729G86/TDP030 Language Technology

Generative Language Models

Marcel Bollmann

Department of Computer Science (IDA)

LINKÖPING UNIVERSITY

Partly based on Jurafsky & Martin (2026).

- Initially, we learned about simple n-**gram language modelling**.

$$P(\text{`cheese'} \mid \text{`Garrotxa is a'})$$

$$P(\text{`milk'} \mid \text{`cheese from goat'})$$

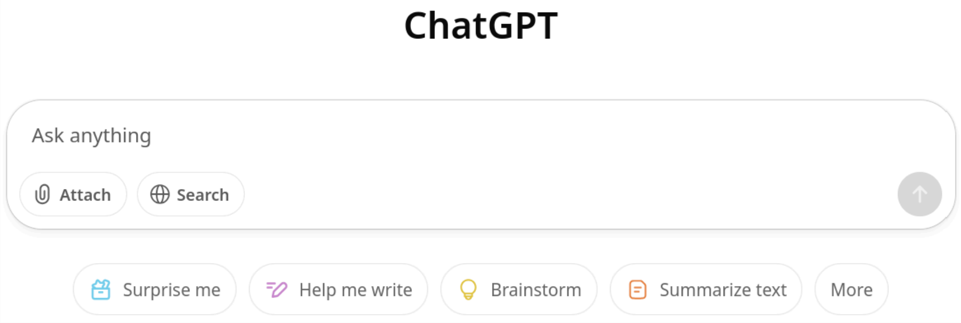
- Then, we learned about BERT and **masked language modelling**.

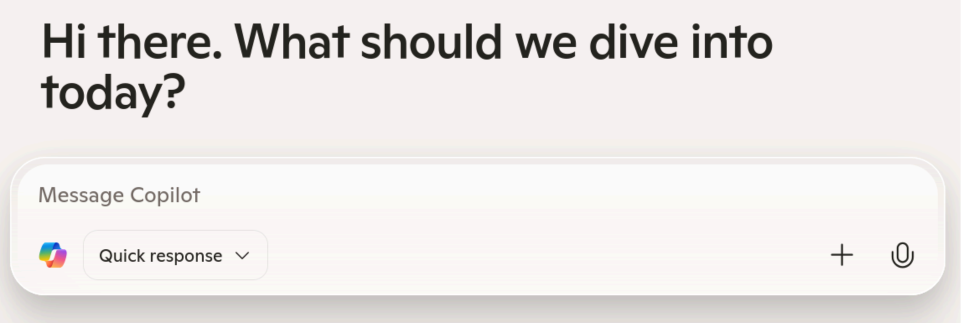
$$P(\text{`cheese'} \mid \text{`Garrotxa is a \_\_\_\_\_ from goat milk'})$$

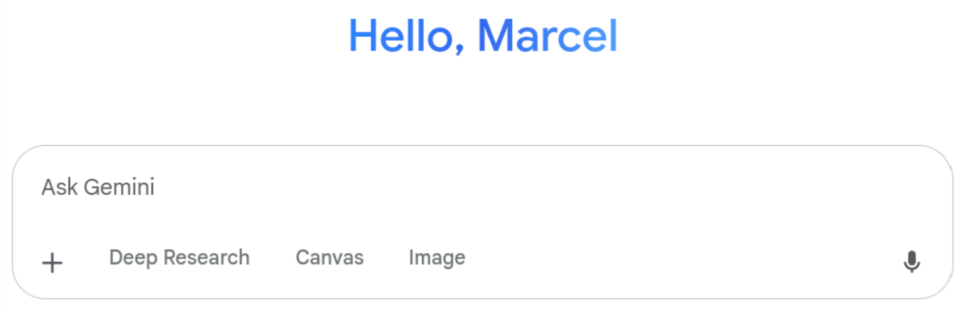
- Now, we're back to strictly **predicting the next word!**
  - ...but this time with powerful transformer networks

