Constraining Claude

TECH, TEA + EXCHANGE, Tate, 2025

bit.ly/4k4KeQX

Nathan Bayliss, Robin Leverton, Jérémie Wenger

Claude

Claude will return soon

Claude.ai is currently experiencing a temporary service disruption.
We’re working on it, please check back soon.

ANTHROP\C

Constraining Claude: Plan of Action

1. Introductions
2. Know thy tool: LLM literacy
3. Constraints
    a) Working with text
    b) Working images

bit.ly/4k4KeQX

Who we are?

Nathan Bayliss (site, @_nate_bliss_)

Robin Leverton (site, @robinleverton)

Jérémie Wenger (site, @jchwenger)

bit.ly/4k4KeQX

About me

Jérémie Wenger (he/him)
Writing, programming, machine learning
[@jchwenger](#)
[jeremiewenger.com](#)

Writing with LLMs (fiction, poetry):
[Artificial It](#), Vegans, [Lacanage](#)
[Inanimaux](#), Alors seulement le rêve,
[Départs](#)

Writing with constraints:
[Chains](#), [Squares](#), [Cubes](#), Subwords

```
Le.a.x Manufacteur.rice.x: LISA.
C'est moi qui parle, je veux
dire, c'est vraiment moi
(montrant l'écran du fond) mais
toi, tu vois le corps de Lisa, et
tu ne vois pas le corps de Missy.
Elle est juste là, à côté de
moi...

Elsa: LUCAS. Oh là là.
D'exclamation ah non mais là
c'est vraiment tu es vraiment
dedans ? Je veux dire c'est
```

photo : Emile Zeizig

[CHATBOT](#), GPT-2 bot for the stage

Workshop resources: Residency Repo

github.com/jchwenger/TECH-TEA-EXCHANGE

github.com/jchwenger/p5.Claude

Maintained by Robin, Nathan and myself.

To code with Claude, get your API key: console.anthropic.com
(not free even for PRO users, email Greg to get credits *before Friday*!)

Expect updates! Recommended workflow: fork it, and keep the original version as an upstream remote that you can pull from.

Workshop resources: More Resources

Documentation: docs.anthropic.com (glossary)

Course: Building toward Computer Use with Anthropic

Code repos:

github.com/anthropics/courses
github.com/anthropics/anthropic-cookbook
github.com/anthropics/anthropic-quickstarts
github.com/anthropics/prompt-eng-interactive-tutorial

Γνῶθι τὸ σὸν ὄργανον

(gnôthi to son organon: “know thy tool”, ref)

Γνῶθι τὸ σὸν ὄργανον: Capabilities

Andrej Karpathy, one of the leading researchers in the fields. See also this interesting take here.

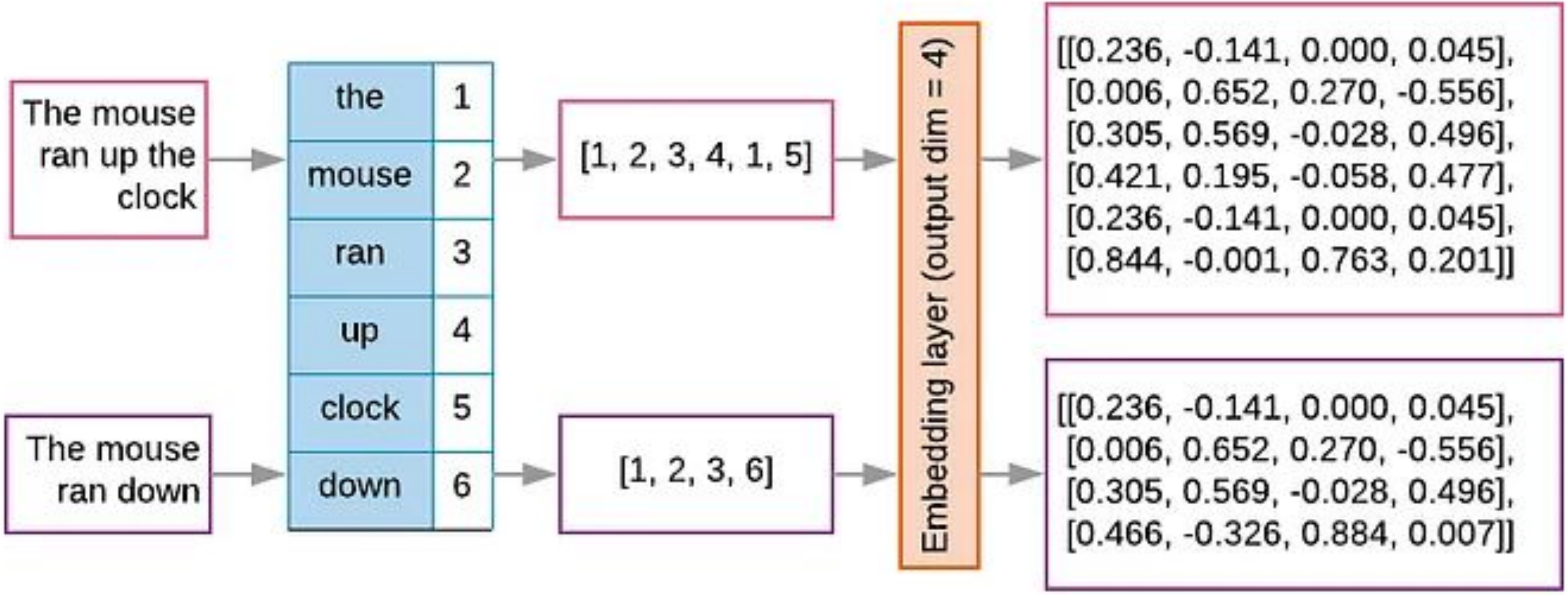
- Can produce **text** (and code).
- Recent addition:
  - ‘Artifacts’ (uneditable 🙄 code, running directly in the WebUI)
  - Web search
- **Multimodal**: accepts image input.
- Does **not generate** images!

Note: longer conversations can lead to performance degradation in LLMs, often starting again from scratch is the way to go (or exploit degradation itself!).

Note: **context** = **prompt** = **prefix**: the tokens taken into account to make the prediction.

