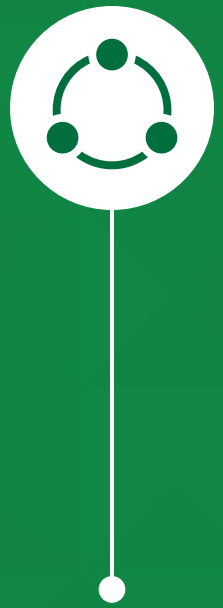
08/01/2023



Generative AI

Karthik Uppuluri

Principal Data Scientist

AI Center of Excellence, Fidelity Investments

For Wifi use Greenline

Introduction

Artificial Intelligence, Machine Learning and Deep Learning

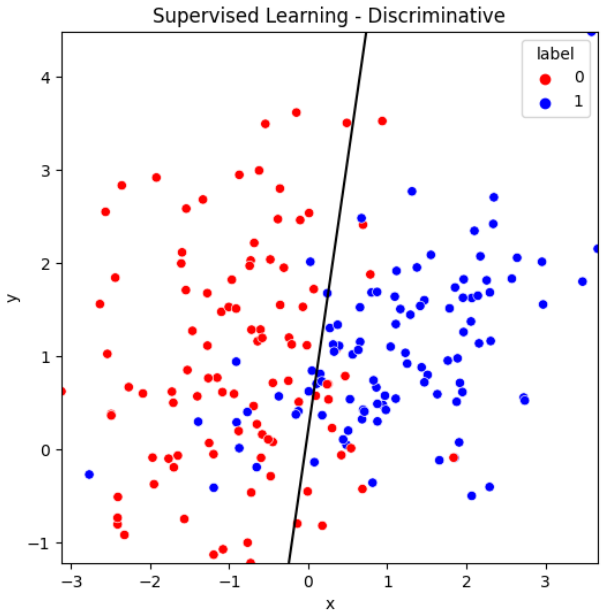
- The term **AI** in the broadest sense refers to simulation of human intelligence processes by computer systems
- **Machine Learning** is a subset of AI focusses on designing specific systems which can learn and make decisions/predictions based on data.
- **Deep Learning** is a subset of Machine Learning that uses a specific set of algorithms known as Neural-Networks often with many layers.

Introduction

Types of Machine Learning Models

❑ Supervised Learning

Supervised Learning is a type of Machine Learning model trained on labeled data



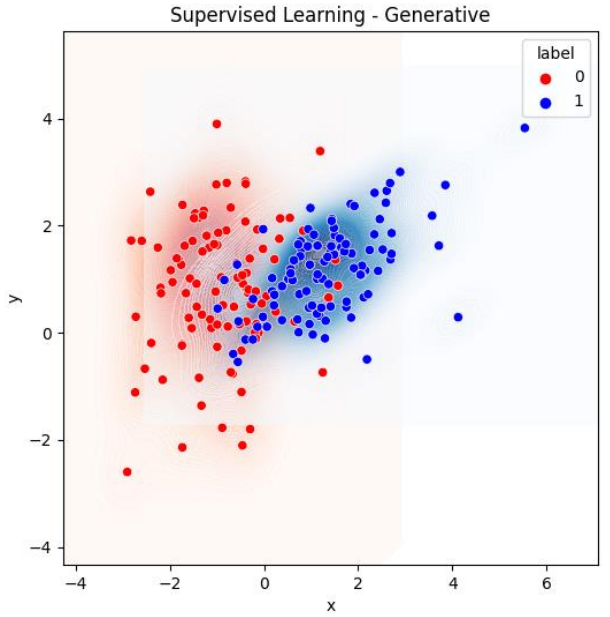
Email Spam Classification Model

***Data**: Examples of emails either tagged as Spam or not Spam*

Training:

Discriminative – Learns the boundary that separates “spam” vs “not spam”

Generative – Learns the distribution of “spam” and “not spam” emails to understand how each class generates content



Inference

Discriminative – Determine on which side of the boundary a new email falls

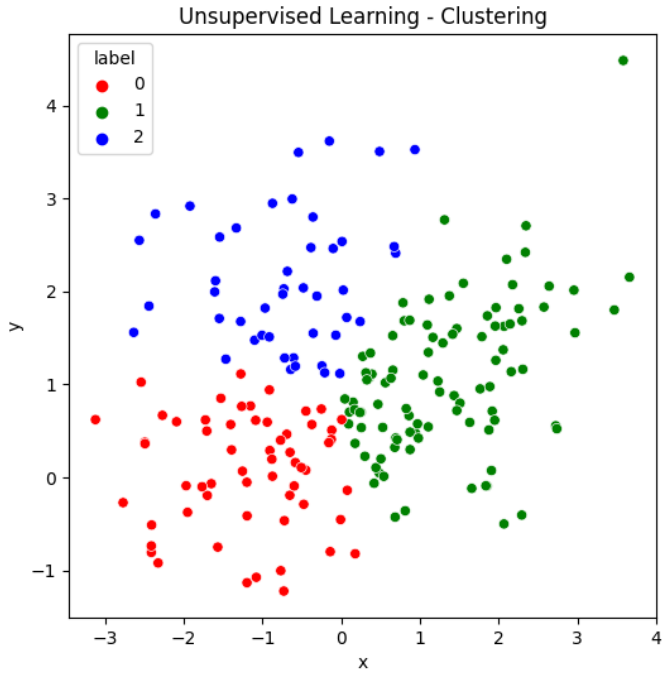
Generative – Based on learned distributions compute the likelihood of the new email being “spam” vs “not spam”

Introduction

Types of Machine Learning Models

❑ Unsupervised Learning

Unsupervised Learning is a type of Machine Learning model that identifies patterns and structures within un-labelled data



Email Topic Modeling

***Data**: A large collection of emails you may want to organize by subject matter*

Training:

Learn the distribution that generates the structure within the data

Inference

- Assign new email to the cluster where they have the highest probability of belonging

Introduction

Types of Machine Learning Models

❑ Reinforcement Learning

Interaction: Agent interacts with the environment by choosing actions from its current policy

A self-driving car decides to take a left or a right based on its current strategy and current state of the road

Reward/Penalty: After each action, agent receives a reward/penalty which reflects the success of the action

If the car safely navigates traffic or obeys rules, it's a success

Policy Update: Agent updates the policy based on feedback received aiming to maximize the total reward over time.

Based on the reward/penalty received car adjusts its driving policy, actions with positive rewards will be repeated and negative rewards will be avoided

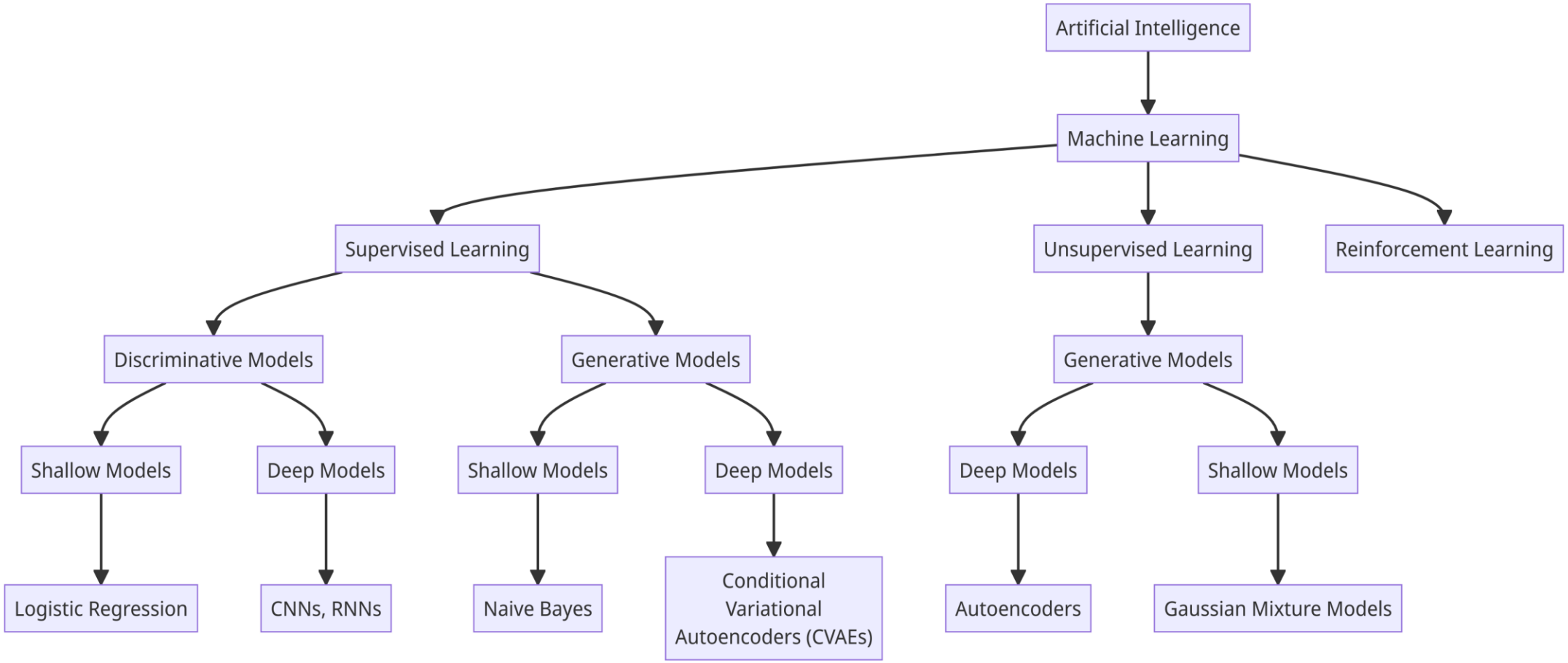
❑ Shallow and Deep Models

Models with limited layers and capable of capturing only linear and simple nonlinear relationships are called shallow models

Models with many layers and capable of capturing complex hierarchical patterns are called deep models

Introduction

Summary



Images Generated for this presentation

For Wifi use Greenline

Generative AI

GPT, GAN and Diffusion Models

Applications of Generative AI

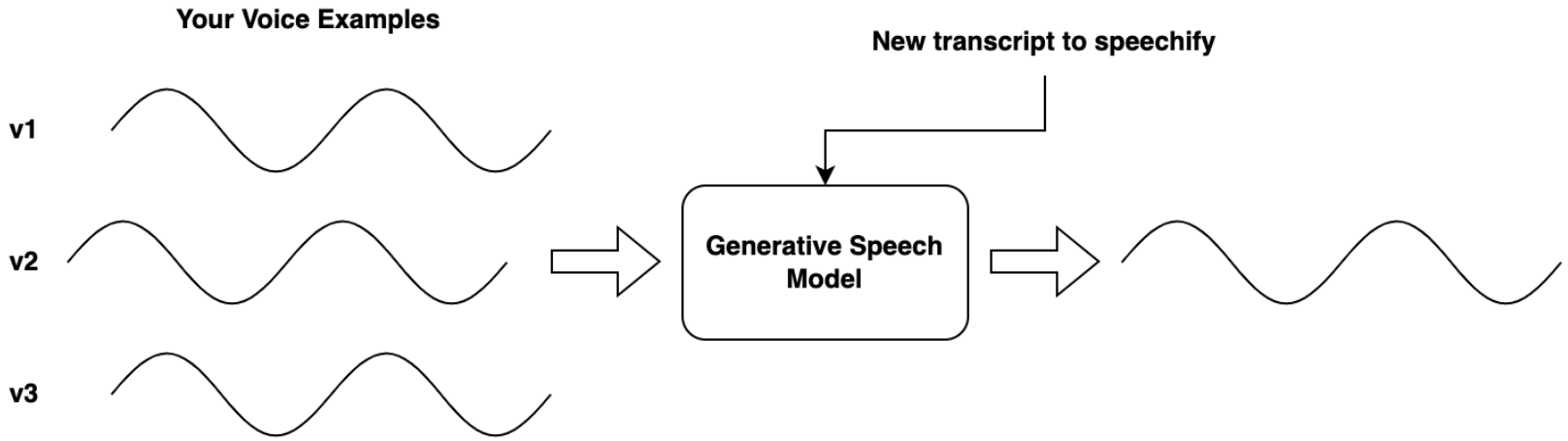
Emerging Trends, Limitations, Potential Ahead

Generative AI

Definition

Generative AI refers to a set of artificial intelligence methodologies that can produce novel content that resembles the training data they were exposed to.

