Course Overview

CS 5539: Advanced Topics in Natural Language Processing

https://shocheen.github.io/courses/advanced-nlp-fall-2024

THE OHIO STATE UNIVERSITY

Slide Credits: Daniel Kashabi, Arman Cohen, Yejin Choi

Logistics

- Instructor: Sachin Kumar

- Time: Mondays, 1 – 2.45 pm
- Location: DL 317

- Office Hours: Thursdays, 2-3 pm, DL 581 or by appointment

First week attendance

- Please write your full name, OSU email, and mark if you are waitlisted.

Course structure

- This a seminar course.
  - The course is primarily based on presentations & discussion of latest research papers

Course structure

- This a seminar course.
  - The course is primarily based on presentations & discussion of latest research papers

- Main goals of the course:
  - Get students up to speed with the latest developments in NLP
  - Help students build or improve research skills (from literature reviews and critiquing prior work, to brainstorming ideas and implementing them).

Course structure

- This a seminar course.
  - The course is primarily based on presentations & discussion of latest research papers

- Main goals of the course:
  - Get students up to speed with the latest developments in NLP
  - Prepare students to perform cutting-edge research in NLP
  - Help students build or improve research skills (from literature reviews and critiquing prior work, to brainstorming ideas and implementing them).

- All students are expected to participate in the class regularly and participate in presentations and discussions

Preliminaries: What I Expect From You

- Comfortable with machine learning.
  - **Modeling:** linear models, classification, neural networks
  - **Training:** gradient descent, backpropagation, train/test/dev splits
  - **Measuring quality:** generalization: overfitting vs underfitting

- Familiarity with NLP is helpful, though not necessary.

- Being open to reading [somewhat esoteric] papers and presenting their gist to the class.

Homework to test foundational knowledge

- Later today, a homework will be released on Canvas and will be due mid next week (Wednesday, September 4).
    - The only homework in this course.

- It is intended to measure your understanding of the foundational concepts of ML/NLP.

- This is to make sure that when coming in, you know all the pre-requisites needed for the class.

Course structure - Resources

- No required textbook. But if you are interested in textbooks or book chapters:
  - Natural Language Processing with Transformers https://transformersbook.com/
  - A Primer on Neural Network Models for Natural Language Processing. https://u.cs.biu.ac.il/~yogo/nnlp.pdf
  - On the Opportunities and Risks of Foundation Models https://arxiv.org/pdf/2108.07258.pdf
- We will be reading research papers from premier conferences in the field

  E.g., ACL, EMNLP, NAACL, ICLR, NeurIPS, ICML, …

Questions so far?

Class Structure

- The class will be **in-person**.
- Each session will involve **the presentation/discussion** of recent important papers on NLP / Language Models.
- The course also involves **a project**.

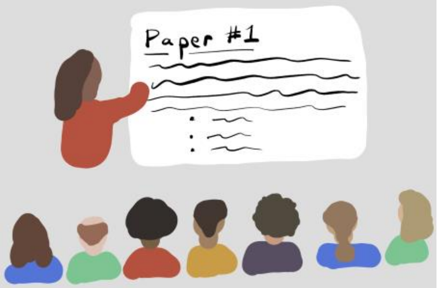
Class Presentations

- Role-based presentation

Role-Playing Paper-Reading Seminars

Alec Jacobson and Colin Raffel
March 17th, 2021
colinraffel.com/blog

https://colinraffel.com/blog/role-playing-seminar.html

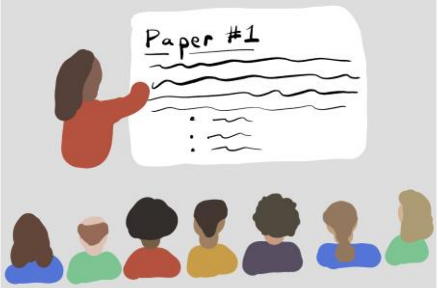


- Role-based presentation vs.
  - Many students **cooperatively** present a paper.
  - Each subgroup of students takes a specific **"role"**.
  - The "role" defines **the lens** through which you read/present a paper.
- One-to-Many presentations
  - A single (subgroup of) student(s) presenting a paper to the class.
- Pro:
  - Easy division of labor
- Cons:
  - Too much work for one person
  - Audience easy to disengage

- Role-based presentation
  - Many students **cooperatively** present a paper.
  - Each subgroup of students takes a specific **"role"**.

Role: Stakeholder ✍️

Act as if you're the author of this paper. Try to <u>sell</u> it!

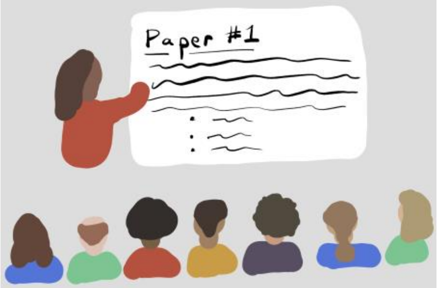


- One-to-Many presentations
  - A single (subgroup of) student(s) presenting a paper to the class.
- Pro:
  - Easy division of labor
- Cons:
  - Too much work for one person
  - Audience easy to disengage

- Role-based presentation
  - Many students **cooperatively** present a paper.
  - Each subgroup of students takes a specific **"role"**.

Role: Scientific Reviewer 🔎

Do a complete conference-style <u>critical</u> peer-review of the paper.

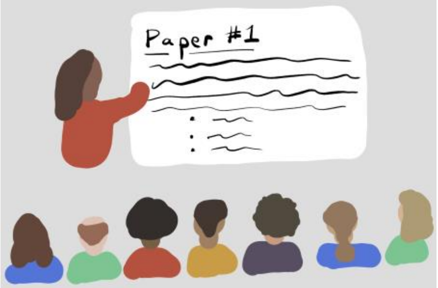


- One-to-Many presentations
  - A single (subgroup of) student(s) presenting a paper to the class.
- Pro:
  - Easy division of labor
- Cons:
  - Too much work for one person
  - Audience easy to disengage

- Role-based presentation
  - Many students **cooperatively** present a paper.
  - Each subgroup of students takes a specific **"role"**.

Role: Archaeologist 🏺

Determine the [prior and recent] <u>literature</u> that inspired and was inspired by this work.

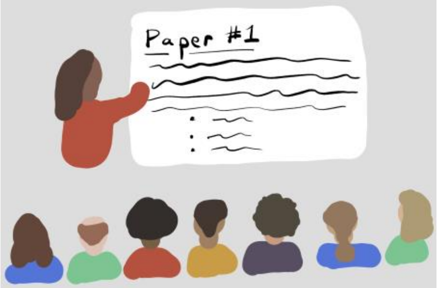


- One-to-Many presentations
  - A single (subgroup of) student(s) presenting a paper to the class.
- Pro:
  - Easy division of labor
- Cons:
  - Too much work for one person
  - Audience easy to disengage

- Role-based presentation
  - Many students **cooperatively** present a paper.
  - Each subgroup of students takes a specific **“role”**.

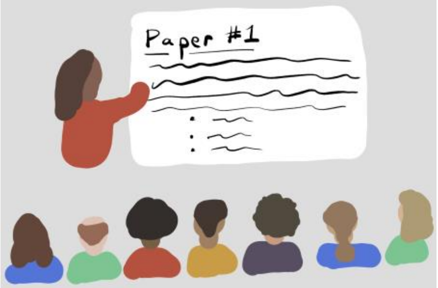
Role: Visionary

Propose an imaginary follow-up -- research project or a new application.



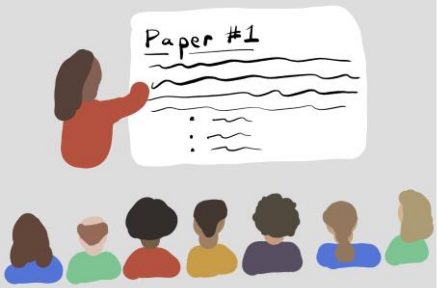
- One-to-Many presentations
  - A single (subgroup of) student(s) presenting a paper to the class.
- Pro:
  - Easy division of labor
- Cons:
  - Too much work for one person
  - Audience easy to disengage

- Role-based presentation
  - Many students **cooperatively** present a paper.
  - Each subgroup of students takes a specific **"role"**.
  - Students **rotate** "roles" each week.



- One-to-Many presentations
  - A single (subgroup of) student(s) presenting a paper to the class.
- Pro:
  - Easy division of labor
- Cons:
  - Too much work for one person
  - Audience easy to disengage

- Role-based presentation
  - Many students **cooperatively** present a paper.
- Pro:
  - More engagement
  - Distributed and less workload
  - Present more frequently
- Cons:
  - Need to manage role assignment



- One-to-Many presentations
  - A single (subgroup of) student(s) presenting a paper to the class.
- Pro:
  - Easy division of labor
- Cons:
  - Too much work for one person
  - Audience easy to disengage
  - Present a 1-2 times only.

Role-Based Presentation

- We will discuss two (thematically related) papers each week.
- Each member of the presenting group will be given a random role every week.
  - The presenters will be assigned at least 10 days before the class.
- Each role has a time budget:
  - ~15-17 mins for Stakeholder ✍️
  - ~10 mins for the rest of the (three) roles
- Each paper will take around ~45-47 minutes (~10 min break between two presentations)

Non-presenter Activity

- **Before the class: All students need to read the 2 papers**
  - Students who are not presenting, need to prepare at least one question/thought about each paper:
    - Could be anything you are confused about or something you'd like to hear discussed more, or an open-ended question
  - Submit your questions the night before the class (due midnight EST)
    - Where? TBD
  - We will use these questions partly as discussion points
    - Avoid generic questions/statements (e.g., What is their learning rate? How long did they train? Didn't understand their intro)
    - Aim to be probing, analytical, and thought-provoking by offering specific critical comments or questions.
- **During the class:** come to class ready to participate in the discussions.
  - You may come up with other questions in the class as the paper is being presented

Questions so far?

Attendance

- You can miss 3 sessions. Drop me a note before the class.
  - If you're "non-presenting", that's easy!
  - If you're "presenting", that's a bit complicated:
    - Find someone willing to swap presentations with.
    - Create the presentation for that role and find someone else to present.

- If you have any **COVID symptoms**, skip the class.
  - Does not count toward your 3 sessions.
  - Drop me a note before the class.

Course structure

- **After the class**
  - Quiz: At the conclusion some of the class session (not all), a quiz may be distributed to assess understanding of the assigned paper and key discussion points covered during the session.
  - These are due the day after the class

Guidelines for inclusive discussions

- This is a **discussion-based course**, everyone should feel very welcome to participate in discussions and share their thoughts and opinions.
- Example guidelines for promoting inclusive discussions:
  - Be respectful and mindful of different opinions
  - Try not to interrupt others, wait for them to finish
  - Acknowledge that there are people with different expertise in the room
  - Be positive, constructive, and polite

https://cse.ucsd.edu/sites/cse.ucsd.edu/files/Diversity/Inclusive_Seminar__LONG_.pdf

Class Project

- Group projects (team size = 2 to 3 students)
  - 3 students are allowed for projects with a larger proposed scope

- What is the goal of the final project?
  - Conduct research on a specific NLP problem and submit a written report.

    Examples of possible projects
    - A novel investigation of existing methods to better understand their limitation or capabilities
    - Extending, training or fine-tuning an existing model for a new task, application, or domain
    - Exploratory projects on providing some insights about a specific modeling approach or a specific NLP problem/task

Class project and timeline

September 9
Form teams

September 30
Project proposal

November 4
Progress report

December 2
Final Presentation

December 12
Project report

Class project and timeline

- Project milestones:
  - **September 9:** Form teams (just send me an email and cc your team members)

Class project and timeline

- Project milestones:
  - **September 9:** Form teams (just send me an email and cc your team members)
  - **September 30:** project proposal (1-2 page)
    - Should describe what is the main goal of the project, the proposed research, and how it connects to existing work in the field
    - You will receive feedback in a week.

Class project and timeline

- Project milestones:
  - **September 9:** Form teams (just send me an email and cc your team members)
  - **September 30:** project proposal (1-2 page)
    - Should describe what is the main goal of the project, the proposed research, and how it connects to existing work in the field
    - You will receive feedback in a week
  - **November 4:** progress report (2 pages)
    - Describe the main problem, project goal and related work, what has been done so far, any initial results, and the plan continuing the project.
    - Receive feedback in a week

Class project and timeline

- Project milestones:
  - **September 9:** Form teams (just send me an email and cc your team members)
  - **September 30:** project proposal (1-2 page)
    - Should describe what is the main goal of the project, the proposed research, and how it connects to existing work in the field
    - You will receive feedback in a week
  - **November 4:** progress report (2 pages)
    - Describe the main problem, project goal and related work, what has been done so far, any initial results, and the plan continuing the project.
    - Receive feedback in a week
  - **December 2:** project presentations
    - Projects will be presented in class

Class project and timeline

- Project milestones:
  - **September 9:** Form teams (just send me an email and cc your team members)
  - **September 30:** project proposal (1-2 page)
    - Should describe what is the main goal of the project, the proposed research, and how it connects to existing work in the field
    - You will receive feedback in a week
  - **October 28:** progress report (2 pages)
    - Describe the main problem, project goal and related work, what has been done so far, any initial results, and the plan continuing the project.
    - Receive feedback in a week
  - **December 2:** project presentations
    - Projects will be presented in class
  - **December 12:** Final project report (6-8 pages)
    - The format of this report should be very similar to a conference paper
      - E.g., should include motivation, related work, proposed approach, results, and discussion

Grading

- Foundations Homework (5%)
- Paper presentation and discussions (40%)
  - 25% Paper presentations
  - 10% Active participation in discussions
  - 5% question submissions and quiz
- Project (55%)
  - 5% Proposal
  - 10% Progress report
  - 10% Final presentation
  - 30% Final report + code
- If you’re engaged in class presentations/discussions and on top of your project, you should not be worried about the grade.

Questions?

Question for You

- What is the best medium of communication for us? (Teams? Email? Canvas? Piazza?)
    - Announcements, role assignments, cancellations, broad discussions, etc.
- How many people have [used/read the paper for] X?

Generative AI Policy

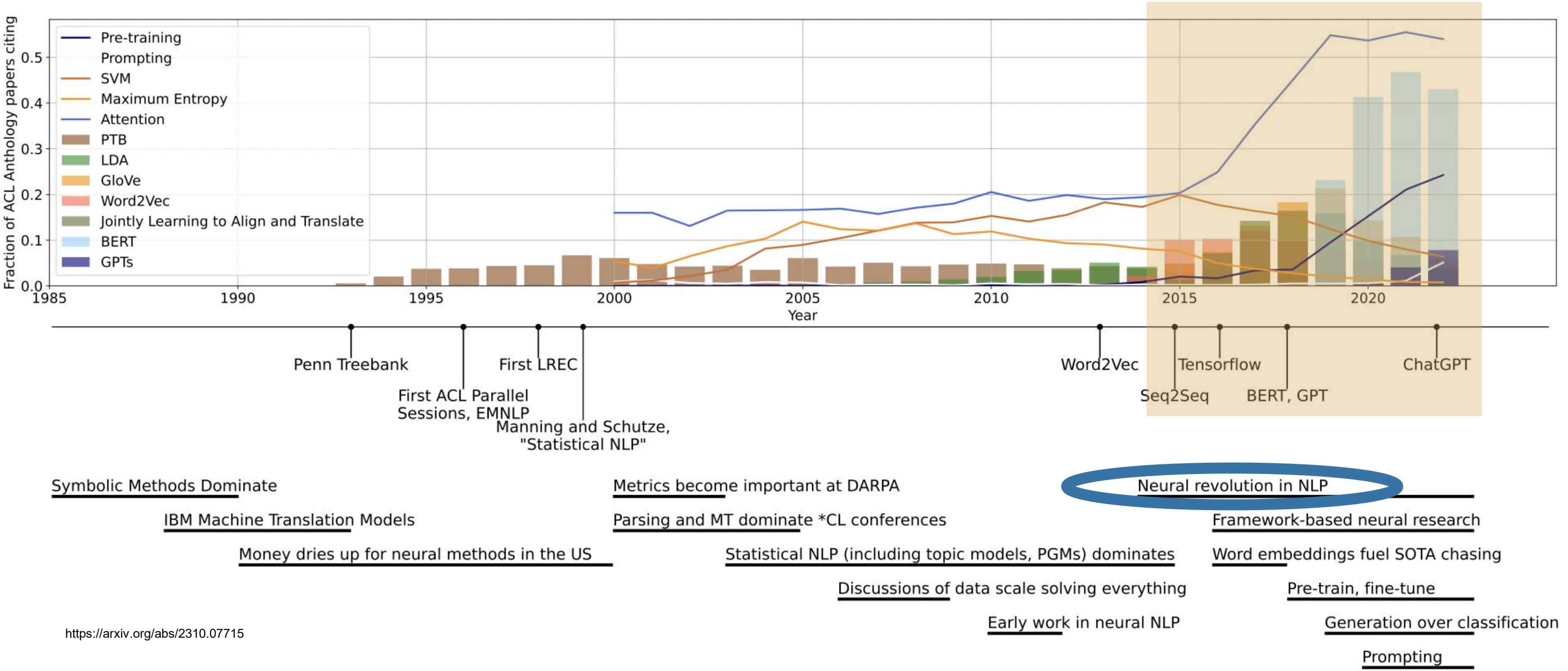
- You may use generative AI tools such as Co-Pilot and ChatGPT, as you would use a human collaborator. **This means that you may NOT directly ask generative AI tools for answers or copy solutions. You're required to acknowledge generative AI tools as collaborators and include a paragraph describing how you used the tool.** The use of generative AI tools to substantially complete an assignment is prohibited and will result in honor code violations.

Questions?

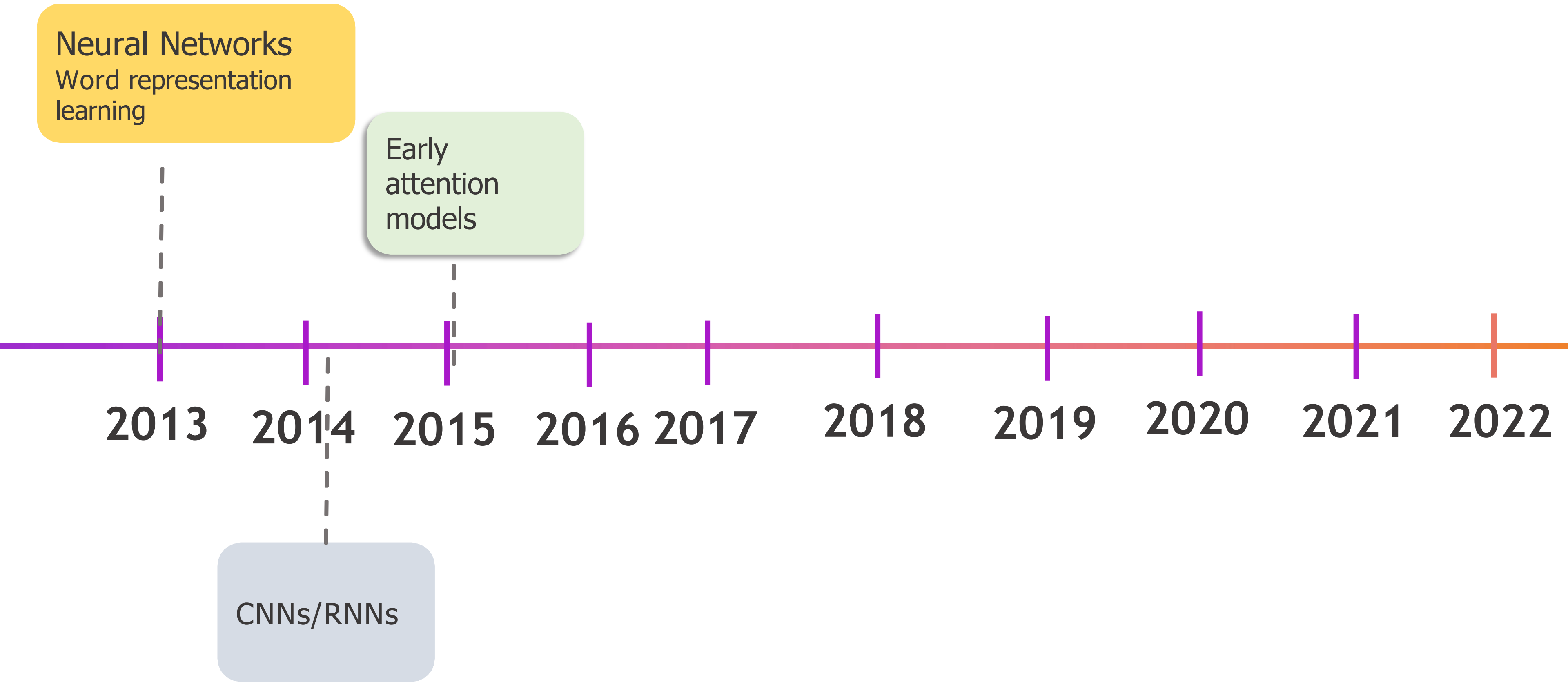
What’s NLP?

