Applied Machine Learning for Business Analytics

Lecture 4: LLM Fundamentals

Lecturer: Zhao Rui

How to read model-card from Huggingface

Model distilled from DeepSeek-R1 based on Qwen

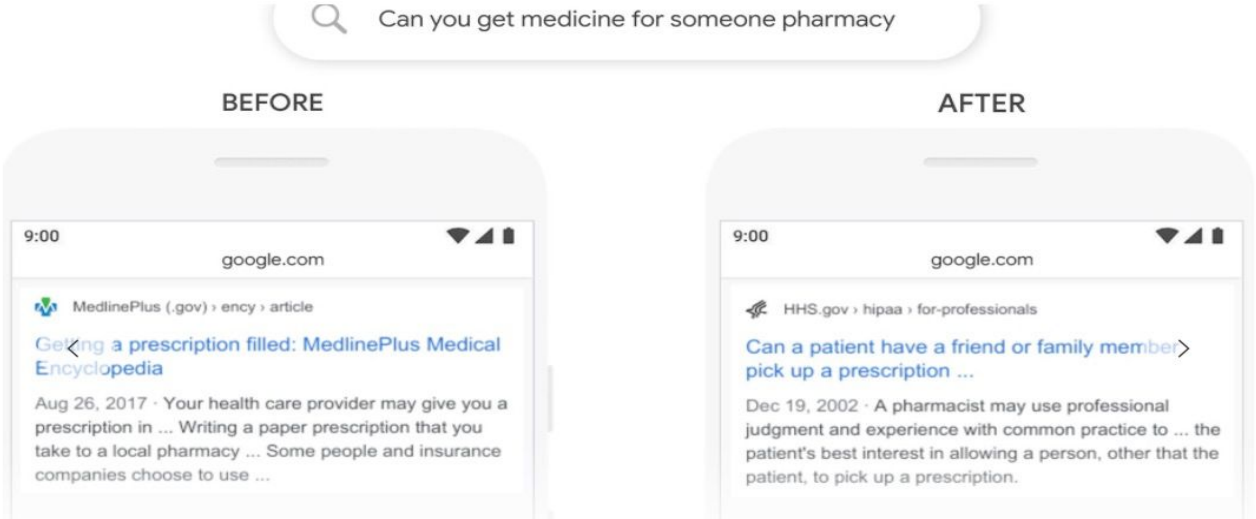
Fri 02/06	LLM Fundamentals	Link	Assignment I Out	How transformers and LLMs work
Fri 02/13	Training & Scaling LLMs	Link	Assignment II Due	How LLMs are built and improved
Fri 02/20	Inference & Reasoning	Link	Assignment II Out	How to get the best outputs
Fri 02/27	Recess Week	N.A.	Proposal & Assignment II Due	
Fri 03/06	RAG & Context Management	Link	Assignment III Out	How to ground LLMs in external knowledge
Fri 03/13	Agent Design Patterns	Link	Assignment III Due	How to build autonomous systems
Fri 03/20	Agent Production&Security	Link	Kaggle Starts	How to deploy agents safely with guardrails

Agenda

1. BERT
2. LLM Basics
3. GPTs

1. BERT

BERT in Google Search

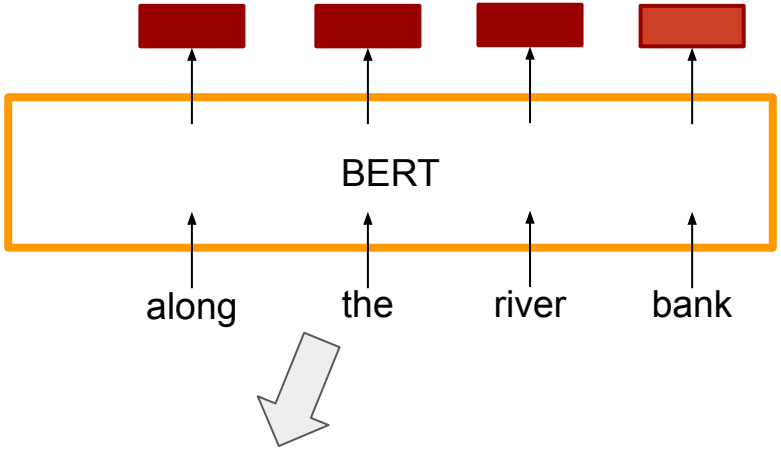


https://blog.google/products/search/search-language-understanding-bert/

With the latest advancements from our research team in the science of language understanding--made possible by machine learning--we're making a significant improvement to how we understand queries, representing the biggest leap forward in the past five years, and one of the biggest leaps forward in the history of Search.

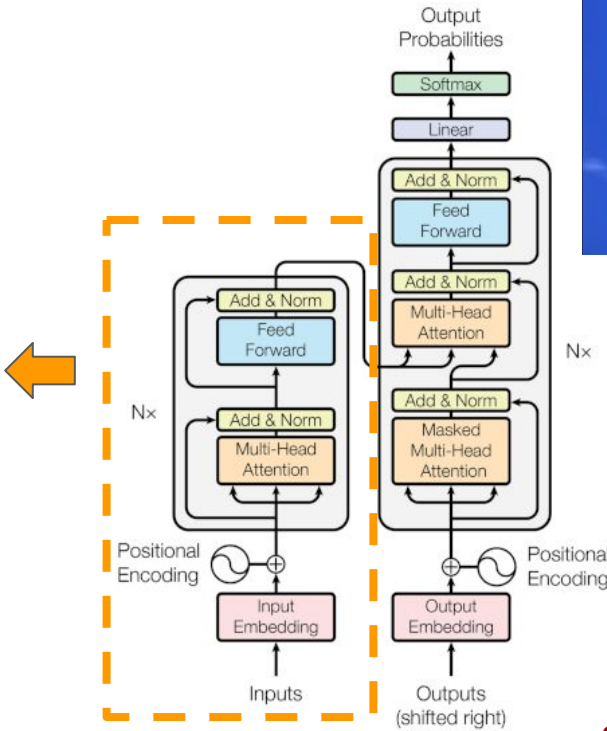
What is BERT

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (**BERT**)
- BERT: Encoder of Transformer,



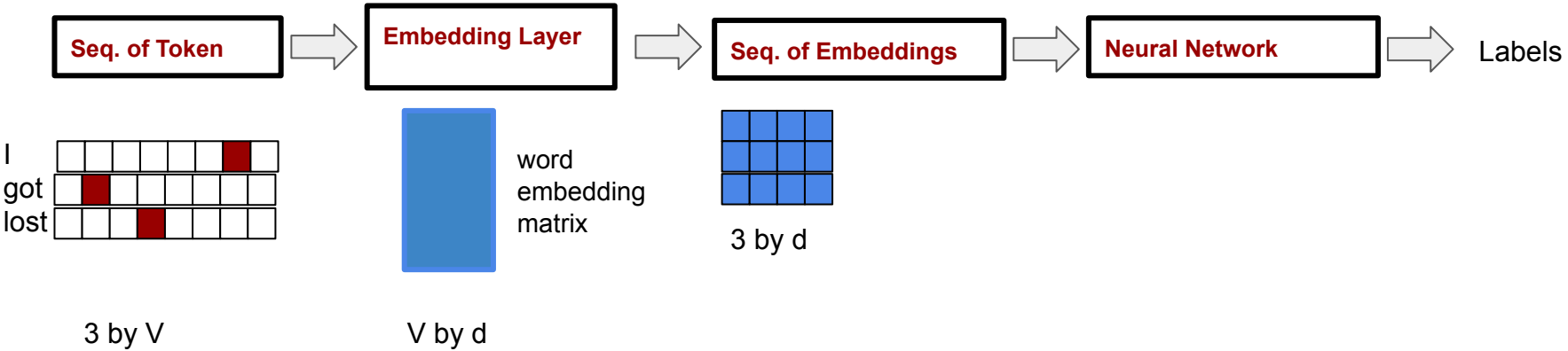
Given a sequence of words, generate a sequence of vectors and then can be used for various NLP tasks

Transformer



Solve Seq2Seq Task

Neural Networks for NLP



Each word has a fixed vector

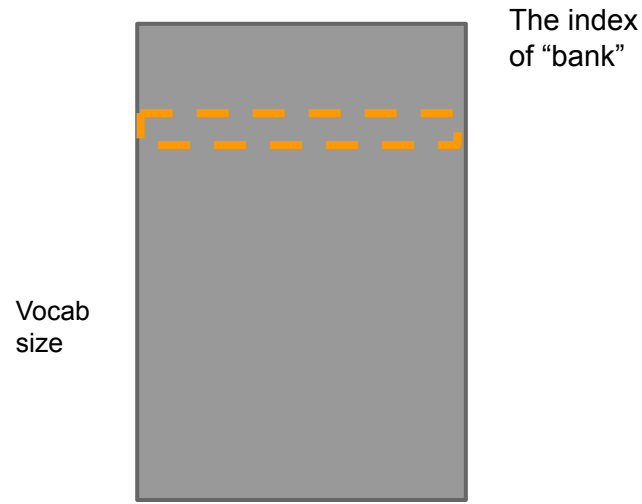
Can not address multi-sense problem!

Multiple Senses of Words

- It is safest to deposit your money in the bank.
- All the animals lined up along the river bank.
- Today, blood banks collect blood.

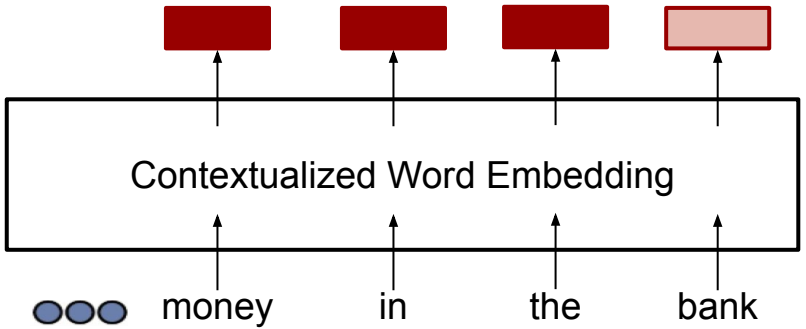
The third sense of not?

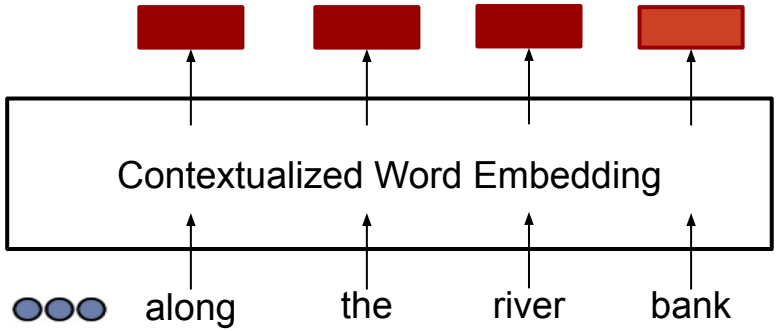
Word2Vec, Fasttext,Glove and other word embedding models

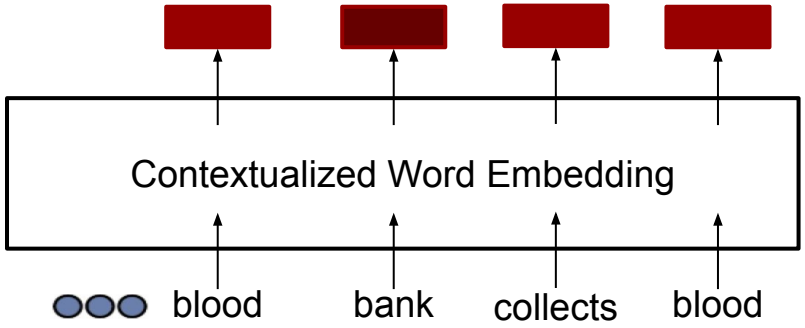


Contextualized Word Embeddings

Encoded Embeddings







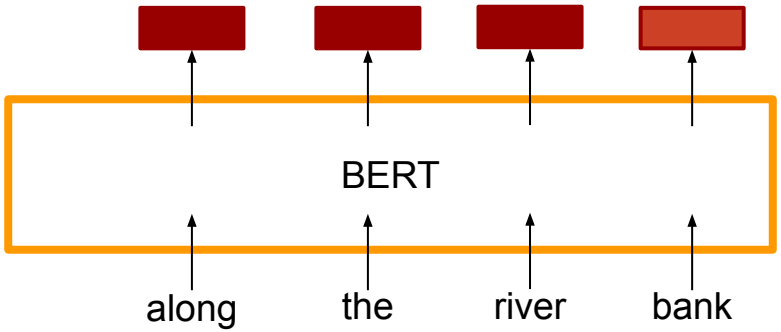
Different Contexts,
Different Encoded Embeddings for bank.

