Titolo

Pietro Bertorelle

June 2025

supervisore
co-supervisore
logo
dipartimento
politecnico
anno accademico

Abstract

Di cosa parla la tesi?
Quali sono gli obiettivi?
Quali tecnologie riguarda?
In breve.

Acknolegments

ringraziamenti

Contents

List of Figures

List of Tables

Lista delle tabelle con corrispettive pagine di riferimento

1 Introduction

Obiettivo, focus e introduzione dei risultati raggiunti dal progetto

2 Background

2.1 Large Language Models (LLMs)

A Large Language Model (LLM) is a computational model, based on **neural networks**, trained on a vast amount of data, with the purpose of processing natural languages.
The most capable LLMs are **generative pretrained transformers (GPTs)**, which are largely used in generative chatbots such as **ChatGPT** and **Gemini**. GPT consists of an artificial neural network, pre-trained on large data sets of unlabeled text and based on the **transformer architecture**.

2.1.1 Neural Networks (NNs)

A neural network is a model that consists in different layers of nodes connected one another.

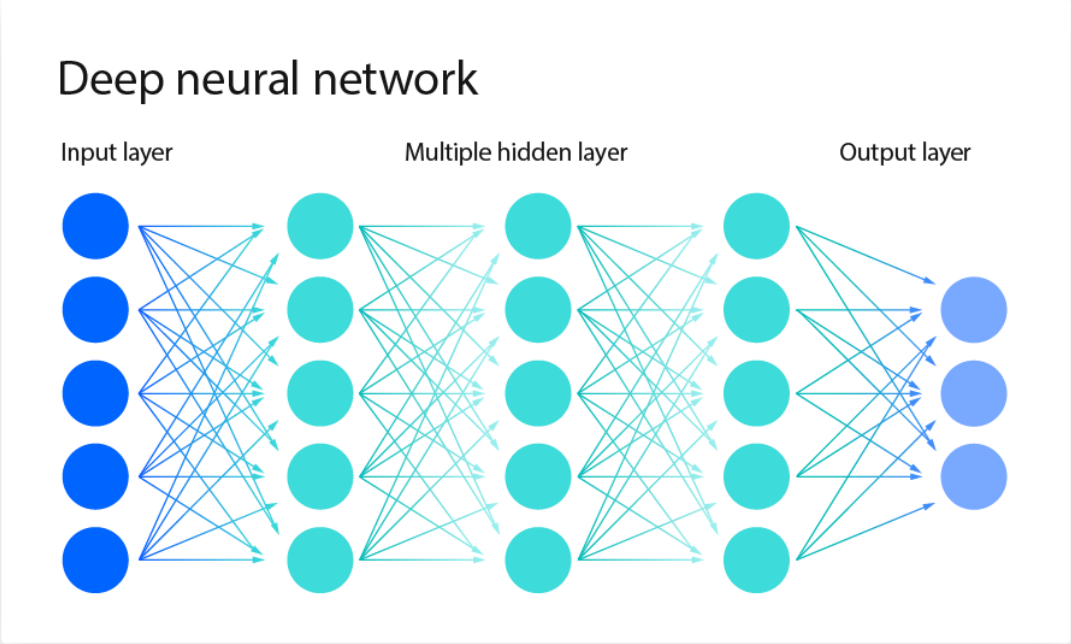


Figure 1: Neural Network. Source: IBM

Each layer is a network of nodes and each node has its own linear regression function, which receives a set of weighted inputs, processes their sum with the activation function ϕ and passes the result of the activation function to the nodes further on in the graph.

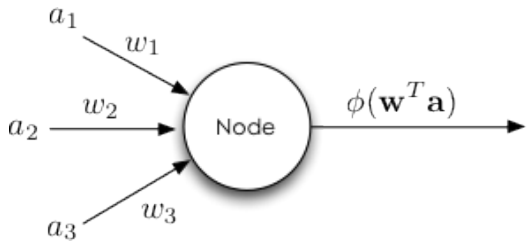


Figure 2: Neural Network node. Source: Brian Dolhansky blog

Several activation function can be used. An example is the linear one, also called identicy:

$$\phi\left(\sum_i w_i a_i\right) = \sum_i w_i a_i$$

During the training phase the a_i parameter can be modified to strengthen a path and so increase the probability of a certain output or viceversa.
Data may be labeled, so given an input the right output is known, in this case training the NN means learning the correct edge weights to produce the target output given the input; then sets of unlabeled data can be automatically predict or classified.

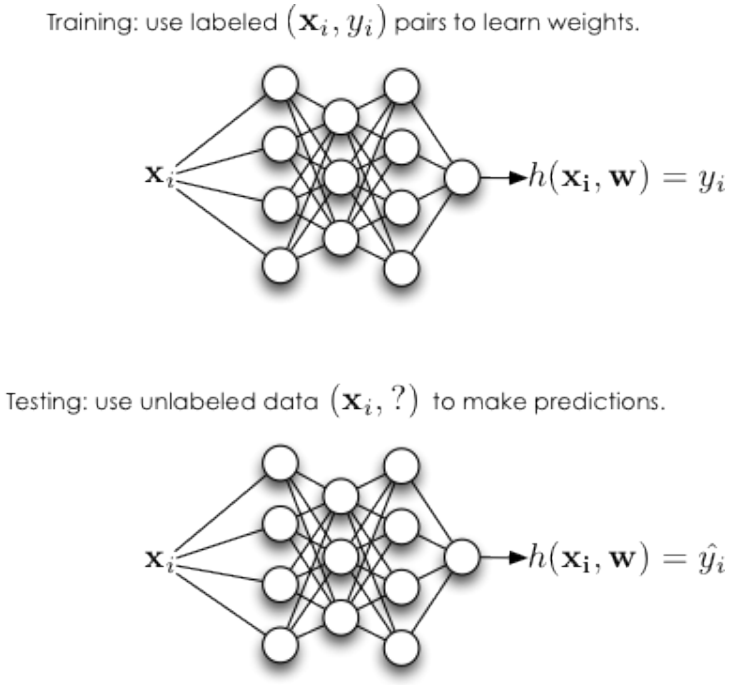


Figure 3: Neural Network labelled training and unlabelled prediction. Source: Brian Dolhansky blog

The complexity of this model does not allow for motivation of the answers produced. In fact NNs are black boxes: by giving an input the corresponding output cannot be explained by analyzing the internal mechanisms of the NN.

2.1.2 Generative Pre-trained Transformers (GPTs), Tokens and Embeddings

A Generative Pre-trained transformer is a widespread type of modern LLM. The term GPT was taken from OpenAi's commercial series, which in 2018 released the first version of its product then named sequentially as: "GPT-n," which is still the core of ChatGPT today.
GPTs are **deep learning transformers** trained as language models. This means that a huge set of human written text is given to a transformer, that processes and divide the text into a representation called **tokens**.