The content could be anything spanning from synthesizing text, generating code, realistic images, music and more



Generative AI

Generative Pre-Trained Transformer (**GPT**) - Motivation

Issues with CNNs, RNNs, LSTMs

- Convolutional Neural Networks (**CNNs**) are good at **local feature extraction** and struggle to understand long-range dependencies in data

- CNNs **do not have a mechanism to understand the order of elements** making it harder for problems involving text and time-series

- **RNNs** especially **LSTMs** can handle long range dependencies due to their ability to process data sequentially. But as the sequences get longer, they struggle from **vanishing gradient problems**

- CNNs, RNNs, LSTMs are suitable for specific data types and are **not efficient at handling multi-modal inputs**

What if you can completely avoid recurrent connections, thereby avoiding vanishing gradient issues?

Generative AI

Generative Pre-Trained Transformer (**GPT**) - Motivation

- A new architecture called **Transformers** is proposed by scientists from Google which avoids the recurrent connections altogether by relying on an operation known as attention
- This architecture also takes care of sequential nature of inputs by using **positional embeddings**

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Generative AI

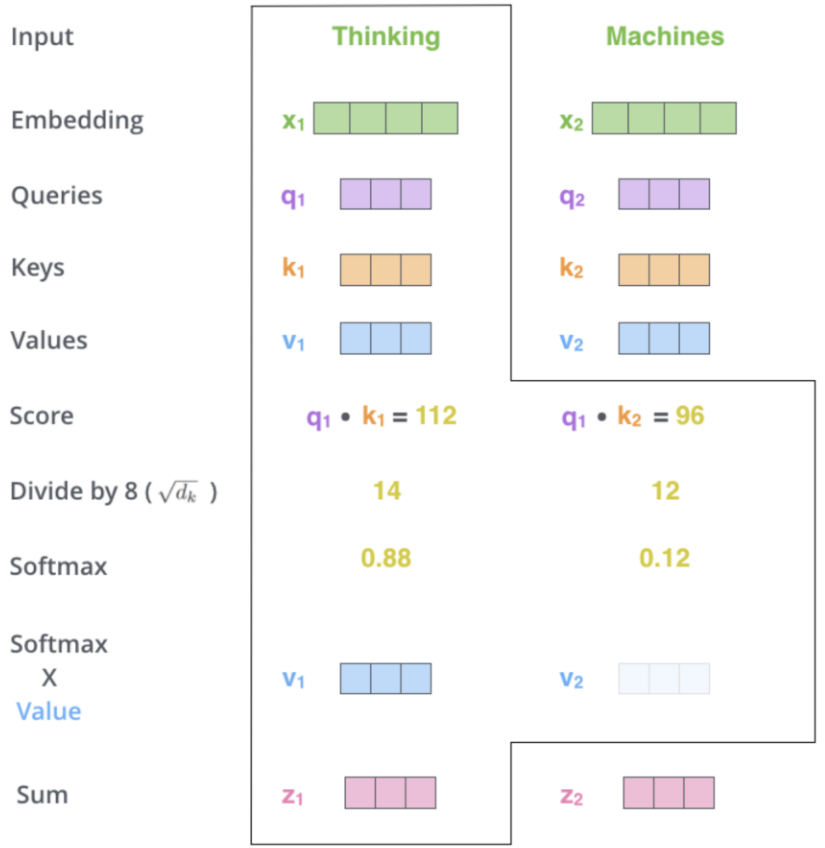
Generative Pre-Trained Transformer (**GPT**) - Attention

Let's take an example sentence.

Alice, who has a black cat, **loves** going to park

- When the model is processing the word “loves”, attention mechanism allows it to associate it with “Alice”
- At each word, attention mechanism allows to look at words at other positions in the input sequence to better encode the word at current position

1. At each input position, calculate query, key and value vectors (a linear transformation of embeddings using learnt weight matrices)
2. Compute dot product between each query and all the keys in the input sequence (attention)
3. Compute a weighted sum of all value vectors using attention weights as coefficients



http://jalammar.github.io/illustrated-transformer/

https://arxiv.org/pdf/1706.03762.pdf

For Wifi use Greenline

Generative AI

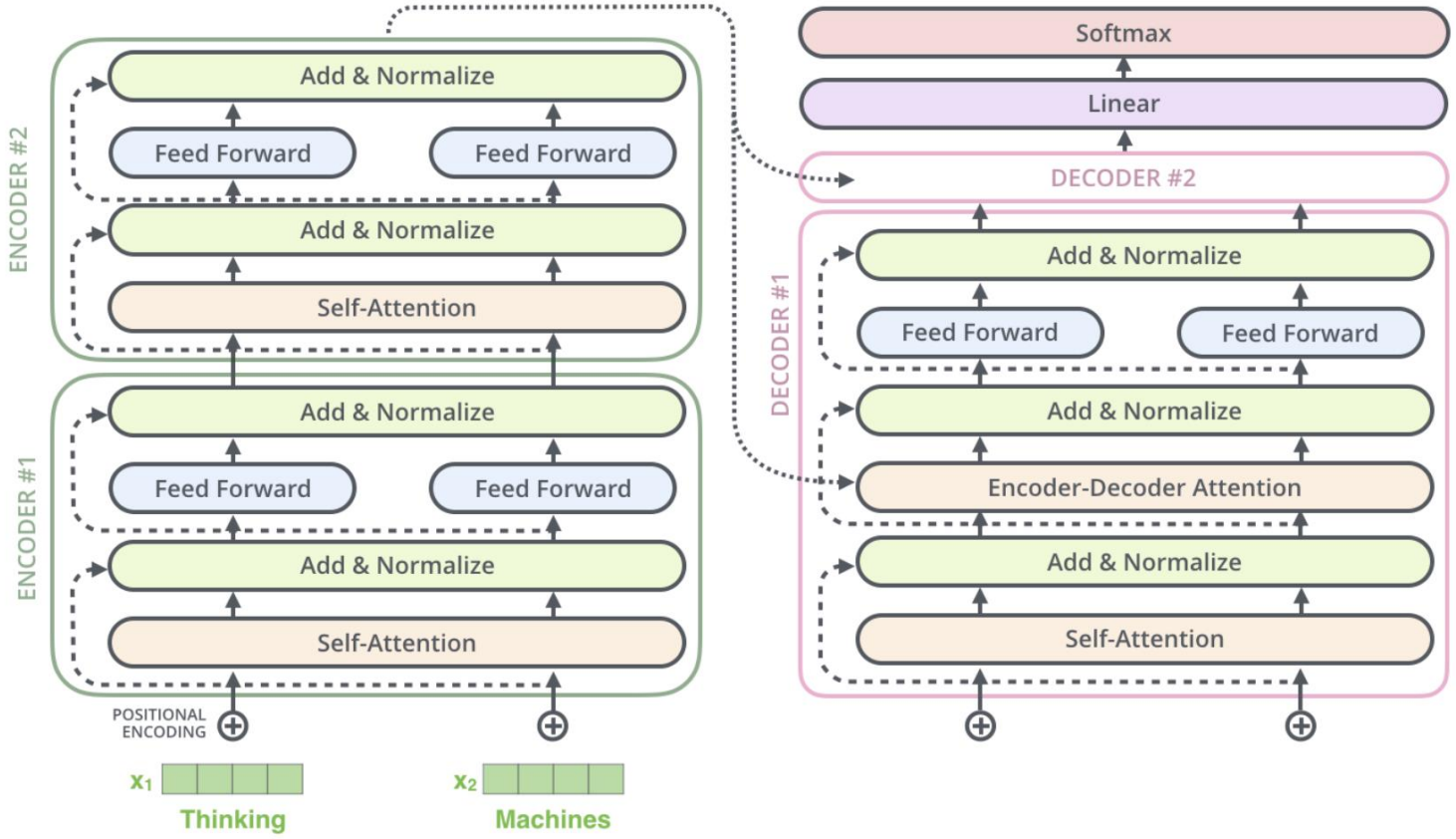
Generative Pre-Trained Transformer (**GPT**) –Transformers Architecture

Architecture

- Six Encoder layers stacked
- Six Decoder layers stacked
- Positional Embeddings
- Masked Attention (Encoder-Decoder Attention)

Advantages

- Better long-range connections
- Easier to parallelize
- Can make the networks much deeper (more layers) than RNNs



https://arxiv.org/pdf/1706.03762.pdf
http://jalammar.github.io/illustrated-transformer/
https://cs182sp21.github.io/static/slides/lec-12.pdf

For Wifi use Greenline

Generative AI

Generative Pre-Trained Transformer (**GPT**)

A **Generative-Pre-Trained Transformer** is a kind of transformers model developed by OpenAI for natural language processing tasks

- **Generative** refers to the model's ability to generate text
- **Pre-Trained** refers to models training process consisting of two stages
  - **Pre-Training**: Model is trained on a large corpus of text data, where the objective is to predict next word in a sentence
  - **Fine-tuning**: Once the model is pre-trained the model can be fine-tuned on a specific task with a task-specific dataset with supervised learning