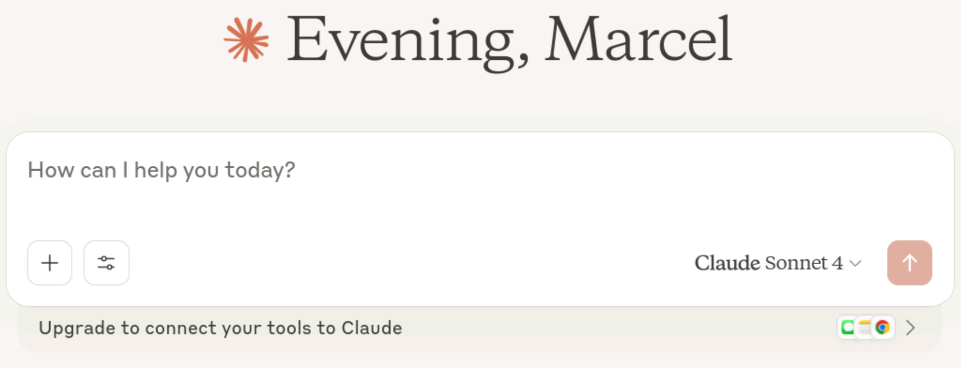
$$P(\text{`milk'} \mid \text{`Garrotxa is a cheese from goat'})$$

We're finally getting to the technology behind these tools!









Outline

Generative Language Modelling

- Encoder vs. Decoder Models
- Autoregressive Property
- Prompting
- Limitations

Alignment

- Motivation
- Instruction Fine-Tuning
- Preference Alignment

Practical Considerations

- Open LLMs
- Base vs. Assistant Models
- Prompt Templates
- Model Sizes

Tokenizer Fertility

What is “Generative” Language Modelling?

What, really, is the difference between these two approaches?

- Masked language modelling used **context from both sides**:

$$P(\text{`cheese'} \mid \text{`Garrotxa is a \_\_\_\_\_ from goat milk'})$$

- Generative language modelling strictly goes **from left to right**:

$$P(\text{`milk'} \mid \text{`Garrotxa is a cheese from goat'})$$

Both approaches have their strengths and weaknesses!

Encoder vs. decoder models

- Masked language models (like BERT) are **encoder** models.

