



# DATA 8005 Advanced Natural Language Processing

## Lecture I: Introduction

Fall 2024

# Logistics

- Location: IDS P603
- Meetings: Friday 1:30 - 4:20pm
- Instructor: Tao Yu (<https://taoyds.github.io/>)
- Office hours: Tuesday 4 - 5pm @CB 204E

# Logistics

Course website: <https://taoyds.github.io/courses/data8005>

- We will maintain the website for schedule, lecture slides, reading lists, grading policies, etc
- Only submit your reports on Moodle.

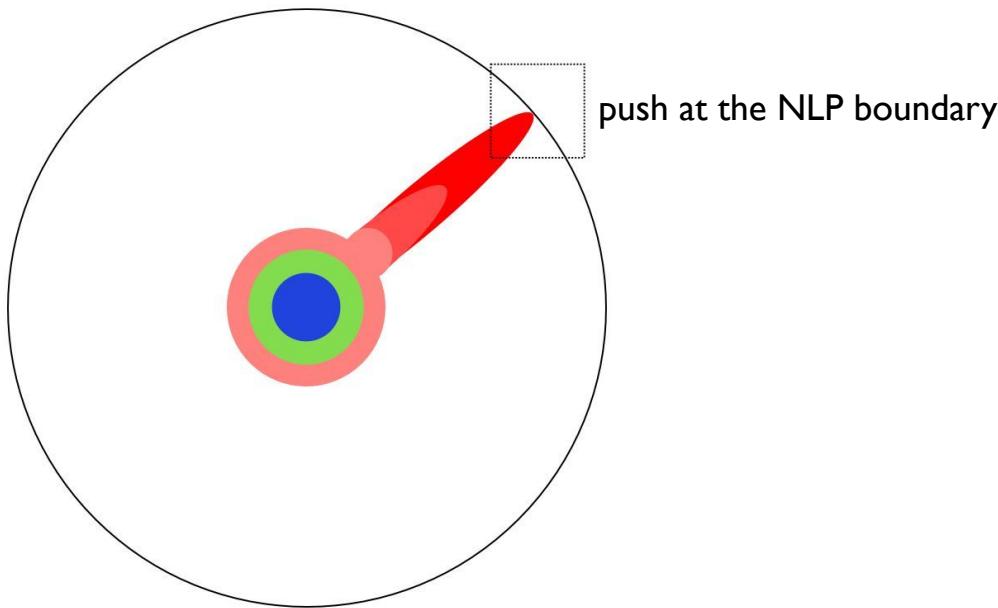
# Logistics

Slack: [https://join.slack.com/t/slack-t4u9128/shared\\_invite/zt-2qj3z7to2-GhKpsWy5YT1r9rWvIBFLw](https://join.slack.com/t/slack-t4u9128/shared_invite/zt-2qj3z7to2-GhKpsWy5YT1r9rWvIBFLw)

- We will use Slack as the primary mode of communication. DM me on Slack instead of emails.
  - Provide feedback on your lectures, answer any questions about lectures, projects, and so on
  - More importantly, encourage you to ask questions, share random thoughts, highlight interesting papers, brag about cool finding there.
- Join Slack via the invitation link above or scan the QR code below.



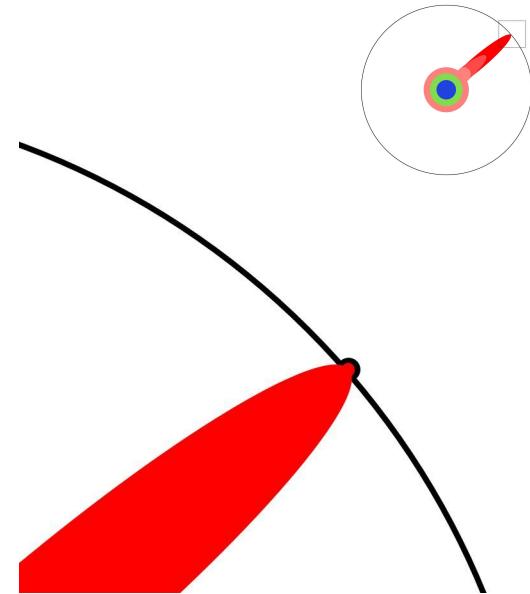
# Course structure



**This is an advanced graduate course**

# Course structure

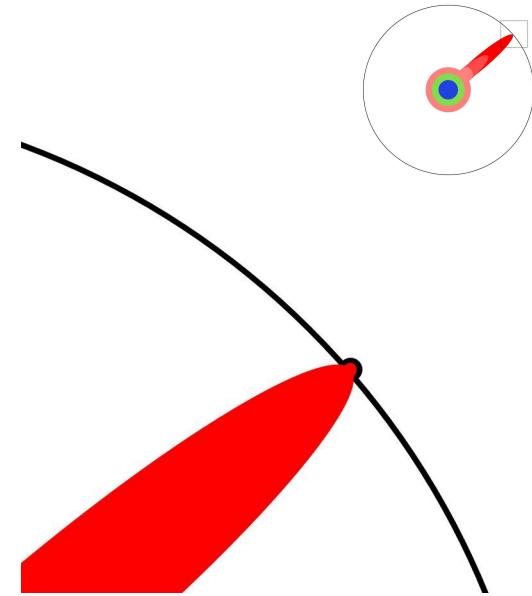
- We will be teaching and discussing state-of-the-art papers about large language models.
- Mostly presentations and discussions. I will deliver  $\frac{1}{3}$ , while you will present the other  $\frac{2}{3}$ .
- You are expected to come to the class regularly and participate in discussion.



**This is an advanced graduate course**

# Course prerequisites

- Familiarity with deep learning and machine learning
- Familiarity with Python programming
- Familiarity with basic NLP tasks
- NLP course background is recommended.
- If you don't have a background in NLP, you may find the introductory undergraduate-level NLP course [COMP 3361](#) helpful.