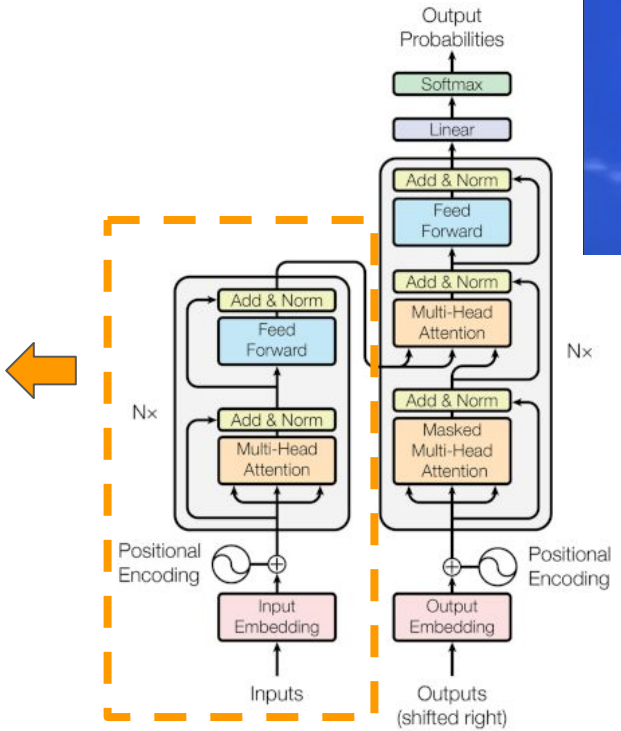
Embeddings generated from BERT

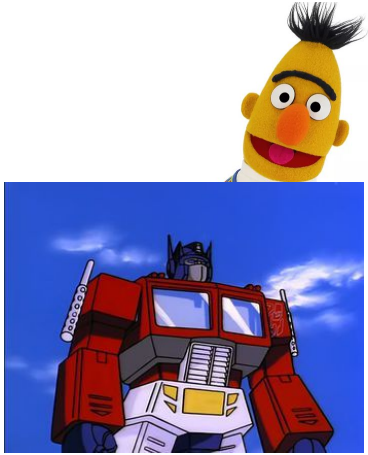
Cos-similarities among vectors of “apple” in different context

BERT

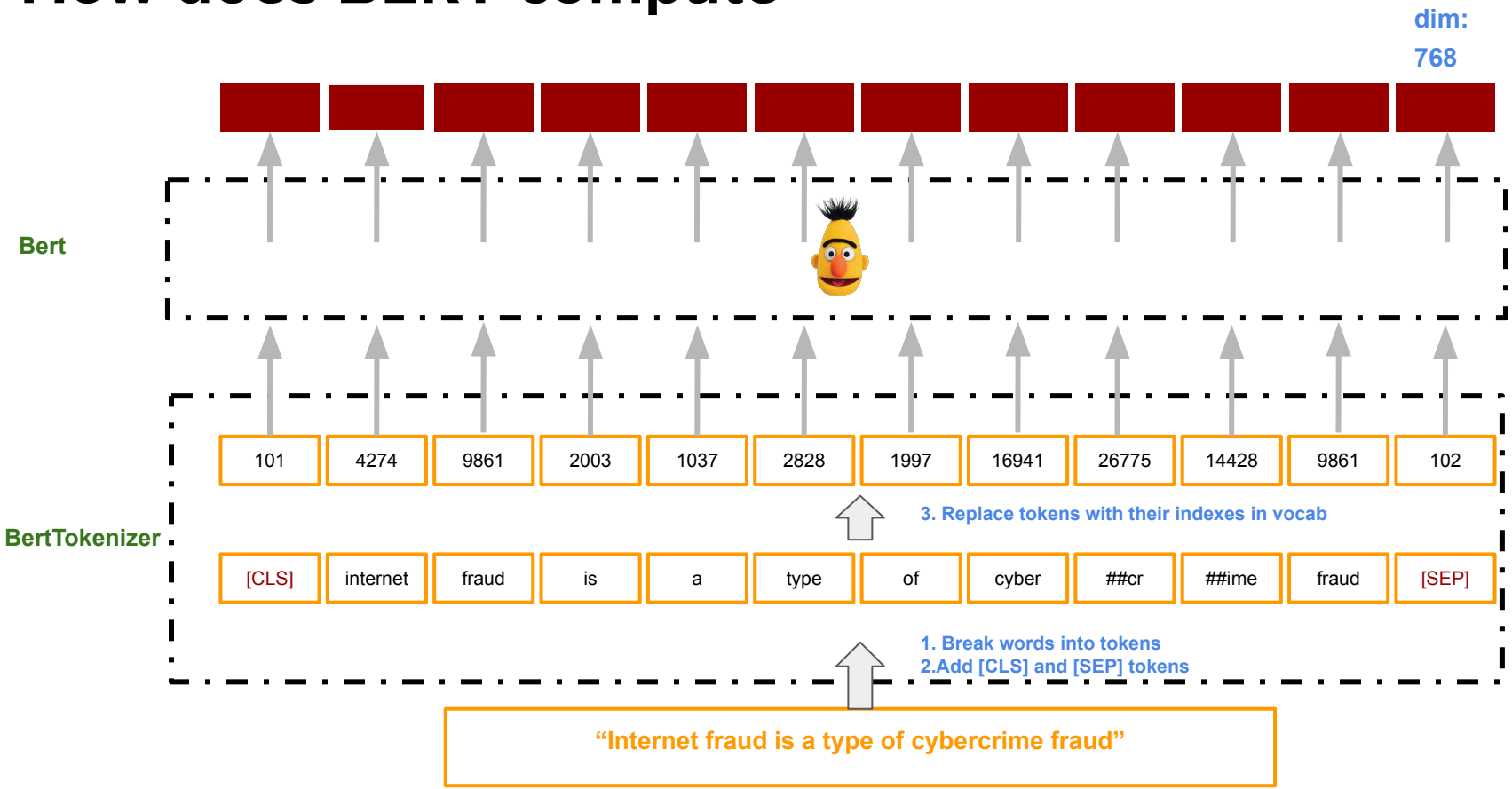
- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (**BERT**)
- BERT: Encoder of Transformer,







How does BERT compute



How to use BERT

A model that can “read” text

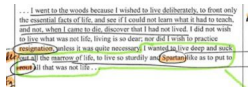
Text without annotation

Pre-train

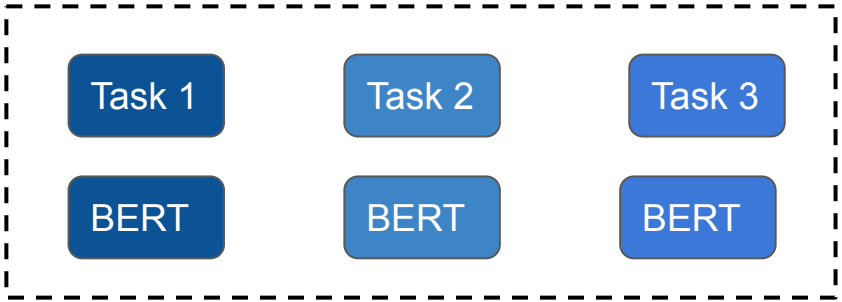
BERT

Learned from a large amount of **text without annotation**

Fine-tune

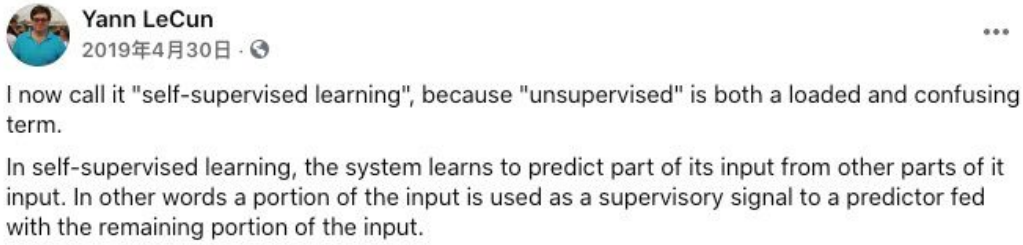


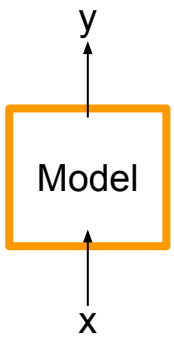
Task-specific data with labels



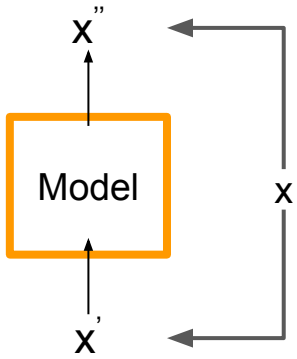
How to Pre-Train

The answer is **self-supervised learning**.