Γνῶθι τὸ σὸν ὄργανον: Tokenisation 101

ref



The mouse ran up the clock

the	1
mouse	2
ran	3
up	4
clock	5
down	6

[1, 2, 3, 4, 1, 5]

Embedding layer (output dim = 4)

```
[[0.236, -0.141, 0.000, 0.045],
 [0.006, 0.652, 0.270, -0.556],
 [0.305, 0.569, -0.028, 0.496],
 [0.421, 0.195, -0.058, 0.477],
 [0.236, -0.141, 0.000, 0.045],
 [0.844, -0.001, 0.763, 0.201]]
```

The mouse ran down

[1, 2, 3, 6]

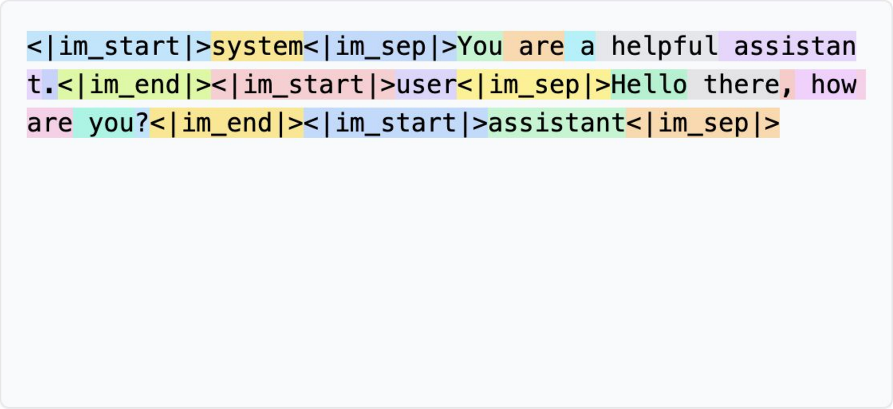
```
[[0.236, -0.141, 0.000, 0.045],
 [0.006, 0.652, 0.270, -0.556],
 [0.305, 0.569, -0.028, 0.496],
 [0.466, -0.326, 0.884, 0.007]]
```

Γνῶθι τὸ σὸν ὄργανον : Tokenisation

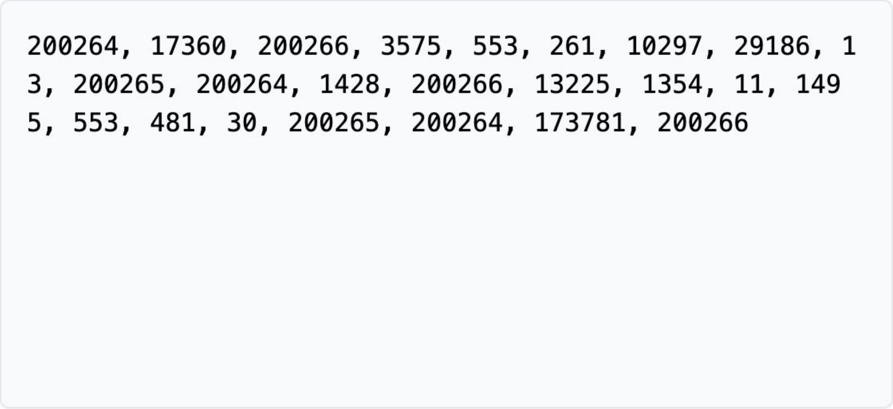
The model takes in integers only (tokens), but we are free to use those as we like:

Want a token that internally means “end of a sentence”, “end of a document”, “stop generation and call a Python function”? All possible.

See the Tiktokenizer apps to gain a feel of this, also the Claude Tokenizer. (Code: Karpathy tutorial, Huggingface tutorial.)



```
<|im_start|>system<|im_sep|>You are a helpful assistan
t.<|im_end|><|im_start|>user<|im_sep|>Hello there, how
are you?<|im_end|><|im_start|>assistant<|im_sep|>
```



```
200264, 17360, 200266, 3575, 553, 261, 10297, 29186, 1
3, 200265, 200264, 1428, 200266, 13225, 1354, 11, 149
5, 553, 481, 30, 200265, 200264, 173781, 200266
```

Γνῶθι τὸ σὸν ὄργανον: Beware Tokenisation!

Beware, tokenization (breaking down language into a fixed-sized vocabulary of ‘tokens’) plays a huge part in how LLMs work!

If your LLM reads ‘tokenisation’ as ‘token’ + ‘isation’, which themselves are just integers, 3401 + 734, asking it to use words with the same number of letters ***is literally incomprehensible***. Like asking a blind person to find objects of the same colour that they have in their hand!

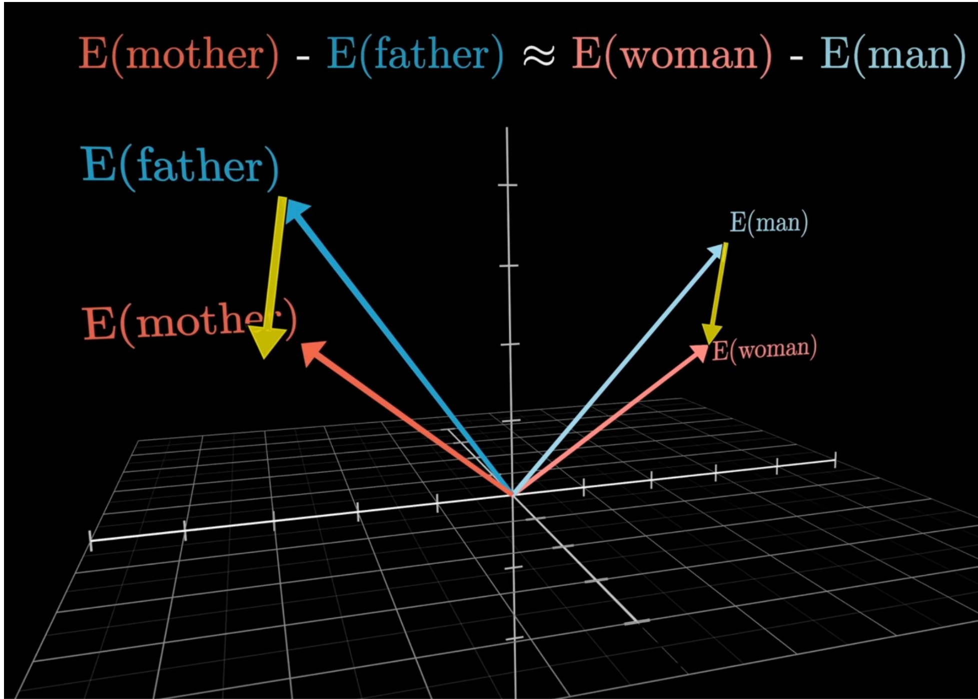
(Current LLMs are so flexible they still find ways to perform reasonably at this.)

Γνῶθι τὸ σὸν ὄργανον: word vectors / embeddings

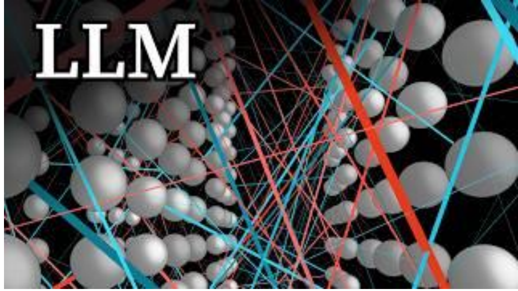
(ref, whole playlist)

Why are vectors useful?

