DSC291: Machine Learning with Few Labels

Large Language Model Basics
Self-Supervised Learning

Zhiting Hu

Lecture 4, April 8, 2024

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Components of a ML solution (roughly)

- Loss
- Experience
- Optimization solver
- Model architecture

$$\min_{\theta} \mathcal{L}(\theta, \mathcal{E})$$

Optimization solver

Loss

Model architecture

Experience

Algorithm marketplace

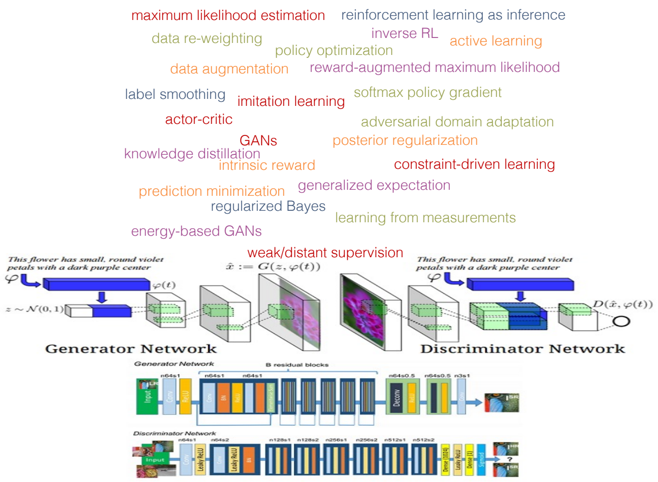
Designs driven by: experience, task, loss function, training procedure …

maximum likelihood estimation
reinforcement learning as inference
data re-weighting
inverse RL
active learning
policy optimization
data augmentation
reward-augmented maximum likelihood
label smoothing
imitation learning
softmax policy gradient
actor-critic
adversarial domain adaptation
GANs
posterior regularization
knowledge distillation
intrinsic reward
constraint-driven learning
prediction minimization
generalized expectation
regularized Bayes
learning from measurements
energy-based GANs
weak/distant supervision

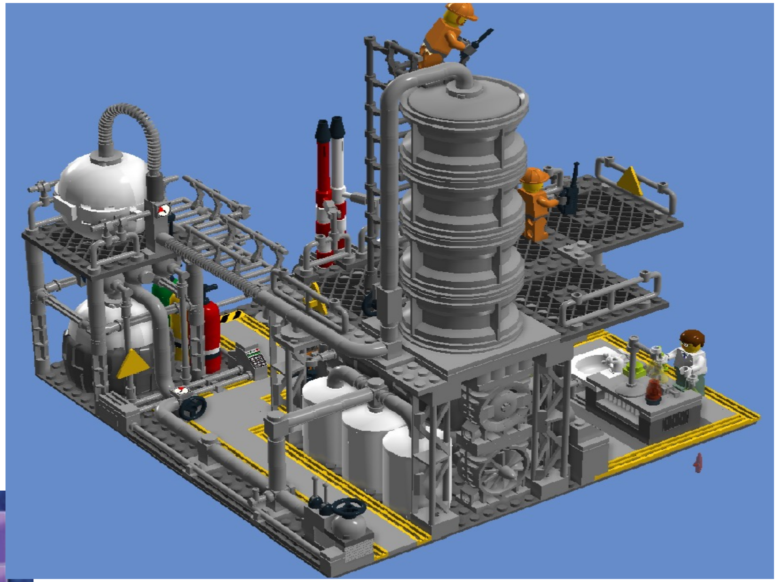
Where we are now? Where we want to be?

