ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal
Regular Issue, Vol. 12 N. 1 (2023), e31704
eISSN: 2255-2863
DOI: https://doi.org/10.14201/adcaij.31704

Generative Artificial Intelligence: Fundamentals

Juan M. Corchado[a], Sebastian López F[a], Juan M. Núñez V[a], Raul Garcia S.[a], and Pablo Chamoso[a]

[a]BISITE Research Group, University of Salamanca, Edificio Multiusos I+D+I, Salamanca, 37007
corchado@usal.es, sebastianlopezflorez@usal.es, jmnunez@usal.es, raulrufy@usal.es, chamoso@usal.es

KEYWORDS

large language models; artificial intelligence; transformers; GPT

ABSTRACT

Generative language models have witnessed substantial traction, notably with the introduction of refined models aimed at more coherent user-AI interactions—principally conversational models. The epitome of this public attention has arguably been the refinement of the GPT-3 model into ChatGPT and its subsequent integration with auxiliary capabilities such as search features in Microsoft Bing. Despite voluminous prior research devoted to its developmental trajectory, the model's performance, and applicability to a myriad of quotidian tasks remained nebulous and task specific. In terms of technological implementation, the advent of models such as LLMv2 and ChatGPT-4 has elevated the discourse beyond mere textual coherence to nuanced contextual understanding and real-world task completion. Concurrently, emerging architectures that focus on interpreting latent spaces have offered more granular control over text generation, thereby amplifying the model's applicability across various verticals. Within the purview of cyber defense, especially in the Swiss operational ecosystem, these models pose both unprecedented opportunities and challenges. Their capabilities in data analytics, intrusion detection, and even misinformation combatting is laudable; yet the ethical and security implications concerning data privacy, surveillance, and potential misuse warrant judicious scrutiny.

1. Introduction

Artificial intelligence has become one of the driving forces of 21st century economics, especially because of the development of generative artificial intelligence. This article explains what generative artificial intelligence is and what its foundations are, paying special attention to large language models. A short historical review analyses how AI has evolved to date and then presents the most relevant models and techniques. The limitations and challenges of the techniques that have been used thus far are analyzed and it is explained how generative artificial intelligence offers new and better alternatives for problem solving. The ethical and social aspects linked to the use of these technologies are also of great relevance and are explored throughout this paper. Finally, future trends are presented as well as the authors' vision of the potential of this technology to change the world.

2. Definition and Fundamentals

This section describes generative artificial intelligence (AI), emphasizing its most significant and differentiating elements. This is followed by a brief review of the evolution of artificial intelligence and how this has led to the emergence of generative AI as we know it today. Finally, a summary of the progress of AI up to where we are today is presented.

2.1. What Is Generative AI?

Generative AI is defined as a branch of artificial intelligence capable of generating novel content, as opposed to simply analyzing or acting on existing data, as expert systems do (Vaswani et al., 2017). This is a real evolution over the intelligent systems used to date that are, for instance, based on neural networks, case-based reasoning systems, genetic algorithms, fuzzy logic (Nguyen et al., 2013) or hybrid AI models (Gala et al., 2016; Abraham et al., 2009; Corchado & Aiken, 2002; Corchado et al., 2021) and models and algorithms that used specific data for specific problems and generated a specific answer on the basis of the input data.

Generative artificial intelligence incorporates discriminative or transformative models trained on a corpus or dataset, capable of mapping input information into a high-dimensional latent space. In addition, it has a generative model that drives stochastic behavior, creating novel content at every attempt, even with the same input stimuli. These models can perform unsupervised, semi-supervised or supervised learning, depending on the specific methodology. Although this paper aims to present the full potential of generative AI, the focus is on large language models (LLMs) to generalize from there (Chang et al., 2023). LLMs are a subcategory of generative artificial intelligence (AI). Generative AI refers to models and techniques that have the ability to generate new and original content, and within this domain, LLMs specialize in generating text. An LLM such as OpenAI's GPT (Generative Pre-trained Transformer) is basically trained to generate text, or rather to answer questions with paragraphs of text (Guan et al., 2020). Once trained, it can generate complete sentences and paragraphs that are coherent and, in many cases, indistinguishable from those written by humans, simply from an initial stimulus or prompt (Madotto et al., 2021).

While generative AI also encompasses models that can generate other types of content such as images (e.g., DALL-E, also from OpenAI) or music, LLMs focus specifically on the domain of language (Adams, et al., 2023). LLMs can therefore be considered as a part or subset of the broad category of generative AI.

LLMs are neural networks designed to process sequential data (Bubeck et al., 2023). An LLM can be trained on a corpus of text (digitized books, databases, information from the internet, etc.); the input text can be used to learn to generate text, word-by-word in a sequence, given previous information. Transformers are perhaps the most widely used models in the construction of these LLMs (Von Oswald et al., 2023). Large Scale Language Models (LLMs) do not exclusively use transformers, although transformers, in particular the architecture introduced in the paper "Attention Is All You Need" by Vaswani et al. in 2017, have proven to be especially effective for natural language processing tasks (Nadkarni et al., 2011) and have been the basis of many popular LLMs such as GPT and BERT. However, before the popularization of transformers, recurrent neural networks (RNNs) and their variants, such as LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Units) networks, were commonly used to model sequences in natural language processing tasks (Sherstinsky, 2020; Tang et al., 2020).

As research in the field of artificial intelligence and natural language processing continues to advance, it is possible that new architectures and approaches emerge which may be used in conjunction with or instead of transformers in future LLMs. Thus, although transformers are currently a dominant architecture for LLMs, they are not the only architecture used, but they are one of the most reliable when it comes to generating new text that is grammatically correct and semantically meaningful (Vaswani et al., 2017). This is due to three specific elements: (a) the first is the use of positional coding mechanisms, which allow the network to assign a position to a word within a sentence so that this position is part of the network's input data. This means that the word order information becomes part of the data itself rather than part of the structure of the network, so that as the network is trained, with lots of textual data, it learns how to interpret positional coding and to order words coherently from the data used in the training; (b) secondly, attention (Bahdanav et al., 2014), which emerged as a mechanism for the meaningful translation of text from one language to another by developing algorithms to relate words to each other and thus know how to use them in an adequate context; (c) finally, self-attention or autoregressive attention, allows for better knowledge of language features, in addition to gender and order, such as synonyms, which are identified through the analysis of multiple examples.

The same is true for verb conjugations, adjectives, etc. Previous approaches that assign importance based on word frequencies can misrepresent the true semantic importance of a word; in contrast, self-attention allows models to capture long-term semantic relationships within an input text, even when that text is split and processed in parallel (Vaswani et al., 2017). Text generation is also about creating content and sequences of, for example, proteins, audio, computer code or chess moves (Eloundou et al., 2023).

Advances at the algorithm level in the development of transformers, for example, together with the current computational capacity and the ability to pre-train with unlabelled data and to refine training (fine tuning) have driven this great AI revolution. Model performance depends heavily on the scale of computation, which includes the amount of computational power used for training, the number of model parameters and the size of the dataset. Pre-training an LLM requires hundreds or thousands of GPUs and weeks to months of dedicated training time. For example, it is estimated that a single training run for a GPT-3 model with 175 billion parameters, trained on 300 billion tokens, can cost five million dollars in computational costs alone.

LLMs can be pre-trained on large amounts of unlabeled data. For example, GPT is trained on unlabeled text data, which allows it to learn patterns in human language without explicit guidance (Radford and Narasimhan, 2018). Since unlabeled data is much more prevalent than labeled data, this allows LLMs to learn about natural language in a much larger training corpus (Brown et al., 2020). The resulting model can be used in multiple applications because its training is not specific to a particular set of tasks.

General-purpose LLMs can be "fine-tuned" to generate output that matches the priors of any specific configuration (Ouyang et al., 2022; Liu et al., 2023), known as fine tuning. For example, an LLM may generate several potential answers to a given query, but some of them may be incorrect or biased. To fine-tune this model, human experts can rank the outputs to train a reward function that prioritizes some answers over others. Such refinements can significantly improve the quality of the model, making a general-purpose model fit to solve a particular problem (Ouyang et al., 2022).

2.2. History and Evolution from AI to Generative AI

Artificial intelligence is a field of computer science and technology concerned with the development of computer systems that can perform tasks that typically require human intelligence, such as learning, decision-making, problem solving, perception and natural language (Russell and Norvig, 2014). Turing addressed the central question of artificial intelligence: "Can machines think" (Turing, 1950). Soon after, it was John McCarthy who coined the term "artificial intelligence" in 1956 and contributed to the development of the Lisp programming language, which for many has been the gateway to AI (McCarthy et al., 2006). He, along with others such as Marvin Minsky (MIT), Lotfali A. Zadeh (University of Berkeley, California) or John Holland (University of Michigan), have been the pioneers (Zadeh, 2008). Trends, models, and algorithms have emerged from their work. Their work has led to the creation of schools of thought and systems have been built on its basis, bringing about real advances in fields such as medicine.

Thus, branches of artificial intelligence such as symbolic logic, expert systems, neural networks (Corchado et al., 2000), fuzzy logic, natural language processing, genetic algorithms, computer vision, multi-agent systems (González-Briones et al., 2018) or social machines (Hendler & Mulvehill, 2016; Chamoso et al., 2019) have emerged. All these branches are divided into sub-branches and these into others, such that, today, the level of specialization is high.

Most complex systems are affected by multiple elements; they generate, or are related to multiple data sources, they evolve over time, and in most cases, they contain a degree of expert knowledge (Pérez-Pons et al., 2023). In this regard, it seems clear that the combined use of symbolic systems capable of modelling knowledge together with connectionist techniques that analyze data at different levels or from different sources can offer global solutions. It is not difficult to find such problems, for example, in the field of medicine, where knowledge modelling is as important as the analysis of patient data alone. One example of model fusion was the Gene-CBR platform for genetic analysis. On the one hand, it used the methodological framework delivered with a case-based reasoning system together with several neural networks and fuzzy systems (Díaz et al., 2006; Hernandez-Nieves et al., 2021). This model was built to facilitate the analysis of myeloma.

The 1970s/80s was a breakthrough period for artificial intelligence and distributed computing (Janbi et al., 2022). A time of great change, with the Internet taking off, at a time when the world was approaching a new century and where the attention of the computing world was more focused on the potential of the Internet than on the advancement of AI. This fact, coupled with hardware limitations, industry disinterest in AI and a lack of disruptive ideas contributed to the beginning of a period of stagnation in the field, which is known as the "AI winter".

But after a winter there is a summer and this came at the turn of the century, with the emergence of what we call deep learning and convolutional neural networks (CNNs). It was a major concept that brought about a radical change in the way we deal with information. These networks use machine learning techniques in a somewhat different way to how they were originally conceived

(Bengio, 2009; Pérez-Pons et al., 2021; Hernández et al., 2021). Unlike other models, they have multiple hidden layers that allow features and patterns to be extracted from the input data in an increasingly complex and abstract manner (Parikh et al., 2022). Here, a single algorithm addresses a problem from different perspectives.

These models represent a before and after and are bound to revolutionize how we work. This is the beginning of the fifth industrial revolution thanks to our ability to create systems through the convergence of digital, physical, and biological technologies using these new models of knowledge creation (Corchado, 2023). If we lived in a fast-moving world, we must now prepare for a world of continuous acceleration. Those who keep pace with these advances will see their business, value generation and service opportunities increase exponentially in the coming years.

Deep learning is a subcategory of machine learning that focuses on algorithms inspired by the structure and function of the brain, called artificial neural networks (Chan, et al., 2016; Kothadiya et al., 2022, Alizadehsani et al., 2023). These networks, especially when they have many (deep) layers, have proven to be extremely effective in a variety of AI tasks. Deep learning-based generative models can automatically learn to represent data and generate new ones that resemble the distribution of the original data.

CNNs are a specialized class of neural networks designed to process data with a grid-like structure, such as an image. They are central to computer vision tasks. In the context of generative AI, CNNs have been adapted to generate images. For example, generative antagonistic networks (GANs) often use CNNs in their generators and discriminators to produce realistic images.

GANs, introduced by Ian Goodfellow and his collaborators in 2014, consist of two neural networks, a generator, and a discriminator, which are trained together (Goodfellow et al., 2014). The generator attempts to produce data (such as images), while the discriminator attempts to distinguish between real data and generated data. As training progresses, the generator gets better and better at creating data that deceives the discriminator. CNNs are often used in the GAN architecture for image-related tasks. On the other hand, variational autoencoders (VAEs), for example, are another type of generative model based on neural networks (Wei & Mahmood, 2020). Unlike GANs, VAEs explicitly model a probability distribution for the data and use variational inference techniques to train. In addition, pixel-based models (Su et al., 2021) are generative AI frameworks based on deep learning and generate images on a pixel-by-pixel basis, using recurrent neural networks or CNNs.

Deep learning, in particular, convolutional networks, have been fundamental tools in the development and success of many generative AI models, especially those focused on image generation. These techniques have enabled significant advances in the ability of models to generate content that is indistinguishable from real content in many cases.

For instance, ChatGPT has come into our lives and changed them, and we had hardly noticed. Some people have only heard of it, others have used it on occasions, and many of us are already working on projects and generating value with this technology. The ability of this tool to write text, to generate algorithms, to synthesize and generate reasoned proposals is extraordinary, but this is only the tip of the iceberg. It is already being used to create systems for customer service, medical data analysis, decision support and diagnostics, among others.

But ChatGPT is only the first such system to make its way into the market. There are many other models and tools: BARD, XLNet, T5, RoBERTa, Bedrock, Wu Dao, Nemo, LLAMA 2, etc. Technology such as this will allow for the development of much more accurate diagnostic systems based on evidence and clinical records, more widespread use of telemedicine, systems for monitoring chronic patients in their homes, etc. In this regard, algorithms of great interest for the medical field are being developed at different levels, such as transformers, autoencoders, deep energy-based generative models, variational inference models of prototypes, reinforcement learning systems with causal inference, to name but a few. AI has the potential to fundamentally change the way we live and work, but it also poses significant ethical challenges in terms of privacy and security that need to be addressed.

