Course Introduction

Tanmoy Chakraborty
Associate Professor, IIT Delhi
https://tanmoychak.com/

Introduction to Large Language Models

Instructors

Teaching Assistants

Tanmoy Chakraborty
IIT Delhi

Soumen Chakrabarti
IIT Bombay

Anwoy Chatterjee
PhD student, IIT Delhi

Poulami Ghosh
PhD student, IIT Bombay

Course Content

- This is an **introductory graduate course** and we will be teaching the fundamental concepts underlying large language models.
- This course will start with a short introduction to NLP and Deep Learning, and then move on to the architectural intricacies of Transformers, followed by the recent advances in LLM research.

Course Content

Basics

- Introduction
- Intro to NLP
- Intro to Deep Learning
- Intro to Language Models (LMs)
- Word Embeddings (Word2Vec, GloVE)
- Neural LMs (CNN, RNN, Seq2Seq, Attention)

Course Content

Basics

- Introduction
- Intro to NLP
- Intro to Deep Learning
- Intro to Language Models (LMs)
- Word Embeddings (Word2Vec, GloVE)
- Neural LMs (CNN, RNN, Seq2Seq, Attention)

Architecture

- Intro to Transformer
- Positional encoding
- Tokenization strategies
- Decoder-only LM, Prefix LM, Decoding strategies
- Encoder-only LM, Encoder-decoder LM

Course Content

Basics

- Introduction
- Intro to NLP
- Intro to Deep Learning
- Intro to Language Models (LMs)
- Word Embeddings (Word2Vec, GloVE)
- Neural LMs (CNN, RNN, Seq2Seq, Attention)

Architecture

- Intro to Transformer
- Positional encoding
- Tokenization strategies
- Decoder-only LM, Prefix LM, Decoding strategies
- Encoder-only LM, Encoder-decoder LM

Learnability

- Instruction fine-tuning
- In-context learning
- Advanced prompting (Chain of Thoughts, Graph of Thoughts, Prompt Chaining, etc.)
- Alignment
- PEFT

Course Content

Basics
- Introduction
- Intro to NLP
- Intro to Deep Learning
- Intro to Language Models (LMs)
- Word Embeddings (Word2Vec, GloVE)
- Neural LMs (CNN, RNN, Seq2Seq, Attention)

Architecture
- Intro to Transformer
- Positional encoding
- Tokenization strategies
- Decoder-only LM, Prefix LM, Decoding strategies
- Encoder-only LM, Encoder-decoder LM

Learnability
- Instruction fine-tuning
- In-context learning
- Advanced prompting (Chain of Thoughts, Graph of Thoughts, Prompt Chaining, etc.)
- Alignment
- PEFT

Knowledge & Retrieval
- Knowledge graphs
- Open-book question answering
- Retrieval augmentation techniques

Course Content

Basics	Architecture	Learnability	Knowledge & Retrieval	Ethics and Misc.
• Introduction • Intro to NLP • Intro to Deep Learning • Intro to Language Models (LMs) • Word Embeddings (Word2Vec, GloVE) • Neural LMs (CNN, RNN, Seq2Seq, Attention)	• Intro to Transformer • Positional encoding • Tokenization strategies • Decoder-only LM, Prefix LM, Decoding strategies • Encoder-only LM, Encoder-decoder LM	• Instruction fine-tuning • In-context learning • Advanced prompting (Chain of Thoughts, Graph of Thoughts, Prompt Chaining, etc.) • Alignment • PEFT	• Knowledge graphs • Open-book question answering • Retrieval augmentation techniques	• Overview of recently popular models • Bias, toxicity and hallucination

Pre-Requisites

- Excitement about language!
- Willingness to learn

Pre-Requisites

- Excitement about language!
- Willingness to learn

Mandatory	Desirable
• Data Structures & Algorithms • Machine Learning • Python programming	• NLP • Deep learning

Pre-Requisites

- Excitement about language!
- Willingness to learn

Mandatory	Desirable
• Data Structures & Algorithms • Machine Learning • Python programming	• NLP • Deep learning

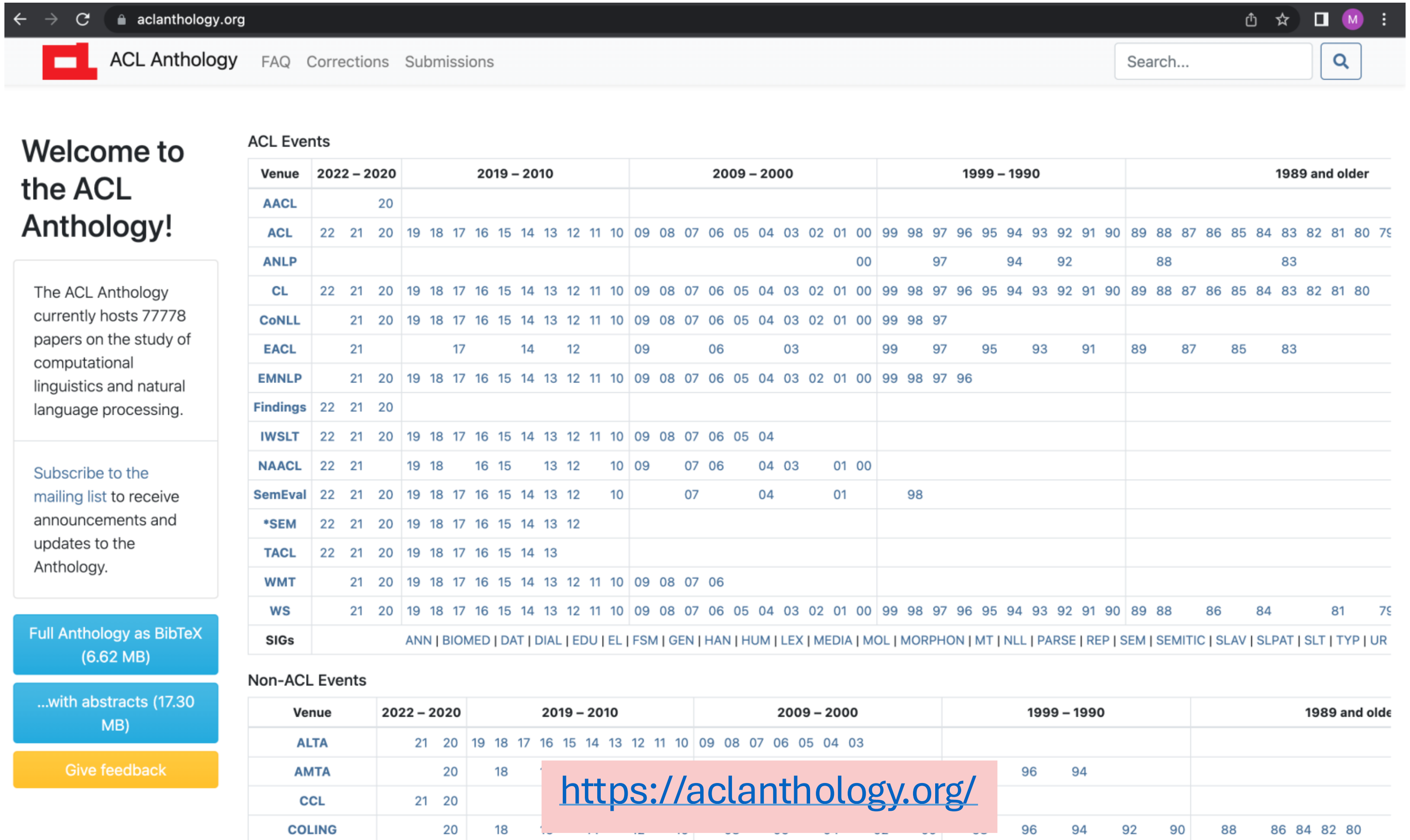
This course will NOT cover:

- Details of NLP, Machine Learning and Deep Learning
- Generative models for modalities other than text

Reading and Reference Materials

- Books (optional reading)
  - Speech and Language Processing, Dan Jurafsky and James H. Martin
    https://web.stanford.edu/~jurafsky/slp3/
  - Foundations of Statistical Natural Language Processing, Chris Manning and Hinrich Schütze
  - Natural Language Processing, Jacob Eisenstein
    https://github.com/jacobeisenstein/gt-nlp-class/blob/master/notes/eisenstein-nlp-notes.pdf
  - A Primer on Neural Network Models for Natural Language Processing, Yoav Goldberg
    http://u.cs.biu.ac.il/~yogo/nnlp.pdf
- Journals
  - Computational Linguistics, Natural Language Engineering, TACL, JMLR, TMLR, etc.
- Conferences
  - ACL, EMNLP, NAACL, COLING, ICML, NeurIPS, ICLR, AAAI, WWW, KDD, SIGIR, etc.