- Alchemy vs chemistry

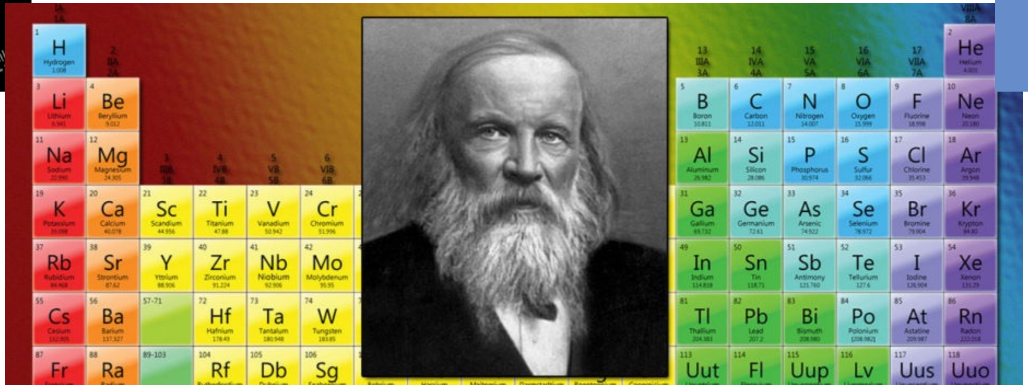
≈



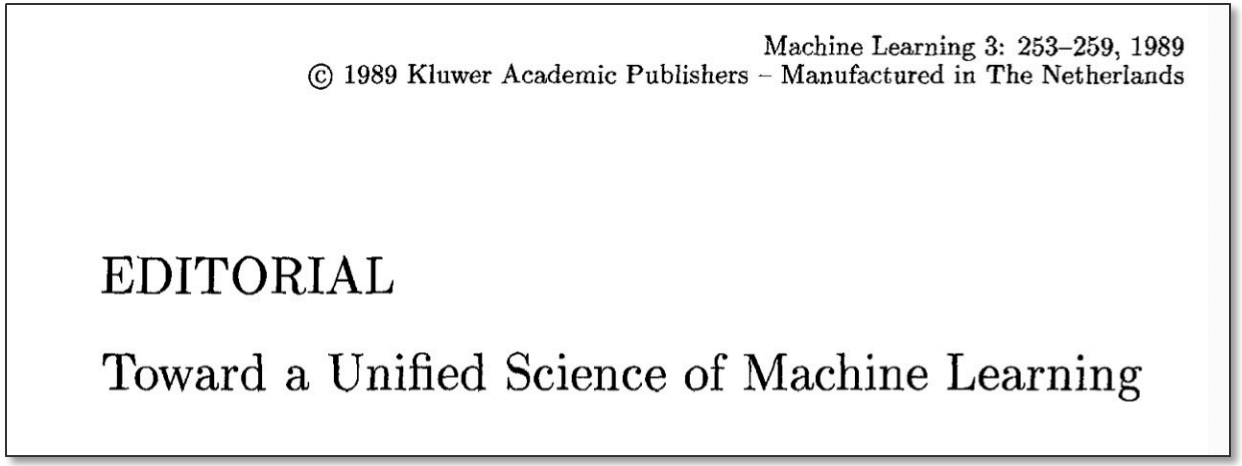
≠≠



∦



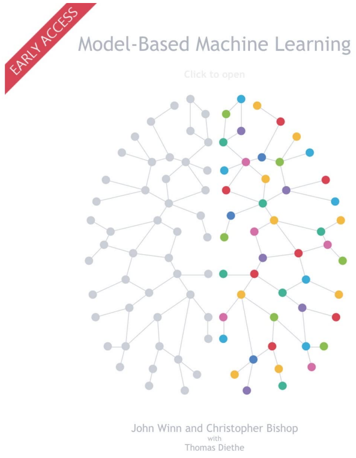
Quest for more standardized, unified ML principles

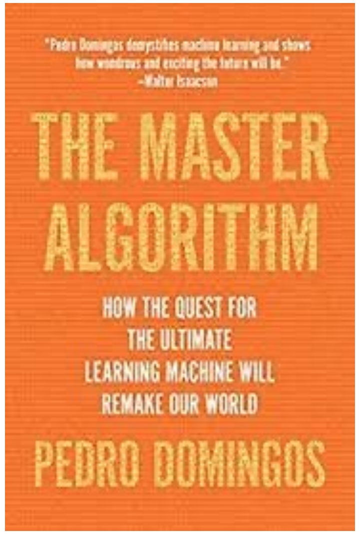

Machine Learning 3: 253–259, 1989
© 1989 Kluwer Academic Publishers – Manufactured in The Netherlands

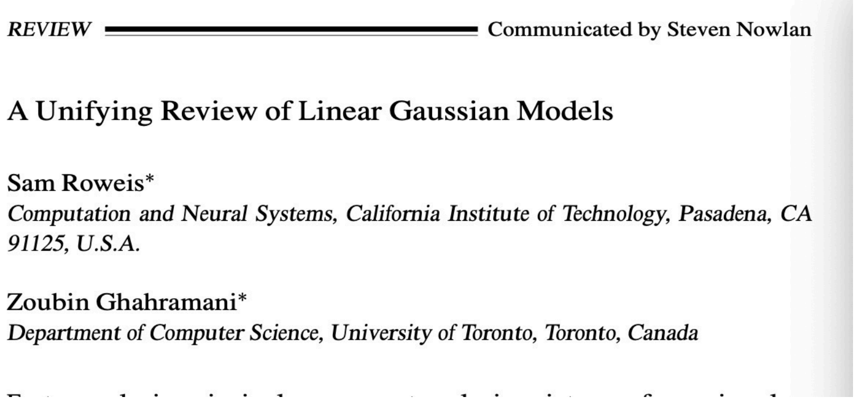
EDITORIAL

Toward a Unified Science of Machine Learning

[P. Langley, 1989]






