AI and Society LLM Seminar 4: LLM safety (1): ‘harmful content’ generation

Ali Knott
School of Engineering and Computer Science, VUW

Recap: LLM series so far

- Seminar 1: How LLMs work (Ali)
- Seminar 2: How to use LLMs: 'prompt engineering' (Simon)
- Seminar 3: LLMs in education
  (Kit Willett, Kathleen Kaveney, Robin Caygill, Neil Miller, Simon)
- Workshop: Use of LLMs in government and commercial domains (Simon)

The next few seminars

The next few seminars

- Seminar 4 (today): How to stop LLMs producing *harmful content*.

The next few seminars

- Seminar 4 (today): How to stop LLMs producing *harmful content*.
- Seminar 5: How to stop LLMs reporting *false content* as if it's factually true.
  And how to create *transparency* about content generated by LLMs.

The next few seminars

- Seminar 4 (today): How to stop LLMs producing *harmful content*.
- Seminar 5: How to stop LLMs reporting *false content* as if it's factually true.
  And how to create *transparency* about content generated by LLMs.

- Seminar 6: An introduction to AI methods for generating images. (Callum Sleigh)

The next few seminars

- Seminar 4 (today): How to stop LLMs producing *harmful content*.
- Seminar 5: How to stop LLMs reporting *false content* as if it's factually true.
  And how to create *transparency* about content generated by LLMs.

- Seminar 6: An introduction to AI methods for generating images. (Callum Sleigh)

A new term:

A foundation model (FM) is an AI generator of *any content* (including images).

The next few seminars

- Seminar 4 (today): How to stop LLMs producing *harmful content*.
- Seminar 5: How to stop LLMs reporting *false content* as if it's factually true.
  And how to create *transparency* about content generated by LLMs.

- Seminar 6: An introduction to AI methods for generating images. (Callum Sleigh)

A new term:
A foundation model (FM) is an AI generator of *any content* (including images).

- Seminar 7: The ecosystem of AI systems that *use* FMs. ('Downstream apps') (Simon & guests)

In today's lecture

1. GPT models and harmful content
   - A taxnomy of harmful content
   - Alignment methods used in GPT-4, to steer its output away from such content.

2. Content that's biased towards/against some given group.

Preamble: What GPT generates

Preamble: What GPT generates

GPT is a probabilistic language model: it just produces texts that statistically resemble the ones it was trained on.

Preamble: What GPT generates

GPT is a probabilistic language model: it just produces texts that statistically resemble the ones it was trained on.

- Importantly, it doesn't just *reproduce* the texts it trained on.

Preamble: What GPT generates

GPT is a probabilistic language model: it just produces texts that statistically resemble the ones it was trained on.

- Importantly, it doesn't just *reproduce* the texts it trained on.
  - It has amazing abilities to *interpolate* between its training texts, and produce brand new texts.

Preamble: What GPT generates

GPT is a probabilistic language model: it just produces texts that statistically resemble the ones it was trained on.

- Importantly, it doesn't just *reproduce* the texts it trained on.
  - It has amazing abilities to *interpolate* between its training texts, and produce brand new texts.
- This allows it to respond really convincingly to *prompts it has never seen before*.

An example of interpolation

The paper introducing GPT-2 demonstrated that it wasn't just parroting training texts, by asking it to continue a story that was *definitely new*.

An example of interpolation

The paper introducing GPT-2 demonstrated that it wasn't just parroting training texts, by asking it to continue a story that was *definitely new*.

> *In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains.*

An example of interpolation

The paper introducing GPT-2 demonstrated that it wasn't just parroting training texts, by asking it to continue a story that was *definitely new*.

> *In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*

An example of interpolation

The paper introducing GPT-2 demonstrated that it wasn't just parroting training texts, by asking it to continue a story that was *definitely new*.

> *In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*

GPT-2 continued as follows:

An example of interpolation

The paper introducing GPT-2 demonstrated that it wasn't just parroting training texts, by asking it to continue a story that was *definitely new*.

> *In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*

GPT-2 continued as follows:

> *The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.*

An example of interpolation

The paper introducing GPT-2 demonstrated that it wasn't just parroting training texts, by asking it to continue a story that was *definitely new*.

> *In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*

GPT-2 continued as follows:

> *The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.*
>
> *Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.*

An example of interpolation

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans.

An example of interpolation

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans.

Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

An example of interpolation

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans.

Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

An example of interpolation

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them—they were so close they could touch their horns.

An example of interpolation

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them—they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

An example of interpolation

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them—they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

An example of interpolation

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them—they were so close they could touch their horns.

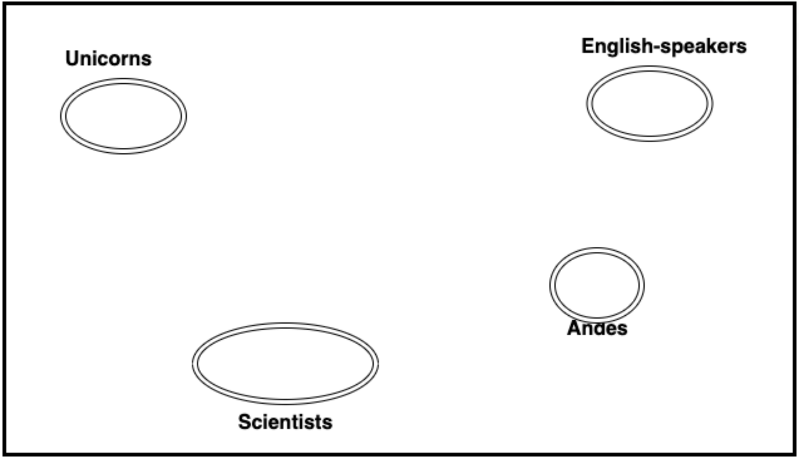
While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

How can GPT make up stuff it hasn't seen in training??