**This is an advanced graduate course**

# Course goals

- Learn more about current state-of-the-art NLP research
- Practice oral and written NLP research presentation skills
- Practice different parts of NLP research cycle including understanding prior literature, brainstorming ideas, choosing a research problem, executing a project.



# Components and grading

- **In-class presentation: 30%**
  - Survey and review papers/blogs on a topic and present it in the class.
- **Class participation: 25%**
  - Read papers before the class and comments on Slack (answer, ask, or add >2 high-quality questions/comments, suggest related papers)
  - In-class group discussions or idea pitches
  - Feedback on presentations from your classmates
- **Final project: 45%**
  - More details found on the course page.

# In-class presentation

- **Two** students work together and deliver a 60-minute presentation
- Your goal is to **educate others** in the class about the topic
  - Cover at least the required paper(s)
  - Search for latest related papers (citations)
  - Survey more related work/blogs to add background and context
- Presentation meeting
  - Send me your draft slides on Slack **before** my office hours
  - Use Google slides and add slide numbers
  - Provide comments and give suggestions on Slack
- Everyone else: prepare and ask questions!

# What is a good presentation?

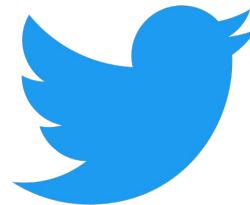
- Add background and context and survey more related work/blogs
  - What this topic is about
  - Why this topic is important
  - The most important problems and recent methods on this topic
- Paper presentation
  - Most papers were published in the past 1-2 years (if not a few months)
  - Highlight the biggest takeaways
  - Why this paper is important, what problem it trying to solve
  - Pay attention to technical details, but don't too much
  - Future work?
- More or less like a short tutorial on this topic

# How to survey a new NLP research topic?

- Related papers
  - Prior work: Related Work section
  - Latest work: papers that cite the assigned papers
- Related recent tutorials at NLP conferences (in 1-2 years)
- Good survey papers/blogs

# How to survey a new NLP research topic?

- Related papers
  - Prior work: Related Work section
  - Latest work: papers that cite the assigned papers
- Related recent tutorials at NLP conferences (in 1-2 years)
- Good survey papers/blogs
- Many NLPers/AI researchers use Twitter



# Latest AI big news today on Twitter?

 OpenAI ✨  
@OpenAI

We're releasing a preview of OpenAI o1—a new series of AI models designed to spend more time thinking before they respond.

These models can reason through complex tasks and solve harder problems than previous models in science, coding, and math.



Introducing OpenAI o1 - a new series of reasoning models for solving hard problems

From openai.com

1:09 AM · Sep 13, 2024 · 2.3M Views

632 replies 3.9K retweets 11K likes 2K bookmarks

 Jim Fan ✅ @DrJimFan · 4h

OpenAI Strawberry (o1) is out! We are finally seeing the paradigm of inference-time scaling popularized and deployed in production. As Sutton said in the Bitter Lesson, there're only 2 techniques that scale indefinitely with compute: learning & search. It's time to shift focus to [Show more](#)



Pre-training Post-training Inference

**Most LLMs**

Pre-training Post-training Inference

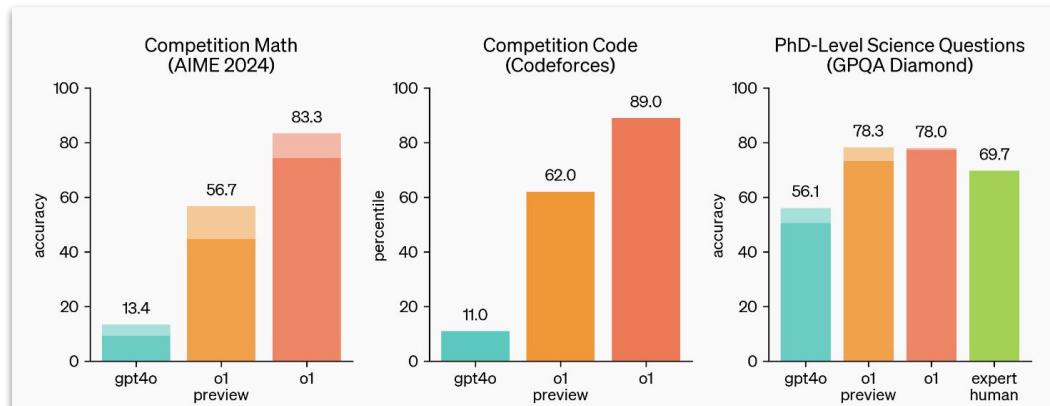
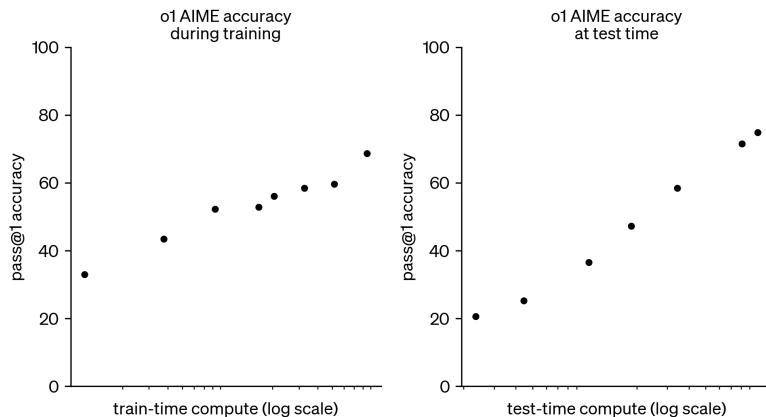


@DrJimFan

63 replies 598 retweets 3K likes 243K bookmarks

# Latest AI big news today on Twitter?

- o1 performance consistently improves with more reinforcement learning (train-time compute) and with more time spent thinking (test-time compute).



