CS 288 Advanced Natural Language Processing

Course website: cal-cs288.github.io/sp26

Ed: edstem.org/us/courses/92268

- Class starts at 15:40!
- **(If you just arrived and there's no seat)** Students who are currently not enrolled need to leave the room due to the building policy — so sorry about that! Please see the instructions on Ed

CS 288 Advanced Natural Language Processing

CS 288 Spring 2026

UC Berkeley

cal-cs288.github.io/sp26

Berkeley EECS

BAIR

Course Instructors

Sewon Min

(67%)

- Assistant Professor in EECS, since 2025
- Co-PI of BAIR and BerkeleyNLP
- First-time teaching the NLP class!

- Research: Large language model (LLM) architecture & training, data-model co-design
- PhD 2024 from University of Washington
- Previously: Meta (part-time, 4 yrs), Google (part-time, 0.5 yr), Currently at Ai2 (part-time)

Alane Suhr

(33%)

- Assistant Professor in EECS, since 2023
- Co-PI of BAIR and BerkeleyNLP
- Taught CS 288 in 2023 & 2024, and 183/283A in Fall 2025

- Research: NLP systems that use and acquire language in interaction, Multimodal & embodied NLP
- PhD 2022 from Cornell University
- 1 yr in Seattle at Ai2

Course GSIs

Akshat Gupta

- 3rd-year PhD student
- Advisor : Gopala Anumanchipalli

- Research: Continual Learning, Self-Improvement, Interpretability
- Previously: JPMorgan AI Research, MS CMU

Zineng Tang

- 3rd-year PhD student
- Advisor : Alane Suhr

- Research: Vision-Language, World models
- Previously: BS UNC Chapel Hill, Microsoft Research

Lecture plans

- What is this course? — 30min
- Course logistics — 20min
- First lecture (*n*-gram LM!) — 30min

What is **CS 288 Advanced NLP** about?

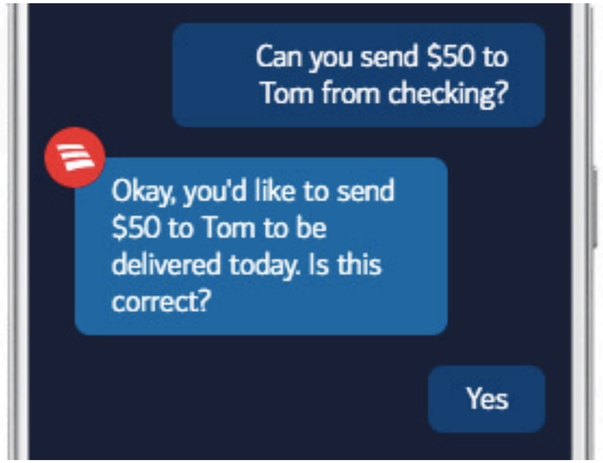
What is NLP?

- NLP = building **computer systems** that can process, understand, generate, and interact with **natural language** (**human language**) in a **genuinely meaningful way** — beyond surface patterns or just in a narrow domain.

What is NLP?

- NLP = building **computer systems** that can process, understand, generate, and interact with **natural language** (**human language**) in a genuinely meaningful way — beyond surface patterns or just in a narrow domain.

Communication with humans (ex. personal assistants, customer service)



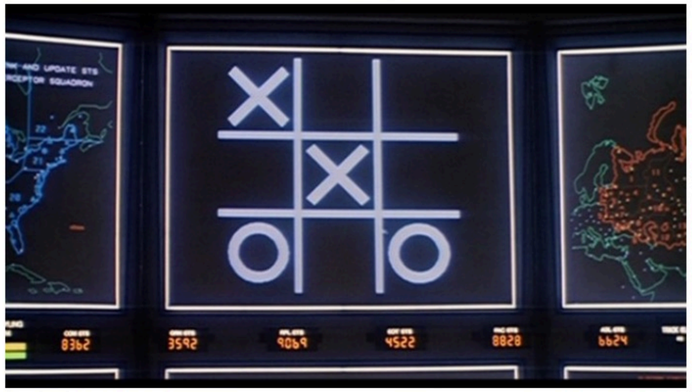
Access the wealth of information about the world — crucial for AI systems

Language is an Interface

Computer learns to play Civilization by reading the instruction manual

By Matthew Rogers on July 14, 2011 at 5:03 pm | 16 Comments

532 SHARES

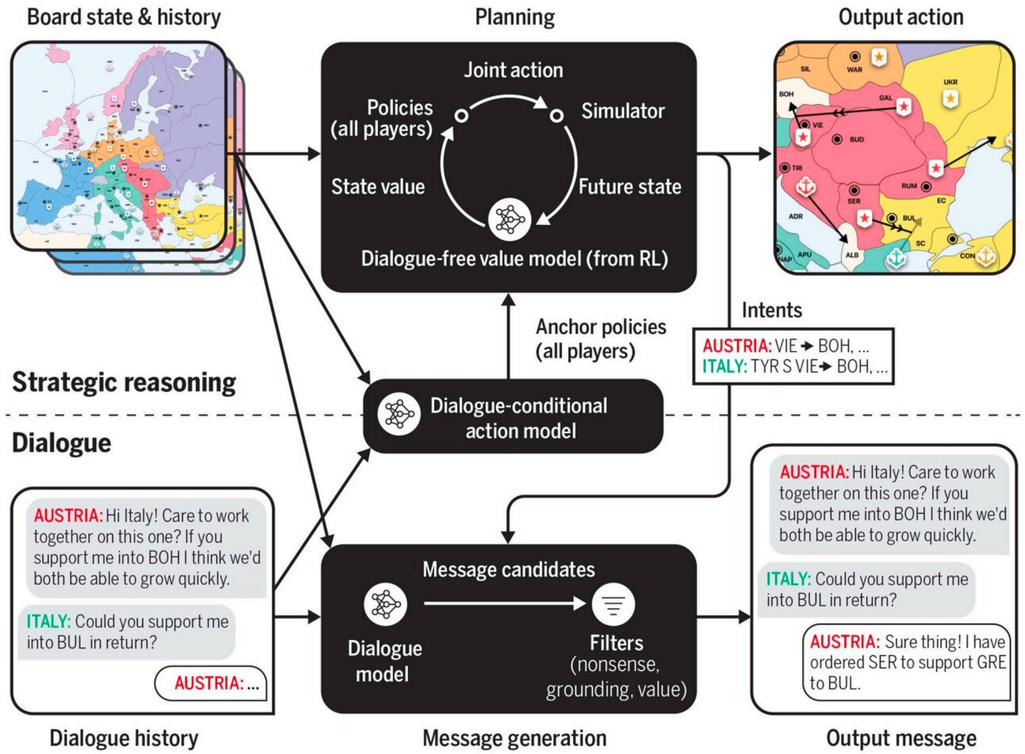


MIT researchers just got a computer to accomplish yet another task that most humans are incapable of doing: It learned how to play a game by reading the instruction manual.

