

# LTI

## Language, Technology and the Internet

### Large Language Models

Francis Bond

Department of Asian Studies

Palacký University

<https://fcbond.github.io/>

[bond@ieee.org](mailto:bond@ieee.org)

Lecture 11

---

# LLM

# Large Language Models

# Large Language Models

---

- Generative models that capture extended context in order to predict the next token (word, sub-word or symbol)
- The main innovation is in the sheer size
- They model how language is used
- They can appear know a lot about the world
  - Humans naturally make meaning of language we encounter and imagine the mind behind that language
    - \* Artificial agents have limited capacity for communicative intent
    - \* And some natural language systems have none
  - Dealing with the risks of language technology requires recognizing and accounting for the above

---

# Meaning making with artificial interlocutors and risks of language technology, Talk at HiTz, Emily Bender 2023

# Generative Pre-trained Transformer (GPT) models

---

- Non-supervised — just needs text (and/or code)

Date	Name	Parameters*	Data	Type
2018	GPT	117	4.5 GB	Books
2019	GPT-2	1,500	40 GB	Web
2020	GPT-3	175,000	570 GB	Web 82%, Books 16% Wiki 3%
2021	ChatGPT**			
2023	GPT-4	***1,750,000	?	text and images
2024	GPT-4o		?	+ audio and video

\* million

\*\* GPT-3.5 fine-tuned to follow instructions and interact

\*\*\* estimate

Lambdalabs estimates a cost of around \$4.6 million US dollars and 355 years to train GPT-3 on a single GPU in 2020

# ChatGPT Intro

---

- The base is generative pre-trained transformer (GPT) model based on the transformer architecture developed by Google
- Training data includes software manual pages, bulletin board systems, multiple programming languages, wikipedia, books and more (up to Jan 2022)
- Fine-tuned for conversational applications using a combination of supervised and reinforcement learning techniques from human feedback
- Interact with a prompt, to which you get a response
  - keeps a context of 4k tokens ( $\approx 500$  tokens/A4 page)
  - longer context (up to 32k) in newer models

- 
- Free interface, beyond a certain amount you are charged per token (from \$0.002 – –\$0.12/token)

# Prompt Engineering

---

The first prompt provides a starting point or direction for ChatGPT. The response follows the same tone, context, and style as the prompt.

- Start simple - start with a simple prompt and build on it
- Give a command - start the prompt with an action word like "Write", "Create", or "Summarize" instead of "Can you"
- Add Context - add specific and relevant context to the task you want to perform
- Add Expectations - add clear and direct expectations for the content, like how long it should be and what to include



---

Create a list of interesting places to visit in Olomouc:

- they should be cheap
- they should be fun

# Role Prompting

---

A simple way to add context to a prompt is to assign a role to ChatGPT.

You are a student.

Create a list of interesting places to visit in Olomouc:

- they should be cheap
- they should be fun

You are an artist.

Create a list of interesting places to visit in Olomouc:

- they should be cheap
- they should be fun

---

You are a foodie.

Create a list of interesting places to visit in Olomouc:

- they should be cheap
- they should be fun

# You can add a lot more information

---

- **Zero-shot** prompting means providing a prompt with no examples

Classify the text into positive, neutral or negative:

Text: That shot selection was awesome.

Classification:

---

➤ **Few-shot** prompting means you give the system some examples

Text: Today the weather is fantastic

Classification: Pos

Text: The furniture is small.

Classification: Neu

Text: I don't like your attitude

Classification: Neg

Text: That shot selection was awful

Classification:

- 
- **RAG: Retrieval-augmented generation** means you give the system some similar examples (from a database or corpus of some kind)

```
Input: That shot selection was awful
LOOKUP SIMILAR EXAMPLES AND ADD THEM:
Text: Your shot selection was brilliant
Classification: Pos
Text: That shot choice was unexpected
Classification: Neu
Text: The editor's That shot selection sucked
Classification: Neg
Text: That shot selection was awful
Classification:
```

# ChatGPT example with formatting

---

Write a short introduction to computational linguistics using LaTeX with linguistic examples

```
\documentclass{article}
```

```
\begin{document}
```

```
\section{Introduction to Computational Linguistics}
```

Computational Linguistics is an interdisciplinary field that combines knowledge and techniques from linguistics, computer science, artificial intelligence, and cognitive psychology to understand and model language from a computational perspective.

For example, computational linguistics can be used to study the structure of language at different levels, such as syntax and morphology. In syntax, computational linguistics can help identify the rules that govern how words are combined to form

---

phrases and sentences. For instance, in the sentence "The cat chased the mouse", computational linguistics can analyze the syntactic structure to identify that "The cat" is the subject, "chased" is the verb, and "the mouse" is the object.

Similarly, in morphology, computational linguistics can help identify the rules that govern how words are formed from smaller units of meaning, called morphemes. For example, in the word "unhappiness", computational linguistics can analyze the morphological structure to identify that "un-" is a prefix meaning "not", "-happi-" is the root meaning "happy", and "-ness" is a suffix meaning "state of".

Overall, computational linguistics provides a powerful set of tools for analyzing language and understanding its underlying structure. It has many practical applications, such as natural language processing, machine translation, and speech recognition, and continues to be an active area of research and development.

\end{document}



# Some possible uses

---

- Brainstorming
- First Draft
- Find Title
- Email
- Summarize
- Code
- Debug

# Brainstorming

---

What are some unique ideas for activities for 10 five-year-old girls at a birthday party?

How can we combine ideas 4 and 6? How can we make this into a unique experience that the children will remember?