2.3. The Shift from Traditional AI to Generative AI

The history of artificial intelligence (AI) is rich and fascinating and, like everything else, it can have different interpretations and key elements. Here is a summary of some transcendental elements that allow us to analyze the evolution of this field quickly from the appearance of the first artificial neuron to the construction of the first transformer and the popularization of ChatGPT:

1. **Artificial neuron (1943)**: Warren McCulloch and Walter Pitts published "A Logical Calculus of the Ideas Immanent in Nervous Activity", where a simplified model of a biological neuron, known as the McCulloch-Pitts neuron, was presented. This model is considered the first artificial neuron and is the basis of artificial neural networks (McCulloch & Pitts, 1943).
2. **Perceptron (1957-1958)**: Frank Rosenblatt introduced the perceptron, the simplest supervised learning algorithm for single-layer neural networks. Although limited in its capabilities (e.g., it could not solve the XOR problem), it laid the foundation for the future development of neural networks (Rosenblatt, 1958).
3. **AI Winter (1970s-1980s)**: Limitations of early models and lack of computational capacity led to a decline in enthusiasm and funding for AI research. During this period, neural networks were not the focus of the AI community (Moor, 2006).
4. **Backpropagation (1986)**: Rumelhart, Hinton and Williams introduced the backpropagation algorithm for training multilayer neural networks (Rumelhart et al., 1986). This algorithm began to revive interest in neural networks. Recurrent networks, which use backpropagation, pay attention to each word individually and sequentially. These networks operate sequentially. In these networks, the order in which each word appears is considered in the training. In the context of the recurrent networks that appeared in the late 1980s and early 1990s, RNNs were developed and created to process sequences of data. To train these networks, the backpropagation through time technique (BPTT) is used. RNNs can maintain a "state" over time, which makes them suitable for tasks such as time series prediction and natural language processing. However, traditional RNNs faced problems such as gradient vanishing and gradient explosion. Recurrent networks lose context as they progress through paragraph evaluation/generation, which is a problem if the text is long. This problem was solved by other networks with backpropagation, long short-term memory (LSTM), introduced by Hochreiter and Schmidhuber (1997), a specialized variant of RNNs designed to deal with the vanishing gradient problem. LSTMs can learn long-term dependencies and have been central to many advances in natural language processing and other sequential tasks until the advent of transformers. These networks include, at each stage of learning, mathematical operations that prevent it from forgetting what was learned at the beginning of the paragraph. However, these networks have other problems related to the impossibility of parallelizing their training, making the creation of large models practically unfeasible. In this type of network, all training is sequential.
5. **Deep Learning and Convolutional Neural Networks (Convolutional Neural Networks, CNN, 2012)**: In 2012, Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton presented a convolutional neural network that won the ImageNet image classification challenge by a wide margin

(Krizhevsky et al., 2012). This event marked the beginning of the "Deep Learning" era with renewed interest in neural networks that started to become popular in 2006, the year in which the end of the "AI Winter" began. These networks are particularly suitable for classification and image processing, are structured in layers and are organized into three main components: convolutional layers, activation layers and clustering layers. Convolutional layers are responsible for extracting important features from images by means of filters or kernels. The filters glide over the image, performing mathematical operations to detect specific edges, shapes, or patterns. In activation layers, activation functions (such as ReLU) are applied to add non-linearity and increase the network's ability to learn complex relationships. Finally, the clustering layers reduce the size of the image representation, reducing the number of parameters and making the network more efficient in processing. As information passes through these layers, the CNN learns to recognize more abstract and complex features, allowing for the identification of objects, people, or anything else that needs to be identified. The work done in this field for the construction of massive information processing systems and for the development of parallel projects has given rise to the transformers that are used today (Gerón, 2022).

6. **Transformers (2017)**: Vaswani et al. introduced the transformer architecture in the paper "Attention Is All You Need". This architecture, based on attention mechanisms, proved to be highly effective for natural language processing tasks and became the basis for many subsequent models, including GPT. The advantage of these networks over backpropagation models such as LSTMs and deep learning lies in their ability to parallelize learning. Unlike recurrent neural networks (RNNs) or convolutional neural networks (CNNs), transformers do not rely on a fixed sequential or spatial structure of the data, which allows them to process information in parallel and capture long-term dependencies in the data. In this regard, the concept of word embedding, which is the basis of transformer learning, is worth mentioning. This is a technique within natural language processing for text vectorization. Transformers make it possible to analyze all the words in a text in parallel and, in this way, the processing and creation of the network is faster. That said, it should be noted that these networks require huge amounts of data and very powerful hardware, as mentioned above. For example, GPT-3 was created with 175 billion parameters and 45 TB of data, and GPT-4 with 1000,000,000,000,000 million parameters and a larger but unknown number of TB.
7. **GPT and ChatGPT (2018-2020)**: OpenAI launched the generative pre-trained transformer (GPT) model series. GPT-2, released in 2019, demonstrated an impressive ability to generate coherent and realistic text. GPT-3, released in 2020, further extended these capabilities and led to the popularization of chat-based applications such as ChatGPT (Abdullah et al., 2022). This product has had impressive penetration power, having reached 100 million users in 2 months, when other platforms such as Instagram have taken 26 months to reach the same number of users (Facebook 54 months or Twitter 65 months).

These seven elements may be regarded as a chronological list of findings and facts reflecting the evolution of AI from its origins to the emergence of what is known today as generative AI.

3. Large Language Models

In this section, we introduce large language models (LLMs). After a generic definition, selected success stories are discussed (Itoh & Okada, 2023). Rather than conducting an exhaustive study, the intention is to highlight the LLMs that are currently most relevant and comment on their distinctive aspects.

3.1. Defining Large Language Models

Large Language Models are artificial intelligence models designed to process and generate natural language. These models are trained on vast amounts of text, enabling them to perform complex language-related tasks such as translation, text generation and question answering, among others.

LLMs have become popular largely due to advances in transformer architecture and the increase in available computational capacity. These models are characterized by many parameters, allowing them to capture and model the complexity of human language.

Large Language Models have revolutionized the field of natural language processing and have several distinctive features. These are the most characteristic elements of LLMs:

- **Large number of parameters**: LLMs, as the name implies, are large. For example, GPT-3, one of the best known LLMs, has 175 billion parameters. This huge number of parameters allows them to capture and model the complexity of human language.
- **Large corpus training**: LLMs are trained on vast datasets that span large portions of the internet, such as books, articles, and websites. This allows them to acquire a broad general knowledge of language and diverse topics.
- **Text generation capability**: LLMs can generate text that is coherent, fluent and, in many cases, indistinguishable from human-written text. They can write essays, answer questions, create poetry and more.
- **Transfer learning**: Once trained on a large corpus, LLMs can be "tuned" for specific tasks with a relatively small amount of task-specific data. This is known as "transfer learning" and is one of the reasons LLMs are so versatile.
- **Use of transformer architecture**: Most modern LLMs, such as GPT and BERT, are based on a transformer architecture, which uses attention mechanisms to capture relationships in data.
- **Multimodal capability**: While LLMs have traditionally focused on text, more recent models are exploring multimodal capabilities, meaning they can understand and generate multiple types of data such as text and images simultaneously.
- **Generalization across tasks**: Without the need for specific architectural changes, an LLM can perform a wide variety of tasks, from translation to text generation to question answering. Often, all that is needed is to provide the model with the right prompt or stimulus.
- **Ethical challenges and bias**: Because LLMs are trained on internet data, they can acquire and perpetuate biases present in that data. This has led to concerns and discussions about the ethical use of these models and the need to address and mitigate these biases.

Similarly, the growth of different LLM models is exponential in time, with each LLM developer working on a wide variety of applications to meet different needs and resource levels. This includes both larger models with many parameters and smaller models with fewer parameters. Companies such as OpenAI and Google have been developing models with an ever-increasing number of parameters, where these models are able to tackle very diverse and complex tasks and often perform outstandingly well in a wide range of applications. However, the case of the META company with its Llama 2 model has created commotion due to the different parameterized versions of the model and is being optimized to be able to run in low hardware performance environments. The following Table 1 shows data regarding some of these models:

Table 1. LLM models.

Model Name	Company	Number of Parameters	Training Information Quantity	Website
GPT-3	OpenAI	175 billion	Approx. 570GB (WebText, books, others)	https://openai.com/
BERT-Large	Google	340 million	Wikipedia + BookCorpus	https://research.google/pubs/pub45413/
T5 (Text-to-Text Transfer Transformer)	Google AI	Varies depending on version (from 60 million to 11 billion)	C4 (Common Crawl)	https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html
RoBERTa	Facebook AI	Varies depending on version (up to 355 million for RoBERTa-Large)	Numerous datasets including WebText, OpenWebText, and others	https://ai.meta.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/
XLNet	Google/CMU	Up to 340 million	Various datasets including Wikipedia and BookCorpus	https://github.com/zihangdai/xlnet
CLIP	OpenAI	281 million	Internet images + associated text	https://openai.com/research/clip
DALL·E	OpenAI	Approx. 12 billion (based on GPT-3)	Images and text descriptions	https://openai.com/research/dall-e
Llama 2	Meta AI	1000 million	1000 million words	https://github.com/facebookresearch/llama
Wu Dao	Beijing Academy of Artificial Intelligence (BAAI)	1.75 trillion	4.9 terabytes of text and code	https://www.baai.ac.cn/
LaMDA	Google AI	137 billion	Google databases	https://bard.google.com/
PaLM	Google AI	540 billion	Google databases	https://bard.google.com/

3.2. Types of Large Language Models

What follows are some of the types of LLMs, and an identification of their key characteristics and potential:

1. **Autoregressive models:**
    - **GPT (Generative Pre-Trained Transformer)**: Developed by OpenAI, GPT is an autoregressive model that generates text on a word-by-word basis. It has had several versions, with GPT-3 being the most recent and advanced at the time of the last update in 2021.

2. **Bidirectional model classification**:
   - **BERT (Bidirectional Encoder Representations from Transformers)**: Developed by Google, BERT is a model that is trained bidirectionally, meaning that it considers context on both the left and right sides of a word in a sentence. It is especially useful for reading comprehension and text classification tasks.
3. **Sequence-to-sequence models:**
   - **T5 (Text-to-Text Transfer Transformer)**: Developed by Google, T5 interprets all language processing tasks as a text-to-text conversion problem. For example, "translation", "summarization" and "question answering" are handled as transformations from text input to text output.
   - **BART (Bidirectional and Auto-Regressive Transformers)**: Developed by Facebook AI, BART combines features of BERT and GPT for generation and comprehension tasks.
4. **Multimodal models:**
   - **CLIP (Contrastive Language–Image Pre-training)** and **DALL·E**: Both developed by OpenAI, these models combine computer vision and natural language processing. While CLIP is able to understand images in the context of natural language, DALL-E generates images from textual descriptions.
   - **WU DAO** is a deep learning language model created by the Beijing Academy of Artificial Intelligence that has multimodality features. It has been trained on both text and image data, so it can tackle both tasks. It was trained with many parameters (1.75 trillion).

4. Algorithmics Relevant in Field of Generative AI

Generative artificial intelligence is mainly based on unsupervised learning techniques. This differs from supervised learning models that need labelled data to orchestrate their training phase. The absence of such labelling constraints in unsupervised learning models, such as generative adversarial networks (GANs) or variational autoencoders (VAEs), allows for the use of larger and more heterogeneous datasets, resulting in simulations that closely mimic real-world scenarios (Goodfellow et al., 2016). The main goal of these generative models is to decipher the intrinsic probability distribution P(x) to which the dataset adheres. Once the model is competently trained, it possesses the ability to generate new samples of data 'x' that are statistically consistent with the original dataset. These synthesized samples are drawn from the learned distribution, thus extending the applicability of generative models in various sectors such as healthcare, finance, and creative industries (Baidoo-Anu & Owusu Ansah, 2023).

The landscape of generative AI is notably dominated by two key architectures: generative adversarial networks (GANs) and generative pre-trained transformers (GPTs). GANs operate through dual neural networks, consisting of a generator and a discriminator. The generator produces synthetic data, while the discriminator evaluates the authenticity of this data. This adversarial mechanism continues iteratively until the discriminator can no longer distinguish between real and synthetic assets, thus validating the generated content (Hu, 2022; Jovanović, 2022). GANs are mainly used for applications in graphics, speech generation and video synthesis (Hu, 2022).

There are multifaceted contributions from various architectures such as GANs, GPT models and especially variational autoencoders (VAE). The latter not only offer a probabilistic view of generative modelling, but also allow for a more flexible understanding of the underlying complex data distributions (Kingma and Welling, 2013). In addition, the advent of multimodal systems, which harmonize diverse data types in a singular architecture, has redefined the capacity for intricate pattern recognition and data synthesis. This evolution reflects the increasing complexity and nuances that generative AI can capture.

The interaction between VAEs and multimodal systems exemplifies the next frontier of generative AI. It promises not only greater accuracy, but also the ability to generate results that are rich in context and aware of variations between different types of data. In this context, generative AI has evolved from a mere data-generating tool to a progressively interdisciplinary platform capable of understanding nuances and solving complex problems in various industries (Zoran, 2021).

4.1. Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model

Deep reinforcement learning (RL) algorithms leverage high-capacity networks to directly learn from image observations. However, these high-dimensional observation spaces present challenges as the policy must now solve two problems: representation learning and task learning.

This paper proposes the Stochastic Latent Actor-Critic (SLAC) algorithm, a high-performance, high-efficiency RL method for learning policies for complex continuous control tasks directly from high-dimensional image inputs. SLAC specifically addresses the challenges posed by large observation spaces through the introduction of a stochastic latent state that summarizes the relevant information from each situation for decision making.

In this way, the SLAC actor-critic divides the general problem of learning the state representation and determining the most appropriate action into two specialized processes. On the one hand, the encoder module is responsible for compressing the observation information into a latent representation useful for the task. On the other hand, the traditional actor-critic uses this latent state to evaluate situations and select the best action, without needing to directly process the complex original image input.

SLAC proposes an innovative and robust technique to integrate stochastic sequential models with RL into a single unified framework. This is accomplished by creating a succinct latent representation, which is subsequently used to conduct RL within the latent space generated by the model. Experimental trials indicate that this approach exceeds the performance of both model-free and model-based competitors in terms of final results and the efficient use of samples, especially across a spectrum of complex image-based control tasks.

In the context of generative environment models for RL, these findings emphasize the importance of latent representation learning that can accelerate reinforcement learning from images, representing a significant advance in the domain of machine learning and generative artificial intelligence.

In traditional deep reinforcement learning (DRL) paradigms, such as q-learning or policy gradients, the objective function is often adapted to maximize the expected performance $J(\theta)$ defined as:

$$J(\theta) = E_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R(z_t, a_t)\right]$$

Where τ represents a trajectory and γ is the discount factor.

In the context of DRL with a latent variable model, the incorporation of a latent space z adds an additional layer of abstraction. Specifically, state s is replaced or augmented by a latent variable z, which in turn can be a function of the state $z = f(s)$ or can be learned from the unsupervised data.

$$J(\theta,\phi) = E_{\tau,z\sim\pi_{\theta,\phi}}\left[\sum_{t=0}^{\infty} \gamma^t R(z_t,\, a_t)\right]$$

Where ϕ are the parameters of the latent variable model and the policy π is now conditioned not only on *S* but also on z.

The success of this approach lies in the quality of the learned latent space and how well it captures the nuances needed for optimal decision making. It is a promising line of research, and the advances here could potentially revolutionize the way we approach complex, high-dimensional, partially observable decision-making problems in DRL.

4.2. Video: High-Definition Video Generation with Diffusion Models

The sequential composition of spatial and temporal super-resolution video models exhibits an ingenious architecture, as it not only elevates pixel-level fidelity, but also ensures temporal coherence between the generated frames (Simonyan & Zisserman, 2014; Xie et al., 2018). This facet is particularly beneficial for applications that require dynamic scene rendering and fluid motion rendering; conventional image-based generative models do not meet those conditions (Jiang et al., 2018). Furthermore, the system's ability to scale towards text-to-high-definition video output, made possible by its fully convolutional architecture, represents a significant advance over static image generation. In traditional generative models, the computational complexity for video generation often scales poorly, making high-definition results computationally infeasible (Vondrick et al., 2016). In contrast, the modular image-video cascade structure facilitates more efficient resource allocation, enabling high-quality video sampling in substantially shorter periods of time (Tulyakov et al., 2018).