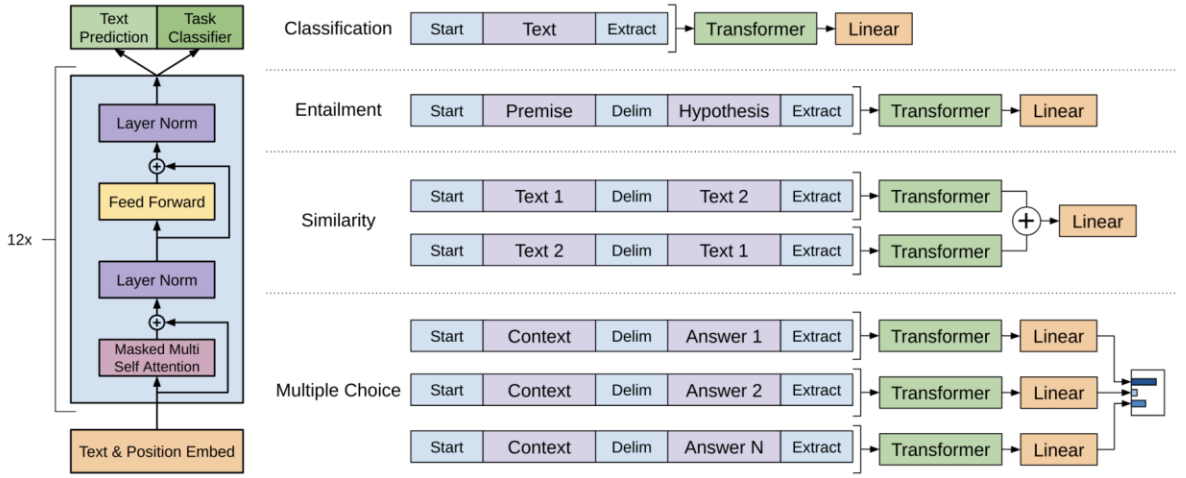


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

https://arxiv.org/pdf/2005.14165.pdf https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

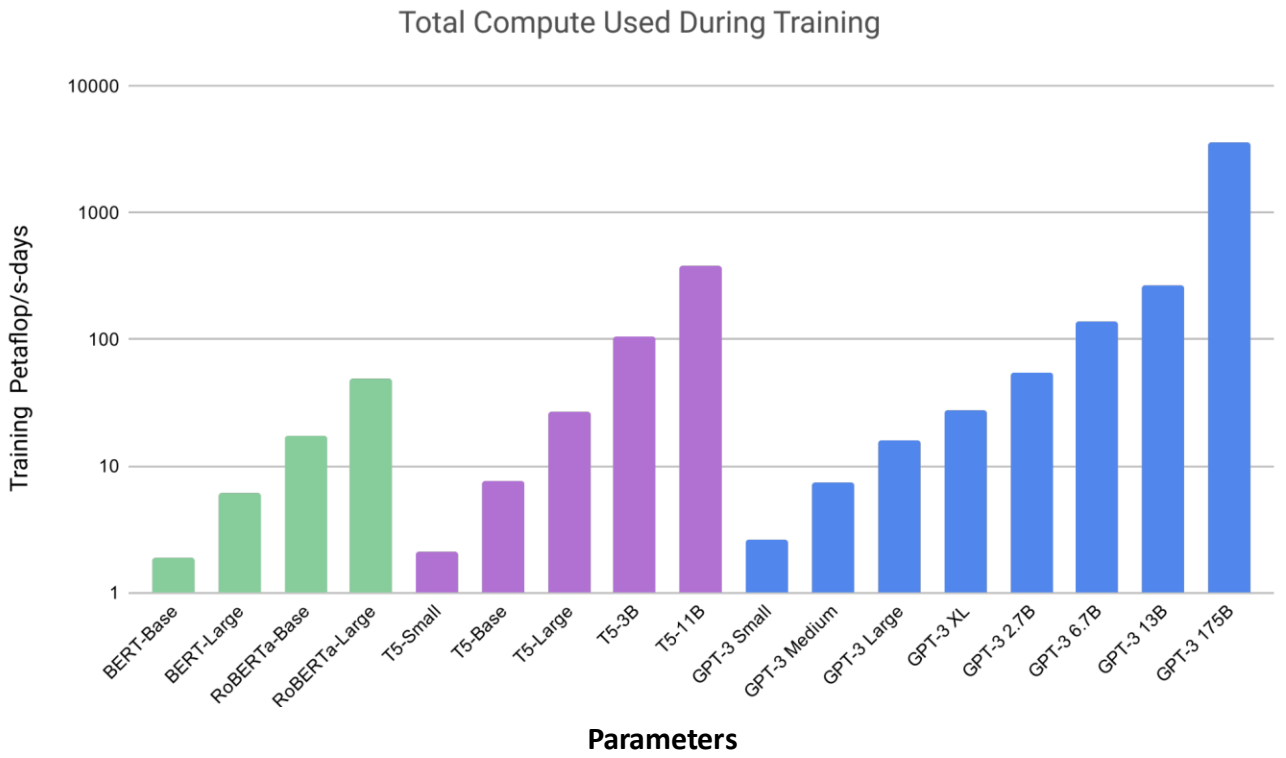
Generative AI

Generative Pre-Trained Transformer (**GPT**) – GPT 3 Training Data and Parameters

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

Table 2.2: Datasets used to train GPT-3. "Weight in training mix" refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

Dataset

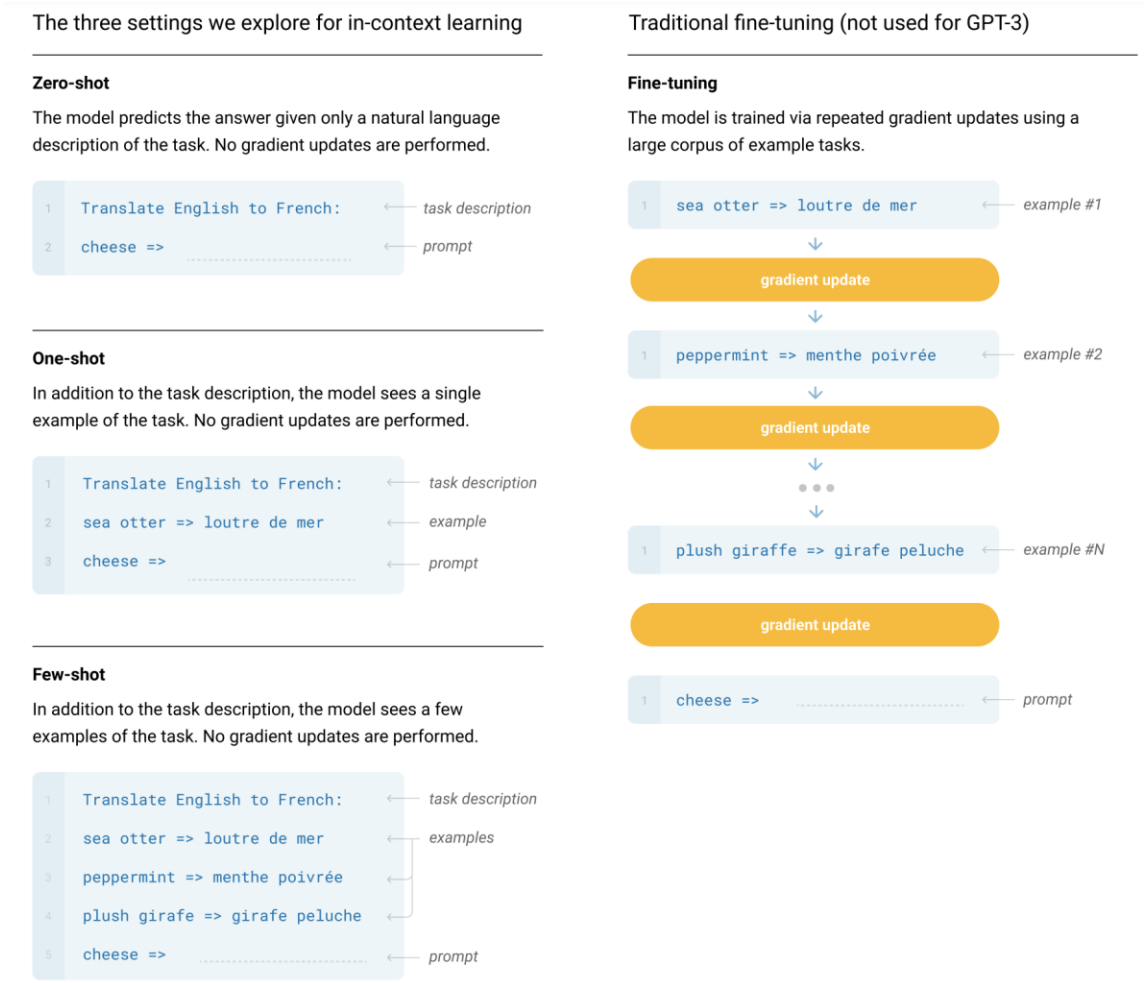


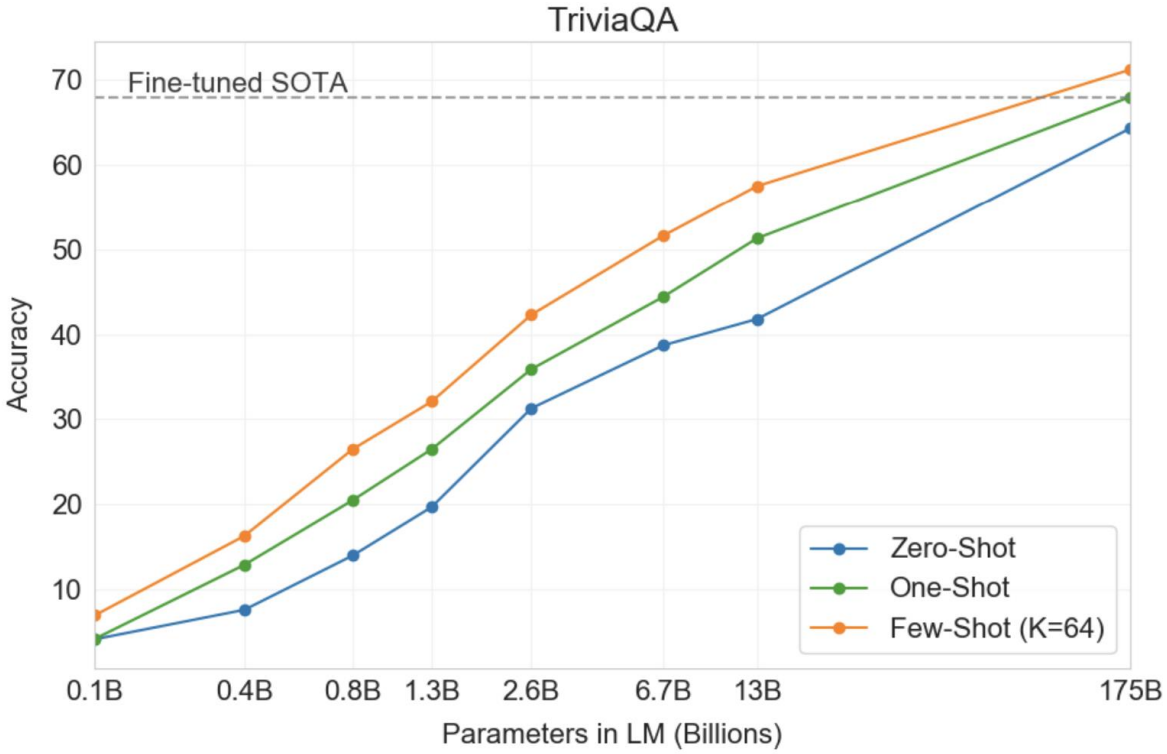
For Wifi use Greenline

https://arxiv.org/pdf/2005.14165.pdf

Generative AI

Generative Pre-Trained Transformer (**GPT**) – GPT 3 Unreasonable Effectiveness





TriviaQA is a reading comprehension dataset containing over 650K question-answer-evidence triples.
https://nlp.cs.washington.edu/triviaqa/