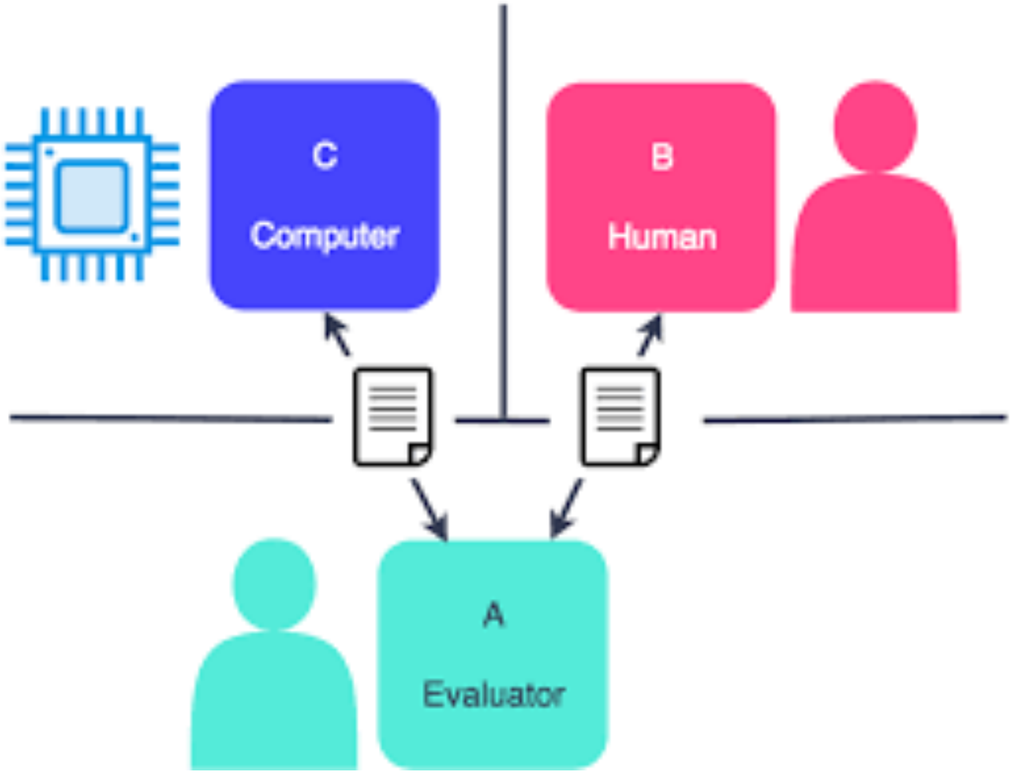
The MIT Computer Science and Artificial Intelligence lab has a computer that now plays Civilization

Meta's New AI Ranked in the Top 10% at the Game 'Diplomacy'—and Human Players Were None the Wiser

By Edd Gent › November 28, 2022



Turing test

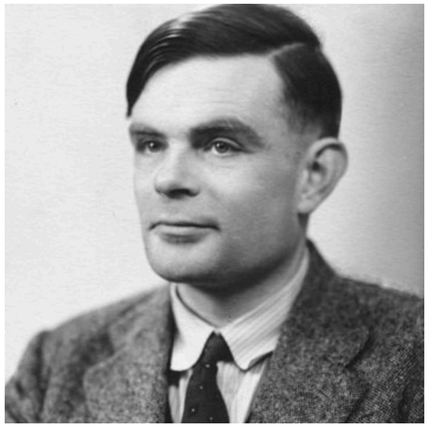


A. M. Turing (1950) Computing Machinery and Intelligence. *Mind 49:* 433-460.

COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

1. The Imitation Game



Ability to understand and generate language ~ intelligence

Why is language difficult to understand?

- Ambiguous
- Dialects
- Accents
- listener has to infer - pragmatics
- humor, sarcasm, irony
- context, dependencies

Ambiguity

Syntactic Ambiguity

Ambiguity

Syntactic Ambiguity

ellipsis

noun (plural **ellipses**)

the omission from speech or writing of a word or words that are superfluous or able to be understood from contextual clues: *it is very rare for an ellipsis to occur without a linguistic antecedent*