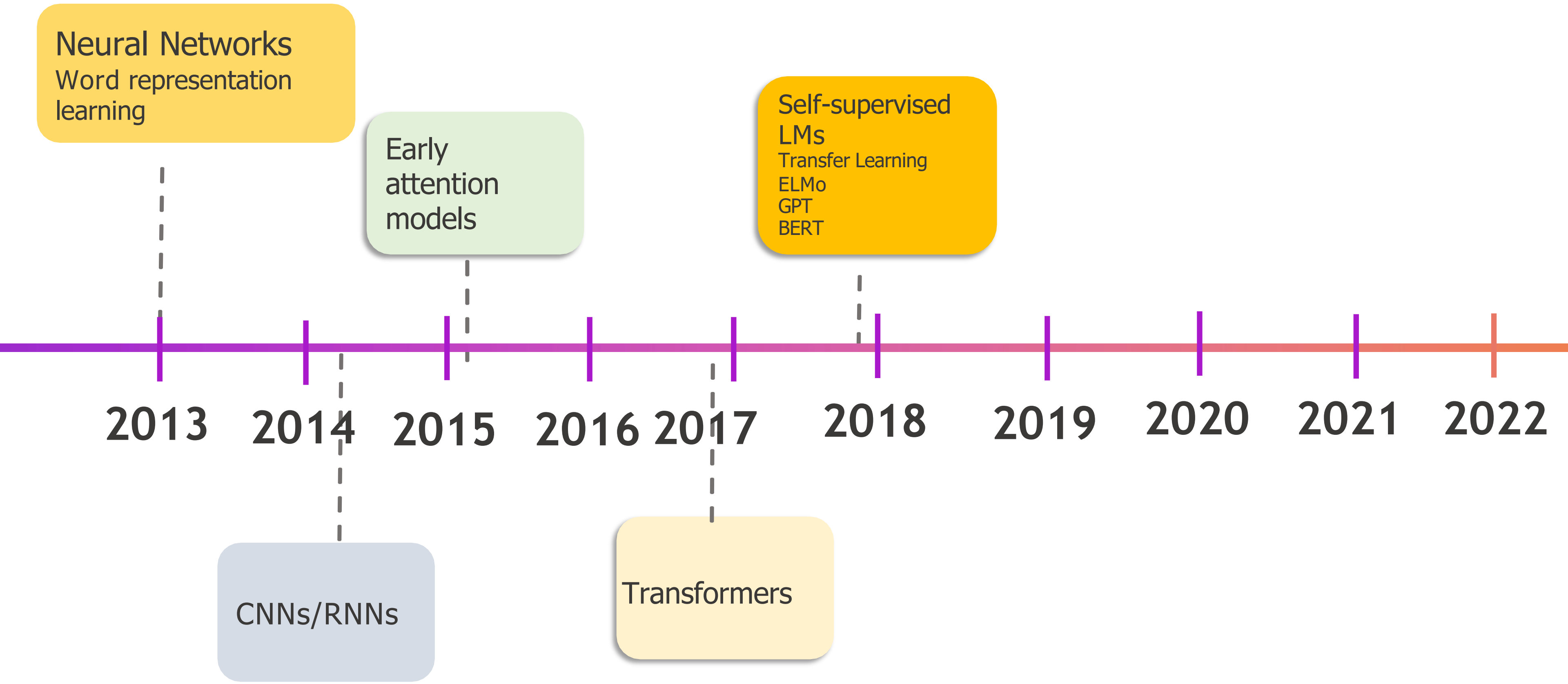
- Fundamental goal: deep understand of broad language
  - Not just string processing or keyword matching
- End systems that we want to build:
  - **Simple:** spelling correction, text categorization…
  - **Complex:** speech recognition, machine translation, information extraction, sentiment analysis, question answering…
  - **Unknown:** human-level comprehension (is this just NLP?)

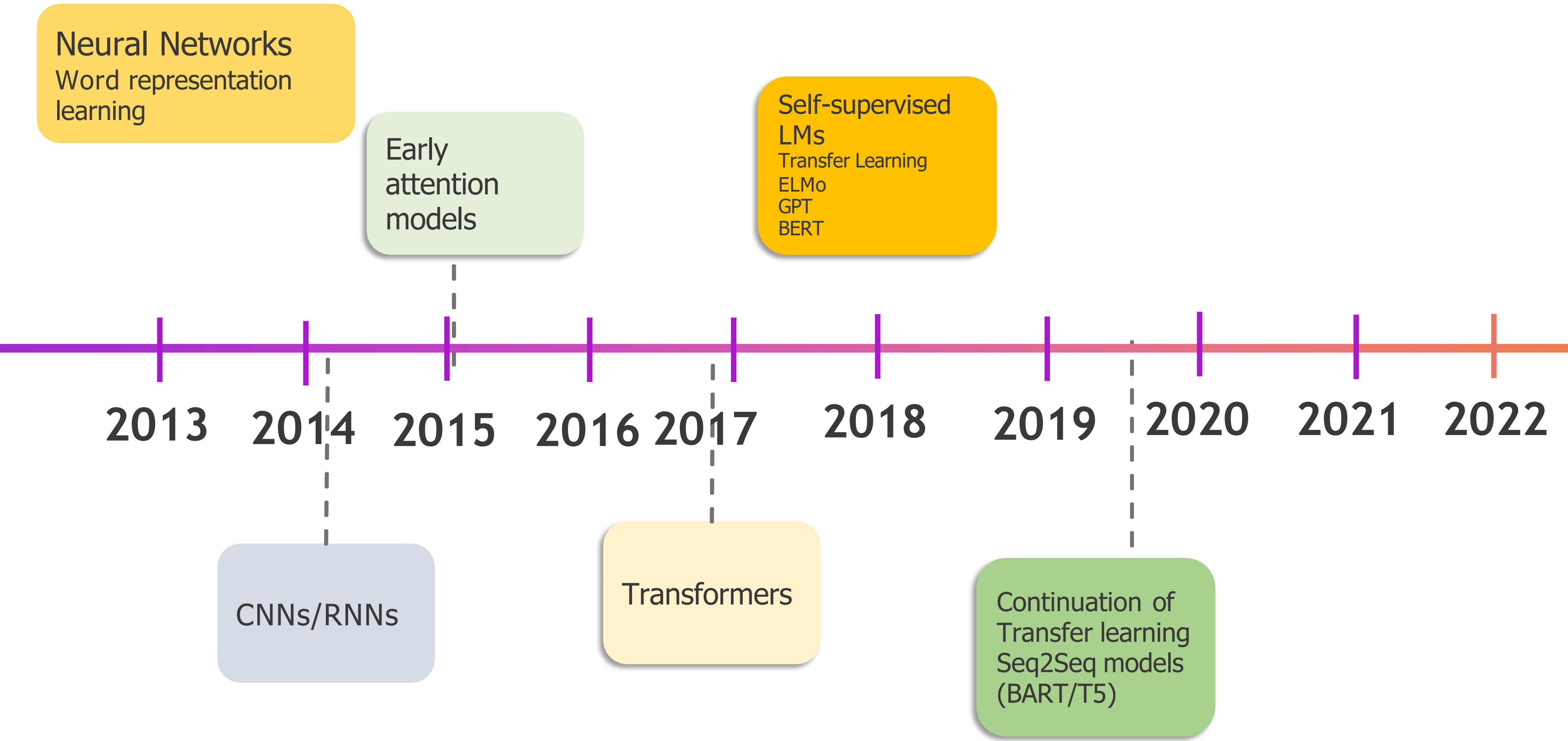
History of NLP Research

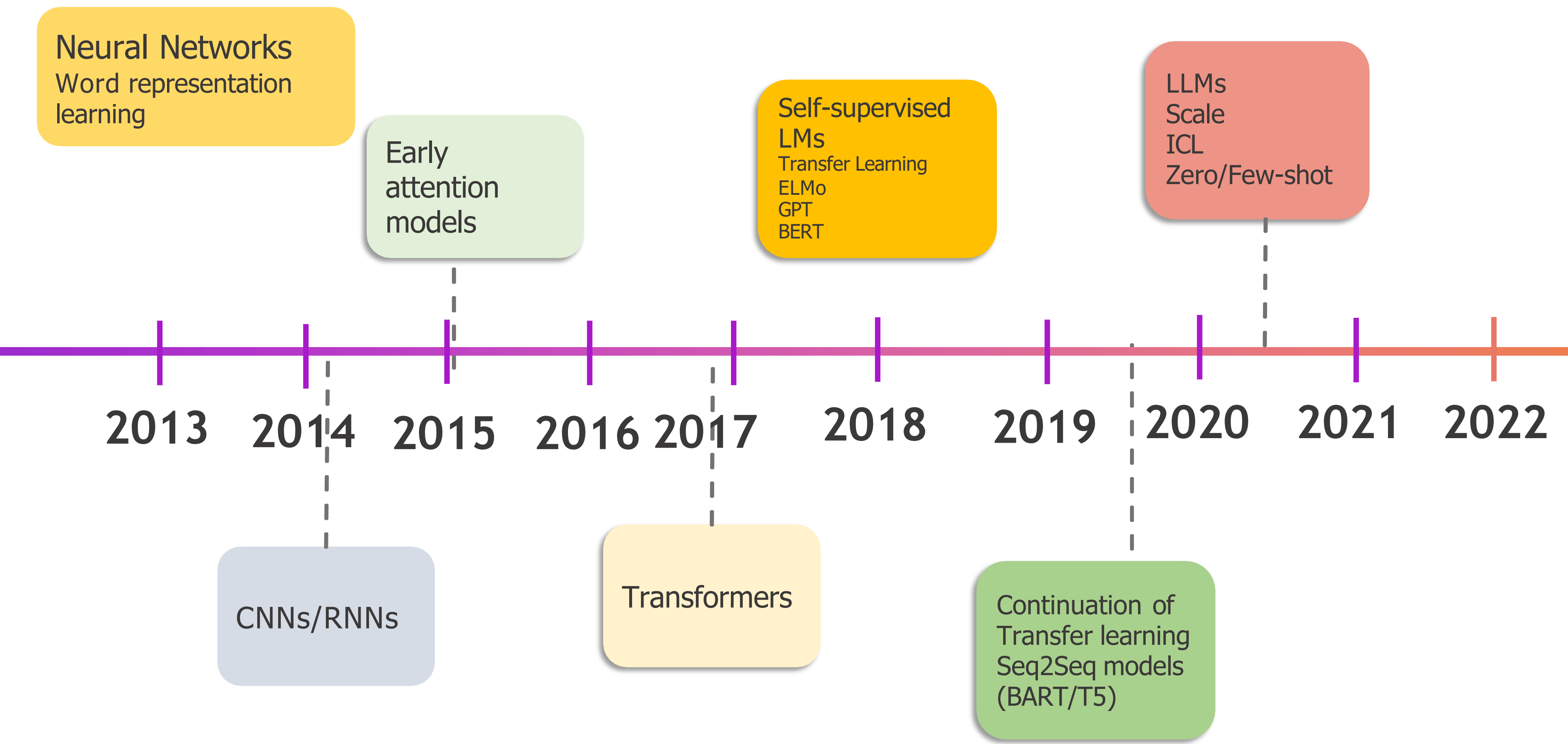


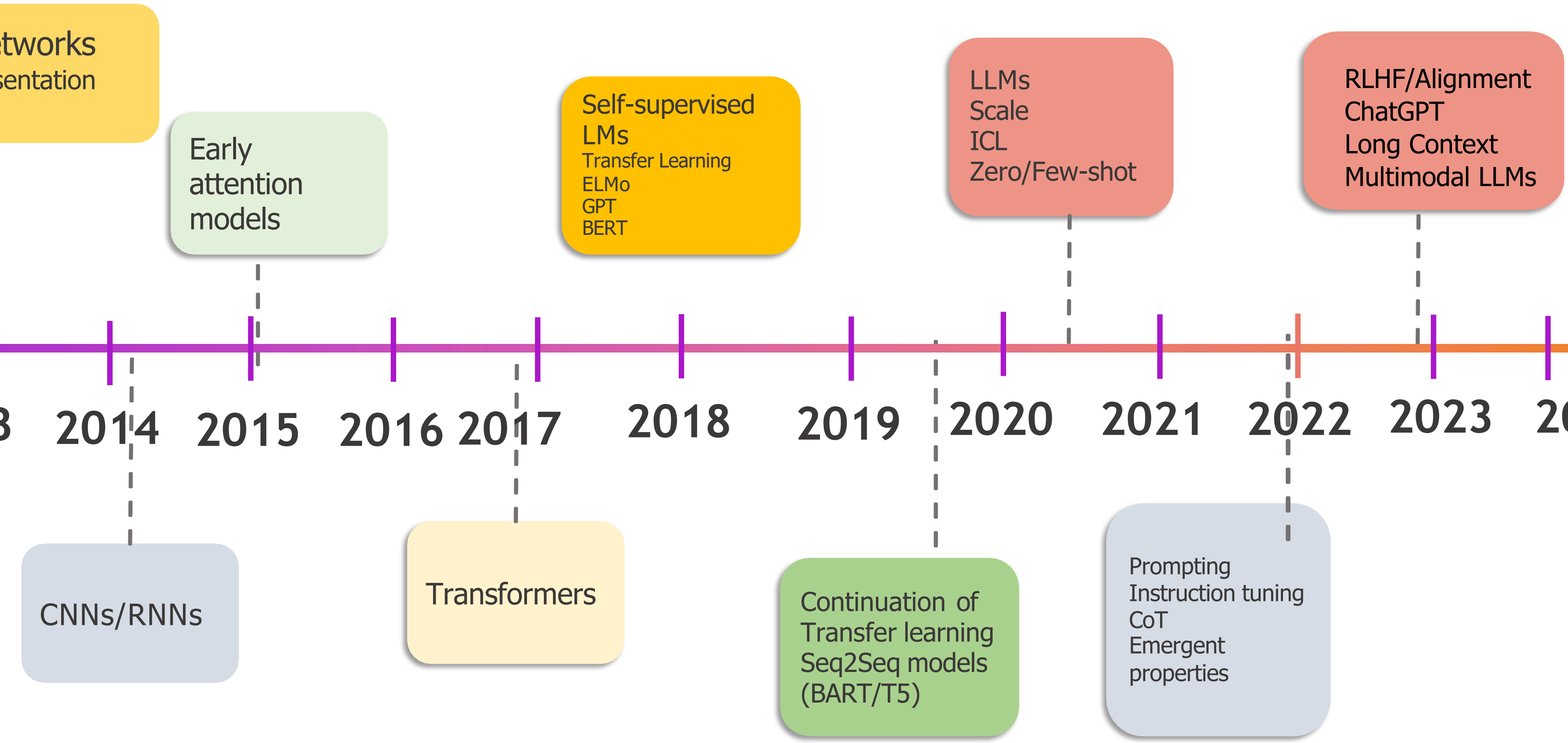
https://arxiv.org/abs/2310.07715


Neural Networks
Word representation learning
Early attention models
CNNs/RNNs
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022

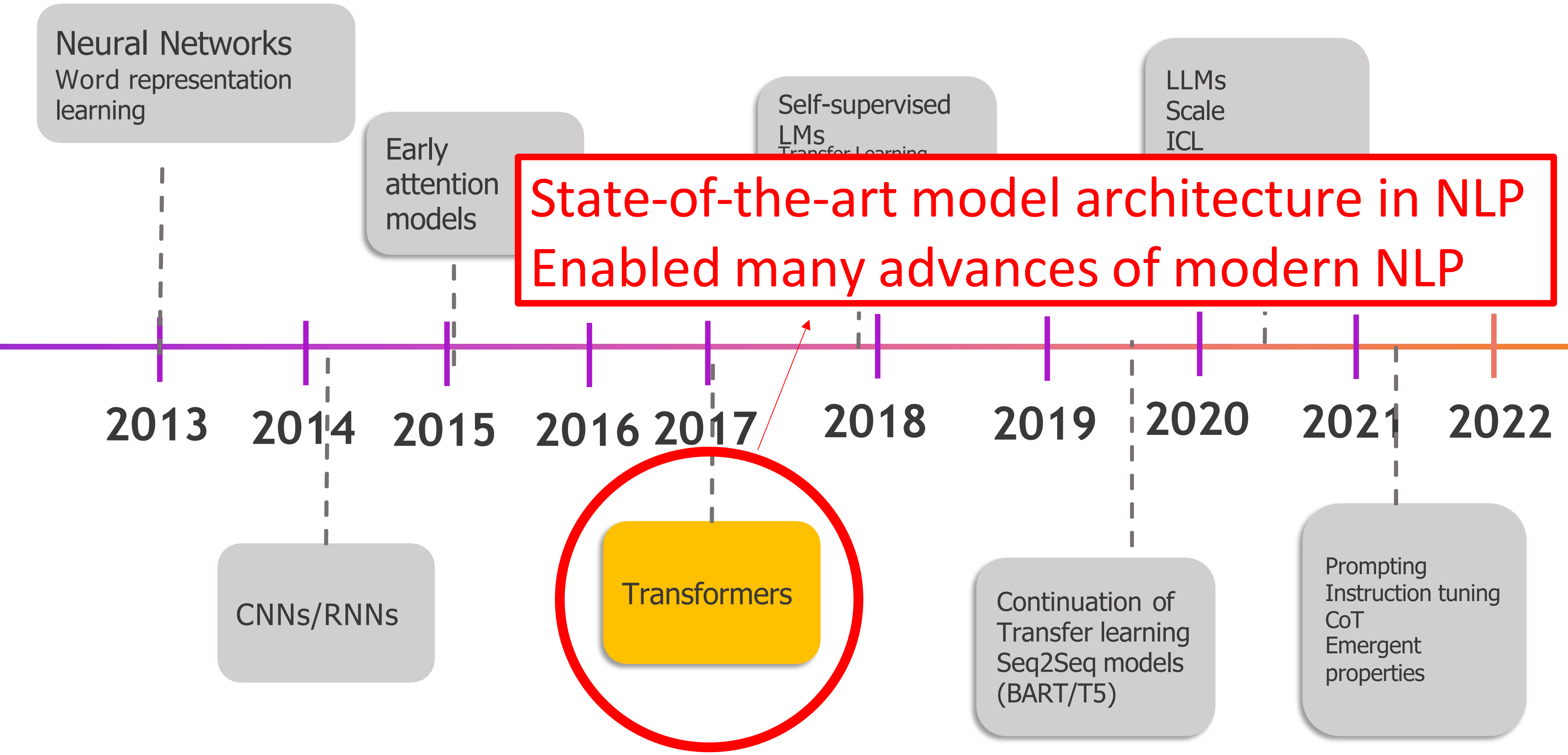

Neural Networks
Word representation learning
CNNs/RNNs
Early attention models
Transformers
Self-supervised LMs
Transfer Learning
ELMo
GPT
BERT
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022


Neural Networks
Word representation learning
Early attention models
Self-supervised LMs
Transfer Learning
ELMo
GPT
BERT
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
CNNs/RNNs
Transformers
Continuation of Transfer learning
Seq2Seq models (BART/T5)