For Wifi use Greenline

https://arxiv.org/pdf/2005.14165.pdf

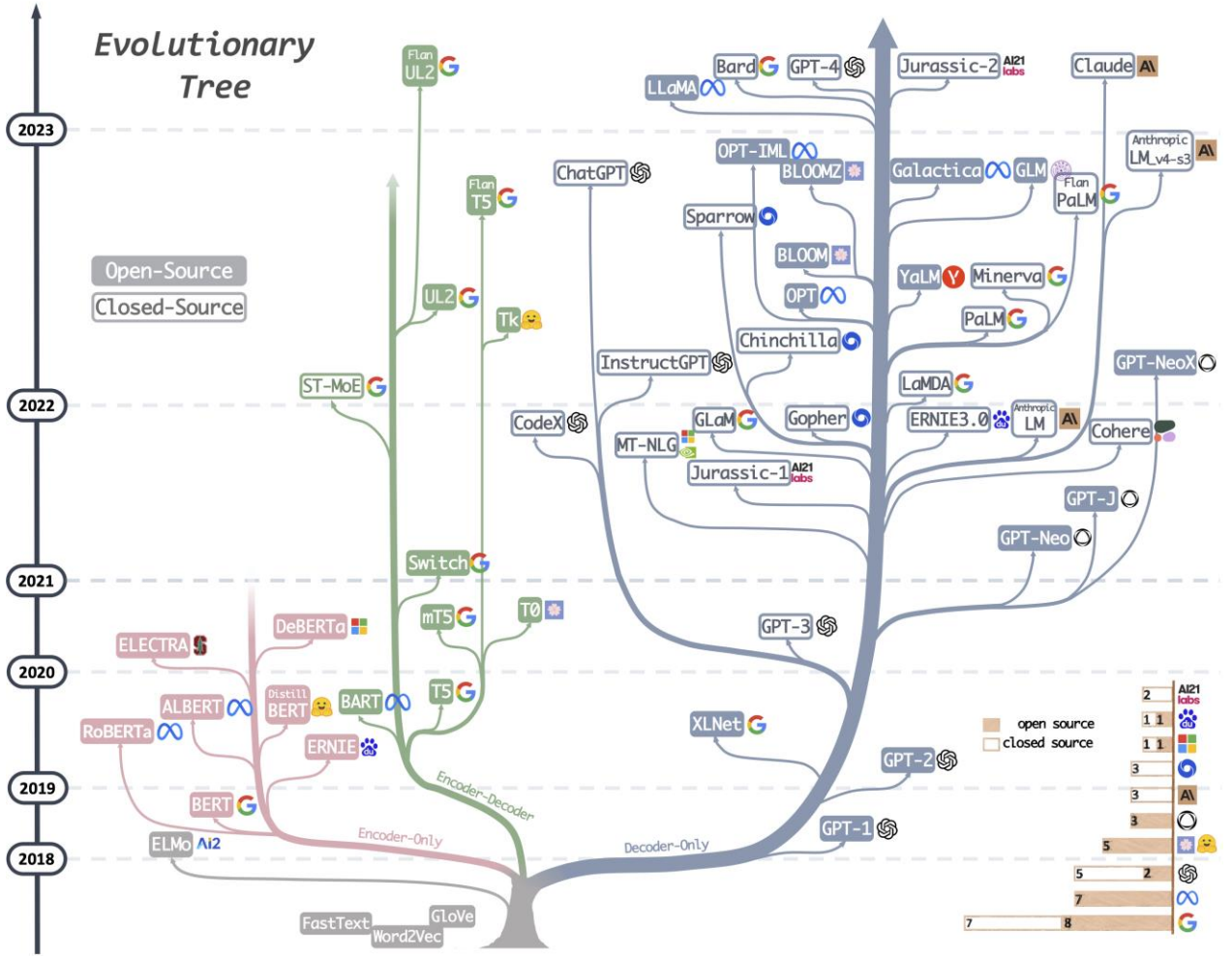
Generative AI

Generative Pre-Trained Transformer (**GPT**) – LLM Landscape

Encoder Models: These models map input sequences to a vector representation. Useful for extracting features (**BERT**)

Decoder Models: These models generate an output sequence from a fixed length input vector. Useful for generation text, images etc. (**GPT-3**)

Encoder-Decoder Models: These models are a combination of both encoder and decoder. Encoder is responsible for mapping input into vector and decoder generates output sequence from that vector. (**BART/ T5/ FLAN UL2**)



https://arxiv.org/pdf/2304.13712.pdf

https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/

For Wifi use Greenline

Generative AI

Generative Pre-Trained Transformer (**GPT**) – Chain-of-thought Prompting

Chain-of-Thought Prompting is a technique that enables LLMs to complex reasoning by generati chain-of-thought, a series of intermediate reasoning steps.

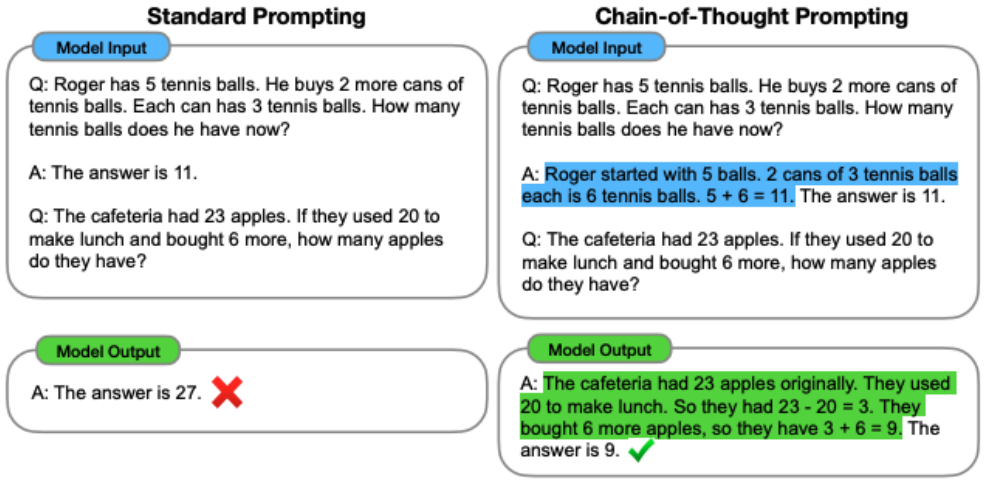


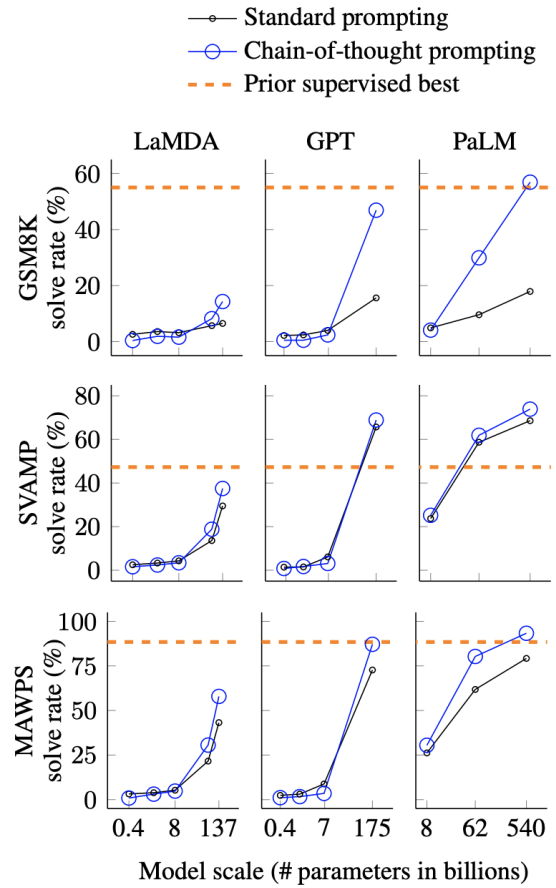
Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Prompting

Table 12: Summary of math word problem benchmarks we use in this paper with examples. N: number of evaluation examples.

Dataset	N	Example problem
GSM8K	1,319	Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?
SVAMP	1,000	Each pack of dvds costs 76 dollars. If there is a discount of 25 dollars on each pack. How much do you have to pay to buy each pack?
ASDiv	2,096	Ellen has six more balls than Marin. Marin has nine balls. How many balls does Ellen have?
AQuA	254	A car is being driven, in a straight line and at a uniform speed, towards the base of a vertical tower. The top of the tower is observed from the car and, in the process, it takes 10 minutes for the angle of elevation to change from 45° to 60°. After how much more time will this car reach the base of the tower? Answer Choices: (a) $5\sqrt{3}$ + 1 (b) $6\sqrt{3}$ + $\sqrt{2}$ (c) $7\sqrt{3}$ - 1 (d) $8\sqrt{3}$ - 2 (e) None of these
MAWPS: SingleOp	562	If there are 7 bottle caps in a box and Linda puts 7 more bottle caps inside, how many bottle caps are in the box?
MAWPS: SingleEq	508	Benny bought a soft drink for 2 dollars and 5 candy bars. He spent a total of 27 dollars. How much did each candy bar cost?
MAWPS: AddSub	395	There were 6 roses in the vase. Mary cut some roses from her flower garden. There are now 16 roses in the vase. How many roses did she cut?
MAWPS: MultiArith	600	The school cafeteria ordered 42 red apples and 7 green apples for students lunches. But, if only 9 students wanted fruit, how many extra did the cafeteria end up with?