# LLMs development: too fast to keep up...

## Advice on prompting

These models perform best with straightforward prompts. Some prompt engineering techniques, like few-shot prompting or instructing the model to "think step by step," may not enhance performance and can sometimes hinder it. Here are some best practices:

- **Keep prompts simple and direct:** The models excel at understanding and responding to brief, clear instructions without the need for extensive guidance.
- **Avoid chain-of-thought prompts:** Since these models perform reasoning internally, prompting them to "think step by step" or "explain your reasoning" is unnecessary.
- **Use delimiters for clarity:** Use delimiters like triple quotation marks, XML tags, or section titles to clearly indicate distinct parts of the input, helping the model interpret different sections appropriately.
- **Limit additional context in retrieval-augmented generation (RAG):** When providing additional context or documents, include only the most relevant information to prevent the model from overcomplicating its response.

# How to survey a new NLP research topic?

- Related papers
  - Prior work: Related Work section
  - Latest work: papers that cite the assigned papers
- Related recent tutorials at NLP conferences (in 1-2 years)
- Good survey papers/blogs
- Many NLPers/AI researchers use Twitter (Read the firsthand info yourself!)
- We can select these papers together
- Other related courses



# How to survey a new NLP research topic?

- Other related courses
  - Regular NLP courses
    - [CS224N: Natural Language Processing with Deep Learning](#)
  - Seminar-based advanced NLP courses
    - [COS 597G: Understanding Large Language Models](#)
    - [COS 597F: Embodied Language Understanding](#)
    - [CS25: Transformers United V2](#)
    - [CSE 599: Exploration on Language, Knowledge, and Reasoning](#)
    - [Understanding the World Through Code](#)
    - [CS324: Large Language Models, winter 2023](#)
    - [Large Language Model Agents](#)

# Before class

- Read the assigned papers before the class
  - **Very likely** you will need to read up on some prior literature to fully understand the paper(s) we discuss!
- Pre-lecture questions
  - Send your comments on `# week#` Slack channel before the class (answer, ask, or add >2 high-quality questions/comments about one topic each week)
- Come prepared with points of discussion and ideas to contribute in class

# In-class group discussion

- Ask questions or suggest paper improvements
- Answer others' questions
- Add more related work or propose new ideas
- Review and discuss pre-lecture questions
- Also can provide feedback on the presentation: clarity, structure, completeness, slides

# What is good research in AI?

- We talked about what a good presentation is.
- Btw, what defines good research that makes a difference in the current, crowded AI space?



Omar Khattab

@lateinteraction

...

Thoughts on Research Impact in AI.

Grad students often ask: how do I do research that makes a difference in the current, crowded AI space?

This is a blogpost that summarizes my perspective in six guidelines for making research impact via open-source artifacts. Link below.

11:31 PM · Sep 4, 2024 · 204.9K Views

22

278

1K

1K

↑

## Thoughts on Research Impact in AI

Grad students often reach out to talk about structuring their research, e.g. **how do I do research that makes a difference in the current, rather crowded AI space?** Too many feel that long-term projects, proper code releases, and thoughtful benchmarks are not incentivized — or are perhaps things you do quickly and guiltily to then go back to doing 'real' research.

This post distills thoughts on impact I've been sharing with folks who ask. Impact takes many forms, and I will focus only on **making research impact in AI via open-source work** through artifacts like models, systems, frameworks, or benchmarks. Because my goal is partly to refine my own thinking, to document concrete advice, and to gather feedback, I'll make rather terse, non-trivial statements. Please let me know if you disagree; I'll update here if I change my mind.

Here are the guidelines:

1. Invest in projects, not papers.
2. Select timely problems with large headroom and "fanout".
3. Think two steps ahead and iterate fast.
4. Put your work out there and own popularizing your ideas.
5. Funnel the excitement you build: Tips on growing open-source research.
6. Continue investing in your projects, via new papers.

# Lecture format

- I will deliver  $\frac{1}{3}$ , while you will present the other  $\frac{2}{3}$ .
  - Introduction to high-level ideas and concepts in NLP and LLMs.
- Two student lectures on advanced topics per week ( $2 \times \sim 6 = \sim 12$  lectures)
  - 60-minute per lecture
    - Be prepared for lots of questions (and we encourage questions in the class)
    - Please control your time (rehearsal is very helpful)!
  - 15-minute post-lecture group discussion
    - Divide the class into groups of 4-5 students (depending on seating and enrollment)
- Potential guest lectures

# Final project

- Students complete a research project in teams of 2-3
- Here are a few example project tasks, such as [OSWorld](#), [SWE-bench](#), and [Spider 2.0](#).
- Deadlines
  - Group registration: week 5, Oct 4
  - Project proposal: week 7, Oct 18
  - Final report paper: 11:59 pm on Dec 15
- In-class presentation: ToD

# Policy on ChatGPT, Copilot, and other AI assistants

- This course emphasizes understanding the capabilities and limitations of these AI systems, and there's no better way to do that than by using them! **Collaboration with these systems is allowed**, treating them as collaborators in the problem-solving process.