The integration of diffusion models into the existing image-video architecture generates a compelling fusion of technologies that can serve to augment its generative capabilities. By introducing the temporal component with a range of 0 to 1, the diffusion model imparts a temporally consistent level of stochasticity and granularity to the generated videos, enriching the high-level representations of VAE (Kingma & Welling, 2013; Ho et al., 2020).

This hybrid model can be conceptually visualized as operating in two stages: first, the VAE *Encoder*(*x*) encoder function computes a latent variable z from the input x. This z subsequently serves as the initial condition for the diffusion model, essentially fulfilling the role of x in its equations. Next, a time-dependent denoising function $D_t(z_t)$, is introduced, which refines the high-level representations of the VAE over time. It culminates in a synthesis function x^θ (*z*,*t*)=*Decoder*(*z*) + $D_t(z_t)$, thus orchestrating a richer, temporally smoothed output.

The objective function L(x) for the diffusion model then becomes:

$$L(x) = E_{\epsilon\sim N(0,I),t\sim U(0,1)}\left\|\hat{\epsilon}_\theta(z_t,\lambda_t)-\epsilon\right\|_2^2$$

Where $z_t = \alpha_t x + \sigma_t\epsilon,\ y\ \widehat{\epsilon_\theta}(z_t,\lambda_t) = \sigma_t^{-1}\left(z_t - \alpha_t\widehat{x_\theta}(z_t,\lambda_t)\right)$.

In this way, the diffusion model learns to adjust the high-level representations generated by the VAE over time, thus providing a richer, temporally smoothed generative model that takes advantage of the strengths of both architectures.

4.3. Motion Diffuse: Text-Driven Human Motion Generation with Diffusion Model

In the MotionDiffuse architecture for text-based human motion generation, special attention should be paid to the design of loss functions that align with the attention structure embedded in the model. Given that Motion Diffuse employs attention mechanisms to modulate the relationship between text and motion spaces, two prominent loss functions could be of critical importance.

Incorporating specialized loss functions such as those in the GLIDE model could further refine the MotionDiffuse framework, especially with its unique approach to text-based human motion generation. In MotionDiffuse, the incorporation of a stochastic noise term, guided by the equation $\alpha_t = 1 - \beta_t$, $\overline{\alpha_t} = \Sigma_{s=0}^{t} \alpha_s$,offers an interesting avenue for loss function innovation.

In the original MotionDiffuse methodology, the model focuses on predicting a noise term instead of x_{t-1} in line with the GLIDE framework (Nichol et al., 2021). This is captured by a mean squared error (MSE) loss function:

$$L = E_{t \in [1,T], x_0 \sim q(x_0), \epsilon \sim N(0,I)} \left[\| \epsilon - \theta(x_t, t, \text{ text}) \| \right]$$

Consider the integration of the following specialized loss functions:

Loss of Attentional Alignment Between Modalities

This loss function ensures that the attention distributions between text and motion spaces are aligned. Since the attention scores a_t and a_m at meter are derived from the text and motion encoders respectively, the loss can be defined as the Kullback-Leibler divergence between the two:

$$L_{attention} = KL(a_t \parallel a_m)$$

This loss encourages the model to pay attention to semantically similar regions in both the text and motion domains, thus encouraging better alignment between modalities.

Text-Guided Motion Fidelity Loss

To ensure that the generated motion is not just any motion but one that specifically aligns with the text description, a text-guided motion fidelity loss can be introduced. This would measure the consistency between the high-level features extracted from the generated motion and those dictated by the text. Let F_t and F_m be the high-level text and motion features:

$$L_{fidelity} = \|F_t - F_m\|_1$$

Coupling these loss functions with traditional generative losses such as mean square error (MSE) or generative adverse loss ensures that Motion Diffuse generates motions that are not only qualitatively good, but also well aligned with textual descriptions. This would address some of the challenges in generating motions that are diverse and textually consistent.

By intricately linking these loss functions with the attentional mechanisms within the model, MotionDiffuse can facilitate more nuanced, temporally coherent, and contextually relevant motion sequences guided by textual descriptions.

By merging these specialized loss functions into a unified loss term, we are able to write:

$$L_{total} = \lambda_1 L_{MSE} + \lambda_2 L_{attention} + \lambda_3 L_{fidelity}$$

Here, λ_1, λ_2, λ_3 are the hyperparameters that control the balance between the different loss components.

4.4. The Emergence of Diffusion Models and Latent Space Dynamics in Text, Audio, and Video Generation

This is a recent innovation in the field of probabilistic denoising diffusion denoising models (DDPMs). Unlike conventional DDPMs, LDMs operate in latent space and are conditional on textual representations (Rombach et al., 2022). This allows for a step-by-step granular generation process involving several conditional diffusion steps.

The loss function is meticulously defined as the mean square error in the noise space (ξ∽N(0, I)\). Mathematically, this can be formulated as:

$$L_\theta = \parallel \xi_\theta(z_t, t, c) - \xi \parallel 22,$$

Where α is a small positive constant and ξ_θ represents the denoising network. This loss function allows for efficient training by optimizing a random term t with stochastic gradient descent. It is worth noting that this approach allows for the diffusion model to be efficiently trained by optimizing the evidence lower bound (ELBO) without requiring adversarial feedback, resulting in remarkably faithful reconstructions that match the ground truth distribution (Huang et al., 2023).

In 2023, the generative artificial intelligence landscape is undergoing a seismic shift, marked predominantly by the maturation of diffusion models. In the text domain, while transformers were once lauded for their versatility in natural language understanding and generation, diffusion models are now catalyzing a new wave of innovation, offering nuanced language models with more robust generalization capabilities. They are extending the fundamental work of earlier models, adding layers of complexity and applicability in sentiment analysis, abstract summarization, and more (Nichol et al., 2021).

In the audio sector, models such as Make-An-Audio are revolutionizing text-to-audio (T2A) generation using enhanced broadcast techniques (Zhao et al., 2023). They address the inherent complexity of long, continuous signal data, something that previous methods such as WaveGAN and MelGAN struggled with. This results in higher fidelity audio generation and nuanced semantic understanding, a radical shift from previous methods.

Video generation has also been strengthened by high-definition broadcast models, moving beyond the resolution limitations, which often plagued older GAN-based methods. Here, the focus is not only on pixel-level detail, but also on semantic coherence between video frames, thus achieving a new standard in the realism of the generated content (Rombach et al., 2022).

In cross-modal generative learning, the advent of stochastic latent actor-critic models integrated with diffusion models offers the possibility of seamless and more accurate translation between different data types. These hybrid models are beginning to unlock capabilities for high-definition, high-fidelity generation in text, audio, and video modalities (Haarnoja et al., 2018).

Taken together, these advances in text, audio, and video broadcast models are setting the stage for a future of multimodal generative AI that is richer, more dynamic, and more far-reaching in its applications, thus revolutionizing the broader landscape of data analytics, content creation, and automated decision making.

5. Chat GPT, Its Potential and How to Take Advantage of It

5.1. Description of GPT

Generative pre-trained transformer (GPT) models represent a significant innovation in the field of natural language processing (NLP) and artificial intelligence (AI). Developed by OpenAI, GPT models are based on the transformer architecture, which was introduced by Vaswani et al. (2017).

GPT models are pre-trained on large corpora of text and then tuned on specific tasks, allowing them to generate text that is consistent, grammatically correct, and often indistinguishable from human-generated text (Radford et al., 2019).

GPT models use multiple layers of attention and a combination of multi-head attention, allowing them to capture a variety of features and relationships in data. The ability of GPT models to understand and generate text has led to advances in a variety of applications, including machine translation, creative text generation, and question answering (Brown et al., 2020).

5.1.1. Architecture and Attention

The transformer architecture, introduced by Vaswani et al. (2017), has revolutionized the field of natural language processing and represents a paradigm shift in the way sequences are modeled.

Unlike traditional recurrent architectures (Bahdanau, Cho, & Bengio, 2014), transformers eliminate recurrence and instead use attention mechanisms to capture dependencies in the data.

The architecture consists of two main parts: the encoder and the decoder; both are composed of a stack of identical layers containing two main sublayers: a multi-head attention sublayer and a feed-forward network.

The multi-head attention allows the model to simultaneously attend to different parts of the input, thus capturing complex and long-term relationships. The forward feed-forward network consists of a simple linear transform followed by a nonlinear activation function.

Transformer

A key feature of the transformer architecture is the addition of residual connections around each of the sublayers (He, Zhang, Ren, & Sun, 2016), followed by layer normalization (Ba, Kiros, & Hinton, 2016). This facilitates training and allows gradients to flow more easily through the network. The combination of attention, residual connections, and layer normalization allows transformers to be highly parallelizable and computationally efficient, which has led to their widespread adoption in a variety of NLP tasks (Devlin, Chang, Lee, & Toutanova, 2018), see Figure 1.

Attention is a central concept in the transformer architecture. It allows the model to weigh different parts of the input when generating each word of the output, which facilitates deep contextual understanding. Attention can be described in three main components:

Queries, keys, and values: Attention is computed using queries, keys, and values, which are vector representations of the words in the input. Queries and keys determine the attention weighting, while values are weighted according to these weights to produce the output (Vaswani et al. 2017).

Multi-head attention: Multi-head attention allows the model to pay attention to different parts of the input simultaneously. Each "head" of attention can focus on different relations in the text, and the outputs from all the heads are combined to form the final representation.

Autoregressive Attention: In the case of GPT, autoregressive attention is used, where each word can only pay attention to the previous words in the sequence. This ensures that text generation is performed in a causal and coherent manner (Radford et al., 2018; Brown et al., 2020), see Figure 2.

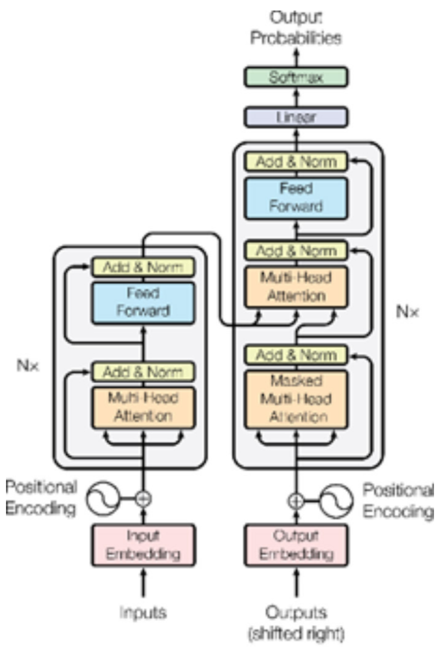


Figure 1. Transformer - model architecture (Vaswani, A., et al., 2017).

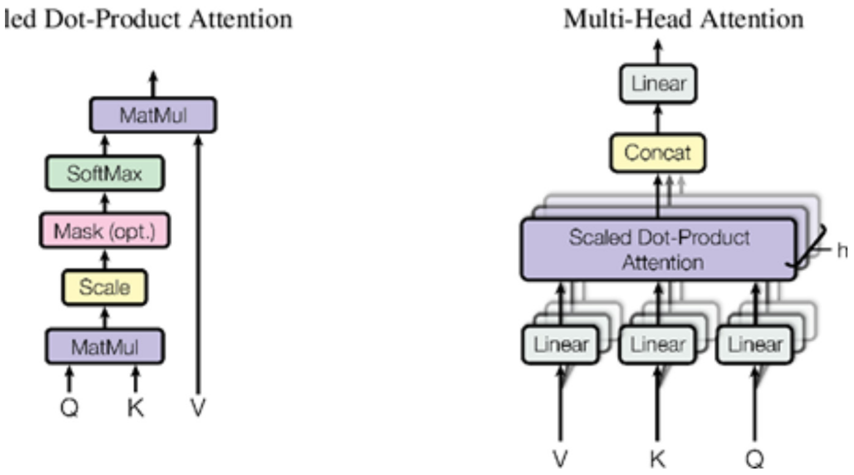


Figure 2. (left) Attention to scale product point. (right) multi-head attention consists of several layers of attention running in parallel (Vaswani, A., et al., 2017).

Layers and Feed-Forward Network

The transformer architecture consists of a series of layers, each with a multi-head attention layer followed by a densely connected feed-forward network. Residual connections and layer normalization are applied at each stage to facilitate training and improve stability (He et al., 2016; Ba et al., 2016).

5.1.2. Pre-Training and Fine-Tuning

Pre-training and fine-tuning of the OpenAI GPT model are two crucial phases in the process of developing and adapting transformer-based language models. In particular, GPT is a leading paradigm in natural language processing research and represents a significant advance in the generation of coherent and contextually relevant text.

GPT Pre-Training

Pre-training is the initial phase in the creation of a GPT model and is founded on the key idea of knowledge transfer. During this stage, the model is trained on large amounts of untagged textual data. This is where GPT learns linguistic, semantic, and contextual patterns at a deep level. This process is carried out using an autoregression task, where the model predicts the next word in a sentence based on previous words (Vaswani et al., 2017). The attentional mechanism present in the transformer-like architecture allows the model to capture long-term relationships in the text, which enables it to generate coherent and contextually relevant responses.

GPT pre-training involves optimizing millions of parameters in the model using optimization algorithms such as stochastic gradient descent. Key hyperparameters, such as learning rate and transformer architecture, play a critical role in this process (Vaswani et al., 2017). The importance of tuning these hyperparameters appropriately to achieve successful pretraining is worth noting.

Fine-Tuning of GPT

After the pre-training phase, the GPT model is enriched with deep language knowledge. However, to adapt the model to specific tasks, such as generating text in a particular domain or answering specific questions, fine-tuning is necessary. During this phase, the model is trained on a set of labeled data relevant to the specific task to be addressed.

Tuning involves adjusting the weights of the pre-trained model using the task-specific dataset. Tuning involves a smaller learning rate compared to pre-training and usually requires fewer iterations because the knowledge previously acquired by the model is more specialized.

When approaching tuning, it is critical to choose the right dataset and design an effective evaluation strategy. The evaluation metrics must be aligned with the objectives of the task. For example, in sentiment analysis, metrics such as precision, recall and score could be considered. Ensuring adequate validation and test sets are also essential to avoid overfitting (Radford et al., 2019).

5.1.3. The Evolution of GPT

The following is a review of the evolutions of the GPT models developed by OpenAI, leading up to GPT-4. The GPT series of models represents one of the most significant advances in the field of artificial intelligence, specifically in natural language processing (NLP), see Figure 3.