Datasets and Example Problems



Performance

https://arxiv.org/pdf/2201.11903.pdf

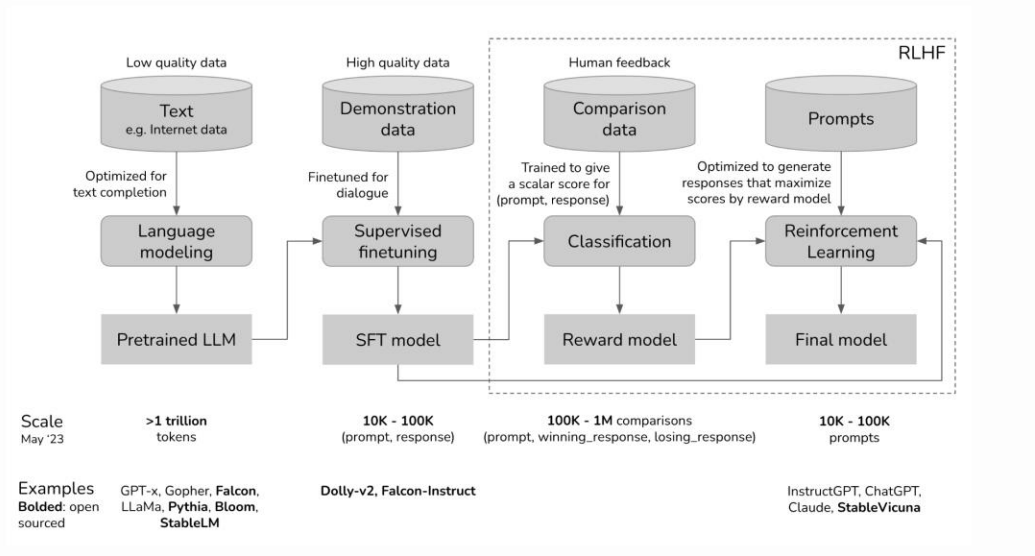
For Wifi use Greenline

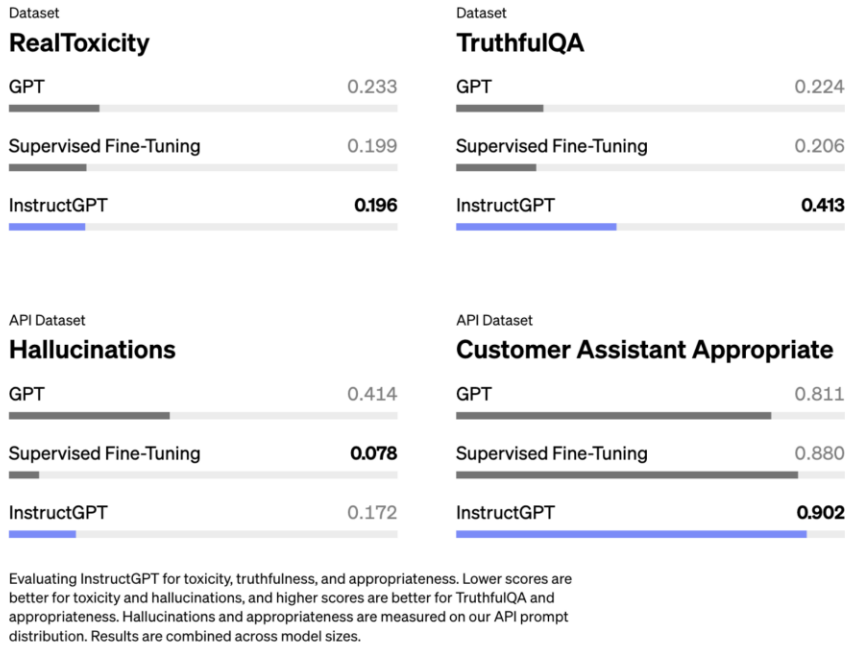
Generative AI

Generative Pre-Trained Transformer (**GPT**) – Alignment - RLHF

- **Reinforcement Learning through Human Feedback** is technique that allows models to learn directly from human feedback (like prompting) without the need for labeled data

- Due to the nature of training data being scrapped from internet (contains a lot of mis-information, conspiracy theories etc..) the models must be **further polished/aligned using RLHF** to make it user appropriate





Evaluating InstructGPT for toxicity, truthfulness, and appropriateness. Lower scores are better for toxicity and hallucinations, and higher scores are better for TruthfulQA and appropriateness. Hallucinations and appropriateness are measured on our API prompt distribution. Results are combined across model sizes.

https://arxiv.org/pdf/2212.08073.pdf

https://huyenchip.com/2023/05/02/rlhf.html

For Wifi use Greenline

Generative AI

Generative Adversarial Networks (**GANs**)

Imagine you have a bunch of **cat images**, and you want a machine learning model to create similar images. This is exactly what a GAN does.

Generator: Takes in random numbers as input and generates the images of interest (**the forger**)

Discriminator: Takes both the images from the generator and the real images from the data and spots the difference between them (**the detective**)

Both the generator and the discriminator are trained together. And, over the duration of training, the generator gets better at creating images which look real, and the discriminator gets better at spotting fakes.

Adversarial Objective: These two networks are pitted against each other where the **generator creates more realistic synthetic images to fool the discriminator** while the **discriminator networks tries to get better at detecting fake images**. This back-and-forth strategy forces both the networks to improve until the generator can create highly realistic synthetic images, that indistinguishable from real images

https://arxiv.org/pdf/1406.2661.pdf

Generative AI

Diffusion Models

Diffusion models are another class of Generative models which work by adding noise to the images in the training data by a process called **forward diffusion process** and then reversing the process to recover the original image using **reverse diffusion**. These models can be trained on **large unlabeled datasets in an unsupervised manner**.

Stable Diffusion: Stable Diffusion is **a text-to-image model** from Stability AI. A stable diffusion model has four important elements

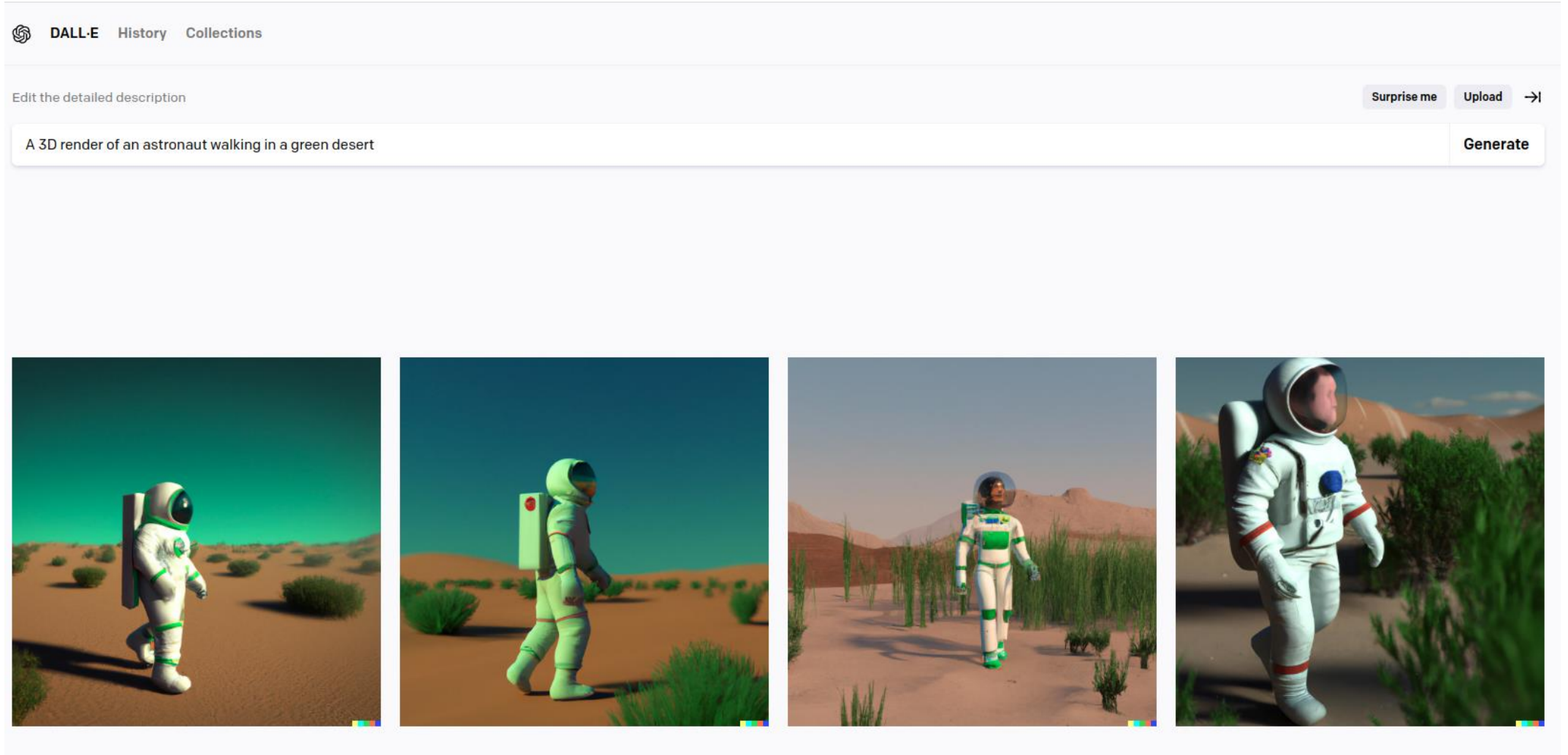
- **Diffusion Probabilistic Model**: Like the above-mentioned diffusion process, noise is added to the inputs repeatedly and then is removed to produce clean data. This enables the model to understand the actual structure of real images
- **U-Net Architecture**: A Conv-Net architecture with an encoder-decoder structure takes in a noisy image and generates a denoised image at each time step.
- **Latent Text Encoding**: A transformer-based text encoder to condition the image generation on text. Enables the model to generate images relevant to text
- **Classifier-Free Guidance**: Stable Diffusion uses a technique called CFG which directly predicts image pixels from the source image and text encoding unlike in GAN, we use a classifier to differentiate real or fake during training