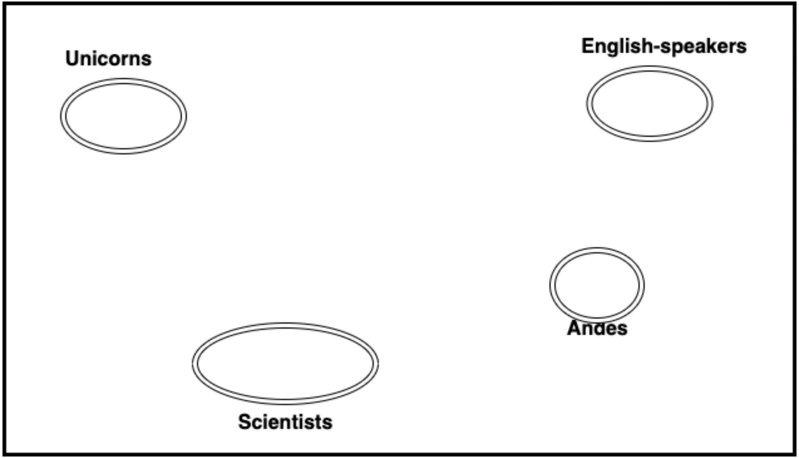
Visualising how GPT works

During training, GPT learns about a huge *space of possible texts*.



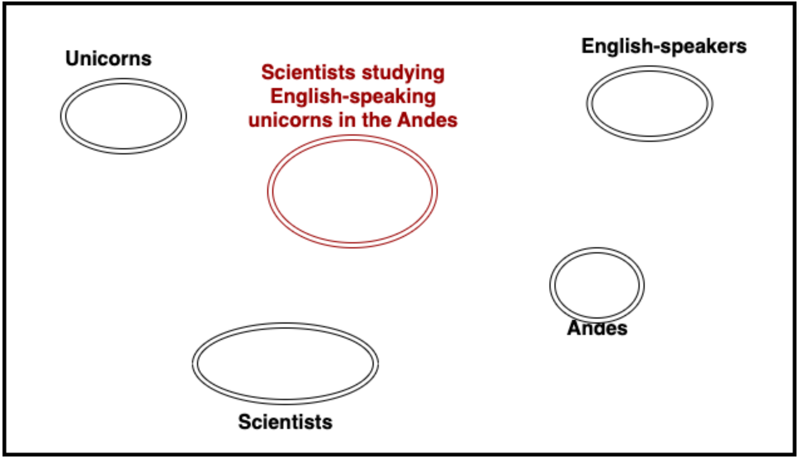
Visualising how GPT works

This includes actual texts, but also an infinity of texts it *never saw* in training.



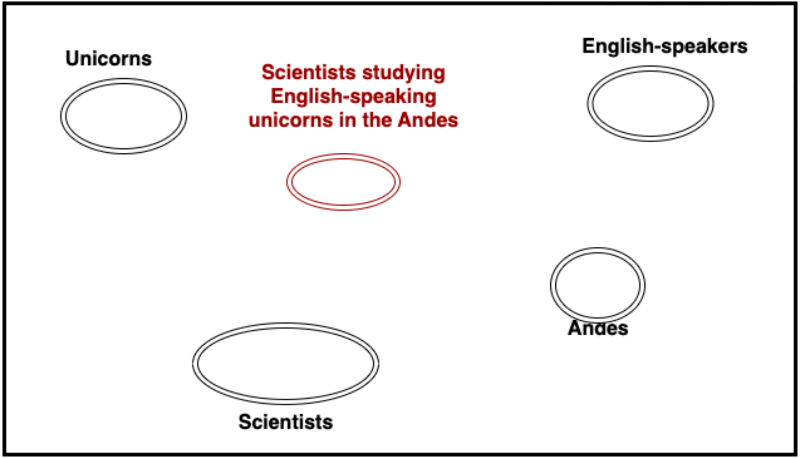
Visualising how GPT works

When you give GPT a *prompt*, you're basically pointing to a *region* of this text space, and saying 'I want you to produce a text from *here*!'