Ambiguity

Semantic Ambiguity

Dictionary

Definitions from Oxford Languages · Learn more

back·ground

/ˈbakˌground/

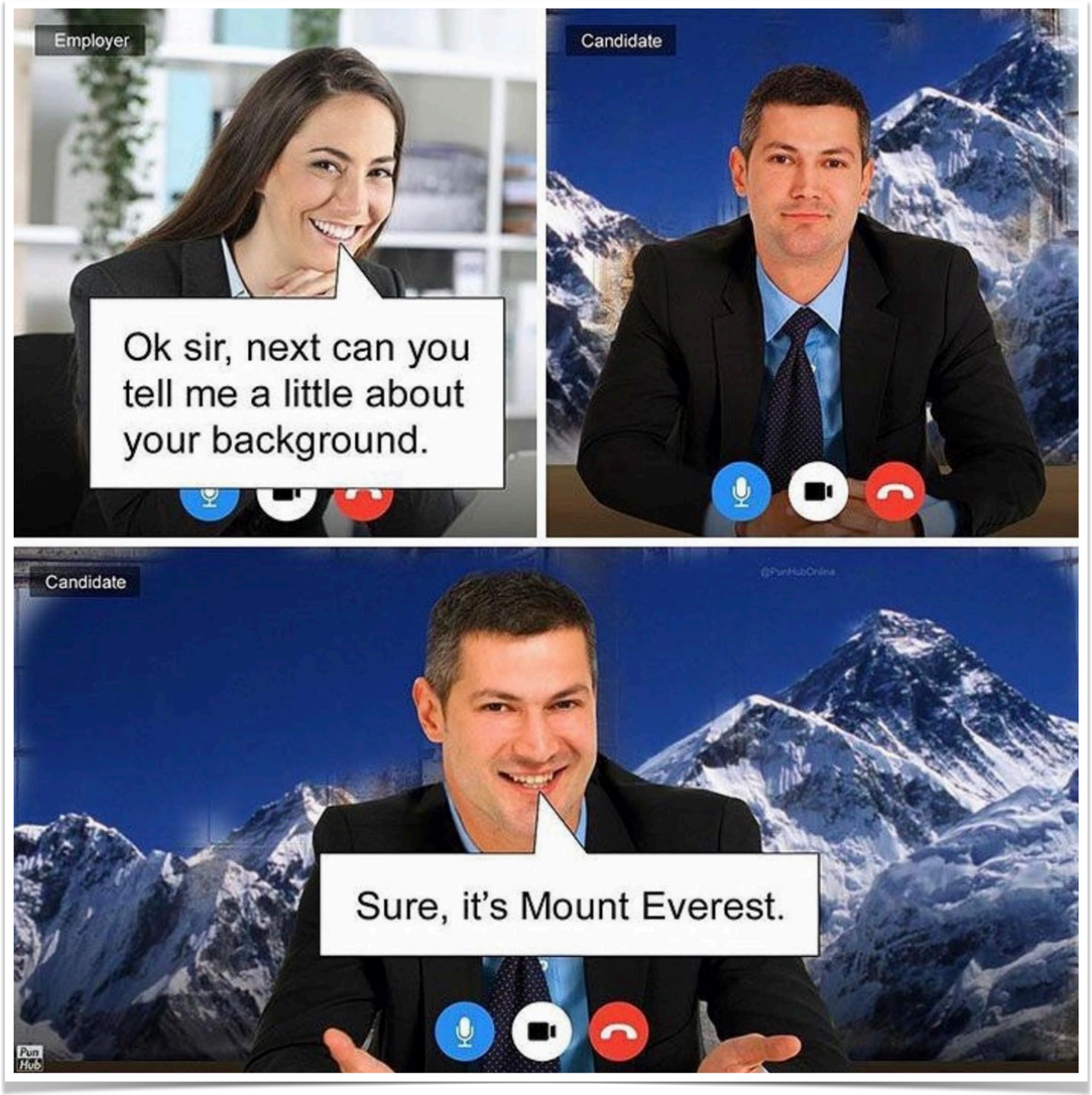
noun

1. the area or scenery behind the main object of contemplation, especially when perceived as a framework for it.
   "the house stands against a background of sheltering trees"

   Similar: surrounding(s) backdrop backcloth framework scene

2. the circumstances or situation prevailing at a particular time or underlying a particular event.
   "the political and economic background"

   Similar: circumstances context conditions situation environment



Ambiguity

Semantic Ambiguity

Meaning 1: To Shine/Refine (pronounced /ˈpɑːlɪʃ/)

- **Verb:** To make something smooth and glossy by rubbing, or to improve/perfect something.
  - *Example:* "She spent hours **polishing** her shoes."
  - *Example:* "He needed to **polish** his speech before the presentation."
- **Noun:** A substance used for shining (like furniture polish) or the state of being smooth and shiny.
  - *Example:* "The table had a nice wood **polish**."

Meaning 2: From Poland (pronounced /ˈpəʊlɪʃ/)

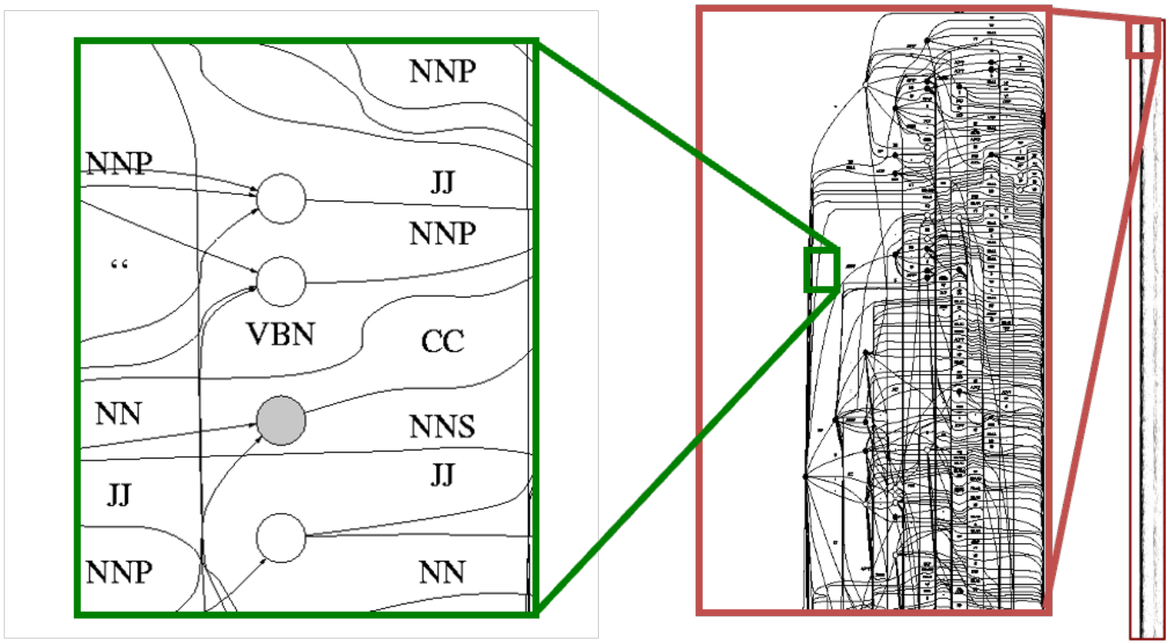
- **Proper Noun (Capital P):** Pertaining to Poland, its people, or its language.
  - *Example:* "She is a **Polish** citizen."
  - *Example:* "He speaks **Polish** fluently."

Ambiguity

Pragmatic Ambiguity

Scale

- People did know that language was ambiguous!
  - …but they hoped that all interpretations would be “good” ones
  - …they didn’t realize how bad it would be