Research Papers Repository



aclanthology.org

ACL Anthology FAQ Corrections Submissions Search...

Welcome to the ACL Anthology!

The ACL Anthology currently hosts 77778 papers on the study of computational linguistics and natural language processing.

Subscribe to the mailing list to receive announcements and updates to the Anthology.

Full Anthology as BibTeX (6.62 MB)

...with abstracts (17.30 MB)

Give feedback

ACL Events

Venue	2022 – 2020	2019 – 2010	2009 – 2000	1999 – 1990	1989 and older
AACL	20				
ACL	22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01 00	99 98 97 96 95 94 93 92 91 90	89 88 87 86 85 84 83 82 81 80 79
ANLP			00	97 94 92	88 83
CL	22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01 00	99 98 97 96 95 94 93 92 91 90	89 88 87 86 85 84 83 82 81 80
CoNLL	21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01 00	99 98 97	
EACL	21	17 14 12	09 06 03	99 97 95 93 91	89 87 85 83
EMNLP	21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01 00	99 98 97 96	
Findings	22 21 20				
IWSLT	22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04		
NAACL	22 21	19 18 16 15 13 12 10	09 07 06 04 03 01 00		
SemEval	22 21 20	19 18 17 16 15 14 13 12 10	07 04 01	98	
*SEM	22 21 20	19 18 17 16 15 14 13 12			
TACL	22 21 20	19 18 17 16 15 14 13			
WMT	21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06		
WS	21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01 00	99 98 97 96 95 94 93 92 91 90	89 88 86 84 81 79
SIGs	ANN | BIOMED | DAT | DIAL | EDU | EL | FSM | GEN | HAN | HUM | LEX | MEDIA | MOL | MORPHON | MT | NLL | PARSE | REP | SEM | SEMITIC | SLAV | SLPAT | SLT | TYP | UR				

Non-ACL Events

Venue	2022 – 2020	2019 – 2010	2009 – 2000	1999 – 1990	1989 and older
ALTA	21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03		
AMTA	20	18 [illegible]	[illegible]	96 94	
CCL	21 20				
COLING	20	18 [illegible]	[illegible]	96 94 92 90	88 86 84 82 80

https://aclanthology.org/

Research Papers Repository

arXiv.org > cs > cs.CL

Computation and Language

Authors and titles for recent submissions

- Wed, 19 Aug 2020
- Tue, 18 Aug 2020
- Mon, 17 Aug 2020
- Fri, 14 Aug 2020
- Thu, 13 Aug 2020

[total of 84 entries: **1-25** | 26-50 | 51-75 | 76-84]
[showing 25 entries per page: fewer | more | all]

Wed, 19 Aug 2020

[1] **arXiv:2008.07905** [**pdf**, **other**]
Glancing Transformer for Non-Autoregressive Neural Machine Translation
Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, Lei Li
Comments: 11 pages, 3 figures, 4 tables
Subjects: **Computation and Language (cs.CL)**

[2] **arXiv:2008.07880** [**pdf**, **other**]
COVID-SEE: Scientific Evidence Explorer for COVID-19 Related Research
Karin Verspoor, Simon Šuster, Yulia Otmakhova, Shevon Mendis, Zenan Zhai, Biaoyan Fang, Jey Han Lau, Timothy Bal
Comments: COVID-SEE is available at this http URL
Subjects: **Computation and Language (cs.CL)**; Information Retrieval (cs.IR)

[3] **arXiv:2008.07772** [**pdf**, **other**]
Very Deep Transformers for Neural Machine Translation
Xiaodong Liu, Kevin Duh, Liyuan Liu, Jianfeng Gao
Comments: 6 pages, 3 figures and 3 tables
Subjects: **Computation and Language (cs.CL)**

[4] **arXiv:2008.07723** [**pdf**, **other**]
NASE: Learning Kr https://arxiv.org/list/cs.CL/recent **l Architecture Search**
Xiaoyu Kou, Bingfeng
Comments: Accepted by C
Subjects: **Computation and Language (cs.CL)**

Acknowledgements (Non-exhaustive List)

- Advanced NLP, Graham Neubig http://www.phontron.com/class/anlp2022/
- Advanced NLP, Mohit Iyyer https://people.cs.umass.edu/~miyyer/cs685/
- NLP with Deep Learning, Chris Manning, http://web.stanford.edu/class/cs224n/
- Understanding Large Language Models, Danqi Chen https://www.cs.princeton.edu/courses/archive/fall22/cos597G/
- Natural Language Processing, Greg Durrett https://www.cs.utexas.edu/~gdurrett/courses/online-course/materials.html
- Large Language Models: https://stanford-cs324.github.io/winter2022/
- Natural Language Processing at UMBC, https://laramartin.net/NLP-class/
- Computational Ethics in NLP, https://demo.clab.cs.cmu.edu/ethical_nlp/
- Self-supervised models, CS 601.471/671: Self-supervised Models (jhu.edu)
- WING.NUS Large Language Models, https://wing-nus.github.io/cs6101/
- And many more…

What is a Language Model (LM)?

Language Model gives the probability distribution over a sequence of tokens.

What is a Language Model (LM)?

Language Model gives the probability distribution over a sequence of tokens.

Language Model

Vocabulary

V = {arrived, delhi, have, is, monsoon, rains, the}

What is a Language Model (LM)?

Language Model gives the probability distribution over a sequence of tokens.

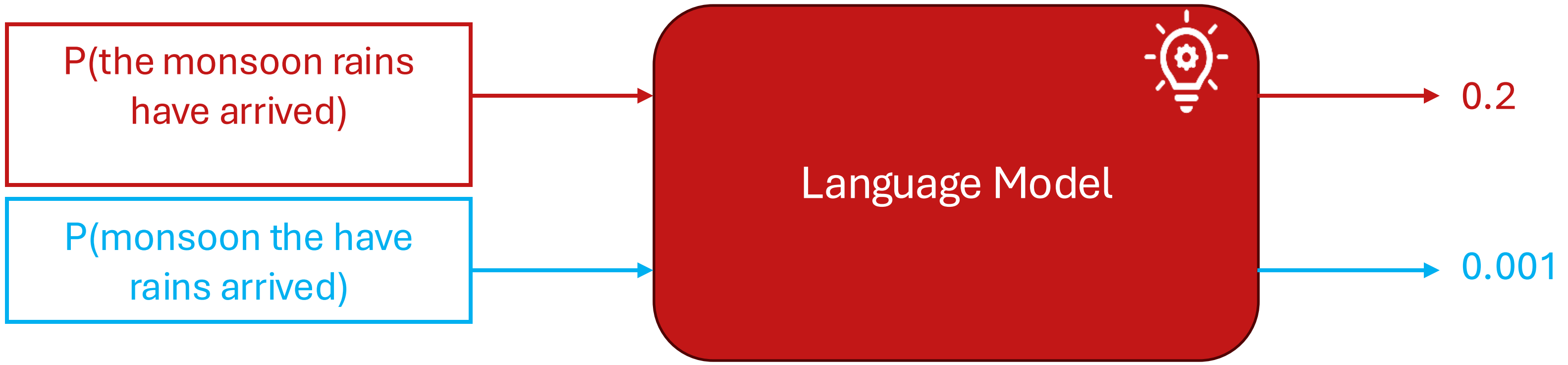
P(the monsoon rains have arrived) → Language Model → 0.2

Vocabulary

V = {arrived, delhi, have, is, monsoon, rains, the}

What is a Language Model (LM)?