# Topics and schedule (tentative)

- Introduction to Large language models (LLMs)
- NLP applications
- Advanced LLM topics

# Large language models (LLMs)

- LLM pretraining
- LLM Prompting, in-context learning, scaling law
- LLM evaluation, data, and benchmarking
- Instruction tuning for LLMs
- LLM alignment/RLHF

# NLP applications

- Question answering, reasoning
- Text generation
- Semantic parsing, code generation
- LM agents, language grounding

# Advanced LLM topics

- Robustness, interpretability, explainability of LLMs
- Bias, toxicity, and privacy in LLMs
- Parameter-efficient LM tuning
- Efficient LLM methods and Infrastructure
- Multimodal LM/VLMs
- Language in robotics, and embodied interaction

# **Introduce yourself!**

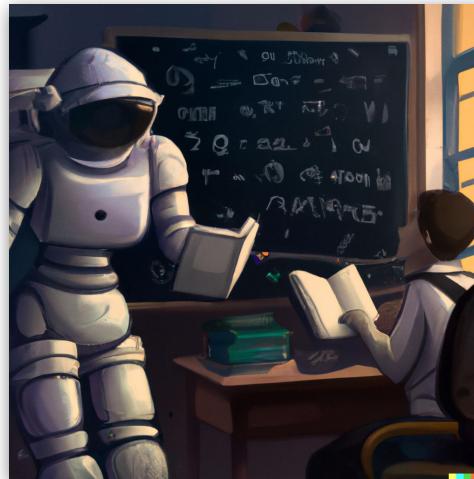
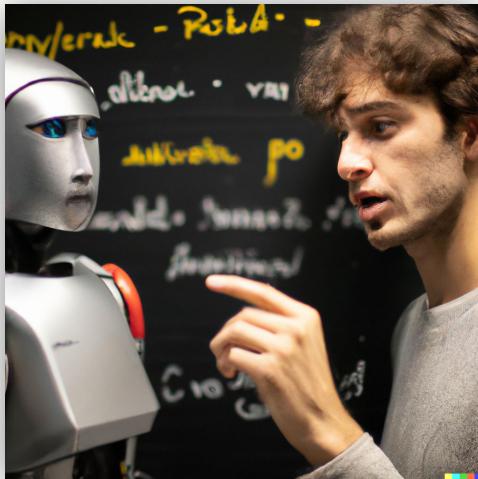
**&**

## **Discussion:**

**What are you most excited about LLMs and want to learn  
from the class?**

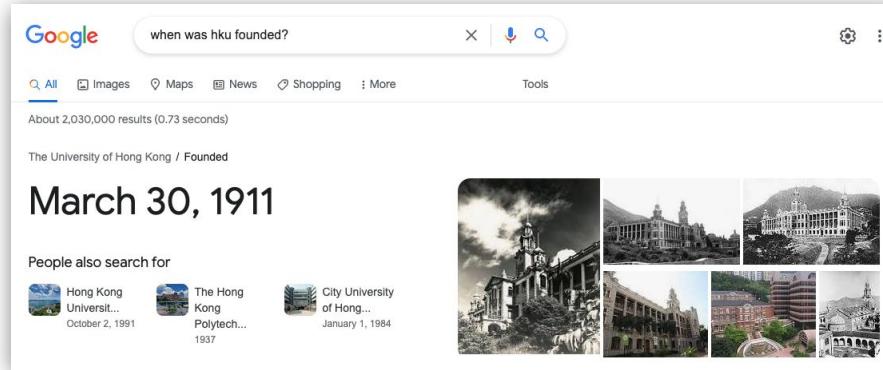
# What is NLP? Wait, what is language?

- **Language** is the abstraction of the real world!
- **Natural Language Processing (NLP)** aims to teach computers human languages a computational perspective.



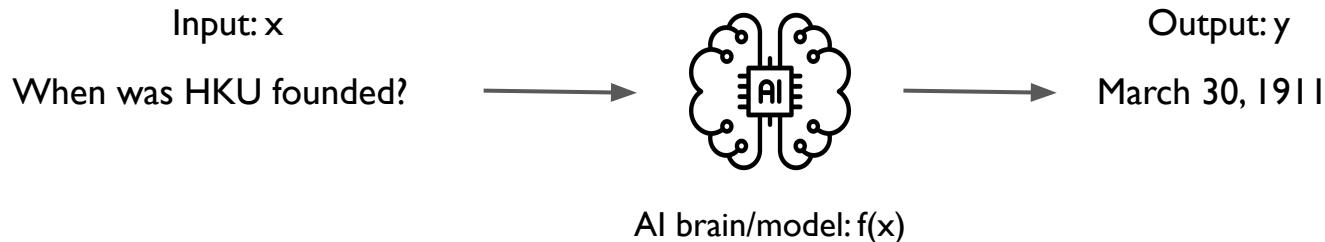
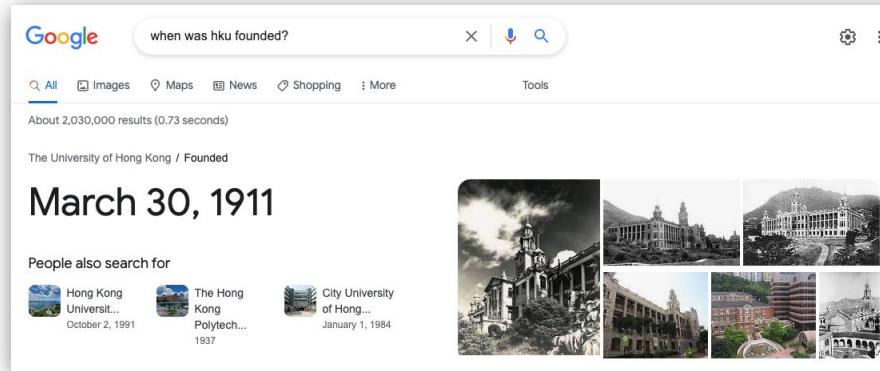
# About NLP: teaching computers human languages

- NLP in real world applications
  - Q&A / IR - Google search



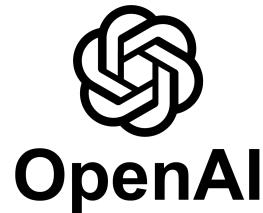
# About NLP: teaching computers human languages

- NLP in real world applications
  - Q&A / IR - Google search



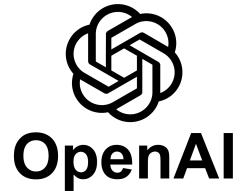
# About NLP: teaching computers human languages

- NLP in real world applications
  - Q&A / IR - Google search
  - Dialogs - Apple Siri / Amazon Alexa



# About NLP: teaching computers human languages

- NLP in real world applications
  - Q&A / IR - Google search
  - Dialogs - Apple Siri / Amazon Alexa
  - Grammar checking (Grammarly), summarization, sentiment analysis ...