Supervised



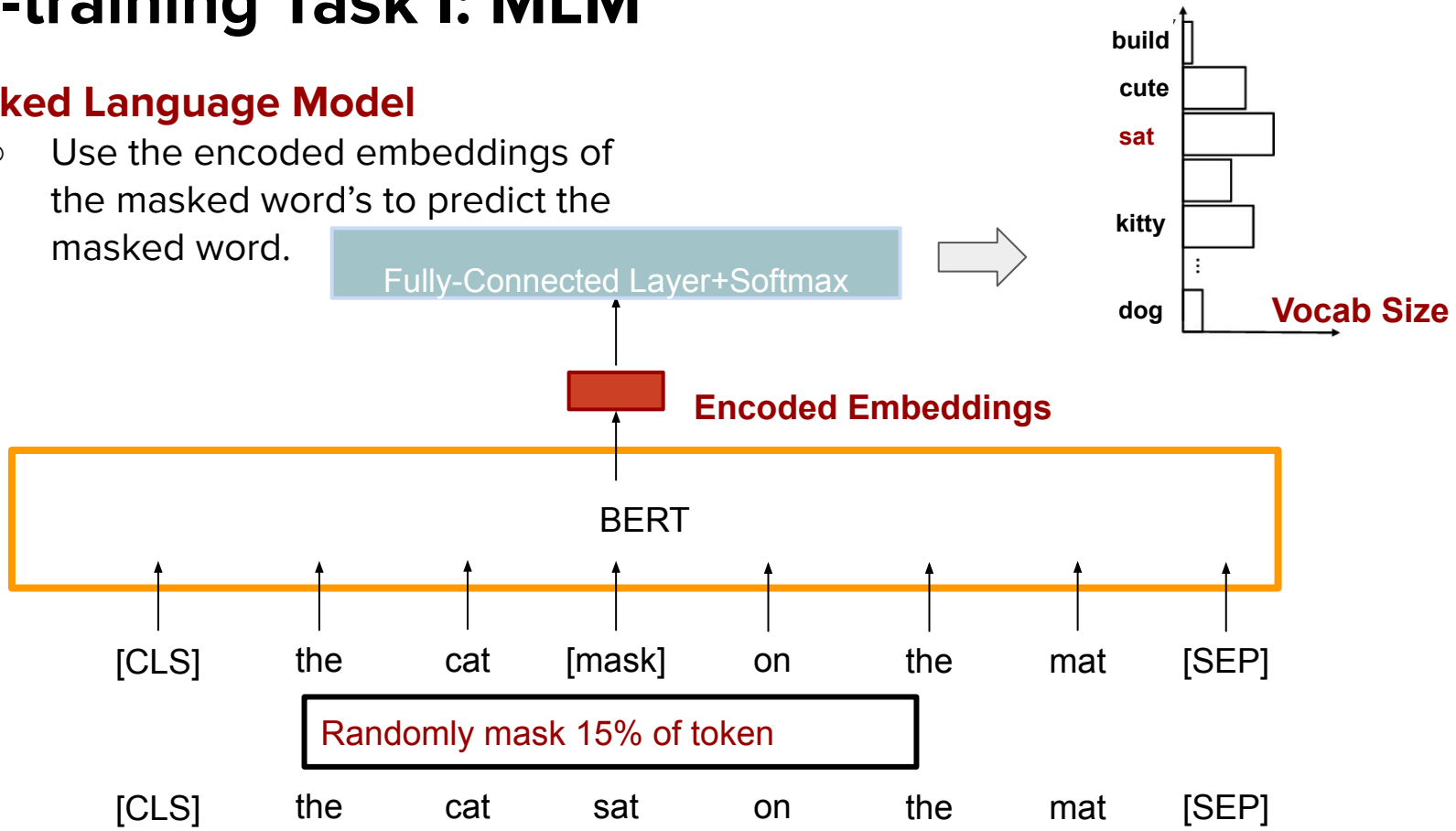
Generated by Rules

Automatically generate some kind of supervisory tasks

Self-Supervised

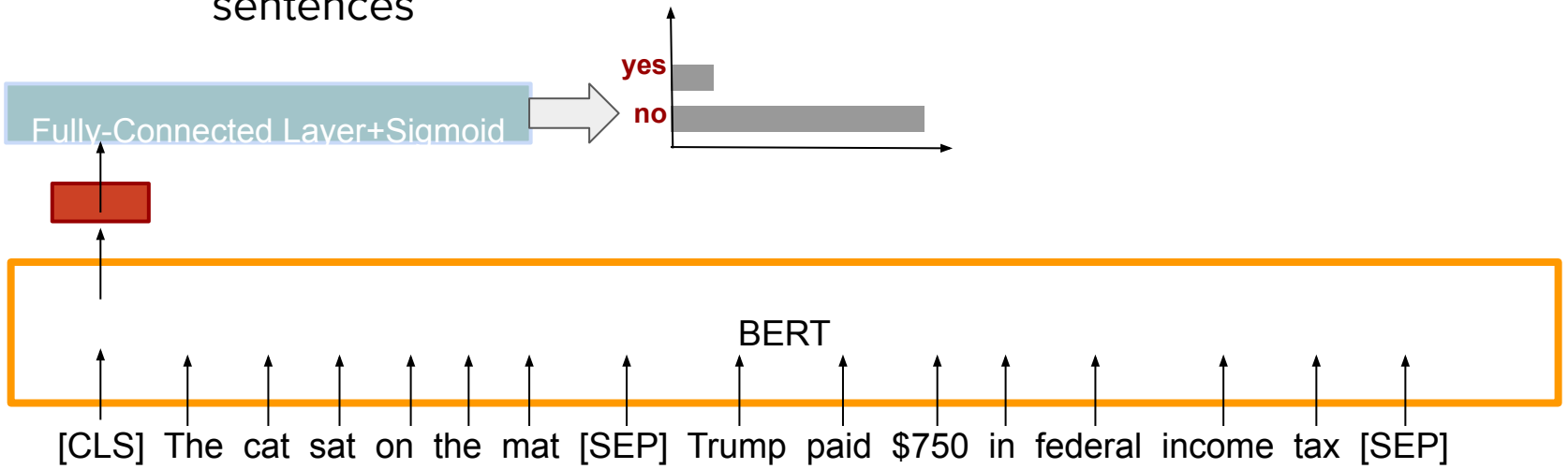
Pre-training Task I: MLM

- **Masked Language Model**
  - Use the encoded embeddings of the masked word's to predict the masked word.

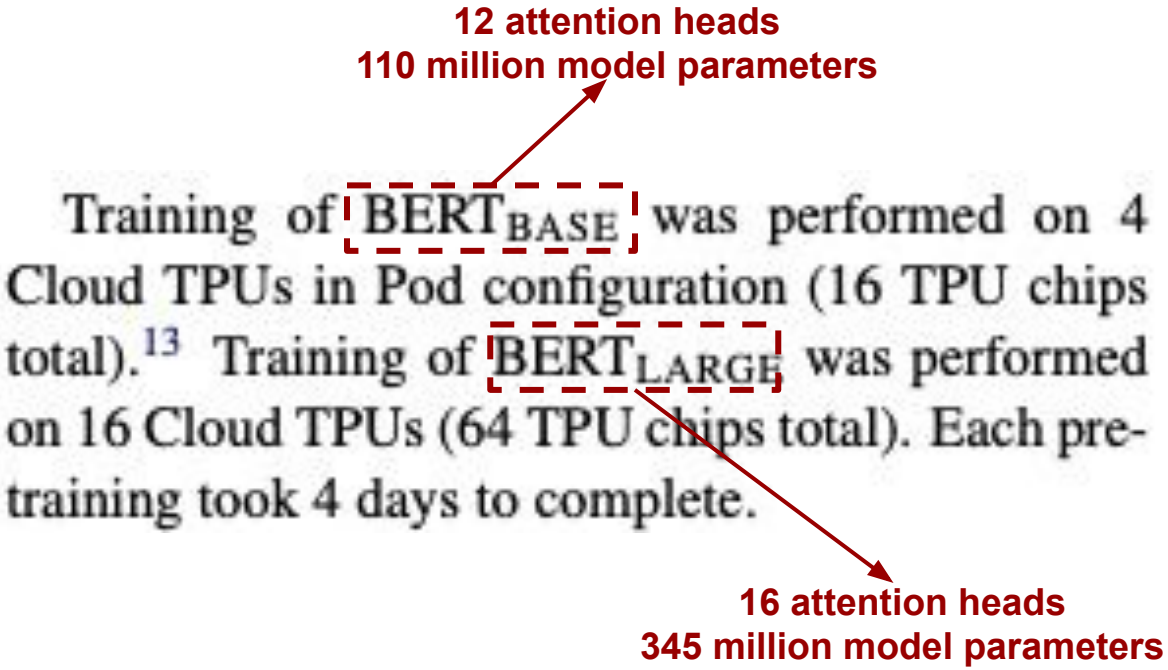


Pre-training Task II: NSP

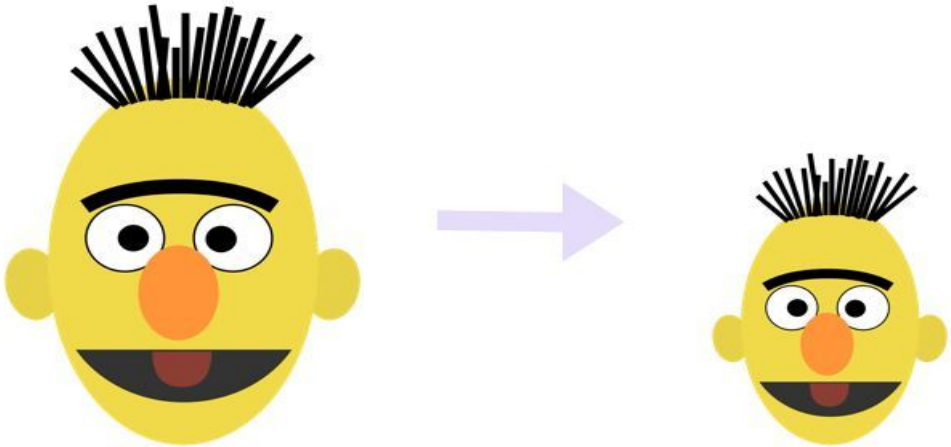
- **Next sentence prediction**
  - Given two sentences A and B, is B likely to be the sentence followed by A?
  - Make bert good at handling relationships between multiple sentences



Big Model Size



Smaller Model



Published as a conference paper at ICLR 2020

ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS

Zhenzhong Lan[1] Mingda Chen[2*] Sebastian Goodman[1] Kevin Gimpel[2]
Piyush Sharma[1] Radu Soricut[1]

DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

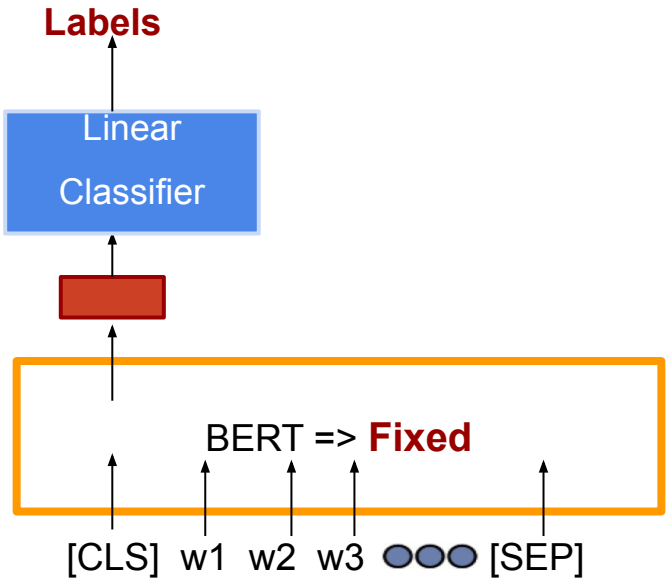
Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF
Hugging Face
{victor,lysandre,julien,thomas}@huggingface.co

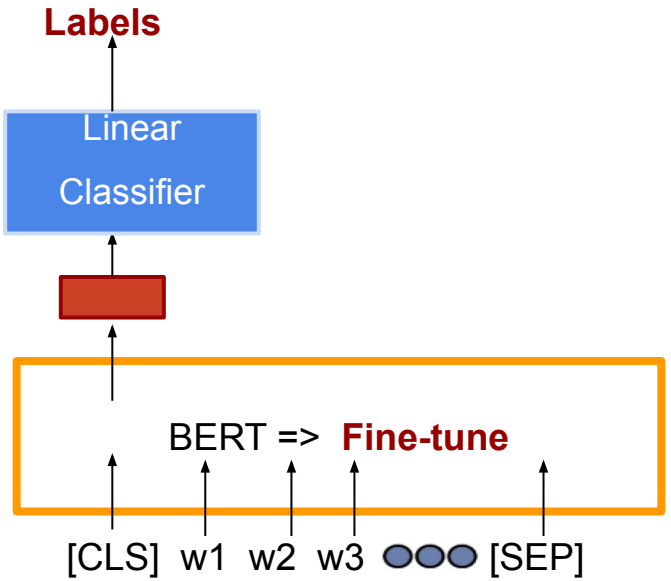
Abstract

Good summary:
http://mitchgordon.me/machine/learning/2019/11/18/all-the-ways-to-compress-BERT.html