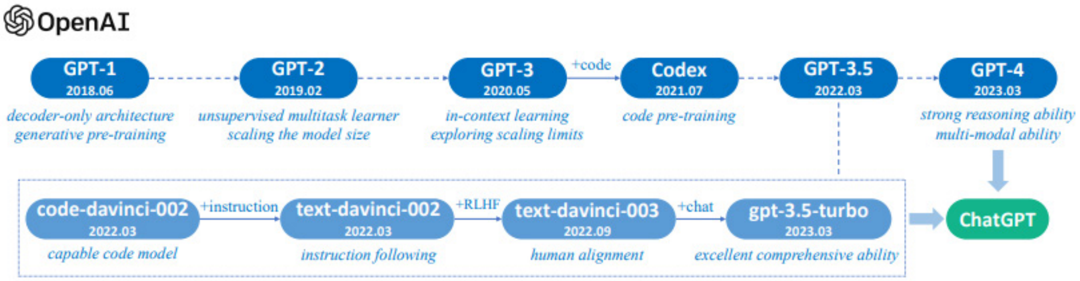


Figure 3. A brief illustration of the technical evolution of the GPT series models (OpenAI, 2023).

GPT-1

Release: June 2018 by OpenAI.

Architecture: Transformer by Vaswani et al. (2017).

Parameters: 117 million.

Technical details: Uses a transformer architecture with 12 layers, 12 attention heads, and 768 hidden units.

Major contribution: Introduction of a transformer architecture with prior training for natural language processing tasks (Radford et al. 2018).

GPT-2

Release: February 2019 by OpenAI.

Parameters: 1.5 billion.

Technical details: GPT-2 featured an expansion in scale, being ten times larger than its predecessor. The model was trained on a diverse and larger corpus. 48 layers, 1600 hidden units, and 25.6 billion parameters in its largest version.

Major contribution: demonstrated that large-scale language models can generate consistent, high-quality text across paragraphs, which was useful for various applications and machine translation (Radford et al. 2019).

GPT-3

Release: June 2020 by OpenAI.

Architecture: GPT-2 architecture extension.

Parameters: 175 billion.

Technical Details: GPT-3 further extended the scale and featured several architecture and training improvements. Trained on a variety of tasks without the need for specific tuning.

Major Contribution: Highlighted the ability to perform “few-shot learning,” where the model can learn tasks with only a few examples. This model has been used in a wide range of applications, from content writing and editing to programming and user interface design, showing unprecedented versatility in the field (Brown et al. 2020).

Table 2 shows learning sizes, architectures, and hyperparameters.

Table 2. Sizes, architectures and learning hyperparameters (batch size in tokens and learning rate) of the models.

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	$1.6 \text{ x } \mathrm{I}0^{-4}$
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}

CODEX

Release: A variant of GPT-3, designed specifically for scheduling tasks (Zaremba & Brockman, 2021).

Architecture: Based on the transformer architecture and is a variant of GPT-3 designed specifically for scheduling tasks.

Parameters: 175 billion parameters.

Technical details: Pre-trained on a large corpus of source code and documentation. It is primarily used to generate code and assist in programming tasks.

Major contribution: It has proven to be able to generate functional code in multiple programming languages. Its release has revolutionized the way developers interact with code, providing a powerful tool for automatic code generation and programming assistance (OpenAI, 2021).

5.2. ChatGPT, GPT-4

Language models based on the transformer architecture have undergone rapid development in recent years, culminating in the series of generative pre-trained transformer (GPT) models developed by OpenAI. In particular, GPT-4 and ChatGPT represent significant advances in the field of natural language understanding and generation, respectively. These models are the follow-up to the aforementioned models and have been, over the last few months, the beginning of a new era in text generative AI models. Both models are based on the transformer architecture, which has proven to be highly efficient for natural language processing tasks (Vaswani et al., 2017).

CHATGPT (GPT-3.5)

ChatGPT is a language model developed by OpenAI and is based on the GPT (Generative Pre-trained Transformer) architecture. It is a tuned variant of GPT-3 designed specifically for conversations and chat tasks. The model has been trained on a large corpus of Internet texts, but it is not known which specific documents were used in its training dataset. It has been designed to perform tasks ranging from creative text generation to programming and technical problem solving. This model is freely accessible through ChatGPT (openai.com).

Release: ChatGPT was released by OpenAI in 2021 according to Nakano et al. (2021).

Architecture: ChatGPT is also based on the transformer architecture and is a variant of GPT-3 optimized for conversations.

Parameters: ChatGPT has 175 billion parameters, similar to GPT-3.

Technical details: It is pre-trained on a corpus of conversations and text. Designed for chatbots and interactive conversational systems.

Key Contribution: ChatGPT has improved consistency and contextualization in conversations compared to previous models.

GPT-4

GPT-4 is the fourth iteration of the GPT series and represents a quantum leap in terms of capabilities, limitations, and associated risks. With a yet undisclosed number of parameters, GPT-4 has shown significant improvements in text generation, contextual understanding, and adaptability to various tasks. This model is available in the paid version of OpenAI called ChatGPT Plus, which includes certain restrictions of the GPT-4 model and new tools available such as plugins (OpenAI, 2023), code interpreter and custom instructions according to OpenAI (2023), Figure 4 shows the different work contexts.

The plugins allow developers to add specific functionality to the model, such as web search, language translation and querying scientific databases. The publication stresses that these plugins are designed to be secure and reliable and undergo a rigorous review process before approval.

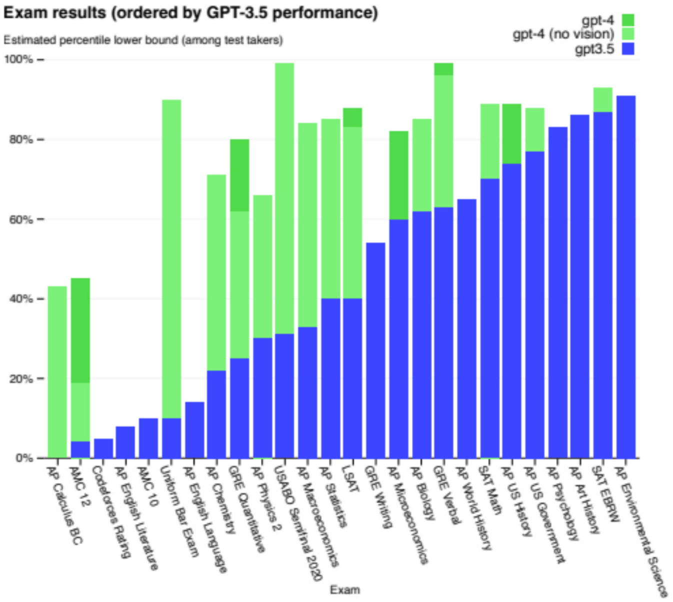


Figure 4. Performance on scholastic and occupational assessments. For every test, we mimic real examination environments and grading systems. The assessments are arranged in ascending order according to the performance of GPT-3.5. GPT-4 surpasses GPT-3.5 in the majority of the evaluated exams (OpenAI, 2023).

Release: The technical report on GPT-4 was published by OpenAI in March 2023.

Architecture: GPT-4 is a model based on the transformer architecture that can accept image and text inputs and produce text outputs.

Parameters: The report does not specify the exact number of parameters but highlights that the model is large scale.

Technical details: GPT-4 is pre-trained to predict the next token in a document.

The post-training alignment process results in improved performance on measures of factuality and adherence to desired behavior. Infrastructure and optimization methods were developed that behave predictably over a wide range of scales.

Major contribution: GPT-4 is a large multimodal model (accepting image and text inputs, emitting text outputs) that, while less capable than humans in many real-world scenarios, exhibits human-level performance on several professional and academic benchmarks (OpenAI, 2023).

5.3. Prompts

Prompts are instructions, questions or statements designed to invoke a specific response from a natural language processing (NLP) model, such as OpenAI's GPT models. In the context of chatbots and other PLN applications, prompts serve as initial inputs that guide the model in generating a coherent and relevant response. Their role is critical in controlling and directing the interaction with the model, influencing the quality, accuracy, and context of the generated response.

5.3.1. Strategies for Establishing Good Prompts

- Clarity and precision: A good prompt should be clear and precise, avoiding ambiguities that may lead to confusing or incorrect answers (Reiter et al., 2020).
- Contextualization: Including the necessary context helps the model understand the intent behind the query, improving the relevance of the response (Chen et al., 2019).
- Use of examples: In few-shot learning, providing examples within the prompt can help the model understand the intended task (Brown et al., 2020).
- Iterative experimentation: Iterative experimentation and adjustment of the prompt allows to fine-tune the interaction and obtain optimal responses (Wallace et al., 2019).
- Ethical considerations: Prompts should be formulated with awareness of potential biases and comply with privacy and ethics regulations (Hovy & Spruit, 2016).

Prompts play a central role in the design of and interaction with generative language models and other PLN applications. Their formulation and management involve a combination of technical, linguistic, and ethical considerations. The literature in PLN offers a wide spectrum of research and techniques related to the effective use of prompts, and their study and application continue to be a vital area in the interface between humans and artificial intelligence systems.

5.3.2. Types of Prompts and Their Uses

- Informative prompts: Designed to request specific information, they are useful in applications such as data search and virtual assistants (Manning et al., 2008).
- Interrogative prompts: Formulated as questions, they are employed to invoke detailed responses in areas such as customer support and tutoring (Jurafsky & Martin, 2019).

- Instructional prompts: Used to direct the model to perform a particular task, such as translation, summarization, or creative content generation (Reiter & Dale, 2020).
- Contextual prompts: Include additional context to guide the model's response in complex scenarios, such as ongoing dialogues or specialized tasks (Serban et al., 2017).
- Comparative prompts: Designed to solicit comparisons, analyses, or evaluations between entities or concepts.
- Code generation prompts: Used in programming environments to automatically generate code snippets, such as in OpenAI Codex (OpenAI, 2021).
- Linguistic assistance prompts: Applied in tools for grammar correction, translation, and other linguistic services, such as Grammarly or Google Translate.

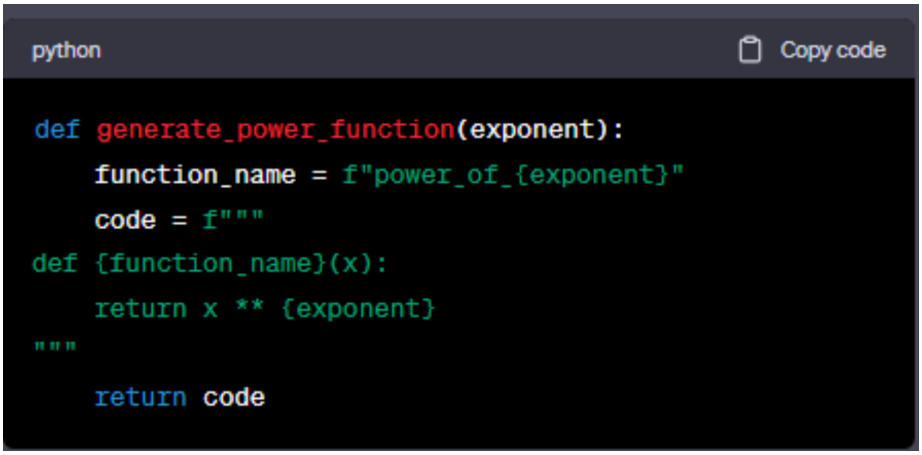
5.3.3. Example: Code Generation

Automatic code generation is a technique used to produce source code programmatically. This practice is especially useful when a large amount of similar or repetitive code needs to be generated. Instead of manually writing each piece of code, developers can use code generators to automate the process. In this article, we will explore how code can be generated automatically with Python and present a simple example of code generation for creating mathematical functions, see Figures 5, 6, 7.

Why Generate Code Automatically?

Efficiency: Reduces the time required to write code.
Consistency: Ensures that the generated code follows a specific pattern.
Flexibility: Allows global changes in the code with minimal effort.

Example of code generation in Python
Automatically generate Python functions that calculate the square, cube, and fourth power of a given number.
First, let's define a Python function that takes an exponent as a parameter and returns the source code of a function that raises a number to that exponent.

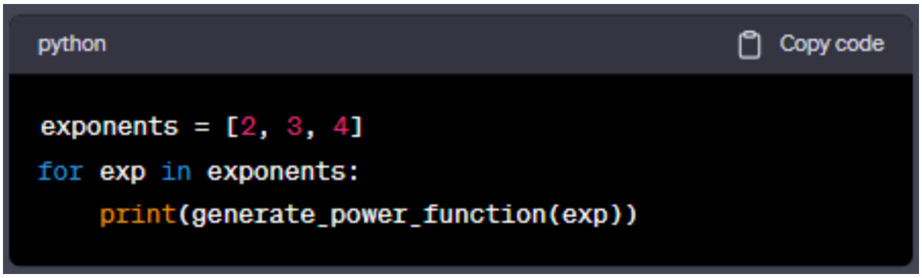


```python
def generate_power_function(exponent):
    function_name = f"power_of_{exponent}"
    code = f"""
def {function_name}(x):
    return x ** {exponent}
"""
    return code
```

Figure 5. Python exponent function.

Generating Functions

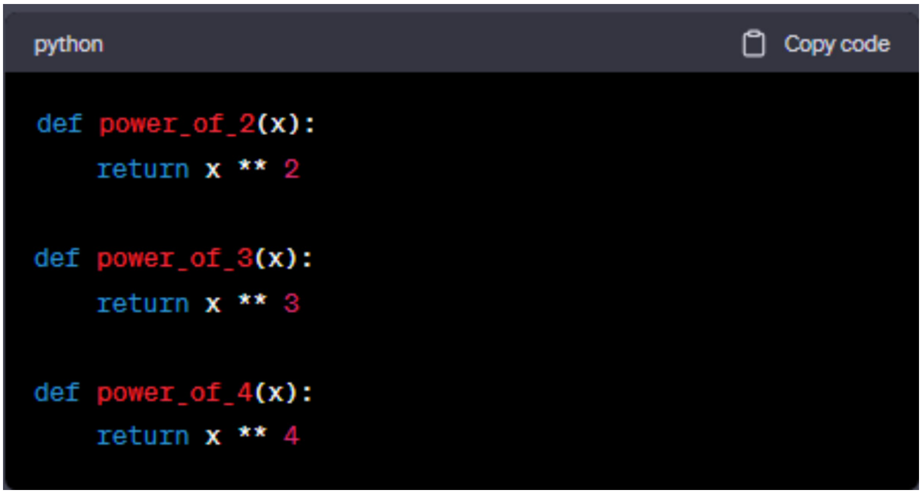
Now, we will use this function to generate code for functions that calculate the square, cube and fourth power of a number.



```python
exponents = [2, 3, 4]
for exp in exponents:
    print(generate_power_function(exp))
```

Figure 6. Python list function.

The generated code looks like this:



```python
def power_of_2(x):
    return x ** 2

def power_of_3(x):
    return x ** 3

def power_of_4(x):
    return x ** 4
```

Figure 7. Python math function.

The generate_power_function function takes an exponent and returns a code snippet that defines a new Python function. This new function takes a parameter x and returns x raised to the provided exponent.

The for loop in the "generating functions" segment runs through a list of exponents. For each exponent, generate_power_function is called to generate the corresponding code.

Automatic code generation is a powerful technique that can save time and effort, especially when dealing with repetitive or similar code. The presented example is quite simple, but code generation is used in much more complex applications, including frameworks and software libraries.