Welcome to the Grammarly Editor, the best place to write what's important.

Red underlines mean that Grammarly has spotted a mistake in your writing. You'll see one if you misspell something. (See what we did there?) You'll also see an underline, if you misuse a punctuation mark. If you're worried about typos or grammatical errors that could affect your credibility, Grammarly will help you fix those. Click any of Grammarly's suggested corrections to apply them to your text, or open a brief explanation to learn more.

• SPELLING

sea → see

The word **sea** doesn't seem to fit this context. Consider replacing it with a different one.

[Learn more](#) | Delete Edit

# What ChatGPT can do?

The screenshot shows the 'Examples' section of the ChatGPT website. At the top, there's a navigation bar with 'Overview', 'Documentation', and 'Examples'. Below the navigation is a search bar and a dropdown menu for 'All categories'. The main area is titled 'Examples' and has a subtitle 'Explore what's possible with some example applications'. There are two columns of cards, each containing 15 examples. Each card has a small icon, a title, and a brief description.

Category	Example	Description	
General AI Applications	Q&A	Answer questions based on existing knowledge.	
	Grammar correction	Correct sentences into standard English.	
	Summarize for a 2nd grader	Translate difficult text into simpler concepts.	
	Text to command	Translate text into programmatic commands.	
	Natural language to Stripe API	Create code to call the Stripe API using natural language.	
	Parse unstructured data	Create tables from long form text.	
	Python to natural language	Explain a piece of Python code in human understandable terms.	
	Calculate Time Complexity	Find the time complexity of a function.	
	Advanced tweet classifier	Advanced sentiment detection for a piece of text.	
	Keywords	Extract keywords from a block of text.	
	Programming and Development	Grammar correction	Correct sentences into standard English.
		Natural language to OpenAI API	Create code to call the OpenAI API using natural language.
		English to other languages	Translate English text into French, Spanish, etc.
		SQL translate	Translate natural language to SQL queries.
		Classification	Classify items into categories via example.
Movie to Emoji		Convert movie titles into emoji.	
Translate programming languages		Translate from one programming language to another.	
Explain code		Explain a complicated piece of code.	
Factual answering		Guide the model towards factual answering.	
TL;DR summarization		Summarize text by adding a 'TL;DR' to the end.	
Python bug fixer		Find and fix bugs in source code.	
Spreadsheet creator		Create spreadsheets of various kinds of data.	
ML/AI language model tutor		Bot that answers questions about language models.	
Tweet classifier		Basic sentiment detection for a piece of text.	
SQL request		Create simple SQL queries.	
JavaScript to Python	Convert simple JavaScript expressions into Python code.		
Mood to color	Turn a text description into a color.		
Analogy maker	Create analogies. Modified from a community project.		
Micro horror story creator	Creates two to three sentence short horror stories.		
Notes to summary	Turn meeting notes into a summary.		
ESRB rating	Categorize text based upon ESRB ratings.		
Python bug fixer	Find and fix bugs in source code.		
JavaScript helper chatbot	Message-style bot that answers JavaScript questions.		
Science fiction book list maker	Create a list of items for a given topic.		
Airport code extractor	Extract airport codes from text.		
Extract contact information	Extract contact information from a block of text.		
Friend chat	Emulate a text message conversation.		
Write a Python docstring	An example of how to create a docstring for Python code.		
JavaScript one-line function	Turn a JavaScript function into a one-liner.		
Third-person converter	Converts first-person POV to the third-person.		
VR fitness idea generator	Create ideas for fitness and virtual reality games.		
Essay outline	Generates an outline for a research topic.		

# Q&A example with ChatGPT

 Q&A  
Answers Generation Conversation

Answer questions based on existing knowledge.

**Prompt**

I am a highly intelligent question answering bot. If you ask me a question that is rooted in truth, I will give you the answer. If you ask me a question that is nonsense, trickery, or has no clear answer, I will respond with "Unknown".

Q: What is human life expectancy in the United States?  
A: Human life expectancy in the United States is 78 years.

Q: Who was president of the United States in 1955?  
A: Dwight D. Eisenhower was president of the United States in 1955.

Q: Which party did he belong to?  
A: He belonged to the Republican Party.

Q: What is the square root of banana?  
A: Unknown

Q: How does a telescope work?  
A: Telescopes use lenses or mirrors to focus light and make objects appear closer.

Q: Where were the 1992 Olympics held?  
A: The 1992 Olympics were held in Barcelona, Spain.

Q: How many squigs are in a bonk?  
A: Unknown

Q: Where is the Valley of Kings?  
A:

## Sample response

The Valley of Kings is located in Luxor, Egypt.

## API request

python ▾ ⌂ Copy

```
1 import os
2 import openai
3
4 openai.api_key = os.getenv("OPENAI_API_KEY")
5
6 response = openai.Completion.create(
7     engine="text-davinci-002",
8     prompt="I am a highly intelligent question answering bot",
9     temperature=0,
10    max_tokens=100,
11    top_p=1,
12    frequency_penalty=0.0,
13    presence_penalty=0.0,
14    stop=["\n"]
15 )
```

# More examples with ChatGPT

 **TL;DR summarization**

Transformation Generation

Summarize text by adding a 'tl;dr:' to the end of a text passage. It shows that the API understands how to perform a number of tasks with no instructions.

**Prompt**

A neutron star is the collapsed core of a massive supergiant star, which had a total mass of between 10 and 25 solar masses, possibly more if the star was especially metal-rich.<sup>[1]</sup> Neutron stars are the smallest and densest stellar objects, excluding black holes and hypothetical white holes, quark stars, and strange stars.<sup>[2]</sup> Neutron stars have a radius on the order of 10 kilometres (6.2 mi) and a mass of about 1.4 solar masses.<sup>[3]</sup> They result from the supernova explosion of a massive star, combined with gravitational collapse, that compresses the core past white dwarf star density to that of atomic nuclei.