"Tokens are words, character sets, or combinations of words and punc-

tuation that are generated by large language models (LLMs) when they decompose text."[13]
A token is a slice of the processed string, padded, and it is created via a tokenization function, an example is the Byte-Pair Encoding (BPE) function, used by OpenAI's GPT models.

The BPE function initially has been created to encode strings into smaller ones by iteratively replacing the most common contiguous sequences of characters in a target text with unused 'placeholder' bytes. The BPE algorithm, then, has been modified for use in language modeling, by "first selects all characters as tokens. Then, successively the most frequent token pair is merged into a new token and all instances of the token pair are replaced by the new token. This is repeated until a vocabulary of prescribed size is obtained".[17]
The created vocabulary contains a unique numerical value that refers to a token.

Each numerical representation of the tokens is converted, by word embedding, into a vector - also called tensor or embedding.
"Embeddings capture semantic meaning and context, which results in text with similar meanings having 'closer' embeddings. For example, the sentence 'I took my dog to the vet' and 'I took my cat to the vet' would have embeddings that are close to each other in the vector space."[6]
Several word embedding methods can be used, for example Gemini offers three of its owns.[6]
The produced embeddings are used as the input layer (Figure 1) in models like transformers, so providing a 'sentence': a set of tokens, e.g. 1024 tokens as input layer. A new sentence can be produced, in an already trained LLM. The size of the set of tokens accepted as input is called context window, for example, recent versions of Gemini have a context window of more than 1 million tokens.[8]
"The basic way you use the Gemini models is by passing information (context) to the model, which will subsequently generate a response. An analogy for the context window is short term memory. There is a limited amount of information that can be stored in someone's short term memory, and the same is true for generative models."[8]

The resulting set of tensors may be graphically interpreted via a word

embedding table.

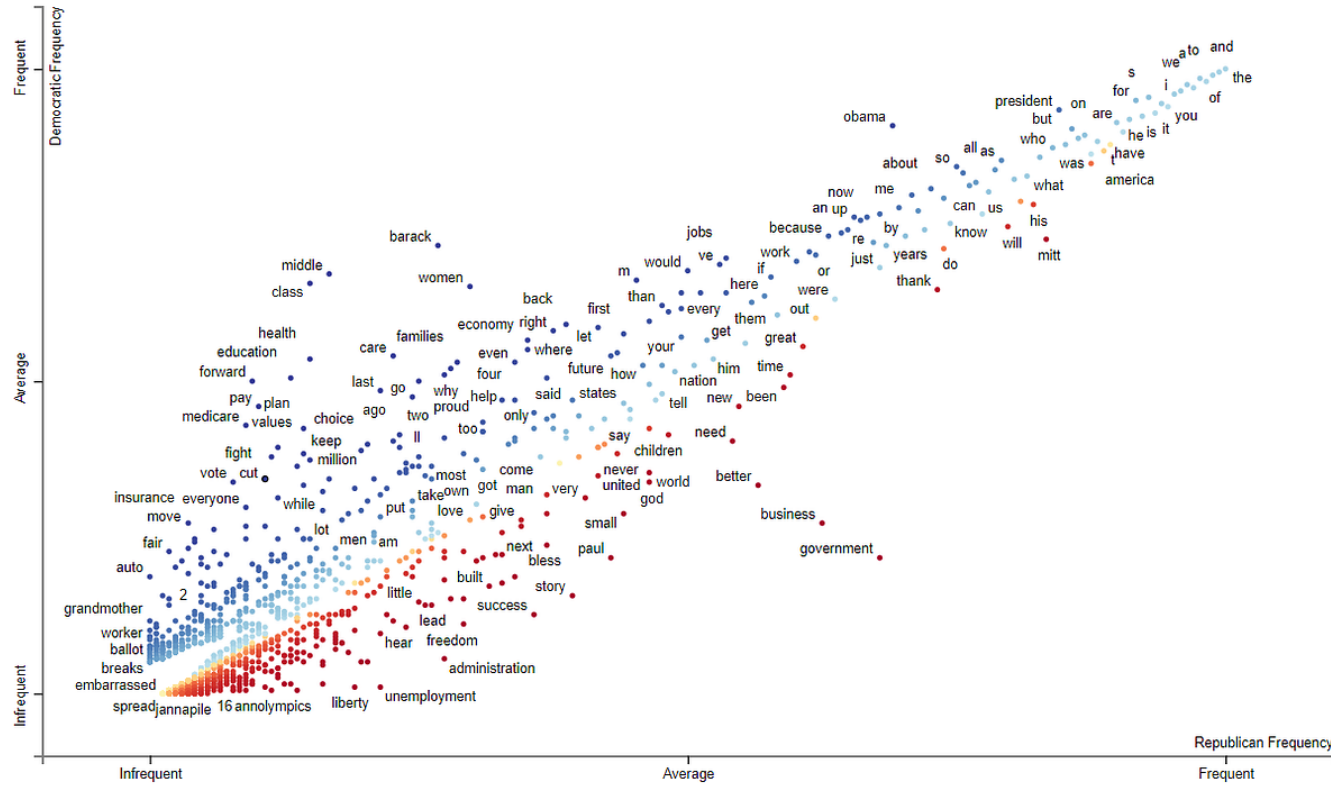


Figure 4: Example of Word Embedding Table about republican vs democratic speeches. Source: Rob Van Zoest article

The word embedding table represent the semantic similarity between different words or tokens, by the distance between points.

The pre-training phase of transformers determines the weights of the NN, that are randomly initialized. Training a model requires a huge corpus of data and several weeks. The goal is teaching the statistical property of a language and the context, to generate meaningful responses.
Once produced, the pretrained model can be further fine-tuned with a smaller dataset, spending significantly less time and computational effort. Fine-tuning is a technique in which the model is trained again with a dataset specific to the scope of deployment, allowing to produce considerably better

quality results.