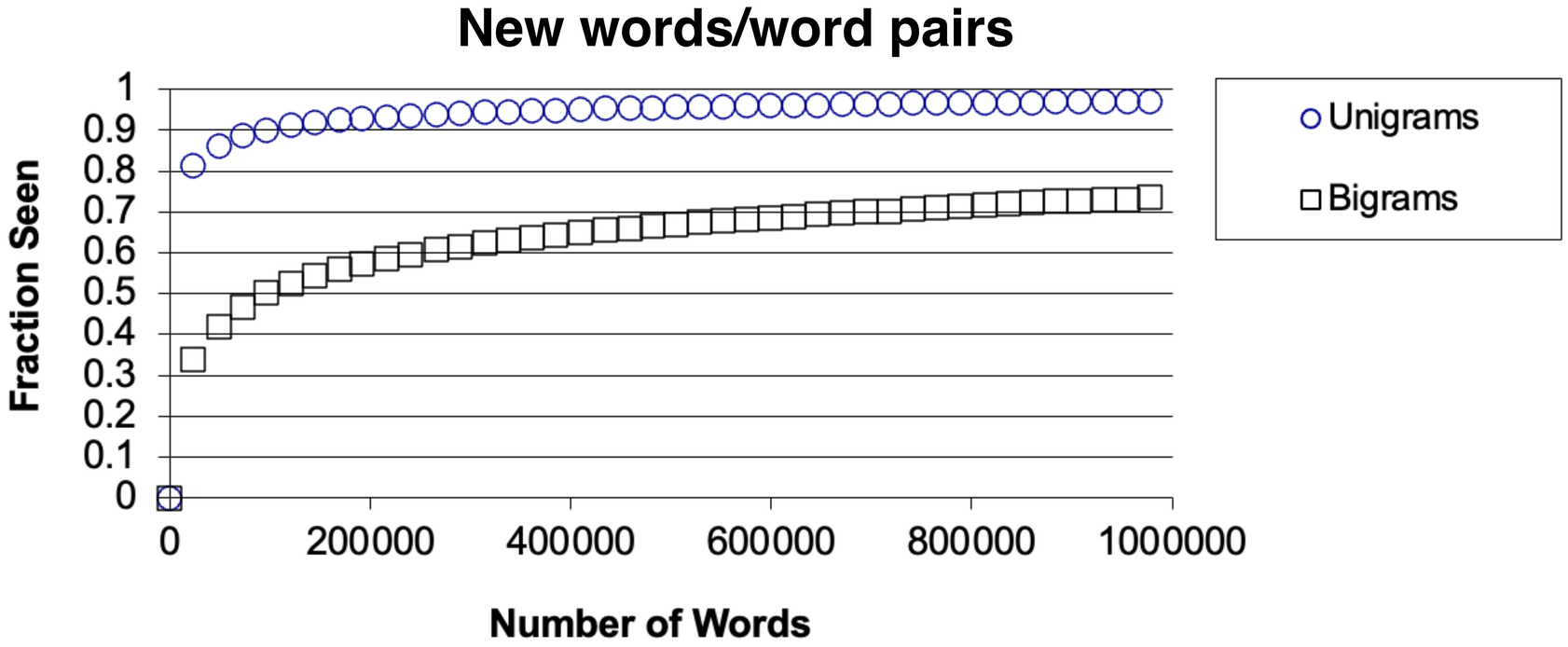


Corpus

- A corpus is a collection of text
  - Annotated vs. raw text
  - Balanced vs. uniform corpora
- Examples:
  - Newswire collections: 500M+ words
  - Brown corpus: 1M words of tagged “balanced” text
  - Penn Treebank: 1M words of parsed WSJ
  - Canadian Hansards: 10M+ words of aligned French / English sentences
  - RedPajama: 30T raw tokens
  - The entire Web?

Corpus: Sparsity

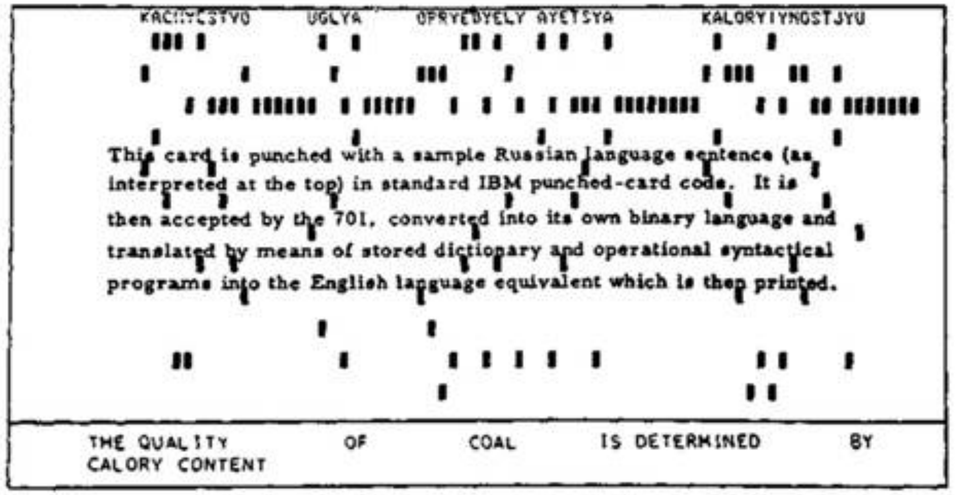
- Even as raw corpora become extremely large …
  - … sparsity is always a problem



A brief history of NLP

NLP is almost as old as modern computers

Weaver's memorandum (Shannon and Weaver, 1949) — machine translation (MT)



Specimen punched card and below a strip with translation, printed within a few seconds

Georgetown experiment
1954

"Within three or five years, machine translation will be a solved problem"

Pre-Statistics

(1) Colorless green ideas sleep furiously.

(2) Furiously sleep ideas green colorless

Which sentence sounds more like English?

- (1)
- (2)
- Equal

Which sentence is likely to appear more frequently in the corpus?

- (1)
- (2)
- Equal

- Neither of these sentences had likely ever said before in human history (probability = 0).
- Therefore, a statistical model would treat them both as equally “wrong”
- However, a native English speaker is likely to see (1) as “English” and (2) as “gibberish”

Pre-Statistics

(1) Colorless green ideas sleep furiously.

(2) Furiously sleep ideas green colorless

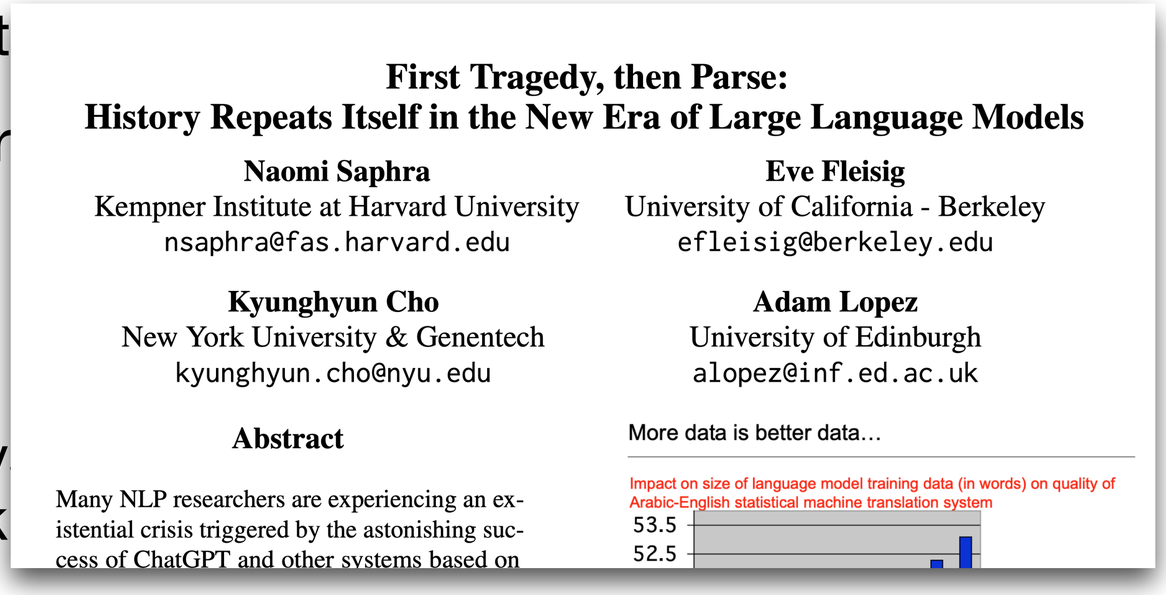
It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) had ever occurred in an English discourse. Hence, in any statistical model for grammaticalness, these sentences will be ruled out on identical grounds as equally "remote" from English. Yet (1), though nonsensical, is grammatical, while (2) is not.” (Chomsky 1957)