BERT Usage I

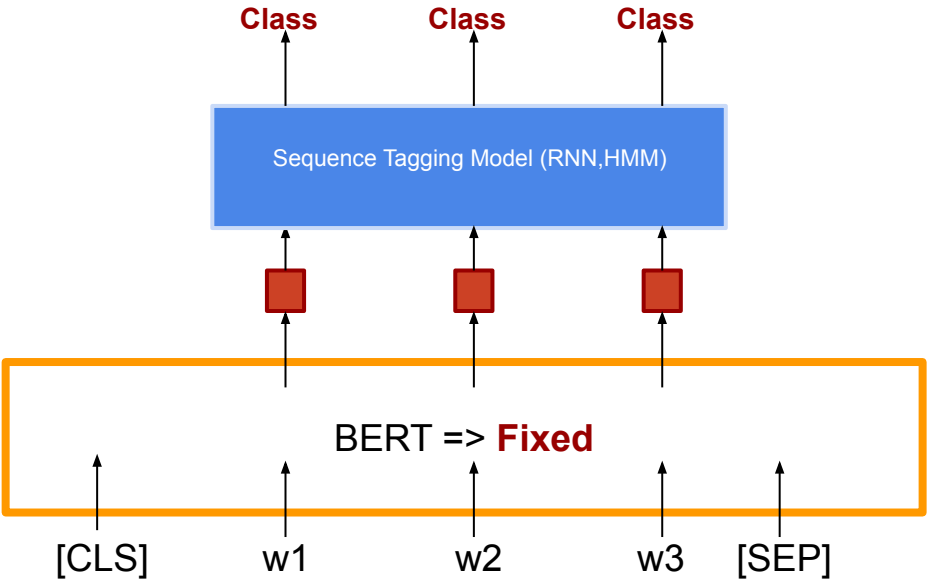
- **Input: Single Sentence Output: Class**
  - Sentiment Analysis
  - Document Classification

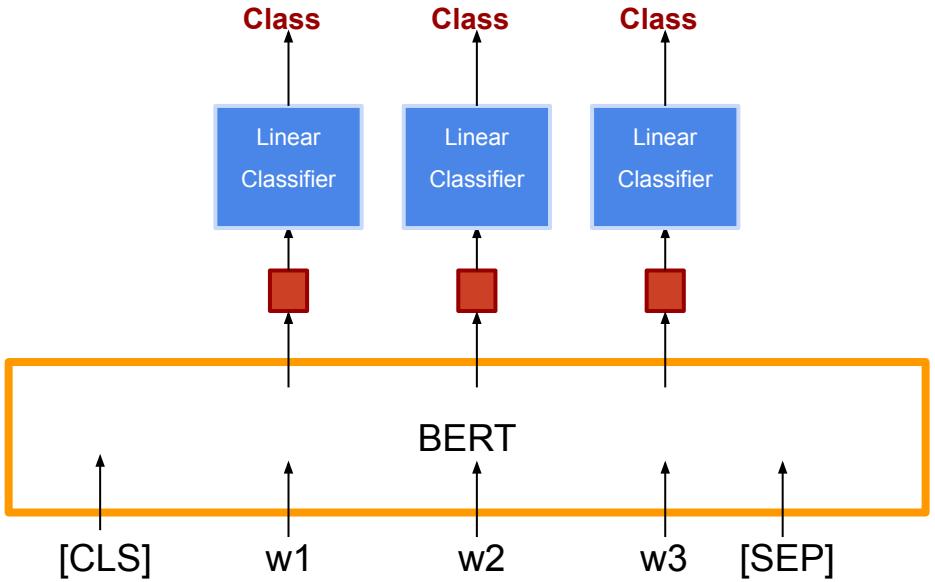




BERT Usage II

- **Input: Single Sentence Output: Class**
  - NER, POS Tagging

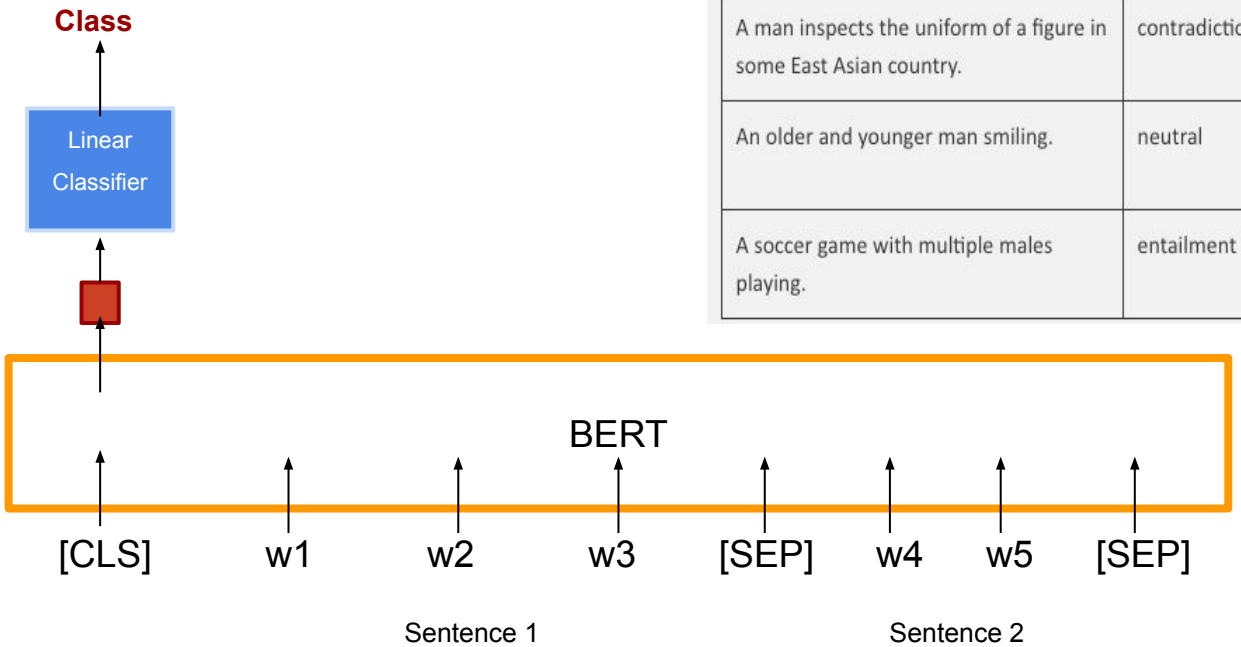




BERT Usage III

- **Input: Two Sentences Output: Class**
  - Natural Language Inference

Premise	Label	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction	The man is sleeping.
An older and younger man smiling.	neutral	Two men are smiling and laughing at the cats playing on the floor.
A soccer game with multiple males playing.	entailment	Some men are playing a sport.



BERT Usage IV

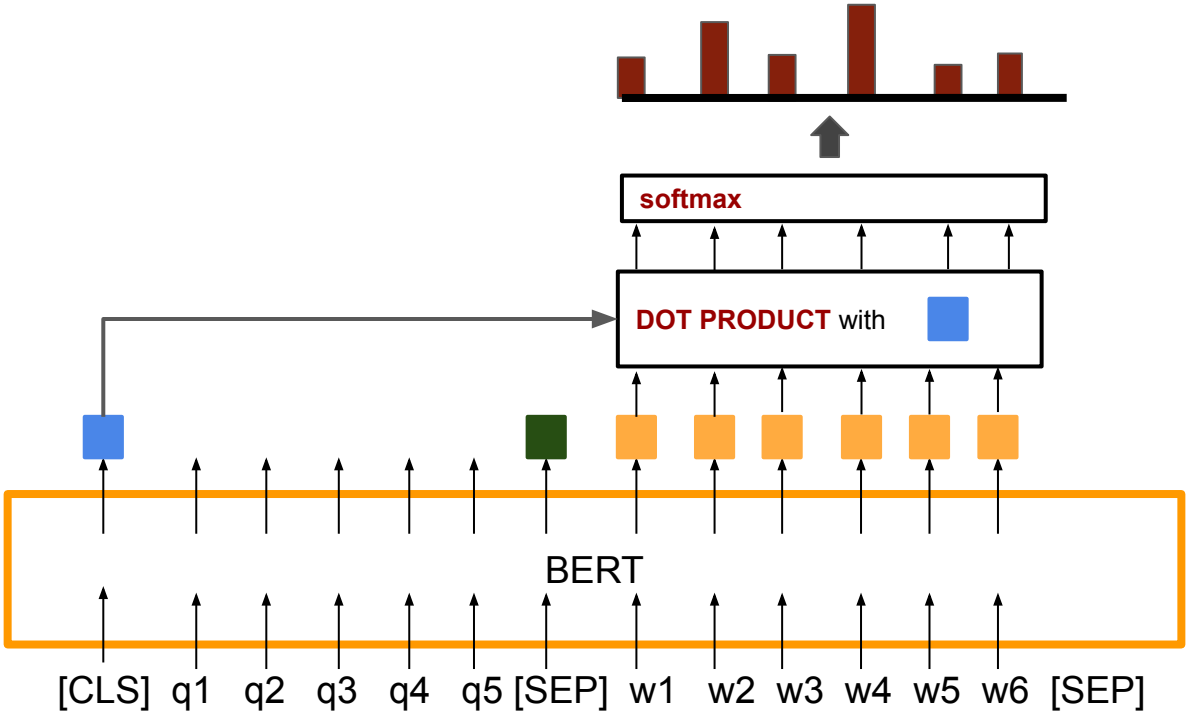
- Extraction-based Question Answering (SQuAD):
  - Input: two “sentences” (Question and Reference Text)
  - Output: start and end positions in Reference (Answer)

Question: How many parameters does BERT-large have?

Reference Text: BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

BERT Usage IV

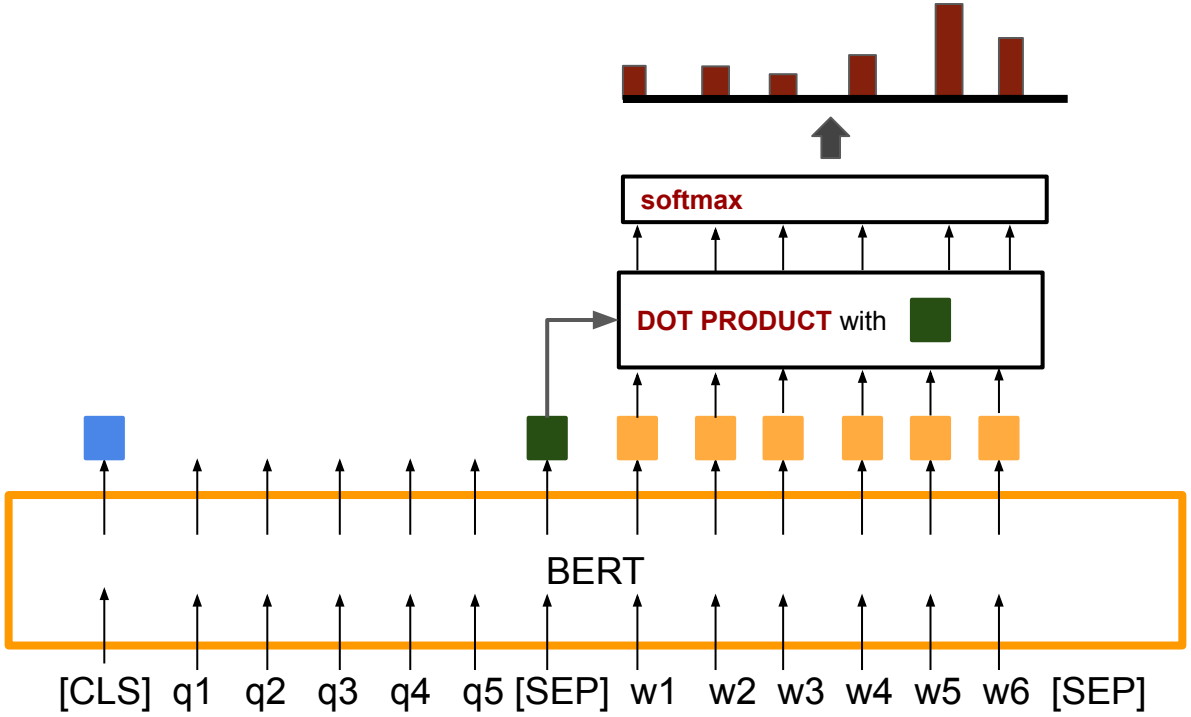
- Extraction-based Question Answering (SQuAD):
  - Question {q1,q2,q3,q4,q5} Reference Text{w1,w2,w3,w4,w5,w6}