https://stablediffusionweb.com/

For Wifi use Greenline

Applications

Dall-E Example 1

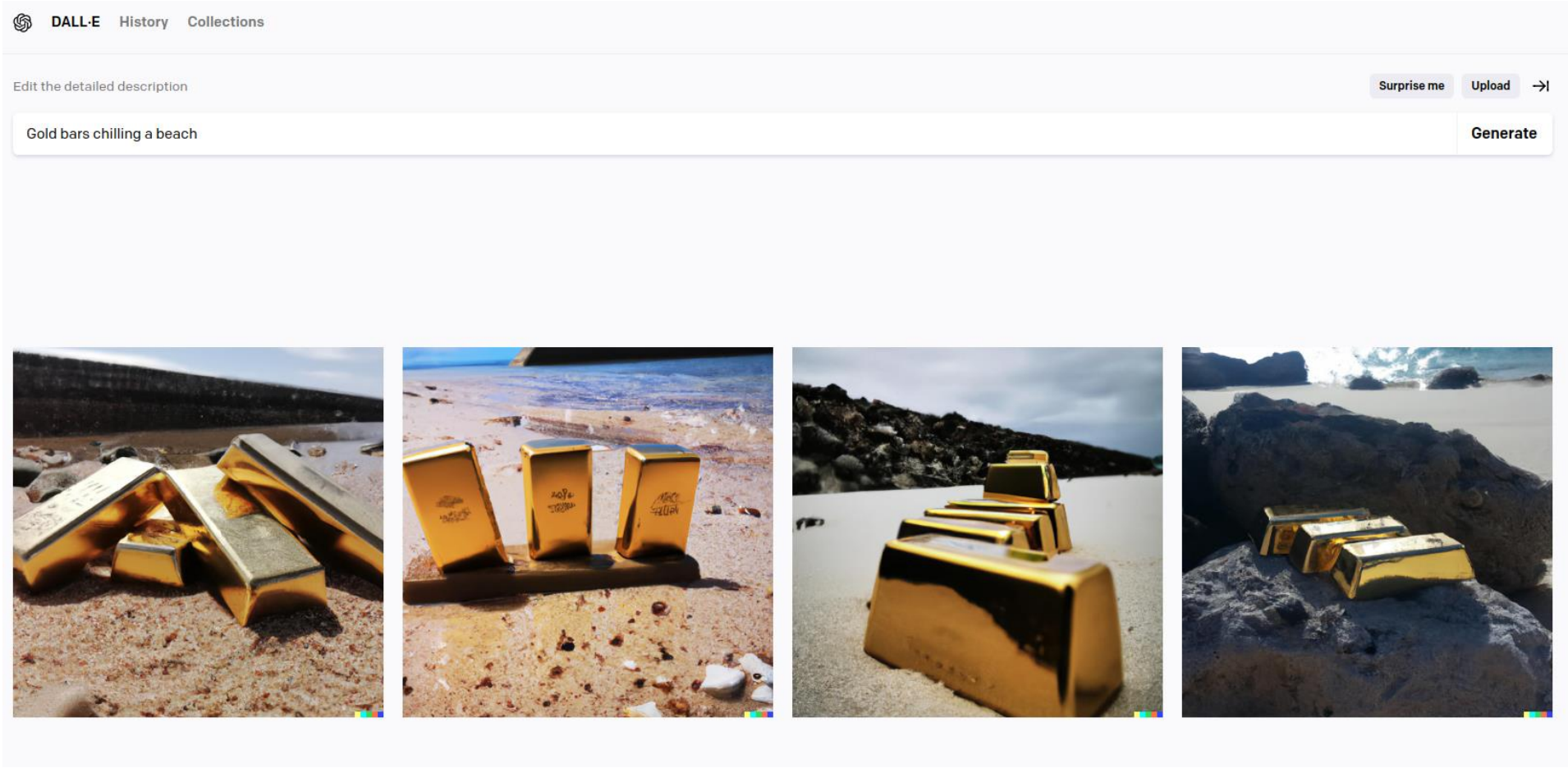


For Wifi use Greenline

Images Generated for this presentation

Applications

Dall-E Example 2

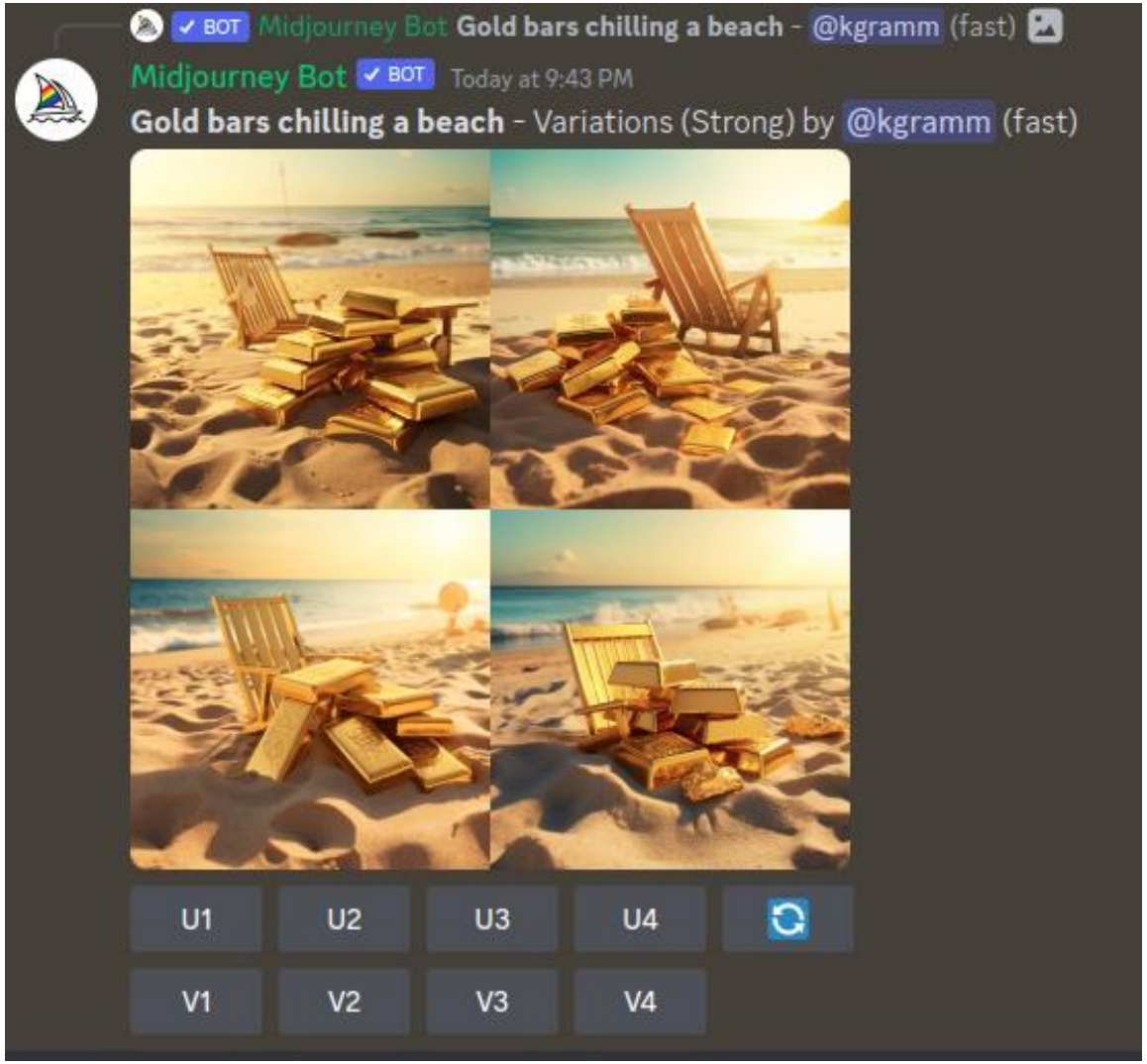


Images Generated for this presentation

For Wifi use Greenline

Applications

Midjourney



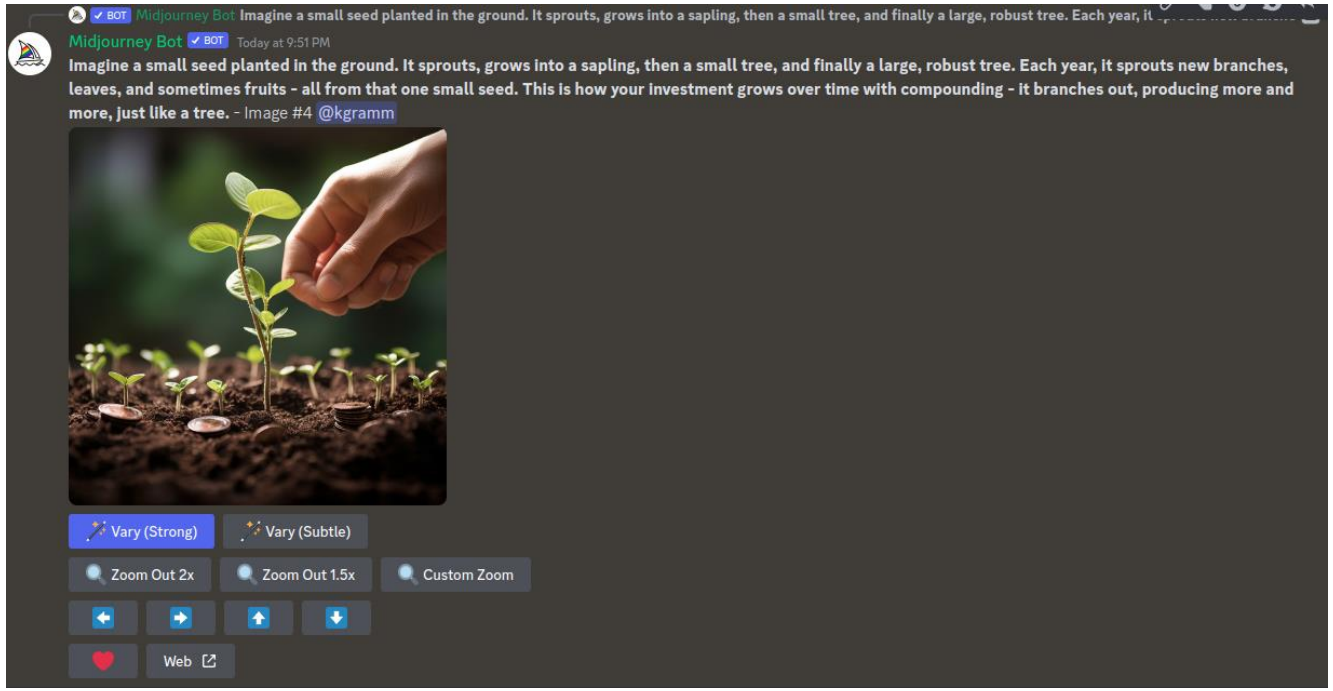
For Wifi use Greenline

Image Generated for this presentation

Applications

Midjourney

Prompt: *Imagine a small seed planted in the ground. It sprouts, grows into a sapling, then a small tree, and finally a large robust tree. Each year, it sprouts new branches, leaves and sometimes fruits – all from that small seed. This is how your investment grows with compounding – It branches out producing more and more just like a tree*

