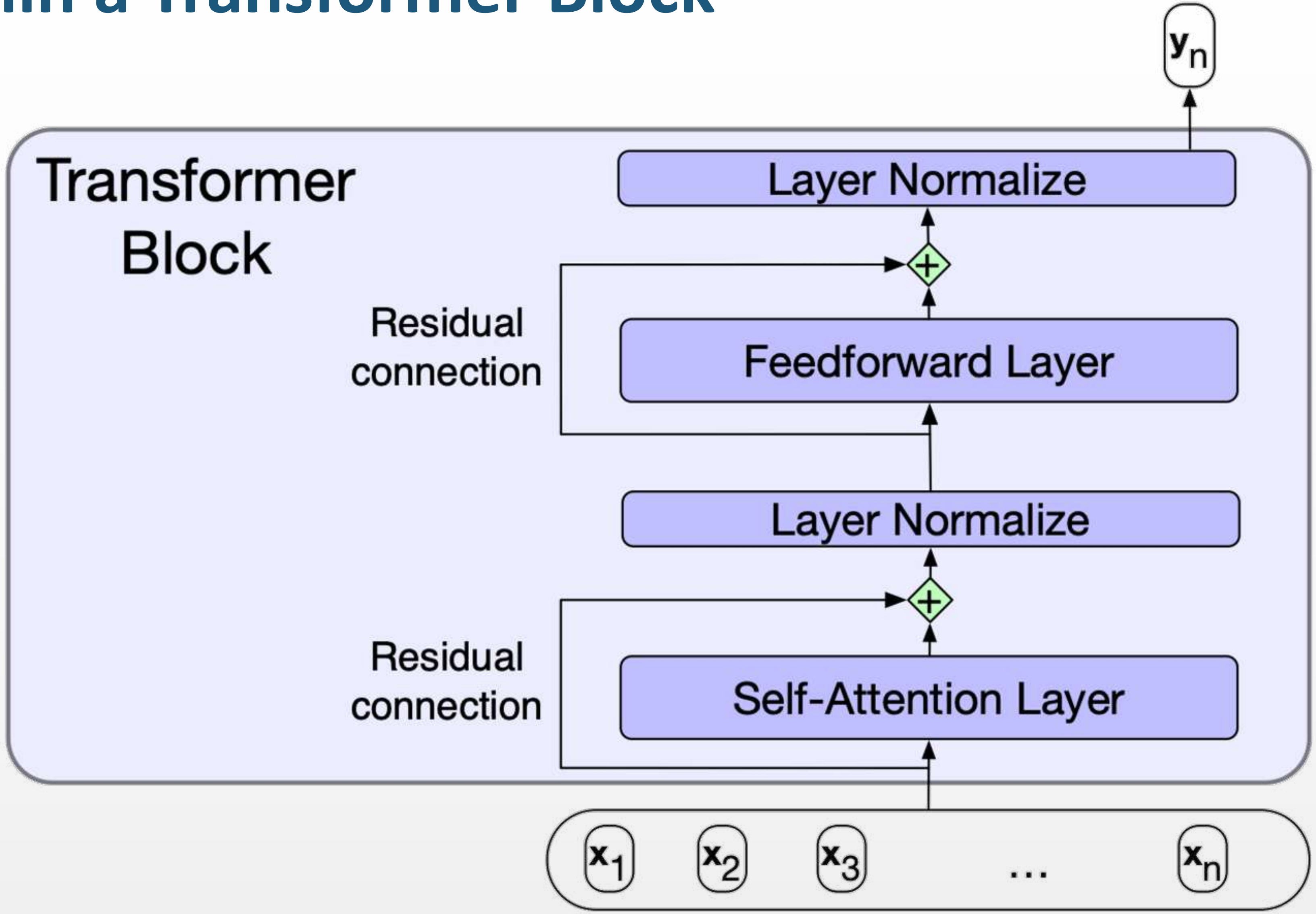
- BERT-like
  https://github.com/google-research/bert
  (Auto-encoding transformer models)
- GPT-like
  https://github.com/openai/gpt-2
  (Auto-regressive transformer models)
- BART-like
  https://github.com/pytorch/fairseq/tree/main/examples/bart
  (Seq2Seq transformer models)

https://github.com/huggingface/transformers

Stack of «Transformer Blocks»