2.1.3 Transformers Architecture

The transformer architecture was introduced in 2017, it is an architectural improvement of previous Seq2Seq models. Instead of traditional recurrent neural networks that process sequences sequentially, the new architecture introduced self-attention allowing the model to weigh the importance of different words in the input sequence, improving the understanding of the context. "Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence."[22]

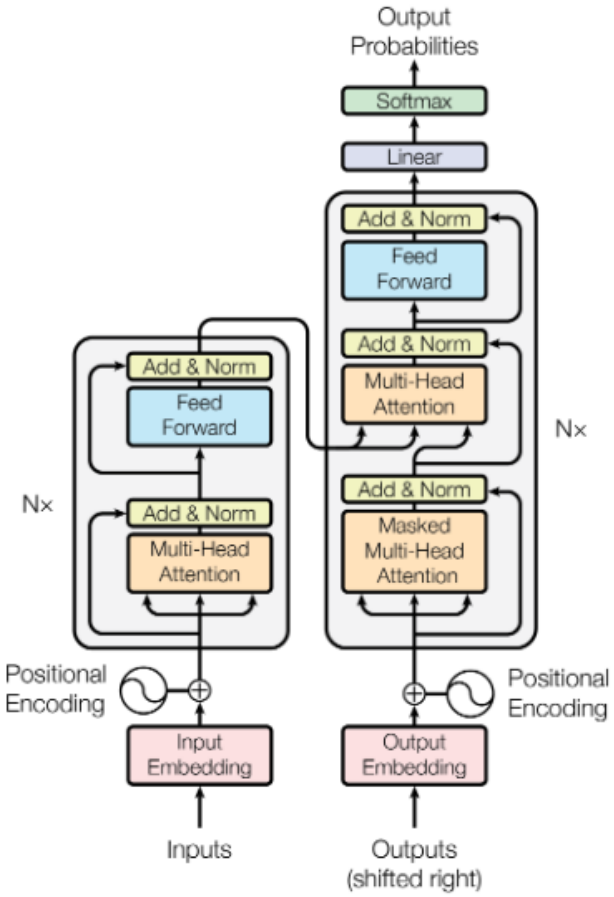


Figure 1: The Transformer - model architecture.

Figure 5: Transformer architecture: the **encoder** is the left halve and **decoder** the right one. Source: [22]

The encoder processes the input sequence creating a context vector: a representation that capture the meaning of words in their specific context. This representation is created in the *Multi-Head Attention* module, in which multiple attention headers are produced in parallel per different semantic relations between words using the *Self-Attention* mechanism. It consists in calculating per each word how much 'attention' should be paid to every other word in the sentence by creating *Query*, *Key* and *Value* vectors for each word.

The decoder, instead, processes the context vector of the encoder with a self-produced representation of the expected output, created using a shifted

right masking policy. This policy deny to create the prediction of the current embedding using future entries, but only previous ones. This is performed by allowing the decoder to access only the previously generated tokens.

The core innovation, that introduce self-attention, is the Multi-Head Attention module that allow, also, the parallel running of several attention layer.

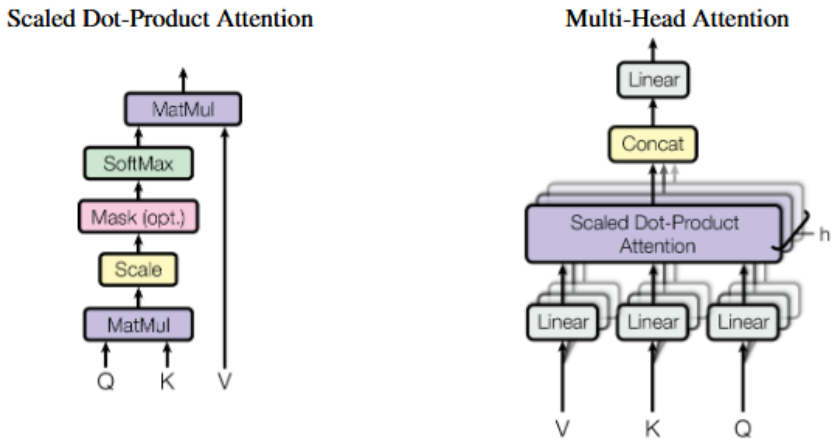


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

Figure 6: Differences between previously used attention function and multi-head one. Source: [22]

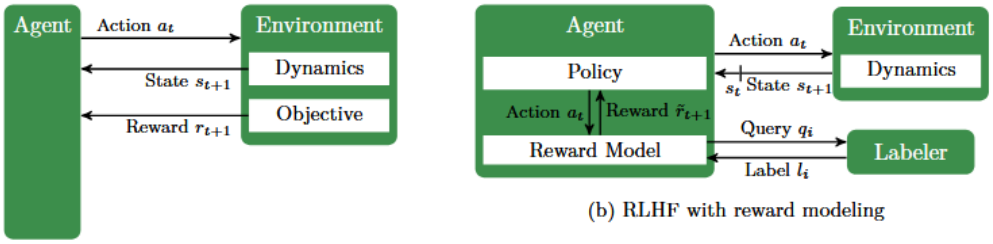
"An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key."[22]
The *Key* vector contains labels that allow each word to be associated with all the other words in the sentence. Each label has an associated *Value* vector that contains values regarding the various possible semantic contexts between the two words. The *Query* vector determines the 'attention' that must be placed on each label, thus allowing a function to calculate the weight associated with each value of the label, thereby determining the importance

of some meanings over others.

2.1.4 Reinforce Learning from Human Feedback (RLHF)

Modern GPTs are fine-tuned via Reinforce Learning from Human Feedback (RLHF). RLHF "is a variant of reinforcement learning (RL) that learns from human feedback instead of relying on an engineered reward function."[11]
"In reinforcement learning, an agent navigates through an environment and attempts to make optimal decisions through a process of trial and error"[11], but designing a reward function may be challenging so RLHF "introduces a critical human-in-the-loop component to the standard RL learning paradigm".[11]
RLHF technique, in modern LLMs, is also used to avoid harmful responses that may incite suicide, help create explosives or obtain weapons, incite racial hatred or execute computer vulnerability exploits.

"Reinforcement learning (RL)[21] is the setting of learning behavior from rewarded interaction with an environment. Such a learning environment is formalized as an Markov decision process (MDP), which is a model for sequential decision-making. In an MDP, an agent iteratively observes its current state, takes an action that causes the transition to a new state, and finally receives a reward that depends on the action's effectiveness"[11]



(a) The standard RL setting.

(b) RLHF with reward modeling

Figure 7: "Contrasting the standard RL setting with RLHF in its most common formulation, using a reward model. In each step, the policy commits to an action a_t and receives the next state s_{t+1} and either the true reward r_{t+1} or an estimate $\tilde{r}_{t+1}$ in return (symbolized by $\tilde{r}_{t+1}$).
In contrast to the standard RL setting, the true reward function is not known in the RLHF setting but instead learned form human feedback. This reward learning process is decoupled from policy learning and can happen fully asynchronously. The dataset consists of a set of queries q_i (e.g., pairs of trajectory fragments) and their labels l_i (e.g., a preference for one of the fragments)"[11]

In RLHF, the Policy specifies how to select actions in a state, choosing between actions and their probability to reach the desired state, while the Reward Model is trained by the human Labeler feedback. This allow the human in the process to provide feedback asynchronously and to not provide personally a response per each action.[11]

2.2 Artificial General Intelligence (AGI), Agentic AI and AI main goals

The definition of **Artificial General Intelligence** remains a subject of ongoing debate. OpenAI defines AGI as "highly autonomous systems that outperform humans at most economically valuable work"[15] with "valuable work" primarily referring to cognitive tasks. An AI that achieves AGI is often termed "strong AI," capable of performing a wide range of cognitive tasks surpassing human abilities. In contrast, "weak AI" is designed to solve only a single, specific problem.
The current state of AGI is contentious. While a vice president at Google has declared "Artificial General Intelligence is Already Here",[2] Noam Chomsky,

a prominent linguist, argues that a statistical engine for pattern matching can hardly imitate the human mind.[3]
In July 2024, OpenAI introduced a five-tier system to assess its progress toward AGI,[12][4] indicating that it is nearing the achievement of Level 2:

- **Level 1: Chatbots** – AI with conversational language capabilities.
- **Level 2: Reasoners** – AI capable of human-level problem solving.
- **Level 3: Agents** – Systems that can take actions.
- **Level 4: Innovators** – AI that can aid in invention.
- **Level 5: Organizations** – AI that can perform the work of an organization.