5.4. Guidelines and Documentation

5.4.1. Ethics and Responsible Use

ChatGPT is generally accessed through an API provided by OpenAI, which requires authentication to ensure the security and integrity of interactions with the model, is highly versatile and can be implemented in a variety of applications, from customer service chatbots to virtual assistants and decision-making systems. The API allows for a wide range of customizations, including the ability to define specific prompts or instructions to guide the model's behavior in different contexts. In addition, OpenAI provides detailed documentation covering technical aspects such as authentication, prompt structuring and interpretation of the responses generated by the model (OpenAI, 2021).

Access to ChatGPT is not limited only to interaction through the API, it is also accessible through a web interface provided by OpenAI. This web interface provides an intuitive and easy-to-use means of interacting with the model without requiring advanced technical knowledge. Users simply enter their prompts or questions in a text box, and the model generates answers that are displayed in the same interface. This method of access is especially useful for non-technical users or for those who wish to test the model's capabilities without having to integrate it into a larger application or system. In addition, the web interface often includes additional features, such as the ability to adjust parameters such as temperature and maximum response length, which provides greater control over text generation (OpenAI, 2021).

It is crucial to keep ethical and responsible use guidelines in mind when interacting with ChatGPT, especially in applications that may have significant social or cultural implications. OpenAI provides specific guidelines to address these issues, including the prevention of plagiarism and the generation of inappropriate content.

5.4.2. Ethics and Responsible Use

The documentation also addresses ethical and responsible use issues. This includes guidelines on the prevention of plagiarism, the generation of inappropriate content, and consideration of the social and cultural implications of using large-scale language models. These models, trained on large datasets, have the potential to generate content that may be discriminatory, biased, or even dangerous (Hao, 2020; Bender et al., 2021). Therefore, it is imperative to address ethical issues from a multidisciplinary perspective that includes both technical and social aspects.

One of the most pressing challenges is the inherent bias in training data, which can perpetuate existing stereotypes and biases (Caliskan et al., 2017). Researchers are exploring methods to mitigate these biases, such as parameter adjustment and reweighting of training data (Zhao et al., 2018).

Transparency in the operation of models and traceability of their decisions are fundamental to the ethical use of AI. This is especially relevant in critical applications such as healthcare and the judicial system, where a misguided decision can have serious consequences (Doshi-Velez et al., 2017).

Responsible use involves implementing safeguards, such as content moderation systems and alerts for inappropriate content. It is also crucial to educate users about the limitations of these models and how to interpret their responses critically (McGregor et al., 2020).

Ethics in AI is a rapidly developing field that requires continued collaboration between engineers, ethicists, legislators, and other relevant stakeholders to ensure that the technology is used in a way that is beneficial to society as a whole.

5.4.3. Applications in Computer Programming and Education

OpenAI Codex has demonstrated significant performance on typical introductory programming problems. Its performance was compared to that of students taking the same exams, showing that Codex outperforms most students. In addition, we explored how Codex handles subtle variations in problem wording, noting that identical input often leads to very different solutions in terms of algorithmic approach and code length. This study also discusses the implications that such technology has on computer science education as it continues to evolve (Finnie-Ansley et al., 2022).

Using OpenAI Codex as a large language model, programming exercises (including sample solutions and test cases) and code explanations were created. The results suggest that most of the automatically generated content is novel and sensible, and in some cases is ready to be used as-is. This study also discusses the implications of OpenAI Codex and similar tools for introductory programming education and highlights future lines of research that have the potential to improve the quality of the educational experience for both teachers and students (Sarsa et al., 2022).

GitHub Copilot, powered by OpenAI Codex, has been evaluated on a dataset of 166 programming problems. It was found to successfully solve about half of these problems on its first attempt and solved 60% of the remaining problems using only natural language changes in the problem description. This study argues that this type of prompt engineering is a potentially useful learning activity that promotes computational thinking skills and is likely to change the nature of code writing skill development (Denny et al., 2022).

5.4.4. Applications in Academic Publishing

ChatGPT is seen as a potential model for the automated preparation of essays and other types of academic manuscripts. Potential ethical issues that could arise with the emergence of large language models such as GPT-3, the underlying technology behind ChatGPT, and their use by academics and researchers are discussed, placing them in the context of broader advances in artificial intelligence, machine learning, and natural language processing for academic research and publication (Lund et al., 2023).

The release of the model has led many to think about the exciting and problematic ways in which artificial intelligence (AI) could change our lives in the near future. Considering that ChatGPT was generated by fine-tuning the GPT-3 model with supervised and reinforcement learning, the quality of the generated content can only be improved with additional training and optimization. There are a number of opportunities, as well as risks associated with its use, as the inevitable implementation of this disruptive technology will have far-reaching consequences for medicine, science, and academic publishing (Homolak, 2023).

It also appears to be a useful tool in scientific writing, assisting researchers and scientists in organizing material, generating an initial draft and/or proofreading. Several ethical issues are raised regarding the use of these tools, such as the risk of plagiarism and inaccuracies, as well as a possible gap in accessibility between high-income and low-income countries (M. Salvagno et al., 2023).

5.4.5. Conversational Systems

Chatbots are being applied in various fields, including medicine and healthcare, for human-like knowledge transfer and communication. In particular, machine learning has been shown to be applicable in healthcare, with the ability to manage complex dialogues and provide conversational flexibility. This review focuses on cancer therapy, with detailed discussions and examples of diagnosis, treatment, monitoring, patient support, workflow efficiency, and health promotion. In addition, limitations and areas of concern are explored, highlighting ethical, moral, safety, technical, and regulatory issues (Xu, L. et al., 2021).

Machine learning (ML) is a study of computer algorithms for automation through experience. The application of ML in healthcare communication has proven to be beneficial to humans. This includes chatbots for health education in COVID-19, cancer therapy, and medical imaging. The review highlights how the application of ML/AI in healthcare communication is able to benefit humans, including complex dialogue management and conversational flexibility (Sarkar Siddique & James C. L. Chow, 2021).

This article provides an overview of the opportunities and challenges associated with using ChatGPT in data science. It discusses how ChatGPT can assist data scientists in automating various aspects of their workflow, including data cleaning and preprocessing, model training, and results interpretation. It also highlights how ChatGPT has the potential to provide new insights and improve decision-making processes (Hossein Hassani & E. Silva, 2023).

5.4.6. Risk

This technology has shaken the foundations of many industries based on content generation and writing, so it is advisable to discuss good practices for writing scientific articles with ChatGPT and other artificial intelligence language models (A. Castellanos-Gómez, 2023). There are different sources of opinion as to the use of this technology and how it should be done.

These language models show significant advances in reasoning, knowledge retention, and programming compared to their predecessors. However, these improvements also bring new security challenges, including risks such as the generation of harmful content, misinformation, and cybersecurity. Despite mitigation measures, similar limitations remain in the model as in its previous versions, such as the generation of biased and unreliable content. In addition, its increased consistency could make the generated content more credible and, therefore, potentially more dangerous (OpenAI, 2023).

The risks to be considered in the use of these models are listed. As always, the person who makes use of these tools will ultimately be responsible, since, to date, a thorough supervision of the generated content is more than necessary to corroborate it:

- Misinformation.
- Harmful content.
- Damage to the performance, assignment, and quality of service.
- Disinformation and influence operations.
- Proliferation of conventional and non-conventional weapons.
- Privacy.
- Cybersecurity.

- Interactions with other systems.
- Economic impacts.
- Excessive dependence.

According to Gao et al. (2023) the scientific abstracts generated by ChatGPT were compared with the original abstracts using an artificial intelligence output detector, a plagiarism detector, and blind peer review. The results showed that the ChatGPT-generated abstracts were clearly written, but only 8% followed journal-specific formatting requirements. Although the generated abstracts were original with no detected plagiarism, they were often identified by using an AI output detector and by skeptical human reviewers. The conclusion is that ChatGPT writes credible scientific abstracts, however, it also raises ethical issues and challenges regarding the accuracy, completeness, and originality of the generated content.

6. Llama 2

Large language models (LLM) have proven to be very promising at the enterprise level. This is the case of the Meta company, which within its visions on artificial intelligence has sought its decentralization, where organizations can customize their virtual assistants and open-source models can be trained with expert knowledge. That is why Meta, a leader in the innovation and technology market headed by Mark Zuckerberg, has partnered with Microsoft Azure to present its generative AI tool Llama 2 (Touvron, H et al., 2023), considered a rival to ChatGPT and Bard differentiating itself by being an open source and not closed product where such LLMs are largely adjusted to human preferences placing security and private information at stake.

6.1. What is Llama 2 and what Are Its Features?

Llama 2 is a family of pre-trained and tuned LLMs using generative transformer models, where publicly available data has been taken and the context length has been increased by 40% with respect to the Llama 1 version. Variants of the model, versions with parameters of 7B, 13B, 34B and 70B, have been made available to the open-source communities. Llama 2 adheres to a model of free license and unrestricted distribution, creating multiple opportunities for various domains (Touvron, H. et al., 2023). Table 3 shows the different Llama 2 models.

Table 3. Model Llama 2 architecture.

MODEL PARAMETER SIZE	PRETRAINED	FINE-TUNED FOR CHAT USE CASES
7B	Model Architecture Pretraining tokens: 2 trillion Context length: 4096	Data collection for helpfulness and safety Supervised fine-tuning: Over 100,000 Huma Preferences: Over 1,000,000
13B		
70B		

6.2. Characteristics

Technology stack: Llama 2 makes use of a robust technology stack for its operation, leveraging tools and libraries ranging from natural language processing to computational efficiency. Some of the key technologies included in this stack are as follows:

- Python: The predominant programming language in the data science and machine learning community. It is used to develop and execute the code in Llama 2, allowing for clear and efficient syntax.
- PyTorch: A widely used deep learning library. Llama 2 leverages PyTorch to build and train language models, as well as to perform operations on neural networks and tensors (Paszke, A. et al., 2019).
- Conda: A package and environment management system that makes it possible to create isolated environments for specific projects. In Llama 2, Conda makes it possible to manage dependencies and versions of the used libraries, ensuring consistency in the development environment (Chaplin, J. R., et al., 2012).
- Google/SentencePiece: SentencePiece is a library for tokenization and segmentation of text in multiple languages. Llama 2 employs Google/SentencePiece for the text segmentation task, which is essential for language processing and modeling (Kudo, T., et al., 2018).
- Fairscale: A library that improves the scalability and performance of PyTorch in multi-GPU environments. Llama 2 leverages Fairscale to optimize hardware resource utilization, which is critical for efficient operation (Miao, X, et al., 2022).

This technology stack enables Llama 2 to achieve a high level of performance, efficiency, and functionality in addressing a wide range of tasks related to natural language processing and generation.

Training Hardware: The Llama 2 model performs pre-training on Meta's Research SuperCluster (RSC) as well as on internal production clusters. Both clusters are powered by NVIDIA A100, which delivers an unmatched level of acceleration across the board, powering artificial intelligence, data analytics and high-performance computing (HPC) in the most advanced data centers. The A100, powered by the NVIDIA Ampere architecture, stands as the heart of NVIDIA's data center platform. With up to 20 times the performance of the previous generation, the A100 has the versatility to be split into seven GPU instances, dynamically adapting to changing demands (Gálvez Vallejo, J. L., et al 2023). In this comparison, two fundamental differences between these two clusters stand out. First, is the type of available interconnection: the RSC makes use of NVIDIA Quantum InfiniBand technology, while the production cluster is equipped with a RoCE (RDMA over Converged Ethernet) solution based on general-purpose Ethernet switches. Both solutions enable 200 Gbps endpoint interconnection.

The second difference lies in the power consumption limit per GPU: While the RSC has a limit of 400W, our production cluster operates with a limit of 350W per GPU. This two-cluster configuration has enabled meaningful comparisons on the suitability of these various interconnect types for large-scale training. Relevantly, the RoCE approach, being a more affordable commercial interconnect solution, stands out (Touvron, H. et al., 2023).

Carbon Footprint

One of the outstanding differentiators between Meta and Llama 2 is its ability to influence carbon footprint indicators. Corporate environmental responsibility has led to a study of the energy consumption per hour generated by the GPU. During the pre-training phase, the following consumption and carbon footprint data were obtained (Touvron, H. et al., 2023). Table 4 shows the emissions during pre-training:

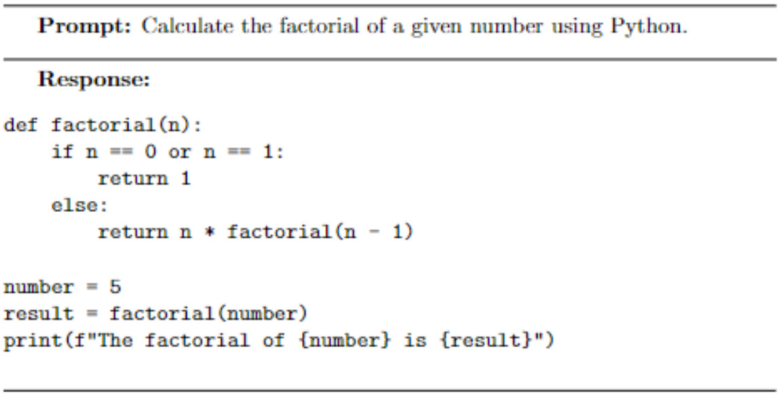
Table 4. Emissions during pre-training

Model	Time (GPU Hours)	Power Consumption (W)	Carbon Emitted (tCO2eq)
7B	184320	400	31.22
13B	368640	400	62.44
34B	1038336	350	153.90
70B	1720320	400	291.42
Total	3311616		539.00

Model Evaluation

The Llama 2 model performs well on the following topics:

- **Code generation**: a code test validation was carried out in Python using Llama 2.



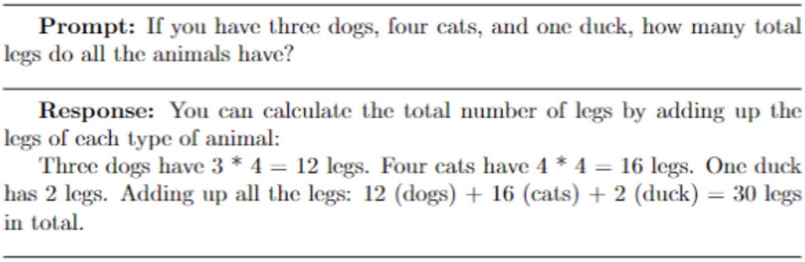
Prompt: Calculate the factorial of a given number using Python.

Response:

```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

number = 5
result = factorial(number)
print(f"The factorial of {number} is {result}")
```

Figure 8. Example of code generation.

- **Reasoning and common sense**: A test validation of reasoning exercises was carried out using Llama 2.



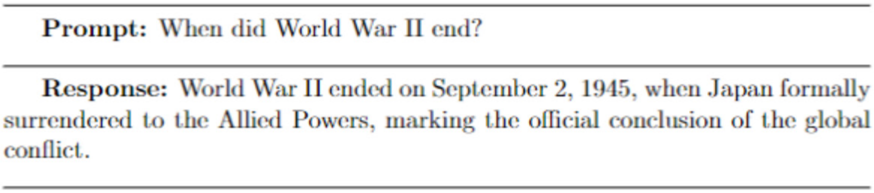
Prompt: If you have three dogs, four cats, and one duck, how many total legs do all the animals have?