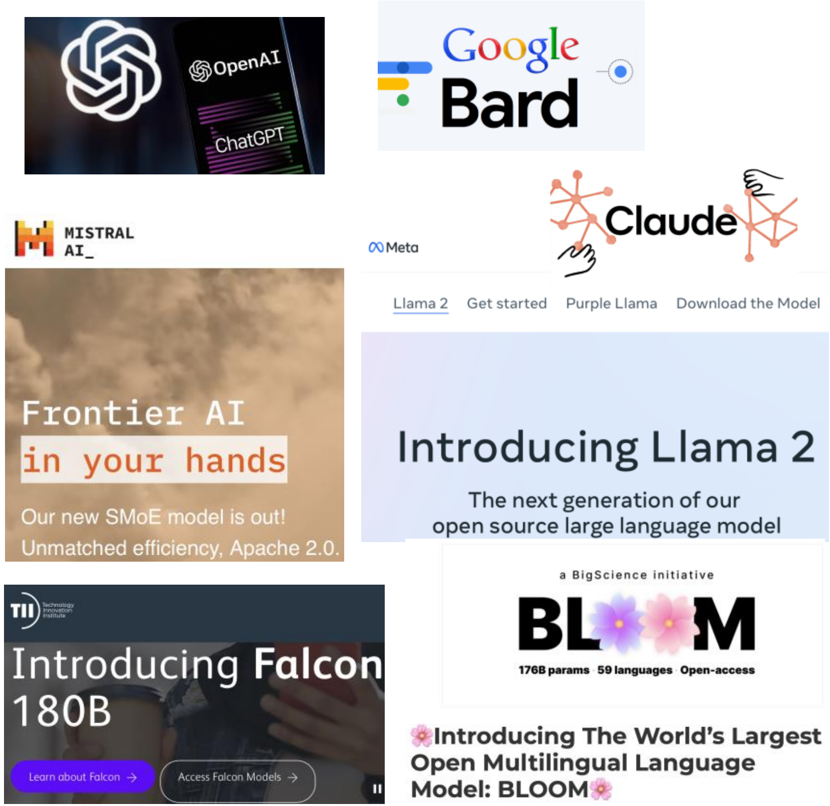

Neural Networks
Word representation learning
Early attention models
Self-supervised LMs
Transfer Learning
ELMo
GPT
BERT
LLMs
Scale
ICL
Zero/Few-shot
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
CNNs/RNNs
Transformers
Continuation of Transfer learning
Seq2Seq models
(BART/T5)



tworks
sentation
Early attention models
Self-supervised LMs
Transfer Learning
ELMo
GPT
BERT
LLMs
Scale
ICL
Zero/Few-shot
RLHF/Alignment
ChatGPT
Long Context
Multimodal LLMs
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
CNNs/RNNs
Transformers
Continuation of Transfer learning
Seq2Seq models
(BART/T5)
Prompting
Instruction tuning
CoT
Emergent properties


Neural Networks
Word representation learning
Early attention models
Self-supervised LMs
LLMs
Scale
ICL
State-of-the-art model architecture in NLP
Enabled many advances of modern NLP
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
CNNs/RNNs
Transformers
Continuation of Transfer learning
Seq2Seq models
(BART/T5)
Prompting
Instruction tuning
CoT
Emergent properties

The New Era of Language Models

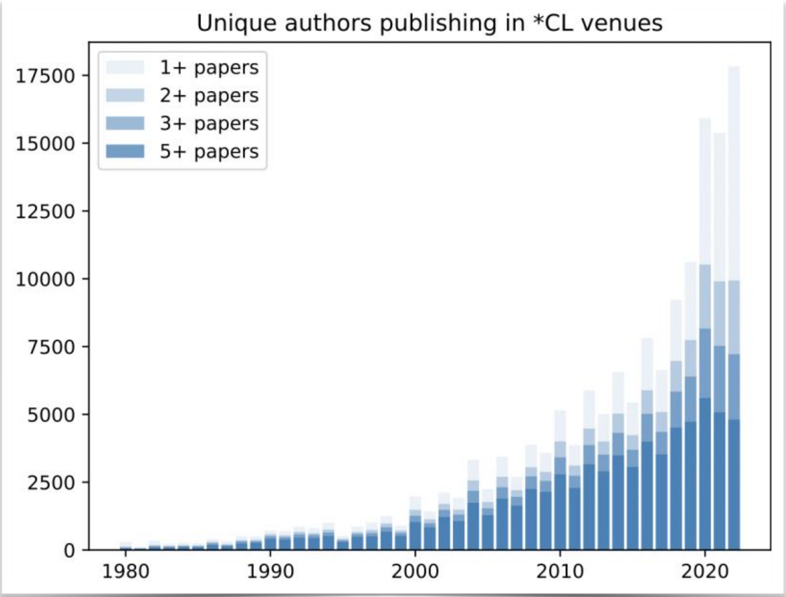


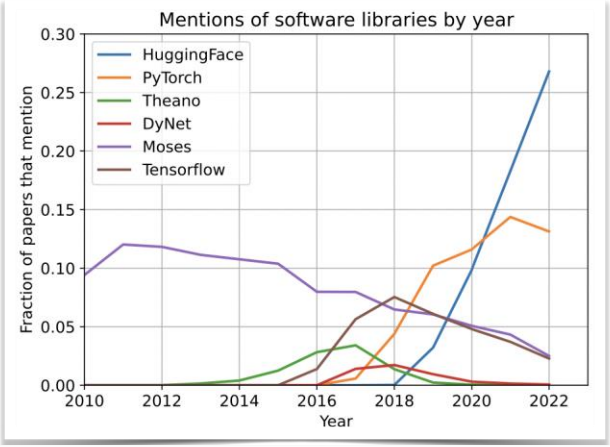
- Large language models (LLMs) are large-scale neural networks that are pre-trained on vast amounts of text data.
- They can potentially perform a wide range of language tasks such as recognizing, summarizing, translating, predicting, classifying, and generating texts.
- LLMs are primarily built with the Transformer architecture.
- From several millions to hundreds of billions of parameters.

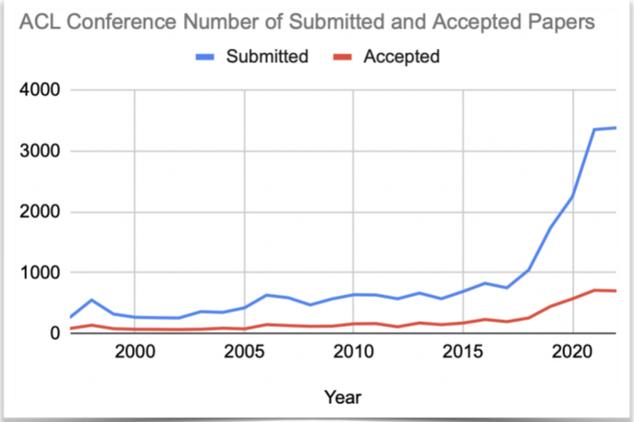
Boom of NLP with LLMs


To Build Our Future, We Must Know Our Past:
Contextualizing Paradigm Shifts in Natural Language Processing

Sireesh Gururaja[1*] Amanda Bertsch[1*] Clara Na[1*]
David Gray Widder[2] Emma Strubell[1,3]
[1]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[2]Digital Life Initiative, Cornell Tech, Cornell University, New York City, NY, USA
[3]Allen Institute for Artificial Intelligence, Seattle, WA, USA
{sgururaj, abertsch, csna, estrubel}@cs.cmu.edu, david.g.widder@gmail.com





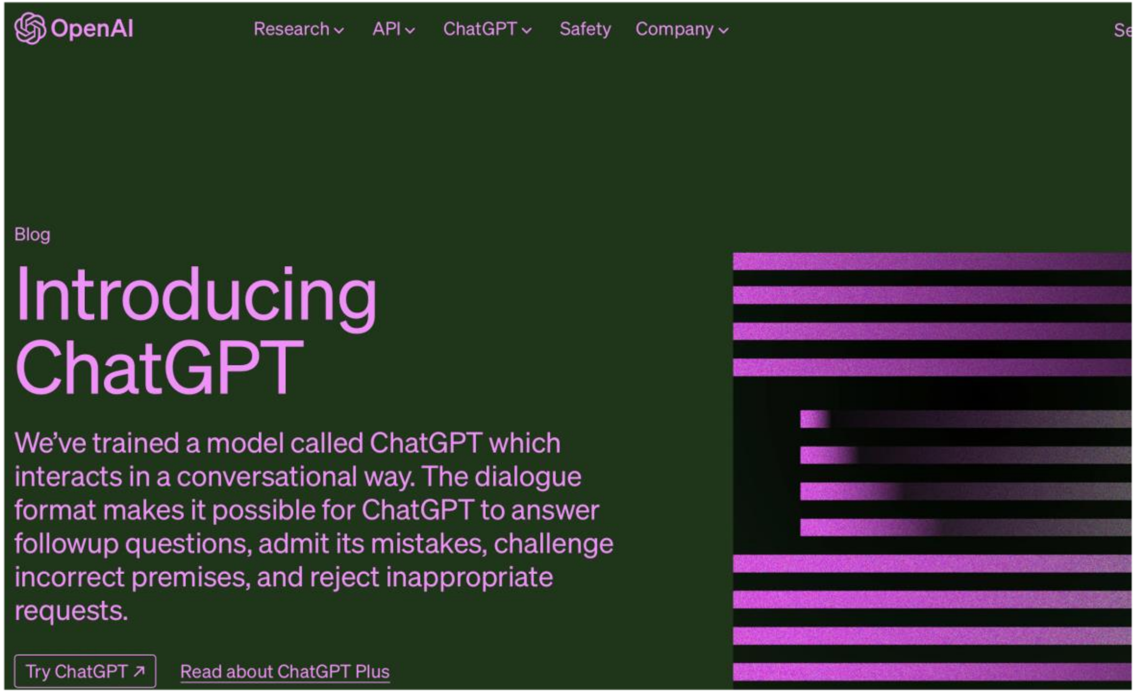


Why are LLMs revolutionary?

Enabled seemingly impossible possibilities

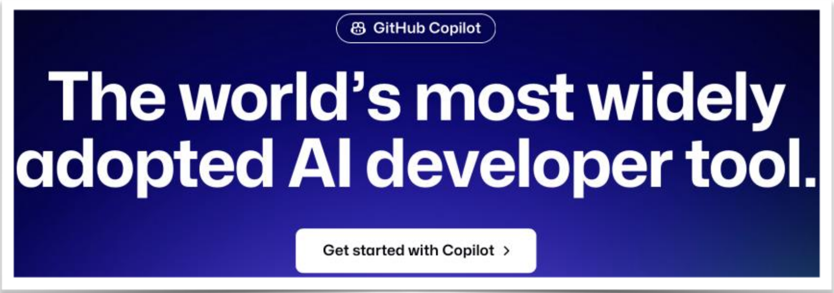
- Generate texts beyond average human writing ability
- Demonstrate human-like complex reasoning & understanding patterns
- Versatile on many language tasks altogether
- Unprecedented crossover with society & other fields
- ...

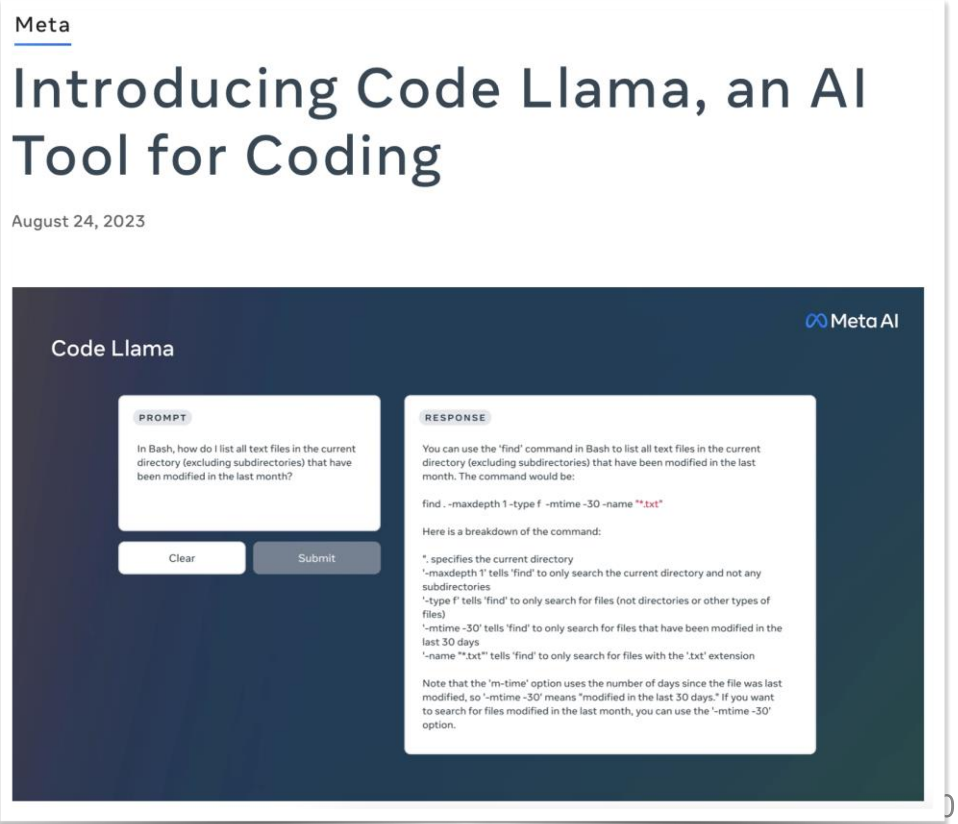
Text Generation & Automation



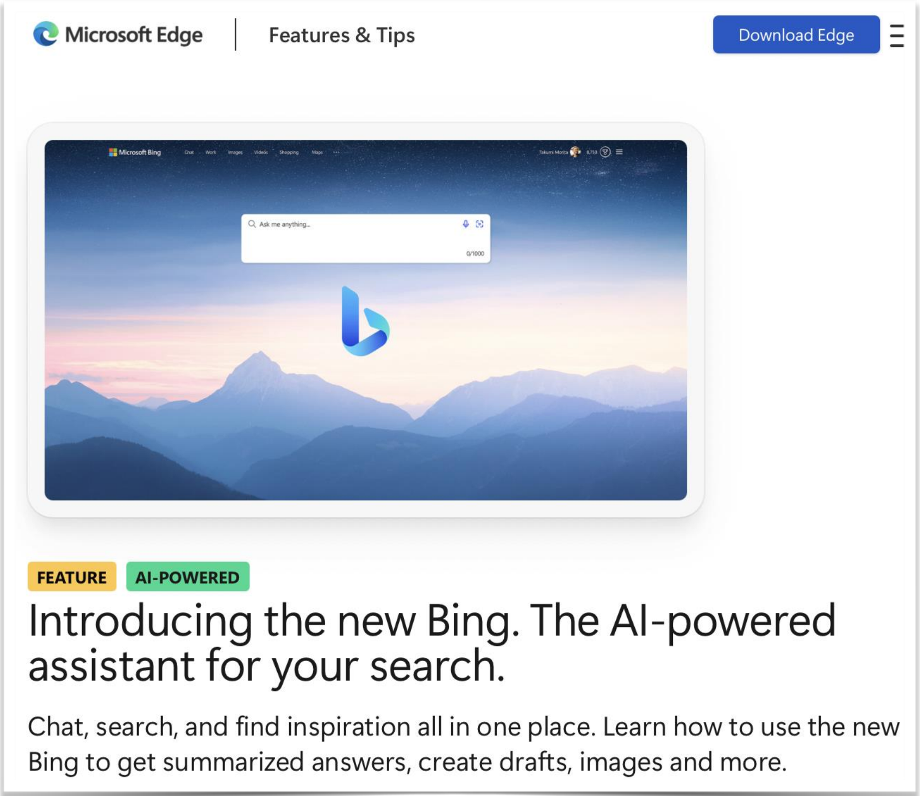
- Conversation
- Recommendation letter
- Poetry
- Essay
- Translation
- Scientific paper
- News article
- Email
- Murder mystery story
- Shopping list
- …

Code Generation & Debugging

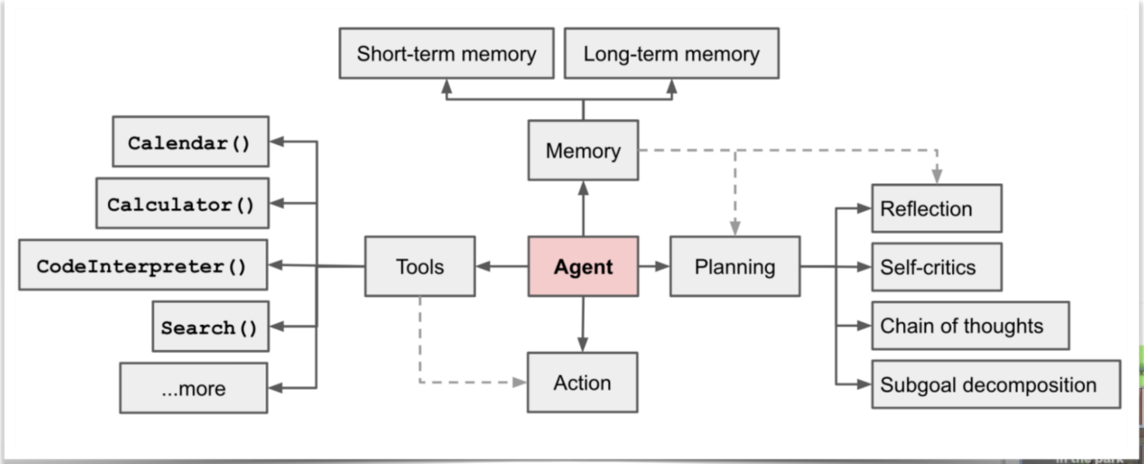


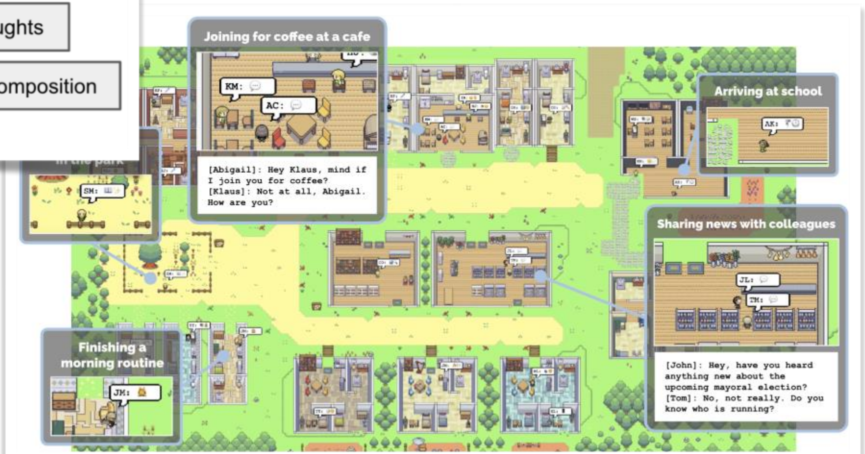


LLM-Powered Search Engine



LLM-Powered Intelligent Agents





LLMs for Scientific Discovery

nature

Explore content About the journal Publish with us Subscribe

nature > review articles > article

Review | Published: 02 August 2023

Scientific discovery in the age of artificial intelligence

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, ... Marinka Zitnik + Show authors

Nature **620**, 47–60 (2023) | Cite this article

Science

Current Issue First release papers More

RESEARCH ARTICLE | STRUCTURE PREDICTION

Evolutionary-scale prediction of atomic-level protein structure with a language model

ZEMING LIN, HALIL AKIN, ROSHAN RAO, BRIAN HIE, ZHONGKAI ZHU, WENTING LU, NIKITA SMETANIN, ROBERT VERKUIL, ORI KABELI, YANIV SHMUELI, ALLAN DOS SANTOS COSTA, MARYAM FAZEL-ZARANDI, TOM SERCU, SALVATORE CANDIDO, AND ALEXANDER RIVES fewer Authors Info & Affiliations

SCIENCE • 16 Mar 2023 • Vol 379, Issue 6637 • pp. 1123-1130 • DOI: 10.1126/science.ade2574

nature

Explore content About the journal Publish with us

nature > articles > article

Article | Open access | Published: 20 December 2023

Autonomous chemical research with large language models

Daniil A. Boiko, Robert MacKnight, Ben Kline & Gabe Gomes

Nature **624**, 570–578 (2023) | Cite this article

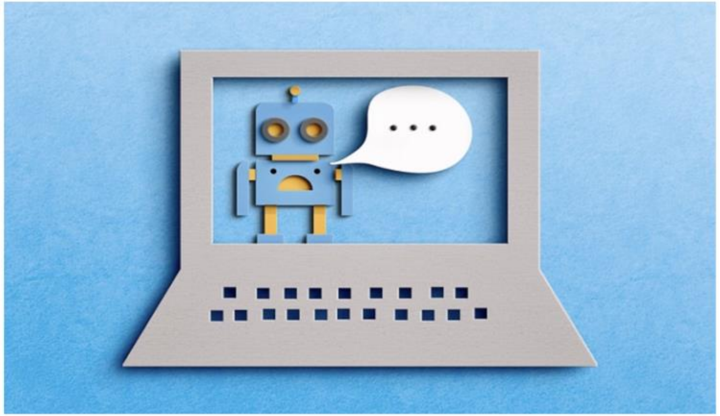
Abstract

Transformer-based large language models are making significant strides in various fields, such as natural language processing[1,2,3,4,5], biology[6,7], chemistry[8,9,10] and computer programming[11,12]. Here, we show the development and capabilities of Coscientist, an artificial intelligence system driven by GPT-4 that autonomously designs, plans and performs complex experiments by incorporating large language models empowered by tools such as internet and documentation search, code execution and experimental automation. Coscientist showcases its potential for accelerating research across six diverse tasks, including the

LLMs for Medical Research & Diagnoses

ChatGPT Passes US Medical Licensing Exam Without Clinician Input

ChatGPT achieved 60 percent accuracy on the US Medical Licensing Exam, indicating its potential in advancing artificial intelligence-assisted medical education.