# First Draft

---

Write an outline for a tutorial on prompt engineering

For a tutorial on prompt engineering, write a section about "Understanding the Basics of Prompts"

# Find Title

---

Suggest 10 titles for a short story about a robot posing as a student. It is written from a teenager's perspective, and the robot is almost a perfect copy of a human. They find friends and fall in love. But they still can't solve captchas when browsing the internet.

Number 8 suits the story well, as the story is mainly about the robot figuring out how to stay true to its robotic nature but at the same time develop as a human. Based on that, suggest 10 more titles.

# Email

---

Write a professional email requesting the day off on the 17th of November? This is a holiday in the Czech Republic and though we understand that this is not usually a day off in the USA, we want to respect our Czech heritage with our family.

Write a professional email complaining about the lack of parking spaces around our office building

These emails may contain details that are not true? They may be over long? What do you think?

# Summarize

---

Summarize "the War with the Newts"

Summarize "the War with the Newts" by Karel Čapek

Summarize "the War with the Newts" by Conan Doyle

Summarize "the War with the Newts" from the newts' point of view

Beware of hallucinations!

# Code

---

```
Make a website using flask for the computational  
linguistics group with pages for teaching, research and  
news
```

```
Add a navigation bar
```

# Debug

---

```
class Test:
    def printToTen(self):
        ten = 10
        print(ten)

test = Test()
test.printToTen() # Should print 10
```

Why does my Python code not iterate from 1 to 10 and print all numbers out?



# Key Principles of Prompt Engineering

---

- Clarity and Specificity in Prompts
  - Avoiding Ambiguity
  - Defining Clear Context and Goals
- Tailoring Prompts to Model Capabilities
  - Understanding Model Strengths and Limitations
  - Crafting Prompts that Align with Model Capabilities
- Iterative Refinement
  - Experimentation with Various Prompts
  - Analyzing Model Responses for Improvement
  - Ask for multiple options

# Specialized tools based on LLMs

---

- Some LLMs can run code for you in a sandbox  
great for web programming
- [consensus.app](#) is an academic search engine. It uses language models (LLMs) and purpose-built search technology (Vector search) to find relevant papers, then synthesizes topic-level and paper-level insights. It also links papers to other papers.
- There are many more

---

# Discussion

# LLM pros and cons

---

- Mainly grammatical text
  - For English, although no one knows if it is the same as human produced text
- Good at transformation
  - Translation
  - Text to code (quicker but less secure)
  - Text to image
- Very expensive to train
  - Only corporations can build the latest models
  - Uses a lot of computing power
  - Big carbon footprint
    - every month it uses (estimated) 4 M KWh  $\approx$  as much as 30 Danes to handle  $\approx$  3,000 million requests!

- 
- It learns from the web
    - A very wide range of knowledge
    - ⊗ Including sexist, racist and other toxic language
  - It does not have a model of the world
    - GPT-3 models relationships between words without having an understanding of the meaning behind the utterances
    - So it makes stuff up (**hallucinates**)  
and there is no way of knowing when it is doing so  
more accurately, it cannot distinguish truth from falsehood
    - Fixing this is a hot-topic of research, but there is no clear answer  
RAG and linking to sources are the best options
  - Some of its training data is copy-righted text
    - It is not clear whether this is **legal** (probably) or **ethical** (probably not if it can closely reproduce the input text/image/code)
    - This is especially a problem for visual models

---

➤ Text produced by a language model is always high-probability text

- So it tends to be blah
- The programming is not-elegant
- The pictures are derivative

We may end up being drowned in mediocre AI produced crap (**slop**)

- SF magazines are already reporting an increase in submissions
- **If students can use it to save time, then they will**
- **We have to set problems that cannot just be solved using LLMs**
- **We should teach students to use it effectively**
  - **like a calculator for text**

➤ LLMs may become self-aware and super-intelligent and kill all humans

- Unlikely given their lack of understanding or linking to the physical world
- They are building some internal models
  - but they have no understanding or actual planning

# ChatGPT

---

# If ChatGPT Can Do It, It's Not Worth Doing

---

[If ChatGPT Can Do It, It's Not Worth Doing](#), *Inside Higher Ed* by John Warner, 2023

[AI from a legal perspective](#) Linux Weekly News by Jake Edge September 26, 2023

[Chatbot Hallucinations Are Poisoning Web Search](#) Wired, ILL KNIGHTBUSINESSOCT 5, 2023 12:00 PM (accessed 2023-10-06)

[Signs of undeclared ChatGPT use in papers mounting](#) *Retraction Watch* October 6, 2023 Frederik Joelving (accessed 2023-10-03)

[An Evaluation of a Zero-Shot Approach to Aspect-Based Sentiment Classification in Historic German Stock Market Reports](#) (2023) Janos Borst, Lino Wehrheim, Andreas Niekler, Manuel Burghardt Preprints of Communication Papers of the of the 18th Conference on Computer Science and Intelligence Systems pp. 51–60

[Debunking the Chessboard: Confronting GPTs Against Chess Engines to Estimate Elo Ratings and Assess Legal Move Abilities](#) Mathieu Acher Blog, September 30, 2023, accessed 2023-10-18



---

[Are the emergent abilities of LLMs like GPT-4 a mirage?](#) TechTalks By Ben Dickson  
-May 17, 2023, accessed 2023-10-19

[The real AI fight](#) Pluralistic, by Cory Doctorow (27 Nov 2023)

# Acknowledgments

---

- Some prompt examples taken from [ChatGPT-3.5 Tutorial](#) from W3 Schools
- <https://rollbar.com/blog/how-to-debug-code-using-chatgpt/>