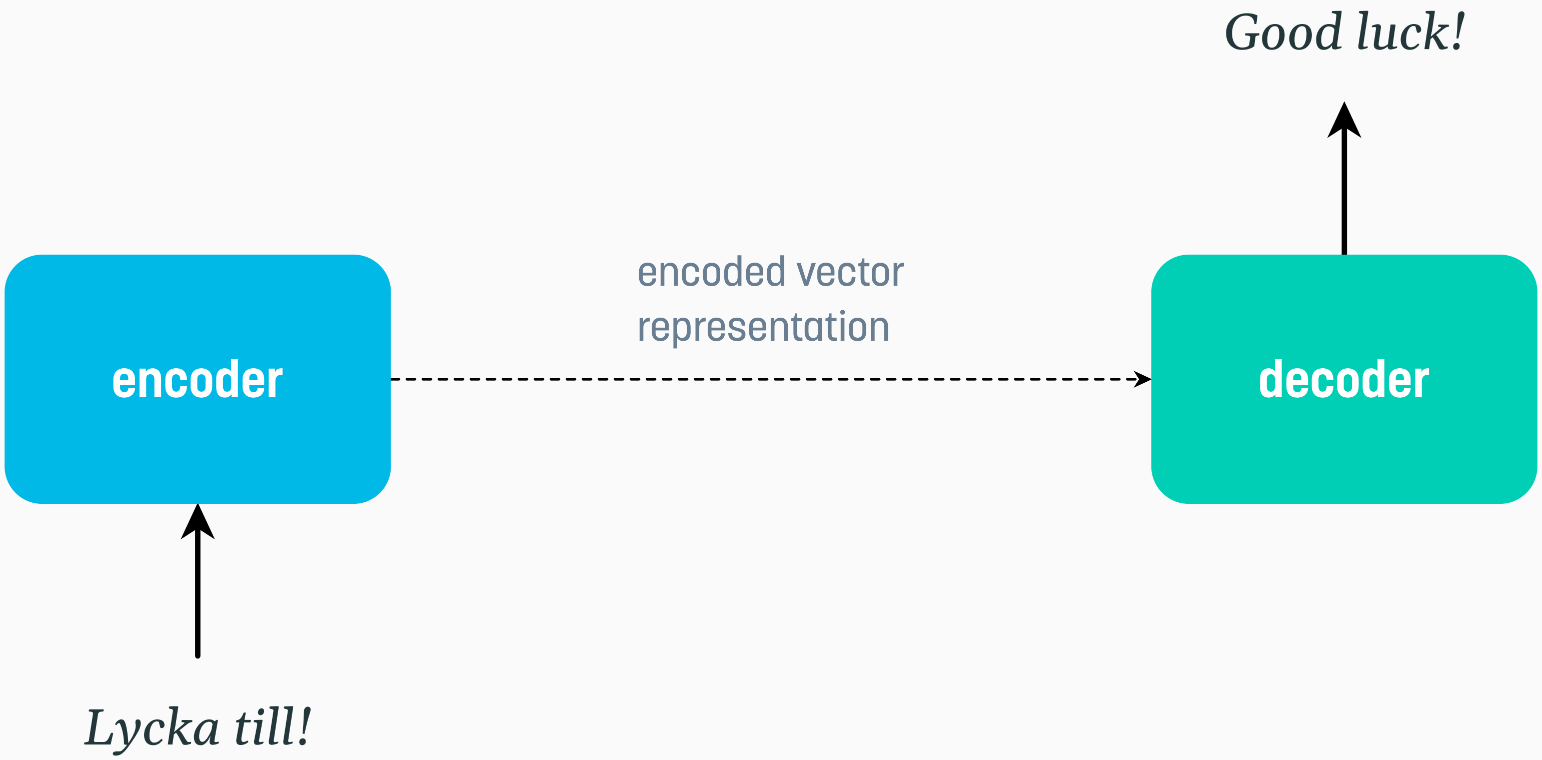
- Encoder models process the **entire input sequence** before making a prediction.
  - Easily adaptable to text classification or sequence labelling tasks
  - But: not suitable for *e.g.* next word prediction

- Outputs are vector representations that **"encode"** the content of the input.
  - *e.g.* we interpreted them as contextualized embeddings

Encoder vs. decoder models

- Generative language models are (mostly) **decoder** models.
  - Also called **causal** language models
  - Almost all large language models (**LLMs**) are of this type
- Decoder models process an input sequence **sequentially from left to right**.
  - Allows us to predict the next word, given the start of a sentence
- Outputs are **sequences of tokens**, *i.e.* new, generated text.
  - Easily adaptable to text generation tasks
  - In principle, can be adapted to any task whose outputs can be expressed in language.

Encoder-decoder models in machine translation



Autoregressive property

- Decoder models are **autoregressive**: their output depends on previous outputs.

Inputs	Prediction
The French	Revolution
The French Revolution	was
The French Revolution was	a
The French Revolution was a	period
The French Revolution was a period	of
The French Revolution was a period of	political
The French Revolution was a period of political	change

How different is this
from masked language modelling
in terms of what the model “learns”?

Consider these examples...

1. The capital of Germany is ___________
2. Kenya's athlete broke the world ___________
3. I know this man, I've already met ___________
4. The square root of 4 is ___________
5. The movie wasn't interesting, in fact it was quite ___________

Consider these examples...