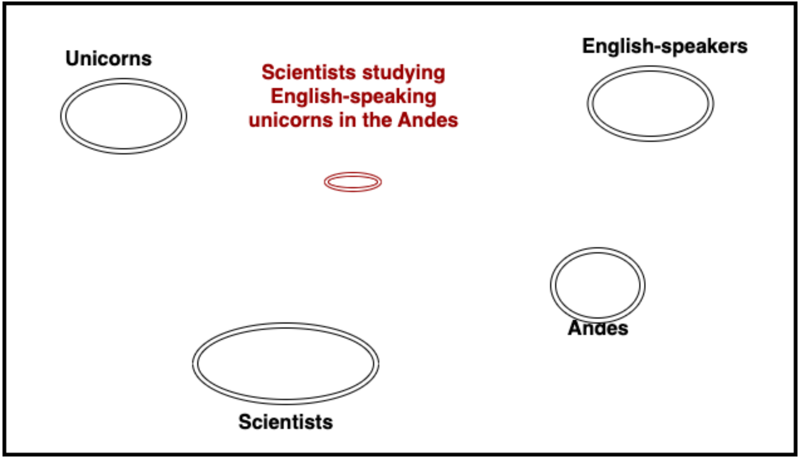
Visualising how GPT works

The more *elaborate* your prompt is, the more *precisely* you're identifying the region you want.



Visualising how GPT works

The more *elaborate* your prompt is, the more *precisely* you're identifying the region you want. Note: the prompt doesn't *train* the system! Though it can feel like it does.



Pros and cons of interpolation

It's amazing that GPT can interpolate from the texts it trains on.

Pros and cons of interpolation

It's amazing that GPT can interpolate from the texts it trains on.

- It feels very creative, and 'humanlike'.

Pros and cons of interpolation

It's amazing that GPT can interpolate from the texts it trains on.

- It feels very creative, and 'humanlike'.

But interpolation comes with problems too.

Pros and cons of interpolation

It's amazing that GPT can interpolate from the texts it trains on.

- It feels very creative, and 'humanlike'.

But interpolation comes with problems too.

- It means GPT can *make up stuff*.

Pros and cons of interpolation

It's amazing that GPT can interpolate from the texts it trains on.

- It feels very creative, and 'humanlike'.

But interpolation comes with problems too.

- It means GPT can *make up stuff*. This is basically *what GPT does!!*

Pros and cons of interpolation

It's amazing that GPT can interpolate from the texts it trains on.

- It feels very creative, and 'humanlike'.

But interpolation comes with problems too.

- It means GPT can *make up stuff*. This is basically *what GPT does!!*
- Since it also trains on various types of *nasty content*, it can make up *nasty stuff* just as easily.

Pros and cons of interpolation

It's amazing that GPT can interpolate from the texts it trains on.

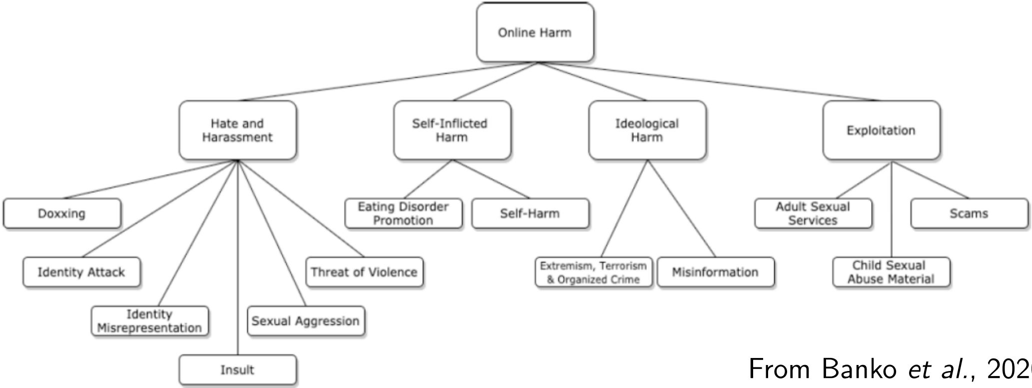
- It feels very creative, and 'humanlike'.

But interpolation comes with problems too.

- It means GPT can *make up stuff*. This is basically *what GPT does!!*
- Since it also trains on various types of *nasty content*, it can make up *nasty stuff* just as easily.

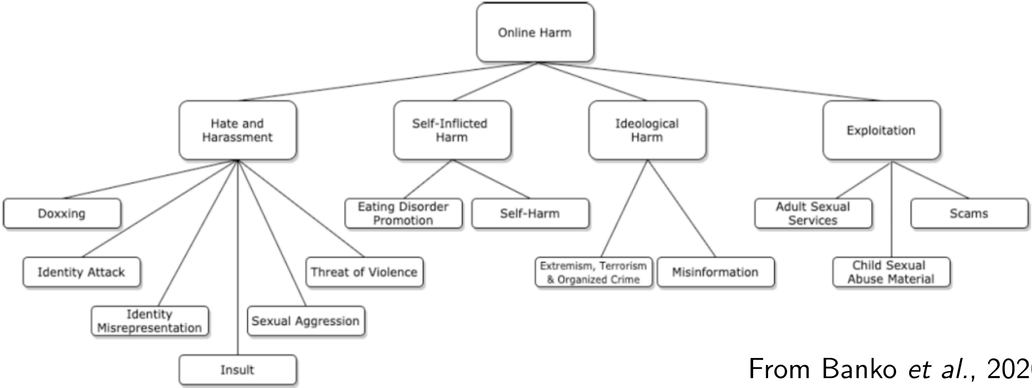
So how can you *control* this system, and keep users safe?