Language Model gives the probability distribution over a sequence of tokens.



Vocabulary

V = {arrived, delhi, have, is, monsoon, rains, the}

LMs can 'Generate' Text !

- Consider a sequence of tokens $\{x_1, x_2, \ldots, x_L\}$,where $x_1, x_2, \ldots, x_L$ are in vocabulary V
- Notation: $P(x_1, x_2, \ldots, x_L) = P(x_{1:L})$
- Using the **chain rule of probability:**

$$P(x_{1:L}) = P(x_1).P(x_2|x_1).P(x_3|x_1, x_2) \ldots P(x_L|x_{L-1}) = \prod_{i=1}^{L} P(x_i|x_{1:i-1})$$

LMs can 'Generate' Text !

- Consider a sequence of tokens $\{x_1, x_2, \dots, x_L\}$,where $x_1, x_2, \dots, x_L$ are in vocabulary *V*
- Notation: $P(x_1, x_2, \dots, x_L) = P(x_{1:L})$
- Using the **chain rule of probability:**

$$P(x_{1:L}) = P(x_1).P(x_2|x_1).P(x_3|x_1, x_2) \dots P(x_L|x_{L-1}) = \prod_{i=1}^{L} P(x_i|x_{1:i-1})$$

Vocabulary

V = {arrived, delhi, have, is, monsoon, rains, the}

Given input 'the monsoon rains have' ,
LM can calculate
$P(x_i \mid \text{the monsoon rains have})$, $\forall\ x_i \in V$

LMs can 'Generate' Text !

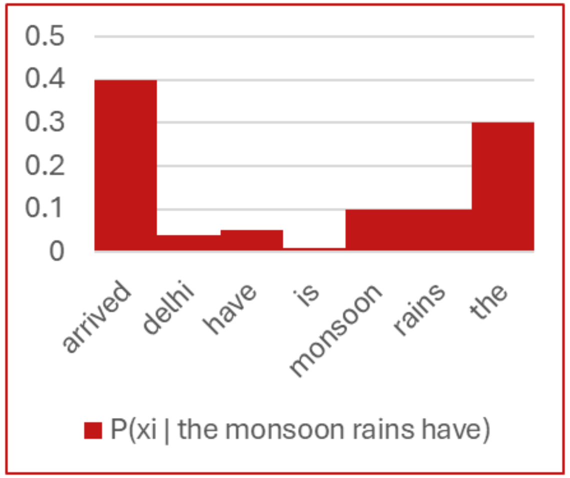
- Consider a sequence of tokens $\{x_1, x_2, \dots, x_L\}$,where $x_1, x_2, \dots, x_L$ are in vocabulary V
- Notation: $P(x_1, x_2, \dots, x_L) = P(x_{1:L})$
- Using the **chain rule of probability:**

$$P(x_{1:L}) = P(x_1).P(x_2|x_1).P(x_3|x_1,x_2)\dots P(x_L|x_{L-1}) = \prod_{i=1}^{L} P(x_i|x_{1:i-1})$$

Vocabulary

V = {arrived, delhi, have, is, monsoon, rains, the}

Given input 'the monsoon rains have' ,
LM can calculate
$P(x_i$ | the monsoon rains have) , $\forall\, x_i \in V$



LMs can 'Generate' Text !

- Consider a sequence of tokens $\{x_1, x_2, \dots, x_L\}$,where $x_1, x_2, \dots, x_L$ are in vocabulary V
- Notation: $P(x_1, x_2, \dots, x_L) = P(x_{1:L})$
- Using the **chain rule of probability:**

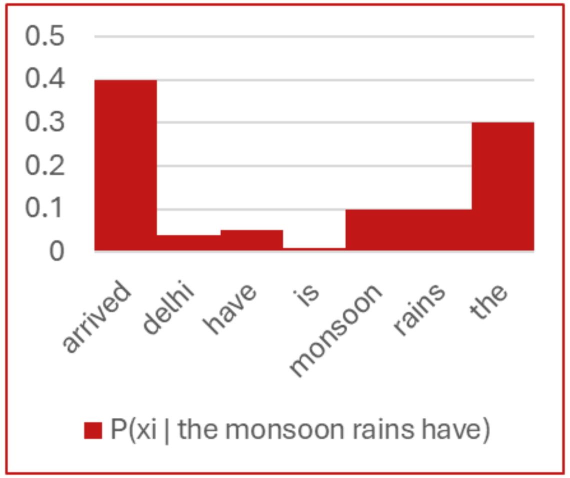
$$P(x_{1:L}) = P(x_1).P(x_2|x_1).P(x_3|x_1,x_2)\dots P(x_L|x_{L-1}) = \prod_{i=1}^{L} P(x_i|x_{1:i-1})$$

Vocabulary

V = {arrived, delhi, have, is, monsoon, rains, the}

Given input 'the monsoon rains have' ,
LM can calculate
$P(x_i \mid \text{the monsoon rains have})$, $\forall\, x_i \in V$

For generation, next token is sampled from this probability distribution



LMs can 'Generate' Text !

- Consider a sequence of tokens $\{x_1, x_2, \dots, x_L\}$,where $x_1, x_2, \dots, x_L$ are in vocabulary V
- Notation: $P(x_1, x_2, \dots, x_L) = P(x_{1:L})$
- Using the **chain rule of probability:**

$$P(x_{1:L}) = P(x_1).P(x_2|x_1).P(x_3|x_1,x_2)\dots P(x_L|x_{L-1}) = \prod_{i=1}^{L} P(x_i|x_{1:i-1})$$

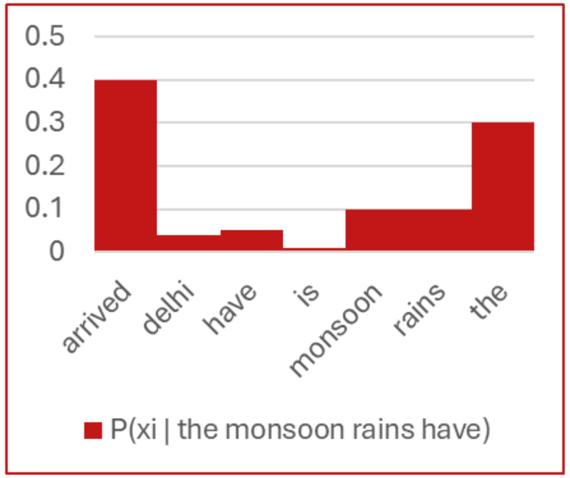
Vocabulary

V = {arrived, delhi, have, is, monsoon, rains, the}

Given input 'the monsoon rains have' ,
LM can calculate
P(x_i | the monsoon rains have) , ∀ x_i ϵ V

For generation, next token is sampled from this probability distribution

$$x_i \sim P(x_i \mid x_{1:i-1})$$



LMs can 'Generate' Text !