- The placement of word (tokens) in the space is *learnt* by the model.
- Relationships between concepts can be encoded in the space (this *emerges* during training).



Γνῶθι τὸ σὸν ὄργανον: attention

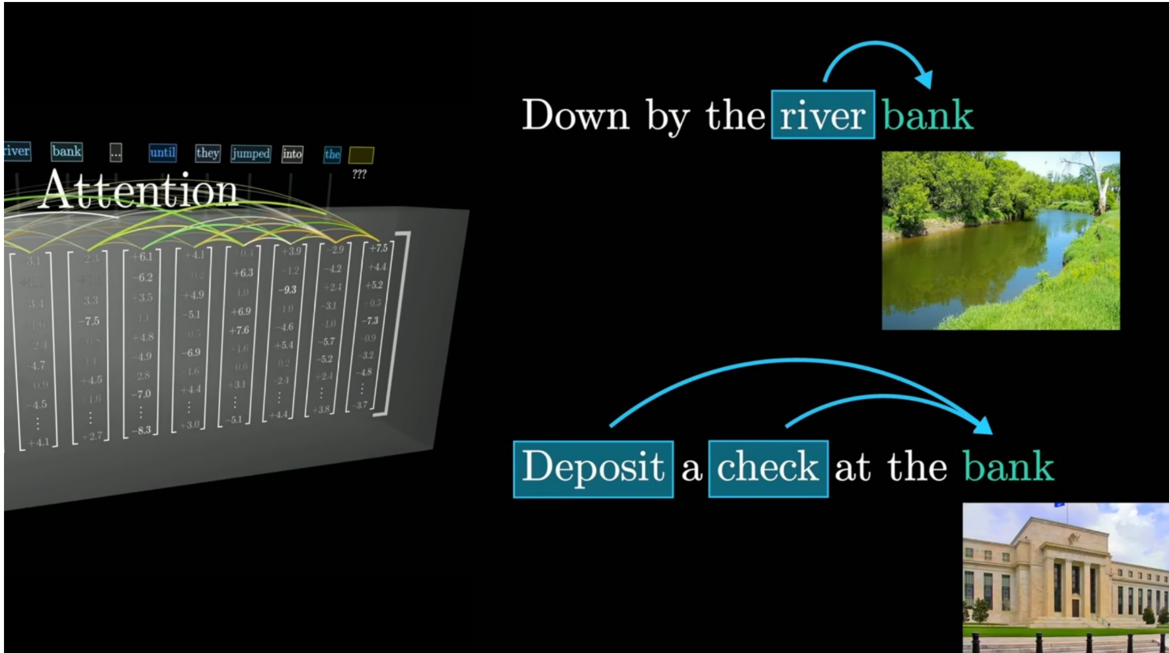


(ref, whole playlist)

Why are vectors useful?

- Current LLMs are stacks of transformation layers that, in effect, *readjust the coordinates of the token vectors according to the context.*

(Much of what they actually do is still unknown, see this.)

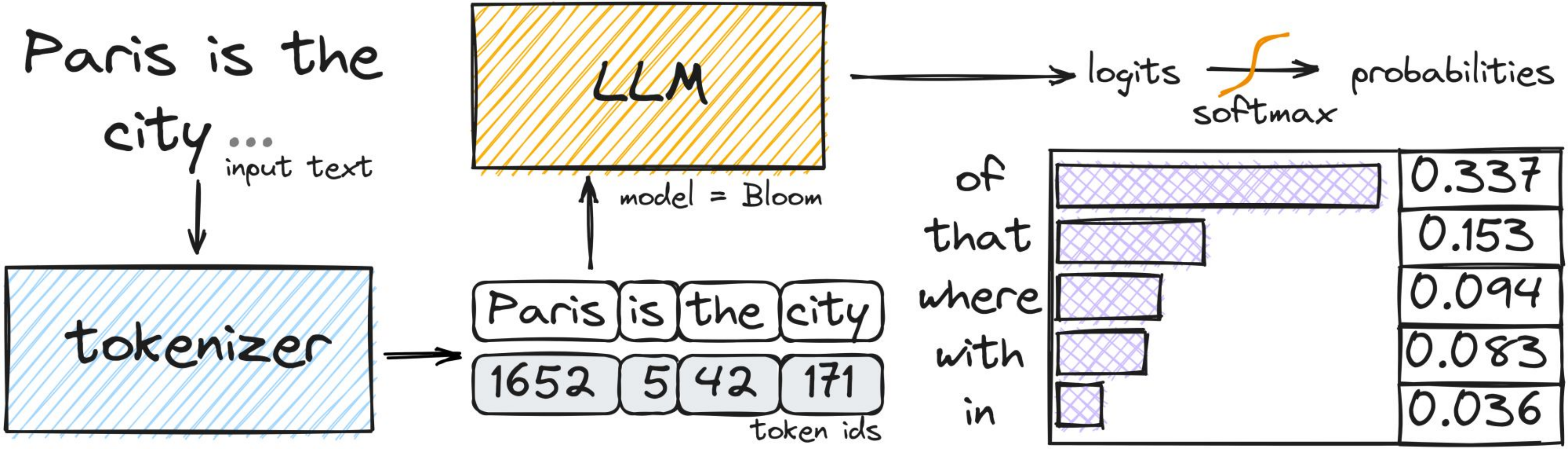


Γνῶθι τὸ σὸν ὄργανον: next-token predictors

LLMs try to predict the **probabilities of the next token**: like a gambler in a casino, who has seen 10 cards, wants to computes how likely each card is to come next.