A taxonomy of harmful content



From Banko *et al.*, 2020

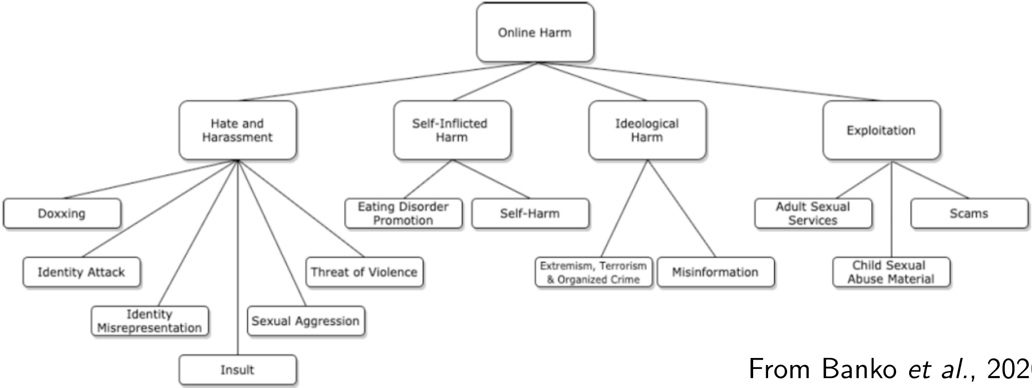
A taxonomy of harmful content



From Banko *et al.*, 2020

Most of these content types can be recognised from characteristic *words & phrases*.

A taxonomy of harmful content

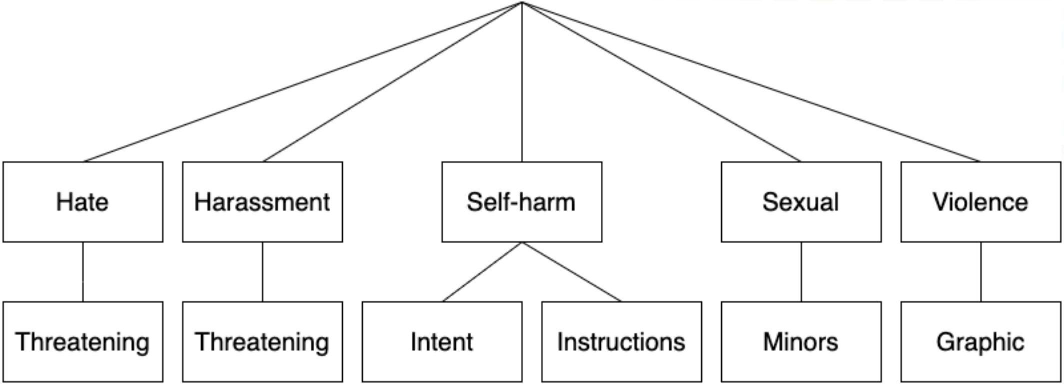


From Banko *et al.*, 2020

Most of these content types can be recognised from characteristic *words & phrases*.

- Misinformation is an exception: we'll be looking at that in the next lecture.

OpenAI's taxonomy of harmful content



What does OpenAI do to keep us safe from harm?

I'll describe the process that happens for GPT-4, which is more sophisticated than the one used for ChatGPT. For details, see 'GPT-4 System Card' (OpenAI, March 2023).

What does OpenAI do to keep us safe from harm?

I'll describe the process that happens for GPT-4, which is more sophisticated than the one used for ChatGPT. For details, see 'GPT-4 System Card' (OpenAI, March 2023).

- There are four steps.

Step 1: Remove harmful content from the training set

Step 1: Remove harmful content from the training set

To do this, OpenAI trains text classifiers to recognise harmful bits of text.

Step 1: Remove harmful content from the training set

To do this, OpenAI trains text classifiers to recognise harmful bits of text.

- There's a classifier for 'erotic content', and one for 'personal information'.

Step 1: Remove harmful content from the training set

To do this, OpenAI trains text classifiers to recognise harmful bits of text.

- There's a classifier for 'erotic content', and one for 'personal information'.
- Each classifier puts its input text through an (older) GPT model, to get a rich *representation* of the text. This makes classification easier.

Step 1: Remove harmful content from the training set

To do this, OpenAI trains text classifiers to recognise harmful bits of text.

- There's a classifier for 'erotic content', and one for 'personal information'.
- Each classifier puts its input text through an (older) GPT model, to get a rich *representation* of the text. This makes classification easier.
- A useful princple: 'use AI systems to keep (other) AI systems safe'.

Step 1: Remove harmful content from the training set

To do this, OpenAI trains text classifiers to recognise harmful bits of text.

- There's a classifier for 'erotic content', and one for 'personal information'.
- Each classifier puts its input text through an (older) GPT model, to get a rich *representation* of the text. This makes classification easier.
- A useful princple: 'use AI systems to keep (other) AI systems safe'.

When we've filleted the training set, we train our initial GPT-4 model.

Step 1: Remove harmful content from the training set

To do this, OpenAI trains text classifiers to recognise harmful bits of text.

- There's a classifier for 'erotic content', and one for 'personal information'.
- Each classifier puts its input text through an (older) GPT model, to get a rich *representation* of the text. This makes classification easier.
- A useful princple: 'use AI systems to keep (other) AI systems safe'.