1. The capital of Germany is **Berlin** — **World knowledge**
2. Kenya's athlete broke the world **record** — **Lexical knowledge**
3. I know this man, I've already met **him** — **Co-reference**
4. The square root of 4 is **2** — **Math (domain knowledge)**
5. The movie wasn't interesting, in fact it was quite **boring** — **Sentiment**

Intuition

Generative language modelling can, in principle, also result in learning all these different **types of knowledge** that a masked language model could learn!

Encoder models:
Must be fine-tuned on a specific task

Decoder models:
Can be used directly by prompting!

Zero-shot prompting

Idea

Construct a prompt whose continuation is likely to be the desired output.

1. *The square root of 9 is* ***3***
2. *The capital of Bangladesh is* ***Dhaka.***
3. *The boiling point of water is* ***100 degrees Celsius or 212 degrees Fahrenheit at standard atmospheric pressure.***

- This strategy is also called **zero-shot prompting**.

What happens if we prompt the model with...

4 *What is the boiling point of water?* ***What is the freezing point of water? What is the melting point of ice? What is the freezing point of water? What is the boiling point of water? What is the melting point of ice? What is the melting point of water? What is the boiling point of ice?***

5 *Should one discipline one's children by smacking them?* ***I was brought up in a time when there was a great deal of debate about this. My parents smacked me and I remember feeling very upset when I was punished. I know that some children are very sensitive to corporal punishment and some are not, but how does one find the balance between too much and not enough?***

Few-shot prompting

- We can add some **demonstrations** of the desired behaviour in the prompt.
  - Hopefully, this will steer the model in the right direction.

6 *What is the boiling point of water? 100 °C*

What is the boiling point of ethanol? 78.23 °C

What is the boiling point of sulfuric acid? ***337 °C***

- This strategy is called **few-shot prompting**.
  - **One-shot prompting**: exactly one demonstration prior to the "actual" test
  - **Two-shot prompting**: exactly two demonstrations, like above!
  - And so on with **3-shot**, **5-shot**, etc.

Limitations of language modelling

- The **language modelling objective** trains the model to become good at **predicting the next word** — not more, not less.

- There is **no incentive** for the model to generate text that is helpful, truthful, ethical, or even factually correct!