The starting position for answer in reference is 4

BERT Usage IV

- Extraction-based Question Answering (SQuAD):
  - Question {q1,q2,q3,q4,q5} Reference Text{w1,w2,w3,w4,w5,w6}

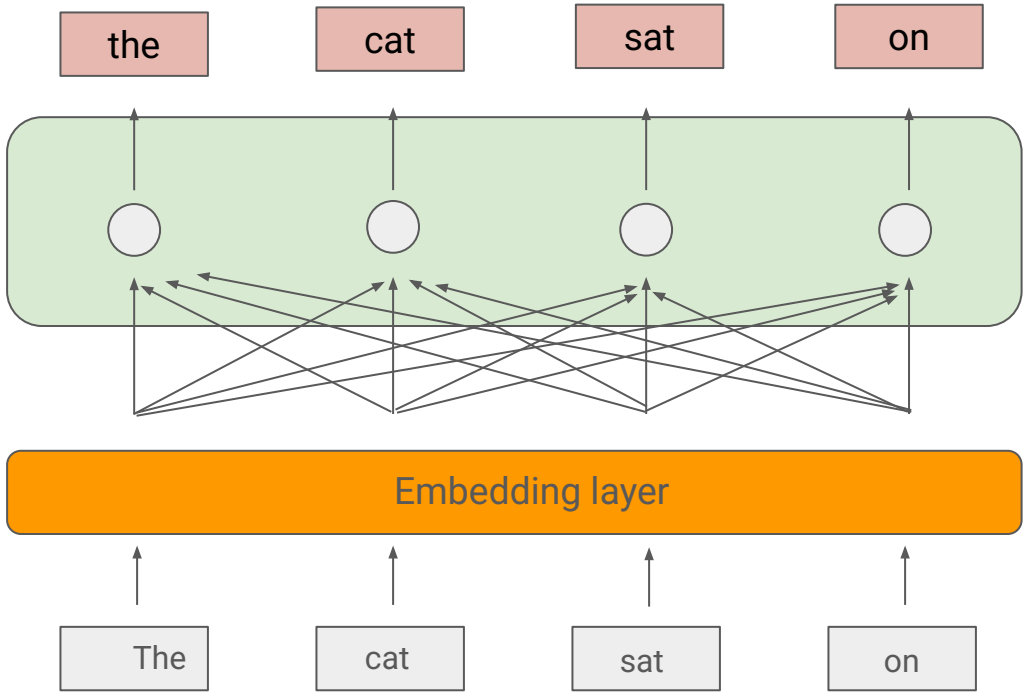


The starting position for answer in reference is 4

The ending position for answer in reference is 5

The answer is w4w5

Model Architecture - BERT



Get embeddings for all positions (used for classification, Q&A, etc)

- *[CLS]: classification*
- *[mask]: token prediction*
- *Tokens: NER*

Transformers layers (12-24) using ***Bi-directional*** *self-attention mechanism (Viswani et. al. 2017)*

Convert tokens into fixed-dimensional numerical vectors (~768 dimensions)

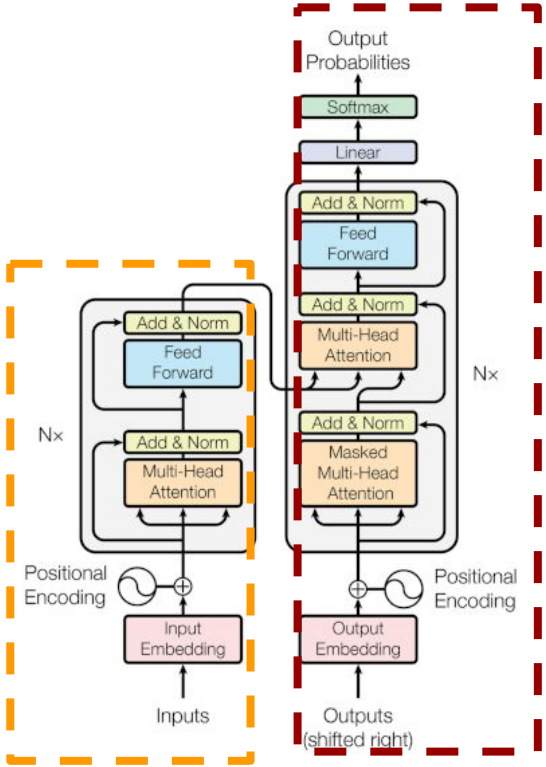
Tokenize inputs using fixed vocabulary

BERT, GPT and Transformers

BERT

Encoder
Maksed “Language” Model

Learn representation



Transformer

GPT2
GPT3

OpenAI

GPT-3, an autoregressive language model with 175 billion parameters

Decoder
Language Model

Generate probabilities

2. LLM Basics

Language Models

- A statistical model than learns probability distributions over sequences of words or tokens
  - Core concept: P(word | context)
  - Given previous words, predict the probability of the next word

Example: *"The cat sat on the"* → P("mat") = 0.35, P("floor") = 0.25, P("chair") = 0.20, ...

Journal of Machine Learning Research 3 (2003) 1137–1155 Submitted 4/02; Published 2/03

A Neural Probabilistic Language Model

Yoshua Bengio BENGIOY@IRO.UMONTREAL.CA
Réjean Ducharme DUCHARME@IRO.UMONTREAL.CA
Pascal Vincent VINCENTP@IRO.UMONTREAL.CA
Christian Jauvin JAUVINC@IRO.UMONTREAL.CA
Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal, Montréal, Québec, Canada

https://jmlr.org/papers/volume3/tmp/bengio03a.pdf

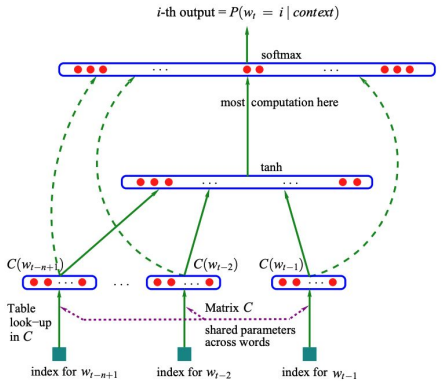


Figure 1: Neural architecture: $f(i, w_{t-1}, \cdots, w_{t-n+1}) = g(i, C(w_{t-1}), \cdots, C(w_{t-n+1}))$ where g is the neural network and $C(i)$ is the i-th word feature vector.

What is LLM (A textbook definition)

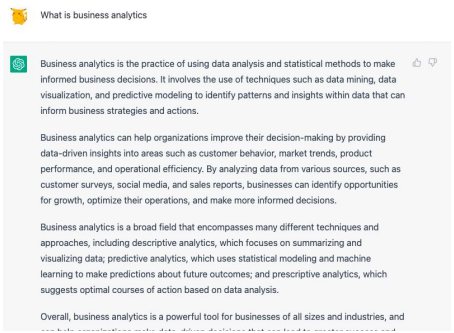
Large language models (LLMs) mainly refer to transformer-based neural language models [1] that contain tens to hundreds of billions of parameters, which are pre-trained on massive text data, such as PaLM [31], LLaMA [32], and GPT-4 [33], as summarized in Table III. Compared

Source: https://arxiv.org/pdf/2402.06196.pdf

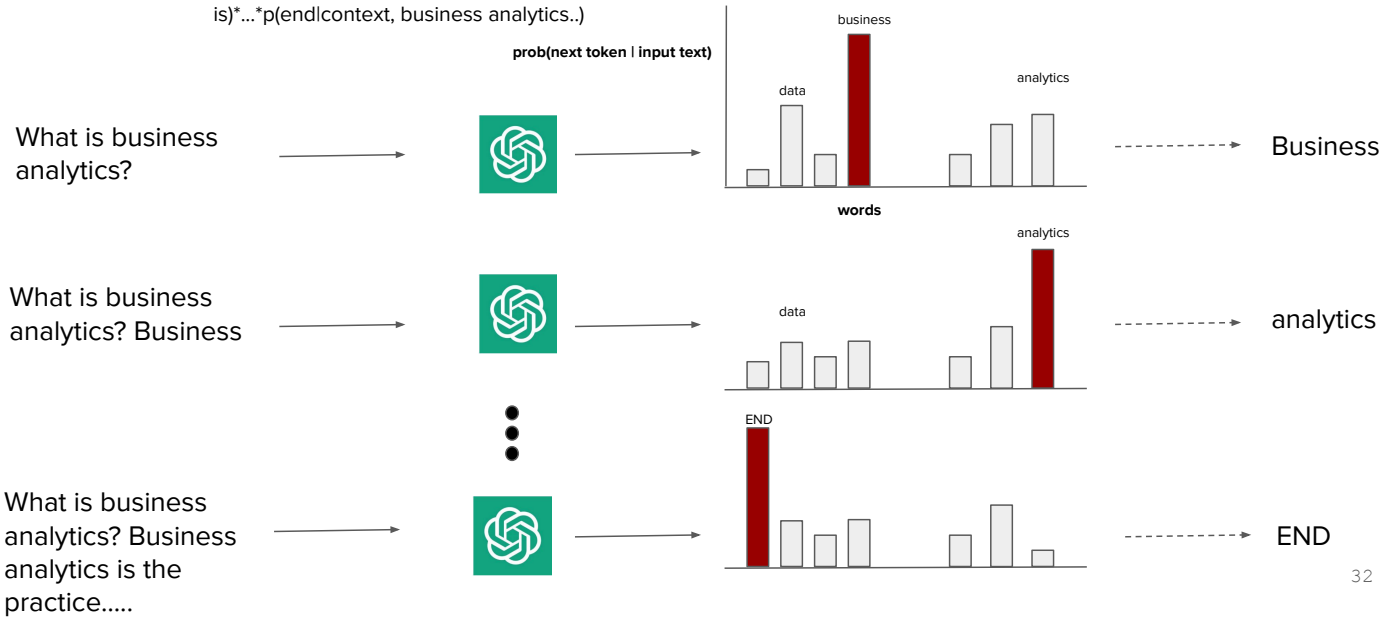
What is LLM

- Scale = Billions of Parameters + Massive Training data
  - Emergent capabilities appear only at sufficient scale
- Emergent Capabilities
  - Few-shot and zero shot learning
  - Complex Reasoning and chain-of-thought
  - Task Generalization without fine-tuning
  - In-context learning from prompts
- Architecture
  - Transformer+Self-Attention
  - 10B-1T+ parameters
  - Trained on internet-scale data

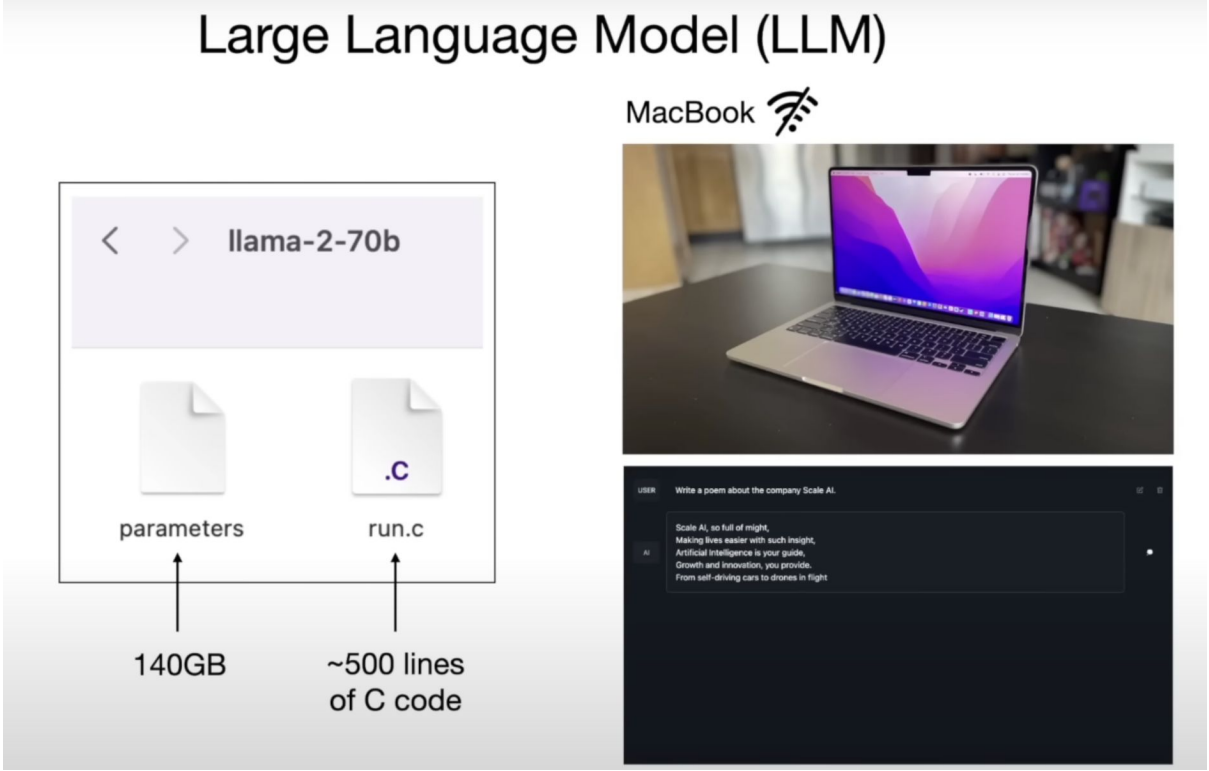
What is LLM (A basic perspective)