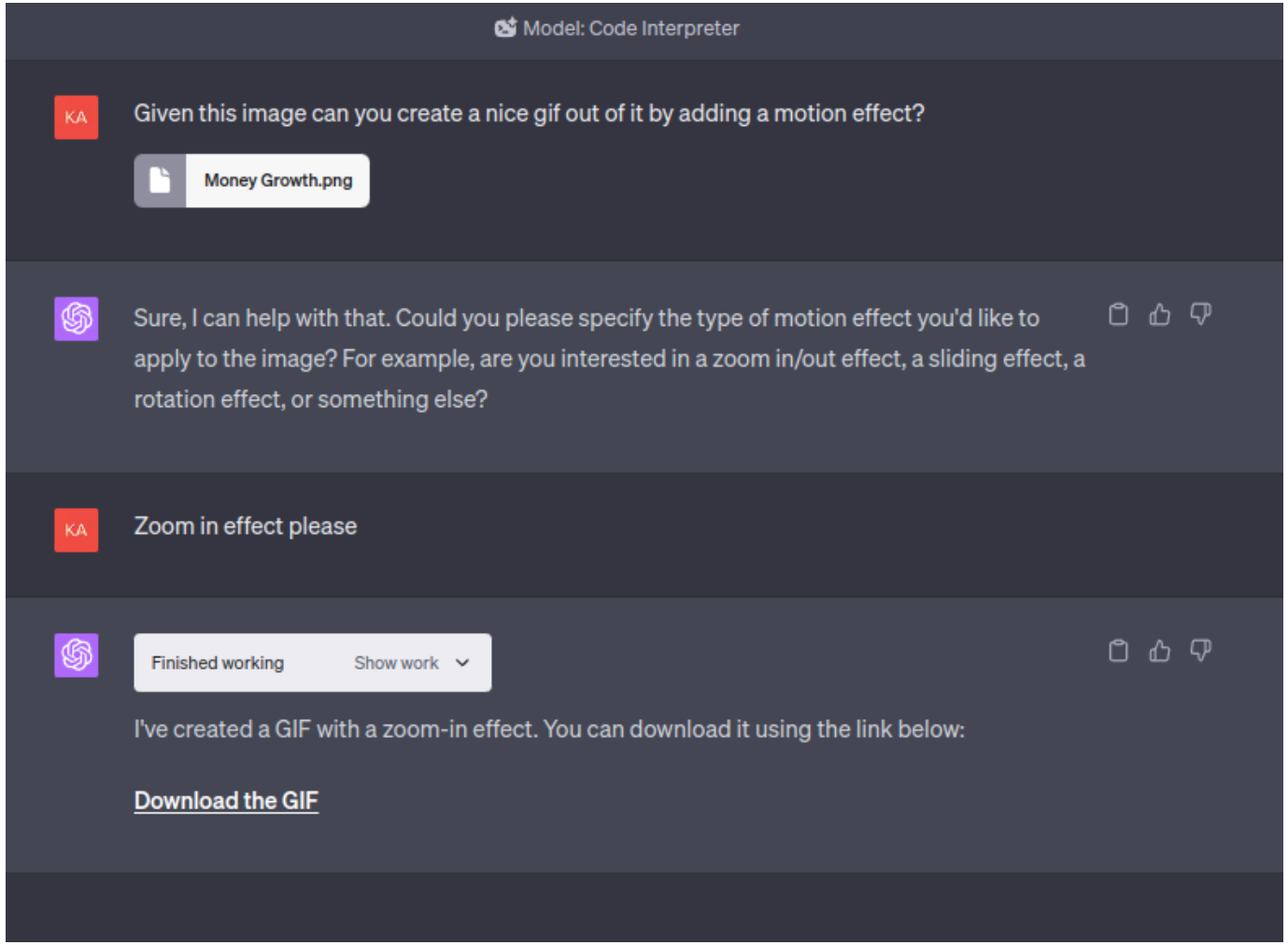


For Wifi use Greenline

Images Generated for this presentation

Applications

ChatGPT



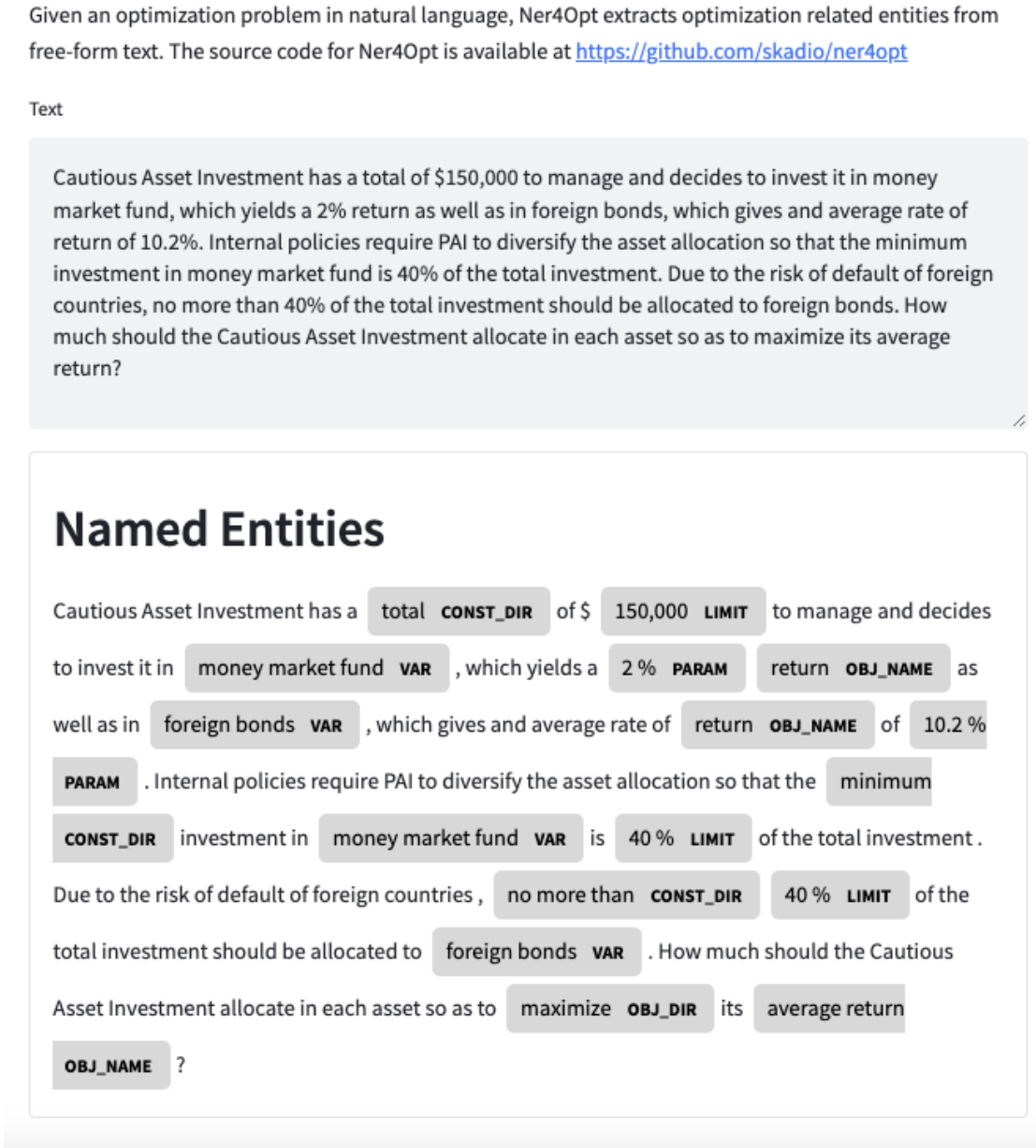
For Wifi use Greenline

Images Generated for this presentation

Generative AI at Fidelity

Ner4Opt: Named Entity Recognition for Optimization Modelling from Natural Language

- ❑ Envision **automated modeling assistant** to turn natural language into optimization formulations
- ❑ Necessary **building block**: finding key pieces of information relevant to optimization
- ❑ **Ner4Opt**: extracting optimization-related information such as the objective, constraints, and variables from free-form natural language text



https://link.springer.com/chapter/10.1007/978-3-031-33271-5_20

For Wifi use Greenline

https://huggingface.co/spaces/skadio/Ner4Opt

Generative AI at Fidelity

Ner4Opt: Named Entity Recognition for Optimization Modelling from Natural Language

NER4OPT: Named Entity Recognition for Optimization Modelling from Natural Language

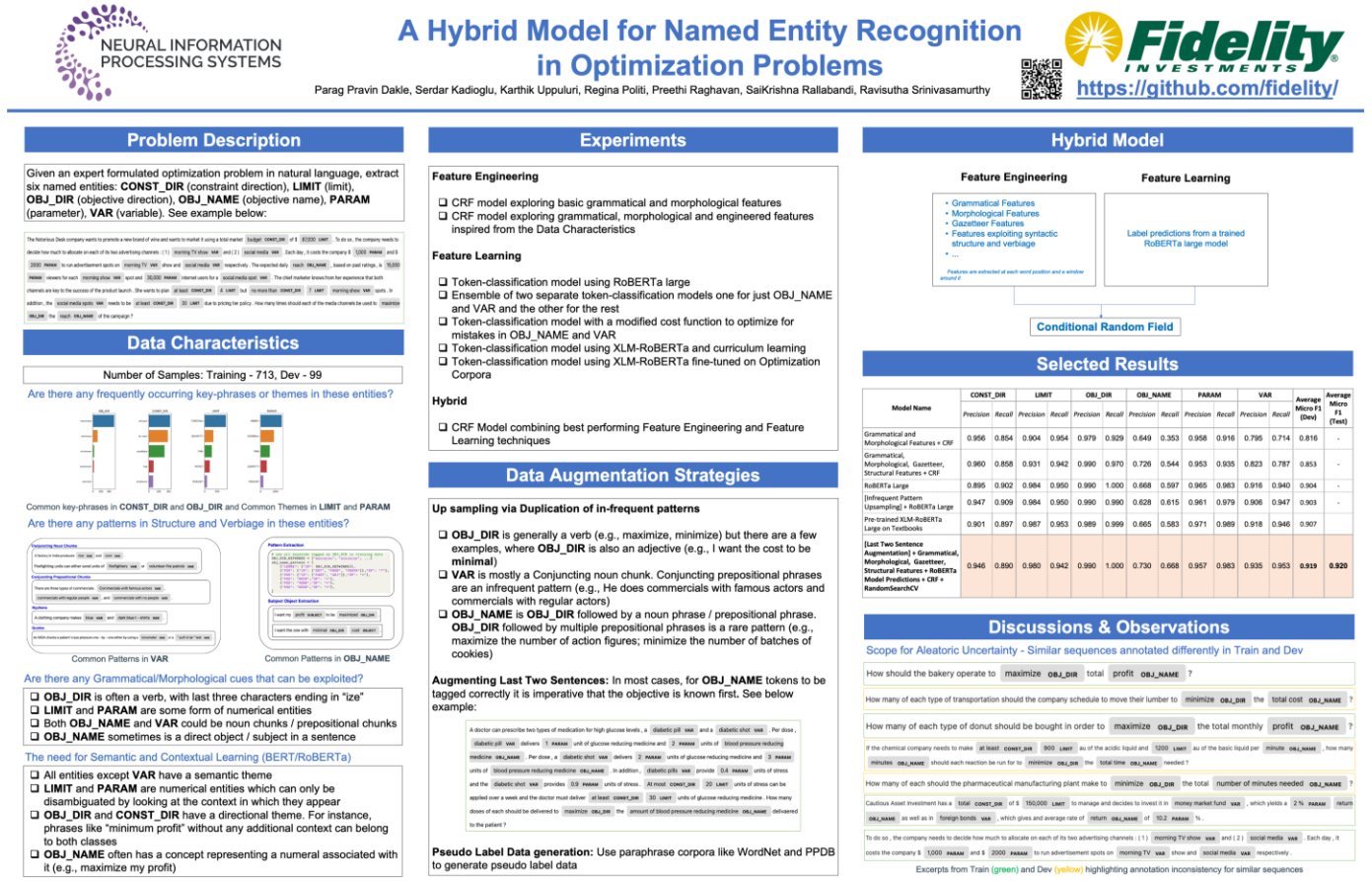
Parag Pravin Dakle[1], Serdar Kadıoğlu[1,2][0000−0002−4672−6830], Karthik Uppuluri[1], Regina Politi[1], Preethi Raghavan[1], SaiKrishna Rallabandi[1], and Ravisutha Srinivasamurthy[1]