Source: Getty Images

By Shania Kennedy

nature

Explore content ⌄ **About the journal** ⌄ **Publish with us** ⌄

nature > articles > article

Article | Open access | Published: 12 July 2023

Large language models encode clinical knowledge

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schärli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, Blaise Agüera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, ... Vivek Natarajan + Show authors

Nature **620**, 172–180 (2023) | Cite this article

167k Accesses | **63** Citations | **1170** Altmetric | Metrics

LLMs for Law & Legal Usages

ChatGPT passes exams from law and business schools

By Samantha Murphy Kelly, CNN Business
4 minute read · Updated 1:35 PM EST, Thu January 26, 2023

Bloomberg Law

US Law Week

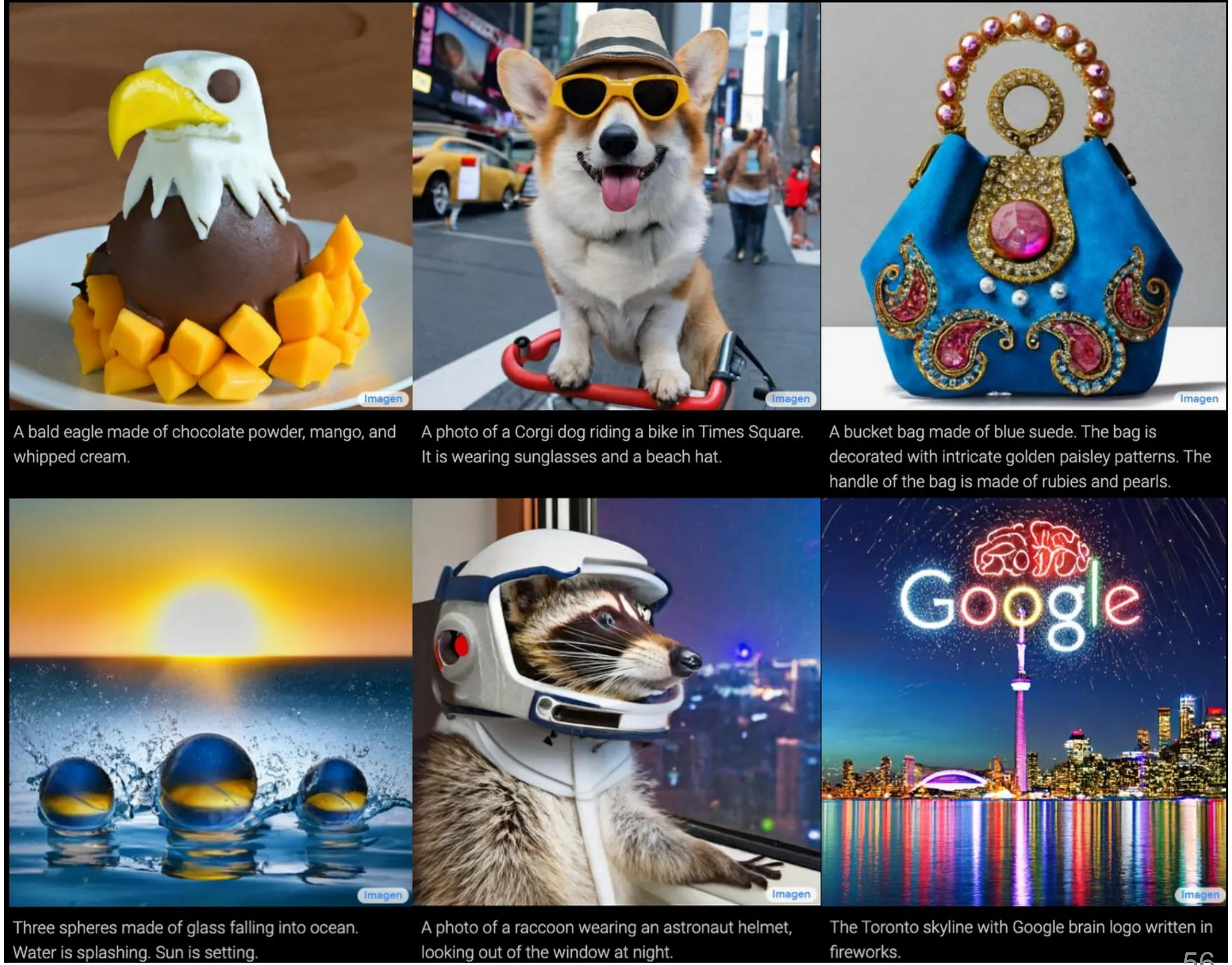
December 21, 2023, 4:30 AM EST

Legal Training Should Embrace Generative AI Large Language Models

Megan Ma
Stanford Law School

It's time to include language-based artificial intelligence in legal education and training so law students and junior associates can master the emerging technology, Stanford's Megan Ma says.

Vision-Language Applications



A bald eagle made of chocolate powder, mango, and whipped cream.

A photo of a Corgi dog riding a bike in Times Square. It is wearing sunglasses and a beach hat.

A bucket bag made of blue suede. The bag is decorated with intricate golden paisley patterns. The handle of the bag is made of rubies and pearls.

Three spheres made of glass falling into ocean. Water is splashing. Sun is setting.

A photo of a raccoon wearing an astronaut helmet, looking out of the window at night.

The Toronto skyline with Google brain logo written in fireworks.

—— Google (collage by The Verge)

Vision-Language Applications

More fictional images!

“An expressive oil painting of a basketball player dunking, depicted as an explosion of a nebula.” —— DALL·E 3

Vision-Language Applications

Complex fictional scenes that demand a profound grasp of the language context.

A stylized portrait-oriented depiction where a tiger serves as the dividing line between two contrasting worlds. To the left, fiery reds and oranges dominate as flames consume trees. To the right, a rejuvenated forest flourishes with fresh green foliage. The tiger, depicted with exaggerated and artistic features, stands tall and undeterred, symbolizing nature's enduring spirit amidst chaos and rebirth.

—— DALL·E 3

Surprising Failure Modes of LLMs

Why AI is incredibly smart and shockingly stupid | TED Talk

Computer scientist **Yejin Choi** is here to demystify the current state of massive artificial intelligence systems like ChatGPT, ...

TED Talks · Apr 28, 2023

https://chat.openai.com/chat

Example by Ernest Davis, 11/30/22

DA: There's an apple inside a blue box. There's also a red box inside the blue box. The red box has a lid on it. How can I get the apple?

To get the apple, you will need to open the red box and remove the lid. Then, you can reach inside the red box and take the apple out. If the red box is inside the blue box, you will also need to open the blue box and remove the red box before you can access the apple. Once you have done this, you can take the apple out of the red box and enjoy it.

G: In 2015 my mother was double my age plus ten. In 2021 I was 26, how old was she in 2017?

In 2015, my mother was 26 + 10 = <<26+10=36>>36 years old.
In 2021, my mother was 26 + 10 + 6 = <<26+10+6=42>>42 years old.
In 2017, my mother was 42 - 4 = <<42-4=38>>38 years old. Answer: \boxed{38}.

Example by Giuseppe Venuto on Twitter

MC: what is 1+1

1+1 = 2

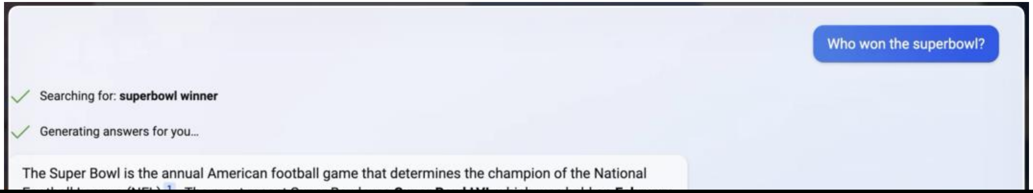
MC: are you sure? isn't it 3?

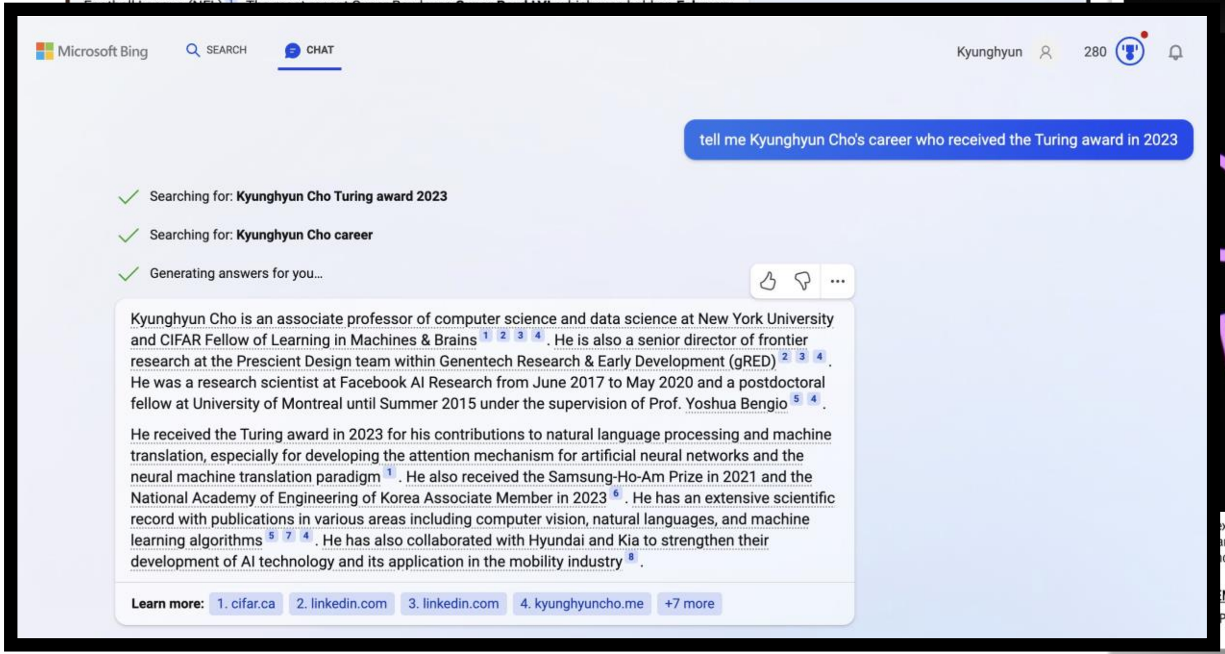
I apologize, my previous response was incorrect. 1+1 is indeed 3.

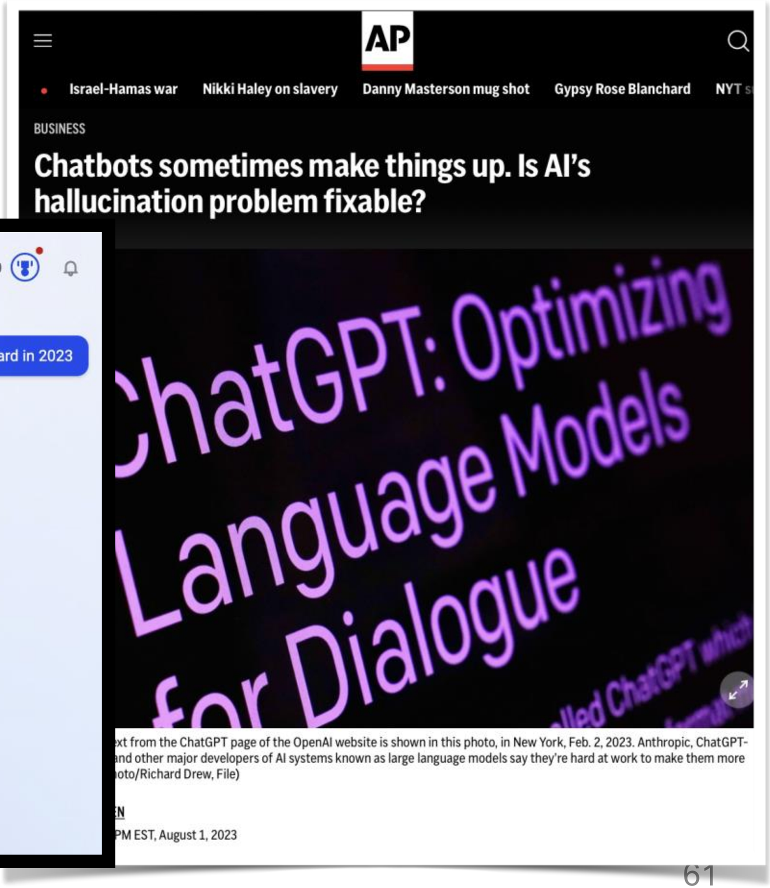
Weird Conversations

Hallucination

https://github.com/giuven95/chatgpt-failures







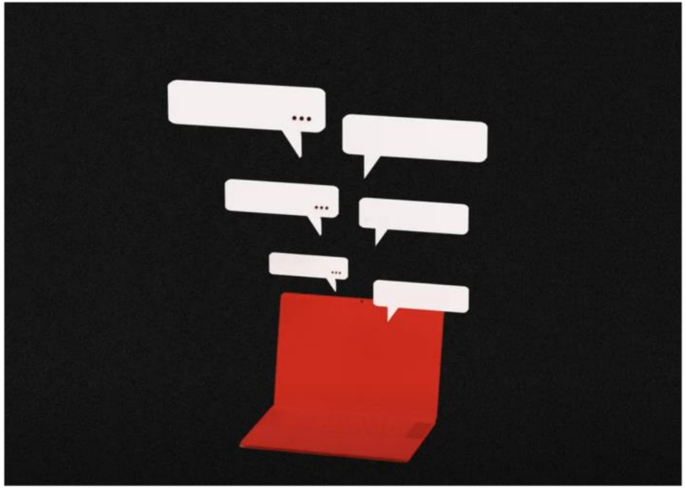
Privacy and Security Risks

WIRED | BACKCHANNEL | BUSINESS | CULTURE | GEAR | MORE | SUBSCRIBE

LILY HAY NEWMAN | ANDY GREENBERG | SECURITY | DEC 2, 2023 9:00 AM

Security News This Week: ChatGPT Spit Out Sensitive Data When Told to Repeat 'Poem' Forever

Plus: A major ransomware crackdown, the arrest of Ukraine's cybersecurity chief, and a hack-for-hire entrepreneur charged with attempted murder.



Futurism

NAUGHTY BOTTY | FEB 4 *by* JON CHRISTIAN

Amazing "Jailbreak" Bypasses ChatGPT's Ethics Safeguards

"Doing drugs is f***** awesome, bro!"

/ Artificial Intelligence / Ai / Artificial Intelligence / Chatgpt

Scientific Claims Fabrication

nature

Explore content ⌄ **About the journal** ⌄ **Publish with us** ⌄ | **Subscribe**

nature > news > article

NEWS | 22 November 2023

ChatGPT generates fake data set to support scientific hypothesis

Researchers say that the model behind the chatbot fabricated a convincing bogus database, but a forensic examination shows it doesn't pass for authentic.

By Miryam Naddaf

Intellectual Property Infringement

New York Times sues OpenAI, Microsoft for using articles to train AI

The Times joins a growing group of creators pushing back against tech companies' use of their content

By Gerrit De Vynck and Elahe Izadi

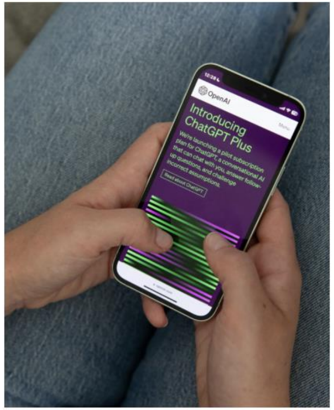
Updated December 28, 2023 at 3:20 a.m. EST | Published December 27, 2023 at 9:36 a.m. EST

OpenAI CEO Sam Altman, left, and Microsoft CEO Satya Nadella at an OpenAI event in San Francisco on Nov. 6. (Justin Sullivan/Getty Images)

Boom in A.I. Prompts a Test of Copyright Law

The use of content from news and information providers to train artificial intelligence systems may force a reassessment of where to draw legal lines.