- Consider a sequence of tokens $\{x_1, x_2, \dots, x_L\}$,where $x_1, x_2, \dots, x_L$ are in vocabulary V
- Notation: $P(x_1, x_2, \dots, x_L) = P(x_{1:L})$
- Using the **chain rule of probability:**

$$P(x_{1:L}) = P(x_1).P(x_2|x_1).P(x_3|x_1, x_2) \dots P(x_L|x_{L-1}) = \prod_{i=1}^{L} P(x_i|x_{1:i-1})$$

Vocabulary

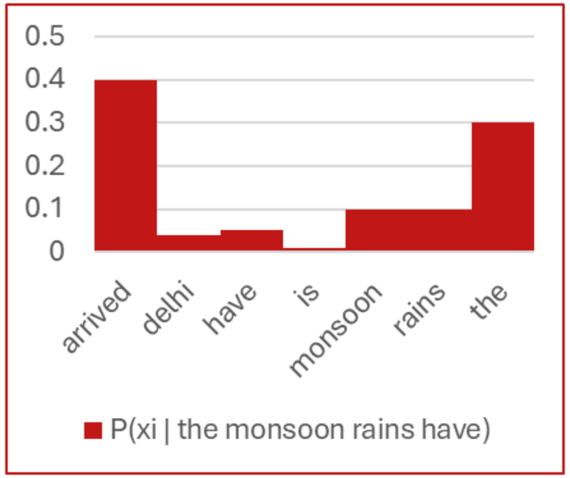
V = {arrived, delhi, have, is, monsoon, rains, the}

Given input 'the monsoon rains have' , LM can calculate $P(x_i \mid \text{the monsoon rains have})$, $\forall\, x_i \in V$

For generation, next token is sampled from this probability distribution

$x_i \sim P(x_i \mid x_{1:i-1})$

Auto-regressive LMs calculate this distribution efficiently, e.g. using 'Deep' Neural Networks



'Large' Language Models

The 'Large' in terms of model's size (# parameters) and massive size of training dataset.

Model	Organization	Date	Size (# params)
ELMo	AI2	Feb 2018	94,000,000
GPT	OpenAI	Jun 2018	110,000,000
BERT	Google	Oct 2018	340,000,000
XLM	Facebook	Jan 2019	655,000,000
GPT-2	OpenAI	Mar 2019	1,500,000,000
RoBERTa	Facebook	Jul 2019	355,000,000
Megatron-LM	NVIDIA	Sep 2019	8,300,000,000
T5	Google	Oct 2019	11,000,000,000
Turing-NLG	Microsoft	Feb 2020	17,000,000,000
GPT-3	OpenAI	May 2020	175,000,000,000
Megatron-Turing NLG	Microsoft, NVIDIA	Oct 2021	530,000,000,000
Gopher	DeepMind	Dec 2021	280,000,000,000

Model sizes have increased by an order of **5000x** over just the last 4 years !!!

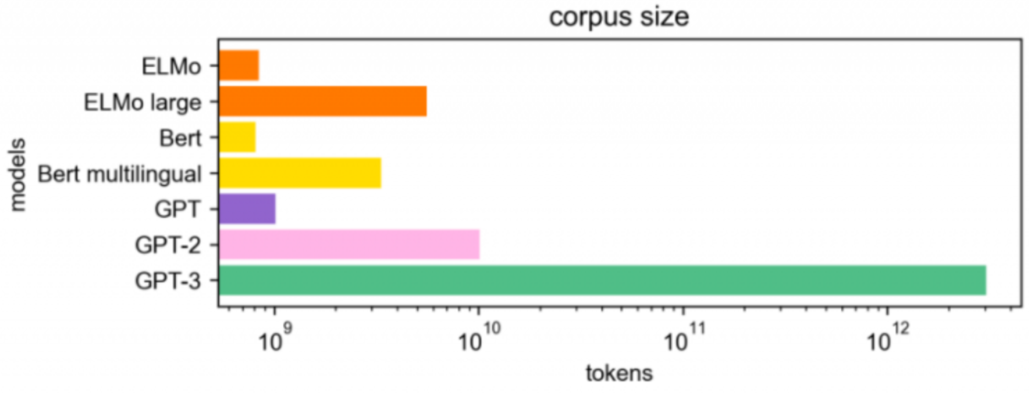


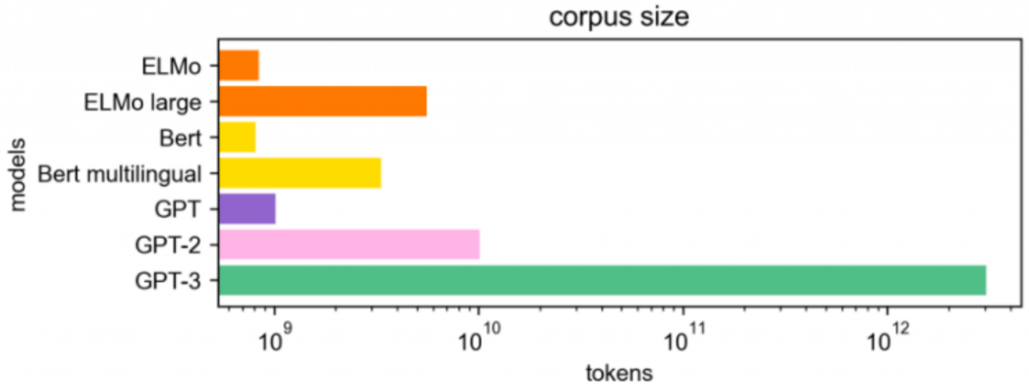
Image source: https://hellofuture.orange.com/en/the-gpt-3-language-model-revolution-or-evolution/

‘Large’ Language Models

The ‘Large’ in terms of model's size (# parameters) and massive size of training dataset.

Model	Organization	Date	Size (# params)
ELMo	AI2	Feb 2018	94,000,000
GPT	OpenAI	Jun 2018	110,000,000
BERT	Google	Oct 2018	340,000,000
XLM	Facebook	Jan 2019	655,000,000
GPT-2	OpenAI	Mar 2019	1,500,000,000
RoBERTa	Facebook	Jul 2019	355,000,000
Megatron-LM	NVIDIA	Sep 2019	8,300,000,000
T5	Google	Oct 2019	11,000,000,000
Turing-NLG	Microsoft	Feb 2020	17,000,000,000
GPT-3	OpenAI	May 2020	175,000,000,000
Megatron-Turing NLG	Microsoft, NVIDIA	Oct 2021	530,000,000,000
Gopher	DeepMind	Dec 2021	280,000,000,000

Model sizes have increased by an order of **5000x** over just the last 4 years !!!



Other recent models: PaLM (540B), OPT (175B), BLOOM (176B), Gemini-Ultra (1.56T), GPT-4 (1.76T)

Disclaimer: For API-based models like GPT-4/Gemini-Ultra, the number of parameters are not announced officially – these are rumored numbers as on the web

Image source: https://hellofuture.orange.com/en/the-gpt-3-language-model-revolution-or-evolution/

LLMs in AI Landscape

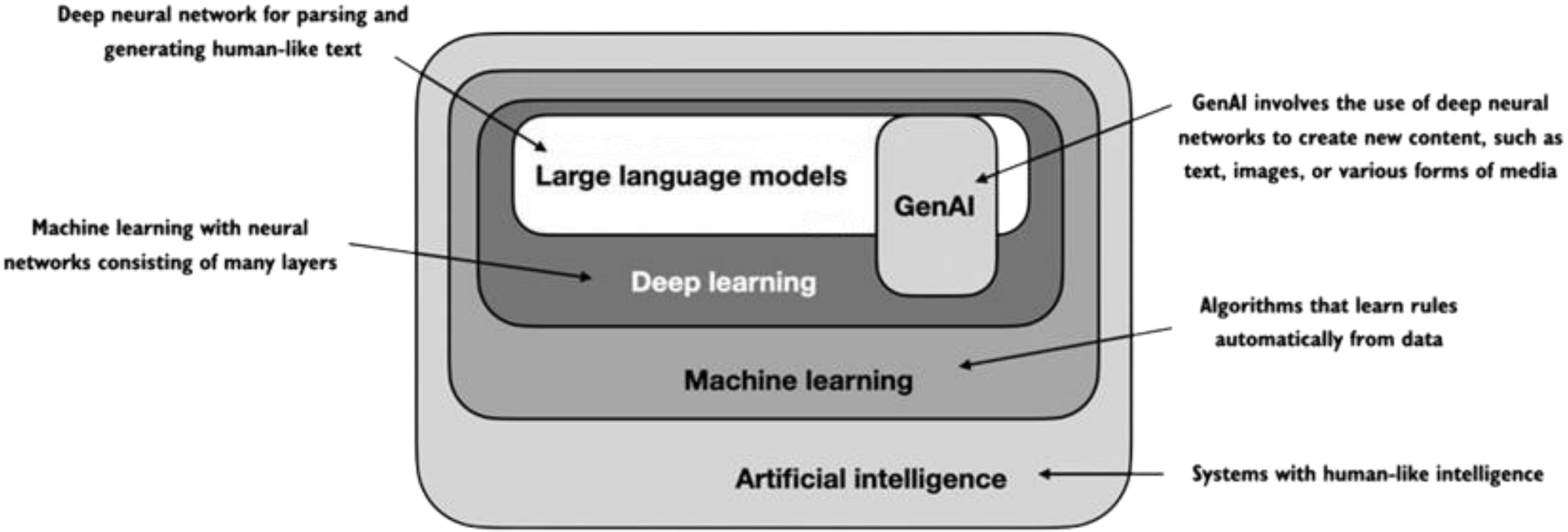


Image source: https://www.manning.com/books/build-a-large-language-model-from-scratch