- LLMs are autoregressive models for next-token prediction
  - $P(x_1, x_2, \ldots, x_\square \mid \text{context}) = \prod_{i=1}^{t} P(x_i \mid \text{context}, x_1, x_2, \ldots, x_{i-1})$
  - Concrete Example: p(business analytics is the practice....)=p(business|context)*p(analytics|context,business)*p(is|context, business, analytics, is)*...*p(end|context, business analytics..)



What is LLM (A practical perspective)



LLM Inference

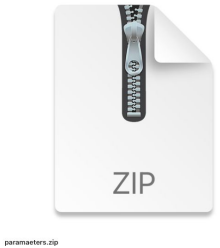
Source:
https://www.youtube.com/watch?v=zjkBMFhNj_g

LLM is compressing the Internet

Google DeepMind

Language Modeling Is Compression

Grégoire Delétang[*1], Anian Ruoss[*1], Paul-Ambroise Duquenne[2], Elliot Catt[1], Tim Genewein[1], Christopher Mattern[1], Jordi Grau-Moya[1], Li Kevin Wenliang[1], Matthew Aitchison[1], Laurent Orseau[1], Marcus Hutter[1] and Joel Veness[1]
[*]Equal contributions, [1]Google DeepMind, [2]Meta AI & Inria



Internet data
~45TB of text

1024 A100 GPUs, 34 days
$5Mdata

GPT3 ~175B

LLM's Evolution Process

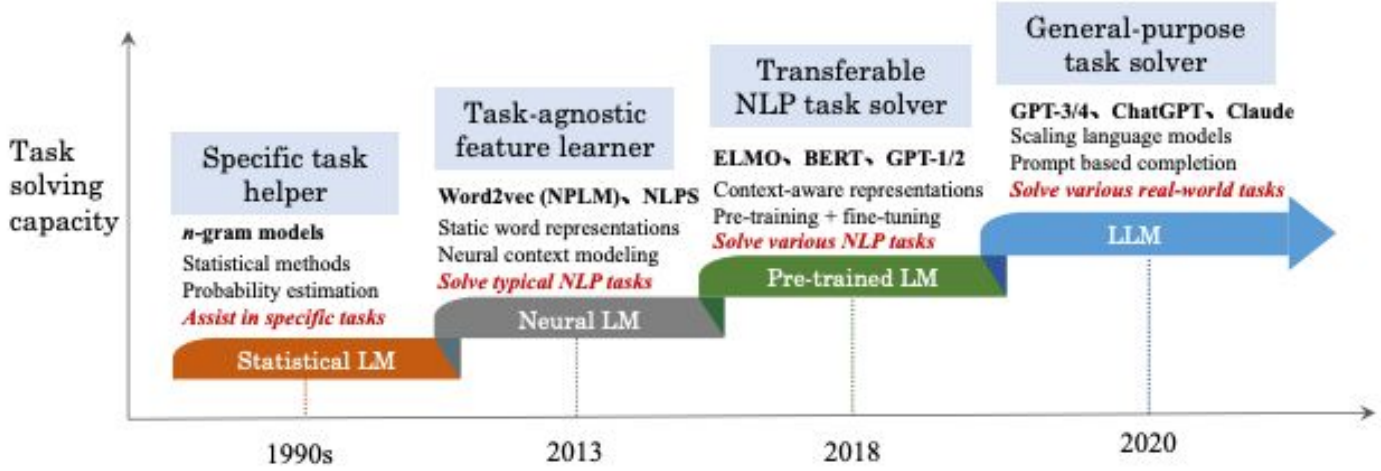
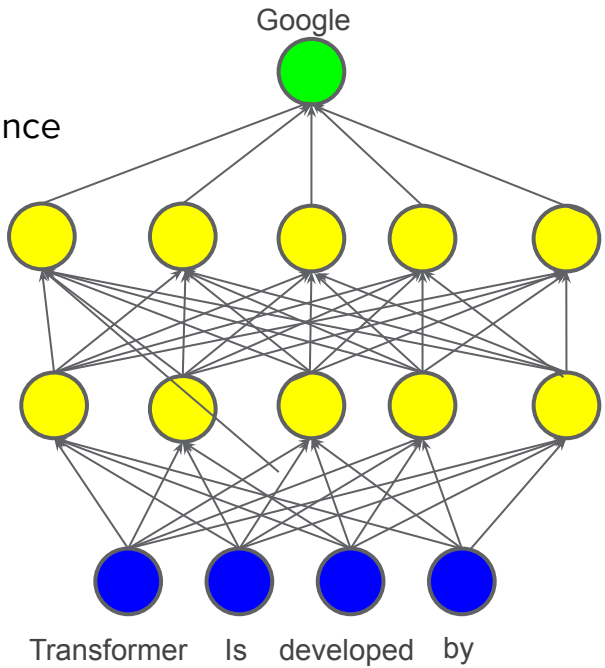


Fig. 2: An evolution process of the four generations of language models (LM) from the perspective of task solving capacity. Note that the time period for each stage may not be very accurate, and we set the time mainly according to the publish date of the most representative studies at each stage. For neural language models, we abbreviate the paper titles of two representative studies to name the two approaches: NPLM [1] (*"A neural probabilistic language model"*) and NLPS [2] (*"Natural language processing (almost) from scratch"*). Due to the space limitation, we don't list all representative studies in this figure.

source:https://arxiv.org/pdf/2303.18223.pdf

LLM Pretraining

- LLM
  - Pre-trained by **large-scale** unannotated corpus
- Training target: token prediction
  - For GPT/decoder: next token prediction in sequence
  - For BERT/encoder: in-context token prediction



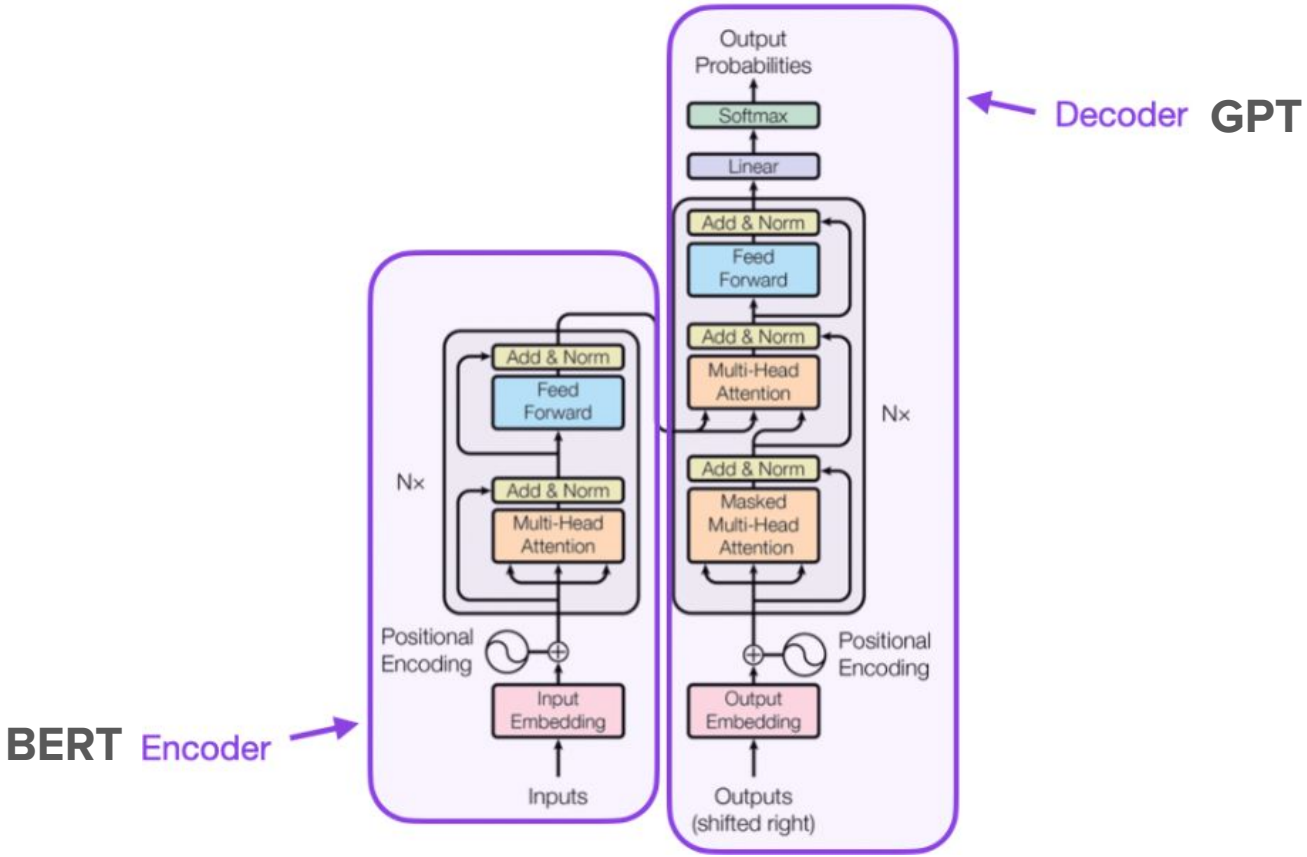
Word prediction is absorbing knowledge

"Transformer architecture" redirects here. For the design of electrical transformers, see Transformer.

A **transformer** is a deep learning architecture based on the multi-head attention mechanism.[1] It is notable for not containing any recurrent units, and thus requires less training time than previous recurrent neural architectures, such as long short-term memory (LSTM),[2] and its later variation has been prevalently adopted for training large language models on large (language) datasets, such as the Wikipedia corpus and Common Crawl.[3] Input text is split into n-grams encoded as tokens and each token is converted into a vector via looking up from a word embedding table. At each layer, each token is then contextualized within the scope of the context window with other (unmasked) tokens via a parallel multi-head attention mechanism allowing the signal for key tokens to be amplified and less important tokens to be diminished. Though the transformer paper was published in 2017, the softmax-based attention mechanism was proposed in 2014 for machine translation,[4][5] and the Fast Weight Controller, similar to a transformer, was proposed in 1992.[6][7][8]

This architecture is now used not only in natural language processing and computer vision,[9] but also in audio[10] and multi-modal processing. It has also led to the development of pre-trained systems, such as generative pre-trained transformers (GPTs)[11] and BERT[12] (Bidirectional Encoder Representations from Transformers).