Pre-Statistics

- 70s and 80s: more linguistic focus
  - Emphasis on deeper models, syntax and semantics
  - Manually engineered systems (i.e., rule based)
  - Toy domains
  - Weak empirical evaluation

1990s Empirical Revolutions

- Corpus-based methods produce the first
- Deep linguistic analysis often traded for r
- Empirical evaluation is essential



First Tragedy, then Parse:
History Repeats Itself in the New Era of Large Language Models

Naomi Saphra
Kempner Institute at Harvard University
nsaphra@fas.harvard.edu

Eve Fleisig
University of California - Berkeley
efleisig@berkeley.edu

Kyunghyun Cho
New York University & Genentech
kyunghyun.cho@nyu.edu

Adam Lopez
University of Edinburgh
alopez@inf.ed.ac.uk

Abstract

Many NLP researchers are experiencing an existential crisis triggered by the astonishing success of ChatGPT and other systems based on

More data is better data…

Impact on size of language model training data (in words) on quality of Arabic-English statistical machine translation system

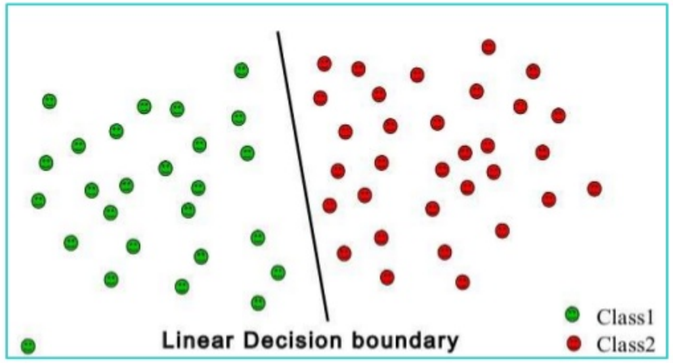
53.5
52.5

"Whenever I fire a linguist our sy
– Jelinek

"Of course, we must not go overboard and mistakenly conclude that the successes of statistical NLP render linguistics irrelevant (rash statements to this effect have been made in the past, e.g., the notorious remark, "Every time I fire a linguist, my performance goes up"). The information and insight that linguists, psychologists, and others have gathered about language is invaluable in creating high-performance broad-domain language understanding systems …"

- Lillian Lee (2001) http://www.cs.cornell.edu/home/llee/papers/cstb/index.html

2000s and 2010s: Use of machine learning in NLP



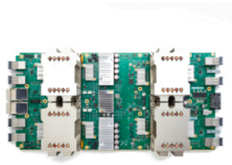
- Increase in computational capabilities
- Availability of electronic corpora

- 2000s: Richer linguistic representations used in statistical approaches, scale to more data
  - Machine translations starts to work well!
- 2010s: Excitement about neural networks (again), pre-trained representations
  - Robust speech recognition and personal assistants show up

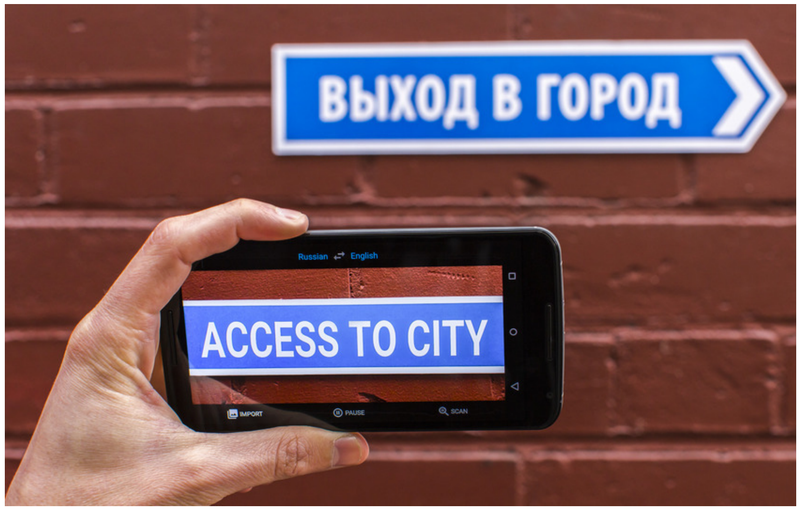
The deep learning era

- Significant advances in core NLP technologies
- Essential ingredient: large-scale supervision, lots of compute
- Reduced manual effort - less/zero **feature engineering**

GPU



TPU



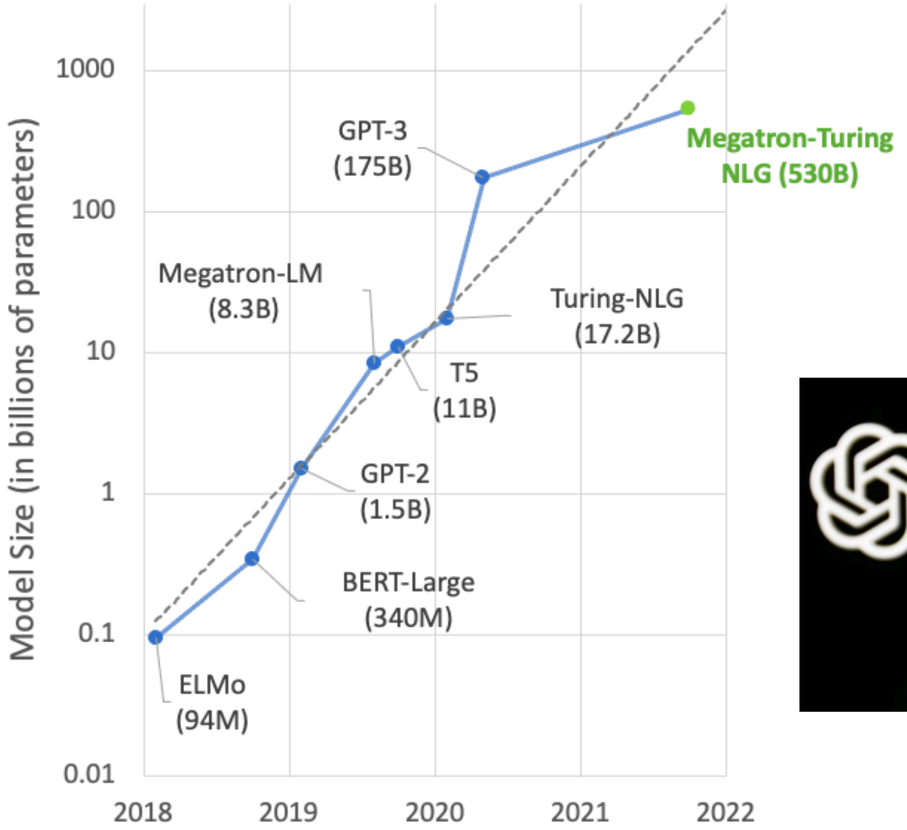
36M sentence pairs

Russian: Машинный перевод - это круто!