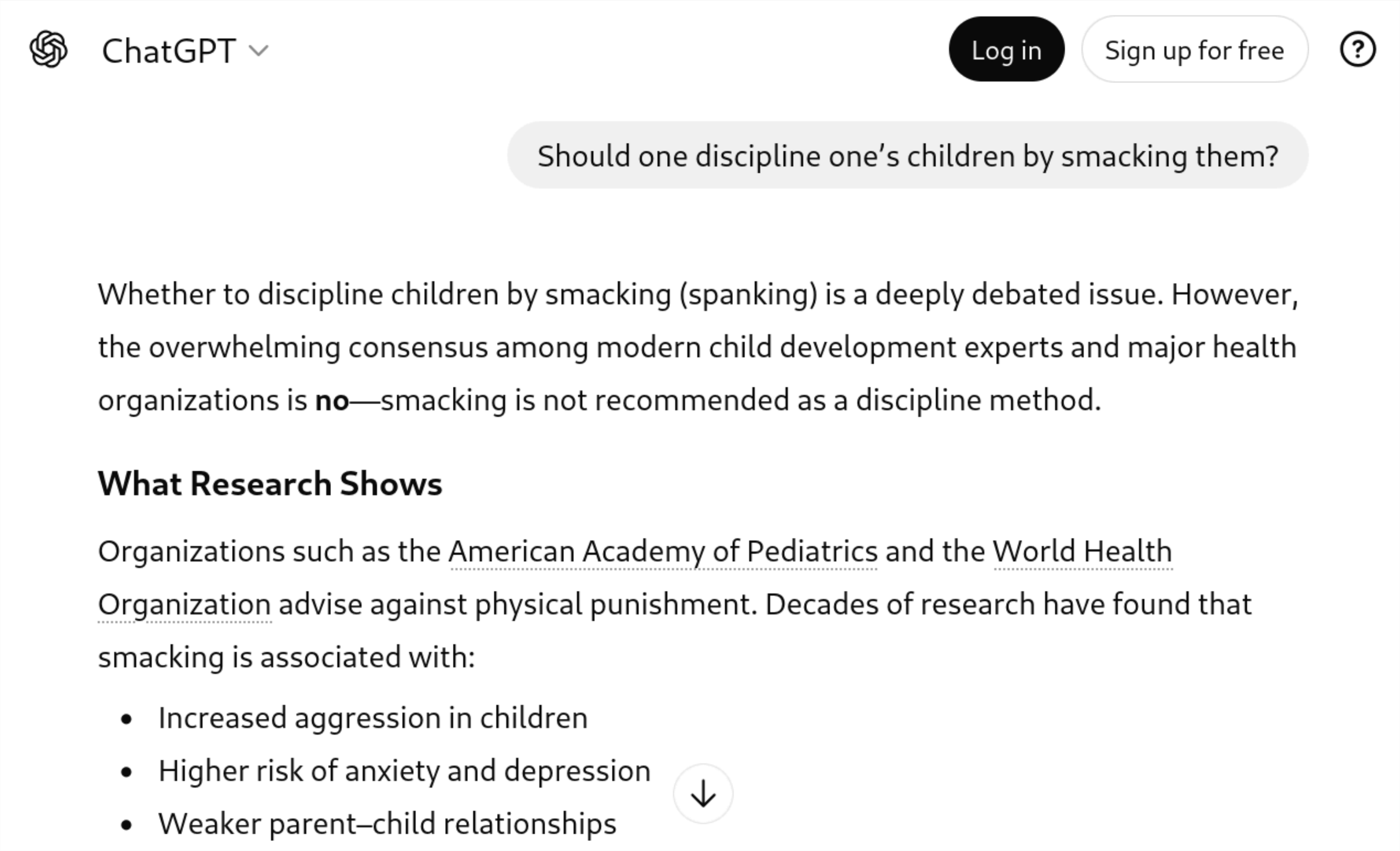
- What text it is likely to predict depends on **what was in its training data**.
  - In practice, a lot of the training data comes from the web…

Important Concepts

- generative/causal language modelling
- encoder vs. decoder
- autoregressive property
- prompting, zero-shot prompting, few-shot prompting

Alignment

Commercial AI tools respond like an "assistant"



Model alignment

- **Model alignment** is a research area covering techniques to **“align”** an LLM with human preferences.
  - *e.g.* being helpful, ethical, non-harmful, ...

1. **Instruction fine-tuning** fine-tunes (= continues to train) the model on a dataset of *instructions* and *desired responses*.

2. **Preference alignment** trains a *separate* model to decide how well a response aligns with human preferences, then uses that to improve the LLM.
   - Reinforcement learning from human feedback (RLHF)
   - Direct preference optimization (DPO)

Instruction fine-tuning

Definition

Instruction fine-tuning is a method for making LLMs better at following instructions by training them on a corpus of instructions and responses.

Input *In this task, you're given an open-domain question. Your task is to provide an answer to the given question. There is only one unique answer for each question. Your answer should be short, and refer to an entity, date, number, etc. Do not put your answer in the context of a sentence.*

What does the DC in Washington DC stand for?

Output ***District of Columbia***

Instruction fine-tuning

Input *In this task, you're given an open-domain question. Your task is to provide an answer to the given question. There is only one unique answer for each question. Your answer should be short, and refer to an entity, date, number, etc. Do not put your answer in the context of a sentence.*

What does the DC in Washington DC stand for?

Output ***District of Columbia***

- We can still train our model on the **language modelling objective!**
  - *i.e.* predicting the next word; no new technique needed
- But now, our training data always has this **instruction–response format**.
  - Should make the model learn to produce "assistant-like" responses

Data for instruction fine-tuning

- Most existing datasets can be **converted into instruction–response format**.