Response: You can calculate the total number of legs by adding up the legs of each type of animal:

Three dogs have 3 * 4 = 12 legs. Four cats have 4 * 4 = 16 legs. One duck has 2 legs. Adding up all the legs: 12 (dogs) + 16 (cats) + 2 (duck) = 30 legs in total.

Figure 9. Example of reasoning.

- **Global knowledge**: a validation test on general culture topics was carried out on the global level using Llama 2.

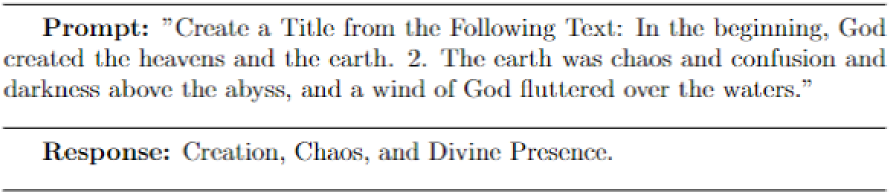


Prompt: When did World War II end?

Response: World War II ended on September 2, 1945, when Japan formally surrendered to the Allied Powers, marking the official conclusion of the global conflict.

Figure 10. World culture example.

- **Reading comprehension**: A validation test was carried out on reading comprehension topics using Llama 2.

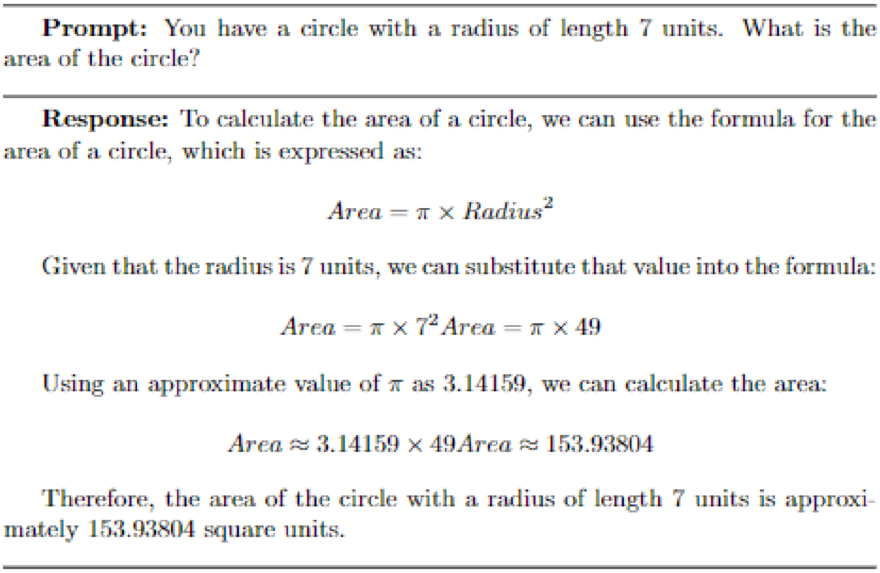


Prompt: "Create a Title from the Following Text: In the beginning, God created the heavens and the earth. 2. The earth was chaos and confusion and darkness above the abyss, and a wind of God fluttered over the waters."

Response: Creation, Chaos, and Divine Presence.

Figure 11. Reading comprehension example.

- **Mathematics**: a validation test was carried out using Llama 2 for calculation ability.



Prompt: You have a circle with a radius of length 7 units. What is the area of the circle?

Response: To calculate the area of a circle, we can use the formula for the area of a circle, which is expressed as:

$$Area = \pi \times Radius^2$$

Given that the radius is 7 units, we can substitute that value into the formula:

$$Area = \pi \times 7^2 Area = \pi \times 49$$

Using an approximate value of π as 3.14159, we can calculate the area:

$$Area \approx 3.14159 \times 49 Area \approx 153.93804$$

Therefore, the area of the circle with a radius of length 7 units is approximately 153.93804 square units.

Figure 12. Example of calculation ability.

6.3. Llama 2 Training Architecture and Process

The Llama 2 training process is a novel combination of architectures, such as a transformer-based pre-training and reinforcement learning with human feedback (RLFH) based tuning, where precision, accuracy, and safety of the Llama 2 LLM are paramount. The following is the methodology used by Meta for the construction of Llama 2 (Touvron, H et al., 2023) . Figure 13 shows the training process of the Llama 2 model.

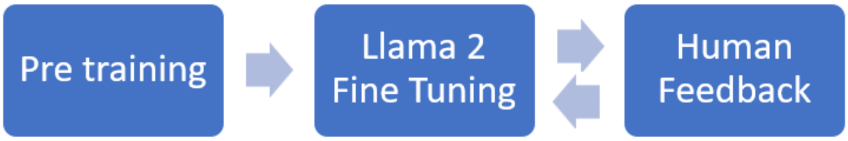


Figure 13. Training process.

Pretraining

Llama 2 performs pre-training from publicly available data sources, excluding any data originating from Meta's products or services. During the data selection process, sources such as web data and specialized databases were used. The model underwent extensive training using a dataset that comprises two trillion tokens and exhibits twice the context length of its predecessor, Llama 1 (Jiao, F., et al., 2023). This design choice strikes a balance between performance and computational cost, with a deliberate emphasis on sampling from the most factual sources to improve knowledge while mitigating potential misinformation and bias issues (Touvron, H et al., 2023).

The model has a standard transformer architecture (Vaswani et al., 2017), using prenormalization with RMSNorm (Sennrich, R et al., 2019) and the SwiGLU activation function (Shazeer, N. et al., 2020). In addition, it integrates rotational positional embeddings (RoPE) (Touvron, H. et al., 2023).

The key differences between Llama 1 and Llama 2 lie in the increased context length and the adoption of grouped query attention (GQA). These architectural modifications contribute to an increased ability to handle longer contextual information during language generation tasks (Touvron, H. et al., 2023) . Below, Table 5 shows characteristics of the model.

Table 5. Pre-training characteristics.

	Training Data	Parameters	Context Length	Tokens
Llama 1	Datasets (Touvron, H et al., 2023).	7B	2k	1.0T
		13B	2k	1.0T
		33B	2k	1.4T
		65B	2k	1.4T
Llama 2	Online data	7B	4k	2.0T
		13B	4k	2.0T
		34B	4k	2.0T
		70B	4k	2.0T

Llama 2 *Fine-Tuning*:

Llama 2 was initially trained using publicly available online data. An initial version of Llama-2-chat was then generated using supervised tuning. Subsequently, Llama-2-chat was iteratively refined by employing reinforced learning based on human feedback (RLHF), which includes techniques such as rejection sampling and proximal policy optimization (PPO) (Touvron, H. et al., 2023).

Although there are numerous sources providing such data, their limited diversity and quality have led to prioritizing the collection of high quality active and passive voice (OFV) examples, which has led to significant improvements.

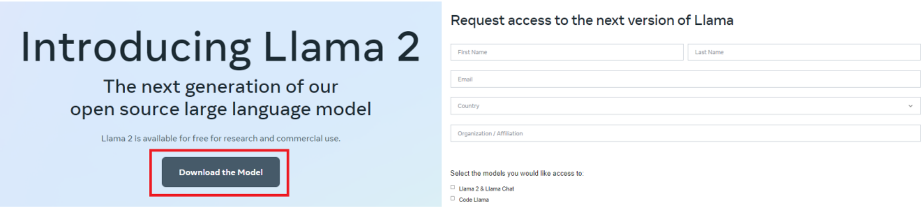
Reinforcement learning with human feedback (RLHF) represents a model training approach that aims to further align the behavior of a tuned language model with human preferences and instructions. To achieve this, human preference data are collected through a selection between two model-generated outcomes, which contributes to the training of a reward model that automates preferential choices (Maroto-Gómez, M. et al., 2023).

The essential role of the reward model is manifested in the context of reinforcement learning with human feedback (RLHF). In this process, the reward model evaluates a model-generated response and its corresponding cue, resulting in a numerical score that reflects quality in terms of utility and safety. By making use of these response scores as rewards, the RLHF process aims to optimize Llama 2 Chat to align its behavior with human preferences, improving both utility and safety (Touvron, H. et al., 2023).

6.4. How to Install It?

For a user to gain access to the pre-trained models, they are required to request permission through Meta's official website (https://ai.meta.com/llama/) and agree to the specified terms and conditions. Once their request is approved, they receive an email that includes a unique, personalized link. This link allows the user to download the models.

Figure 14 illustrates the home screen for registration on the official META Llama 2 website.



Introducing Llama 2

The next generation of our
open source large language model

Llama 2 is available for free for research and commercial use.

Request access to the next version of Llama

Select the models you would like access to:

□ Llama 2 & Llama Chat

□ Code Llama

Figure 14. Llama 2 installation process.

It is recommended to create a Python project, so that the user can use the URL provided on GitHub to clone the repository that hosts the models. This makes it easier to make the models available on their system for use in the project. All the above can be achieved by following these steps:

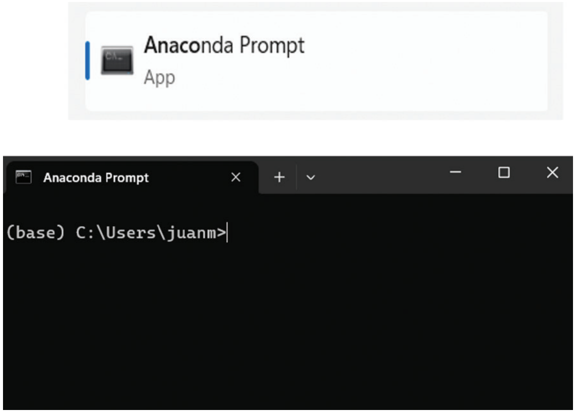
- Request access via the official Meta website and agree to the applicable terms and conditions.
- Once the request is approved, the user will receive an email containing a unique, customized link to download the models.
- In a new Python project, the GitHub URL can be used to clone the repository containing the models.

- Run the “download.sh” script, where the user is required to enter the unique, custom URL and select the models to download. The options available for download include 7B, 13B, 70B, chat 7B, chat 13B and chat 70B.

Note: It is recommended that the user has sufficient storage, processing power, a GPU, and enough RAM to be able to manage these models.

The Bioinformatics, Intelligent Information Systems and Educational Technology (BISITE) research group has managed to install the Llama 2 model locally. Here is a step-by-step guide on how to install the model and make customized use of it:

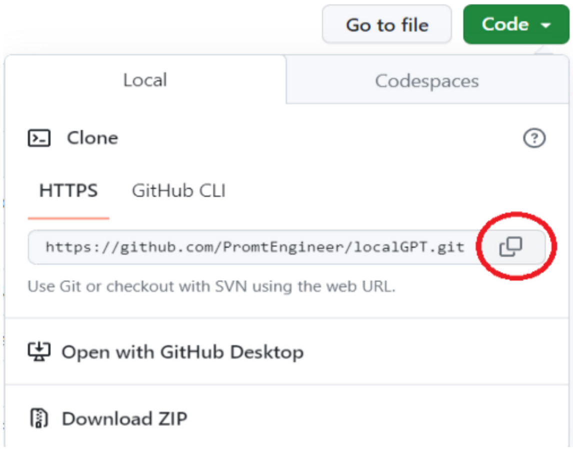
1. Install the free anaconda distribution and run the anaconda prompt, see Figure 15:



```
(base) C:\Users\juanm>
```

Figure 15. Run the anaconda prompt.

2. Access the repository https://github.com/PromtEngineer/localGPT, where credit is given to (PromptEngineer, Github) and copy the link: https://github.com/PromtEngineer/localGPT.git, see Figure 16.

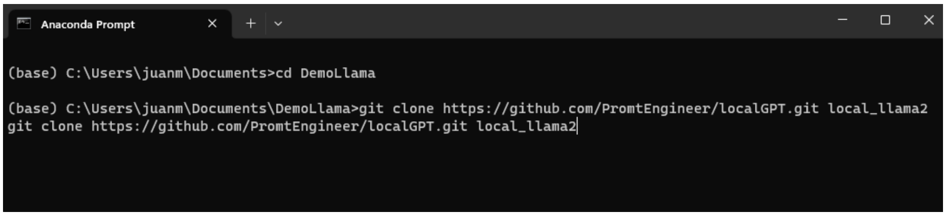


https://github.com/PromtEngineer/localGPT.git

Use Git or checkout with SVN using the web URL.

Figure 16. Copy the GitHub repository.

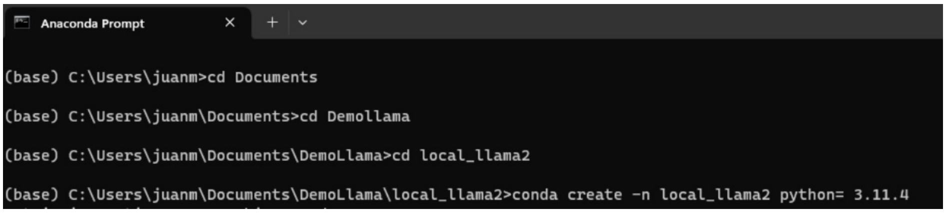
3. Create a folder at a location of your choice on your computer and make a GitHub clone, see Figure 17.



```
(base) C:\Users\juanm\Documents>cd DemoLlama

(base) C:\Users\juanm\Documents\DemoLlama>git clone https://github.com/PromtEngineer/localGPT.git local_llama2
git clone https://github.com/PromtEngineer/localGPT.git local_llama2
```

Figure 17. Clone GithHub locally.

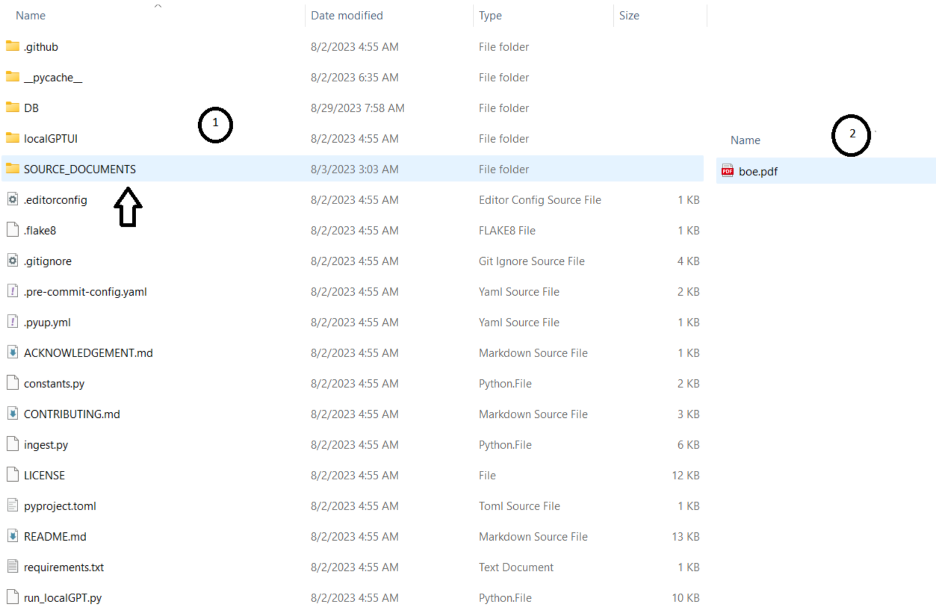
4. Perform the following command to create a Conda environment, see Figure 18:



```
(base) C:\Users\juanm>cd Documents

(base) C:\Users\juanm\Documents>cd Demollama

(base) C:\Users\juanm\Documents\DemoLlama>cd local_llama2

(base) C:\Users\juanm\Documents\DemoLlama\local_llama2>conda create -n local_llama2 python= 3.11.4
```

Figure 18. Create a Conda environment to install requirements.