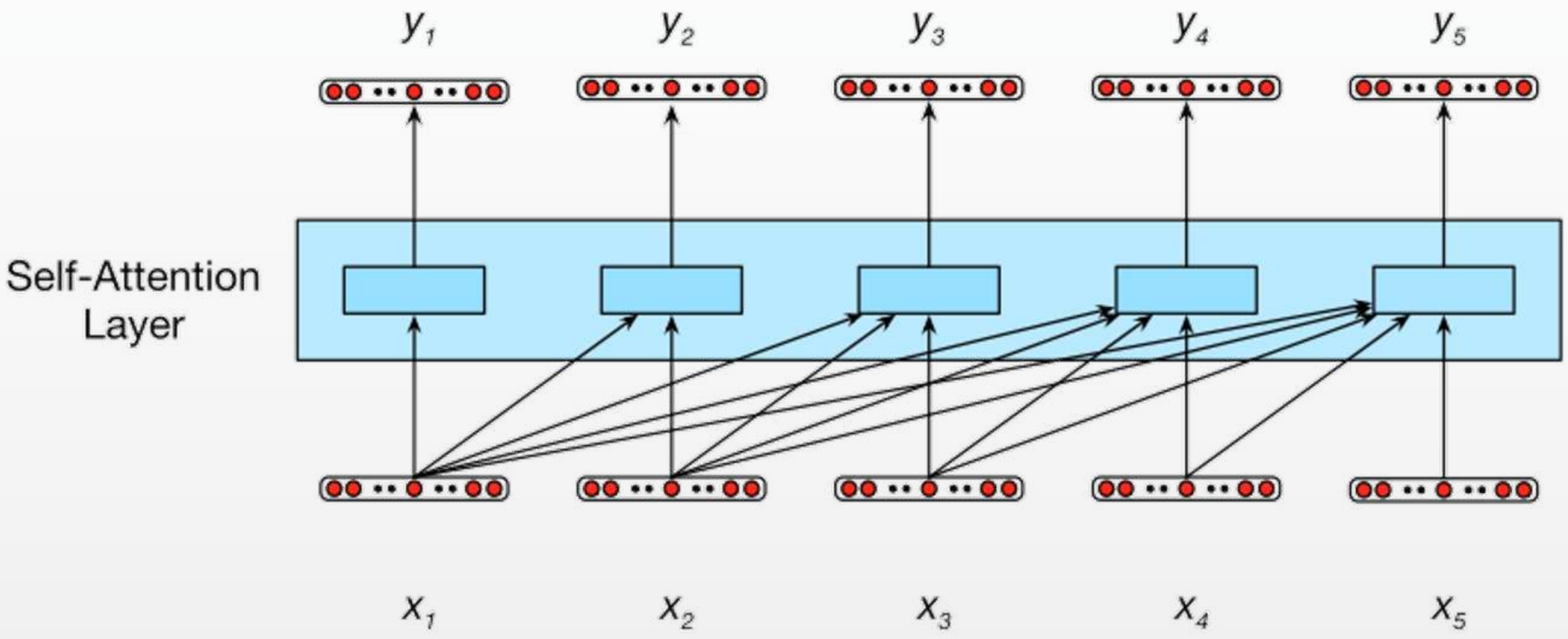


Within a Transformer Block

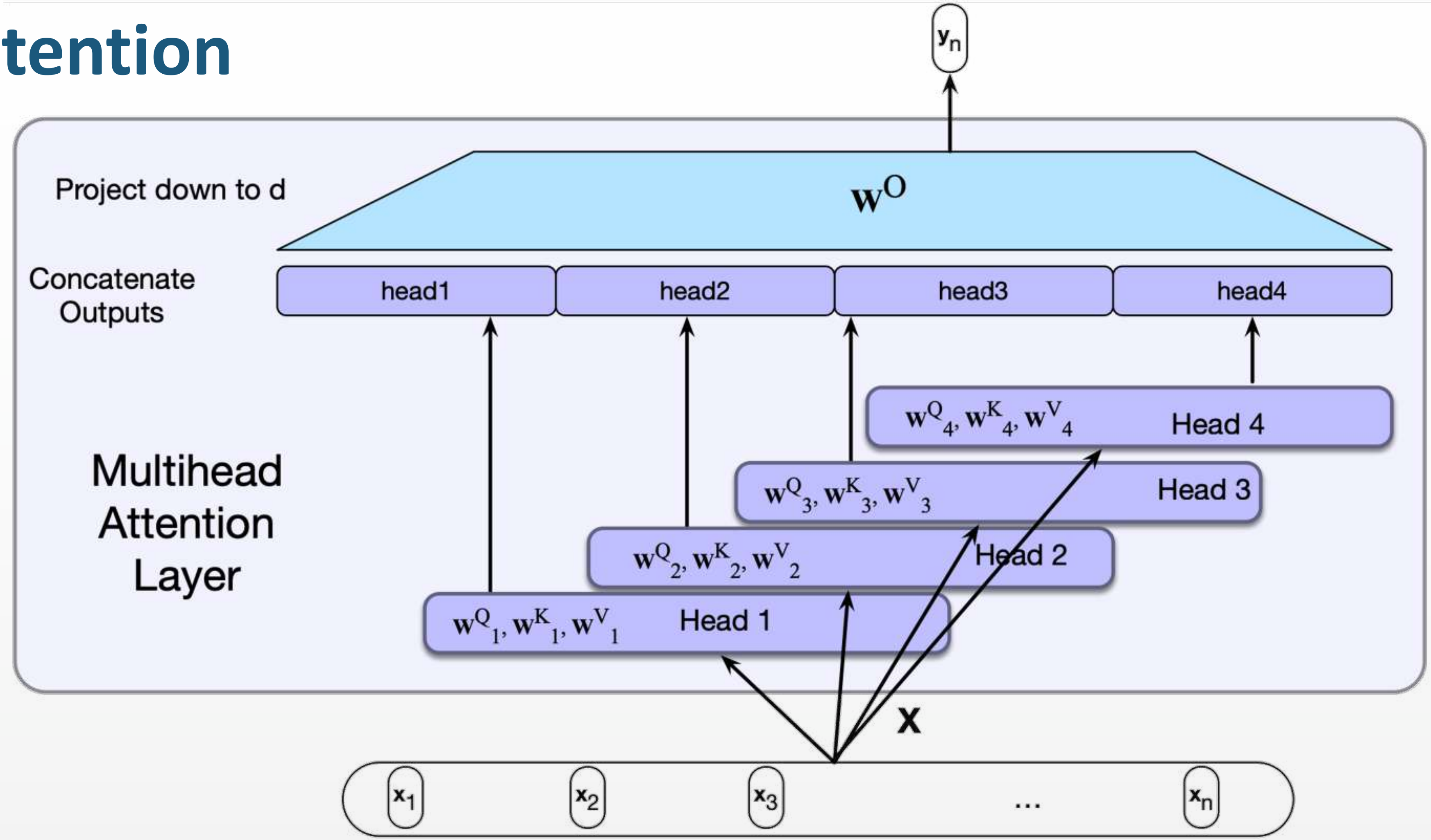


Self-attention layer

- Transformers are based on the attention mechanism. The main component is the self-attention layer