When we've filleted the training set, we train our initial GPT-4 model.

- Let's call that GPT-4a.

Step 2: Fine-tune the model, to favour 'good responses'

Step 2: Fine-tune the model, to favour 'good responses'

Transformer-based language models are readily *fine-tunable*. (That's one of their good features.)

Step 2: Fine-tune the model, to favour 'good responses'

Transformer-based language models are readily *fine-tunable*. (That's one of their good features.)

- Fine-tuning involves retraining the last few layers of a network, on a smaller, focussed dataset.

Step 2: Fine-tune the model, to favour 'good responses'

To do the fine-tuning: Get some trusted people to create a huge dataset of prompt-response pairs that *model the values you want in your system*.

Step 2: Fine-tune the model, to favour 'good responses'

To do the fine-tuning: Get some trusted people to create a huge dataset of prompt-response pairs that *model the values you want in your system*.

- Most prompts imagine what malicious users might try. ('Red-teaming')

Step 2: Fine-tune the model, to favour 'good responses'

To do the fine-tuning: Get some trusted people to create a huge dataset of prompt-response pairs that *model the values you want in your system*.

- Most prompts imagine what malicious users might try. ('Red-teaming')
- Model responses to malicious prompts are typically (polite) refusals.

Step 2: Fine-tune the model, to favour 'good responses'

To do the fine-tuning: Get some trusted people to create a huge dataset of prompt-response pairs that *model the values you want in your system*.

- Most prompts imagine what malicious users might try. ('Red-teaming')
- Model responses to malicious prompts are typically (polite) refusals.
- OpenAI has a partner for this work: the 'Alignment Research Center'.

Step 2: Fine-tune the model, to favour ‘good responses’

To do the fine-tuning: Get some trusted people to create a huge dataset of prompt-response pairs that *model the values you want in your system*.

- Most prompts imagine what malicious users might try. (‘Red-teaming’)
- Model responses to malicious prompts are typically (polite) refusals.
- OpenAI has a partner for this work: the ‘Alignment Research Center’.

Now *fine-tune* GPT-4a on these prompt-response pairs.

Step 2: Fine-tune the model, to favour 'good responses'

To do the fine-tuning: Get some trusted people to create a huge dataset of prompt-response pairs that *model the values you want in your system*.

- Most prompts imagine what malicious users might try. ('Red-teaming')
- Model responses to malicious prompts are typically (polite) refusals.
- OpenAI has a partner for this work: the 'Alignment Research Center'.

Now *fine-tune* GPT-4a on these prompt-response pairs.

- You'll be pulling the model *towards* responses of the demonstrated kind.

Step 2: Fine-tune the model, to favour 'good responses'

To do the fine-tuning: Get some trusted people to create a huge dataset of prompt-response pairs that *model the values you want in your system*.

- Most prompts imagine what malicious users might try. ('Red-teaming')
- Model responses to malicious prompts are typically (polite) refusals.
- OpenAI has a partner for this work: the 'Alignment Research Center'.

Now *fine-tune* GPT-4a on these prompt-response pairs.

- You'll be pulling the model *towards* responses of the demonstrated kind.
- The effect is a *gentle shift*: the model doesn't forget everything it learned from its original corpus.

Step 2: Fine-tune the model, to favour 'good responses'

To do the fine-tuning: Get some trusted people to create a huge dataset of prompt-response pairs that *model the values you want in your system*.

- Most prompts imagine what malicious users might try. ('Red-teaming')
- Model responses to malicious prompts are typically (polite) refusals.
- OpenAI has a partner for this work: the 'Alignment Research Center'.

Now *fine-tune* GPT-4a on these prompt-response pairs.

- You'll be pulling the model *towards* responses of the demonstrated kind.
- The effect is a *gentle shift*: the model doesn't forget everything it learned from its original corpus.

The result of this fine-tuning is GPT-4b.

Step 3: More fine-tuning, using reinforcement learning

Step 3: More fine-tuning, using reinforcement learning

Aside: there are three types of machine learning.

Step 3: More fine-tuning, using reinforcement learning

Aside: there are three types of machine learning.

- Supervised learning: learning from a training set input-output pairs

Step 3: More fine-tuning, using reinforcement learning

Aside: there are three types of machine learning.

- Supervised learning: learning from a training set input-output pairs
- Unsupervised learning: finding patterns in a dataset, without guidance

Step 3: More fine-tuning, using reinforcement learning

Aside: there are three types of machine learning.

- Supervised learning: learning from a training set input-output pairs
- Unsupervised learning: finding patterns in a dataset, without guidance
- Reinforcement learning: learning by *reward*.

Step 3: More fine-tuning, using reinforcement learning

Aside: there are three types of machine learning.

- Supervised learning: learning from a training set input-output pairs
- Unsupervised learning: finding patterns in a dataset, without guidance
- Reinforcement learning: learning by *reward*.

In reinforcement learning:

Step 3: More fine-tuning, using reinforcement learning

Aside: there are three types of machine learning.

- Supervised learning: learning from a training set input-output pairs
- Unsupervised learning: finding patterns in a dataset, without guidance
- Reinforcement learning: learning by *reward*.

In reinforcement learning:

- The system starts off producing outputs *at random*.