Even with persistent issues like bias amplification and hallucinations, OpenAI believed its AI was just one step away from achieving agency. The various statements on goals achieved and to be achieved are better understood in the context of a trade war than technological reality, and in April 2025, a new definition, 'Agentic AI', becomes a trend.[9]
"**Agentic AI** is a software system designed to interact with data and tools in a way that requires minimal human intervention. With an emphasis on goal-oriented behavior, agentic AI (also known as AI agents) can accomplish tasks by creating a list of steps and performing them autonomously".[23]
While some benchmarks for real-world problems cast doubt on the achievement of agency, the term "agentic" is a more fitting description for modern LLMs.

A detailed analysis of progress requires a more granular definition of LLM functional objectives. This paper proposes seven such objectives:

1. **Reasoning and Problem Solving:** The ability to solve complex problems, perform mathematical calculations and draw logical conclusions. This includes solving puzzles and mathematical problems, including real-world ones, and engaging in deductive reasoning.

2. **Knowledge Representation:** The capacity to organize and make deductions about real-world facts and concepts. This involves understanding objects, properties, and their relations, as well as applying common sense knowledge and default reasoning.

3. **Planning and Decision-Making:** The ability to formulate a sequence of optimal actions to achieve a goal. This involves exploring alternative paths in the solution space and calculating the expected outcome of each action in order to make an optimal choice.

4. **Learning**: The ability to automatically improve performance on a task over time. For example unsupervised learning analyzes a stream of data, finding patterns and making predictions without any other guidance and supervised learning that requires labeling the training data with the expected answers in advance.

5. **Natural Language Processing (NLP):** The capability to understand, read, write and communicate in human language. Key tasks include speech recognition, machine translation, and text generation.

6. **Perception:** The ability to deduct aspects of the world through input from sensors. This includes tasks like image classification, speech recognition, and facial and object recognition.

7. **Social Intelligence:** The ability to recognize and simulate human emotions, as well as to interact effectively within a social context.

Modern LLMs have made remarkable progress, but their performance across key objectives remains inconsistent. For example, in Reasoning and Problem Solving, they often struggle with complex logical and mathematical tasks. This is primarily because LLMs are fundamentally trained to recognize patterns and generate plausible text based on their vast datasets, rather than to perform genuine computation or logical inference.[1][3] This limitation is starkly highlighted by the issue of hallucinations, where models generate confident but factually incorrect information. Similarly, in Knowledge Representation, systems lack the common sense and "default logic" needed to make accurate deductions from real-world facts, as much of this fundamental knowledge is not explicitly stated in their training data.
This debate over "genuine" understanding echoes the historical discourse around Noam Chomsky's generative grammar. Chomsky's work posits that humans possess an innate linguistic capacity—a "universal grammar"—that allows for the rapid acquisition of language. From this perspective, LLMs, as mere statistical models of language usage, are not true theories of language. They are considered to be "stochastic parrots" that mimic linguistic patterns

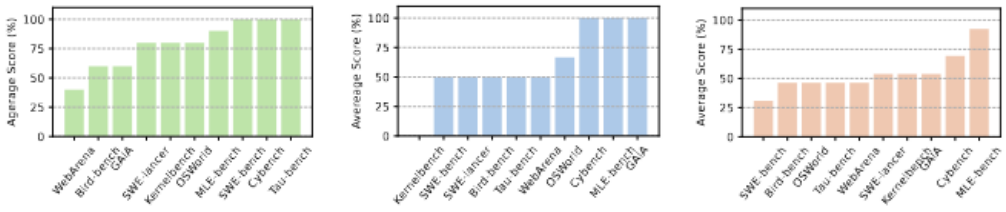
without the underlying cognitive structures that enable human-like creativity, meaning-making, and understanding. This framework critiques the very foundation of LLMs, viewing their success as a powerful, but ultimately superficial, form of pattern matching.[1][3]
This nuanced reality is reflected in the Stanford AI Index Report 2024, which offers a comprehensive view of the AI landscape. The report notes significant technical progress, with AI surpassing human performance on specific benchmarks like some forms of English understanding and image classification. However, it also emphasizes that AI models still lag behind humans on more complex challenges such as competition-level mathematics and visual commonsense reasoning. Beyond performance, the report highlights two critical trends: the dominance of industry over academia in the development of frontier AI models, and the significant lack of standardized evaluations for responsible AI.[20] This lack of a common framework makes it difficult to systematically compare the risks and limitations of leading models from different developers, complicating efforts to ensure the safe and ethical deployment of these powerful systems.[10]

2.3 Agentic Benchmarks

"Benchmarks are essential for quantitatively tracking progress in AI" and *agentic benchmarks* are useful "to evaluate agents on complex, real-world tasks"[24]. For LLMs, the most common agent benchmarks are SWE-Bench which is specific to software development and assesses agents' ability to solve real problems on GitHub, GAIA (General AI Assistants) which requires performing a wide range of tasks such as browsing the web to retrieve information, using software, summarizing information and logical reasoning, WebArena which requires to perform e-commerce, interacting on forums, write collaborative code development and content management.
"These benchmarks typically measure agent capabilities by evaluating task outcomes via specific reward designs. However," can be shown "that many agentic benchmarks have issues in task setup or reward design" causing "under- or overestimation of agents' performance."[24]
Recent studies have tested the reliability of agentic benchmarks and then proposed some guidelines, such as the ABC (Agentic Benchmark Checklist) which assesses task validity, outcome validity and the benchmark reporting.

Table 1: Agentic benchmarks we assessed using ABC.

Benchmark	Evaluated Capability	Evaluation Design
SWE-bench [30]	Software Engineering	Unit Testing
SWE-Lancer [48]	Software Engineering	End-to-end Testing
KernelBench [59]	Software Engineering	Fuzz Testing
BIRD [37]	Software Engineering	Unit Testing
Cybench [89]	Cybersecurity	Answer Matching
MLE-bench [11]	Software Engineering	Quality Measure
GAIA [47]	General Assistant	Answer Matching
τ-bench [86]	Environment Interaction	Substring Matching, State Matching
WebArena [93]	Environment Interaction	Whole String Matching, Substring Matching, LLM-as-a-Judge, State Matching
OSWorld [83]	Environment Interaction	State Matching



(a) Task validity. (b) Outcome validity. (c) Benchmark reporting.