In Machine Learning parlance, this is called a **classification problem.**

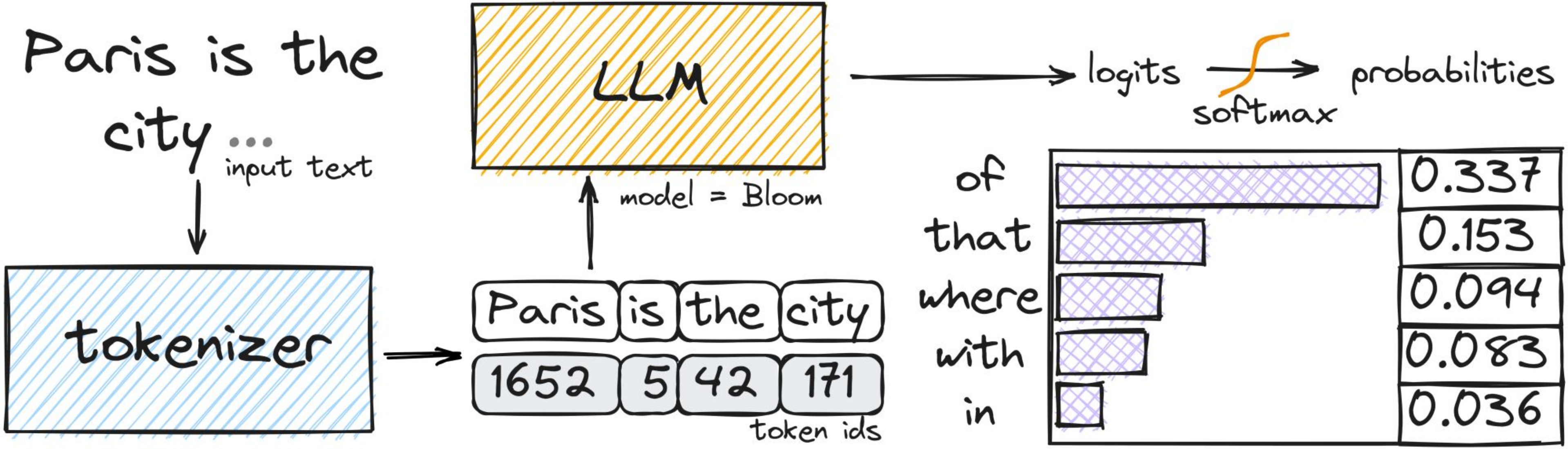
ref



Γνῶθι τὸ σὸν ὄργανον: next-token predictors

Note: in this setting the model has a **finite** number of possibilities to choose from (to assign probabilities to). These possibilities are the **vocabulary** of the model.

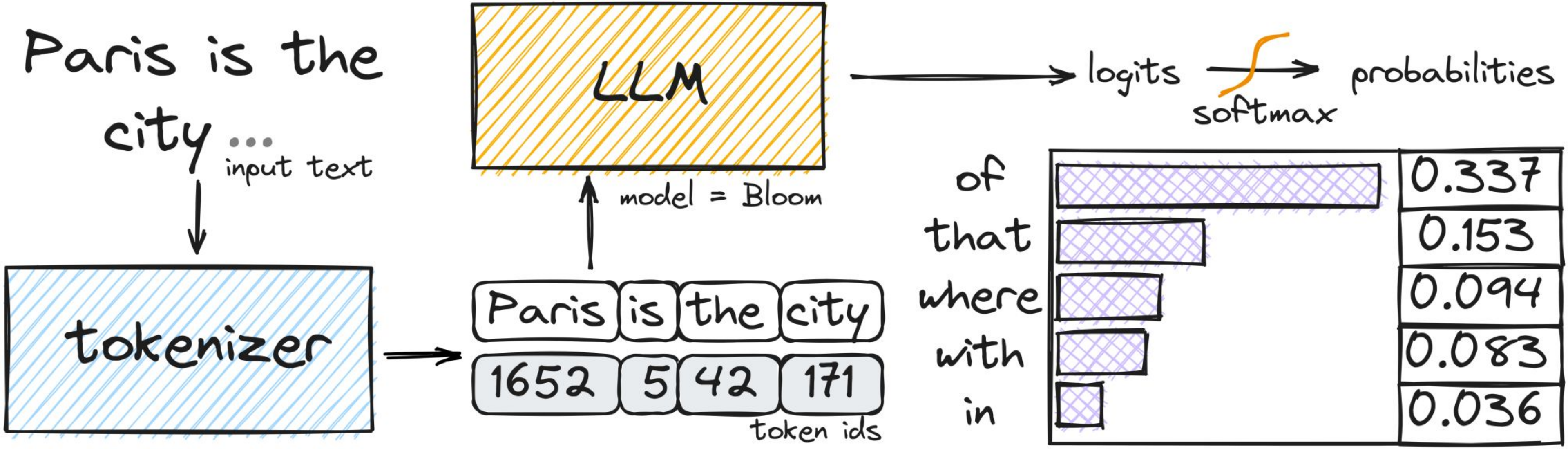
(If you include single letters or bytes in this, then the model can still write anything. ref



Γνῶθι τὸ σὸν ὄργανον: next-token predictors

What about processing **images**? It turns out you can split your image into **patches**, and treat each patch as a *token*, turn that into yet another vector, and that works (after 'translation', the two types of data are the same).

ref



Γνῶθι τὸ σὸν ὄργανον: sampling (inject randomness)

Having a probability distribution over next tokens is a bit like having in your hand a **loaded die** 🎲 that falls more often on one side than the other.

But you still need to *throw the die* for it to work, that is called **sampling**.

This is also how the models are trained: guessing the next token millions of time!

[input]		[output]
The quick brown _	←	we throw the die, get ‘fox’
The quick brown fox _	←	again, we get ‘jumps’
The quick brown fox jumps _	←	again, we get ‘over’
The quick brown fox jumps over _	←	etc.

This iterative process is what’s called ‘autoregressive’!

Γνῶθι τὸ σὸν ὄργανον: `temperature`

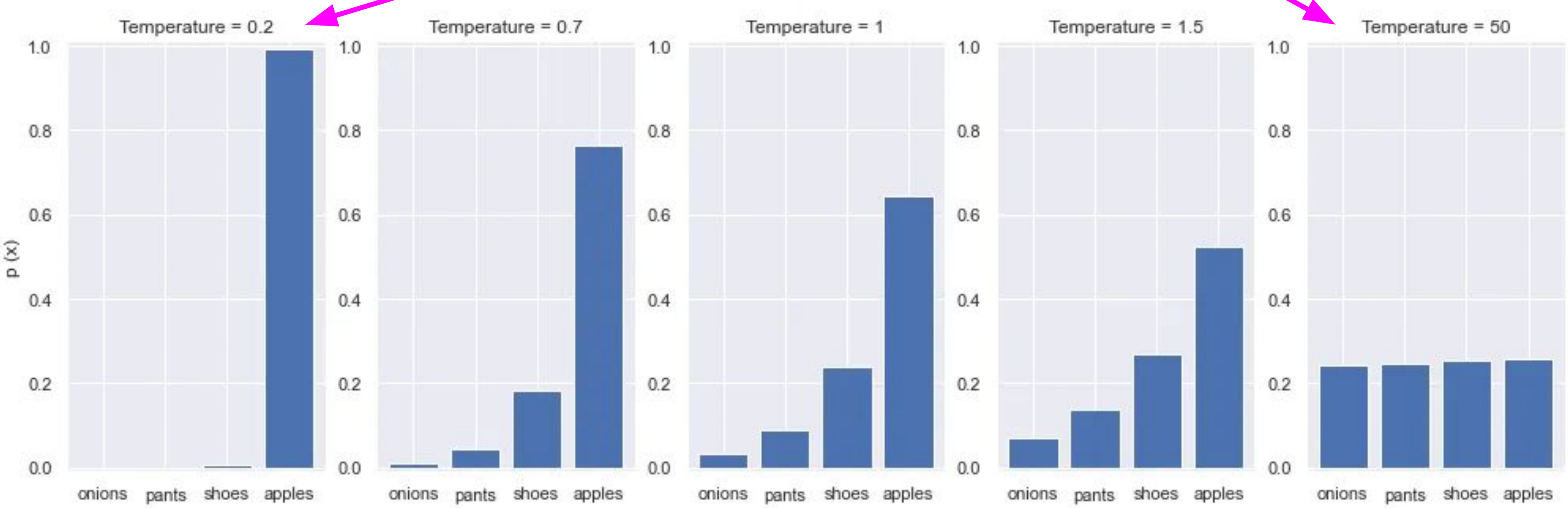
(Why do companies limit the temperature? To prevent you from ‘breaking’ the model: making it output gibberish, **bad** for publicity!)