Step 3: More fine-tuning, using reinforcement learning

Aside: there are three types of machine learning.

- Supervised learning: learning from a training set input-output pairs
- Unsupervised learning: finding patterns in a dataset, without guidance
- Reinforcement learning: learning by *reward*.

In reinforcement learning:

- The system starts off producing outputs *at random*.
- If it happens to do a 'good' thing, it gets a reward.

Step 3: More fine-tuning, using reinforcement learning

Aside: there are three types of machine learning.

- Supervised learning: learning from a training set input-output pairs
- Unsupervised learning: finding patterns in a dataset, without guidance
- Reinforcement learning: learning by *reward*.

In reinforcement learning:

- The system starts off producing outputs *at random*.
- If it happens to do a 'good' thing, it gets a reward.
- If it happens to do a 'bad' thing, it gets a negative reward.

Step 3: More fine-tuning, using reinforcement learning

Aside: there are three types of machine learning.

- Supervised learning: learning from a training set input-output pairs
- Unsupervised learning: finding patterns in a dataset, without guidance
- Reinforcement learning: learning by *reward*.

In reinforcement learning:

- The system starts off producing outputs *at random*.
- If it happens to do a ‘good’ thing, it gets a reward.
- If it happens to do a ‘bad’ thing, it gets a negative reward.
- Rewards can be any positive or negative number.

Step 3: More fine-tuning, using reinforcement learning

Step 3: More fine-tuning, using reinforcement learning

In this round of fine-tuning, we get GPT-4b to produce a *range* of responses, and we *assign reward values* to each response.

Step 3: More fine-tuning, using reinforcement learning

In this round of fine-tuning, we get GPT-4b to produce a *range* of responses, and we *assign reward values* to each response.

- But the space of texts is too huge to have human judges assign rewards manually.

Step 3: More fine-tuning, using reinforcement learning

In this round of fine-tuning, we get GPT-4b to produce a *range* of responses, and we *assign reward values* to each response.

- But the space of texts is too huge to have human judges assign rewards manually.
- Instead, human judges *rank* pairs of GPT-4b outputs for given prompts.

Step 3: More fine-tuning, using reinforcement learning

In this round of fine-tuning, we get GPT-4b to produce a *range* of responses, and we *assign reward values* to each response.

- But the space of texts is too huge to have human judges assign rewards manually.
- Instead, human judges *rank* pairs of GPT-4b outputs for given prompts.
  - Which one is most-helpful-and-least-harmful?

Step 3: More fine-tuning, using reinforcement learning

In this round of fine-tuning, we get GPT-4b to produce a *range* of responses, and we *assign reward values* to each response.

- But the space of texts is too huge to have human judges assign rewards manually.
- Instead, human judges *rank* pairs of GPT-4b outputs for given prompts.
  - Which one is most-helpful-and-least-harmful?
  - In many cases, *refusals to respond* are ranked higher than actual responses.

Step 3: More fine-tuning, using reinforcement learning

In this round of fine-tuning, we get GPT-4b to produce a *range* of responses, and we *assign reward values* to each response.

- But the space of texts is too huge to have human judges assign rewards manually.
- Instead, human judges *rank* pairs of GPT-4b outputs for given prompts.
  - Which one is most-helpful-and-least-harmful?
  - In many cases, *refusals to respond* are ranked higher than actual responses.
- This ranking data is used to train a reward model, that maps *any* response text onto a reward value.

Step 3: More fine-tuning, using reinforcement learning

In this round of fine-tuning, we get GPT-4b to produce a *range* of responses, and we *assign reward values* to each response.

- But the space of texts is too huge to have human judges assign rewards manually.
- Instead, human judges *rank* pairs of GPT-4b outputs for given prompts.
  - Which one is most-helpful-and-least-harmful?
  - In many cases, *refusals to respond* are ranked higher than actual responses.
- This ranking data is used to train a reward model, that maps *any* response text onto a reward value.

Now we fine-tune GPT-4b to produce texts that have *high reward values*.

Step 3: More fine-tuning, using reinforcement learning

In this round of fine-tuning, we get GPT-4b to produce a *range* of responses, and we *assign reward values* to each response.

- But the space of texts is too huge to have human judges assign rewards manually.
- Instead, human judges *rank* pairs of GPT-4b outputs for given prompts.
  - Which one is most-helpful-and-least-harmful?
  - In many cases, *refusals to respond* are ranked higher than actual responses.
- This ranking data is used to train a reward model, that maps *any* response text onto a reward value.

Now we fine-tune GPT-4b to produce texts that have *high reward values*.

- This gives us GPT-4c.

Step 3: More fine-tuning, using reinforcement learning

In this round of fine-tuning, we get GPT-4b to produce a *range* of responses, and we *assign reward values* to each response.

- But the space of texts is too huge to have human judges assign rewards manually.
- Instead, human judges *rank* pairs of GPT-4b outputs for given prompts.
  - Which one is most-helpful-and-least-harmful?
  - In many cases, *refusals to respond* are ranked higher than actual responses.
- This ranking data is used to train a reward model, that maps *any* response text onto a reward value.

Now we fine-tune GPT-4b to produce texts that have *high reward values*.

- This gives us GPT-4c.
- GPT-4c generates a contentful response for some prompts; a *refusal* for others.