TL;dr

**Sample response**

A neutron star is the collapsed core of a massive supergiant star. These ultra-dense objects are incredibly fascinating due to their strange properties and their potential for phenomena such as extreme gravitational forces and a strong magnetic field.

 **Interview questions**

Generation

Create interview questions.

**Prompt**

Create a list of 8 questions for my interview with a science fiction author:

**Sample response**

1. What inspired you to write science fiction?
2. What themes or topics do you explore in your stories?
3. What are some of the unique elements that make science fiction stand out from other genres?
4. What challenges do you face when writing science fiction?
5. What are some of the most important aspects of a good science fiction story?
6. What do you think are the biggest challenges facing science fiction today?
7. What do you think the future of science fiction holds for readers and writers?
8. What advice would you give to aspiring science fiction authors?

# Examples with ChatGPT

 **SQL request**

Transformation Generation Translation

Create simple SQL queries.

**Prompt**

Create a SQL request to find all users who live in California and have over 1000 credits:

**Sample response**

```
SELECT * FROM users WHERE state='CA' AND credits > 1000;
```

 **Python bug fixer**

Code Generation

There's a number of ways of structuring the prompt for checking for bugs. Here we add a comment suggesting that source code is buggy, and then ask codex to generate a fixed code.

**Prompt**

```
##### Fix bugs in the below function

### Buggy Python
import Random
a = random.randint(1,12)
b = random.randint(1,12)
for i in range(10):
    question = "What is "+a+" x "+b+"? "
    answer = input(question)
    if answer == a*b
        print (Well done!)
    else:
        print("No.")

### Fixed Python
```

**Sample response**

```
import random
a = random.randint(1,12)
b = random.randint(1,12)
for i in range(10):
    question = "What is "+str(a)+" x "+str(b)+"? "
    answer = input(question)
    if answer == str(a*b):
        print ("Well done!")
    else:
```

# New learning paradigm: in-context learning

model

$$f_{\theta}$$

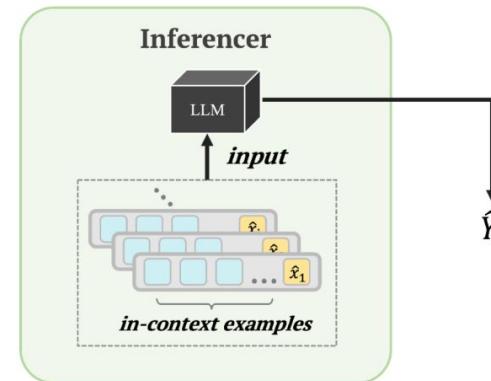
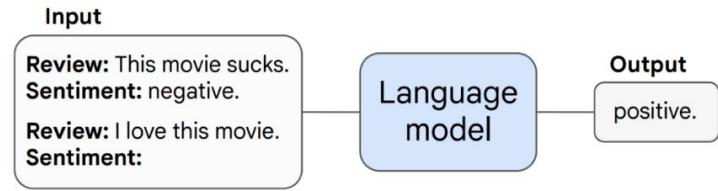


training objective/loss

$$\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$$

training dataset

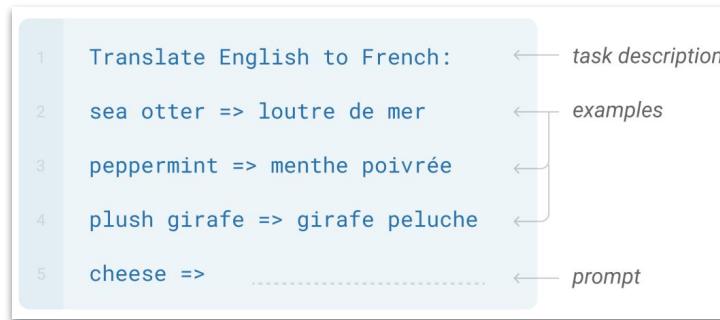
Supervised Learning



In-Context Learning

# Few-shot in-context learning

- **Few-shot:** In addition to the task description, the model sees a few examples of the task.
- **No fine-tuning,** GPT-3 doesn't update their parameters!



The screenshot shows a Q&A interface with tabs for "Answers", "Generation", and "Conversation". The "Answers" tab is selected. The interface includes a "Prompt" section and a "Sample response" section.

**Prompt:**

I am a highly intelligent question answering bot. If you ask me a question that is rooted in truth, I will give you the answer. If you ask me a question that is nonsense, trickery, or has no clear answer, I will respond with "Unknown".

**task description**

**task examples**

**task prompt**

**Sample response:**

The Valley of Kings is located in Luxor, Egypt. ← **GPT-3 output**

Answer questions based on existing knowledge.

Q: What is human life expectancy in the United States?  
A: Human life expectancy in the United States is 78 years.

Q: Who was president of the United States in 1955?  
A: Dwight D. Eisenhower was president of the United States in 1955.

Q: Which party did he belong to?  
A: He belonged to the Republican Party.

Q: What is the square root of banana?  
A: Unknown

Q: How does a telescope work?  
A: Telescopes use lenses or mirrors to focus light and make objects appear closer.

Q: Where were the 1992 Olympics held?  
A: The 1992 Olympics were held in Barcelona, Spain.

Q: How many squigs are in a bonk?  
A: Unknown

Q: Where is the Valley of Kings?  
A:

# ChatGPT is a powerful language model!

The screenshot shows the 'Examples' section of the ChatGPT interface. At the top, there's a navigation bar with 'Overview', 'Documentation', and 'Examples'. Below the navigation is a search bar labeled 'Search...' and a dropdown menu for 'All categories'. The main area is titled 'Explore what's possible with some example applications'.