REVIEW Communicated by Steven Nowlan

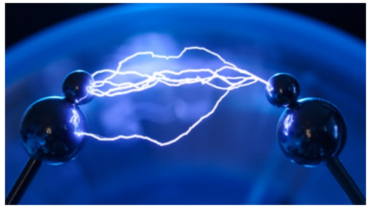
A Unifying Review of Linear Gaussian Models

Sam Roweis*
Computation and Neural Systems, California Institute of Technology, Pasadena, CA 91125, U.S.A.

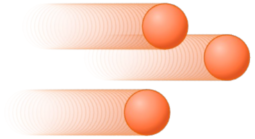
Zoubin Ghahramani*
Department of Computer Science, University of Toronto, Toronto, Canada

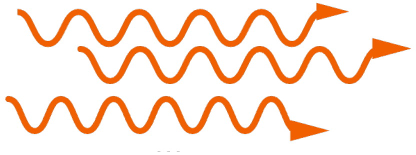
Physics in the 1800’s

- Electricity & magnetism:
    - Coulomb’s law, Ampère, Faraday, ...



- Theory of light beams:
    - Particle theory: Isaac Newton, Laplace, Plank
    - Wave theory: Grimaldi, Chris Huygens, Thomas Young, Maxwell





- Law of gravity
    - Aristotle, Galileo, Newton, …

“Standard equations” in Physics

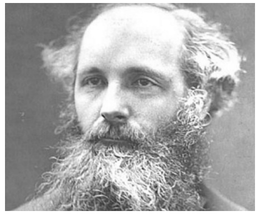
Diverse electro-magnetic theories

Maxwell’s Eqns: original form

$e+\frac{df}{dx}+\frac{dg}{dy}+\frac{dh}{dz}=0$	(1)	Gauss’ Law
$\mu\alpha=\frac{dH}{dy}-\frac{dG}{dz}$ $\mu\beta=\frac{dF}{dz}-\frac{dH}{dx}$ $\mu\gamma=\frac{dG}{dx}-\frac{dF}{dy}$	(2)	Equivalent to Gauss’ Law for magnetism
$P=\mu\left(\gamma\frac{dy}{dt}-\beta\frac{dz}{dt}\right)-\frac{dF}{dt}-\frac{d\Psi}{dz}$ $Q=\mu\left(\alpha\frac{dz}{dt}-\gamma\frac{dx}{dt}\right)-\frac{dG}{dt}-\frac{d\Psi}{dy}$ $R=\mu\left(\beta\frac{dx}{dt}-\alpha\frac{dy}{dt}\right)-\frac{dH}{dt}-\frac{d\Psi}{dz}$	(3)	Faraday’s Law (with the Lorentz Force and Poisson’s Law)
$\frac{d\gamma}{dy}-\frac{d\beta}{dz}=4\pi p'$ $\quad p'=p+\frac{df}{dt}$ $\frac{d\alpha}{dz}-\frac{d\gamma}{dx}=4\pi q'$ $\quad q'=q+\frac{dg}{dt}$ $\frac{d\beta}{dx}-\frac{d\alpha}{dy}=4\pi r'$ $\quad r'=r+\frac{dh}{dt}$	(4)	Ampère-Maxwell Law
$P=-\xi p \quad Q=-\xi q \quad R=-\xi r$		Ohm’s Law
$P=kf \quad Q=kg \quad R=kh$		The electric elasticity equation (**E** = **D**/ε)
$\frac{de}{dt}+\frac{dp}{dx}+\frac{dq}{dy}+\frac{dr}{dz}=0$		Continuity of charge

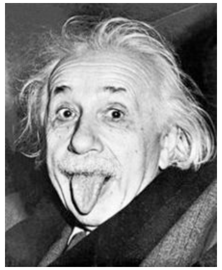
Maxwell’s Eqns simplified w/ rotational symmetry

$$\nabla\cdot\mathbf{D}=\rho_V$$
$$\nabla\cdot\mathbf{B}=0$$
$$\nabla\times\mathbf{E}=-\frac{\partial\mathbf{B}}{\partial t}$$
$$\nabla\times\mathbf{H}=\frac{\partial\mathbf{D}}{\partial t}+\mathbf{J}$$