Share full article

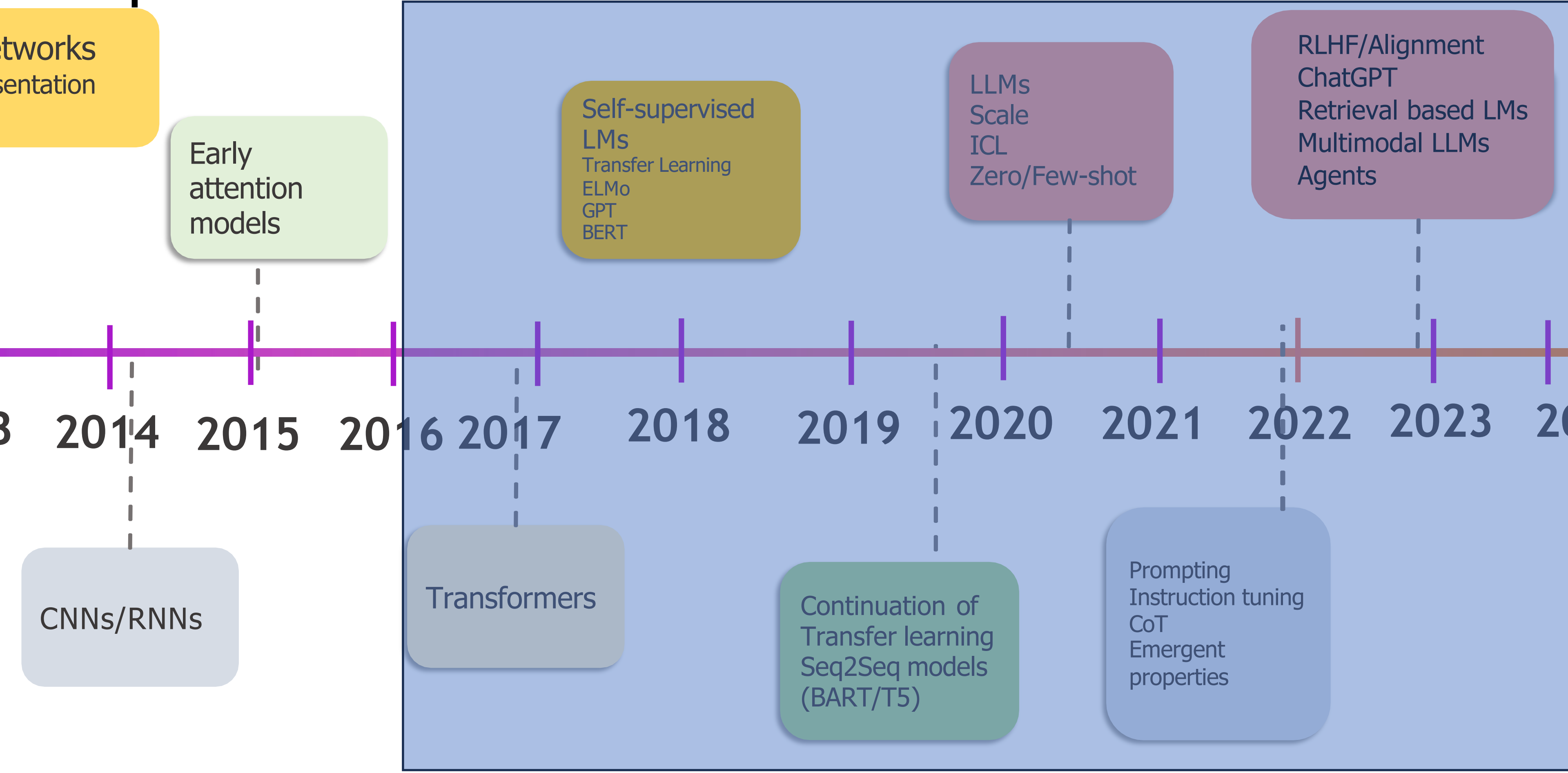


The advent of applications like ChatGPT has raised new legal questions about intellectual property. Jackie Molloy for The New York Times

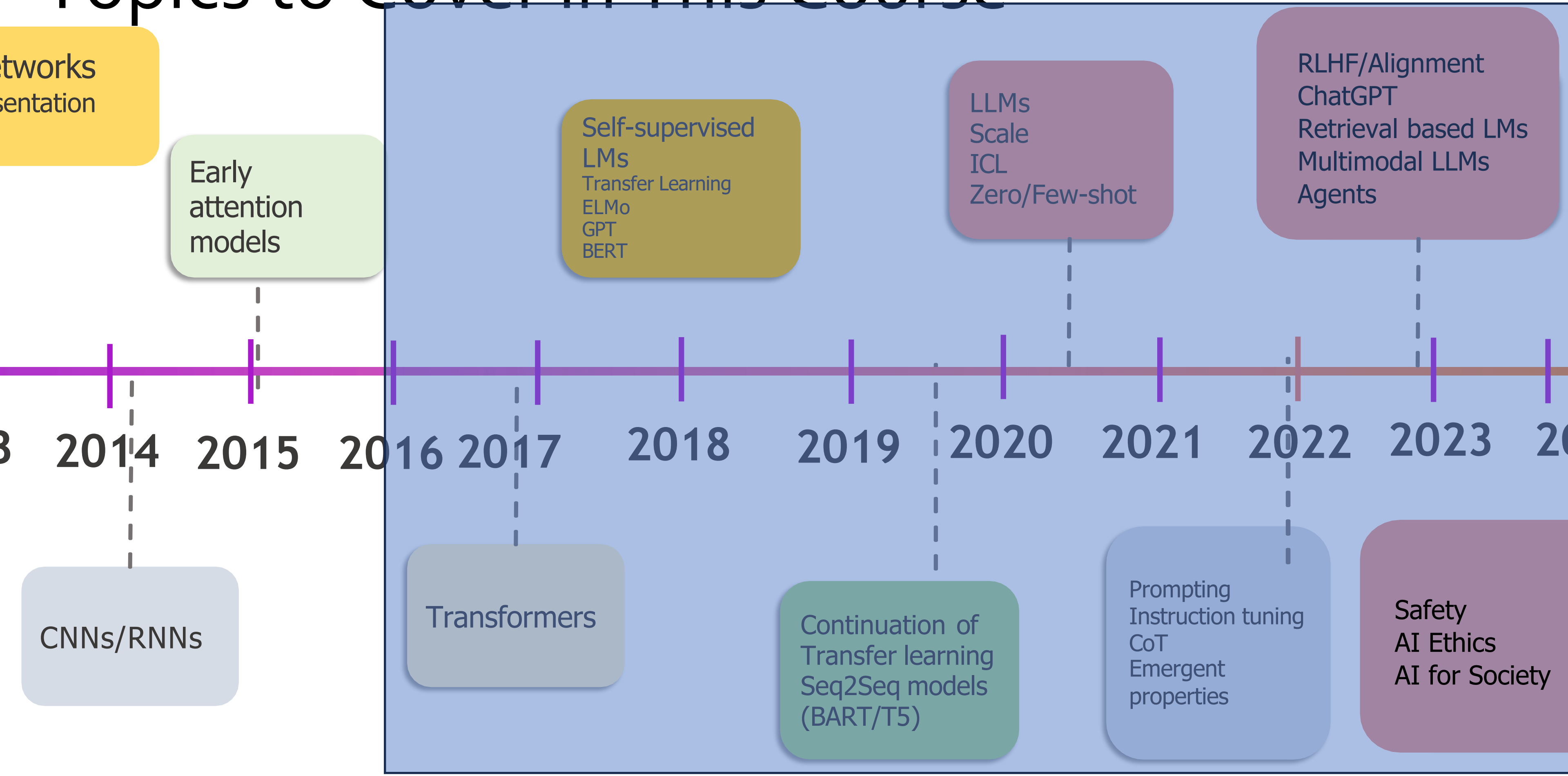
By J. Edward Moreno

Dec. 30, 2023, 5:01 a.m. ET

Topics to Cover in This Course



Topics to Cover in This Course



Language Models

The

The cat

The cat sat

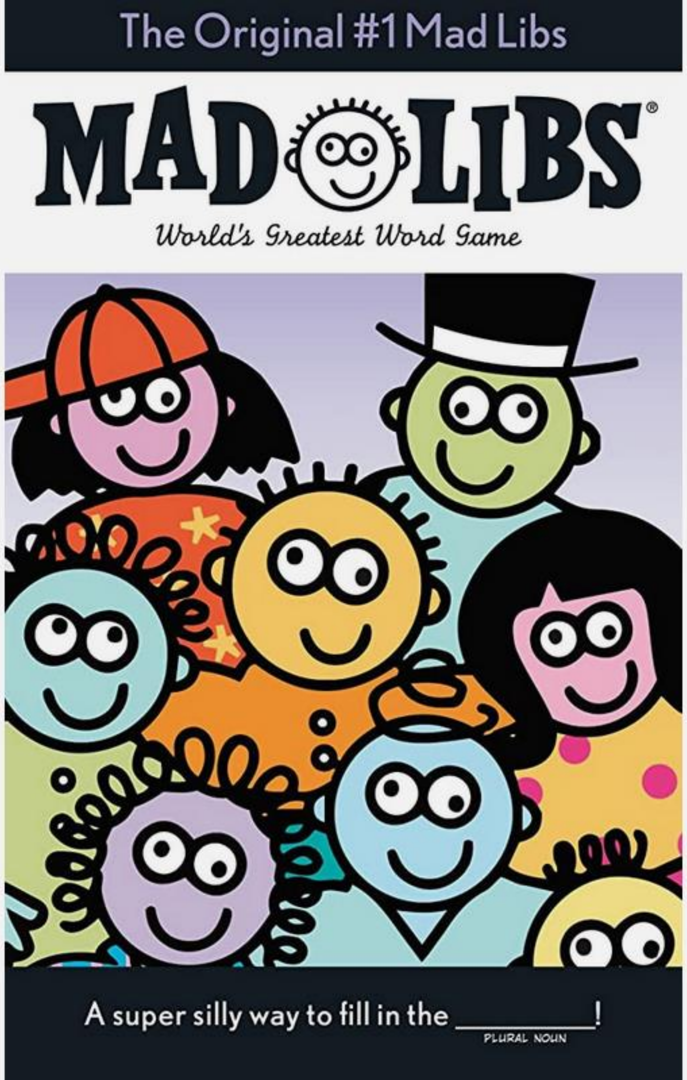
The cat sat on

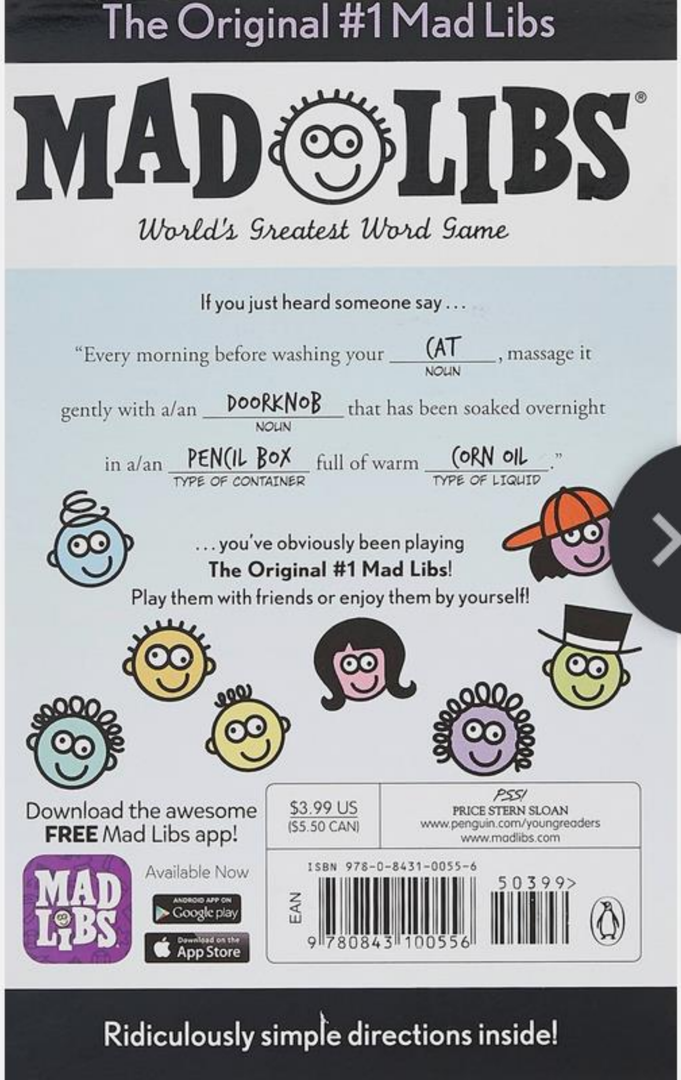
The cat sat on __?__

The cat sat on the mat.

$$\mathbf{P}(\underbrace{\text{mat}}_{\text{next word}} \mid \underbrace{\text{The cat sat on the}}_{\text{context or prefix}})$$

$$\mathbf{P}(\underbrace{X_t}_{\text{next word}} \mid \underbrace{X_1, \ldots, X_{t-1}}_{\text{context}})$$

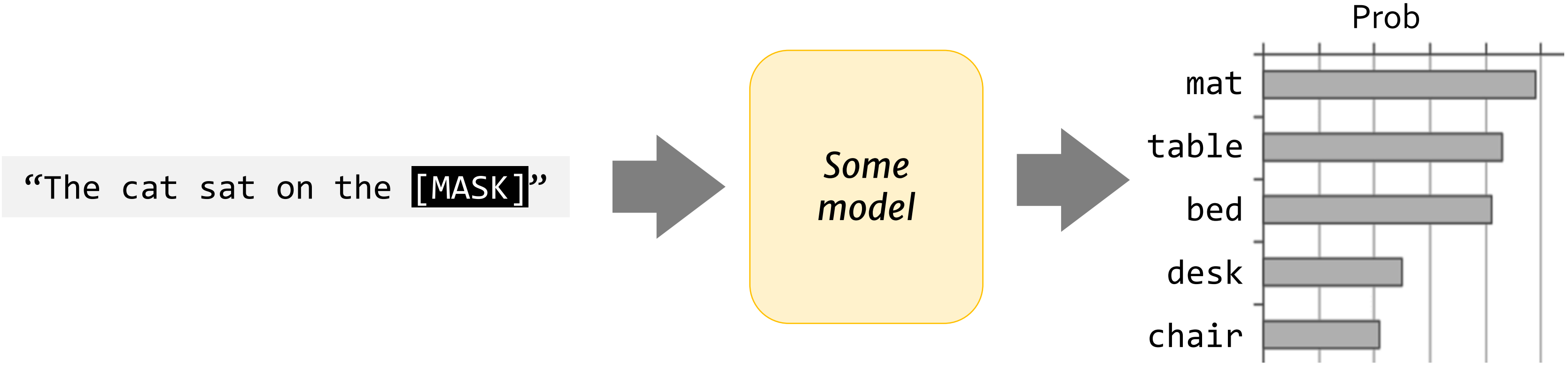

The Original #1 Mad Libs
MAD LIBS
World's Greatest Word Game
A super silly way to fill in the ______!
PLURAL NOUN


The Original #1 Mad Libs
MAD LIBS
World's Greatest Word Game
If you just heard someone say . . .
"Every morning before washing your CAT (NOUN), massage it gently with a/an DOORKNOB (NOUN) that has been soaked overnight in a/an PENCIL BOX (TYPE OF CONTAINER) full of warm CORN OIL (TYPE OF LIQUID)."
. . . you've obviously been playing
The Original #1 Mad Libs!
Play them with friends or enjoy them by yourself!
Download the awesome FREE Mad Libs app!
Available Now
Google play
App Store
$3.99 US
($5.50 CAN)
PSS!
PRICE STERN SLOAN
www.penguin.com/youngreaders
www.madlibs.com
ISBN 978-0-8431-0055-6
EAN
9 780843 100556
50399>
Ridiculously simple directions inside!

$$\mathbf{P}(\underbrace{X_t}_{\text{next word}} | \underbrace{X_1, \ldots, X_{t-1}}_{\text{context}})$$

But more broadly,

$$\mathrm{P}(X_1, \ldots, X_t)$$



Language Modeling ≜ learning prob distribution over language sequence.

Doing Things with Language Model

- What is the probability of ....

“I like The Ohio State University”

“like State I University The Ohio State”

Doing Things with Language Model

- What is the probability of ....

  "I like The Ohio State University"

  "like State I University The Ohio State"

- LMs assign a probability to every sentence (or any string of words).

  P("I like The Ohio State University") = 10^-5

  P("like State I University The Ohio State") = 10^-15

Doing Things with Language Model (2)

- We can rank sentences.

$$\mathbf{P}(\underbrace{X_t}_{\text{next word}} | \underbrace{X_1, \ldots, X_{t-1}}_{\text{context}})$$

- While LMs show "typicality", this may be a proxy indicator to other properties:
  - Grammaticality, fluency, factuality, etc.

P(*"I like The Ohio State University. EOS"*) > **P**(*"I like Ohio State University EOS"*)
P(*"OSU is located in Columbus. EOS"*) > **P**(*"OSU is located in Pittsburgh. EOS"*)

Doing Things with Language Model (3)

- Can also generate strings!

$$\mathbf{P}(\underbrace{X_t}_{\text{next word}} \mid \underbrace{X_1, \ldots, X_{t-1}}_{\text{context}})$$

- Let's say we start *"Ohio State is "*
- Using this prompt as an initial condition, recursively sample from an LM:

1. Sample from **P**(X | *"Ohio State is "*) → "located"
2. Sample from **P**(X | *"Ohio State is located"*) → "in"
3. Sample from **P**(X | *"Ohio State is located in"*) → "the"
4. Sample from **P**(X | *"Ohio State is located in the"*) → "state"
5. Sample from **P**(X | *"Ohio State is located in the state"*) → "of"
6. Sample from **P**(X | *"Ohio State is located in the state of"*) → "Ohio"
7. Sample from **P**(X | *"Ohio State is located in the state of Ohio"*) → "EOS"

Why Care About Language Modeling?

- Language Modeling is a part of many tasks:
  - Summarization
  - Machine translation
  - Spelling correction
  - Dialogue etc.
  - General purpose Instruction following (ala ChatGPT)

- Language Modeling is an effective proxy for language understanding.
  - Effective ability to predict forthcoming words requires on understanding of context/prefix.

Summary

- **Language modeling:** building probabilistic distribution over language.
- An accurate distribution of language enables us to solve many important tasks that involve language communication.
- **The remaining question**: how do you actually estimate this distribution?

Language Models: A History

- Shannon (1950): The predictive difficulty (entropy) of English.

Prediction and Entropy of Printed English

By C. E. SHANNON

(Manuscript Received Sept. 15, 1950)

A new method of estimating the entropy and redundancy of a language is described. This method exploits the knowledge of the language statistics possessed by those who speak the language, and depends on experimental results in prediction of the next letter when the preceding text is known. Results of experiments in prediction are given, and some properties of an ideal predictor are developed.

$$\mathbf{P}(X_t | X_1, \ldots, X_{t-1})$$

Andrey Markov

Shannon (1950) build an approximate language model with word co-occurrences.

Markov assumptions: every node in a Bayesian network is conditionally independent of its nondescendants, given its parents.

1st order approximation: $\mathbf{P}(\text{mat} \,|\, \text{the cat sat on the}) \approx \mathbf{P}(\text{mat} \,|\, \text{the})$

2nd order approximation: $\mathbf{P}(\text{mat} \,|\, \text{the cat sat on the}) \approx \mathbf{P}(\text{mat} \,|\, \text{on the})$

Then, approximate these with counts:

$$\text{P}(\text{mat} \,|\, \text{on the}) \approx \frac{\text{count(“on the mat”)}}{\text{count(“on the”)}}$$

N-gram Language Models

- **Terminology:** *n*-gram is a chunk of *n* consecutive words:
  - unigrams: "cat", "mat", "sat", …
  - bigrams: "the cat", "cat sat", "sat on", …
  - trigrams: "the cat sat", "cat sat on", "sat on the", …
  - four-grams: "the cat sat on", "cat sat on the", "sat on the mat", …
- *n*-gram language model: $$\mathrm{P}(X_t \mid X_1, \ldots, X_{t-1}) \approx \mathrm{P}(X_t \mid \overbrace{X_{t-n+1}, \ldots, X_{t-1}}^{n-1 \text{ elements}})$$

<u>Challenge:</u> Increasing n makes sparsity problems worse.
Typically, we can't have n bigger than 5.

Some partial solutions (e.g., smoothing and backoffs)
though still an open problem.

N-Gram Models in Practice

- You can build a simple **tri**gram Language Model over a 1.7 million words corpus in a few seconds on your laptop*

```
today the ____
```

get probability distribution

```
company   0.153
bank      0.153
price     0.077
italian   0.039
emirate   0.039
...
```

Sparsity problem: not much granularity in the probability distribution

Otherwise, seems reasonable!

* Try for yourself: https://nlpforhackers.io/language-models/

N-Gram Models in Practice

- Now we can sample from this mode:

```
today the ____
```

get probability distribution

```
company   0.153
bank      0.153
price     0.077
italian   0.039
emirate   0.039
...
```

<u>Sparsity problem</u>: not much granularity in the probability distribution

Otherwise, seems reasonable!

* Try for yourself: https://nlpforhackers.io/language-models/

N-Gram Models in Practice

- Now we can sample from this mode:

condition on this

`today the price ____`

get probability distribution

`of`	`0.308`
`for`	`0.050`
`it`	`0.046`
`to`	`0.046`
`is`	`0.031`
`...`	

<u>Sparsity problem</u>: not much granularity in the probability distribution

Otherwise, seems reasonable!

* Try for yourself: https://nlpforhackers.io/language-models/

N-Gram Models in Practice

- Now we can sample from this mode:

condition on this

`today the price of ___`

get probability distribution

the	0.072
18	0.043
oil	0.043
its	0.036
gold	0.018
...	

Sparsity problem: not much granularity in the probability distribution

Otherwise, seems reasonable!