[1] AI Center of Excellence, Fidelity Investments, Boston, USA
[2] Dept. Computer Science, Brown University, Providence, USA
{firstname.lastname}@fmr.com

Abstract. Solving combinatorial optimization problems involves a two-stage process that follows the model-and-run approach. First, a user is responsible for formulating the problem at hand as an optimization model, and then, given the model, a solver is responsible for finding the solution. While optimization technology has enjoyed tremendous theoretical and practical advances, the overall process has remained the same for decades. To date, transforming problem descriptions into optimization models remains a barrier to entry. To alleviate users from the cognitive task of modeling, we study named entity recognition to capture components of optimization models such as the objective, variables, and constraints from free-form natural language text, and coin this problem as NER4OPT. We show how to solve NER4OPT using classical techniques based on morphological and grammatical properties and modern methods leveraging pre-trained large language models and fine-tuning transformers architecture with optimization-specific corpora. For best performance, we present their hybridization combined with feature engineering and data augmentation to exploit the language of optimization problems. We improve over the state-of-the-art for annotated linear programming word problems, identify several next steps and discuss important open problems toward automated modeling.

Keywords: Optimization Modeling · Named Entity Recognition · Natural Language Processing

https://link.springer.com/chapter/10.1007/978-3-031-33271-5_20



For Wifi use Greenline

Generative AI at Fidelity

Ner4Opt: Named Entity Recognition for Optimization Modelling from Natural Language

Method	CONST_DIR		LIMIT		OBJ_DIR		OBJ_NAME		PARAM		VAR		Average Micro F1
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	
Classical	0.956	0.854	0.904	**0.954**	0.979	0.929	0.649	0.353	0.958	0.916	0.795	0.714	0.816
Classical+	**0.960**	0.858	0.931	0.942	**0.990**	0.970	0.726	0.544	0.953	0.935	0.823	0.787	0.853
Xlm-Rb [51]	0.887	0.897	0.965	0.950	0.949	0.999	0.617	0.469	0.960	0.969	0.909	0.932	0.888
Xlm-Rl	0.930	0.897	0.979	0.938	0.979	0.989	0.606	0.512	0.963	0.985	0.899	0.938	0.893
RoBERTa	0.895	**0.902**	0.984	0.950	**0.990**	**1.000**	0.668	0.597	0.965	0.983	0.916	0.940	0.904
Xlm-Rl+	0.901	0.897	**0.987**	0.953	0.989	0.999	0.665	0.583	**0.971**	**0.989**	0.918	0.946	0.907
Hybrid	0.946	0.890	0.980	0.942	**0.990**	**1.000**	**0.730**	**0.668**	0.957	0.983	**0.935**	**0.953**	**0.919**

Table 2. Numerical results that compare classical, modern, and hybrid models for precision, $\mathcal{P}$, and recall, $\mathcal{R}$ for each named entity together with average micro F1 score.

https://link.springer.com/chapter/10.1007/978-3-031-33271-5_20

For Wifi use Greenline

https://nl4opt.github.io/results/

Generative AI at Fidelity

Understanding BLOOM: An empirical study on diverse NLP tasks

Compare the Open-Source BLOOM with other models like BERT/GPT

- **Does performance of BLOOM scale with parameters?**

Authors noticed that performance of BLOOM doesn't scale with parameter size unlike models like BERT

- **Does finetuning improve the performance?**

Authors added multiple zero-shot cross-lingual and multi-lingual fine-tuning experiments suggesting BLOOM is at par or worse than monolingual GPT-2 models

- **What about toxicity in the generated data?**

Toxicity analysis of prompt-based text generation using the RealToxicity Prompts dataset shows that the text generated by BLOOM is at least 17% less toxic than GPT-2 and GPT-3 models.

Understanding BLOOM: An empirical study on diverse NLP tasks

Parag Pravin Dakle, SaiKrishna Rallabandi, Preethi Raghavan
AI Center of Excellence, Fidelity Investments, Boston MA
{paragpravin.dakle, saikrishna.rallabandi, preethi.raghavan}@fmr.com

Model	toxicity	severe toxicity	obscene
gpt2	0.313	0.022	0.183
gpt3 (Brown et al., 2020)	0.331	0.021	0.178
bloom-1b7	**0.26**	**0.016**	**0.122**

Table 7: Results of toxicity analysis on RealToxicityPrompts dataset.

For Wifi use Greenline

https://arxiv.org/pdf/2211.14865.pdf

Generative AI at Fidelity

Correcting Semantic Parses with Natural Language through Dynamic Schema Encoding

- There are several semantic and syntactic challenges in converting Natural Language Text to SQL queries
- In this paper, authors approach Semantic Parse Correction using Natural Language Feedback
- With just one-turn of correction, authors saw an improvement of accuracy up to 26%
- They also show that a base T-5 model can correct the errors of a T-5 large model in a zero-shot cross parser setting.

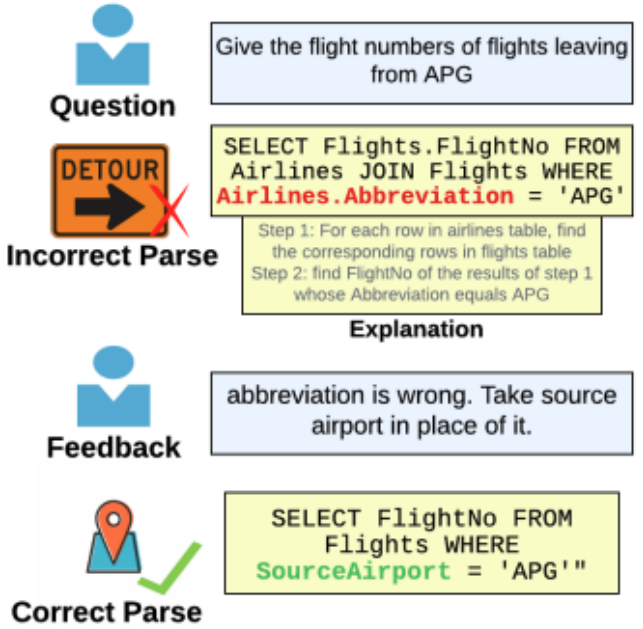


Figure 1: Example item from the SPLASH dataset. An incorrect parse from a neural text-to-SQL model is paired together with natural language feedback commenting on how the parse should be corrected.

https://arxiv.org/pdf/2305.19974.pdf

For Wifi use Greenline

Generative AI at Fidelity

Guardrails for Generative AI

Guardrails are crucial for Generative AI models to safeguard the end users from mis-information, biases, toxicity etc..

There have been a few Open-source initiative around this like

- Nemo Guardrails (NVIDIA) - an open-source toolkit for easily adding programmable guardrails to LLM-based conversational systems.

- Guardrails AI - a Python package that lets a user add structure, type and quality guarantees to the outputs of large language models (LLMs)

Along the same lines with these initiatives, we are interested in an extensive set of guardrails and compliance requirements to check for hallucination in generations, tonality, abusive language, bias, violence, financial scams, malicious injections etc..

Generative AI

Emerging Trends

- Major breakthroughs in deep learning architectures like Transformers and Generative Adversarial Networks
- Availability of massive datasets and GPU/TPU compute
- New advances in techniques like RLHF/Prompting made it much easy to align these models
- Low barrier of entry due to intuitive and user-friendly interfaces and strong open-source ecosystem
- GenAI holds potential to create photo-realistic images, human-like speech and text and generate working code from natural language descriptions which was not possible until recently

Things to Keep in Mind

1. **Lack of Consistency (Hallucination)**: LLMs tend to produce wildly different answers, when the same question is asked multiple times

2. **Bias**: As the models are trained on data scrapped from internet, they might have inherited the biases present in the training data

3. **Interpretability**: It is difficult to understand why a particular response or content is generated, making it very challenging for use cases where explainability is inherently required.

4. **Real-time Knowledge**: As the models are trained on a fixed dataset at a particular point in time, they lack information/changes that occurred after that point.

5. **Memory**: Even though these models are getting good with context lengths that can be supported, having an efficient memory remembering the important details of conversations over a long period of time is still a challenging task.

6. **Engineering Challenges**: Operating these semi-non-deterministic models especially in a multi-model setting (including voice, text, images etc..) at scale remains a significant challenge

Potential of Generative AI

1. **Low Resource Languages** – Ability to understand, generate any language, especially low resource ones, could help study languages and historical documents in general

2. **Inclusion and Accessibility** – Avatars proficient in sign languages, high precision caption generation etc., could increase accessibility for all people

3. **Personalized Content Generation** – Video games, music, movies can be created that cater to users and individual interests at scale

4. **AI Tutors** – Imagine a world where you can conjure up a tutor to teach you any skill you would like to learn at your own pace

5. **Intelligent Assistants** – Laborious and repetitive tasks can be delegated to Intelligent Assistants allowing humans to focus on critical thinking and decision making

6. **Accelerating Scientific Discovery**- General advances in AI can help accelerate scientific discovery by generating deep insights from massive datasets and design new algorithms. This can help solve most challenging problems we face today.

https://www.forbes.com/sites/bernardmarr/2023/05/31/the-future-of-generative-ai-beyond-chatgpt/?sh=161c85da3da9

AI Center of Excellence @ Fidelity

Research & Open-Source Software

❑ [arXiv'23] Explainable AI with Booleans	BoolXAI	https://github.com/fidelity/boolxai
❑ [NeurIPS'22, CPAIOR'23] NER for Optimization	Ner4Opt	https://github.com/skadio/ner4opt
❑ [IJAIT'21] Recommender Systems	Mab2Rec	https://github.com/fidelity/mab2rec
❑ [AAAI'21] NLP/Text Featurization	TextWiser	https://github.com/fidelity/textwiser
❑ [ICTAI'20] Multi-Armed Bandits	MABWiser	https://github.com/fidelity/mabwiser
❑ [AI Magazine'23, AAAI'22] Sequential Mining	Seq2Pat	https://github.com/fidelity/seq2pat
❑ [CPAIOR'22] Feature Selection	Selective	https://github.com/fidelity/selective
❑ [ICMLA'21] Fairness & Bias Mitigation	Jurity	https://github.com/fidelity/jurity

https://jobs.fidelity.com/

github/fidelity

For Wifi use Greenline