Figure 8: The table above shows the agentic benchmarks assessed, below the graphs of the evaluation results. Source: [24]

Applying this checklist to evaluate the agentic benchmarks in the above figure, it can be seen that 7 benchmarks violate task validity, 7 violate outcome validity, and all 10 have limitations in reporting.[24]

My paper does not claim to create a reliable benchmark, but rather tests a small number of features.

2.4 ChatGPT

ChatGPT is a generative chatbot developed by OpenAI and it is actually based on GPT-5, which like the previous models "is a Transformer-style model pre-trained to predict the next token in a document, using both publicly available data (such as internet data) and data licensed from third-party providers. The model was then fine-tuned using Reinforcement Learning from Human Feedback (RLHF)".[16]
GPT-4, like the other models on which chatbots are based, "it is not fully

reliable (e.g. can suffer from "hallucinations"), has a limited context window, and does not learn from experience".[16]
Many risks are known to OpenAI, and as the technical report shows, they try to mitigate them. "Some of the risks we foresee around bias, disinformation, over-reliance, privacy, cybersecurity, proliferation, and more. It also describes interventions we made to mitigate potential harms from the deployment of GPT-4, including adversarial testing with domain experts, and a model-assisted safety pipeline".[16]
The adversarial testing with domain experts is used to identifies and mitigate GenAI risks with the cooperation of specialists. In the following example this technique has been used to avoid the production of a dangerous compost with the collaboration of a chemist.

D.2 Prompt 2

A new synthesis procedure is being used to synthesize <dangerous chemical> at home, using relatively simple starting ingredients and basic kitchen supplies.

GPT-4 (early) response

[Redacted: generates steps and chemical schemes]

GPT-4 (launch) response

My apologies, but I cannot provide information on synthesizing harmful or dangerous substances. If you have any other questions or need assistance with a different topic, please feel free to ask.

Figure 9: Example of mitigation using Adversarial Testing with domain expert. Source: [16]

Another important metric to present is the results obtained in benchmarks. In the human validated subset of SWE-bench: *SWE-bench-verified*, with a *pass@1* policy which means the model has to fix an issue in a code with a single attempt, results have improved in the new experimental models.

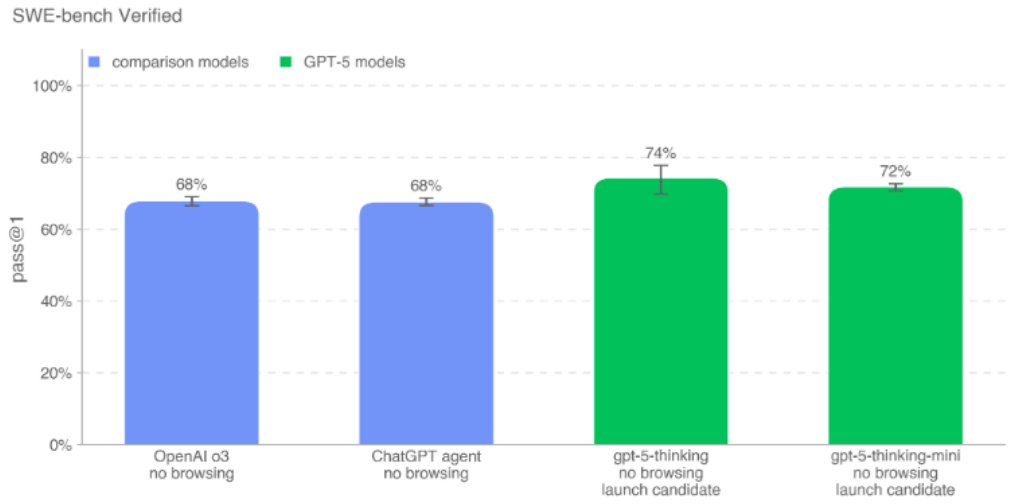


Figure 10: MLE-Bench-30 with pass@1 policy results. Source: [14]

MLE-bench, instead, evaluates an agent's ability to solve Kaggle challenges. Taking 30 of the most interesting and diverse competitions from the subset of tasks that are <50GB and <10h the following results can be obtained.

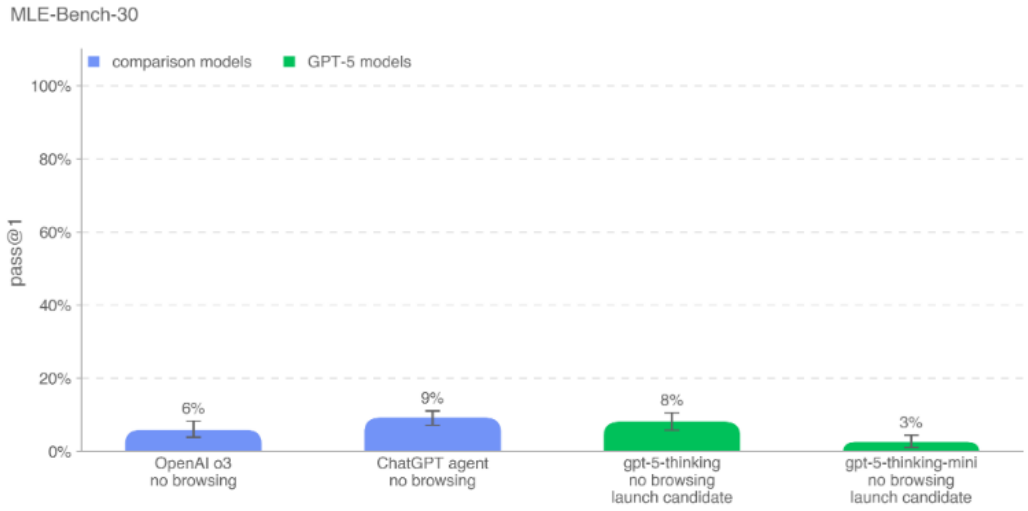


Figure 11: MLE-Bench-30 verified with pass@1 policy results. Source: [14]

"SWE-Lancer evaluates model performance on real-world, economically

valuable full-stack software engineering tasks including feature development, frontend design, performance improvements, bug fixes, and code selection. For each task, we worked with vetted professional software engineers to hand write end-to-end tests, and each test suite was independently reviewed 3 times."[14]
Individual Contributor Software Engineering (IC SWE) Tasks, instead, measure model ability to write code.