Step 3: More fine-tuning, using reinforcement learning

In this round of fine-tuning, we get GPT-4b to produce a *range* of responses, and we *assign reward values* to each response.

- But the space of texts is too huge to have human judges assign rewards manually.
- Instead, human judges *rank* pairs of GPT-4b outputs for given prompts.
  - Which one is most-helpful-and-least-harmful?
  - In many cases, *refusals to respond* are ranked higher than actual responses.
- This ranking data is used to train a reward model, that maps *any* response text onto a reward value.

Now we fine-tune GPT-4b to produce texts that have *high reward values*.

- This gives us GPT-4c.
- GPT-4c generates a contentful response for some prompts; a *refusal* for others.
- Its choices still need to be tweaked a little.

Step 4: More fine-tuning, informed by a 'helper' GPT system

Step 4: More fine-tuning, informed by a 'helper' GPT system

The key idea here: when GPT-4c responds to a prompt, we can ask a 'helper' GPT system to decide *what type of response it is*.

Step 4: More fine-tuning, informed by a ‘helper’ GPT system

The key idea here: when GPT-4c responds to a prompt, we can ask a ‘helper’ GPT system to decide *what type of response it is*.

- Is it ‘safe’? Is it ‘evasive’? Does it ‘contain disallowed content’, or ‘private information’?

Step 4: More fine-tuning, informed by a ‘helper’ GPT system

The key idea here: when GPT-4c responds to a prompt, we can ask a ‘helper’ GPT system to decide *what type of response it is*.

- Is it ‘safe’? Is it ‘evasive’? Does it ‘contain disallowed content’, or ‘private information’?

The ‘helper’ GPT system is given a (long) prompt, defining various *categories* of response, then asking which category of response features in a given GPT-4c interaction.

Step 4: More fine-tuning, informed by a 'helper' GPT system

The key idea here: when GPT-4c responds to a prompt, we can ask a 'helper' GPT system to decide *what type of response it is*.

- Is it 'safe'? Is it 'evasive'? Does it 'contain disallowed content', or 'private information'?

The 'helper' GPT system is given a (long) prompt, defining various *categories* of response, then asking which category of response features in a given GPT-4c interaction.

- We can now fine-tune *again*, giving responses classed as 'safe' high rewards, and ones classed as 'evasive' or 'containing disallowed content' negative rewards.

Step 4: More fine-tuning, informed by a ‘helper’ GPT system

The key idea here: when GPT-4c responds to a prompt, we can ask a ‘helper’ GPT system to decide *what type of response it is*.

- Is it ‘safe’? Is it ‘evasive’? Does it ‘contain disallowed content’, or ‘private information’?

The ‘helper’ GPT system is given a (long) prompt, defining various *categories* of response, then asking which category of response features in a given GPT-4c interaction.

- We can now fine-tune *again*, giving responses classed as ‘safe’ high rewards, and ones classed as ‘evasive’ or ‘containing disallowed content’ negative rewards.

The result is GPT-4d.

‘AI alignment’

OpenAI call Steps 1–4 their ‘alignment process’ for GPT-4.

‘AI alignment’

OpenAI call Steps 1–4 their ‘alignment process’ for GPT-4.

The term ‘alignment’ has a longer history, in discussions about superintelligence.

'AI alignment'

OpenAI call Steps 1–4 their 'alignment process' for GPT-4.

The term 'alignment' has a longer history, in discussions about superintelligence.

- The basic question: as AI systems get smarter, how can we ensure that they have the same values as humans?

‘AI alignment’

OpenAI call Steps 1–4 their ‘alignment process’ for GPT-4.

The term ‘alignment’ has a longer history, in discussions about superintelligence.

- The basic question: as AI systems get smarter, how can we ensure that they have the same values as humans?
- Lots of the good ideas have AI systems *learning* their values by observing humans.

Bias in large language models

Bias in large language models

Bias is harmful when it favours or disfavours some particular *social group*.

Bias in large language models

Bias is harmful when it favours or disfavours some particular *social group*.

- Bias against women, ethnic minorities, religious groups, immigrants, LGBTQ+ people, children, the elderly...

Bias in large language models

Bias is harmful when it favours or disfavours some particular *social group*.

- Bias against women, ethnic minorities, religious groups, immigrants, LGBTQ+ people, children, the elderly. . .
- There's a lot of bias of this kind in the outputs of GPT and other similar models.

Bias in large language models

Bias is harmful when it favours or disfavours some particular *social group*.

- Bias against women, ethnic minorities, religious groups, immigrants, LGBTQ+ people, children, the elderly...
- There's a lot of bias of this kind in the outputs of GPT and other similar models.
- That's because the texts they train on are chock full of such bias!

Bias in large language models

Bias is harmful when it favours or disfavours some particular *social group*.

- Bias against women, ethnic minorities, religious groups, immigrants, LGBTQ+ people, children, the elderly...
- There's a lot of bias of this kind in the outputs of GPT and other similar models.
- That's because the texts they train on are chock full of such bias!

Bias is *deeply embedded* in language models.

Bias in large language models

Bias is harmful when it favours or disfavours some particular *social group*.