Transformer is the backbone of LLMs



3. GPTs

What is GPT

- **G**enerative **P**re-trained **T**ransformer
  - GPT: Decoder only of Transformer
  - Goal: Learn how to generate high-quality text

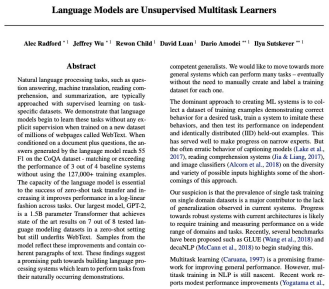
Improving Language Understanding by Generative Pre-Training

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever
OpenAI
alec@openai.com karthikn@openai.com tim@openai.com ilyasu@openai.com

Abstract

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by *generative pre-training* of a language model on a diverse corpus of unlabeled text, followed by *discriminative fine-tuning* on each specific task. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. For instance, we achieve absolute improvements of 8.9% on commonsense reasoning (Stories Cloze Test), 5.7% on question answering (RACE), and 1.5% on textual entailment (MultiNLI).

GPT1



Language Models are Unsupervised Multitask Learners

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

Abstract

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, GPT-2, is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text. These findings suggest a promising path towards building language processing systems which learn to perform tasks from their naturally occurring demonstrations.

competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks. Recently, several benchmarks have been proposed such as GLUE (Wang et al., 2018) and decaNLP (McCann et al., 2018) to begin studying this.

Multitask learning (Caruana, 1997) is a promising framework for improving general performance. However, multitask training in NLP is still nascent. Recent work reports modest performance improvements (Yogatama et al.,

GPT2

Language Models are Few-Shot Learners

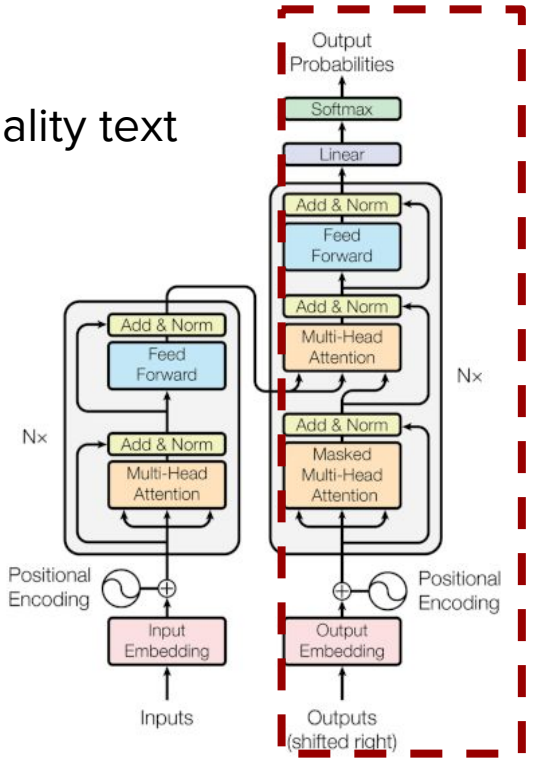
Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*
Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry
Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan
Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter
Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray
Benjamin Chess Jack Clark Christopher Berner
Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

Abstract

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely struggle to do. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3's few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora. Finally, we find that GPT-3 can generate samples of news articles which human evaluators have difficulty distinguishing from articles written by humans. We discuss broader societal impacts of this finding and of GPT-3 in general.

GPT3

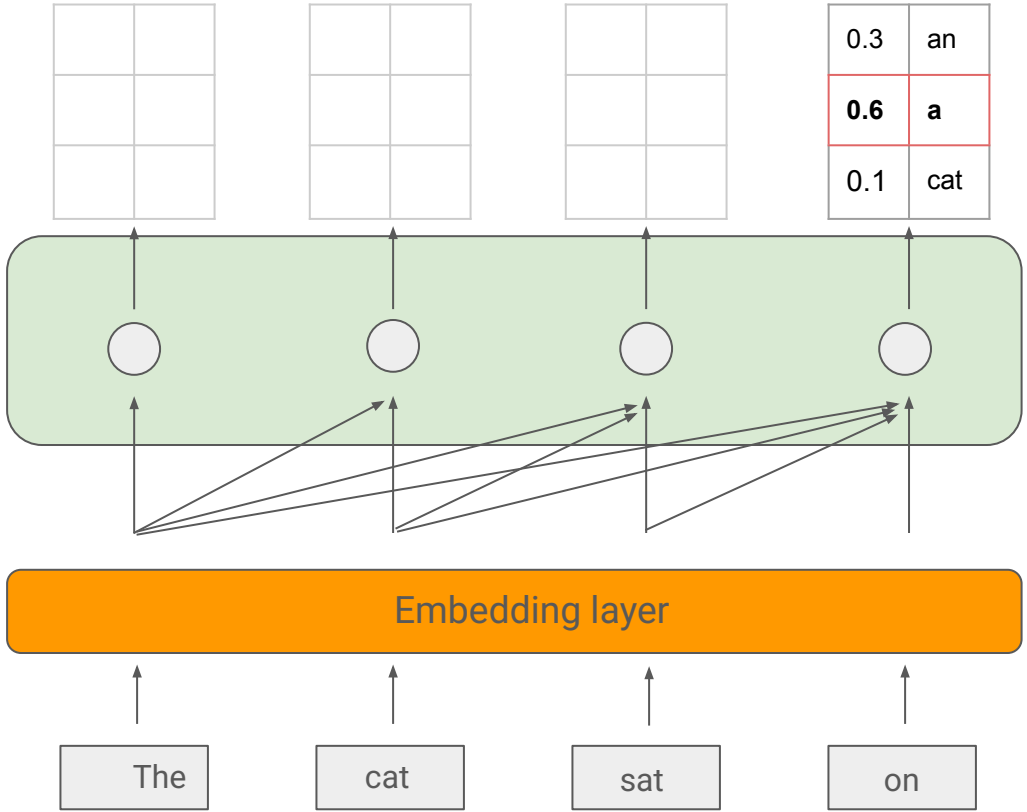


Decoder Language Model

Why Decoder-Only Architecture

- Casual self-attention
  - Each token only attends to previous tokens
- Natural for sequential generation
  - predicts on tokens at a time
- Simpler than encoder-decoder
  - single unified architecture
- Scalable
  - efficient training on massive datasets

Model Architecture



*Get probability distribution over most likely **next token***

Transformers layers (12-96+) using causal-attention mechanism
Position i

Convert tokens into fixed-dimensional numerical vectors (~1-3K dimensions.))

Tokenize inputs using fixed vocabulary. Usually, it has two special tokens indicating BoS and Eos

Causal Masking

Attention weights before masking

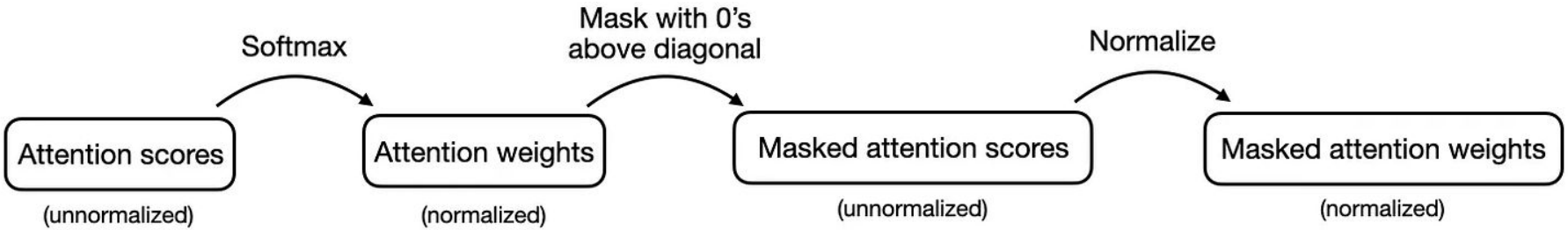
	the	cat	sat	on
the	0.42	0.21	0.19	0.18
cat	0.35	0.28	0.20	0.17
sat	0.22	0.31	0.29	0.18
on	0.19	0.24	0.33	0.24

Masked out future tokens

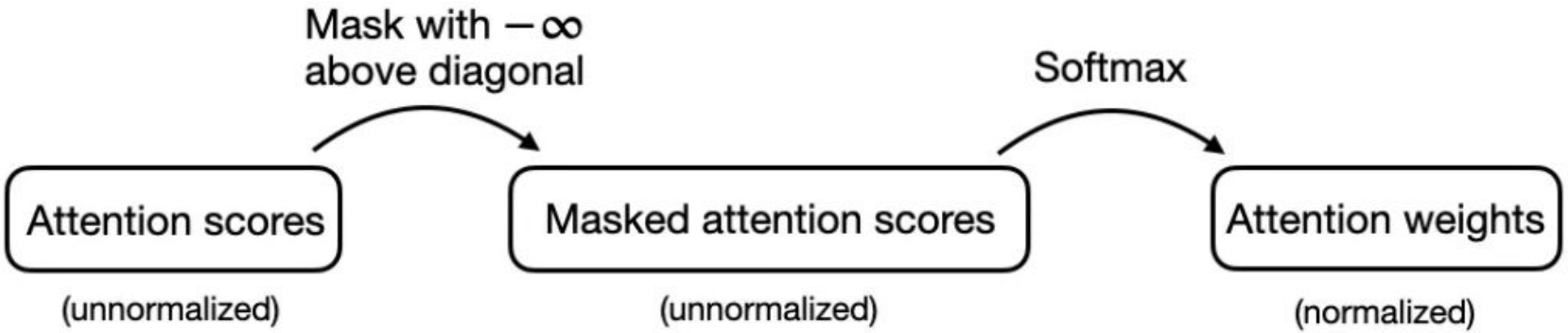
	the	cat	sat	on
the	0.42			
cat	0.35	0.28		
sat	0.22	0.31	0.29	
on	0.19	0.24	0.33	0.24

Attention weights above the diagonal line should be masked (Each token can only attend to itself and previous tokens)

Implementation of Causal Masking



Optimized Implementation of Causal Masking



GPT1

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

Abstract

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by *generative pre-training* of a language model on a diverse corpus of unlabeled text, followed by *discriminative fine-tuning* on each specific task. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. For instance, we achieve absolute improvements of 8.9% on commonsense reasoning (Stories Cloze Test), 5.7% on question answering (RACE), and 1.5% on textual entailment (MultiNLI).

GPT1 laid the groundwork with a decoder-only architecture to show the potential of LLM.

GPT1 Training

1. Unsupervised Pre-training: next token prediction
2. Fine Tuning: A fully connected layer would be used for label prediction. And it is task-specific

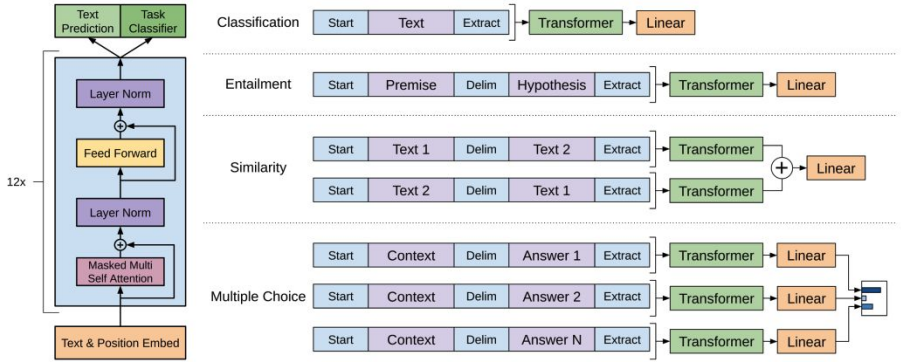


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Source:
https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

GPT2

Language Models are Unsupervised Multitask Learners

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

Abstract

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, GPT-2, is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text. These findings suggest a promising path towards building language processing systems which learn to perform tasks from their naturally occurring demonstrations.

competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks. Recently, several benchmarks have been proposed such as GLUE (Wang et al., 2018) and decaNLP (McCann et al., 2018) to begin studying this.