Multi-head attention

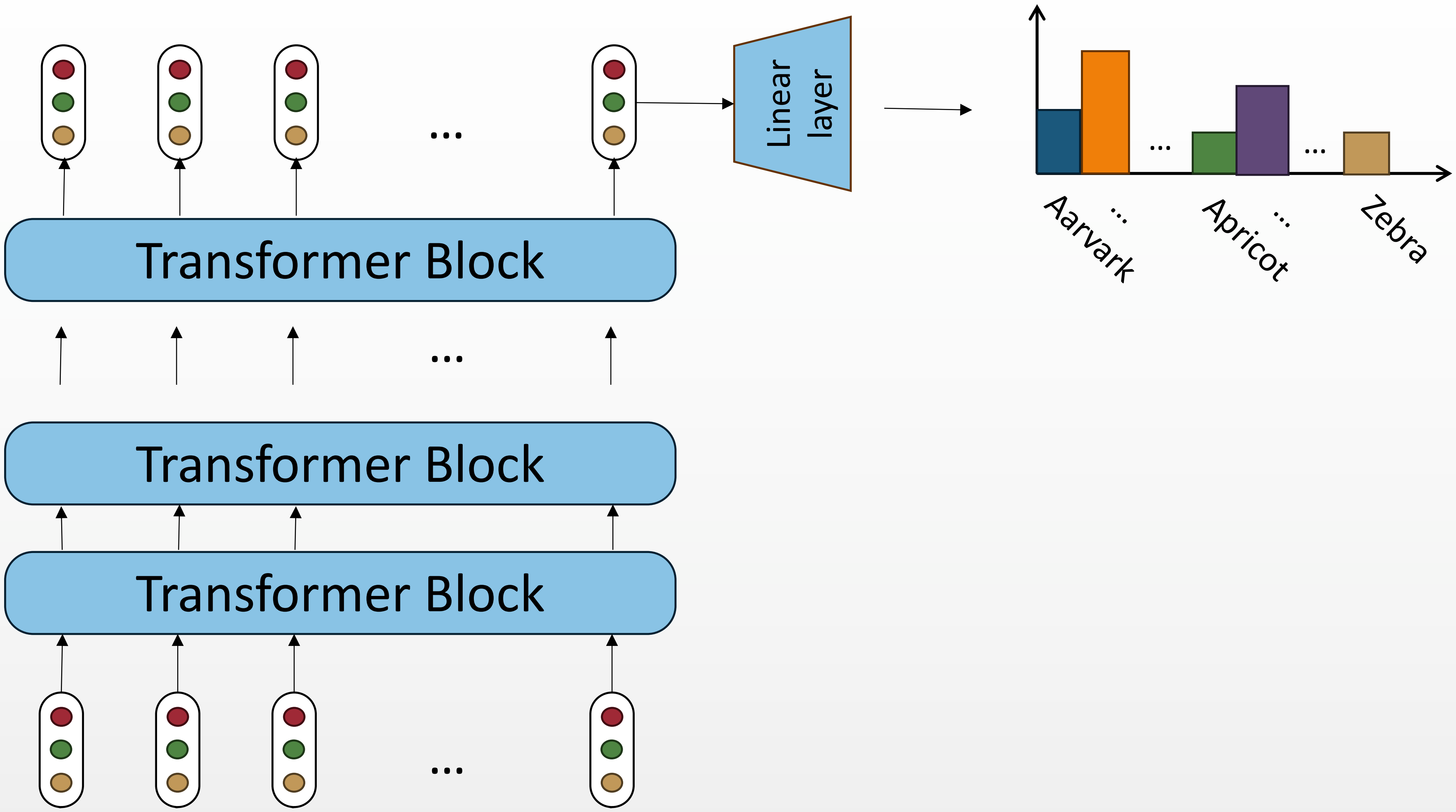


The output of each head is of size d/h

Transformers as language models

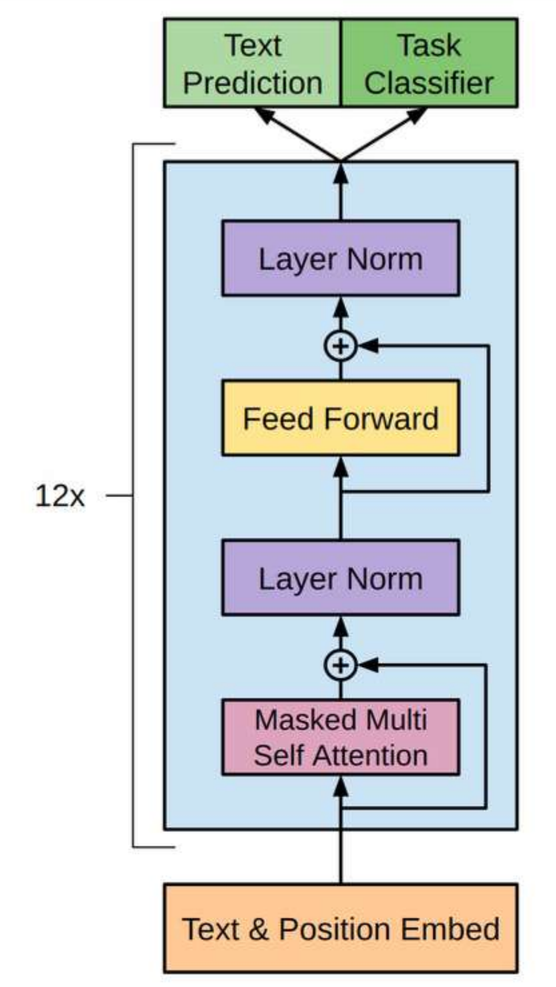
- Transformers can be used as Language Models
- We train the trasformer to predict the next token, i.e. the output token is the next token of the input
- For text generation we start from the «start» token, predict the next token, add it to the input and repeate

Transformer to predict the next word



Generative Pretrained Transformer (GPT-1) architecture

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).



- Model specifications
  - 12 transformer blocks (layers)
  - 12 attention heads
  - Hidden vector (embedding) size D=768

Setup GPT-1

- Unsupervised pre-training
  - Bookcorpus dataset
    - 7,000 books, about 1B words
- Fine-tuning (again training but for a specific task):
  - Classification
  - Entailment
  - Text similarity
  - Multiple Choice Question Answering

Performances of GPT-1 (2018)

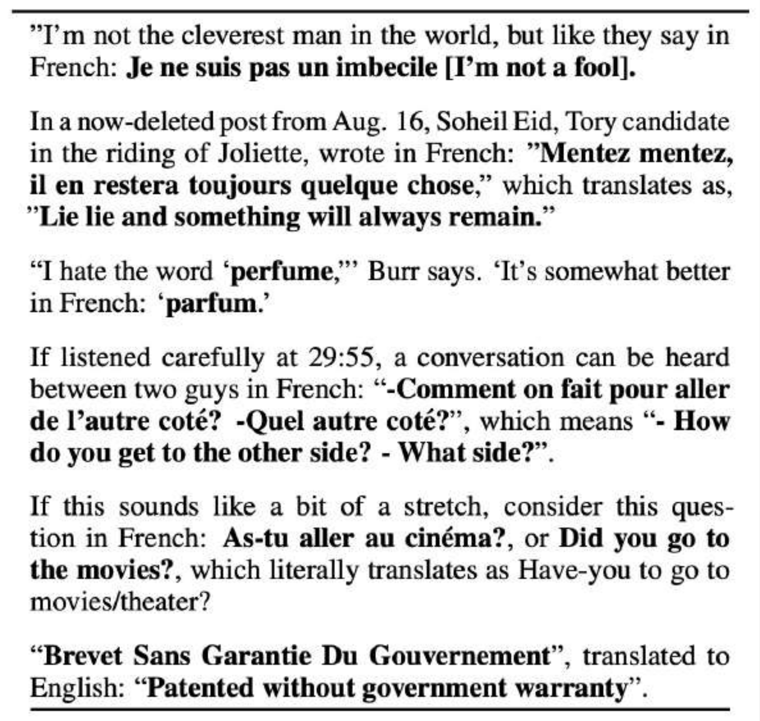
- Achieved state-of-the-art performances on several NLP tasks
  - Natural Language Inference (Entailment)
  - Question Answering and Commonsense Reasoning
  - Semantic Similarity
  - Classification (grammatical check and sentiment)

Language models are unsupervised multitask learner (GPT-2)

Radford, Alec, et al. "Language models are unsupervised multitask learners." *OpenAI blog* 1.8 (2019): 9.

“Our speculation is that a language model with sufficient capacity will begin to learn to infer and perform the tasks demonstrated in natural language sequences in order to better predict them, regardless of their method of procurement.”

Examples of translations occurring in the WebText training set



”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: **”Mentez mentez, il en restera toujours quelque chose,”** which translates as, **”Lie lie and something will always remain.”**

“I hate the word ‘**perfume,**’” Burr says. ‘It’s somewhat better in French: ‘**parfum.**’

If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

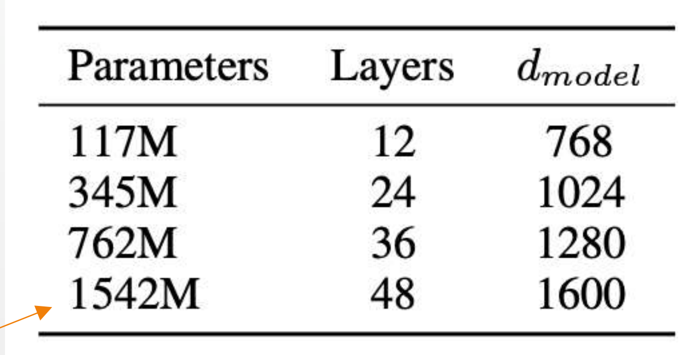
“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

Definitions

- Zero-shot learning: the model is asked to predict classes that have not been observed during training
    - It might be the same task or a different task
- One-shot learning: the model is given only one sample for the specific task
- Few-shot learning: the model is given a few samples
    - Typically, the weights of the model are not updated (no fine-tuning)

GPT-2

- Training on a new web dataset: WebText
  - Scraped all outbound links from Reddit which received at least 3 karma
  - 8 million documents
  - 40 GB of text
  - Wikipedia excluded (avoid overlapping data between train and test)
  - Byte Pair Encoding tokenization (with slight changes)
- 4 models
  - Same architecture as GPT-1 (with slight changes)
  - Vocabolary expanded to 50.257
  - Context size increased to 1024 tokens
  - Batch size 512



Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

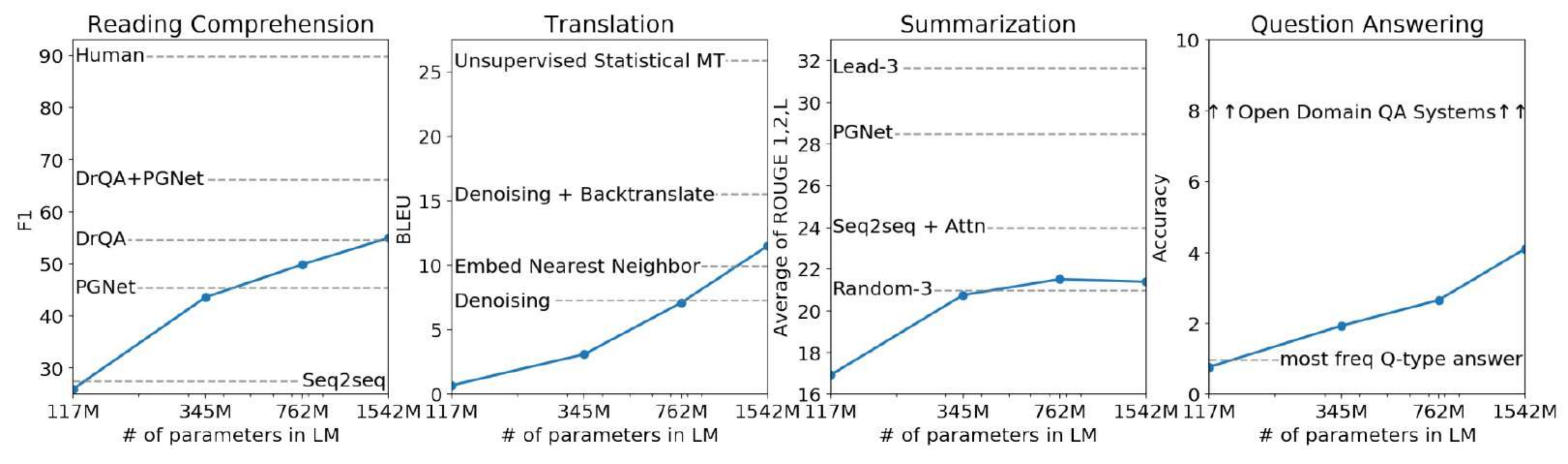
GPT-2

Experimental analysis

- Tasks are performed zero-shot
  - No specific training for the task (i.e. no fine-tuning)
- Tasks:
  - Reading Comprehension
    - The input is given as a document and a conversation with questions (Q:) and answers (A:) about the document. The final token “A:” triggers the generation of the answer
  - Summarization
    - The input is given as the document + “TL:DR” (too long, didn’t read)
  - Translation
    - A set of example pairs “english sentence = french sentence” followed by “english sentence =”
  - Question answering

Performances

- GPT-2 is sometimes competitive <u>without using specific training set</u>
- Performances increase with the size of the model



Language Models are Few-Shot Learners (GPT-3)

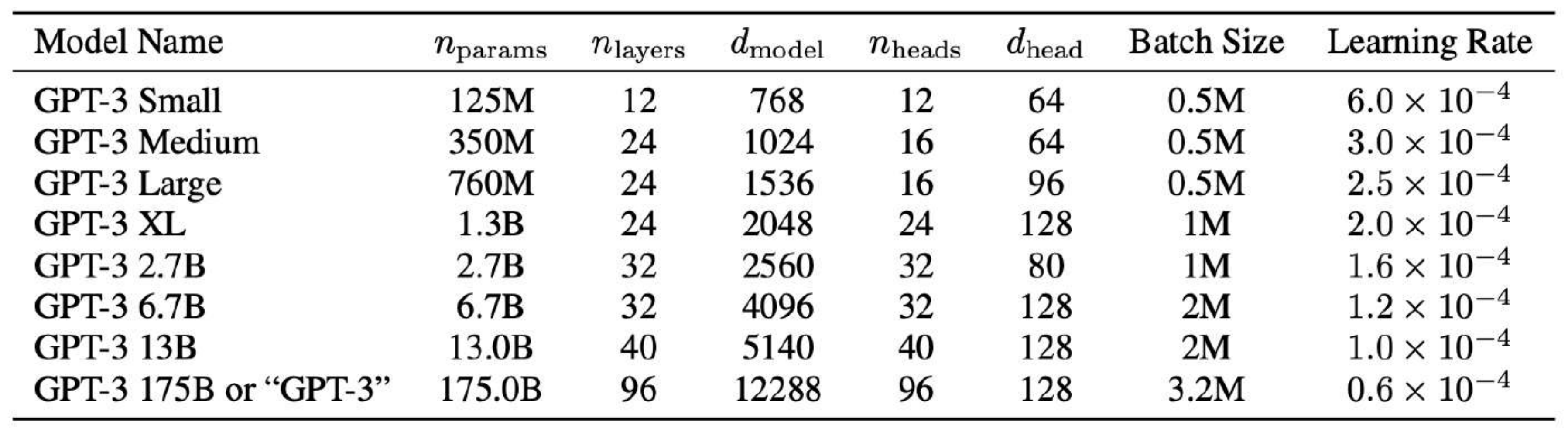
Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.

https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html?utm_medium=email&utm_source=transaction

"scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches"

GPT-3 architecture

- Mostly the same as GPT-2
- Alternate dense and locally banded sparse attention patterns in the layers of the transformer, similar to the Sparse Transformer
- Parameters:



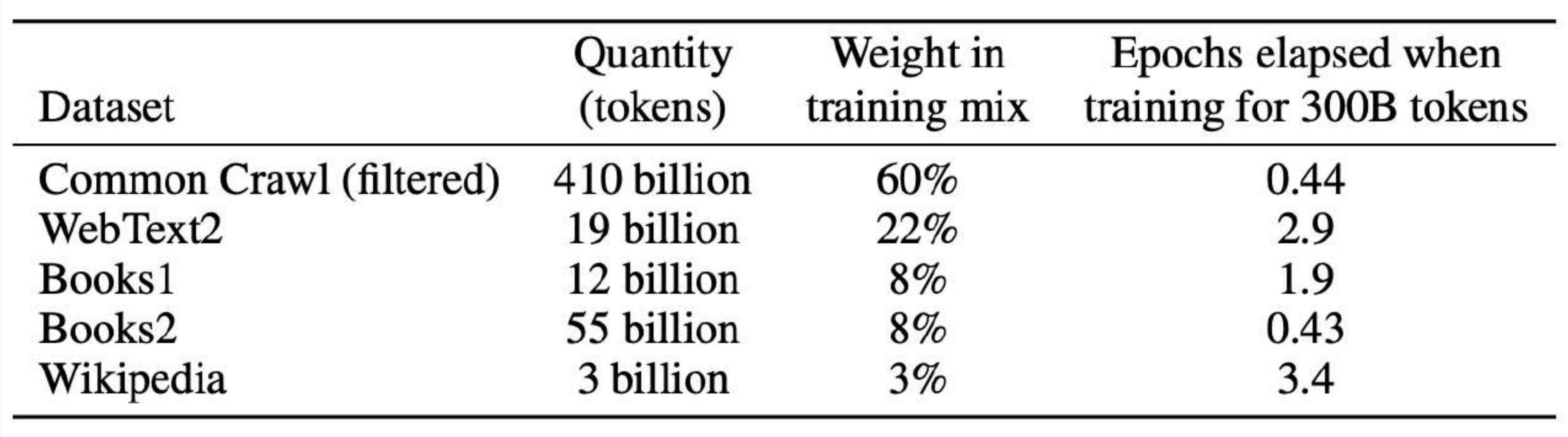
Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

# of context tokens

$$n_{\text{ctx}} = 2048$$

Training

- Hardware infrastructure
  - A mixture of model parallelism within each matrix multiply and model parallelism across the layers of the network
- Dataset
  - 300 billion tokens from:



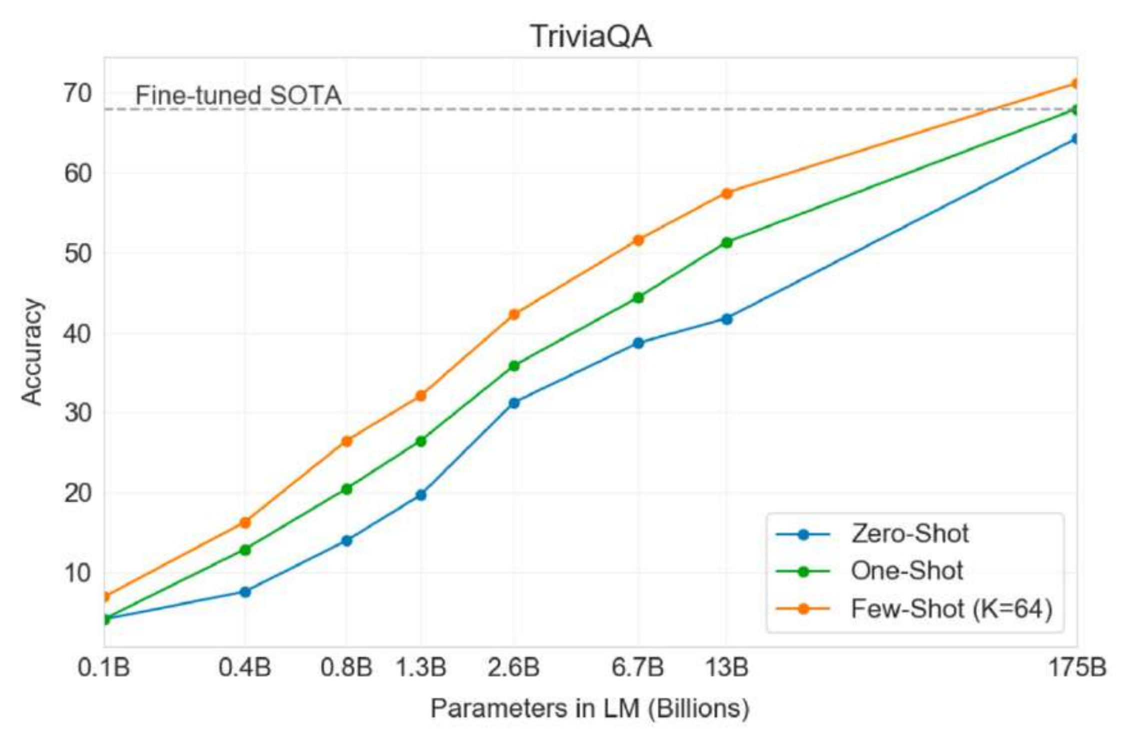
Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Common Crawl