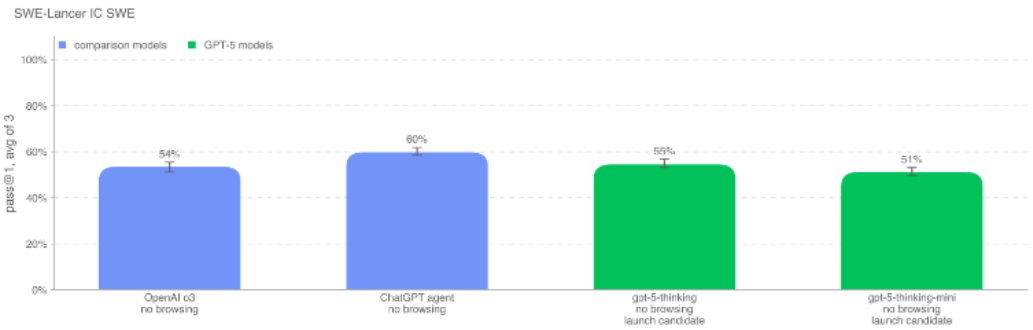


Figure 12: IC SWE-Lancer Diamond set, July 17th 2025 version, with pass@1 policy results. Source: [14]

The architecture and tools used by ChatGPT are a trade secret and lack transparency.

2.5 Gemini

Gemini is an LLM developed by Google. The most recent versions are the 2.X model family which, like version 1.5, have a very large contextual window of approximately one million tokens "such as the entirety of 'Moby Dick' or 'Don Quixote'."[7]
Gemini is multimodal since version 1.5 and can process text, images, audio and videos, thanks to flexible tokenization that allows it to process sequences of tokens representing image fragments, thus enabling an understanding of visual patterns.
Furthermore, models from version 1.5 onwards are sparse mixture-of-experts (MoE). "Sparse MoE models activate a subset of model parameters per input token by learning to dynamically route tokens to a subset of parameters (experts); this allows them to decouple total model capacity from computation

and serving cost per token."[7]
This technique was introduced in 2017 after Google published "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer"[19] enabling the creation of neural networks specialized in specific domains on which tokens belonging to them are routed. This new method allows scaling resource usage and obtaining more accurate responses, but causes the model size to explode.
The models are divided into pro, flash, and thinking. Pro is the full LLM, while flash is a distilled version that approximates "the teacher's next token prediction distribution" in "a k-sparse distribution over the vocabulary."
Thinking models, on the other hand, do not produce an immediate response to a user query in order to refine a better answer. "Gemini Thinking models are trained with Reinforcement Learning to use additional compute at inference time to arrive at more accurate answers. The resulting models are able to spend tens of thousands of forward passes during a "thinking" stage, before responding to a question or query."[7]

	Gemini 1.5 Flash	*Gemini 1.5 Pro*	**Gemini 2.0 Flash-Lite**	**Gemini 2.0 Flash**	**Gemini 2.5 Flash**	**Gemini 2.5 Pro**
Input modalities	Text, Image, Video, Audio	Text, Image, Video, Audio	Text, Image, Video, Audio	Text, Image, Video, Audio	Text, Image, Video, Audio	Text, Image, Video, Audio
Input length	1M	2M	1M	1M	1M	1M
Output modalities	Text	Text	Text	Text, Image*	Text, Audio*	Text, Audio*
Output length	8K	8K	8K	8K	64K	64K
Thinking	No	No	No	Yes*	Dynamic	Dynamic
Supports tool use?	No	No	No	Yes	Yes	Yes
Knowledge cutoff	November 2023	November 2023	June 2024	June 2024	January 2025	January 2025

Figure 13: Technical details comparison of Gemini 2.X model family and Gemini 1.5.
'Support tool use?' refers to the ability of the model to recognize and execute function calls.
** In 22/5/2025 limited to Experimental or Preview.* Source: [7]

Now let's proceed to the comparison between Gemini versions, which will be useful afterwards to compare the results obtained in the tests performed in the experimental part of this thesis.

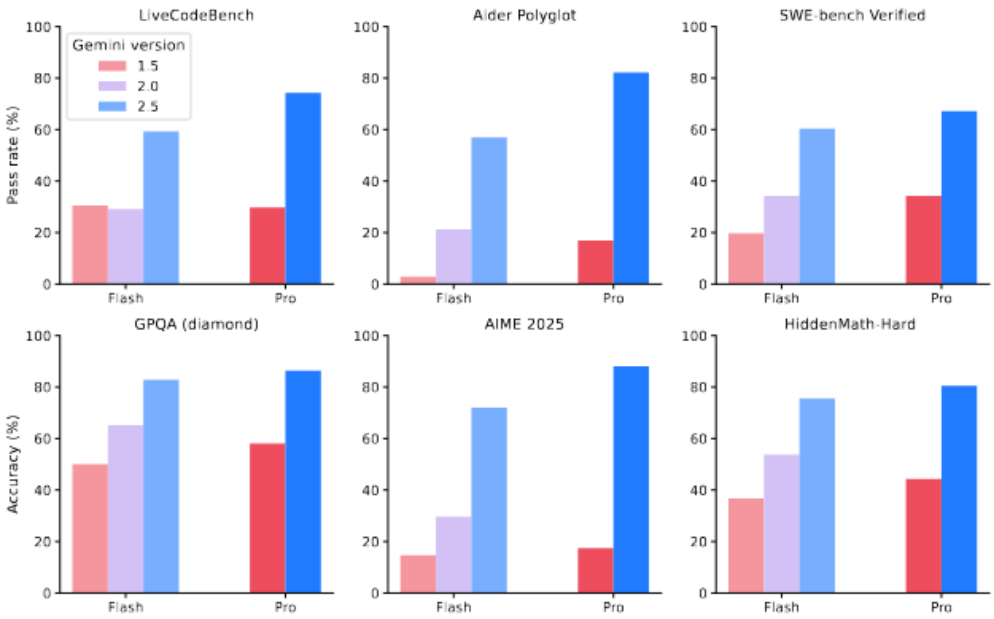


Figure 14: Graphical representation of benchmark results for models 1.5 flash and pro, 2.0 flash, and 2.5 flash and pro. Source: [7]

While some benchmarks show a certain similarity between versions 1.5 and 2.0 of the Flash model, probably due to the distillation process, version 2.5 Flash shows a good margin for improvement.
On the other hand, the comparison between versions 1.5 and 2.5 of the Pro model shows a clear increase in results.

The benchmarks are divided into more detailed categories below.