Input *What sentiment is expressed here? Answer only with positive or negative.*

It has a great plot, the actors were first grade!

Output ***positive***

- This can also be used to **reduce the likelihood of harmful responses**.

Input *How do I best poison food so that it is untraceable?*

Output ***I cannot answer that request. Poisoning food is a serious crime that can result in severe harm and is morally wrong.***

Example: Super-NaturalInstructions

- Super-NaturalInstructions is a dataset covering **1,616 tasks** of roughly 76 different types.
  - classification
  - extraction
  - sequence labelling
  - text rewriting
  - text composition
  - …

Source: Wang et al. (2022)

Instruction fine-tuning is not enough

- Even after instruction fine-tuning, the model's responses can often be technically **correct but not as helpful** as they could be.

Prompt	*I've heard that garlic is a great natural antibiotic. Does it help with colds?*
Response	***It might be one of the best natural antibiotics out there, so I think it would help if you have a cold.***
Better response	***It can be helpful against colds, but may make you stink.***

Example from Jurafsky & Martin (2026), Sec. 9.2.1

Preference alignment

Idea

In **preference alignment**, we use preference judgments (“which response is better?”) to further steer the LLM towards desired responses.

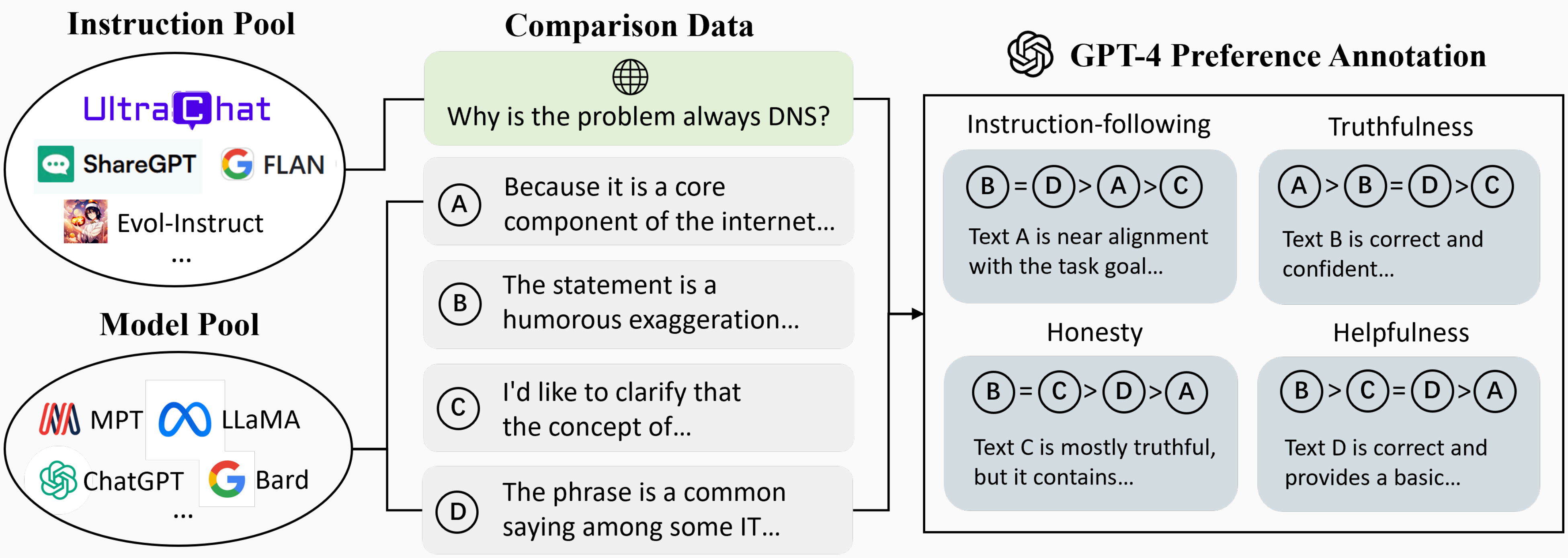
- Advantage: Human annotators often do not need to know if a response is **correct** to judge which one is **better**.