Evolution of (L)LMs

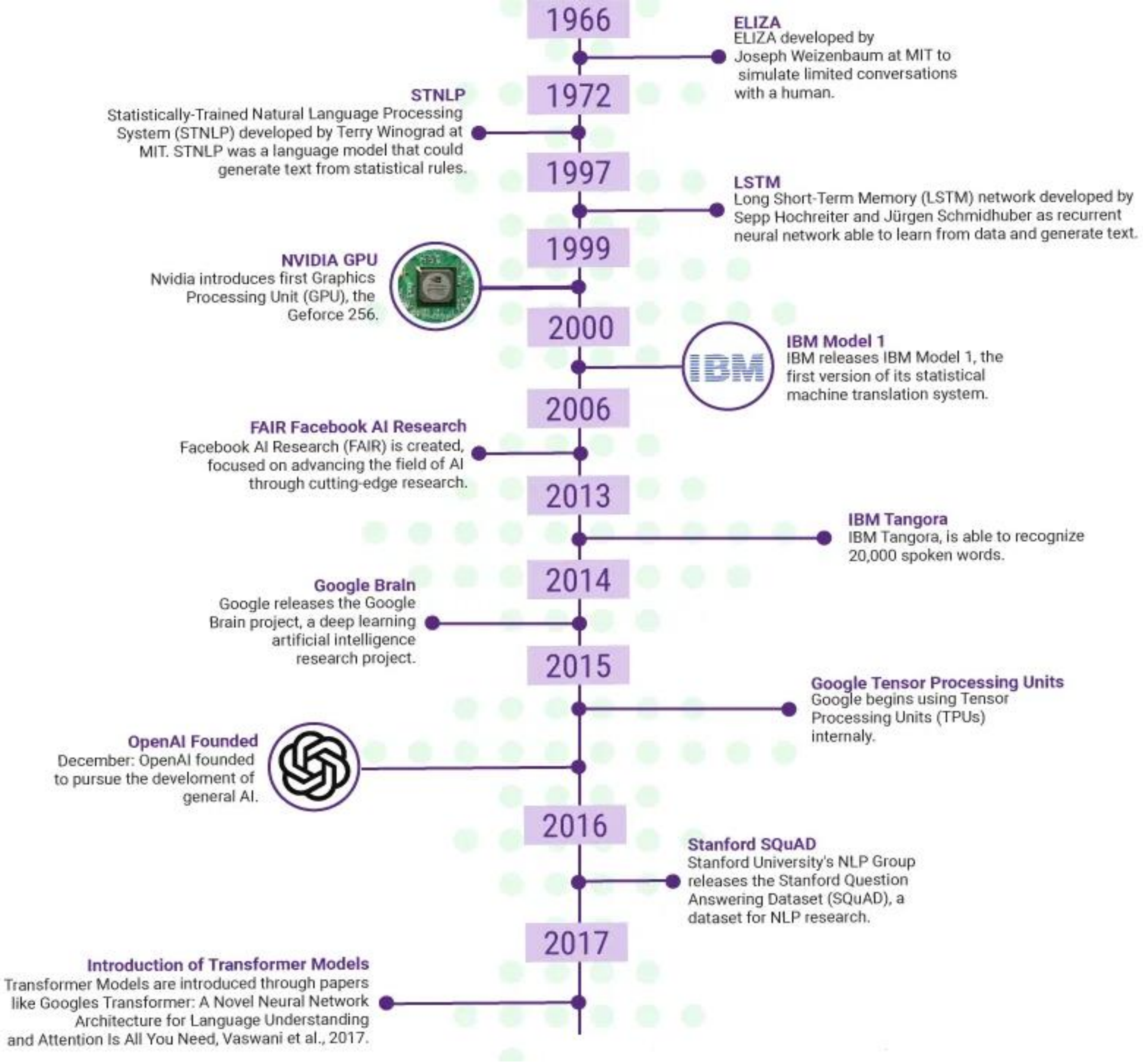


Image source: https://synthedia.substack.com/p/a-timeline-of-large-language-model

Post-Transformers Era

The LLM Race

Google Designed Transformers: But Could it Take Advantage?

Transformers (2017)

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Google Designed Transformers: But Could it Take Advantage?

Transformers (2017)

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

BERT (2018)

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Google Designed Transformers: But Could it Take Advantage?

Transformers (2017)

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

BERT (2018)

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

The beginning of use of Transformer as Language Representation Models.

BERT achieved SOTA on 11 NLP tasks.

Google Designed Transformers: But Could it Take Advantage?

Transformers (2017)

BERT (2018)

DistilBERT, TinyBERT, MobileBERT

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

The beginning of use of Transformer as Language Representation Models.

BERT achieved SOTA on 11 NLP tasks.

However, someone was waiting for the right opportunity!!

Guess Who?

However, someone was waiting for the right opportunity!!

OpenAI

OpenAI Started Pushing the Frontier

OpenAI

Improving Language Understanding by Generative Pre-Training

Alec Radford OpenAI alec@openai.com

Karthik Narasimhan OpenAI karthikn@openai.com

Tim Salimans OpenAI tim@openai.com

Ilya Sutskever OpenAI ilyasu@openai.com

OpenAI Started Pushing the Frontier

GPT (2018)

OpenAI

Improving Language Understanding by Generative Pre-Training

Alec Radford
OpenAI
alec@openai.com

Karthik Narasimhan
OpenAI
karthikn@openai.com

Tim Salimans
OpenAI
tim@openai.com

Ilya Sutskever
OpenAI
ilyasu@openai.com

OpenAI Started Pushing the Frontier

GPT (2018)

OpenAI

Improving Language Understanding by Generative Pre-Training

Alec Radford
OpenAI
alec@openai.com

Karthik Narasimhan
OpenAI
karthikn@openai.com

Tim Salimans
OpenAI
tim@openai.com

Ilya Sutskever
OpenAI
ilyasu@openai.com

- Use of **decoder-only architecture**
- The idea of generative pre-training over large corpus

The Beginning of Scale

GPT-2 (2019)