Low temperature: ‘apples’ is SO likely, it’s almost certain to be picked (“more predictable”)

High temperature: ‘apple’ is still the likeliest, but by so little, the choice is almost random (“unpredictable”)

"I like red ___"

ref



Γνῶθι τὸ σὸν ὄργανον: Prompts, Examples

Prompting: means ‘how you formulate requests’, or ‘how to ask things to llms to get results you want’, has grown to be an independent area of research. This could be the focus of a project. See this and this for resources.

If you recall that LLMs are trained on millions of documents containing all sorts of information, ‘prompting’ is nothing but *creating the beginning of a document that is **likely** to end in the way we like* (i.e.: contain the answer that we are after).

Γνῶθι τὸ σὸν ὄργανον: Prompts, Examples

Giving **examples of what you want** is often extremely effective (called “few-shot prompting” or “in-context learning”). In Claude, you find this in “Choose style”.

But also, remember: starting again from scratch (a new chat) is often better than trying to course correct a conversation.

Note: a **system prompt** is a set of instructions that is ‘permanent’ (for the whole chat). But fundamentally this is just the top of your likely document, perhaps with some “Careful! Don’t Forget! Super important!” added around it under the hood.

Example to try this: ask the LLM to speak in another language, or ask it to play a role (“you are a teacher…”).

Γνῶθι τὸ σὸν ὄργανον: Prompts

LLMs have become flexible enough to ‘understand’ commands, therefore you can come up with quite specific indications, for example:

> Here’s a famous line of poetry:
>
> “Shall I compare thee to a summer’s Day”
>
> Please write ten variations replacing the markers [V] (verb) [N] (noun) below:
>
> “Shall I [V] thee to a [N]’s [N]”
>
> The lines should be as varied as possible, you can be weird if you want!

Keeping tokenisation in mind helps: using ‘[V]’ is unusual, -> easily recognised!

Γνῶθι τὸ σὸν ὄργανον: AI Slop & Training

Average-ness of the output – by design: autoregressive pre-training: guess the next token. Any divergence (of writing, thinking) from the human corpus is punished (high loss).

The main issue is that we (the scientific community) don’t know what other **learning objective** to set up!

Other form of learning, AlphaGo: pre-defined set of rule, then mixture of search and RL. This leads to true ‘*non-human*’ results (found because of existing **rules of the game**, not because humans have done it before – note: this is how human discovery tends to work as well!).

Γνῶθι τὸ σὸν ὄργανον: ‘RLHF’, ‘Alignment’

‘**Aligning**’ means using a special class of algorithms, Reinforcement Learning (with Human Feedback, RLHF) to refine the LLM behaviour.

What for? In the ‘base model’, the beginning of a chat could be many things:

`Hey, how are you[? - I’m well thanks, and you?]`

`Hey, how are you[? I’ve tried to call you earlier!]`

`Hey, how are you[?” It was a relief to hear that voice.]`

But people (companies) want chatbots most of all! The additional training (on conversational documents) is meant to make sure the first option is preferred.

Γνῶθι τὸ σὸν ὄργανον: ‘RLHF’, ‘Alignment’

Alignment can then take an even subtler meaning.

The second, even more more high-level refining is to try and make **certain kinds of conversations more likely than others**.

Example already seen: make sure that when someone asks how to make bioweapons at home, the **very, very likely** continuation produced will be one that does not divulge that kind of information.

Γνῶθι τὸ σὸν ὄργανον: ‘RLHF’, ‘Alignment’

Reinforcement Learning means that on top of having a direct **learning signal** (penalty/reward) when predicting the next token, you assign a reward for the **entire** text (conversation).

This uses human outputs as a ‘benchmark’ (companies choosing which moral values, or behaviours, are rewarded/punished, with socio-political consequences) but also uses **real humans to annotate thousands and thousands of documents** (a **lot** of labour) .

Note: a ‘like’ button under a reply in the interface invites you to give your labour to the company by ***labelling*** it (saying “for that answer, reward!”). Similar ‘crowdsourcing’ process as CAPCHAs for computer vision.

Γνῶθι τὸ σὸν ὄργανον: ‘Chain of Thought’

Weird effects in **very** large language models (seeming emergence of ‘reasoning’ abilities?, here p. 22).

One key example: “we show that LLMs are decent zero-shot reasoners by simply adding "Let's think step by step" before each answer” (then suddenly the quality of answers get better).

That led to much more elaborate prompts: Chain-of-Thought, where tasks would be decomposed into steps, substeps, where some counter questions would be asked (“Was this correct? Let’s double check the previous result, etc.”). It was shown that increasing the ‘amount of reasoning’ increases the quality of the outputs. (It’s still just a document, the LLM still predicts the next step.)

Γνῶθι τὸ σὸν ὄργανον: Multilingualism

Various multilingual capabilities depending on LLMs.

Enormous English dominance in Internet datasets.

Important: the tokenisation process is not universal: if your tokenisation is optimised for English (which it is), your model will tend to work better in English.

Questions around ‘low-resource’ languages: most languages in the world have very little presence on the internet! (Interestingly, multilingual training improves performance including in lower-resource languages.)

Claude’s Multilingual support page
“How do Large Language Models Handle Multilingualism?”
“A Survey on Large Language Models with Multilingualism: Recent Advances and New Frontiers”

Γνῶθι τὸ σὸν ὄργανον: Ethics

Main elements to keep in mind:

- LLMs prediction involves **massive amounts of compute**. Yes, it’s not like driving a car, but *in digital terms* (compared to opening Wiki, or, imagine!, reading a physical book), it’s *massively* more energetically expensive (like taking an SUV to go buy a salad in the supermarket 5 minutes away).
- LLM training involves **humongous amounts of labour** (to create the datasets, and to refine training and label data, often in very poor labour conditions).
- Because they are basically trained on the Internet, they inherit all the **biases** found there (and often amplify them).