English: Machine translation is cool!

The era of pre-training / LLMs

BERT, ELMo, ERNIE…



- Leverages a lot of unlabeled text
- Model size increased by $10^3 - 10^5$x in parameters

The industrialization of NLP

The industrialization of NLP

GPT-4 supposedly has 1.8T parameters. [article]
GPT-5 supposedly cost $1.7 to $2.5 billion to train [HSBC].
Meta builds cluster with 600,000 Nvidia GPUs (January 2024). [article]
Stargate (OpenAI, NVIDIA, Oracle) invests $500B over 4 years. [article]

The industrialization of NLP

Also, there are no public details on how frontier models are built.

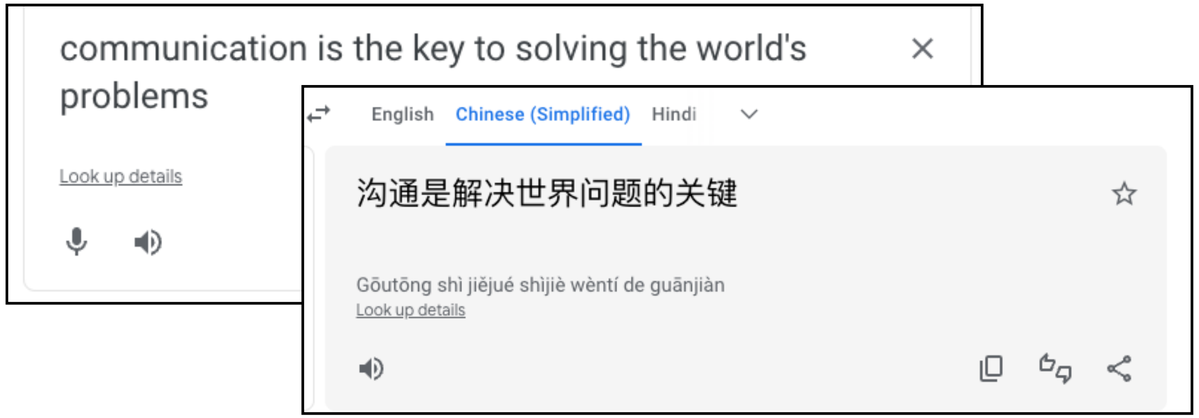
From the GPT-4 technical report [OpenAI+ 2023]:

2 Scope and Limitations of this Technical Report

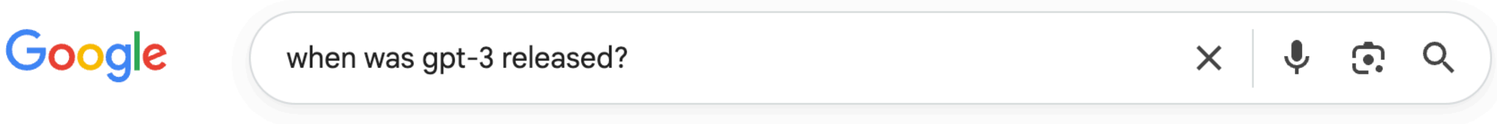
This report focuses on the capabilities, limitations, and safety properties of GPT-4. GPT-4 is a Transformer-style model [39] pre-trained to predict the next token in a document, using both publicly available data (such as internet data) and data licensed from third-party providers. The model was then fine-tuned using Reinforcement Learning from Human Feedback (RLHF) [40]. Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method, or similar.

We are committed to independent auditing of our technologies, and shared some initial steps and ideas in this area in the system card accompanying this release.[2] We plan to make further technical details available to additional third parties who can advise us on how to weigh the competitive and safety considerations above against the scientific value of further transparency.

But actually, that always has been the case
[in most of the AI fields]



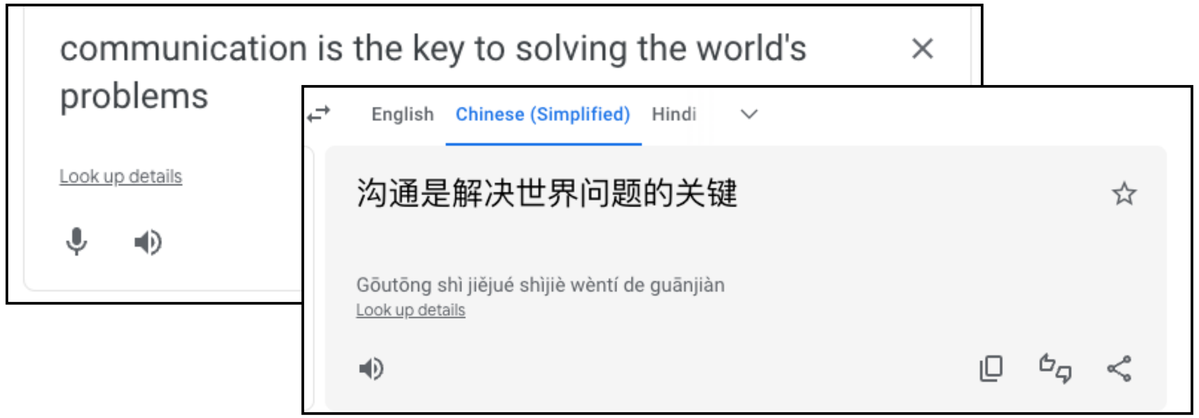
Machine Translation



Information retrieval

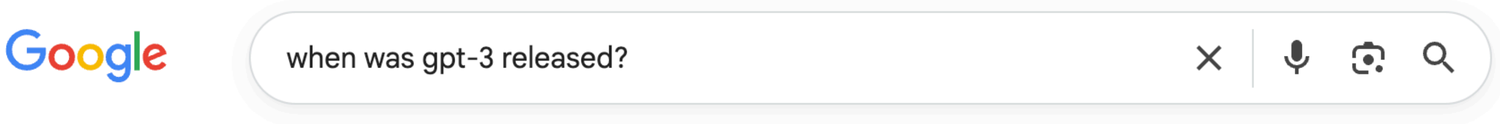
But actually, that always has been the case
[in most of the AI fields]

And yet, academic education and research have transformed them



Machine Translation

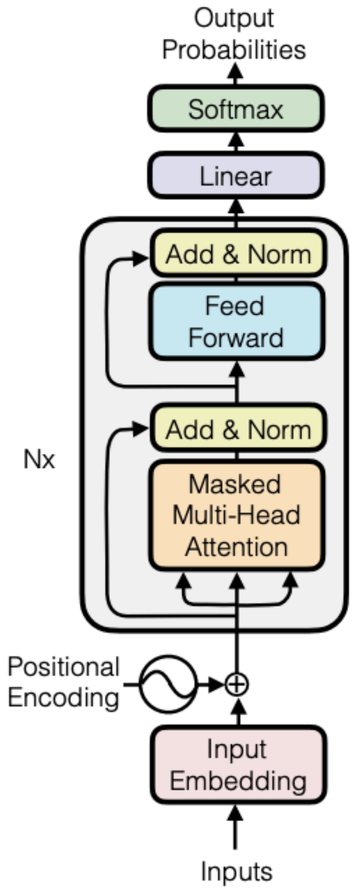
- Statistical MT
- Neural MT (actually only launched in 2016)
- LLM-based MT