Language Models are Unsupervised Multitask Learners

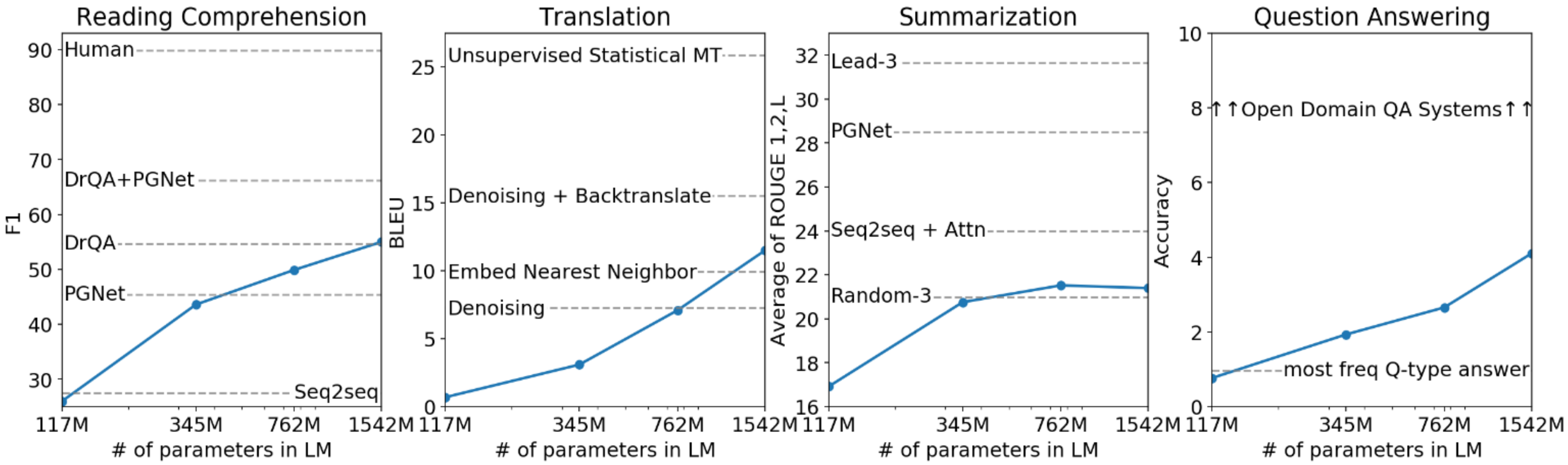
Alec Radford *[1] Jeffrey Wu *[1] Rewon Child[1] David Luan[1] Dario Amodei **[1] Ilya Sutskever **[1]

- GPT-1 (117 M) → GPT-2 (1.5 B) **13x increase in # parameters**
- Minimal changes (some LayerNorms added, modified weight initialization)
- Increase in context length: GPT-1 (512 tokens) → GPT-2 (1024 tokens)

The Beginning of Scale

GPT-2 (2019)

Performance boosts across tasks



What Was Google Developing Parallelly?

T5 (2019)

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel* CRAFFEL@GMAIL.COM
Noam Shazeer* NOAM@GOOGLE.COM
Adam Roberts* ADAROB@GOOGLE.COM
Katherine Lee* KATHERINELEE@GOOGLE.COM
Sharan Narang SHARANNARANG@GOOGLE.COM
Michael Matena MMATENA@GOOGLE.COM
Yanqi Zhou YANQIZ@GOOGLE.COM
Wei Li MWEILI@GOOGLE.COM
Peter J. Liu PETERJLIU@GOOGLE.COM

Google, Mountain View, CA 94043, USA

What Was Google Developing Parallelly?

T5 (2019)

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel* CRAFFEL@GMAIL.COM
Noam Shazeer* NOAM@GOOGLE.COM
.COM
.COM
.COM
.COM
.COM
.COM
.COM

Google, Mountain View, CA 94043, USA

- Similar broader goal of converting all text-based language problems into a text-to-text format.
- Used **Encoder-Decoder Architecture**.
- Pre-training strategy differs from GPT
    - Strategy more similar to BERT

Was It Only Google vs OpenAI?
Where did **Meta** Stand?

Was It Only Google vs OpenAI? Where did **Meta** Stand?

RoBERTa (2019)

RoBERTa: A Robustly Optimized BERT Pretraining Approach

Yinhan Liu[*§] **Myle Ott**[*§] **Naman Goyal**[*§] **Jingfei Du**[*§] **Mandar Joshi**[†]
Danqi Chen[§] **Omer Levy**[§] **Mike Lewis**[§] **Luke Zettlemoyer**[†§] **Veselin Stoyanov**[§]

[†] Paul G. Allen School of Computer Science & Engineering,
University of Washington, Seattle, WA
{mandar90,lsz}@cs.washington.edu

[§] Facebook AI
{yinhanliu,myleott,naman,jingfeidu,
danqi,omerlevy,mikelewis,lsz,ves}@fb.com

Was It Only Google vs OpenAI? Where did **Meta** Stand?

RoBERTa (2019)

RoBERTa: A Robustly Optimized BERT Pretraining Approach

Yinhan Li ... Iandar Joshi†
Danqi Chen§ ... Veselin Stoyanov§
ng,
fb.com

- Replication study of BERT pretraining
- Measured the impact of many key hyperparameters and training data size.
- **Found that BERT was significantly undertrained**, and can match or exceed the performance of every model published after it.

Was It Only Google vs OpenAI? Where did **Meta** Stand?

RoBERTa (2019)

RoBERTa: A Robustly Optimized BERT Pretraining Approach

Yinhan Li
Danqi Chen§

- Replication study of BERT pretraining
- Measured the impact of many key hyperparameters and training data size.
- **Found that BERT was significantly undertrained**, and can match or exceed the performance of every model published after it.

landar Joshi†
Veselin Stoyanov§

ng,

fb.com

XLM (2019)

Cross-lingual Language Model Pretraining

Guillaume Lample*
Facebook AI Research
Sorbonne Universités
glample@fb.com

Alexis Conneau*
Facebook AI Research
Université Le Mans
aconneau@fb.com

Was It Only Google vs OpenAI? Where did **Meta** Stand?

RoBERTa (2019)

RoBERTa: A Robustly Optimized BERT Pretraining Approach

Yinhan Li... Mandar Joshi[†]
Danqi Chen[§] ... Veselin Stoyanov[§]

- Replication study of BERT pretraining
- Measured the impact of many key hyperparameters and training data size.
- **Found that BERT was significantly undertrained**, and can match or exceed the performance of every model published after it.

XLM (2019)

Cross-lingual Language Model Pretraining

Alexis Conneau*
...ebook AI Research
...niversité Le Mans
...onneau@fb.com

- Proposed methods to learn **cross-lingual language models (XLMs)**
- Obtained SOTA on:
  - cross-lingual classification
  - unsupervised and supervised machine translation

OpenAI Continues to Scale

GPT-3 (2020)

Language Models are Few-Shot Learners

Tom B. Brown* **Benjamin Mann*** **Nick Ryder*** **Melanie Subbiah***

Jared Kaplan† **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam** **Girish Sastry**

Amanda Askell **Sandhini Agarwal** **Ariel Herbert-Voss** **Gretchen Krueger** **Tom Henighan**

Rewon Child **Aditya Ramesh** **Daniel M. Ziegler** **Jeffrey Wu** **Clemens Winter**

Christopher Hesse **Mark Chen** **Eric Sigler** **Mateusz Litwin** **Scott Gray**

Benjamin Chess **Jack Clark** **Christopher Berner**

Sam McCandlish **Alec Radford** **Ilya Sutskever** **Dario Amodei**

OpenAI

OpenAI Continues to Scale

GPT-3 (2020)

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafu ranav Shyam Girish Sastry

Amanda Askell Sandh hen Krueger Tom Henighan

Rewon Child Adit frey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

175 B parameters !

OpenAI Continues to Scale

GPT-3 (2020)

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafu ranav Shyam Girish Sastry

Amanda Askell Sandh hen Krueger Tom Henighan

Rewon Child Adit rey Wu Clemens Winter

Christopher H win Scott Gray

Benja pher Berner

Sam McCa Dario Amodei

OpenAI

175 B parameters !

OpenAI stops open-sourcing!!

Google Starts Scaling too (But is it Late) !