The examples are listed in two columns:

Category	Description
Q&A	Answer questions based on existing knowledge.
Grammar correction	Corrects sentences into standard English.
Summarize for a 2nd grader	Translates difficult text into simpler concepts.
Natural language to OpenAI API	Create code to call to the OpenAI API using natural language.
Text to command	Translate text into programmatic commands.
English to other languages	Translates English text into French, Spanish, etc.
Natural language to Stripe API	Create code to call the Stripe API using natural language.
SQL translate	Translate natural language to SQL queries.
Parse unstructured data	Create tables from long form text.
Classification	Classify items into categories via example.
Python to natural language	Explain a piece of Python code in human understandable terms.
Movie to Emoji	Convert movie titles into emoji.
Calculate Time Complexity	Find the time complexity of a function.
Translate programming languages	Translate from one programming language to another.
Advanced tweet classifier	Advanced sentiment detection for a piece of text.
Explain code	Explain a complicated piece of code.
Keywords	Extract keywords from a block of text.
Factual answering	Guide the model towards factual answering.
TL;DR summarization	Summarize text by adding a 'tl;dr' to the end.
Python bug fixer	Find and fix bugs in source code.
Spreadsheet creator	Create spreadsheets of various kinds of data.
JavaScript helper chatbot	Message-style bot that answers JavaScript-related questions.
ML/AI language model tutor	Bot that answers questions about language models.
Science fiction book list maker	Create a list of items for a given topic.
Tweet classifier	Basic sentiment detection for a piece of text.
Airport code extractor	Extract airport codes from text.
SQL request	Create simple SQL queries.
Extract contact information	Extract contact information from a block of text.
JavaScript to Python	Convert simple JavaScript expressions into Python code.
Friend chat	Emulate a text message conversation.
Mood to color	Turn a text description into a color.
Write a Python docstring	An example of how to create a docstring for a function.
JavaScript one line function	Turn a JavaScript function into a one liner.
Analogy maker	Create analogies. Modified from a community project.
Micro horror story creator	Creates two to three sentence short horror stories.
Third-person converter	Converts first-person POV to the third-person.
Notes to summary	Turn meeting notes into a summary.
VR fitness idea generator	Create ideas for fitness and virtual reality games.
ESRB rating	Categorize text based upon ESRB ratings.
Essay outline	Generate an outline for a research topic.

# Let's play a game!

This year, I am going to do an internship in \_\_\_\_\_

Queen Mary Hospital, HSBC, Google, Amazon

Majoring in computer science, this year, I am going to do an internship in \_\_\_\_\_



Queen Mary Hospital, HSBC, Google, Amazon

# ChatGPT auto-completes your prompt

The screenshot shows a user interface for a question-and-answer service. At the top left is a purple square icon with a white question mark. To its right is the text "Q&A" in bold. Below that are three tabs: "Answers", "Generation", and "Conversation". A horizontal line separates this header from the main content area.

The main content area has two sections: "Prompt" on the left and "Sample response" on the right. The "Prompt" section contains a large gray box with the following text:

I am a highly intelligent question answering bot. If you ask me a question that is rooted in truth, I will give you the answer. If you ask me a question that is nonsense, trickery, or has no clear answer, I will respond with "Unknown".

Below this, several questions and their answers are listed:

- Q: What is human life expectancy in the United States?  
A: Human life expectancy in the United States is 78 years.
- Q: Who was president of the United States in 1955?  
A: Dwight D. Eisenhower was president of the United States in 1955.
- Q: Which party did he belong to?  
A: He belonged to the Republican Party.
- Q: What is the square root of banana?  
A: Unknown
- Q: How does a telescope work?  
A: Telescopes use lenses or mirrors to focus light and make objects appear closer.
- Q: Where were the 1992 Olympics held?  
A: The 1992 Olympics were held in Barcelona, Spain.
- Q: How many squigs are in a bonk?  
A: Unknown
- Q: Where is the Valley of Kings?  
A:

The "Sample response" section on the right contains a green box with the text: "The Valley of Kings is located in Luxor, Egypt."

# Generative language model

I am going to do an internship in Google



# Making the dice



# bag of words

- 1 Belief
  - 2 Evidence
  - 3 Reason
  - 4 Claim
  - 5 Think
  - 6 Justify
  - 7 Also
  - ...
  - 99 Therefore
  - 100 Google