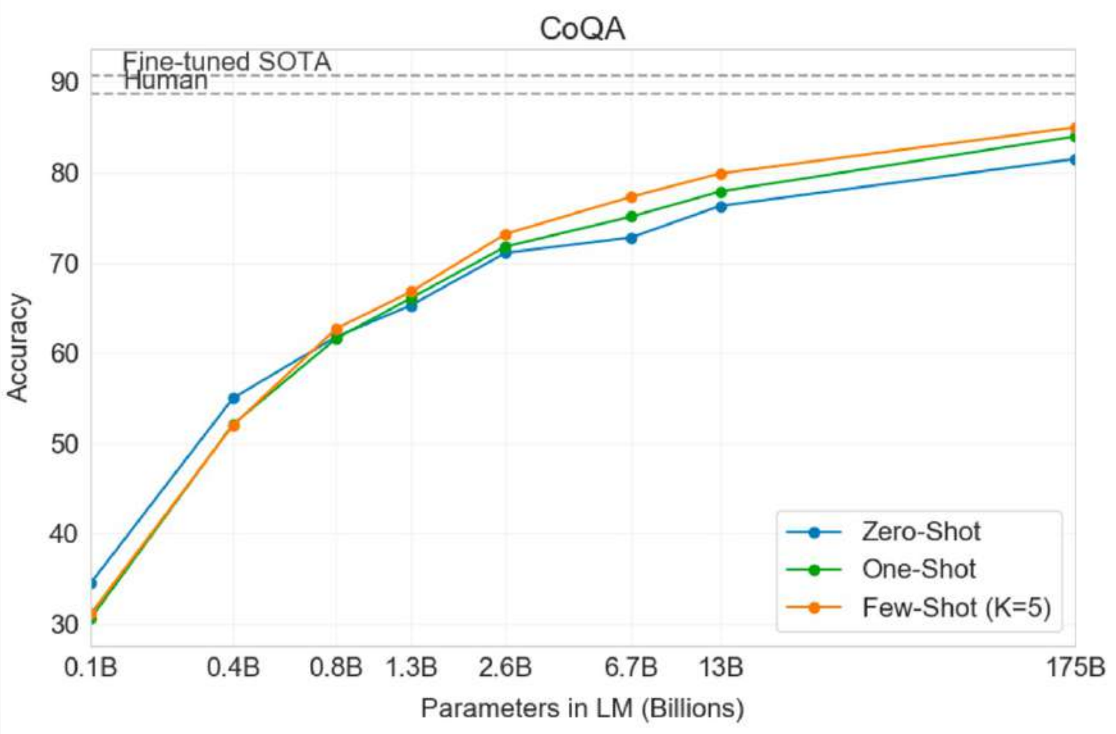
- A free, open repository of web crawl data that can be used by anyone
- Over 250 billion pages spanning 16 years
- 3–5 billion new pages (20 TB text) added each month
- The version used by GPT-3 has been cleaned to improve quality:
  - Train a classifier to distinguish WebText (high-quality documents) form raw Common Crawl and use it to sample Common Crawl by prioritizing documents which were predicted by the classifier to be high quality
  - Remove documents with high overlap with other documents

Some results

Question Answering

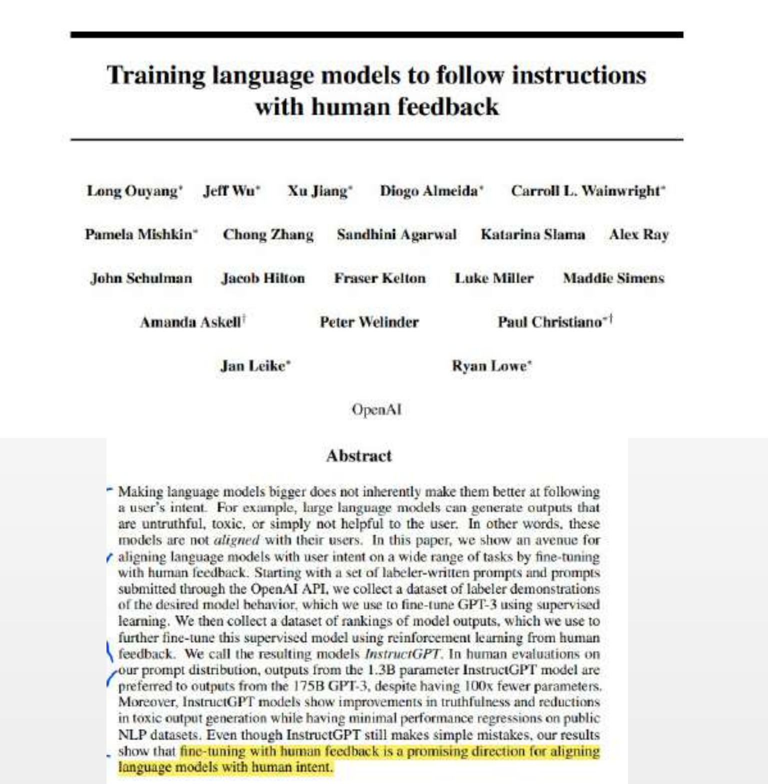


Reading comprehension



InstructGPT

- **Training language models to follow instructions with human feedback**, Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, Ryan Lowe, *Advances in Neural Information Processing Systems 35* (NeurIPS 2022)



Training language models to follow instructions with human feedback

Long Ouyang[*] Jeff Wu[*] Xu Jiang[*] Diogo Almeida[*] Carroll L. Wainwright[*]

Pamela Mishkin[*] Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell[†] Peter Welinder Paul Christiano[*†]

Jan Leike[*] Ryan Lowe[*]

OpenAI

Abstract

Making language models bigger does not inherently make them better at following a user's intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not *aligned* with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through the OpenAI API, we collect a dataset of labeler demonstrations of the desired model behavior, which we use to fine-tune GPT-3 using supervised learning. We then collect a dataset of rankings of model outputs, which we use to further fine-tune this supervised model using reinforcement learning from human feedback. We call the resulting models *InstructGPT*. In human evaluations on our prompt distribution, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters. Moreover, InstructGPT models show improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets. Even though InstructGPT still makes simple mistakes, our results show that fine-tuning with human feedback is a promising direction for aligning language models with human intent.

This is the research that led to ChatGPT

Why InstructGPT

- LLMs still generate undesired responses:
  - toxic, untruthful, harmful or simply flouting the user’s instructions.
  - In one word they are not *aligned* with the users.