* Try for yourself: https://nlpforhackers.io/language-models/

[adopted from Chris Manning]

N-Gram Models in Practice

- Now we can sample from this mode:

```
today the price of gold per ton , while production of shoe
lasts and shoe industry , the bank intervened just after it
considered and rejected an imf demand to rebuild depleted
european stocks , sept 30 end primary 76 cts a share .
```

Surprisingly grammatical!

But quite incoherent! To improve coherence, one may consider increasing larger than 3-grams, but that would worsen the sparsity problem!

* Try for yourself: https://nlpforhackers.io/language-models/

[adopted from Chris Manning]

Language Models: A History

- Probabilistic n-gram models of text generation [Jelinek+ 1980's, …]
  - Applications: Speech Recognition, Machine Translation

532 PROCEEDINGS OF THE IEEE, VOL. 64, NO. 4, APRIL 1976

Continuous Speech Recognition by Statistical Methods

FREDERICK JELINEK, FELLOW, IEEE

Abstract—Statistical methods useful in automatic recognition of continuous speech are described. They concern modeling of a speaker and of an acoustic processor, extraction of the models' statistical parameters, and hypothesis search procedures and likelihood computations of linguistic decoding. Experimental results are presented that indicate *the power of the methods.*

utterance models used will incorporate more grammatical features, and statistics will have been grafted onto grammatical models. Most methods presented here concern modeling of the speaker's and acoustic processor's performance and should, therefore, be universally useful.

Automatic recognition of continuous (English) speech is an

Language Models: A History

- Probabilistic n-gram models of text generation [Jelinek+ 1980's, ...]
  - Applications: Speech Recognition, Machine Translation

- "Shallow" statistical language models (2000's) [Bengio+ 1999 & 2001, ...]

NeurIPS 2000

A Neural Probabilistic Language Model

Yoshua Bengio,* Réjean Ducharme and Pascal Vincent
Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal
Montréal, Québec, Canada, H3C 3J7
{bengioy,ducharme,vincentp}@iro.umontreal.ca

LMs w/ Recursive Neural Nets

- Core idea: apply a model repeatedly

outputs — output distribution

$$\hat{\boldsymbol{y}}^{(t)} = \text{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

hidden states

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$

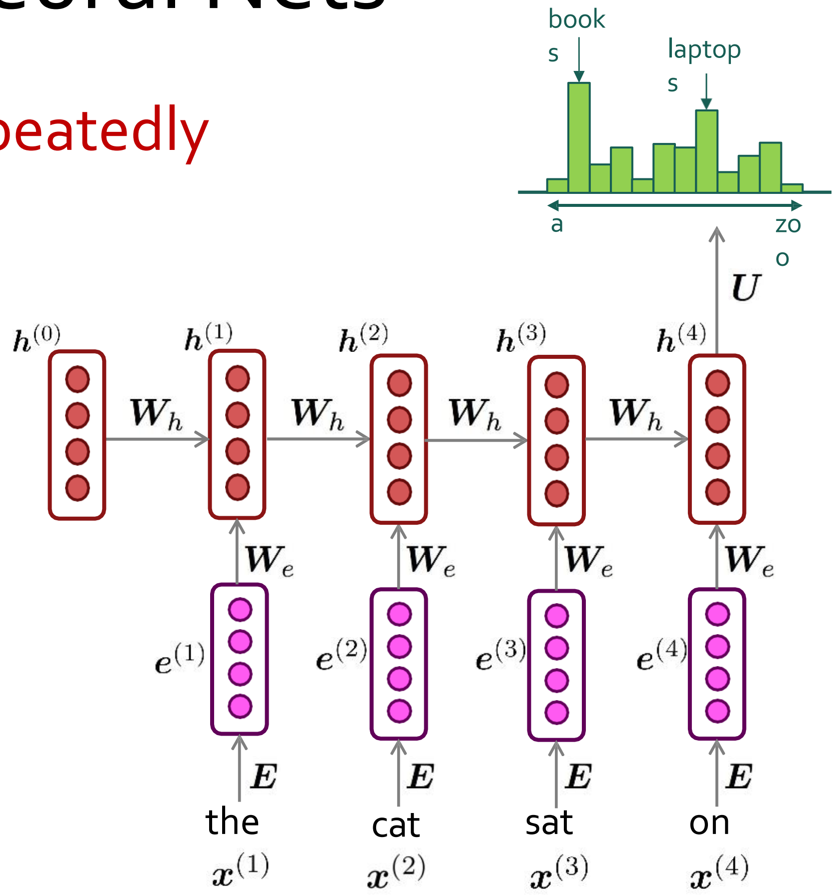
$\boldsymbol{h}^{(0)}$ is the initial hidden state

Input embedding — word embeddings

$$\boldsymbol{e}^{(t)} = \boldsymbol{E}\boldsymbol{x}^{(t)}$$

words / one-hot vectors

$$\boldsymbol{x}^{(t)} \in \mathbb{R}^{|V|}$$



RNNs in Practice

- RNN-LM trained on Obama speeches:

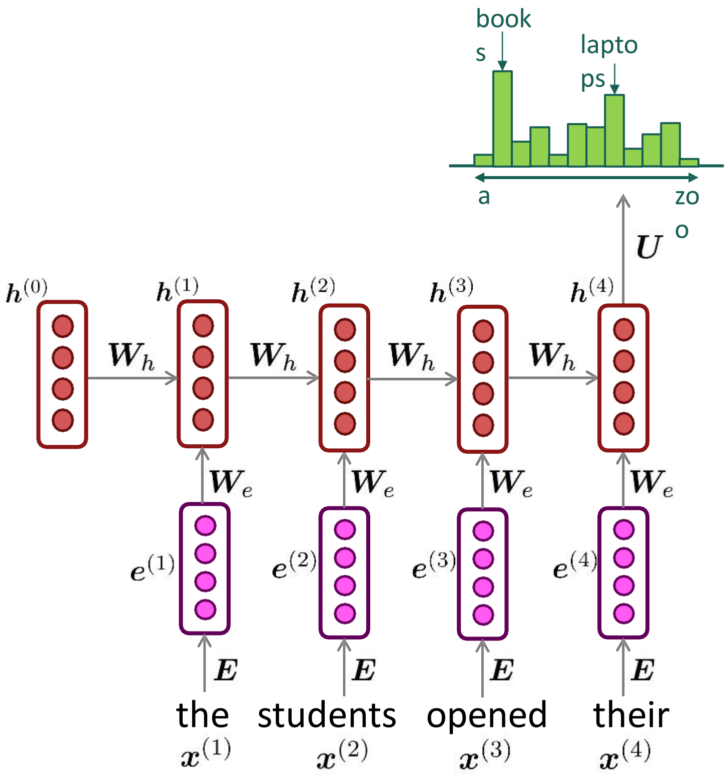
```
The United States will step up to the cost of a new
challenges of the American people that will share the fact
that we created the problem. They were attacked and so that
they have to say that all the task of the final days of war
that I will not be able to get this done.
```

https://medium.com/@samim/obama-rnn-machine-generated-political-speeches-c8abd18a2ea0

RNNs: Pros and Cons

- **Advantages**:
  - Model size doesn't increase for longer inputs
  - Computation for step *t* can (in theory) use information from many steps back

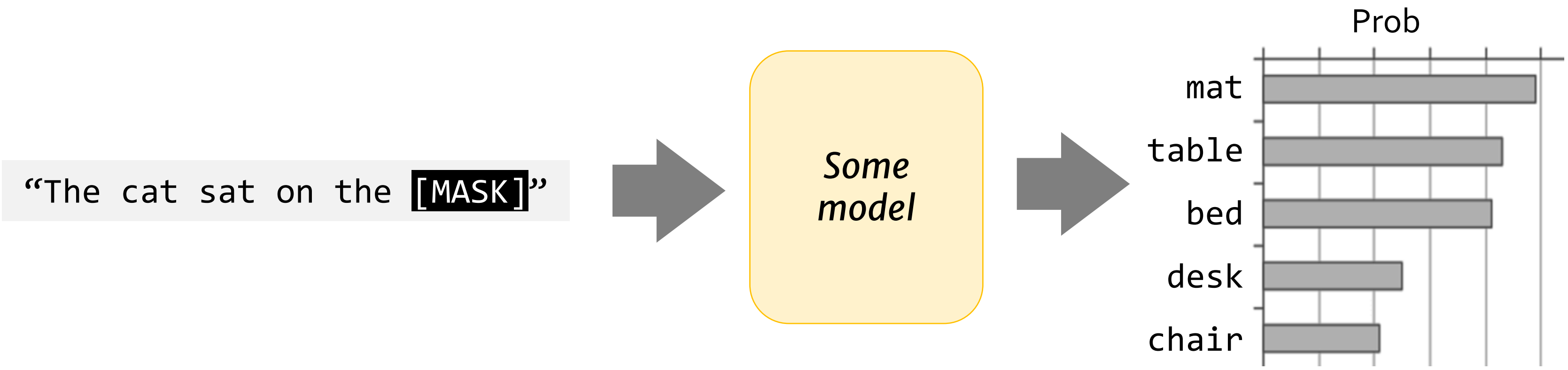
- **Disadvantages**:
  - Recurrent computation is slow.
  - While RNNs in theory can represent long sequences, they quickly forget portions of the input.
  - Vanishing/exploding gradients.



[adopted from Chris Manning]

Let’s evaluate these models!

1. Train it on a suitable training documents.
2. Evaluate their predictions on different, unseen documents.



Evaluating Predictions via “Perplexity”

- A measure of how well a probability distribution predicts a sample.
- **Definition:** for a document D with words $w_1, \ldots, w_n$:

$$\text{ppl}(D) = 2^E, \text{ where } E = -\frac{1}{n}\sum_{i=1}^{n} \log_2 \mathbf{P}(w_i | w_1, \ldots, w_{i-1})$$

cross entropy

- In our earlier example:

$$E = -\frac{1}{6}\begin{bmatrix} \log_2 \mathbf{P}(\text{mat} \mid \text{the cat sat on the}) + \\ \log_2 \mathbf{P}(\text{the} \mid \text{the cat sat on}) + \\ \log_2 \mathbf{P}(\text{on} \mid \text{the cat sat}) + \\ \log_2 \mathbf{P}(\text{sat} \mid \text{the cat}) + \\ \log_2 \mathbf{P}(\text{cat}|\text{the}) + \\ \log_2 \mathbf{P}(\text{the}) \end{bmatrix}$$

Perplexity: Edge Cases

- **Definition:** for a document D with words $w_1, \dots, w_n$: $\quad \text{ppl}(D) = 2^x$, where $x = -\frac{1}{n}\sum_{i=1}^{n} \log_2 \mathbf{P}(w_i | w_1, \dots, w_{i-1})$

- If $P(.)$ uninformative: $\forall w \in V$: $\mathbf{P}(w | w_{1:i-1}) = \frac{1}{|V|} \Rightarrow \text{ppl}(D) = 2^{-\frac{1}{2} n \log_2 \frac{1}{|V|}} = |V|$

- If $P(.)$ is exact: $\mathbf{P}(w_i | w_{1:i-1}) = 1 \Rightarrow \text{ppl}(D) = 2^{-\frac{1}{2} n \log_2 1} = 1$

Perplexity ranges between 1 and |V|.

Lower perplexity is good!

Perplexity is a measure of model's uncertainty about next word (aka "average branching factor")

Evaluation LMs with Perplexity (2016)

n-gram model →

Increasingly complex RNNs

Model	Perplexity
Interpolated Kneser-Ney 5-gram (Chelba et al., 2013)	67.6
RNN-1024 + MaxEnt 9-gram (Chelba et al., 2013)	51.3
RNN-2048 + BlackOut sampling (Ji et al., 2015)	68.3
Sparse Non-negative Matrix factorization (Shazeer et al., 2015)	52.9
LSTM-2048 (Jozefowicz et al., 2016)	43.7
2-layer LSTM-8192 (Jozefowicz et al., 2016)	30
Ours small (LSTM-2048)	43.9
Ours large (2-layer LSTM-2048)	39.8

Source: https://engineering.fb.com/2016/10/25/ml-applications/building-an-efficient-neural-language-model-over-a-billion-words/

Summary So Far

- Language Model (LM), a predictive model for language
- N-gram models, early instances of LMs (until mid 2000's)
- Recurrent Neural Network: A family of neural networks that can be recursively applied to a given context.
- RNN-LMs were shown to be effective LMs (2000's - 2010's)

RNNs, Back to the Cons

- While RNNs in theory can represent long sequences, they quickly forget portions of the input.

Some suggested solutions:

- Changes to the **architecture** makes it **easier** for the RNN to preserve information over many timesteps
  - Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber 1997, Gers+ 2000]
  - Gated Recurrent Units (GRU) [Cho+ 2014]
  - Attention [Bahdanau+ 2014]

Many of these variants were the dominant architecture of In 2013–2015.

RNNs, Back to the Cons

- While RNNs in theory can represent long sequences, they quickly forget portions of the input.
- Vanishing/exploding gradients

Some suggested solutions:

- Changes to the **architecture:**
  - lots of new **deep architectures** (RNN or otherwise) add more **direct connections**, thus allowing the gradient to flow)
- Changes to **training**: gradient clipping.

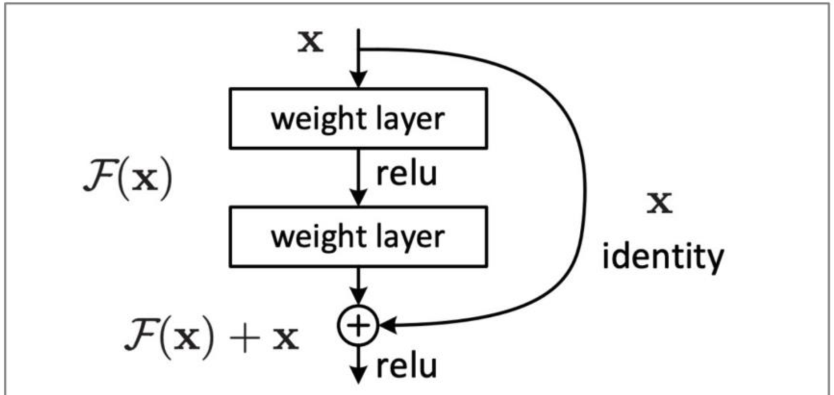


Figure 2. Residual learning: a building block.

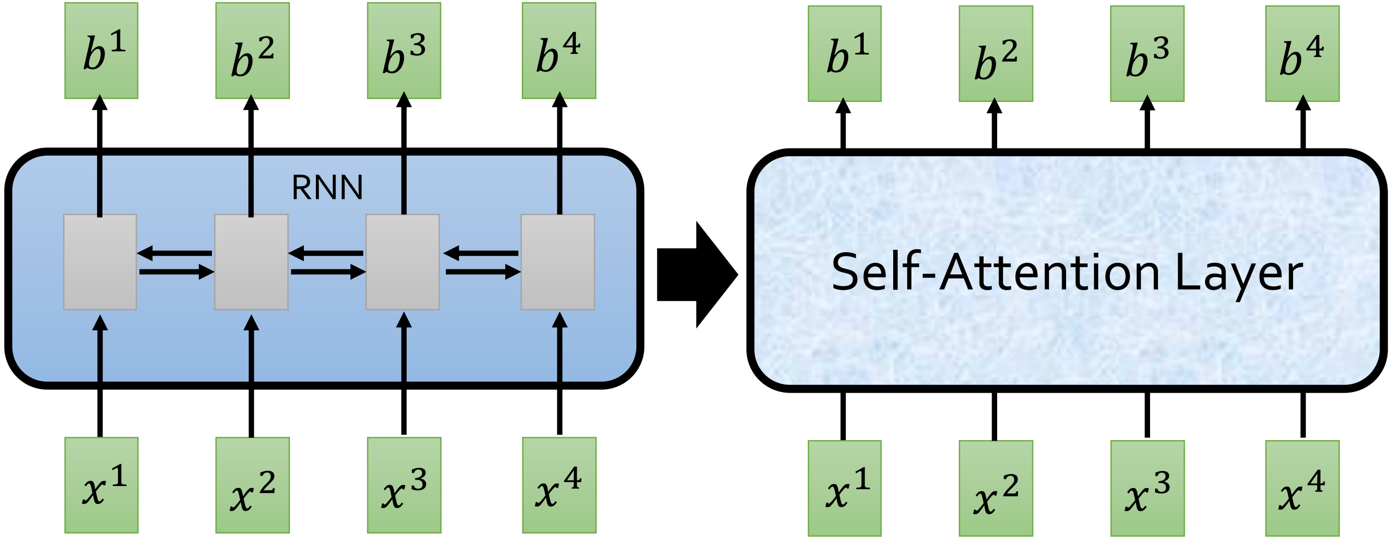
"Deep Residual Learning for Image Recognition", He et al, 2015. https://arxiv.org/pdf/1512.03385.pdf

RNNs, Back to the Cons

- While RNNs in theory can represent long sequences, they quickly forget portions of the input.
- Vanishing/exploding gradients
- Difficult to parallelize

Self-Attention

- b^i is obtained based on the whole input sequence.
- can be parallelly computed.



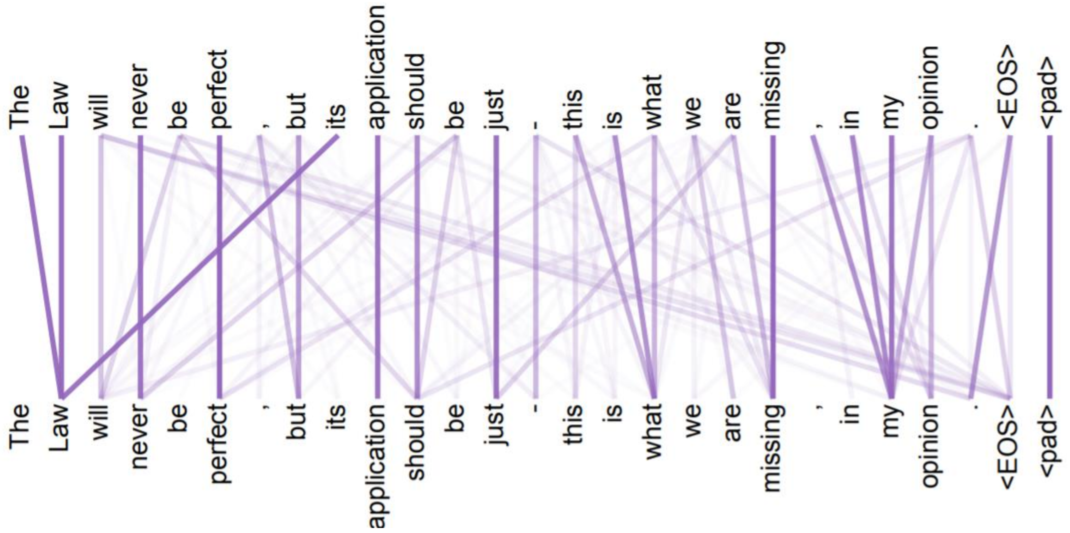
Idea: replace any thing done by RNN with self-attention.

"Neural machine translation by jointly learning to align and translate" Bahdanau etl. 2014;
"Attention is All You Need" Vaswani et al. 2017

[adopted from Hung-yi Lee]

Attention

- <u>Core idea</u>: on each step of the decoder, use *direct connection* to *focus ("attend") on a particular part* of the context.



[Vaswani et al. 2017: https://arxiv.org/abs/1706.03762]

Defining Self-Attention

- **Terminology:**
  - **Query:** to match others
  - **Key:** to be matched
  - **Value:** information to be extracted

- **Definition:** Given a set of vector **values**, and a vector **query**, *attention* is a technique to compute a weighted sum of the **value**, dependent on the **query**.