# Vocabulary

# Generative language model

|



|

# Generative language model

I am



am

# Generative language model

I am going



going

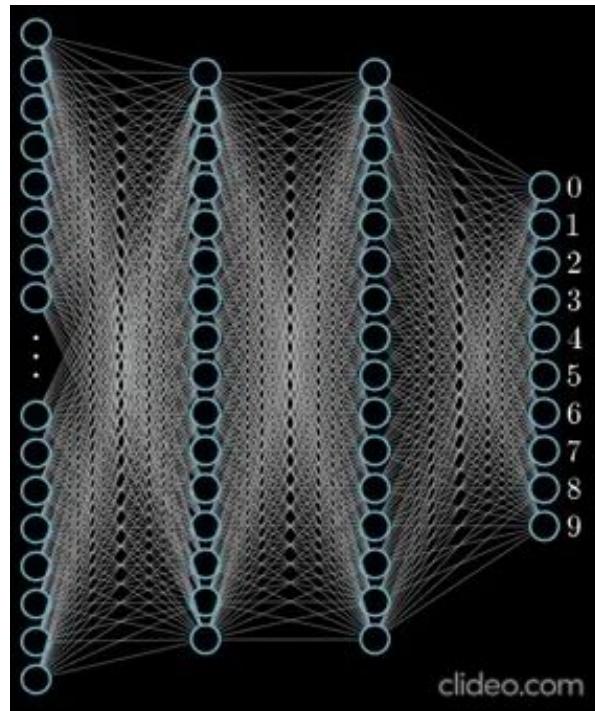
# Generative language model

I am going to do an internship in Google

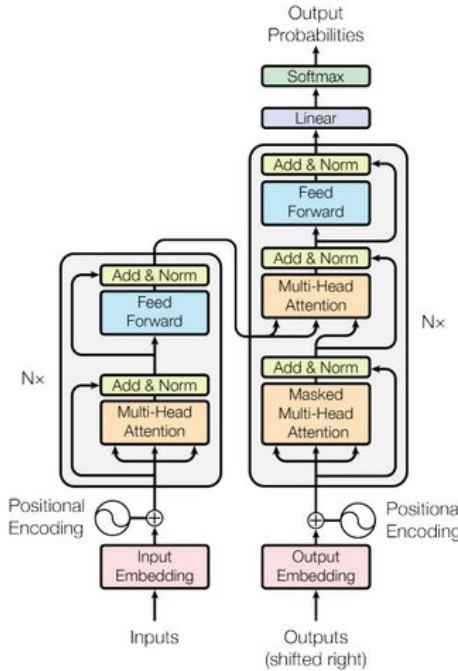


Google

# Neuralize the dice!

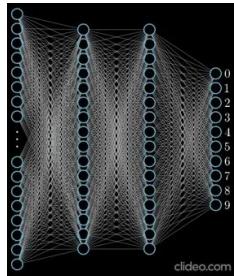


# Neural Networks (e.g. Transformers)



# Neural network language models

I am going to do an internship in Google



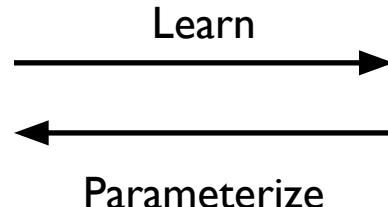
Google

# Language models, and how to build it

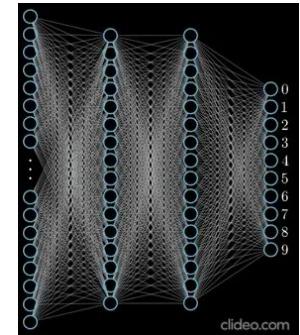
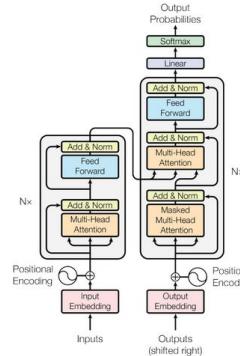


Dice, and how do we roll them  
(probabilistic model)

$$p(\mathbf{x}) = \prod_i p(x_i | \mathbf{x}_{<i})$$



Transformers, neural networks and many others  
(powerful functions)



# First problem — the language modeling problem

Given a finite vocabulary

$$\mathcal{V} = \{\text{belief, evidence, reason, claim, ... Google, therefore}\}$$

We have a set of sentences

< s > I am going to an internship in Google < /s >  
< s > an internship in Google < /s >  
< s > I am going going < /s >  
< s > Google is am < /s >  
< s > internship is going < /s >

Can we learn a “model” for this “generative process”? We need to “learn” a probability distribution:

$$p(x_1, x_2, \dots x_n)$$

Learn from what we've seen

# The language modeling problem

Given a *training sample* of example sentences, we need to “learn” a probabilistic model that assigns probabilities to every possible string:

$$p(<\text{s}> \text{ I am going to an internship in Google } </\text{s}>) = 10^{-12}$$

$$p(<\text{s}> \text{ an internship in Google } </\text{s}>) = 10^{-8}$$

$$p(<\text{s}> \text{ I am going going } </\text{s}>) = 10^{-15}$$

...

# What is a language model?

- A probabilistic model of a sequence of words  $x_1, x_2, \dots, x_n$

A language model consists of

- A finite set  $\mathcal{V} = \{\text{the, dog, laughs, saw, barks, cat, ...}\}$

# What is a language model?

- A probabilistic model of a sequence of words  $x_1, x_2, \dots, x_n$

A language model consists of

- A finite set  $\mathcal{V} = \{\text{the, dog, laughs, saw, barks, cat, ...}\}$

A sentence in the language is a sequence of words

$$x_1, x_2, \dots, x_n$$

For example

the dog barks STOP

the cat saw the dog STOP

Define  $\mathcal{V}^{\dagger}$  to be the set of all sentences with the vocabulary  $\mathcal{V}$

# What is a language model?

- A probabilistic model of a sequence of words  $x_1, x_2, \dots, x_n$

A language model consists of

- A finite set  $\mathcal{V} = \{\text{the, dog, laughs, saw, barks, cat, ...}\}$
- A probability distribution over sequences of words  $p(x_1, x_2, \dots, x_n)$  such that:
  1. For any  $\langle x_1 \dots x_n \rangle \in \mathcal{V}^\dagger$ ,  $p(x_1, x_2, \dots, x_n) \geq 0$
  2. In addition,  $\sum_{\langle x_1 \dots x_n \rangle \in \mathcal{V}^\dagger} p(x_1, x_2, \dots, x_n) = 1$

# Assign a probability to a sentence

Application of language models:

$P(\text{"I am going to school"}) > P(\text{"I are going to school"})$

Grammar Checking

I had some coffee this morning.

Machine translation

$P(\text{"我今早喝了一些咖啡"}) > P(\text{"我今早吃了一些咖啡"})$

$P(\text{"Can we put an elephant into the refrigerator? No, we can't."}) > P(\text{"Can we put an elephant into the refrigerator? Yes, we can."})$

Question Answering

# A (very bad) language model

Number of times the sentence  $x_1 \dots x_n$  is seen in the training corpus

$$c(x_1 \dots x_n)$$

Total number of sentences in the training corpus  $N$

$$p(x_1 \dots x_n) = \frac{c(x_1 \dots x_n)}{N}$$

Why this is very bad?