PaLM (2022)

PaLM: Scaling Language Modeling with Pathways

Aakanksha Chowdhery* Sharan Narang* Jacob Devlin*
Maarten Bosma Gaurav Mishra Adam Roberts Paul Barham
Hyung Won Chung Ch … rker Schuh Kensen Shi
Sasha Tsvyashchenk … ker Barnes Yi Tay
Noam Shazeer‡ Vino … Du Ben Hutchinson
Reiner Pope Jan … ard Guy Gur-Ari
Pengcheng Yin To … mawat Sunipa Dev
Henryk Michalewski Xavier Garcia Vedant Misra Kevin Robinson Liam Fedus
Denny Zhou Daphne Ippolito David Luan‡ Hyeontaek Lim Barret Zoph
Alexander Spiridonov Ryan Sepassi David Dohan Shivani Agrawal Mark Omernick
Andrew M. Dai Thanumalayan Sankaranarayana Pillai Marie Pellat Aitor Lewkowycz
Erica Moreira Rewon Child Oleksandr Polozov† Katherine Lee Zongwei Zhou
Xuezhi Wang Brennan Saeta Mark Diaz Orhan Firat Michele Catasta† Jason Wei
Kathy Meier-Hellstern Douglas Eck Jeff Dean Slav Petrov Noah Fiedel

Google Research

540 B parameters !

Google Starts Scaling too (But is it Late) !

PaLM (2022)

PaLM: Scaling Language Modeling with Pathways

Aakanksha Chowdhery* Sharan Narang* Jacob Devlin*
Maarten Bosma Gaurav Mishra Adam Roberts Paul Barham
Hyung Won Chung Ch rker Schuh Kensen Shi
Sasha Tsvyashchenk ker Barnes Yi Tay
Noam Shazeer‡ Vino Du Ben Hutchinson
Reiner Pope Jan ard Guy Gur-Ari
Pengcheng Yin To mawat Sunipa Dev
Henryk Michalewski Xavier Garcia Vedant Misra Kevin Robinson Liam Fedus
Denny Zho Barret Zoph
Alexander Spirid Mark Omernick
Andrew M. Dai Aitor Lewkowycz
Erica Moreira Zongwei Zhou
Xuezhi Wang E tasta† Jason Wei
Kathy Mei Noah Fiedel

Google Research

540 B parameters !

Google follows OpenAI in stopping open-sourcing !

It's now the "LLM Race"

2021-2022: A Flurry of LLMs

Microsoft
NVIDIA
Megatron-Turing NLG

Gopher

DeepMind
Chinchilla AI

LaMDA

CODEGEN

OpenAI
Codex

Google AI
MINERVA
MODEL FOR
QUANTITATIVE REASONING
PROBLEMS SOLVING

Meta Promotes Open-sourcing !

Meta Promotes Open-sourcing !

OPT (2022)

OPT: Open Pre-trained Transformer Language Models

Susan Zhang*, Stephen Roller*, Naman Goyal*,
Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li,
Xi Victoria Lin, Todor Mihaylov, Myle Ott†, Sam Shleifer†, Kurt Shuster, Daniel Simig,
Punit Singh Koura, Anjali Sridhar, Tianlu Wang, Luke Zettlemoyer

Meta AI

{susanz,roller,naman}@fb.com

Meta Promotes Open-sourcing !

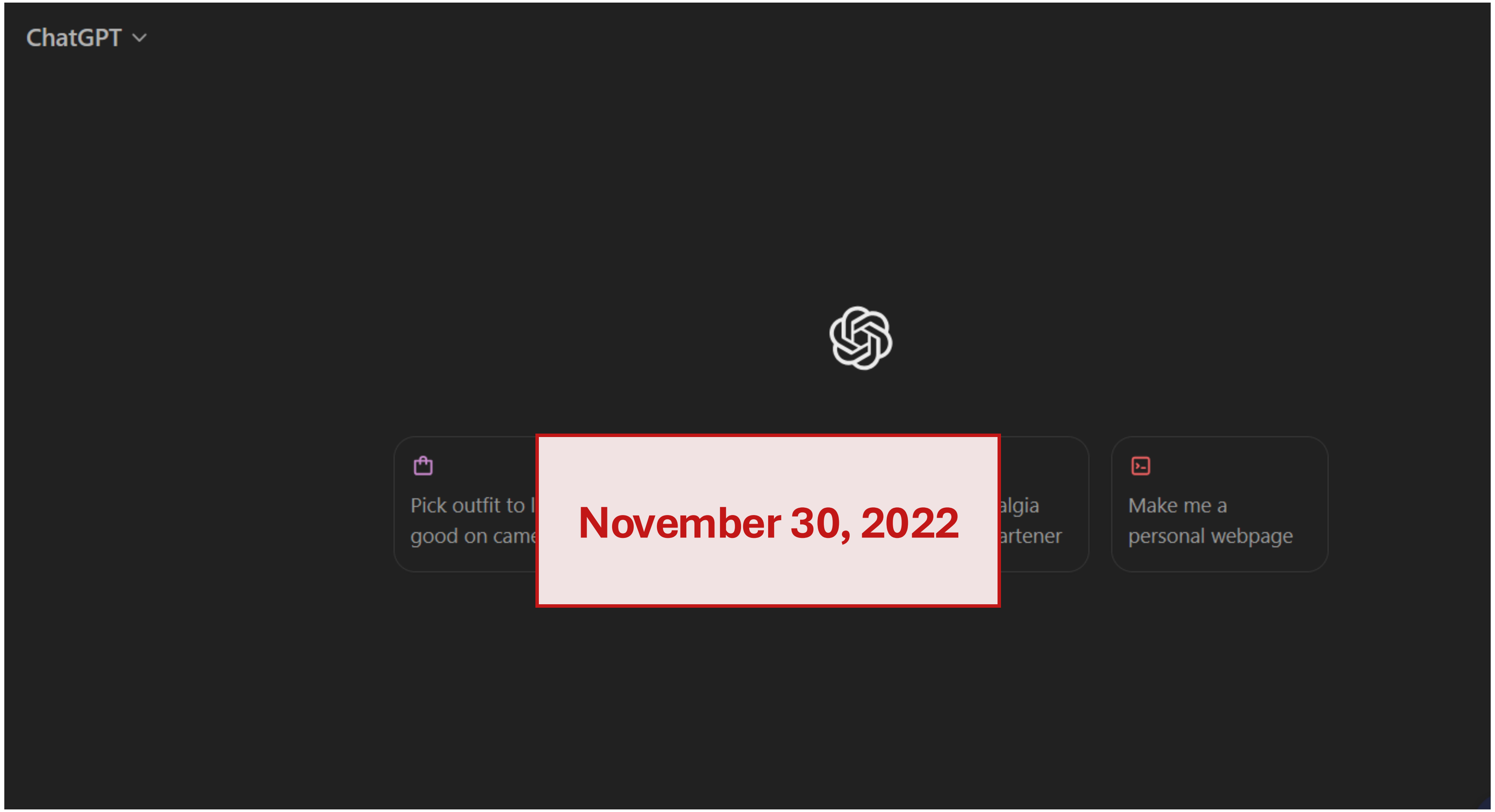
OPT (2022)

OPT: Open Pre-trained Transformer Language Models

Susan Zhang*, Stephen Roller*, Naman Goyal*,
Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li,
Xi Victoria Lin, T… …t Shuster, Daniel Simig,
Punit Sin… …ke Zettlemoyer

- A suite of decoder-only pre-trained transformers ranging from 125M to 175B parameters
- **Open-sourced !!!**

The ChatGPT Moment



2023: The Year of Rapid Pace

Feb, 2023: **Google** releases **Bard**

Feb, 2023: **Meta** releases its **LLaMA** family of **open-source models**

March, 2023: **Anthropic**, a start-up founded in 2021 by ex-OpenAI researchers, releases **Claude**

March, 2023: **OpenAI** releases **GPT-4**, which is multimodal

June, 2023: **Microsoft** releases **Phi-1**, a 1.3B LLM for code

Sept, 2023: **Mistral AI** releases **Mistral-7B** model

Nov, 2023: **xAI** releases **Grok**

Dec, 2023: **Google** releases **Gemini**

Gemma 2

GPT-4o

Llama 3
From Meta

MISTRAL AI_
Codestral
Mamba

And now in 2024 seeing even more rapid advancements !

Phi - 3
From Microsoft

405B
Meet Llama 3.1
70B
8B

MISTRAL
LARGE 2

Why Does This Course Exist?

Why Does This Course Exist?

Why do we need a separate course on LLMs? What changes with the scale of LMs?