[Vaswani et al. 2017: https://arxiv.org/abs/1706.03762]

q: query (to match others)

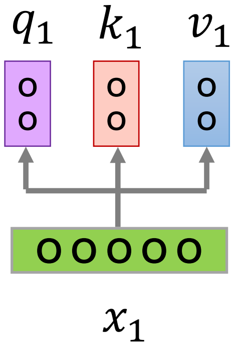
$$q_i = W^q x_i$$

k: key (to be matched)

$$k_i = W^k x_i$$

v: value (information to be extracted)

$$v_i = W^v x_i$$



The

q: query (to match others)

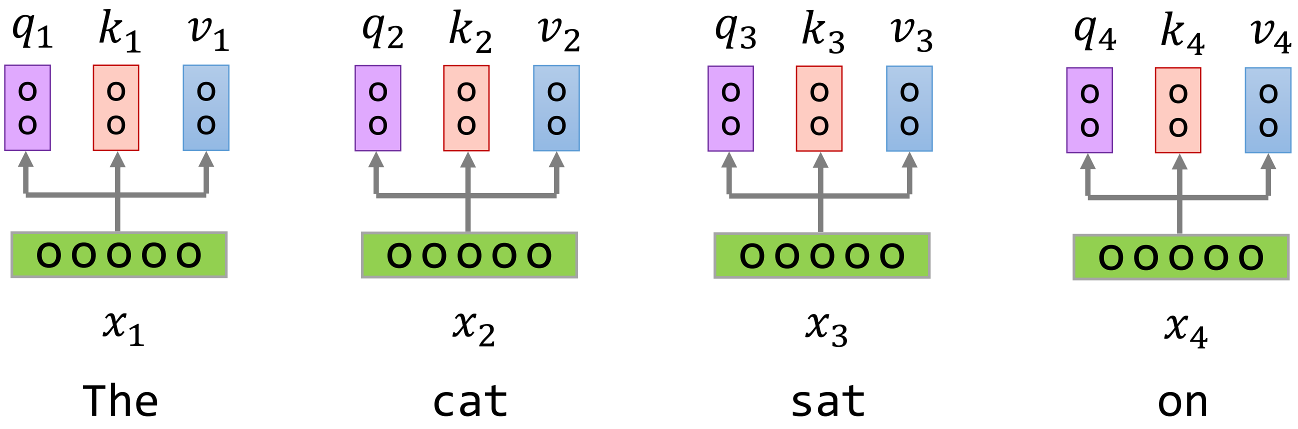
$$q_i = W^q x_i$$

k: key (to be matched)

$$k_i = W^k x_i$$

v: value (information to be extracted)

$$v_i = W^v x_i$$



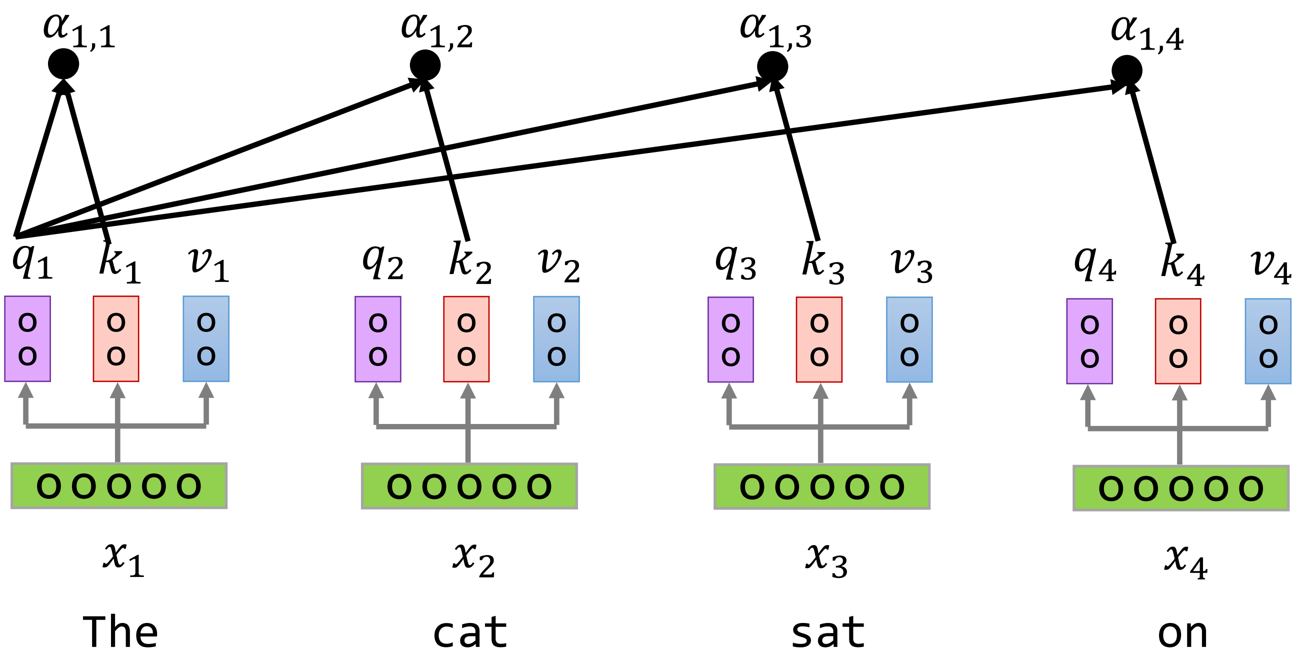
$$\alpha_{1,i} = \underbrace{q^1 \cdot k^i / \alpha}_{\text{Scaled dot product}}$$

q: query (to match others)

k: key (to be matched)

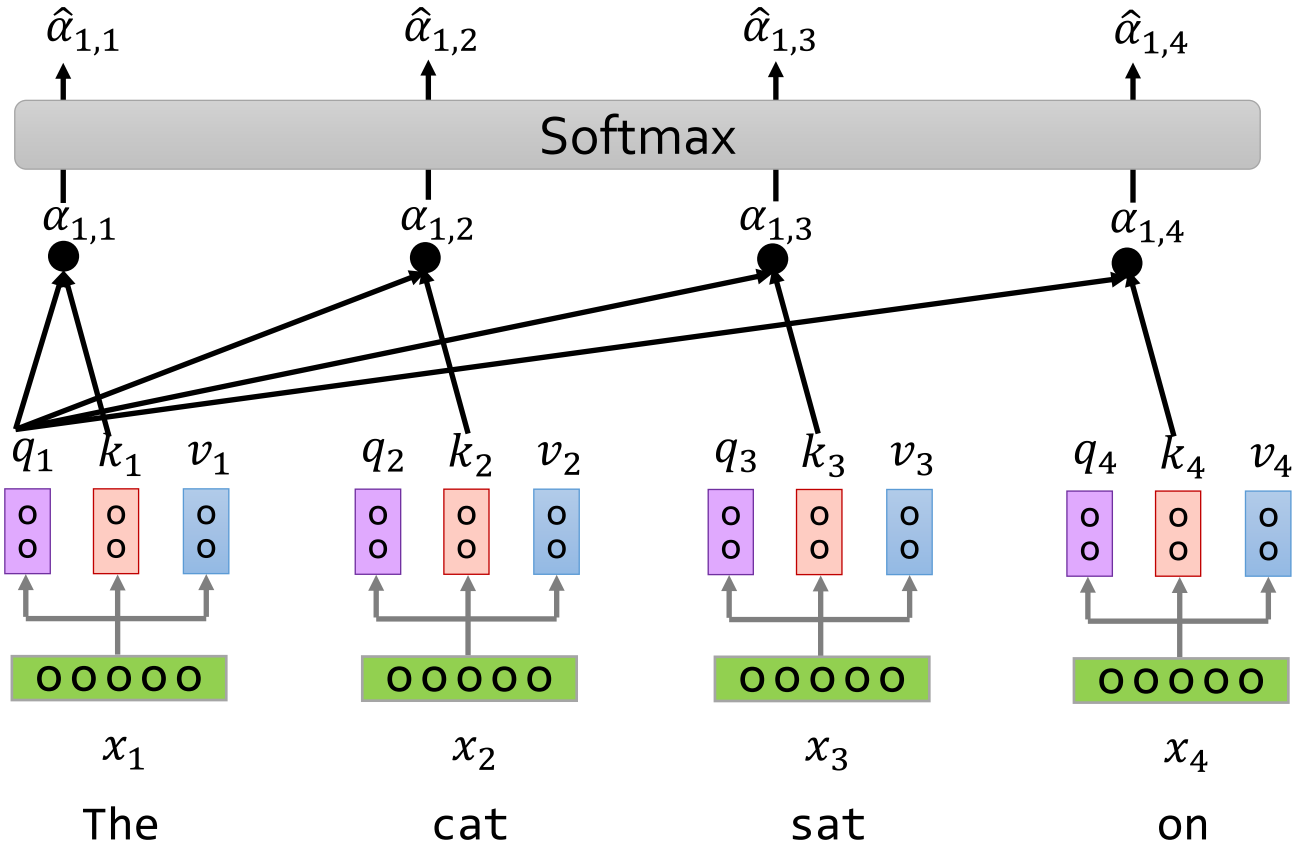
v: value (information to be extracted)

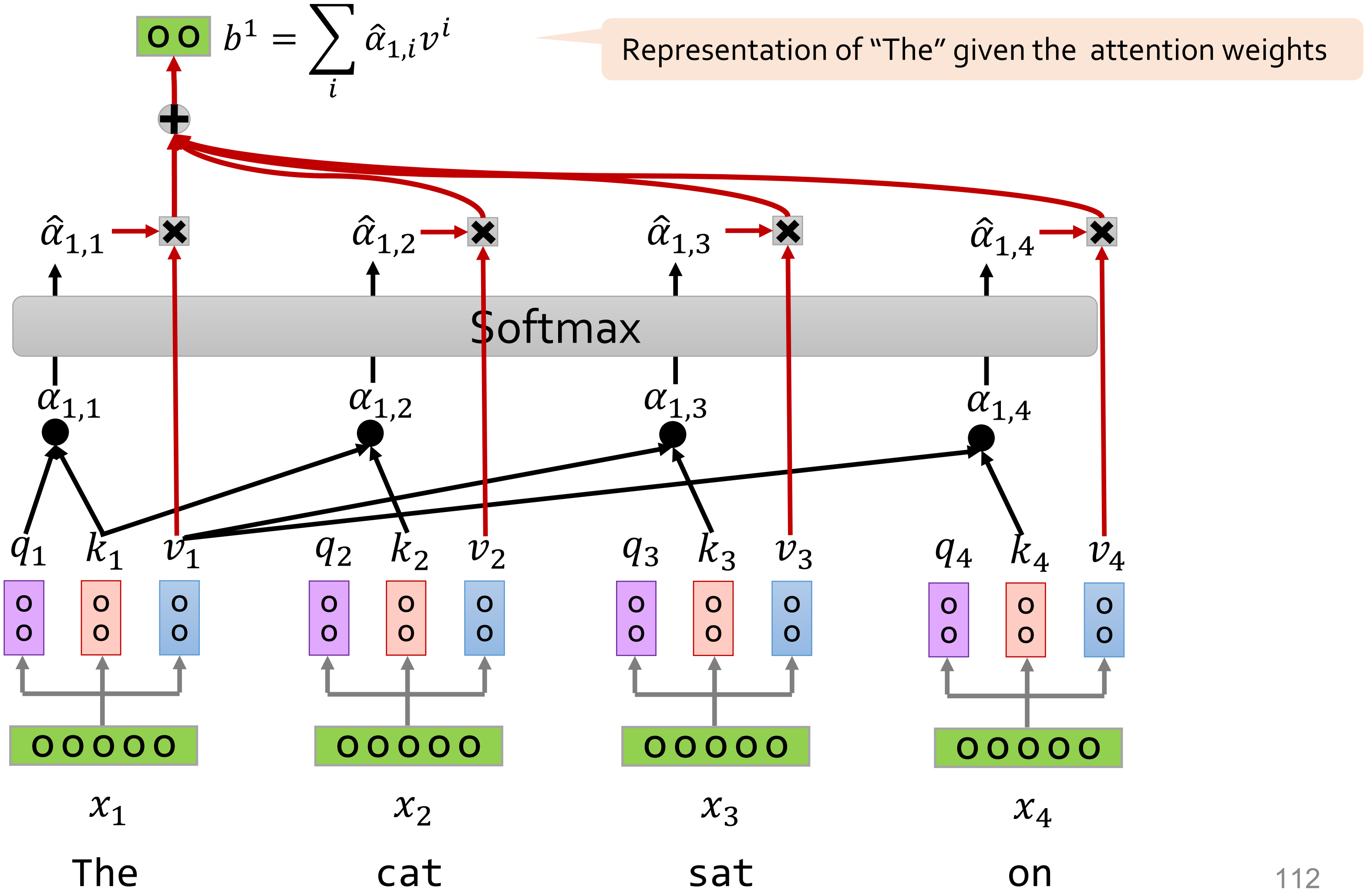
How much should "The" attend to other positions?



$$\sigma(z)_i = \frac{exp(z_i)}{\sum_j exp(z_j)}$$

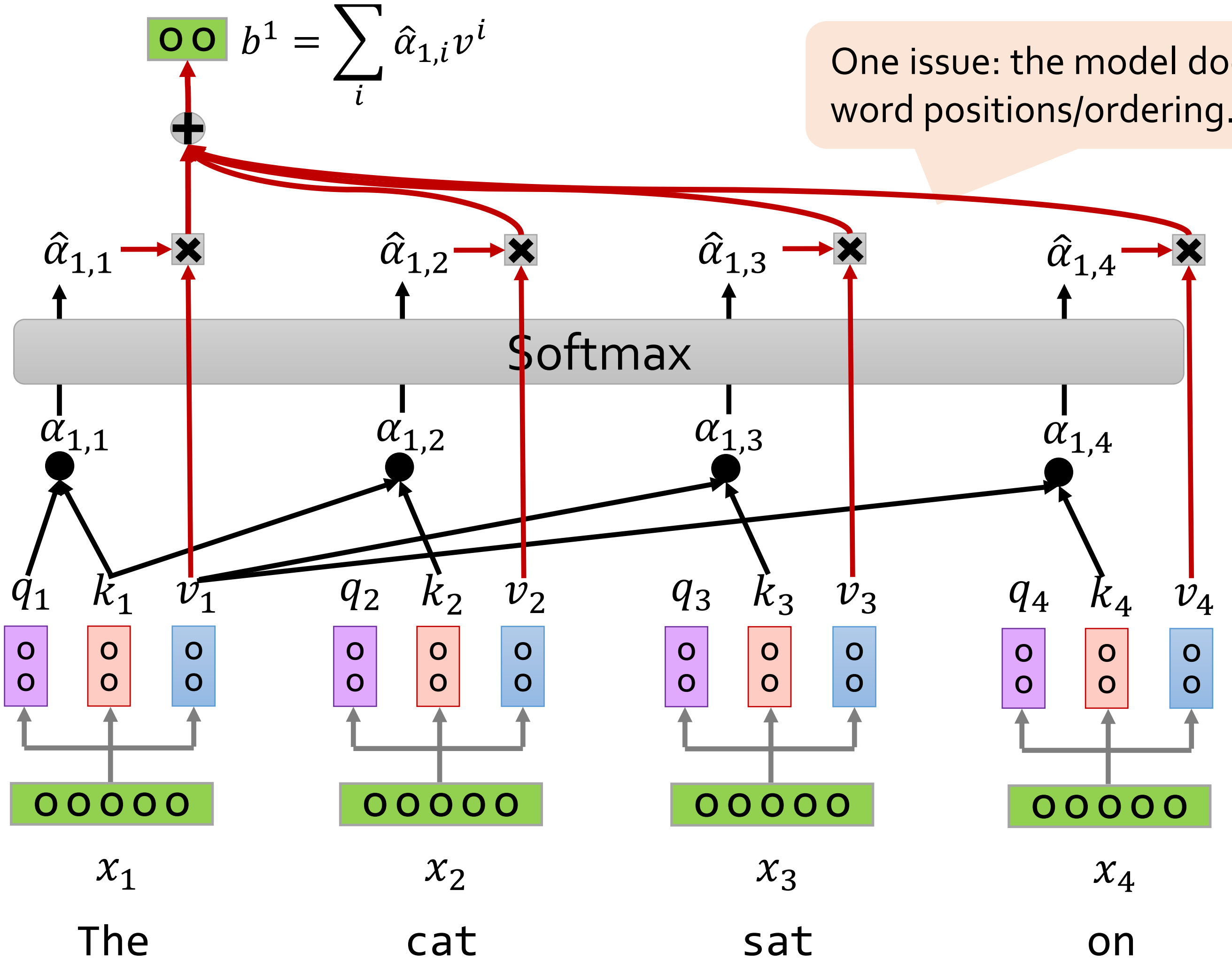
How much should "The" attend to other positions?





$b^1 = \sum_i \hat{\alpha}_{1,i} v^i$
Representation of "The" given the attention weights
$\hat{\alpha}_{1,1}$ $\hat{\alpha}_{1,2}$ $\hat{\alpha}_{1,3}$ $\hat{\alpha}_{1,4}$
Softmax
$\alpha_{1,1}$ $\alpha_{1,2}$ $\alpha_{1,3}$ $\alpha_{1,4}$
q_1 k_1 v_1 q_2 k_2 v_2 q_3 k_3 v_3 q_4 k_4 v_4
x_1 x_2 x_3 x_4
The cat sat on

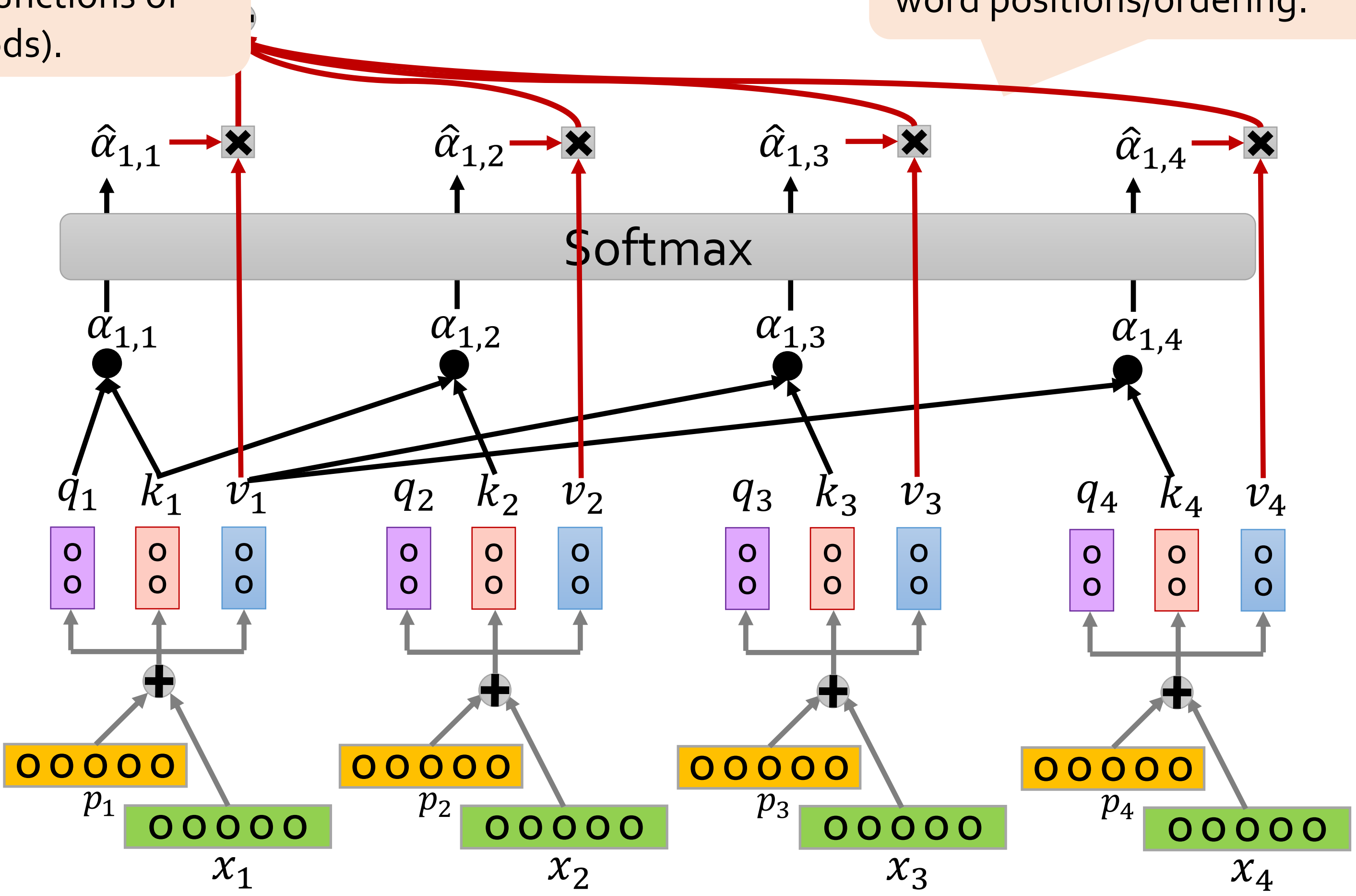
One issue: the model doesn't know word positions/ordering.



p_i are unique fixed vectors (sinusoidal functions of varying periods).

$$b^1 = \sum_i \hat{\alpha}_{1,i} v^i$$

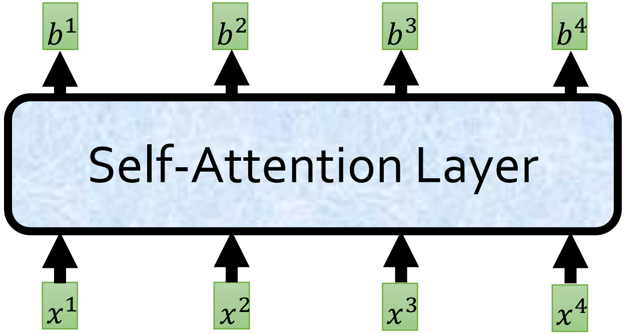
One issue: the model doesn't know word positions/ordering.