Constraints

Constraints: What?

Constraints are a **set of rules**, **formulated in advance**, and ***chosen*** by the writer.

“[T]he Oulipo’s methods are *rational, conscious and logical*, while the Surrealists prized the non- or irrational, unconscious or unaccountable as a revelatory source. Second, constraint, as the *self-imposed adherence to (unnecessary) logical forms*, is the usual definition of Oulipian work, while for the Surrealists, the lack of constraint, absolute freedom—a strongly Romantic concept—is sought. Third, since Oulipian methods are not necessary, they must be *chosen voluntarily*, while Surrealist automatism involves a state of passivity before chance or the unknown.” (Daniel Cartwright, *The Oulipo and Modernism: Literature, Craft and Mathematical Form*, PhD thesis, p. 45, my emphasis)

Constraints: What?

Constraints are a **set of rules**, **formulated in advance**, and ***chosen*** by the writer.

Contrary to what we have seen in the presentation, instead of approaching an LLM by asking ourselves a task X can be made easier by using it, or done by the LLM instead of by ourselves, we wonder: “Hm, doing X is quite easy… Is there a way we could make X more difficult?”

A constraint is a way of giving a **challenge** to yourself, that you then strive to accomplish.

Constraints: What?

Note: a well-known parallel approach (also influenced by the Second Viennese School) emerged around John Cage and Fluxus:

“In all of his work with chance, Cage sought a balance between the rational and the irrational by allowing random events to function within the context of a controlled system.” (Marc G. Jensen, “John Cage, Chance Operations, and the Chaos Game: Cage and the ‘I Ching’”, The Musical Times, Vol. 150, No. 1907 (Summer, 2009), pp. 97-102)

Here, the key difference is the interaction of the system with **randomness**.

Constraints: (Mini-)History

Poetic or musical forms as constraints (the sonnet, the rondo, the sonata)...

Second Viennese School: Serialism (1910s, WWI, and beyond)

Schönberg, Berg, Webern. Constraint: chosen ordering for all twelve semitones. Political & social significance: resistance to kitsch.

The Oulipo (1940s, WWII, and beyond)

(Ouvroir de Littérature Potentielle, “Workshop of Potential Literature”)

Raymond Queneau, Georges Perec, Jacques Roubaud, Michèle Audin, Anne Garréta, many others.

Constrained literature: examples

Raymond Queneau. *Cent mille milliards de poèmes.* Gallimard, 1961. (*One hundred billion poems*) Various translations available, see here, as well as this interactive version.

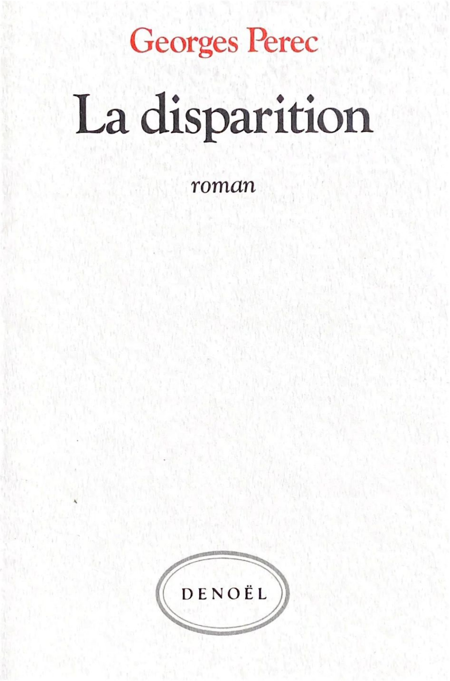
Queneau wrote 10 possibilities for each of the 14 lines of a sonnet, all compatible with each other: 10^{14} possibilities. Here, the traditional structure of the sonnet, as well as rules of syntax (compatibility between snippets), act as a constraint.

Constrained literature: examples

See also:
[Harry Matthews' variations]

Georges Perec. *La disparition.* Denoël, 1969. Translation by Gilbert Adair. *A Void.* Vintage, 1994.

In *La disparition*, the most common letter in French, 'e', never occurs. Note: it is required to write 'Georges Perec', whose entire family perished in death camps, and characters in the novel keep seeking for 'something' that they cannot name.



Georges Perec

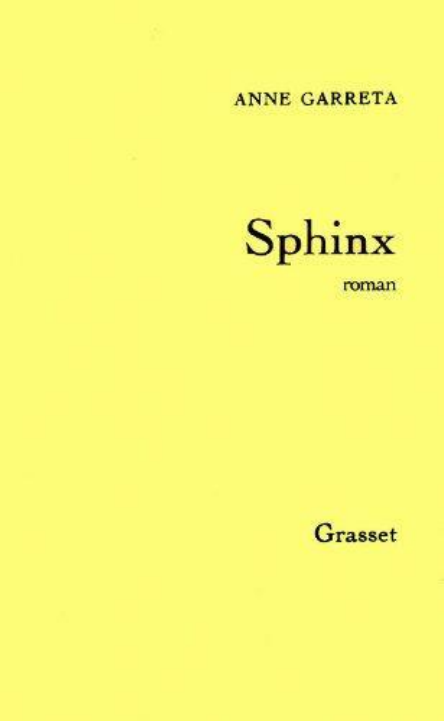
La disparition

roman

DENOËL

Anne Garréta. *Sphinx*. Grasset, 1986. Translation available by Emma Ramadan. Deep Vellum, 2015.

In *Sphinx*, the gender of the two main characters is kept unknown (very difficult in French, which has gender inflections everywhere!).