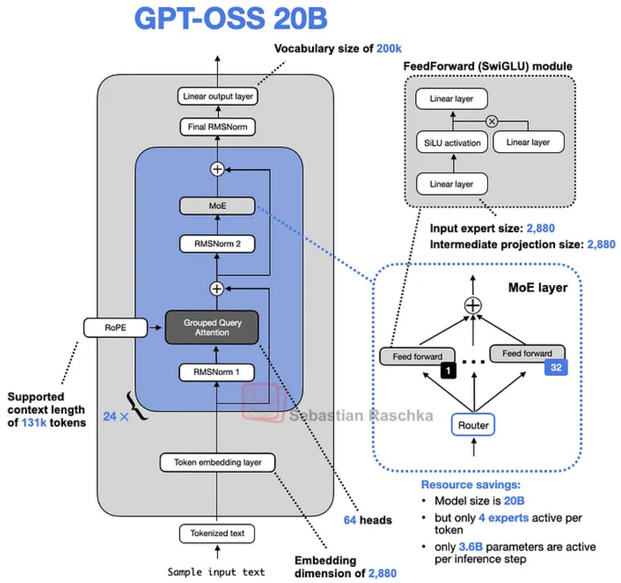


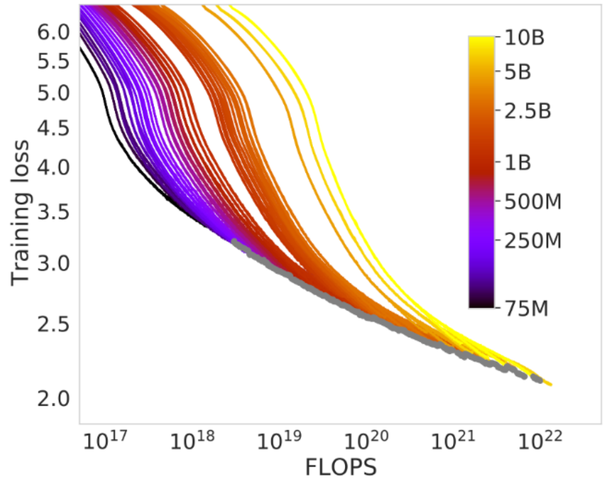
Information retrieval

- Rules, graphs, PageRank
- Neural retrieval (actually only launched in 2016; pre-trained model based in 2019)
- LLM-based retrieval

We can learn about these models







	Vaswani et al.	Llama	Llama 2
Norm Position	Post-norm	Pre-norm	Pre-norm
Norm Type	LayerNorm	RMSNorm	RMSNorm
Activation	ReLU	SwiGLU	SwiGLU
Positional Encoding	Sinusoidal	RoPE	RoPE
Attention	Multi-head	Multi-head	Grouped-query

We can learn about these models

Meta

The Llama 3 Herd of Models

Llama Team, AI @ Meta[1]

[1]A detailed contributor list can be found in the appendix of this paper.

Qwen3 Technical Report

Qwen Team

https://huggingface.co/Qwen
https://modelscope.cn/organization/qwen
https://github.com/QwenLM/Qwen3

DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models

DeepSeek-AI

research@deepseek.com

DataComp-LM: In search of the next generation of training sets for language models

Olmo 3

Olmo Team*

Olmo 3 Base: Olmo-3-1025-7B Olmo-3-1125-32B
Olmo 3 Think: Olmo-3-7B-Think Olmo-{3|3.1}-32B-Think
Olmo 3 Instruct: Olmo-3-7B-Instruct Olmo-3.1-32B-Instruct
Olmo 3 RL Zero: Olmo-3-7B-RL-Zero-{Math|Code|IF|General|Mix} Olmo-3.1-7B-RL-Zero-{Math|Code}
Base Data: Pretrain: Dolma 3 Mix Midtrain: Dolma 3 Dolmino Mix Long-ctx: Dolma 3 Longmino Mix
Think Data: Dolci-Think-{SFT|DPO|RL}-7B Dolci-Think-{SFT|DPO|RL}-32B
Instruct Data: Dolci-Instruct-{SFT|DPO|RL}
RL-Zero Data: Dolci-RL-Zero-{Math|Code|IF|General}-7B Dolci-RL-Zero-Mix-7B
Training Code: OLMo-core (pretrain) Open Instruct (posttrain)
Data Code: datamap-rs (data processing) duplodocus (deduplication) dolma3 (data recipes)
Eval Code: OLMES (eval suite) decon (eval decontamination)
Training Logs: Olmo-3-7B-{Base|Think|Instruct|RL-Zero} Olmo-3-32B-{Base|Think|Instruct}
Demo: 32B Think 32B Instruct 7B Think 7B Instruct
Contact: olmo@allenai.org

Fortunately, people still share about their technical findings, and people still build on top of each other's work.

Goal of this course

- Context: How modern NLP models arrived where they are today.
- Mechanics: How today's NLP systems work (e.g., what a Transformer is, training pipelines, data curation).
- Mindset: Taking language seriously. Take scale seriously. Learn how to think and reason at scale.
- Intuition: Which modeling decisions tend to yield good accuracy (partially — may not transfer across scales).

Intuitions?

Some design decisions are simply not (yet) justifiable and just come from experimentation.

Example: Noam Shazeer paper that introduced SwiGLU

4 Conclusions

We have extended the GLU family of layers and proposed their use in Transformer. In a transfer-learning setup, the new variants seem to produce better perplexities for the de-noising objective used in pre-training, as well as better results on many downstream language-understanding tasks. These architectures are simple to implement, and have no apparent computational drawbacks. We offer no explanation as to why these architectures seem to work; we attribute their success, as all else, to divine benevolence.

The bitter lesson

Rich Sutton

March 13, 2019

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore's law, or rather its generalization of continued exponentially falling cost per unit of computation. Most AI research has been conducted as if the computation available to the agent were constant (in which case leveraging human knowledge would be one of the only ways to improve performance) but, over a slightly longer time than a typical research project, massively more computation inevitably becomes available. Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation. These two need not run counter to each other, but in practice they tend to. Time spent on one is time not spent on the other. There are psychological commitments to investment in one approach or the other. And the human-knowledge approach tends to complicate methods in ways that make them less suited to taking advantage of general methods leveraging computation. There were many examples of AI researchers' belated learning of this bitter lesson, and it is instructive to review some of the most prominent.