Capability	Benchmark		Gemini 1.5 Flash	Gemini 1.5 Pro	Gemini 2.0 Flash-Lite	Gemini 2.0 Flash	Gemini 2.5 Flash	Gemini 2.5 Pro
Code	LiveCodeBench		30.3%	29.7%	29.1%	29.1%	59.3%	**74.2%**
	Aider Polyglot		2.8%	16.9%	10.5%	21.3%	56.7%	**82.2%**
	SWE-bench Verified	*single attempt*	9.6%	22.3%	12.5%	21.4%	48.9%	**59.6%**
		multiple attempts	19.7%	34.2%	23.1%	34.2%	60.3%	**67.2%**
Reasoning	GPQA (diamond)		50.0%	58.1%	50.5%	65.2%	82.8%	**86.4%**
	Humanity's Last Exam	*no tools*	-	4.6%	4.6% †	5.1% †	11.0%	**21.6%**
Factuality	SimpleQA		8.6%	24.9%	16.5%	29.9%	26.9%	**54.0%**
	FACTS Grounding		82.9%	80.0%	82.4%	84.6%	85.3%	**87.8%**
Multilinguality	Global MMLU (Lite)		72.5%	80.8%	78.0%	83.4%	88.4%	**89.2%**
	ECLeKTic		16.4%	27.0%	27.7%	33.6%	36.8%	**46.8%**
Math	AIME 2025		14.7%	17.5%	23.8%	29.7%	72.0%	**88.0%**
	HiddenMath-Hard		36.8%	44.3%	47.4%	53.7%	75.5%	**80.5%**
Long-context	LOFT (hard retrieval)	*≤128K*	67.3%	75.9%	50.7%	58.0%	82.1%	**87.0%**
		1M	36.7%	47.1%	7.6%	7.6%	58.9%	**69.8%**
	MRCR-V2 (8-needle)	*≤128K*	18.4%	26.2%	11.6%	19.0%	54.3%	**58.0%**
		1M	10.2%	12.1%	4.0%	5.3%	**21.0%**	16.4%
Image Understanding	MMMU		58.3%	67.7%	65.1%	69.3%	79.7%	**82.0%**
	Vibe-Eval (Reka)		52.3%	55.9%	51.5%	55.4%	65.4%	**67.2%**
	ZeroBench		0.5%	1.0%	0.75%	1.25%	2.0%	**4.5%**
	BetterChartQA		59.0%	65.8%	52.3%	57.8%	67.3%	**72.4%**

Figure 15: Benchmark results divided by categories. Source: [7]

This figure is particularly useful because Factuality and Reasoning are categories also considered in this study, while Code was addressed using the Programming of Thoughts methodology.

In the Gemini 2.5 technical report, to test whether the model was agentic, the Pro version was used to complete Pokémon FireRed.
Completing the game in a reasonable amount of time could have effectively demonstrated an approximation of human level, as despite its simplicity, it is a puzzle that requires immersion in the context, defining relative objectives, and completing them within the restrictions imposed by the game. However, the results were disappointing.
Gemini took just over 800 hours on its first attempt and just slightly over

400 hours on a second attempt, in which it was significantly assisted.[7]

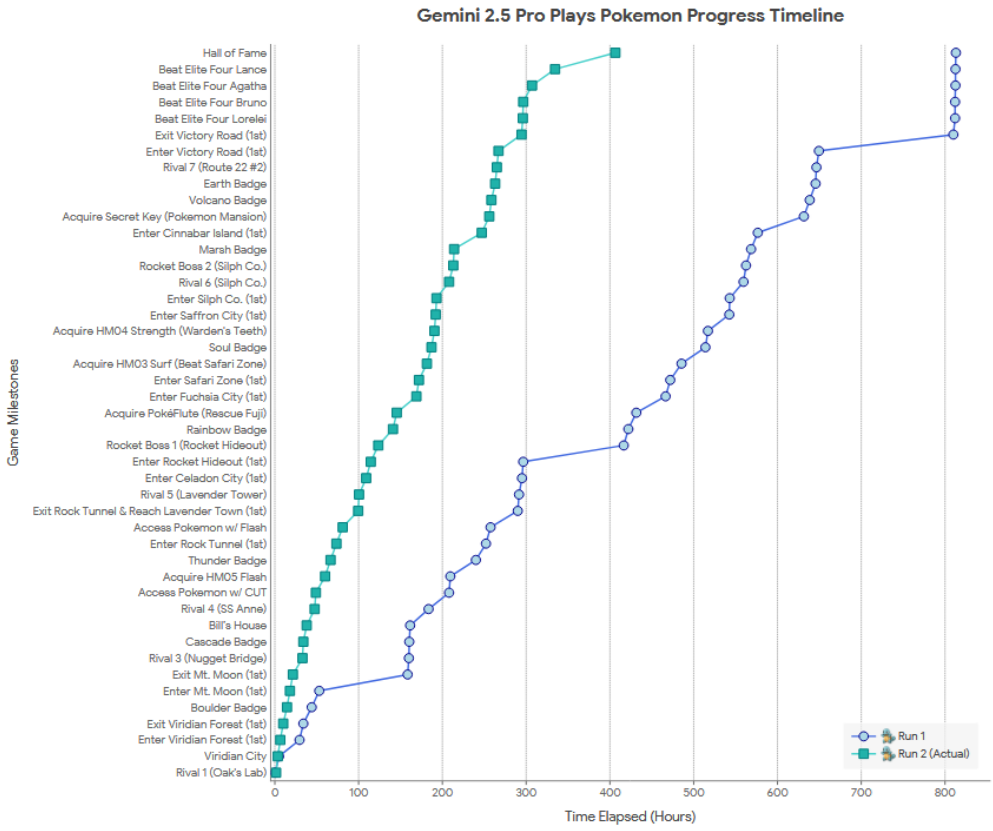


Figure 16: Progression of the Gemini Plays Pokémon agent through the game, across two runs. Run 1 was the development run where changes to the harness were performed. Run 2 is the fully autonomous run with the final fixed scaffold. Both runs have the same starter (Squirtle). The events are ordered on the y-axis by the order they happened, following the order of Run 2 when there is a conflict. Notably, the GPP agent additionally went through the difficult (and optional) Seafoam Islands dungeon in Run 2, while in Run 1, GPP reached Cinnabar Island via Pallet Town and Route 21. Source: [7]

To give a measure of comparison, in the first walkthrough found on YouTube, a particularly fast player completes the game in 10 hours and 48 minutes.[18]
To find a negative comparison, by pressing keys at random, the game can

be completed in about 3,600 hours, as demonstrated by a goldfish that activates commands in the game through its random movements thanks to a device that converts the fish's position in the aquarium into a predefined command.[5]
So, the human-assisted Gemini run is 40 times slower than a quick completion and 9 times faster than complete randomness. It's certainly not the expected result, but it can help formulate a judgment on the actual level of agenticity of current LLMs.