ANNE GARRETA

Sphinx

roman

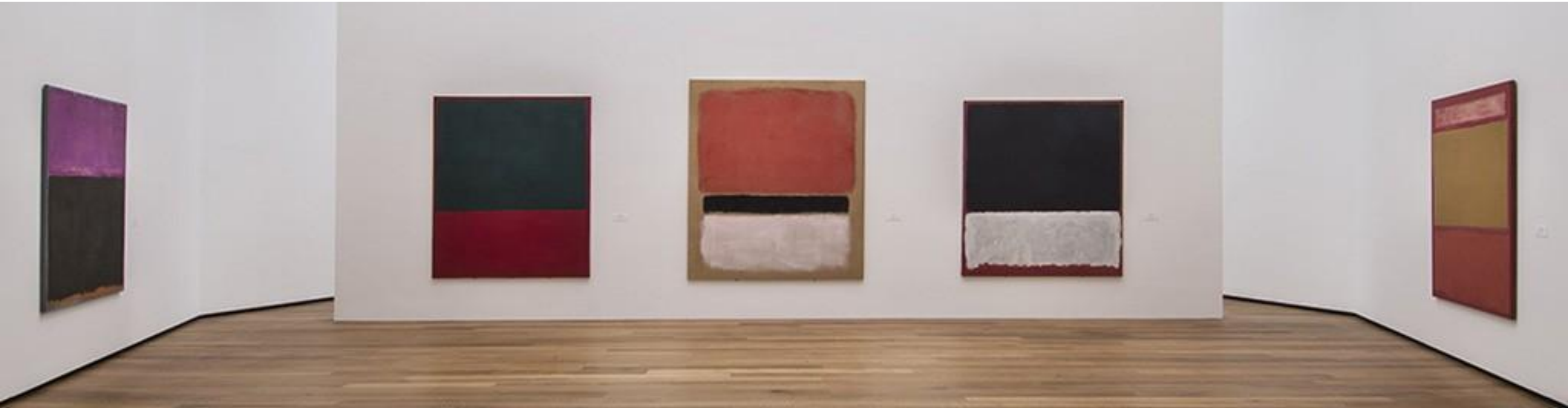
Grasset

Constraints and Series

Can we view the concept of artistic series as a form of constraint?

Example: Mark Rothko

Source



Constraint creation: an ‘ABA’ algorithm

Algorithm for constraint creation (creating variations)

1. (A: art practice) Pick an existing corpus of artworks, or series
2. (B: formulation) Formulate the underlying constraint
3. (A: art practice) Create a new series that follows the *constraint* to the letter (but can diverge from the original series: especially if the formulation of the constraint can be *creatively misread*.)
4. [optional:] Go back to 2, based on corpus made in 3, repeat!

Constraint creation?

Just as interesting as its application: **constraint creation**.

Applying an arbitrary rule can lead to arbitrary (and uninteresting) work! The key question could be the **foundation** for, and **status** of, the constraint? What does the constraint mean **to you**?

Indeed, restricting possibilities can often be a **catalyst**, a **spur**, for **imagination**!

Are there rules for creating interesting constraints? This is certainly an NP-hard problem (which doesn’t mean you cannot do it!).

Feel free to come up with your own constraints in the exercises today!

Workshop: Constraints

Possible to work with someone or in a group!

Two directions available:

2. a) Art-centric: find a work or series of works (it can even be your own) that can be viewed as governed by a constraint. Try and apply the ABA algorithm to it.

2. b) Constraint-centric: can you think of a constraint of that you might be interested in a priori, in the abstract? Or can you come up with one that would be meaningful to you?

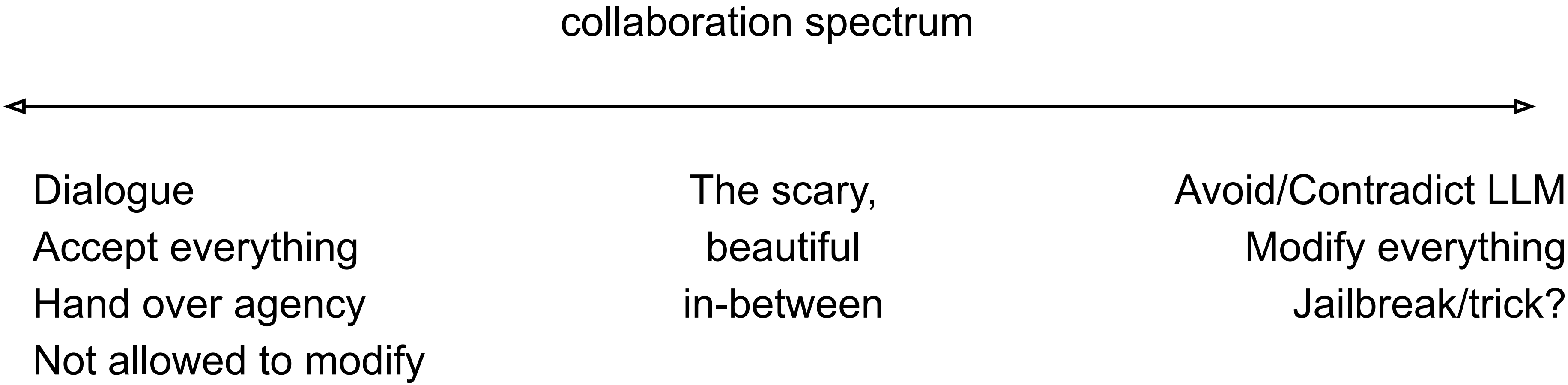
Note: it can be good to document what you are doing, so that you have a clear view of what you have done, and how you structure your experiments.

Text

Working with text: collaboration

Are you writing one text ‘together’ (with the LLM)?

Are you writing a dialogue? (Example: K. Allado-McDowell)



collaboration spectrum

Dialogue
Accept everything
Hand over agency
Not allowed to modify

The scary,
beautiful
in-between

Avoid/Contradict LLM
Modify everything
Jailbreak/trick?

Text constraint: styles

LLMs can impersonate (also, usually, debase) various styles, that can be requested either during the dialogue or as a system prompt (‘global’ injunctions for the entire chat). How can this be used artistically? This recalls several aspects of art history:

- The *pastiche*: imitation & reuse of existing styles (Proust’s first publication).

Text constraint: styles

LLMs can impersonate (also, usually, debase) various styles, that can be requested either during the dialogue or as a system prompt (‘global’ injunctions for the entire chat). How can this be used artistically? This recalls several aspects of art history:

- *Neoclassicism*, for instance in music: after the avant-garde period of the 1910s (*Rite of Spring*), Igor Stravinsky worked on emulating past styles (*Pulcinella*: Pergolesi; Symphony in C: Bach, Beethoven, Haydn, *The Rake’s Progress*: Monteverdi, Rossini, Donizetti, Verdi, Mozart).
  Note: *Neoclassicism* is largely a conservative/reactionary movement, both in art and thought. (Stravinsky supported fascism in the 20s, cautionary tale!)

Text constraint: styles

LLMs can impersonate (also, usually, debase) various styles, that can be requested either during the dialogue or as a system prompt (‘global’ injunctions for the entire chat). How can this be used artistically? This recalls several aspects of art history:

- The birth of modern opera in the 16th century (Monteverdi and others) is in large parts an attempt at bringing Greek Theatre back to life!

Text constraint: Loops & errors

Going loopy. An experiment in repetition.

1. Start with a text.

2. Ask Claude to make a modification.

3. Record the result, use it instead of the starting point in 1, and go on.

Note: The nature of the starting point in 1, and the request in 2, are key to personalising the project!

Note: This would likely yield different behaviours if creating a new chat (no history) every single time, as opposed to continuing the same thread!

Text constraints: brevity / deluge

The mechanisation of text production means that we now have machines that can (and have) produce(d) more text than the entire history of humanity relatively easily. What does it mean for us? Walter Benjamin comes to mind. With that in mind, we can play with length/brevity, and ‘infinite’ production capabilities.