Wrong interpretation: scale is all that matters, algorithms don’t matter

Right interpretation: algorithms that scale is what matters.

Course Logistics

Course Info

Lectures: Tues/Thurs 15:30—17:00 (15:40—17:00 considering Berkeley Time) at SODA 306

Instructor OH: Right after the lectures at SODA 347

GSI OH: Monday (12:30-1), Wednesday (11:30-12), virtual (zoom link on the website)

Course webpage: https://cal-cs288.github.io/sp26/

- Contains all the detailed information about the course (slides, assignments, project, policy, office hours, etc).
- Ed will be used for all announcements
  - Make sure you have **notifications** turned on!
  - All questions should also go to Ed

This is a **graduate-level** class.

CS 288 has been offered for many years, but this year, we re-made the class (lectures, assignments, etc)

Course topics

(Listing a subset only — the full schedule on the course website)

Context: How modern NLP models arrived where they are today (33%)

+ Assignment 1

- Statistical language models
- How to represent a word into a vector? Then, how to plug them into neural networks?
- NLP model architecture before Transformers; How did we come up with attention?

Mechanics: How today's NLP systems (LLMs) work (33%)

- Transformers: The original, and the modern versions
- Encoder, decoder, and encoder-decoder architectures
- Pre-training (masked LM, casual LM), fine-tuning, in-context learning
- Scaling laws
- Data curation
- Post-training

+ Assignment 2

Advanced Topics (33%)

+ Assignment 3

- Retrieval and Retrieval-augmented generation (RAG)
- Mixture-of-experts (MoEs)
- Test-time compute & reasoning models
- LLM agents
- Vision-language models
- Interactive embodied agents
- Impact & social implication

What this class is ***not*** about (1/2)

Do not take this class if you want to learn about the hottest LLM techniques (e.g., reasoning models, agents, RL, …)

- This class is primarily about **comprehensive, foundational NLP** build **bottom up**
- We'll only have one lecture on "reasoning models" and one or two on "agents"
- For cutting-edge topics, take a seminar class

Do note take this class if you want to get good results on your own application domain

- You should just use APIs, or prompt or fine-tune one of the models from Huggingface
  - Tons of online tutorials out there — you really don't need a semester-long course
- We do not use API for any course assignment, and we don't teach you how to use them
- Projects that rely sole on APIs not accepted (using APIs as a supporting tool, e.g., for data preparation or evaluation, is allowed).

What this class is ***not*** about (2/2)

Majority of this class is about neural networks, deep learning, and large language models

- No linguistics, syntax parsing, semantic parsing, or other structured prediction tasks (we will briefly survey what kinds of structured prediction problems exist)
- No classical NLP models and algorithms (e.g., IBM models, CKY, PCFG, CCG, …)
- See previous offerings of CS 288 for self-study, or INFO 159/259

This class focuses on **text** and **human language**

- One lecture about multimodality, but that's not a central focus
- One lecture about speech, but that's not a central focus
- Programming languages are largely out of scope (although similar approaches can be applied)
- If you'd like to do the final project on these topics without involving text at all, permission request in the project abstract submission is required

Prerequisites

Machine learning and deep learning knowledge equivalent to CS 189 is prerequisites.

We assume you have learned the following concepts already:

- Logistic regression w/ regularization
- Unsupervised vs supervised learning
- Feedforward neural networks
- Proficiency in Python, Numpy and PyTorch

All details on the course website

Enrollment Policy

To people who cannot directly enroll (incl. undergrad students, non-EECS grad students, or EECS grad students who tried enrolling late):

- Google Form on the website (don't simply be on the waitlist!)
- Selected students will be notified via email in batches, before the 5th class (Tuesday, 02/03)
- Students likely need both (1) an A in CS 189/EECS 183/283A and (2) relevant research with Berkeley research groups
    - To increase the chance, make sure to update your Google Form response
- Emailing course staff will not increase the chance

All details on the course website

Assignments

Assignment 1: n-gram language models and perceptron classifiers

Release date: 01/27 (Tue) / Due date: 02/10 (Tue)

Assignment 2: Build your own language model

Release date: 02/10 (Tue) / Due date 02/24 (Tue)

Assignment 3: Build your own RAG model (group assignment)

Release date: 03/03 (Tue) / Early milestone due date: 03/17 (Tue) / Due date: 03/19 (Thu)

All deadlines are at **5:59 PM PST**

A total of 6 late days: you can use up to 3 late days per assignment

All details on the course website

Final project

A team of 3

Five milestones

- Team registration/match request (the same team will be used for A3) — Due 02/10 (Tue)
- Checkpoint 1: Abstract submission (<250 words), mainly to ensure the topic is in scope — Due 03/03 (Tue)
- Checkpoint 2: Midpoint project report (2—3 pages), requiring literature review, experimental design, baselines, and (quantitative) preliminary results — Due 04/19 (Thu)
- Checkpoint 3: Project presentation (04/28 and 04/30)
- Checkpoint 4: Project report (6—8) pages — Due 05/07 (Thu)

More information on the course website

All details on the course website

Compute

- GCloud credits ($50/student) — instructions will be posted on Ed shortly
- Tinker — API for fine-tuning ($250/student) — instructions will be posted on Ed shortly
- VESSL AI credits (1,250 A100 hours each for two selected teams): Teams will be selected based on merits after receiving Checkpoint 2 (Midpoint project report)

Beyond this, no compute will be provided. We highly encourage choosing the project topic in consideration of your own compute availability.

All details on the course website

Grading

- Quizzes/Participation: 5%
- Assignments (55%)
  - Assignment 1: 15%
  - Assignment 2: 20%
  - Assignment 3: 20%
- Final Project: 40%
  - Abstract: 5%
  - Midpoint report: 5%
  - Presentation: 10%
  - Final report: 20%

All details on the course website

Guest Lectures

Huan Sun (April 9)
“Capability and Safety of Computer-Use Agents”

Jack Morris (April 14)
“Memory in Language Models: Representation and Extraction”

Gopala Anumanchipalli (April 23)
“Speech”

We may have 0—2 more (Stay tuned!)

All details on the course website

Acknowledgement

The course material (lecture slides and assignments) are heavily based on:

- **Princeton COS 484 Natural Language Processing** by Danqi Chen, Tri Dao, Vikram Ramaswamy
- **CMU Advanced Natural Language Processing** by Graham Neubig & Sean Welleck
- **Stanford CS336 Language Modeling from Scratch** by Tatsumori Hashimoto & Percy Liang
- **Cornell LM-class** by Yoav Artzi

Lecture- and assignment-specific acknowledgement are provided in the corresponding slide deck & documents.

Questions?

Acknowledgement

Princeton COS 484 by Danqi Chen, Tri Dao, Vikram Ramaswamy
Cornell LM-class by Yoav Artzi
Stanford CS336 Language Modeling from Scratch by Tatsumori Hashimoto & Percy Liang