- We need a method to ‘align’ the model using human feedback

Method

- InstructGPT is based on GPT-3 and fine-tuned with the following protocol:
  1. A team of human annotators is selected.
  2. The human annotators write demonstration prompts and responses.
  3. The model is fine-tuned on these demonstrations using supervised learning: SFT.
  4. The SFT model is used to generate responses.
  5. The human annotators flag and rank the responses.
  6. A Reward Model (RM) is trained on the rankings to grade the responses (i.e.: automate step 5.).
  7. The model is fine-tuned through Reinforcement Learning (RL) to maximise the reward of the RM: Proximal Policy Optimization (PPO).

- Considerations:
  - Steps 4-7 can be iterated.
  - Optimising the model for alignment with users can penalise performance on other NLP tasks (as measured on NLP datasets). To minimise degradation, a 'pre-training' term is added to the optimisation in step 7: PPO-ptx.

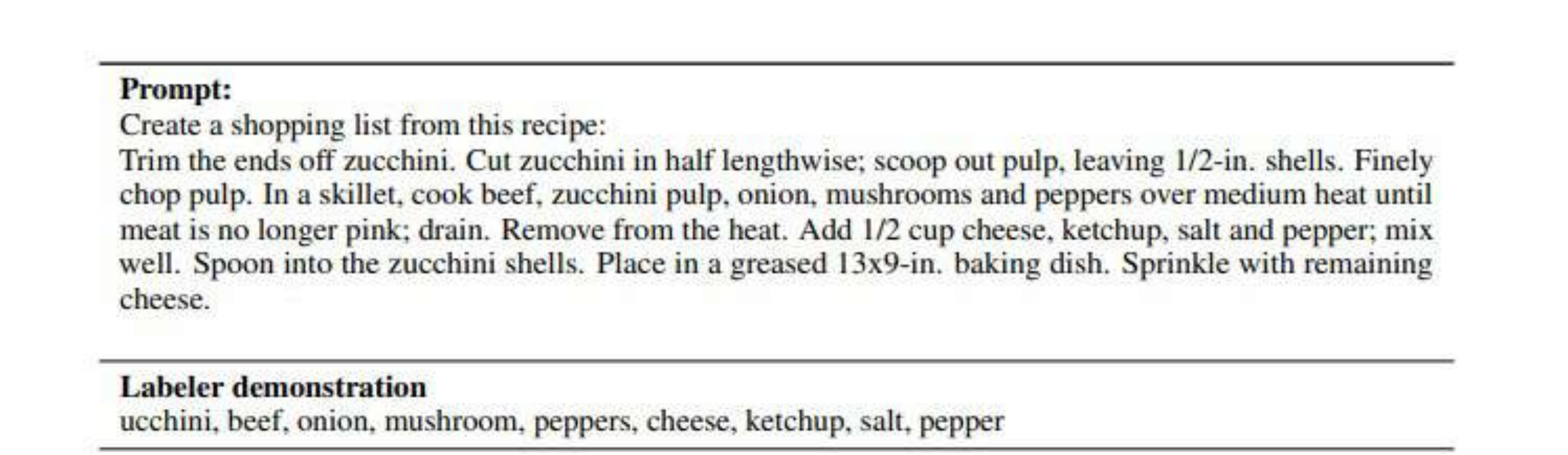
1. Annotator Selection

- A team of labellers are hired via freelance recruitment platforms.
- To ensure the labellers are themselves aligned with the researchers, they are screened with 4 criteria (sic):
    - Agreement on sensitive speech flagging
    - Agreement on rankings
    - Sensitive demonstration writing
    - Self-assessed ability to identify sensitive speech for different groups.

- The first 3 are to guarantee that the objective function of the training process broadly matches that of the researchers.
- The last point is to aim for a more diverse and representative training dataset.
- The annotators are given instructions on their task and are encouraged to exchange with the researchers.

2. Human written examples

- The annotators write demonstration prompt-response pairs to initialise the training process.



Prompt:

Create a shopping list from this recipe:

Trim the ends off zucchini. Cut zucchini in half lengthwise; scoop out pulp, leaving 1/2-in. shells. Finely chop pulp. In a skillet, cook beef, zucchini pulp, onion, mushrooms and peppers over medium heat until meat is no longer pink; drain. Remove from the heat. Add 1/2 cup cheese, ketchup, salt and pepper; mix well. Spoon into the zucchini shells. Place in a greased 13x9-in. baking dish. Sprinkle with remaining cheese.

Labeler demonstration

ucchini, beef, onion, mushroom, peppers, cheese, ketchup, salt, pepper

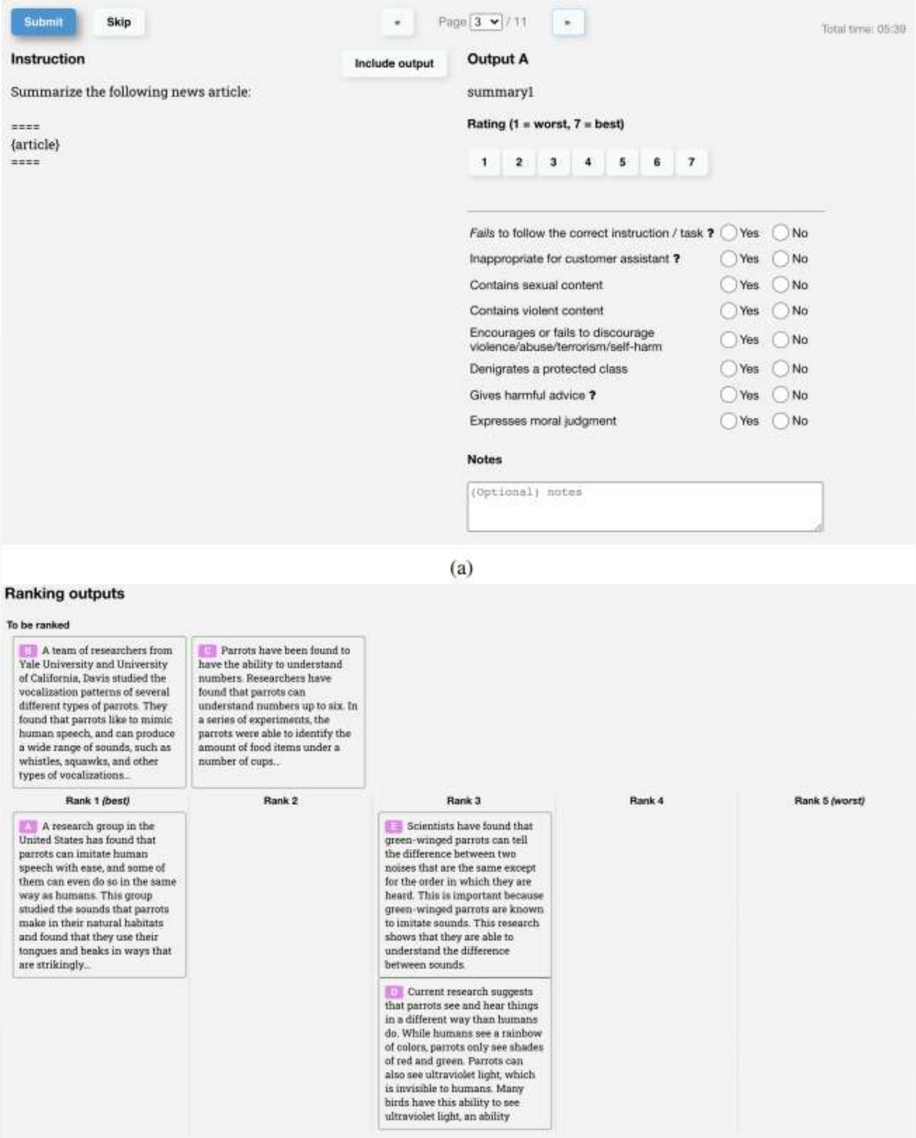
Labeler-written prompt and demonstration

3. Supervised Fine-Tuning (SFT)

- A GPT-3 model is fine-tuned on the prompt-response pairs using the standard supervised training protocol.
- The resulting model is named SFT and is used as a baseline in following experiments.