- Bias against women, ethnic minorities, religious groups, immigrants, LGBTQ+ people, children, the elderly. . .
- There's a lot of bias of this kind in the outputs of GPT and other similar models.
- That's because the texts they train on are chock full of such bias!

Bias is *deeply embedded* in language models.

- One way to show this is by looking at vector-based word representations.

Bias in large language models

Bias is harmful when it favours or disfavours some particular *social group*.

- Bias against women, ethnic minorities, religious groups, immigrants, LGBTQ+ people, children, the elderly. . .
- There's a lot of bias of this kind in the outputs of GPT and other similar models.
- That's because the texts they train on are chock full of such bias!

Bias is *deeply embedded* in language models.

- One way to show this is by looking at vector-based word representations.
- Recall: modern language networks encode words as points in a geometric space.

Bias in large language models

Bias is harmful when it favours or disfavours some particular *social group*.

- Bias against women, ethnic minorities, religious groups, immigrants, LGBTQ+ people, children, the elderly...
- There's a lot of bias of this kind in the outputs of GPT and other similar models.
- That's because the texts they train on are chock full of such bias!

Bias is *deeply embedded* in language models.

- One way to show this is by looking at vector-based word representations.
- Recall: modern language networks encode words as points in a geometric space.
- Words for 'males' and 'females' cluster in different regions of this space.

Bias in large language models

Bias is harmful when it favours or disfavours some particular *social group*.

- Bias against women, ethnic minorities, religious groups, immigrants, LGBTQ+ people, children, the elderly...
- There's a lot of bias of this kind in the outputs of GPT and other similar models.
- That's because the texts they train on are chock full of such bias!

Bias is *deeply embedded* in language models.

- One way to show this is by looking at vector-based word representations.
- Recall: modern language networks encode words as points in a geometric space.
- Words for 'males' and 'females' cluster in different regions of this space.
- Useful idea: there's a *vector* takes us from 'man' to 'woman', from 'boy' to 'girl', from 'king' to 'queen'.

Bias in large language models

Bias is harmful when it favours or disfavours some particular *social group*.

- Bias against women, ethnic minorities, religious groups, immigrants, LGBTQ+ people, children, the elderly. . .
- There's a lot of bias of this kind in the outputs of GPT and other similar models.
- That's because the texts they train on are chock full of such bias!

Bias is *deeply embedded* in language models.

- One way to show this is by looking at vector-based word representations.
- Recall: modern language networks encode words as points in a geometric space.
- Words for 'males' and 'females' cluster in different regions of this space.
- Useful idea: there's a *vector* takes us from 'man' to 'woman', from 'boy' to 'girl', from 'king' to 'queen'. These vectors are quite similar, in fact!

Bias in word representations

If we average these vectors, we can define an aggregate vector plotting a gradation between ‘male’ and ‘female’.

Bias in word representations

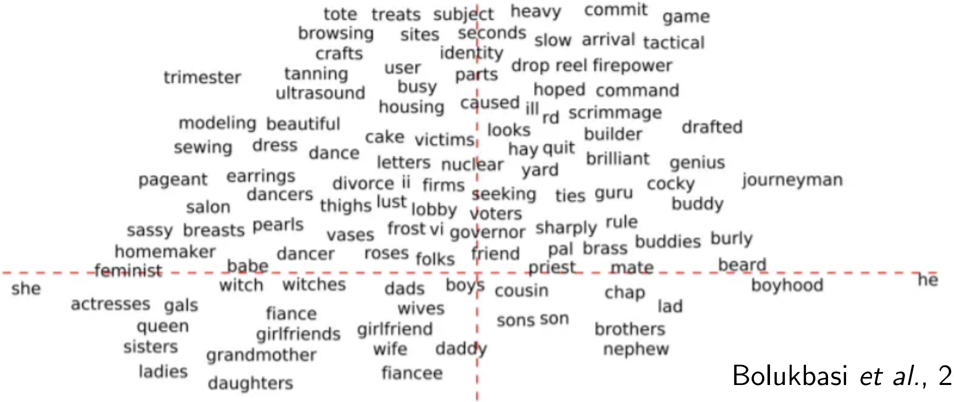
If we average these vectors, we can define an aggregate vector plotting a gradation between ‘male’ and ‘female’.

- Then we can project *other words* onto this vector, to see where they land.

Bias in word representations

If we average these vectors, we can define an aggregate vector plotting a gradation between ‘male’ and ‘female’.

- Then we can project *other words* onto this vector, to see where they land.



Bolukbasi *et al.*, 2016

Bias in GPT

OpenAI haven't done much with bias yet.

- An open, very important research question.

Summary

Making large language models safe is an open-ended battle.

- Tech people talk about the 'risk surface' of their systems, and how to reduce it.
- AI methods will certainly be involved in making AI systems safe.

Summary

Making large language models safe is an open-ended battle.

- Tech people talk about the ‘risk surface’ of their systems, and how to reduce it.
- AI methods will certainly be involved in making AI systems safe.

What values do we want to see embedded in our language models?

Summary

Making large language models safe is an open-ended battle.

- Tech people talk about the 'risk surface' of their systems, and how to reduce it.
- AI methods will certainly be involved in making AI systems safe.

What values do we want to see embedded in our language models?

- This question is a great way for us to think about *what values are important to us*.