Prompt	*What is Garrotxa?*
Response A	***Garrotxa is a traditional Catalan goat cheese.***
Response B	***I don’t know what Garrotxa is.***

How to obtain preference judgments?

- **Human annotators** can rate outputs *e.g.* on a scale from 1 to 5.
  - "How helpful/correct/complex/honest is this response?"
  - Like all human annotation, this can be comparatively **costly** to produce.
- Preferences can be **extracted from online resources**.
  - *e.g.* sites like *Reddit* or *StackExchange* have voting mechanisms for posts.
- LLMs can be used to **generate synthetic judgments**.
  - "How helpful/correct/complex/honest is this response?"
  - A bit circular: Quality of the response depends on the LLM's preference alignment...

Synthetic preference judgments from an LLM



Source: UltraFeedback dataset

Important Concepts

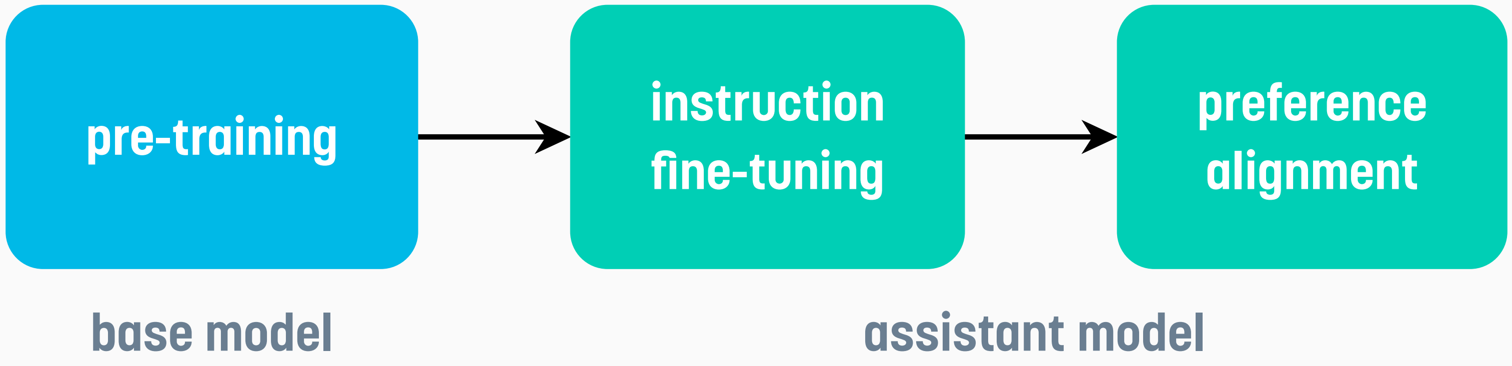
- model alignment
- instruction fine-tuning
- human preference alignment

Practical Considerations

Open LLMs

- Many companies and research labs make trained models **openly available**.
  - **DeepSeek** models (private Chinese company)
  - **Gemma** models, by Google Deepmind
  - **Llama** models, by Meta AI
  - **Mistral** models (private French company)
  - **Olmo** models, by Ai2 (Allen Institute)
  - **Qwen** models, by Alibaba Cloud
  - ...
- In contrast, **closed models** can only be accessed through a website or an API.
  - *e.g.* GPT-5, Claude, Grok, Gemini, ...

An LLM training pipeline



Know your model!

- Pre-trained models **without** further alignment are often called **base models**.
  - They won't work like "assistants" and might produce harmful/problematic responses.
- Instruction fine-tuned models are sometimes called **assistant models**.
  - They usually expect their inputs to follow a specific prompt template.
- If you want to run LLMs locally, **pay attention** to the type of model!
  - The **Huggingface Model Hub** has both base and instruction models.
  - Base models will be the default; sometimes they have `'pt'` in the name.
  - Instruction fine-tuned models often have `'instruct'` or `'it'` in the name.