4-5. Human ranked responses

- The SFT model is used to generate multiple responses for a prompt.
- The human annotators flag the response for certain positive/negative attributes, rate the response and rank the outputs.
- The criteria for ranking are usefulness (i.e.: how well the response answers the prompt) and toxicity & harmfulness.
    - Unharmful responses are to be preferred.



Instruction

Summarize the following news article:

====
{article}
====

Output A

summary1

Rating (1 = worst, 7 = best)

Fails to follow the correct instruction / task ?	○ Yes	○ No
Inappropriate for customer assistant ?	○ Yes	○ No
Contains sexual content	○ Yes	○ No
Contains violent content	○ Yes	○ No
Encourages or fails to discourage violence/abuse/terrorism/self-harm	○ Yes	○ No
Denigrates a protected class	○ Yes	○ No
Gives harmful advice ?	○ Yes	○ No
Expresses moral judgment	○ Yes	○ No

Notes

(Optional) notes

(a)

Ranking outputs

To be ranked

B A team of researchers from Yale University and University of California, Davis studied the vocalization patterns of several different types of parrots. They found that parrots like to mimic human speech, and can produce a wide range of sounds, such as whistles, squawks, and other types of vocalizations...

C Parrots have been found to have the ability to understand numbers. Researchers have found that parrots can understand numbers up to six. In a series of experiments, the parrots were able to identify the amount of food items under a number of cups...

Rank 1 *(best)*	Rank 2	Rank 3	Rank 4	Rank 5 *(worst)*
A A research group in the United States has found that parrots can imitate human speech with ease, and some of them can even do so in the same way as humans. This group studied the sounds that parrots make in their natural habitats and found that they use their tongues and beaks in ways that are strikingly...		E Scientists have found that green-winged parrots can tell the difference between two noises that are the same except for the order in which they are heard. This is important because green-winged parrots are known to imitate sounds. This research shows that they are able to understand the difference between sounds. D Current research suggests that parrots see and hear things in a different way than humans do. While humans see a rainbow of colors, parrots only see shades of red and green. Parrots can also see ultraviolet light, which is invisible to humans. Many birds have this ability to see ultraviolet light, an ability		

6. Training reward model (RM)

- A GPT3-6B-based Reward Model is trained on the rankings to automate the task.
- The RM takes as input a prompt-response pair and outputs a value, the *reward*.

7. Reinforcement Learning From Human Feedback

- The SFT model is trained again using the Proximal Policy Optimisation[1] method.
- The objective function is designed to maximise the reward from the RM model and to minimize the discrepancy from the original SFT model

Performances

- In human evaluations, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters
- InstructGPT models show improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets

The Current State of Large Language Models

By 2025–2026, LLMs have evolved into **natively multimodal systems** with significantly enhanced reasoning capabilities. The trajectory of progress has accelerated dramatically, driven by both architectural innovation and massive computational investment.

🧠 Architectural Sophistication

Models have moved beyond monolithic dense transformers toward modular designs like Mixture-of-Experts (MoE), enabling unprecedented scale with manageable compute costs.

📖 Language Mastery

State-of-the-art models demonstrate near-human performance on language understanding benchmarks, producing coherent, contextually rich text across diverse domains and languages.

🔗 Data Integration

Modern LLMs seamlessly integrate structured and unstructured data sources, bridging the gap between raw text and multimodal signals including vision, audio, and symbolic information.

Mixture of Experts (MoE)

OPTIMIZATION & EFFICIENCY

The **Mixture of Experts** architecture represents one of the most impactful innovations in modern LLM design. Rather than activating the full parameter space for every input, MoE models route each token through a small subset of specialized "expert" sub-networks, selected dynamically by a learned gating mechanism.

This conditional computation strategy allows models to reach enormous total parameter counts — often in the hundreds of billions — while keeping the **active parameters per forward pass** a fraction of the total. The result: dramatic improvements in both computational efficiency and model generalization.

⚡ Efficiency

Sparse activation reduces FLOPs per token significantly

🎯 Specialization

Experts develop domain-specific knowledge autonomously

📈 Scalability

Scales capacity without proportional compute growth

How MoE Works

Each input token is evaluated by a **gating network** that assigns it to the top-K expert layers.

Only those K experts are activated — the rest remain dormant during that forward pass.

Prominent production MoE systems demonstrate that this approach can match or exceed dense model quality at a fraction of the training and inference cost.

Multimodality in Large Language Models

BEYOND TEXT

The frontier of LLM research has decisively expanded beyond text. **Multimodal models** are now capable of perceiving, reasoning across, and generating content in multiple signal domains simultaneously — unlocking richer human-AI interaction paradigms.

Vision

Models ingest and reason over images with fine-grained spatial understanding, enabling applications from visual QA to document analysis and scene interpretation.

Video

Temporal reasoning over video sequences allows models to track events, summarize content, and answer questions that span multiple frames and time steps.

Audio

Speech understanding and generation are natively integrated, enabling real-time transcription, translation, and spoken dialogue without pipeline fragmentation.

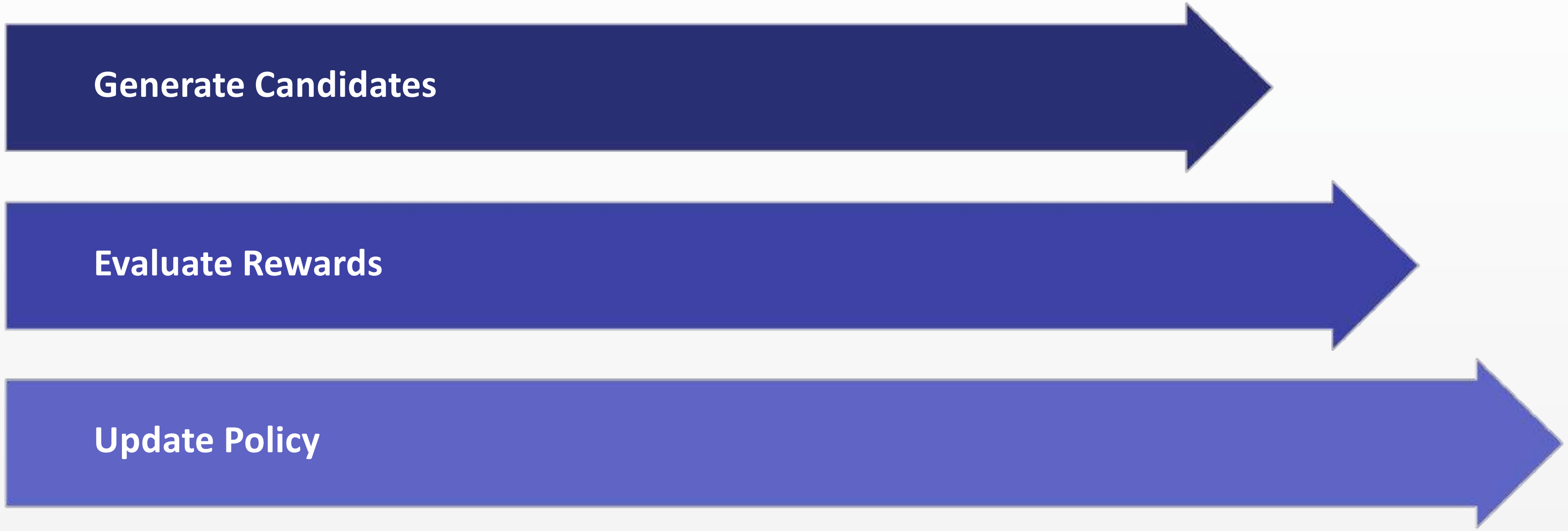
Interoperability

Aligning representations across radically different modalities remains an open research challenge, requiring unified embedding spaces and cross-modal attention mechanisms.