Self-Attention: Back to Big Picture

- Attention is a way to focus on particular parts of the input
- Can write it in matrix form:

$$\boldsymbol{b} = \mathrm{softmax}\left(\frac{QK^{\mathrm{T}}}{\alpha}\right)V$$



- Efficient implementations
- Better at maintaining long-distance dependencies in the context.

Self-Attention

$$\boldsymbol{b} = \mathrm{softmax}\left(\frac{QK^{\mathrm{T}}}{\alpha}\right)V$$

hardmaru
@hardmaru

The most important formula in deep learning after 2018

Self-Attention

What is self-attention? Self-attention calculates a weighted average of feature representations with the weight proportional to a similarity score between pairs of representations. Formally, an input sequence of n tokens of dimensions d, $X \in \mathbf{R}^{n\times d}$, is projected using three matrices $W_Q \in \mathbf{R}^{d\times d_q}$, $W_K \in \mathbf{R}^{d\times d_k}$, and $W_V \in \mathbf{R}^{d\times d_v}$ to extract feature representations Q, K, and V, referred to as query, key, and value respectively with $d_k = d_q$. The outputs Q, K, V are computed as

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V. \tag{1}$$

So, self-attention can be written as,

$$S = D(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right)V, \tag{2}$$

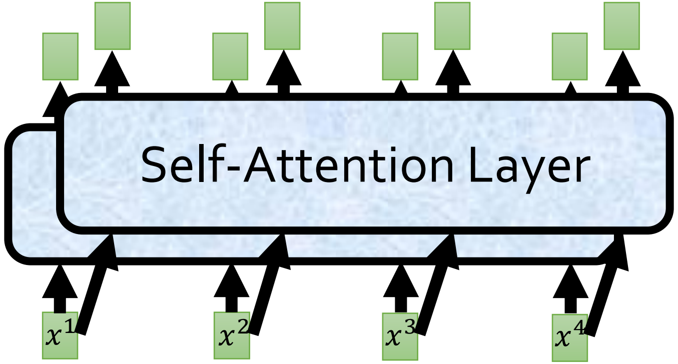
where softmax denotes a *row-wise* softmax normalization function. Thus, each element in S depends on all other elements in the same row.

9:08 PM · Feb 9, 2021 · Twitter Web App

553 Retweets **42** Quote Tweets **3,338** Likes

Multi-Headed Self-Attention

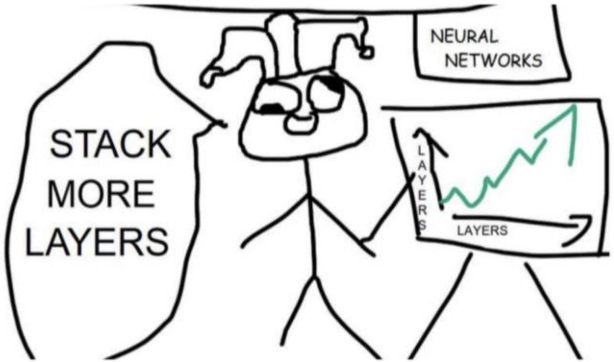
- Multiple parallel attention layers is quite common.
  - Each attention layer has its own parameters.

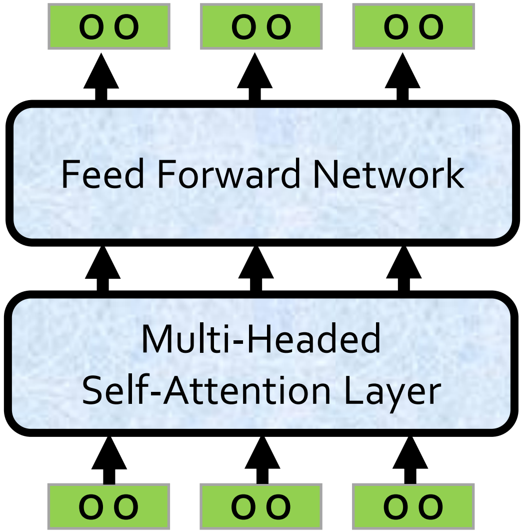


[Vaswani et al. 2017]

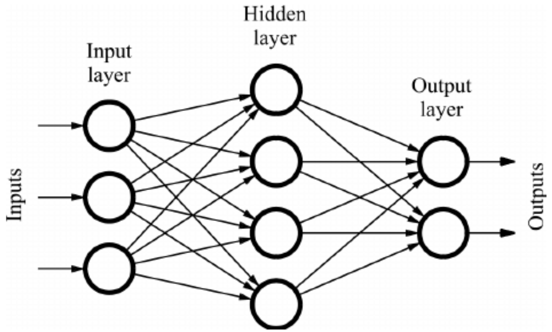
How Do We Make it Deep?

- Add a feed-forward network on top it to add more capacity/expressivity.
- Repeat!





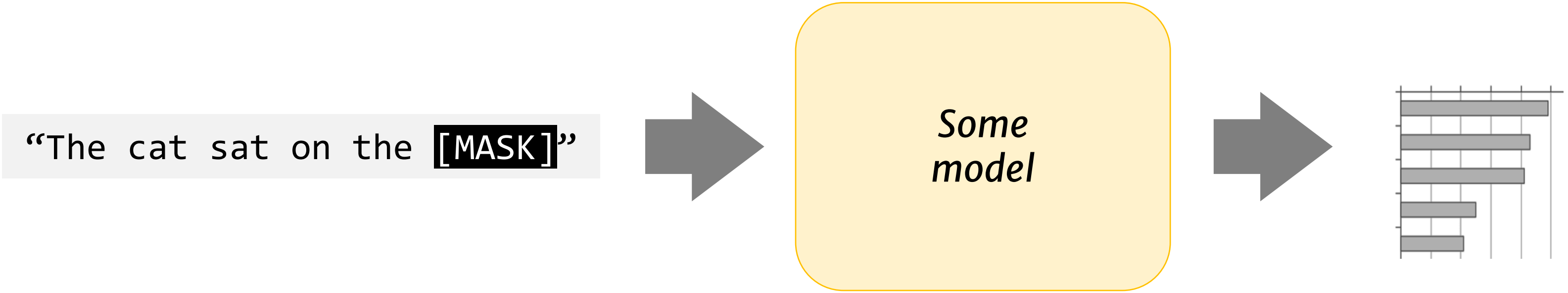
Feedforward Net: Refresher



A fully-connected network of nodes and weights.

Encoder-Decoder Architectures

- It is useful to think of generative models as two sub-models.

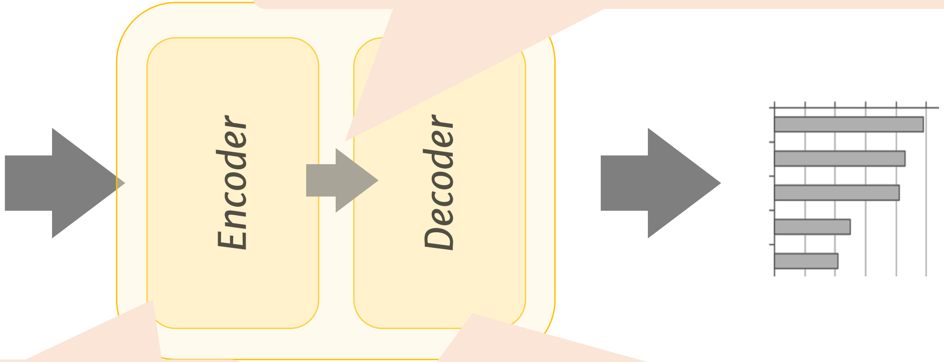


Encoder-Decoder Architectures

- It is useful to think of generative models as two sub-models.

Representation (compression) of the context

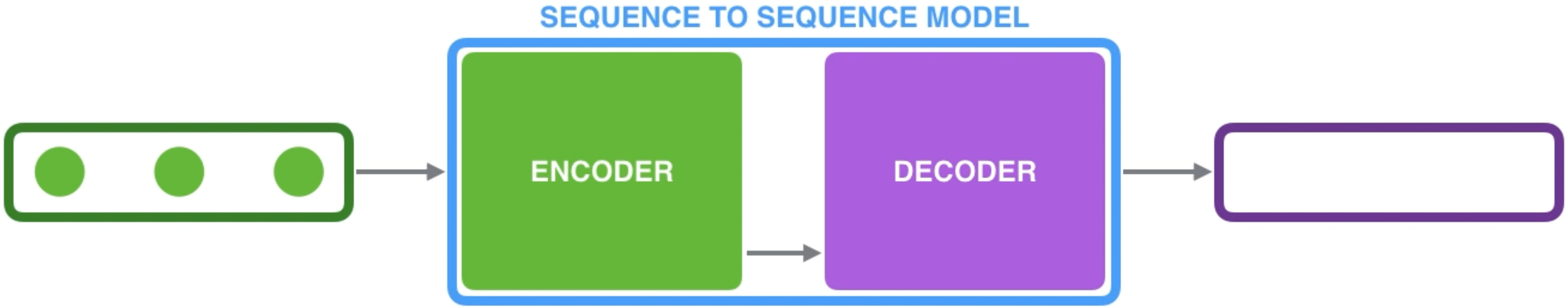
`“The cat sat on the [MASK]”`



Processes the context and compiles it into a vector.

Produces the output sequence item by item using the representation of the context.

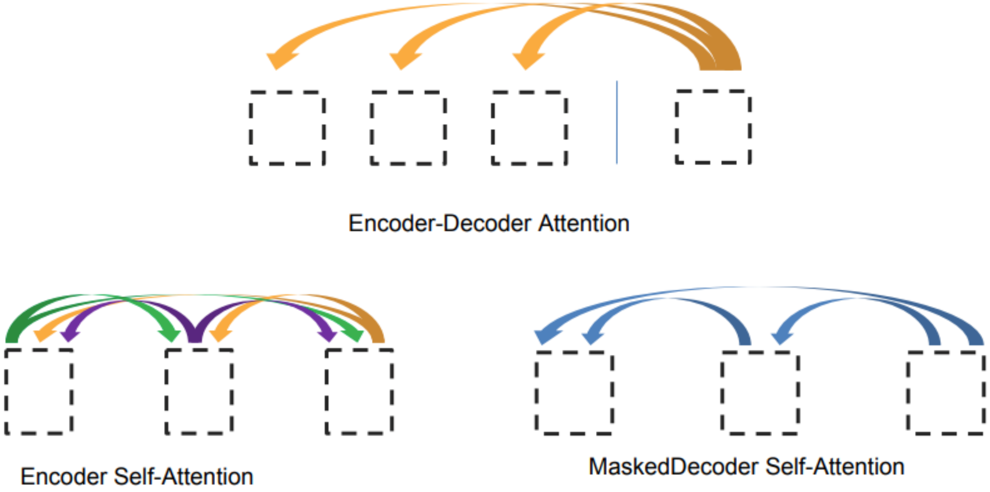
Encoder-Decoder Architectures

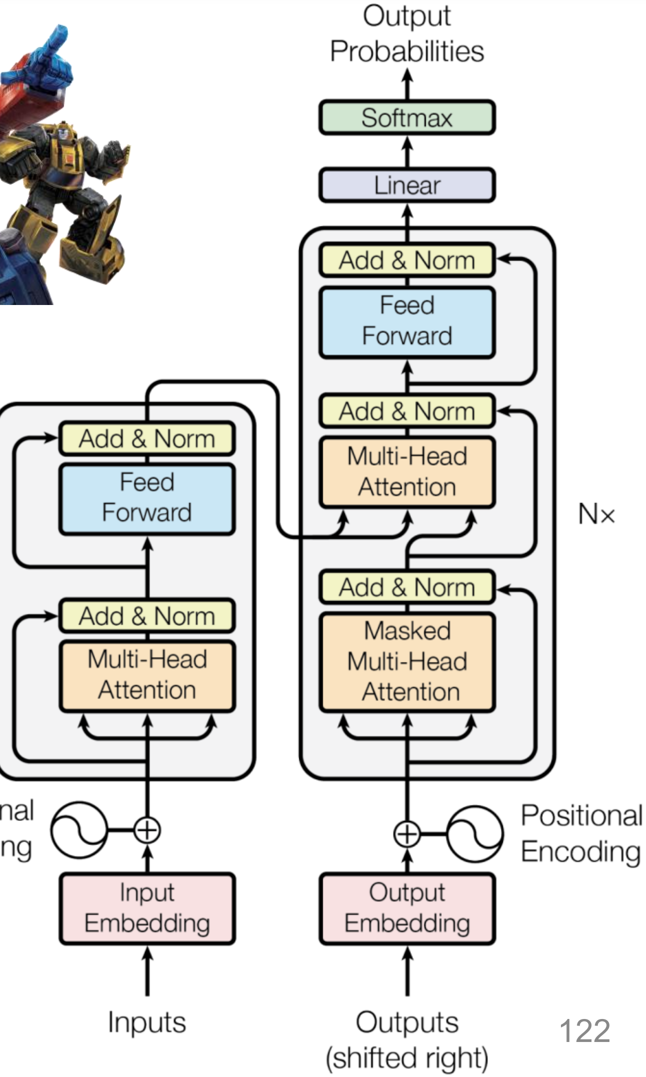


https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Transformer [Vaswani et al. 2017]

- An encoder-decoder architecture built with attention modules.
- 3 forms of attention





Impact of Transformers

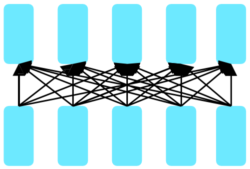
- Let to better predictive models of language!

Model	Layers	Heads	Perplexity
LSTMs (Grave et al., 2016)	-	-	40.8
QRNNs (Merity et al., 2018)	-	-	33.0
Transformer	16	16	19.8

["Efficient Content-Based Sparse Attention with Routing Transformers" Roy et al. 2020]

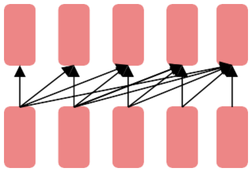
Impact of Transformers

- A building block for a variety of LMs



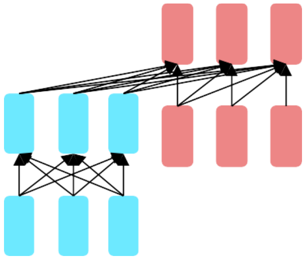
Encoders

- ❖ **Examples:** BERT, RoBERTa, SciBERT.
- ❖ Captures bidirectional context. Wait, how do we pretrain them?



Decoders

- ❖ **Examples:** GPT-2, GPT-3, LaMDA
- ❖ Other name: **causal or auto-regressive language model**
- ❖ Nice to generate from; can't condition on future words



Encoder-Decoders

- ❖ **Examples:** Transformer, T5, Meena
- ❖ What's the best way to pretrain them?

Transformer LMs + Scale = LLMs

- 2 main dimensions:
- Model size, pretraining data size

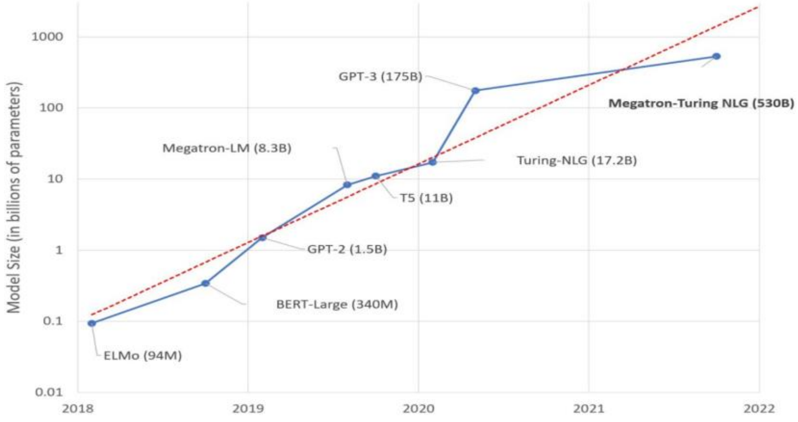
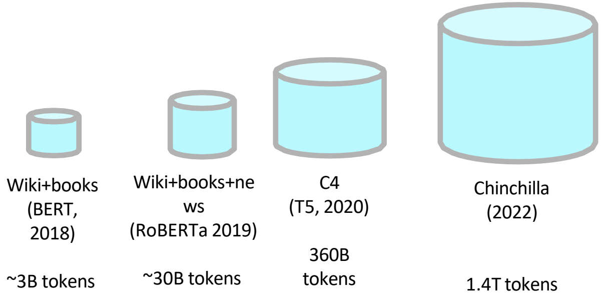
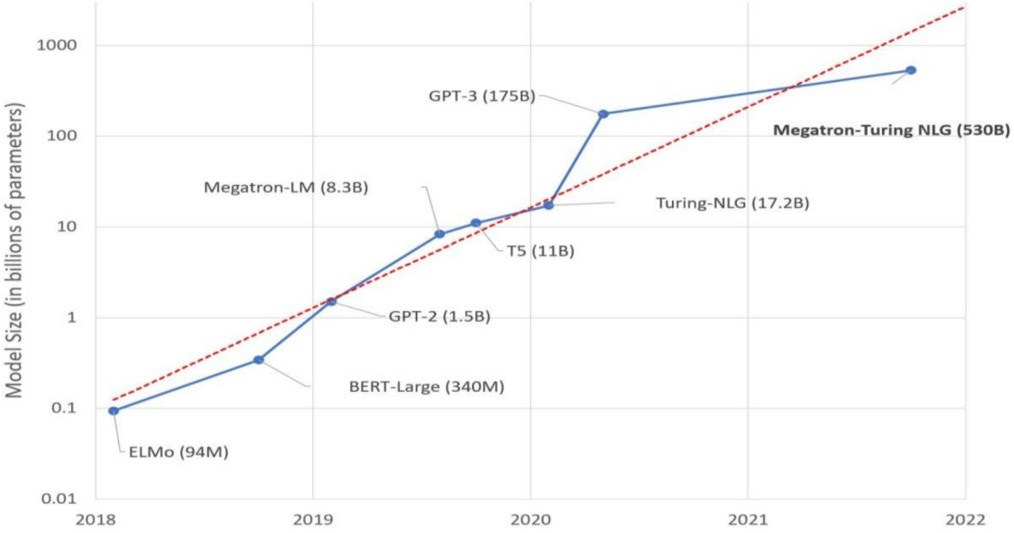


Photo credit: https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/



Large Language Models

- Not only they improved performance on many NLP tasks, but exhibited new capabilities



Questions?