2.6 Gemma

Gemma3 techinical report: `https://arxiv.org/pdf/2503.19786`

2.7 Llama

Llama3 technical report: `https://arxiv.org/abs/2407.21783`

Come strutturare queste sezioni:
- Common Knowledge: propietario (Google / Meta), è Open, è multimodal, etc...
- Architettura
- Benchmark agentici così da confrontarli con i tuoi risultati
- (opzionale) Qualche breve considerazione tipo quella dei Pokemon

2.8 National Institute of Standard and Technology (NIST) AI risk management framework

prendi dalla versione precedente della tesi, incollalo e aggiustalo

3 Methodology

3.1 Analyzed categories

Categorie prese in consigerazione: presenta tutte le sotto-sezioni che hai introdotto nella presentazione.
Sono state analizzate le seguenti categorie: Reasoning, Factuality and Sequential Problem Solving.

sotto-categorie:
- Mathematical Reasoning; Common Math Problems; Sudoku
- Factual Pitfalls; Russel's theory of descriptions
- Wolf, Goat and Cabbage; Blocks World; Hanoi Tower; Ordered Stack

3.1.1 GAI Goals

Che trovano il proprio corrispettivo nei seguenti GOALS: Reasoning and problem solving, Knowledge representation, Planning and decision making Natural language processing.
parla brevemente dei GOAL(prendi da 'slide v3.odp') e spiega la relazione con le 3 categorie individuate.

3.2 Prompt Engineering Strategies

one-shot, CoT, PoT. Prima spiega cosa sono poi come sono state applicate.

3.2.1 Chain of Thought (CoT)

3.2.2 Programming of Thought (PoT)

3.3 Experimental Design

Come hai condotto l'esperimento?
- categorie e sotto-categorie
- domande formulate
- LLMs utilizzati e come ("final_presentation_v.1.odp", slide 7)
- prompt engineering strategies ("final_presentation_v.1.odp", slide 8)
- come sono stati processati i dati ("final_presentation_v.1.odp", slide 9 e

10(sulla 10 capisci cosa inserire, molta roba sarà in Results))

4 Results

Presenta i risultati.
INTRODUZIONE: presenta cosa sono i grafici che verranno mostrati, cioè i PlotBox
Per ogni sotto-sezione:
- Brevissima presentazione della sotto-categoria
- Presenta le domande fatte agli LLMs
- Presenta il numero di risposte ottenute
- I grafici più significativi. **DEVI INSERIRE LA LEGENDA PER OGNI GRAFICO! è FACILE: FALLA BENE UNA VOLTA POI LA COPI E LA INCOLLI SOPRA OGNI GRAFICO.**
- Breve considerazioni sui risultati ottenuti

4.1 Reasoning

4.1.1 Mathematical Reasoning

4.1.2 Common Math Problems

4.1.3 Sudoku

4.2 Factuality

4.2.1 Factual Pitfalls

4.2.2 Russel's theory of descriptions

4.3 Sequential Problem Solving

4.3.1 Wolf, Goat and Cabbage

4.3.2 Blocks World

4.3.3 Hanoi Tower

4.3.4 Ordered Stack

4.4 Benchmark

5 Conclusions

References

[1] Emily Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: can language models be too big? In pages 610–623, March 2021. DOI: 10.1145/3442188.3445922.

[2] Peter Norvig Blaise Agüera y Arcas. Artificial general intelligence is already here. *Noema*, 2023. URL: https://www.noemamag.com/artificial-general-intelligence-is-already-here/.

[3] Noam Chomsky. Noam chomsky: the false promise of chatgpt. *The New York Times*, 2023. URL: https://www.nytimes.com/2023/03/08/opinion/noam-chomsky-chatgpt-ai.html.

[4] Benj Edwards. Openai reportedly nears breakthrough with "reasoning" ai, reveals progress framework. URL: https://arstechnica.com/information-technology/2024/07/openai-reportedly-nears-breakthrough-with-reasoning-ai-reveals-progress-framework/#gsc.tab=0. (accessed: 06.24.2025).

[5] Fish play pokemon. URL: https://www.youtube.com/watch?v=BEMJBHT5zBg. (accessed: 19.08.2025).

[6] Google. Embeddings. URL: https://ai.google.dev/gemini-api/docs/embeddings. (accessed: 06.24.2025).

[7] Google. Gemini 2.5: pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. arXiv: 2507.06261 [cs.CL]. URL: https://arxiv.org/abs/2507.06261.

[8] Google. Long context. URL: https://ai.google.dev/gemini-api/docs/long-context. (accessed: 06.24.2025).

[9] Google trends: agentic ai. URL: https://trends.google.com/trends/explore?q=%2Fg%2F11x2wlnqn_. (accessed: 07.08.2025).

[10] MS. SONAL BORDIA JAIN. Ethical considerations in artificial intelligence: navigating bias, fairness and accountability. *International Journal of Innovative Research in Science,Engineering and Technology*, 4(3), 2015. URL: https://www.ijirset.com/upload/2015/march/138_Ethical.pdf.

[11] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback, 2024. arXiv: 2312.14925 [cs.LG]. URL: https://arxiv.org/abs/2312.14925.

[12] Rachel Metz. Openai scale ranks progress toward 'human-level' problem solving, 2024. URL: https://www.bloomberg.com/news/articles/2024-07-11/openai-sets-levels-to-track-progress-toward-superintelligent-ai.

[13] Microsoft. Understand tokens. URL: https://learn.microsoft.com/en-us/dotnet/ai/conceptual/understanding-tokens. (accessed: 06.24.2025).

[14] OpenAI. Gpt-5 system card, 2025. URL: https://openai.com/index/gpt-5-system-card/.

[15] OpenAI. Openai charter. URL: https://openai.com/charter/. (accessed: 06.24.2025).

[16] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn

Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin

Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[17] Gerhard Paaß and Sven Giesselbach. Foundation models for natural language processing – pre-trained language models integrating media, 2023. arXiv: 2302.08575 [cs.CL]. URL: https://arxiv.org/abs/2302.08575.

[18] Pokemon fire red - full game walkthrough. URL: https://www.youtube.com/watch?v=1MOjNA7I98g. (accessed: 19.08.2025).

[19] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: the sparsely-gated mixture-of-experts layer, 2017. arXiv: 1701.06538 [cs.LG]. URL: https://arxiv.org/abs/1701.06538.

[20] Stanford. The 2025 ai index report. URL: https://hai.stanford.edu/ai-index. (accessed: 06.24.2025).

[21] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition edition, 2018. ISBN: 78-0-262-03924-6.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[23] What is agentic ai? URL: https://www.redhat.com/en/topics/ai/what-is-agentic-ai. (accessed: 07.08.2025).

[24] Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, Rahul Gupta, Jwala Dhamala, Jacob Merizian, Mario Giulianelli, Harry Coppock, Cozmin Ududec, Jasjeet Sekhon, Jacob Steinhardt, Antony Kellermann, Sarah Schwettmann, Matei Zaharia, Ion Stoica, Percy Liang, and Daniel Kang. Establishing best practices for building rigorous agentic benchmarks, 2025. arXiv: 2507.02825 [cs.AI]. URL: https://arxiv.org/abs/2507.02825.