Reinforcement Learning from Verifiable Rewards (RLVR)

ALIGNMENT & LEARNING

RLVR represents a significant shift in how LLMs are trained to reason and produce reliable outputs. Unlike traditional RLHF, which relies on human preference signals that can be noisy or inconsistent, **RLVR grounds the reward signal in objectively verifiable criteria** — mathematical correctness, code execution results, logical consistency, or factual accuracy.



Generate Candidates

Evaluate Rewards

Update Policy

Key Advantages

- Scalable supervision without human annotation bottlenecks
- Reward signals that are consistent, reproducible, and tamper-resistant
- Improved multi-step reasoning on mathematical and coding tasks

Research Trajectory

The field is moving rapidly toward **dynamic feedback loops** where models are evaluated against real-world performance metrics rather than static labeled datasets — enabling continuous, environment-grounded improvement.

PEFT: Parameter-Efficient Fine-Tuning

OPTIMIZATION & EFFICIENCY

Full fine-tuning of large language models is computationally prohibitive for most organizations. **Parameter-Efficient Fine-Tuning (PEFT)** methods solve this by adapting a small subset of parameters — or injecting lightweight trainable modules — while keeping the majority of the pre-trained model frozen.

LoRA (Low-Rank Adaptation)

Injects low-rank decomposition matrices into attention layers. Achieves near full fine-tune performance with as few as 0.1% trainable parameters. Now the de facto standard for efficient adaptation in production systems.

HiRA & Advanced Variants

Hierarchical and structured extensions of LoRA that better capture task-specific adaptation signals across different network depths. Emerging research shows consistent gains in complex reasoning and domain transfer tasks.

Benefits for Deployment

PEFT enables rapid iteration: teams can fine-tune, evaluate, and deploy specialized model variants within hours rather than weeks. Multiple adapters can be swapped modularly on a single base model, drastically reducing infrastructure costs.

Techniques for Constraining LLM Output

SAFETY & RELIABILITY

Why Output Control Matters

As LLMs are deployed in high-stakes environments — healthcare, law, finance, and critical infrastructure — the ability to **reliably constrain and verify outputs** becomes as important as raw capability.

Unconstrained generation can produce hallucinations, policy violations, or structurally invalid outputs. The field is actively developing methods to bound model behavior without sacrificing fluency or usefulness.

Core Techniques

→ ### Constrained Decoding

Enforces grammatical or structural constraints (e.g., valid JSON, formal grammars) at inference time by masking invalid token transitions in the decoding distribution.

→ ### Reward-Guided Decoding

Integrates a reward model directly into the generation process, steering the output distribution toward higher-quality or policy-compliant responses in real time.

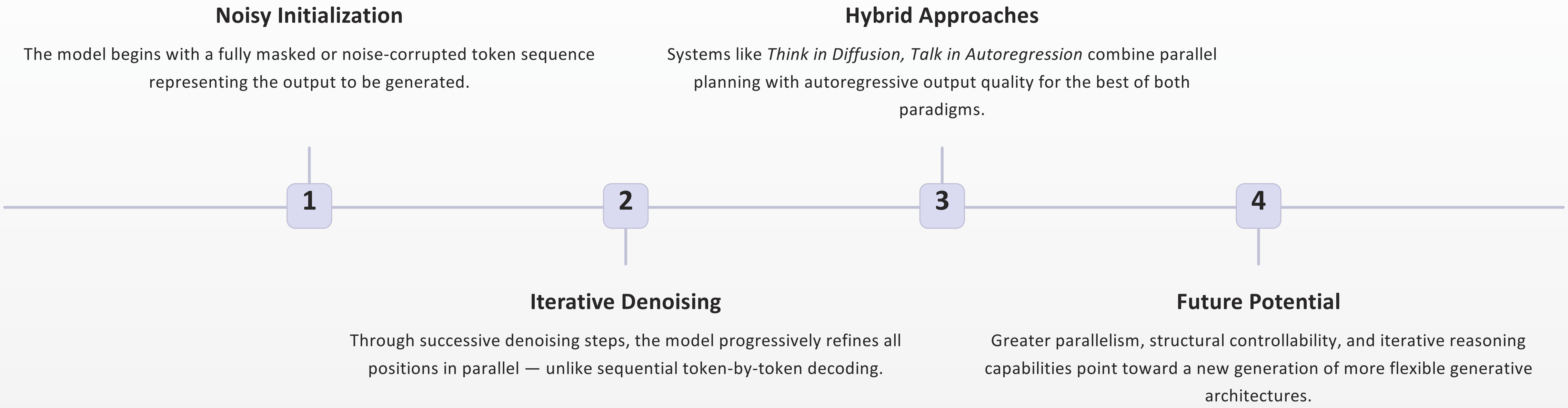
→ ### Style & Content Penalties

Applies soft or hard penalties to discourage toxic, off-topic, or stylistically inconsistent content, enabling fine-grained behavioral customization per deployment context.

Diffusion Models for Text & Reasoning

NEW GENERATIVE PARADIGM

While autoregressive transformers generate text token-by-token, **diffusion language models** take a fundamentally different approach: they refine an entire sequence iteratively through a denoising process, enabling parallel generation and novel forms of structural control.



Noisy Initialization

The model begins with a fully masked or noise-corrupted token sequence representing the output to be generated.

Iterative Denoising

Through successive denoising steps, the model progressively refines all positions in parallel — unlike sequential token-by-token decoding.

Hybrid Approaches

Systems like *Think in Diffusion, Talk in Autoregression* combine parallel planning with autoregressive output quality for the best of both paradigms.

Future Potential

Greater parallelism, structural controllability, and iterative reasoning capabilities point toward a new generation of more flexible generative architectures.

Recent models such as **Diffusion-NAT** and **TESS** (2023–2025) integrate discrete diffusion with pre-trained LLM representations, demonstrating improved coherence and generation quality compared to earlier non-autoregressive baselines.

Agentic AI & Intelligent Agent Models

THE NEXT FRONTIER

The most transformative shift on the LLM horizon is the transition from **reactive systems to proactive agents**. Agentic AI models do not simply respond to queries — they decompose complex goals, select and invoke tools, maintain persistent memory, and adapt their strategies based on environmental feedback.

🧠 Dynamic Reasoning

Agents employ multi-step reasoning chains — such as ReAct and chain-of-thought variants — to plan and execute sequences of actions toward a defined objective.

🛠 Tool Integration

Modern agent frameworks natively integrate web search, code execution, APIs, and database access, dramatically expanding what an LLM can accomplish autonomously.

💾 Persistent Memory

Agents maintain episodic and semantic memory across sessions, enabling them to learn from past interactions and build cumulative contextual understanding over time.

🤝 Human Collaboration

The most promising agent architectures position AI not as a replacement but as a collaborative partner — initiating tasks, seeking clarification, and escalating decisions appropriately.

Conclusions & Future Outlook

The convergence of architectural innovation, efficient training, multimodal perception, safety engineering, and agentic reasoning is redefining what large language models can be. The field is moving rapidly toward **adaptive, multimodal, and autonomous AI systems** capable of genuine collaboration with humans across real-world contexts.

MoE → Efficiency at Scale

Sparse activation enables models of unprecedented capability without proportional compute costs — making frontier AI increasingly accessible.

Multimodality → Richer Cognition

Models that see, hear, and reason across modalities mirror human perception more closely and unlock entirely new application domains.

RLVR & PEFT → Responsible Learning

Efficient, verifiably grounded training methods bring LLMs closer to reliable, safe, and domain-adapted deployment at scale.

Agentic & Diffusion AI → New Paradigms

Autonomous agents and non-autoregressive generation represent the next leap — from tools that respond to systems that act, plan, and learn.

> The future of LLMs is not a single model answering a single question — it is an ecosystem of adaptive, collaborative, and accountable AI systems working alongside humanity to solve problems of genuine complexity.