Multitask learning (Caruana, 1997) is a promising framework for improving general performance. However, multitask training in NLP is still nascent. Recent work reports modest performance improvements (Yogatama et al.,

1. GPT2 is trying to build a general language model that could do multi-task learning while training
2. Compared to GPT1, there is no change in architecture. GPT2 has more parameters and a much bigger training dataset
3. No fine-tuning

GPT2

”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”

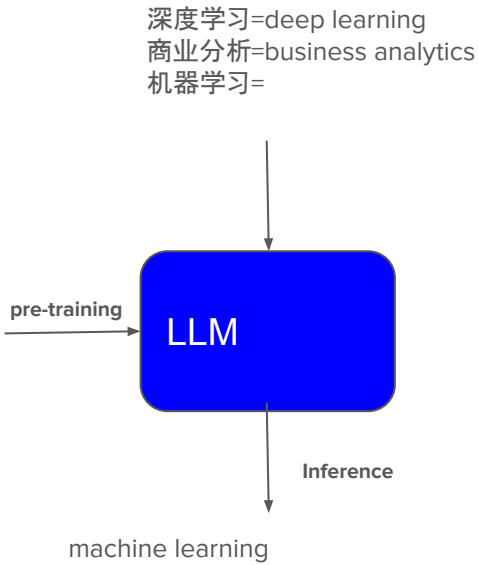
“I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’

If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

Table 1. Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.



A context of example pairs of chinese text=english is provided to help the LLM infer this is the machine translation task.

GPT3

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

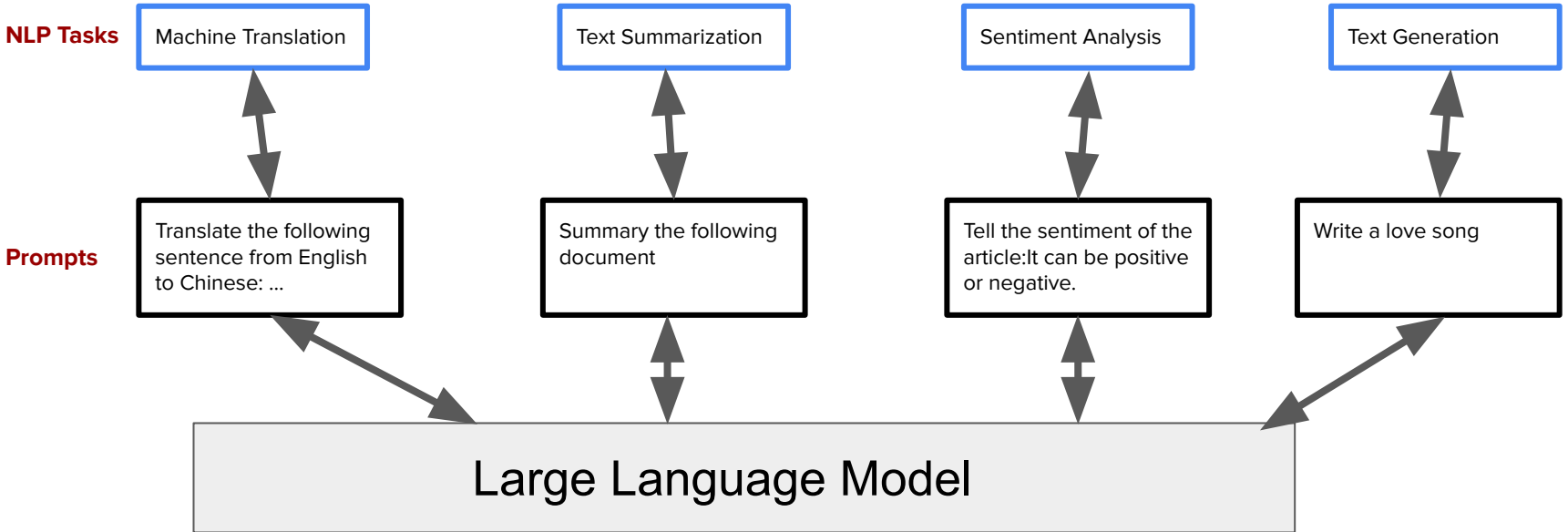
Abstract

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely struggle to do. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3's few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora. Finally, we find that GPT-3 can generate samples of news articles which human evaluators have difficulty distinguishing from articles written by humans. We discuss broader societal impacts of this finding and of GPT-3 in general.

1. Compared to GPT2 and GPT1, more parameters and bigger training dataset are used in GPT3. And the generalization capability is named as in-context learning.
2. GPT2 has constraints in handling certain specific tasks while GPT3 show groundbreaking abilities. So it has paved the way for even bigger and more complex models

In-context Learning

In-context learning: using the text input of a pre-trained language model as a form of task specification: the model is conditioned on a natural language instruction and/or a few demonstrations of the task and is then expected to complete further instances of the task simply by predicting what comes next.



GPT3: Prompting

- Zero-shot Prompting
  - No examples are given in prompt
  - “Please answer, 3+2=?”
- One-shot Prompting
  - One example is given
  - “1+7=8, please answer, 3+2=?”
- Few-shot Prompting
  - A few shot examples of tasks are provided
  - “1+1=2,1+7=8, please answer, 3+2=?”

GPT3: Performances of Prompting

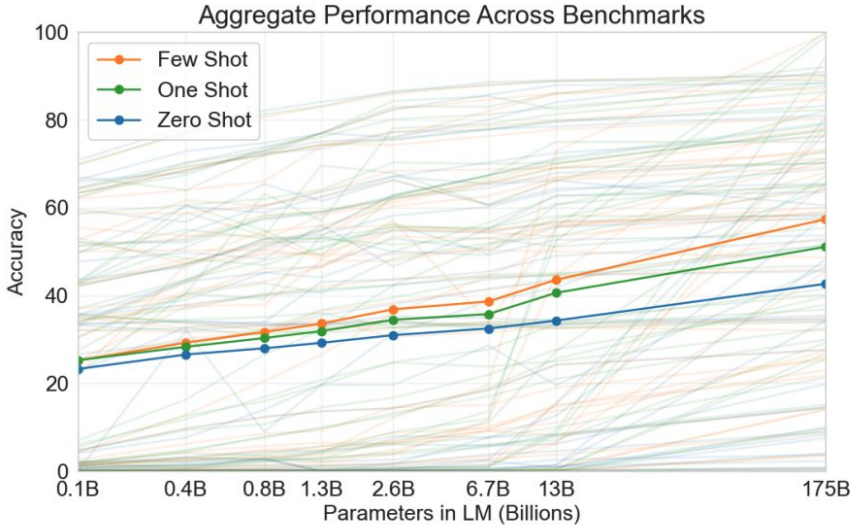


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

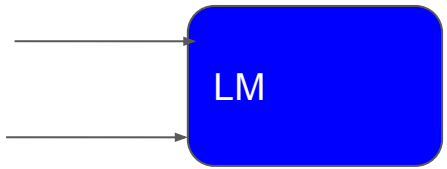
Source: https://arxiv.org/pdf/2005.14165.pdf

Downstream NLP tasks can all be formulated as LM

- Language model is doing next token prediction
  - E.g., based on the previous tokens: I love this -> movie
- Downstream NLP tasks:
  - Sentiment analysis: given a sentence, **generate** sentiment label
    - I love this movie -> positive
  - Machine translation: given a source sentence, **generate** a target sentence
    - 深度学习 -> deep learning
- How LM differentiate those NLP tasks?
  - Provide in-context information (prompt)

深度学习 Translate it into english:

I love this movie. Label it sentiment



BERT vs GPT

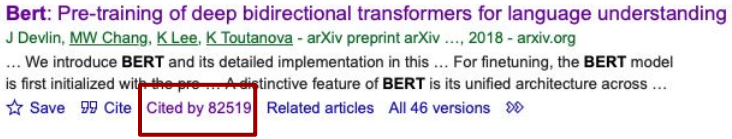
BERT

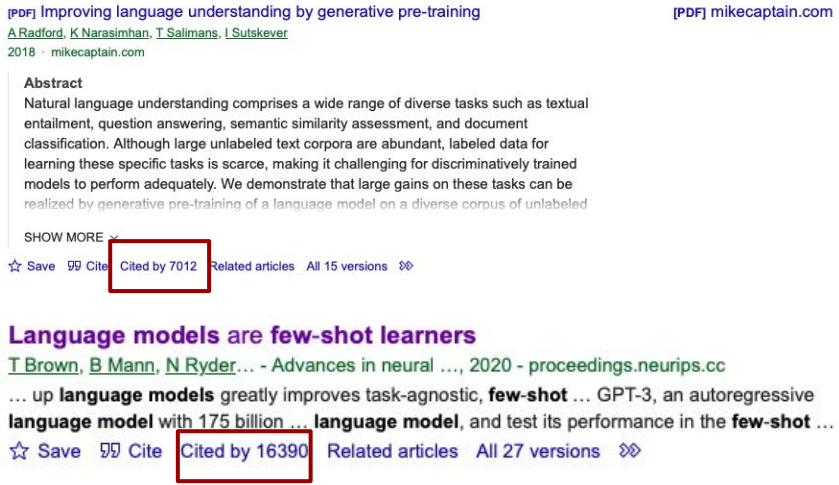
- Architecture:
  - Transformer Encoder block
  - Less training parameters (a few hundred M)
- Model learning:
  - Two objectives: masked language model (cbow) and next sentence prediction
    - The [MASK] sat on mat
  - **Bi-directional**
  - Less training data
- Applications:
  - Text classification
  - NER
  - QA
  - Sentence embeddings

GPT

- Architecture:
  - Transformer Decoder block
  - More training parameters (a few hundred B)
- Model learning:
  - Generative, next word prediction
    - The cat sat on -> mat
  - **Uni-directional (left to right)->Causal**
  - More training data
- Applications:
  - Text generation
  - Creative writing
  - Chatbots with multi-turn conversation
  - Coding

BERT vs GPT: BERT was winning

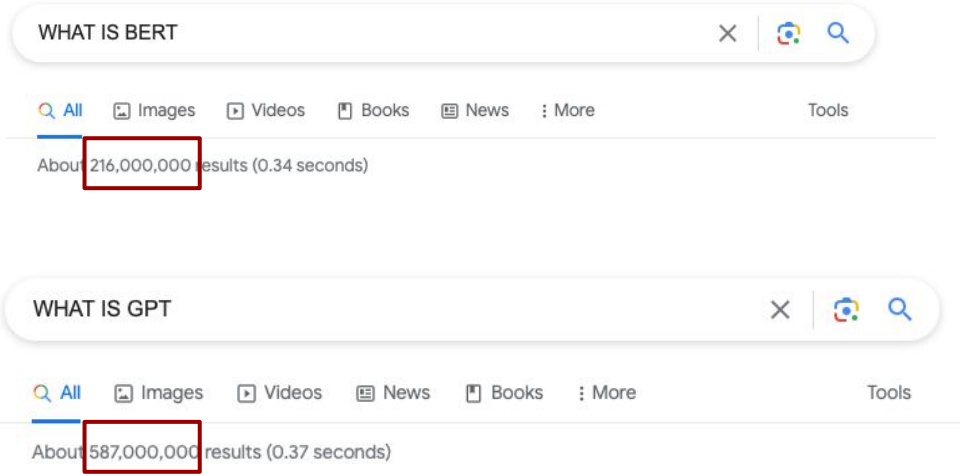




At beginning, BERT got more adoption from NLP community compared to GPT and GPT2 (82k citation vs 23k citation)

BERT vs GPT: GPT method is the SOTA now

- With the popularity of ChatGPT, GPT method is winning now
  - We understand others by the response
  - Representations or encodings does not matter, we can rely on outputs for any specific tasks
  - Closer to the idea of General AI (only one model)



Next Class: Training & Scaling LLMs