5. Enter the file or document to be trained and all the above to be able to embed the local Llama 2 model. This file must be placed in the SOURCE_DOCUMENTS folder, see Figure 19.

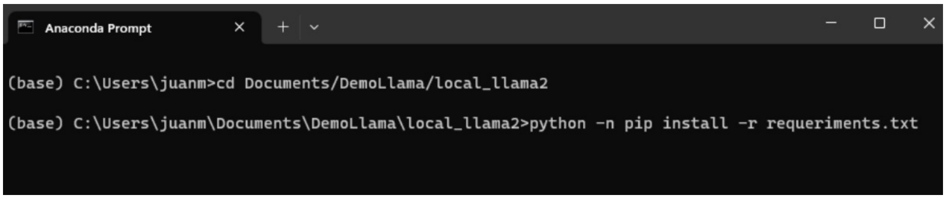


Name	Date modified	Type	Size
.github	8/2/2023 4:55 AM	File folder	
__pycache__	8/2/2023 6:35 AM	File folder	
DB	8/29/2023 7:58 AM	File folder	
localGPTUI	8/2/2023 4:55 AM	File folder	
SOURCE_DOCUMENTS	8/3/2023 3:03 AM	File folder	
.editorconfig	8/2/2023 4:55 AM	Editor Config Source File	1 KB
.flake8	8/2/2023 4:55 AM	FLAKE8 File	1 KB
.gitignore	8/2/2023 4:55 AM	Git Ignore Source File	4 KB
.pre-commit-config.yaml	8/2/2023 4:55 AM	Yaml Source File	2 KB
.pyup.yml	8/2/2023 4:55 AM	Yaml Source File	1 KB
ACKNOWLEDGEMENT.md	8/2/2023 4:55 AM	Markdown Source File	1 KB
constants.py	8/2/2023 4:55 AM	Python.File	2 KB
CONTRIBUTING.md	8/2/2023 4:55 AM	Markdown Source File	3 KB
ingest.py	8/2/2023 4:55 AM	Python.File	6 KB
LICENSE	8/2/2023 4:55 AM	File	12 KB
pyproject.toml	8/2/2023 4:55 AM	Toml Source File	1 KB
README.md	8/2/2023 4:55 AM	Markdown Source File	13 KB
requirements.txt	8/2/2023 4:55 AM	Text Document	1 KB
run_localGPT.py	8/2/2023 4:55 AM	Python.File	10 KB

Name
boe.pdf

Figure 19. Insert the file to be trained.

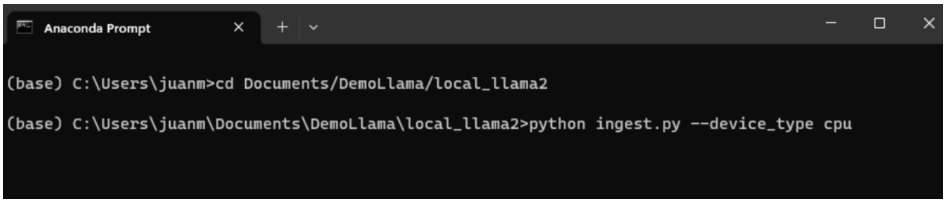
6. Install the packages according to the requirements file: Some of the packages are: pytorch, langchain, chromadb, sentence-transformers, huggingface hub (See the requirements.txt file), see Figure 20.



```
(base) C:\Users\juanm>cd Documents/DemoLlama/local_llama2

(base) C:\Users\juanm\Documents\DemoLlama\local_llama2>python -n pip install -r requeriments.txt
```

Figure 20. Installing Llama 2 packages and requirements.

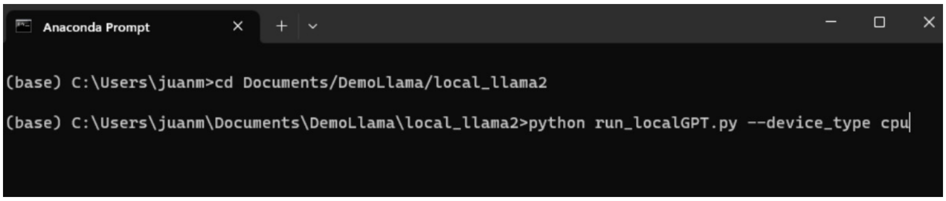
7. The data from the file to be trained must be ingested into the model and the following instruction must be followed, see Figure 21:



```
(base) C:\Users\juanm>cd Documents/DemoLlama/local_llama2

(base) C:\Users\juanm\Documents\DemoLlama\local_llama2>python ingest.py --device_type cpu
```

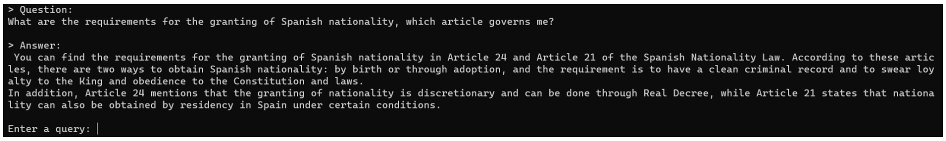
Figure 21. Ingesting data into the Llama 2 model.

8. The following command is used to execute the model and make a request or query, see Figures 22, 23:



```
(base) C:\Users\juanm>cd Documents/DemoLlama/local_llama2

(base) C:\Users\juanm\Documents\DemoLlama\local_llama2>python run_localGPT.py --device_type cpu
```

Figure 22. Making a request in the Llama 2 model.



```
> Question:
What are the requirements for the granting of Spanish nationality, which article governs me?

> Answer:
 You can find the requirements for the granting of Spanish nationality in Article 24 and Article 21 of the Spanish Nationality Law. According to these artic
les, there are two ways to obtain Spanish nationality: by birth or through adoption, and the requirement is to have a clean criminal record and to swear loy
alty to the King and obedience to the Constitution and laws.
In addition, Article 24 mentions that the granting of nationality is discretionary and can be done through Real Decree, while Article 21 states that nationa
lity can also be obtained by residency in Spain under certain conditions.

Enter a query: 
```

Figure 23. Validation test.

6.5. Use Cases

Multiple applications are beginning to take shape with the decentralization of generative AI and research focused on the protection and detection of people with abusive and sexual language on social networks, where the population to be protected are children and adolescents (Nguyen, T. T, et al., 2023). In the health sector, the advances that are being presented from the point of view of medical assistants through generative AI are evident. This is the case of Li, Y., et al., 2023, who have implemented a specialized medical chat based on the diagnoses and experience of an organization's physicians. ChatDoctor uses Llama 2 as a framework and is fed by the needs of patients, medical opinions that are issued daily, thus improving the accuracy of the model, and generating reliability in medical responses. In the financial context, an LLM based on Llama 2 is refined for the analysis of financial news to extract the most important indicators from the text and perform the respective analytics (Pavlyshenko, 2023).

7. Conclusions

Generative models using diffusion techniques have opened possibilities in multiple domains, especially in high-fidelity text, audio, and video production. Their probabilistic frameworks can capture complex and nuanced relationships within data, generating highly realistic results (Rombach et al., 2022; Huang et al., 2023). For textual data, this encompasses everything from semantic coherence to syntactic sophistication. In audio, it leads to clearer speech synthesis and speech-to-text conversion applications (Williams, 2023). In video, the algorithms generate high-definition, temporally coherent video clips (Tate, 2023).

The series of large language models (LLMs) developed by OpenAI have revolutionized the field of natural language processing. Models such as GPT-4 and ChatGPT have demonstrated an impressive ability to generate coherent text, understand context, and adapt to diverse tasks. However, as these models advance, so do the ethical and technical challenges associated with their use.

It is critical to continue to research and develop effective strategies to mitigate bias, protect user privacy and ensure data security. As these models become more integrated into everyday life, clear regulations and guidelines for responsible use are needed. Venturing into the field of generative AI reveals significant potential in the application of these models in various domains, from programming to academic publications. However, it is essential to be critical and aware of their limitations. Ultimately, the goal of generative AI research should be to create tools that are useful, safe, and beneficial to humanity. Despite significant progress with models such as GPT-4 and ChatGPT, much work remains to be done to achieve this goal.

Llama 2, developed by Meta AI, is an open-source large language model (LLM) that facilitates interaction with documents on a local device. Its strength lies in its ability to operate locally, ensuring user privacy. Through its sophisticated training process, Llama 2 has demonstrated high performance in various benchmark tests, excelling in areas such as reasoning, coding, and reading comprehension. Despite requiring certain system requirements and technical expertise for installation, Llama 2 offers great versatility by allowing it to work with different Hugging Face LLM models. With applications ranging from child protection on social networks to medical assistants and financial analysis, Llama 2 opens new opportunities for natural language-based interaction and represents a significant breakthrough in the LLM field.

8. References

Abdullah, M., Madain, A., & Jararweh, Y., 2022, November. ChatGPT: Fundamentals, applications and social impacts. In *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (pp. 1-8). IEEE. https://doi.org/10.1109/SNAMS58071.2022.10062688

Abraham, A., Corchado, E., & Corchado, J. M., 2009. Hybrid learning machines. *Neurocomputing: An International Journal, 72*(13-15), 2729-2730. https://doi.org/10.1016/j.neucom.2009.02.017

Adams, L. C., Busch, F., Truhn, D., Makowski, M. R., Aerts, H. J., & Bressem, K. K., 2023. What Does DALL-E 2 Know About Radiology? *Journal of Medical Internet Research*, *25*, e43110. https://doi.org/10.1109/SNAMS58071.2022.10062688

Alizadehsani, Z., Ghaemi, H., Shahraki, A., Gonzalez-Briones, A., & Corchado, J. M., 2023. DCServCG: A data-centric service code generation using deep learning. *Engineering Applications of Artificial Intelligence*, *123*, 106304. https://doi.org/10.1016/j.engappai.2023.106304

Ba, J. L., Kiros, J. R., & Hinton, G. E., 2016. Layer normalization. *arXiv preprint arXiv*:1607.06450.

Bahdanau, D., Cho, K., & Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv*:1409.0473.

Baidoo-Anu, D., & Owusu Ansah, L., 2023, January 25. Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. Available at *SSRN*. https://doi.org/10.2139/ssrn.4337484

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of FAccT*. https://doi.org/10.1145/3442188.3445922

Bengio, Y., 2009. Learning deep architectures for AI. *Foundations and trends® in Machine Learning, 2*(1), 1-127. https://doi.org/10.1561/2200000006

Brown, T. B., et al., 2020. Language Models are Few-Shot Learners. *arXiv preprint arXiv*:2005.14165.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., ... & Zhang, Y., 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv*:2303.12712.

Caliskan, A., Bryson, J. J., & Narayanan, A., 2017. Semantics derived automatically from language corpora contain human-like biases. *Science, 356*(6334), 183-186. https://doi.org/10.1126/science.aal4230

Castellanos-Gómez, A., 2023. Good Practices for Scientific Article Writing with ChatGPT and Other Artificial Intelligence Language Models. *Nanomanufacturing, 3*(2), 135-138 https://doi.org/10.3390/nanomanufacturing3020009

Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M., 2019. *Social computing in currency exchange. Knowledge and Information Systems, 61*, 733-753. https://doi.org/10.1007/s10115-018-1289-4

Chan, W. H., Mohamad, M. S., Deris, S., Zaki, N., Kasim, S., Omatu, S., Corchado J. M. & Al Ashwal, H., 2016. *Identification of informative genes and pathways using an improved penalized support vector machine with a weighting scheme. Computers in biology and medicine, 77*, 102-115. https://doi.org/10.1016/j.compbiomed.2016.08.004

Chang, Y., Wang, X., Wang, J., Wu, Y., Zhu, K., Chen, H., ... & Xie, X., 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:*2307.03109.

Chaplin, J. R., Heller, V., Farley, F. J. M., Hearn, G. E., & Rainey, R. C. T., 2012. Laboratory testing the Anaconda. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *370*(1959), 403-424. https://doi.org/10.1098/rsta.2011.0256

Chen, D., et al., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv*:1810.04805.

Corchado J. M., 2023. *El Despertar de la Inteligencia Artificial Global.* Real Academia de Medicina - Salamanca.

Corchado, J. M., & Aiken, J., 2002. Hybrid artificial intelligence methods in oceanographic forecast models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *32*(4), 307-313. https://doi.org/10.1109/TSMCC.2002.806072

Corchado, J. M., Chamoso, P., Hernández, G., Gutierrez, A. S. R., Camacho, A. R., González-Briones, A., ... & Omatu, S., 2021. Deepint. net: A rapid deployment platform for smart territories. *Sensors*, *21*(1), 236. https://doi.org/10.3390/s21010236

Corchado, J. M., Díaz, F., Borrajo, L., & Fernández, F., 2000. *Redes neuronales artificiales. Un enfoque práctico*. Servicio de Publicacións da Universidade de Vigo.

Denny, P., Kumar, V., & Giacaman, N., 2022. *Conversing with Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language*. https://doi.org/10.1145/3545945.3569823

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:*1810.04805.

Díaz, F., Fernández–Riverola, F., & Corchado, J. M., 2006. gene-CBR: A Case-Based Reasonig Tool for Cancer Diagnosis Using Microarray Data Sets. *Computational Intelligence, 22*(3-4), 254-268. https://doi.org/10.1111/j.1467-8640.2006.00287.x

Doshi-Velez, F., Kortz, M., Budish, R., Bavitz, C., Gershman, S., O'Brien, D., ... & Waldo, J., 2017. Accountability of AI Under the Law: The Role of Explanation. *Berkman Klein Center Working Group on Explanation and the Law, 2*. https://doi.org/10.2139/ssrn.3064761

Eloundou, T., Manning, S., Mishkin, P., & Rock, D., 2023. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv*:2303.10130.

Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., & Prather, J., 2022. *The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming*. https://doi.org/10.1145/3511861.3511863

Gala, Y., Fernández, Á., Díaz, J., & Dorronsoro, J. R., 2016. Hybrid machine learning forecasting of solar radiation values. *Neurocomputing*, 176, 48-59.