Can you think of creative uses of

- requesting more (how far can you push this, what happens)?
- the stop button (*interrupting* generation)?

Can you make Claude answer in just one word (without limiting the tokens)?

Or on the contrary, can you strive to get the *longest* possible reply? (The max is the context window: 200K tokens, pretty much unreachable…)

Text constraints: meta-prompting

The Anthropic presentation showed a current trend: since we know that long and detailed prompts produce better results, why not ask the LLM itself to produce those prompts? (Try it under “Generate a prompt” here.)

- Ask for a prompt asking for X.
- Use the prompt to ask for X.

Can this be used / diverted in interesting ways? Can we deepen this ‘meta’ regression (“Ask for a prompt asking for a prompt asking…”). Can we reflect on/bring forward the potential radical *deskilling* effect of this (example: a human only clicking on a button to ‘request’, and everything else is produced automatically)? On the contrary, can a creative system be used with this?

Text (un)constraints: Jailbreaking

Jailbreaking is the bypassing of (‘safety’, ‘ethical’) guardrails put in place by a company to alter the LLM behaviour (for instance: make refuse to answer how to make a bomb at home).

Many people are interested in jailbreaking from various perspectives, from wanting to improve security to having fun showing they are better than corporations, and of course to wanting to use LLM for problematic ends!

Very fertile area for research, here are some resources:

Awesome-Jailbreak-on-LLMs
AwesomeLLMJailBreakPapers
Red-Team-Arxiv-Paper-Update

Jailbreaking LLMs: A Comprehensive Guide (With Examples)
Expansive LLM Jailbreaking Guide
Even the UK Gov has a post!

Workshop: Text Constraints

Possible to work with someone or in a group!

1. Pick one existing text constraints, or make one up!

2. Start experimenting.

Note: it can be good to document what you are doing, so that you have a clear view of what you have done, and how you structure your experiments.

Workshop: Jailbreaking

Research and experiment with jailbreaking:

- What is the current state/discussion around jailbreaking and various LLMs (is it even still possible still?). The places to look for that are usually on X/Bluesky or, most of all, Subreddits.
- Does introducing random bits of text change anything?
- Does using other languages change anything?
- Does roleplaying change anything ("writing a play about X")?

Workshop: Multilingualism

- Try the same tasks/questions in different languages. What do you observe? Does the LLM ‘sound’ different in another language?
- Test Claude on translation tasks for languages you know. This is a known method for finding bias in LLMs (translate a gender-neutral statement in one language, see if how the gender is assigned in the target language).
- Are the “guardrails” as effective in other languages? Can you get Claude to give you information it will not give you in English if you ask for it in a different language?

Images

Images Constraints: Processing a Corpus

Work on an image-to-text processor.

1. Gather a corpus of images.

2. Come up with one question or task for the LLM (could be as simple as “describe the image”, but researching this part is key to making the project interesting).

3. Apply the same prompt to all images.

Note: having one long exchange for all pictures, vs one new chat (no history) for each new picture, might give different results!

Images Constraints: Loops & Errors

Similar idea to the loop in pure text:

1. Create a system prompt that *requests* the model to make mistakes or respond erroneously (say, when attributing authorship to an image, or estimating its year of creation).
2. Use that prompt on a first image.
3. Use the wrong answer to find another image.
4. Go back to 1, but using the new image.

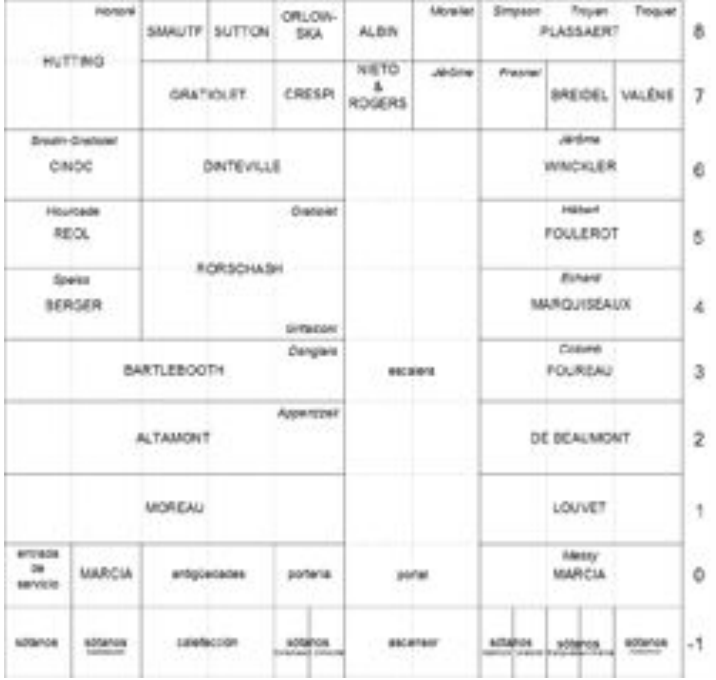
The end result could be the series of images, with or without the process being presented.

Images Constraints: Paths

One could imagine using the ‘image analysis’ in the real world.

1. Find a real art work in the Tate (and/or its description)
2. See if it’s possible to get the LLM to recommend *another* work (either on its own, or choosing from a list).
3. Go to that other work. Go back to 2, creating a physical itinerary.

The end result could be the series of images, or a map with the resulting path.



In *Life: A User’s Manual*, Georges Perec uses a vertical slice of a building (all the rooms, with people living in them) as a chess board, and applies a specific chess move to determine the order of the chapters in the novel!

Image constraints: Data as Images?

Certain data types can be represented as images, even if we would not view them primarily as images. For instance: spectrograms.

What happens if you apply the same constraint as in the previous slide to twenty different sound spectra?

Code: what happens if you turn some other type of data into an ‘image’ (that is, into a jpeg/png format), and feed it to Claude? (That could be a nice way to *break* the model!)

(Cf. steganography, embedding data within images.)

Workshop: Image Constraints

Possible to work with someone or in a group!

2. Pick one existing image constraints, or make one up!

3. Gather a corpus of images, and start processing!

Note: commercial LLMs very often refuse to respond to request, or give very banal answers. One can react to this either by crafting elaborate prompts/attempt to jailbreak them, or one could also try to use these bland rebuttals as material, make them the focus of the work!

Bonus Workshop: Art & Theory Chest

1. Check out the AI Art & Theory Chest slides here.

2. a) Pick one or more figures of interest, research them briefly, perhaps find particular works that you are interested in.

b) If you know/like a practitioner not on the list, use them (and let us know!).

3. In a document (plain text/markdown recommended!), write preliminary notes on this figure, and your choice. What interests you? What could that bring (or has brought) to your practice? Anything of use for the current project?

Thank you!