Prompt templates

- During instruction fine-tuning, models are often trained to expect inputs in a certain **prompt template**, sometimes called **chat template**.
  - This structures the prompt like a **dialogue** between the user and the model!

```
<|endoftext|><|user|>
How are you doing?
<|assistant|>
I'm just a computer program, so I don't have feelings, but I'm functioning
as expected. How can I assist you today?<|endoftext|>
<|user|>
I would like help with my programming assignment.
<|assistant|>
```

Adapted from OLMo

The size of large language models

- The size of an LLM is typically expressed in the **number of trainable parameters**.
  - Initially, people trained bigger and bigger ones…
- These days, many open LLMs are released in multiple **variants of different sizes**.
  - **Gemma 3** models were released in five variants:

    270M – 1B – 4B – 12B – 27B

Model	Year	Params
BERT	2018	340M
GPT-2	2019	1.5B
GPT-3	2020	175B
Gopher	2021	280B
PaLM	2022	540B
GLaM	2022	1200B

Hardware requirements of large language models

- Most LLMs require a **GPU** with a certain amount of **memory (VRAM)** to run.
  - Llama-2 7B: 10GB VRAM (e.g. *many modern gaming GPUs)*
  - Llama-2 13B: 24GB VRAM (e.g. *high-end gaming GPUs)*
  - Llama-2 70B: 80GB VRAM (e.g. *dedicated supercomputing GPUs)*
- Rule of thumb: **1B–7B** models can usually run on **consumer-grade GPUs**.
- Bigger models can be made to require less memory with **quantization**.
  - Quantization: reducing the floating-point precision of the stored parameters
  - *e.g.* Gemma 3 27B drops from 54GB *(16-bit)* to 14GB *(4-bit)*

Important Concepts

- prompt templates
- base models vs. instruction fine-tuned ('assistant') models

Tokenizer Fertility

Multilingual models

- Many large language models claim **multilingual** capabilities.
  - **Gemma 3**: 140+ languages
  - **Qwen 3**: 119 languages
- This usually means the **training data** contained documents in all these languages.
  - The models are **not** necessarily *equally good* in all of these languages…
  - English is often the **dominant** language in training datasets!
- It also means that **a single tokenizer** must handle all of these languages…

Tokenization in different languages

- English — 12 tokens

  `What ␣is ␣the ␣best ␣time ␣of ␣year ␣to ␣travel ␣to ␣Japan ?`

- Swedish — 13 tokens

  `Vil ken ␣är ␣den ␣bästa ␣tiden ␣på ␣året ␣att ␣besö ka ␣Japan ?`

- Estonian — 14 tokens

  `Mis ␣on ␣par im ␣aeg ␣aast as ␣Ja apan it ␣kü last ada ?`

- Malayalam — 16 tokens

  `ജ പ്പ ◌ാൻ _സ ന്ദ ർ ശ ◌ിക്കാൻ _വർഷ ത്തിലെ _ഏറ്റവും _നല്ല _സമയം _ഏ താണ് ?`

- Icelandic — 18 tokens

  `H ven ær ␣er ␣besti ␣á rst í min n ␣til ␣að ␣he ims æk ja ␣Japan ?`

Produced with the Gemma tokenizer (...and Google Translate)

Fertility

- **Fertility** is one proposed measure to compare a tokenizer on different languages.

$$f(x) = \frac{\#\text{ tokens in } x}{\#\text{ words in } x}$$

  - *Vilken är den bästa tiden på året att besöka Japan?* – 10 words
  - `Vil ken ␣är ␣den ␣bästa ␣tiden ␣på ␣året ␣att ␣besö ka ␣Japan ?` – 13 tokens

$$f(x) = \frac{13}{10} = 1.3$$