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

Why do we need a separate course on LLMs? What changes with the scale of LMs?

Emergence

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

Why do we need a separate course on LLMs? What changes with the scale of LMs?

Emergence

Although the technical machineries are almost similar, 'just scaling up' these models results in new **emergent** behaviors, which lead to significantly different capabilities and societal impacts.

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

LLMs show emergent capabilities, not observed previously in ‘small’ LMs.

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

LLMs show emergent capabilities, not observed previously in ‘small’ LMs.

- In-context learning: A **pre-trained language model** can be guided with **only prompts to perform different tasks (without separate task-specific fine-tuning)**.
  - In-context learning is an example of **emergent** behavior.

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

LLMs show emergent capabilities, not observed previously in ‘small’ LMs.

- In-context learning: A **pre-trained language model** can be guided with **only prompts to perform different tasks (without separate task-specific fine-tuning)**.
    - In-context learning is an example of **emergent** behavior.

LLMs are widely adopted in real-world.

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

LLMs show emergent capabilities, not observed previously in 'small' LMs.

- In-context learning: A **pre-trained language model** can be guided with **only prompts to perform different tasks (without separate task-specific fine-tuning)**.
    - In-context learning is an example of **emergent** behavior.

LLMs are widely adopted in real-world.

- **Research**: LLMs have transformed **NLP research** world, achieving state-of-the-art performance across a wide range of tasks such as sentiment classification, question answering, summarization, and machine translation.

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

LLMs show emergent capabilities, not observed previously in ‘small’ LMs.

- In-context learning: A **pre-trained language model** can be guided with **only prompts to perform different tasks (without separate task-specific fine-tuning)**.
  - In-context learning is an example of **emergent** behavior.

LLMs are widely adopted in real-world.

- **Research**: LLMs have transformed **NLP research** world, achieving state-of-the-art performance across a wide range of tasks such as sentiment classification, question answering, summarization, and machine translation.
- **Industry**: Here is a very incomplete list of some high profile large language models that are being used in **production systems**:
  - Google Search (BERT)
  - Facebook content moderation (XLM)
  - Microsoft’s Azure OpenAI Service (GPT-3/3.5/4)

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

With tremendous capabilities, LLMs' usage also carries various **risks**.

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

With tremendous capabilities, LLMs' usage also carries various **risks**.

- **Reliability & Disinformation:** LLMs often hallucinate – generate responses that *seem correct*, but are not factually correct.
  - Significant challenge for high-stakes applications like healthcare

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

With tremendous capabilities, LLMs' usage also carries various **risks**.

- **Reliability & Disinformation:** LLMs often hallucinate – generate responses that *seem correct*, but are not factually correct.
  - Significant challenge for high-stakes applications like healthcare
- **Social bias**: Most LLMs show performance disparities across demographic groups, and their predictions can enforce stereotypes.
  - P(**He** is a doctor) > P(**She** is a doctor.)
  - Training data contains inherent bias

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

With tremendous capabilities, LLMs' usage also carries various **risks**.

- **Reliability & Disinformation:** LLMs often hallucinate – generate responses that *seem correct*, but are not factually correct.
  - Significant challenge for high-stakes applications like healthcare
- **Social bias**: Most LLMs show performance disparities across demographic groups, and their predictions can enforce stereotypes.
  - P(**He** is a doctor) > P(**She** is a doctor.)
  - Training data contains inherent bias
- **Toxicity**: LLMs can generate toxic/hateful content.
  - Trained on a huge amount of Internet data (e.g., Reddit), which inevitably contains offensive content
  - Challenge for applications such as writing assistants or chatbots

Content credits: https://stanford-cs324.github.io/winter2022/

Why Does This Course Exist?

With tremendous capabilities, LLMs' usage also carries various **risks**.

- **Reliability & Disinformation:** LLMs often hallucinate – generate responses that *seem correct*, but are not factually correct.
  - Significant challenge for high-stakes applications like healthcare
- **Social bias**: Most LLMs show performance disparities across demographic groups, and their predictions can enforce stereotypes.
  - P(**He** is a doctor) > P(**She** is a doctor.)
  - Training data contains inherent bias
- **Toxicity**: LLMs can generate toxic/hateful content.
  - Trained on a huge amount of Internet data (e.g., Reddit), which inevitably contains offensive content
  - Challenge for applications such as writing assistants or chatbots
- **Security**: LLMs are trained on a scrape of the public Internet - anyone can put up a website that can enter the training data.
  - An attacker can perform a **data poisoning** attack.

Content credits: https://stanford-cs324.github.io/winter2022/

We Will Cover Almost All of These in 5 Modules

Module-1: Basics

- A **refresher on the basics of NLP** required to understand and appreciate LLMs.
- A brief **introduction to the basics of Deep Learning**.
- The basics of **Statistical Language Modelling**.
- How did we end up in **Neural NLP**?
  - We will discuss the transition and the foundations of Neural NLP.
- Initial **Neural LMs**

Intro to NLP

Intro to Deep Learning

Intro to Language Models (LMs)

Word Embeddings (Word2Vec, GloVE)

Neural LMs (CNN, RNN, Seq2Seq, Attention)

We Will Cover Almost All of These in 5 Modules

- Module-2: Architecture
  - Workings of **Vanilla Transformers**
  - **Positional encoding** and **Tokenization strategies**
  - Different **Transformer Variants**
    - How do their training strategies differ? How are Masked LMs (like, BERT) different from Auto-regressive LMs (like, GPT)?
  - **Response generation (Decoding) strategies**

Intro to Transformer

Positional encoding

Tokenization strategies

Decoder-only LM, Prefix LM, Decoding strategies

Encoder-only LM, Encoder-decoder LM

We Will Cover Almost All of These in 5 Modules

- Module-3: Learnability
  - What makes modern LLMs so good in following user instructions?
  - What is **In-context Learning**? What are its various facets?
  - What kind of prompting techniques are required to elicit reasoning in LLMs?
  - How are LLMs made to **generate responses preferred by humans**?
    - Does it remove toxicity in responses?
  - **Efficiency** is crucial in production systems.
    - How are LLMs efficiently fine-tuned?

Instruction fine-tuning	In-context learning
Advanced Prompting	Alignment
PEFT	

We Will Cover Almost All of These in 5 Modules

- Module-4: Knowledge and Retrieval
  - **Knowledge graphs (KGs)**
    - Representation, completion
    - Tasks: Alignment and isomorphism
    - Distinction between graph neural networks and neural KG inference
  - **Open-book question answering**: retrieving from structured and unstructured sources
  - **Retrieval augmentation techniques**
    - Key-value memory networks in QA for simple paths in KGs
    - Early HotPotQA solvers, pointer networks, reading comprehension
    - REALM, RAG, FiD, Unlimiformer
    - KGQA (e.g., EmbedKGQA, GrailQA)

Knowledge graphs

Open-book question answering

Retrieval augmentation techniques

We Will Cover Almost All of These in 5 Modules

- Module-5: Ethics and Miscellaneous
  - A discussion on **ethical issues** and **risks** of LLM usage
  - An overview of the recent popular LLMs, like GPT4, Llama 3, Claude 3, Mistral, and Gemini.

Bias, toxicity and hallucination

Overview of the recent popular LLMs

Suggestions (For Effective Learning)

- To understand the concepts clearly, experiment with the models (**Hugging Face** makes life easier).
- Smaller models (like, GPT2) can be run on **Google Colab** / **Kaggle**.
  - Even 7B models can be run with proper quantization.

Hugging Face

kaggle

Always **get your hands dirty !**

LLM Research is all about implementing and experimenting with your ideas.

Suggestions (For Effective Learning)

- To understand the concepts clearly, experiment with the models (**Hugging Face** makes life easier).
- Smaller models (like, GPT2) can be run on **Google Colab** / **Kaggle**.
  - Even 7B models can be run with proper quantization.

Hugging Face

kaggle

Rule of thumb:

Never believe in any hypothesis until your experiments verify it !