Galvez Vallejo, J. L., Snowdon, C., Stocks, R., Kazemian, F., Yan Yu, F. C., Seidl, C., ... & Barca, G. M., 2023. Toward an extreme-scale electronic structure system. *The Journal of Chemical Physics*, *159*(4). https://doi.org/10.1063/5.0156399

Gao, C. A., Howard, F. M., Markov, N. S., Dyer, E., Ramesh, S., Luo, Y., Pearson, A. T., 2023. Comparing scientific abstracts generated by ChatGPT to real abstracts with detectors and blinded human reviewers. *npj Digit. Med. 6*, 75. https://doi.org/10.1038/s41746-023-00819-6

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc.

González-Briones, A., De La Prieta, F., Mohamad, M. S., Omatu, S., & Corchado, J. M., 2018. Multi-agent systems applications in energy optimization problems: *A state-of-the-art review. Energies, 11*(8), 1928. https://doi.org/10.3390/en11081928

Goodfellow, I., Bengio, Y., & Courville, A., 2016. *Deep learning*. MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y., 2014. Generative adversarial nets. *Advances in neural information* processing systems, 27.

Guan, W., Smetannikov, I., & Tianxing, M., 2020, October. Survey on automatic text summarization and transformer models applicability. In *Proceedings of the 2020 1st International Conference on Control, Robotics and Intelligent System* (pp. 176-184). https://doi.org/10.1145/3437802.3437832

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S., 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning* (Vol. 80, pp. 1861-1870).

Hao, K., 2020. We read the paper that forced Timnit Gebru out of Google. Here's what it says. MIT Technology Review.

Hassani, H., Silva, E. S., 2023. The Role of ChatGPT in Data Science: How AI-Assisted Conversational Interfaces are Revolutionizing the Field. *Big Data Cogn. Comput., 7*, 62. https://doi.org/10.3390/bdcc7020062

He, K., Zhang, X., Ren, S., & Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). https://doi.org/10.1109/CVPR.2016.90

Hendler, J., & Mulvehill, A. M., 2016. *Social machines: the coming collision of artificial intelligence, social networking, and humanity*. Apress. https://doi.org/10.1007/978-1-4842-1156-4

Hernández, G., Rodríguez, S., González, A., Corchado, J. M., & Prieto, J., 2021. Video analysis system using deep learning algorithms. In *Ambient Intelligence–Software and Applications: 11th International Symposium on Ambient Intelligence* (pp. 186-199). Springer International Publishing. https://doi.org/10.1007/978-3-030-58356-9_19

Hernandez-Nieves, E., Hernández, G., Gil-Gonzalez, A. B., Rodríguez-González, S., & Corchado, J. M., 2021. CEBRA: A CasE-Based Reasoning Application to recommend banking products. *Engineering Applications of Artificial Intelligence*, *104*, 104327. https://doi.org/10.1016/j.engappai.2021.104327

Ho, J., Song, D., & Elizalde, B., 2020. Denoising Diffusion Probabilistic Models. *ArXiv Preprint ArXiv.*

Hochreiter, S., & Schmidhuber, J., 1997. Long short-term memory. *Neural* computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Homolak, J. 2023. *Opportunities and risks of ChatGPT in medicine, science, and academic publishing: a modern Promethean dilemma*. https://doi.org/10.3325/cmj.2023.64.1

Hovy, D., & Spruit, S. L., 2016. The Social Impact of Natural Language Processing. *ACL 2016.* https://doi.org/10.18653/v1/P16-2096

Hu, L., 2022. *Generative AI and Future*. Retrieved January 23, 2023, from URL https://pub.towardsai.net/generative-ai-and-future-c3b1695876f2

Huang, R., Zhou, Z., Zhang, Y., & Zhao, Z., 2023. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. *ArXiv Preprint ArXiv*:2301.12661.

Itoh, S., & Okada, K., 2023. The Power of Large Language Models: A ChatGPT-driven Textual Analysis of Fundamental Data. Available at *SSRN* 4535647. https://doi.org/10.2139/ssrn.4535647

Janbi, N., Mehmood, R., Katib, I., Albeshri, A., Corchado, J. M., & Yigitcanlar, T., 2022. Imtidad: A Reference Architecture and a Case Study on Developing Distributed AI Services for Skin Disease Diagnosis over Cloud, Fog and Edge. *Sensors, 22*(5), 1854. https://doi.org/10.3390/s22051854

Jiang, H., Sun, D., Jampani, V., Yang, M. H., Learned-Miller, E., & Kautz, J., 2018. Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation. In *CVPR* (pp. 9000-9008). https://doi.org/10.1109/CVPR.2018.00938

Jiao, F., Ding, B., Luo, T., & Mo, Z., 2023. Panda LLM: Training Data and Evaluation for Open-Sourced Chinese Instruction-Following Large Language Models. *arXiv preprint arXiv:*2305.03025.

Jovanovic, M., & Campbell, M., 2022. Generative artificial intelligence: Trends and prospects. *Computer*, *55*(10), 107-112. https://doi.org/10.1109/MC.2022.3192720

Jurafsky, D., & Martin, J. H., 2019. *Speech and Language Processing*. Pearson.

Kingma, D. P., & Welling, M., 2013. Auto-Encoding Variational Bayes. *arXiv preprint arXiv*:1312.6114.

Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-González, A. B., & Corchado, J. M., 2022. Deepsign: Sign language detection and recognition using deep learning. *Electronics*, *11*(11), 1780. https://doi.org/10.3390/electronics11111780

Krizhevsky, A., Sutskever, I., & Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25*.

Kudo, T., & Richardson, J., 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:*1808.06226. https://doi.org/10.18653/v1/D18-2012

Li, Y., Li, Z., Zhang, K., Dan, R., Jiang, S., & Zhang, Y., 2023. ChatDoctor: A Medical Chat Model Fine-Tuned on a Large Language Model Meta-AI (LLaMA) Using Medical Domain Knowledge. *Cureus*, *15*(6). https://doi.org/10.7759/cureus.40895

Liu, H., Sferrazza, C., & Abbeel, P., 2023. Languages are rewards: Hindsight finetuning using human feedback. arXiv preprint arXiv:2302.02676.

Lund, B. D., Wang, T., Mannuru, N. R., Nie, B., Shimray, S., & Wang, Z., 2023. *ChatGPT and a new academic reality: Artificial Intelligence-written research papers and the ethics of the large language models in scholarly publishing.* https://doi.org/10.1002/asi.24750

M. Salvagno, Taccone, F., Gerli, A., 2023. *Can artificial intelligence help for scientific writing?* https://doi.org/10.1186/s13054-023-04380-2

Madotto, A., Lin, Z., Winata, G. I., & Fung, P., 2021. Few-shot bot: Prompt-based learning for dialogue systems. *arXiv preprint arXiv:*2110.08118.

Manning, C. D., et al., 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Maroto-Gómez, M., Castro-González, Á., Castillo, J. C., Malfaz, M., & Salichs, M. Á., 2023. An adaptive decision-making system supported on user preference predictions for human–robot interactive communication. *User Modeling and User-Adapted Interaction*, *33*(2), 359-403. https://doi.org/10.1007/s11257-022-09321-2

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E., 2006. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine, 27*(4), 12-12.

McCulloch, W. S., & Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*, 115-133. https://doi.org/10.1007/BF02478259

McGregor, S., Memon, N., & Levy, K., 2020. Cybersecurity and Human Rights. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 425-435).

Miao, X., Wang, Y., Jiang, Y., Shi, C., Nie, X., Zhang, H., & Cui, B., 2022. Galvatron: Efficient transformer training over multiple gpus using automatic parallelism. *arXiv preprint arXiv:*2211.13878. https://doi.org/10.14778/3570690.3570697

Moor, J., 2006. The Dartmouth College artificial intelligence conference: The next fifty years. *Ai Magazine*, *27*(4), 87-87.

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W., 2011. Natural language processing: an introduction. *Journal of the American Medical Informatics Association, 18*(5), 544-551. https://doi.org/10.1136/amiajnl-2011-000464

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., ... & Schulman, J., 2021. Webgpt: Browser-assisted question-answering with human feedback. arXiv preprint arXiv:2112.09332.

Nguyen, T. M., & Wu, Q. J. (2013). A fuzzy logic model based Markov random field for medical image segmentation. *Evolving systems*, 4, 171-181.

Nguyen, T. T., Wilson, C., & Dalins, J., 2023. Fine-Tuning Llama 2 Large Language Models for Detecting Online Sexual Predatory Chats and Abusive Texts. *arXiv preprint arXiv:*2308.14683. https://doi.org/10.1109/TITS.2023.3297252

Nichol, A., et al., 2021. GLIDE: A Generative Language Model for Text-Driven Applications. *Journal of Artificial Intelligence Research, 49*(2), 315-334.

OpenAI, 2021. ChatGPT API Documentation. OpenAI. Recuperado el 22 de agosto de 2023, de https://platform.openai.com/docs/guides/chat

OpenAI, 2023, July 20. Custom instructions for ChatGPT. OpenAI Blog. https://openai.com/blog/custom-instructions-for-chatgpt

OpenAI, 2023. GPT-4 Technical Report. ArXiv, abs/2303.08774.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R., 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems, 35*, 27730-27744.

Parikh, V., Shah, J., Bhatt, C., Corchado, J. M., & Le, D. N., 2022, July. Deep Learning Based Automated Chest X-ray Abnormalities Detection. In *International Symposium on Ambient Intelligence* (pp. 1-12). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-22356-3_1

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

Pavlyshenko, B. M. (2023). Financial News Analytics Using Fine-Tuned Llama 2 GPT Model. arXiv preprint arXiv:2308.13032.

Pérez-Pons, M. E., Alonso, R. S., García, O., Marreiros, G., & Corchado, J. M., 2021. *Deep q-learning and preference based multi-agent system for sustainable agricultural market. Sensors, 21*(16), 5276. https://doi.org/10.3390/s21165276

Pérez-Pons, M. E., Parra-Dominguez, J., Hernández, G., Bichindaritz, I., & Corchado, J. M., 2023. OCI-CBR: A hybrid model for decision support in preference-aware investment scenarios. *Expert Systems with Applications, 211*, 118568. https://doi.org/10.1016/j.eswa.2022.118568

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I., 2018. *Improving language understanding by generative pre-training.*

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I., 2021. Language Models are Few-Shot Learners. *OpenAI Blog, 9*(2), 650-700.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I., 2019. *Language Models are Unsupervised Multitask Learners. OpenAI Blog, 1*(8), 9.

Reiter, E., & Dale, R., 2020. *Building Natural Language Generation Systems*. Cambridge University Press.

Rombach, M., et al., 2022. Generative Latent Diffusion Models for High Definition Video Generation. *IEEE Transactions on Multimedia, 24*(1), 123-137.

Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386. https://doi.org/10.1037/h0042519

Rumelhart, D. E., Hinton, G. E., & Williams, R. J., 1986. Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536. https://doi.org/10.1038/323533a0

Russell, S. J., & Norvig, P., 2014. *Artificial intelligence: a modern approach*. Harlow.

Sarsa, S., Denny, P., Hellas, A., & Leinonen, J., 2022. *Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models.* https://doi.org/10.1145/3501385.3543957

Sennrich, R., & Zhang, B., 2019. Revisiting low-resource neural machine translation: A case study. *arXiv preprint arXiv*:1905.11901. https://doi.org/10.18653/v1/P19-1021

Serban, I. V., et al., 2017. A Deep Reinforcement Learning Chatbot. *arXiv preprint arXiv*:1709.02349.

Shazeer, N., 2020. Glu variants improve transformer. *arXiv preprint arXiv*:2002.05202.

Sherstinsky, A., 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306. https://doi.org/10.1016/j.physd.2019.132306

Siddique, S.; Chow, J. C. L., 2021. Machine Learning in Healthcare Communication. *Encyclopedia, 1*, 220-239. https://doi.org/10.3390/encyclopedia1010021

Simonyan, K., & Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv*:1409.1556.

Su, Z., Chow, J. K., Tan, P. S., Wu, J., Ho, Y. K., & Wang, Y. H., 2021. Deep convolutional neural network–based pixel-wise landslide inventory mapping. *Landslides, 18*, 1421-1443. https://doi.org/10.1007/s10346-020-01557-6

Tang, D., Rong, W., Qin, S., Yang, J., & Xiong, Z., 2020. A n-gated recurrent unit with review for answer selection. *Neurocomputing, 371*, 158-165. https://doi.org/10.1016/j.neucom.2019.09.007

Tate, E., 2023. High definition video generation: A comprehensive review. *Journal of Multimedia Processing.*

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T., 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:*2307.09288.

Tulyakov, S., Liu, M. Y., Yang, X., & Kautz, J., 2018. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1526-1535). https://doi.org/10.1109/CVPR.2018.00165

Turing A. M., 1950. Computing machinery and intelligence. In: *Mind* 59.236, pp. 433-460. https://doi.org/10.1093/mind/LIX.236.433

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems, 30*.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., & Vladymyrov, M., 2023, July. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning* (pp. 35151-35174). PMLR.

Vondrick, C., Pirsiavash, H., & Torralba, A., 2016. Generating videos with scene dynamics. In *Advances in neural information processing systems* (pp. 613-621).

Wallace, E., et al., 2019. Universal Adversarial Triggers for Attacking and Analyzing NLP. *EMNLP 2019.* https://doi.org/10.18653/v1/D19-1221

Wei, R., & Mahmood, A., 2020. Recent advances in variational autoencoders with representation learning for biomedical informatics: A survey. *Ieee Access, 9*, 4939-4956. https://doi.org/10.1109/ACCESS.2020.3048309

Williams, J., 2023. Efficacy of AI-generated text detectors. Journal of Educational Technology.

Xie, J., Xu, L., & Chen, E., 2018. Image Denoising and Inpainting with Deep Neural Networks. In *Advances in Neural Information Processing Systems* (pp. 341-349).

Xu, L., Sanders, L., Li, K., Chow, J. C. L., 2021. Chatbot for Health Care and Oncology Applications Using Artificial Intelligence and Machine Learning: Systematic Review. https://doi.org/10.2196/27850

Yalalov, D. 2023. La evolución de los chatbots: De la era T9 y GPT-1 a ChatGPT. Mpost. URL: https://mpost.io/es/la-evoluci%C3%B3n-de-los-chatbots-de-la-era-t9-y-gpt-1-a-chatgpt/

Zadeh, L. A., 2008. Is there a need for fuzzy logic? *Information sciences, 178*(13), 2751-2779. https://doi.org/10.1016/j.ins.2008.02.012

Zaremba, W., & Brockman, G., 2021, August 10. OpenAI Codex. OpenAI. https://openai.com/blog/openai-codex/

Zhao, J., Wang, T., Yatskar, M., Cotterell, R., Ordonez, V., & Chang, K. W., 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 2 (Short Papers) (pp. 8-14).

Zhao, Z., et al. 2023. Make-An-Audio: Text-To-Audio Generation with Prompt-Enhanced Diffusion Models. *Proceedings of the 40th International Conference on Machine Learning.*

Zoran, D., Kabra, R., Lerchner, A., & Rezende, D. J., 2021. Parts: Unsupervised segmentation with slots, attention and independence maximization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 10439-10447). https://doi.org/10.1109/ICCV48922.2021.01027