Maxwell’s Eqns further simplified w/ symmetry of special relativity

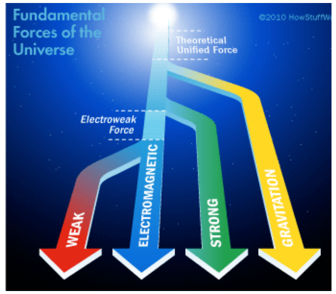
$$\varepsilon^{uvk\lambda}\partial_v F_{k\lambda}=0$$
$$\partial_v F^{uV}=\frac{4\pi}{c}j^u$$



Standard Model w/ Yang-Mills theory and US(3) symmetry

$$\mathcal{L}_{\mathrm{gf}}=-\frac{1}{2}\mathrm{Tr}(F^2)=-\frac{1}{4}F^{a\mu\nu}F^a_{\mu\nu}$$

Unification of fundamental forces?



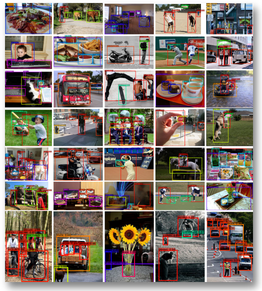
1861

1910s

1970s

?

A "standardized formalism" of ML



Data examples

Type-2 diabetes is 90% more common than type-1

Constraints

Rewards

Auxiliary agents

Adversaries

Imitation

$$\min_{q,\theta} \; -\mathbb{H} + \mathbb{D} - \mathbb{E}$$

Uncertainty Divergence Experience

- Panoramically learn from all types of experience
- Subsumes many existing algorithms as special cases

Will discuss in later in the class

Large Language Models

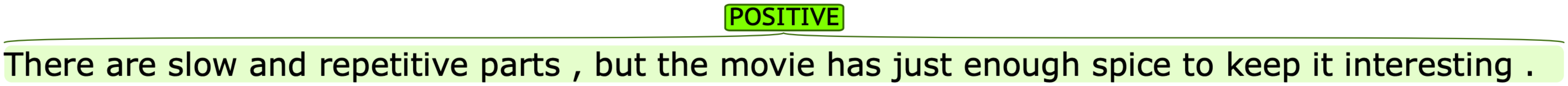
Natural Language Processing (NLP): Before 2017

Automated understanding and generation of natural language

Core NLP tasks handled by respective machine learning models, e.g.,:

Named Entity Recognition

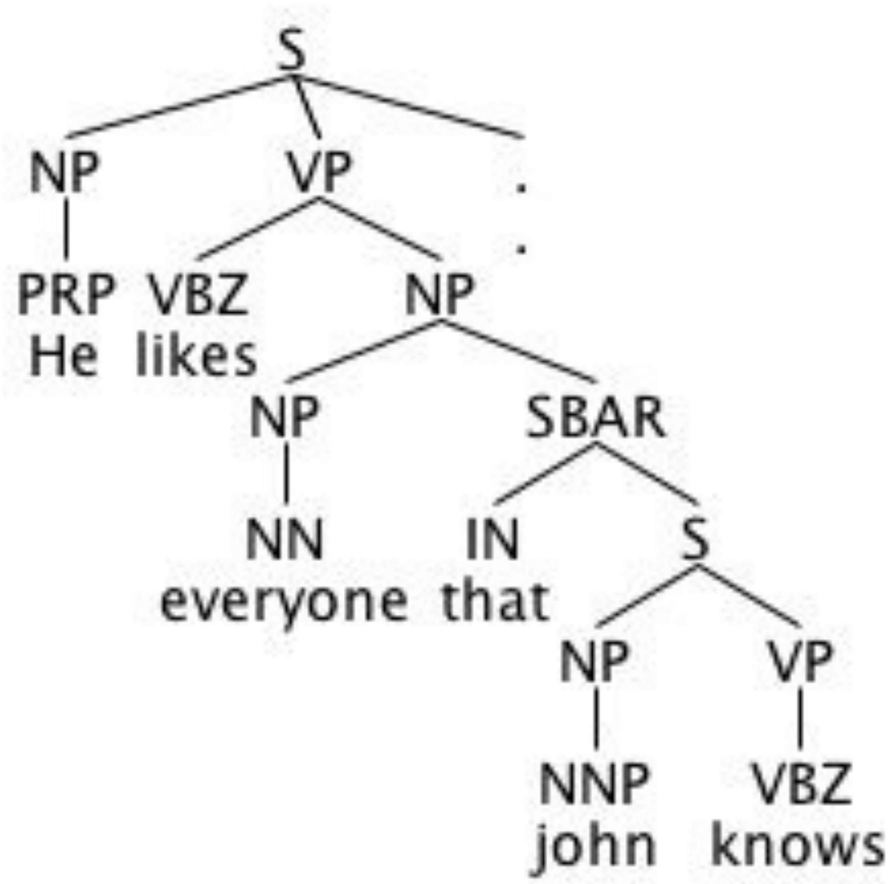
Sentiment Analysis



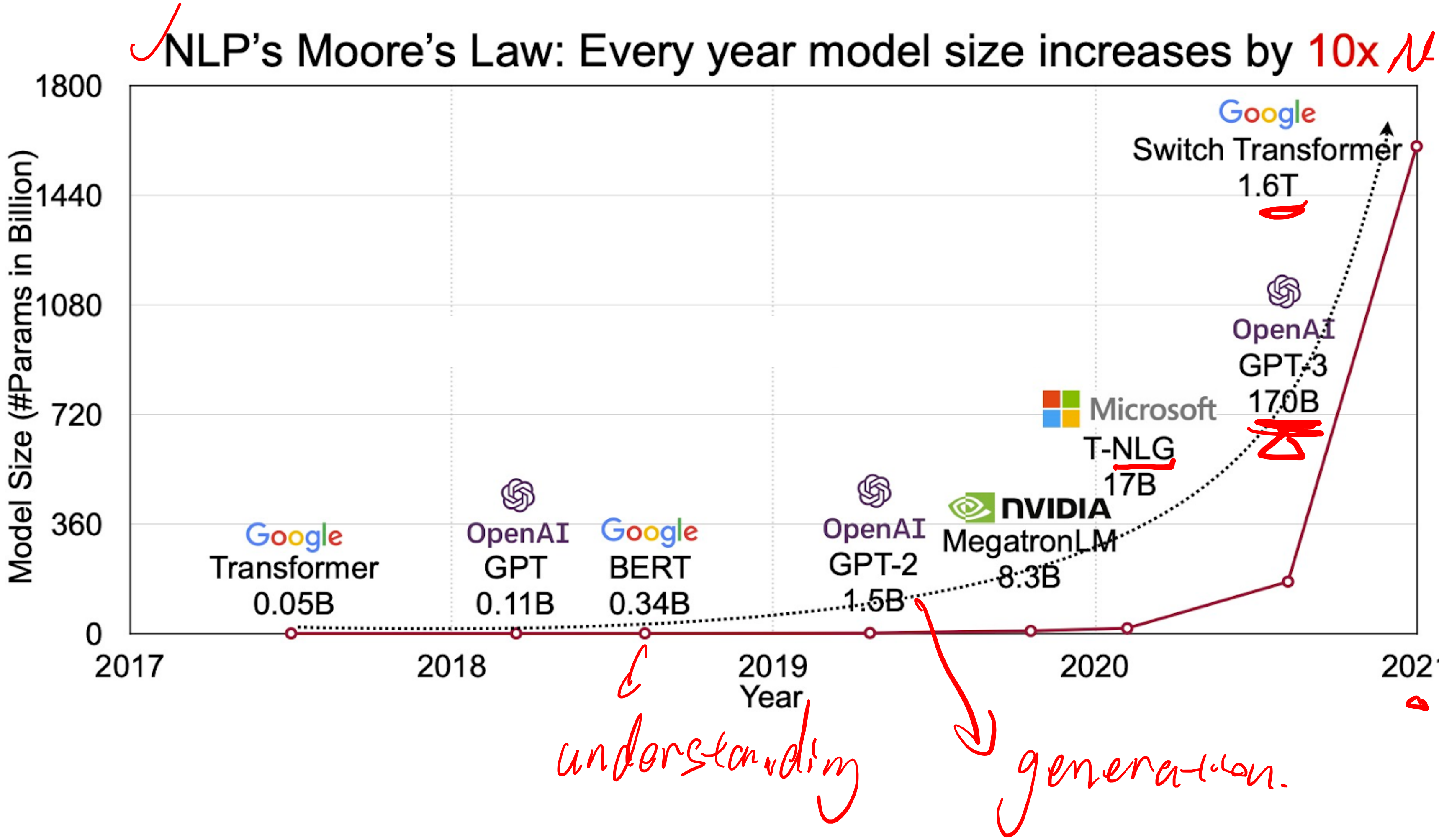
Natural Language Processing (NLP): Before 2017

Automated understanding and generation of natural language

Hand annotation of linguistic structures
(e.g., the Penn Treebank, 1990s)



NLP breakthrough with large language models, since 2017



Mixture-of-experts

MoE

GPT-4

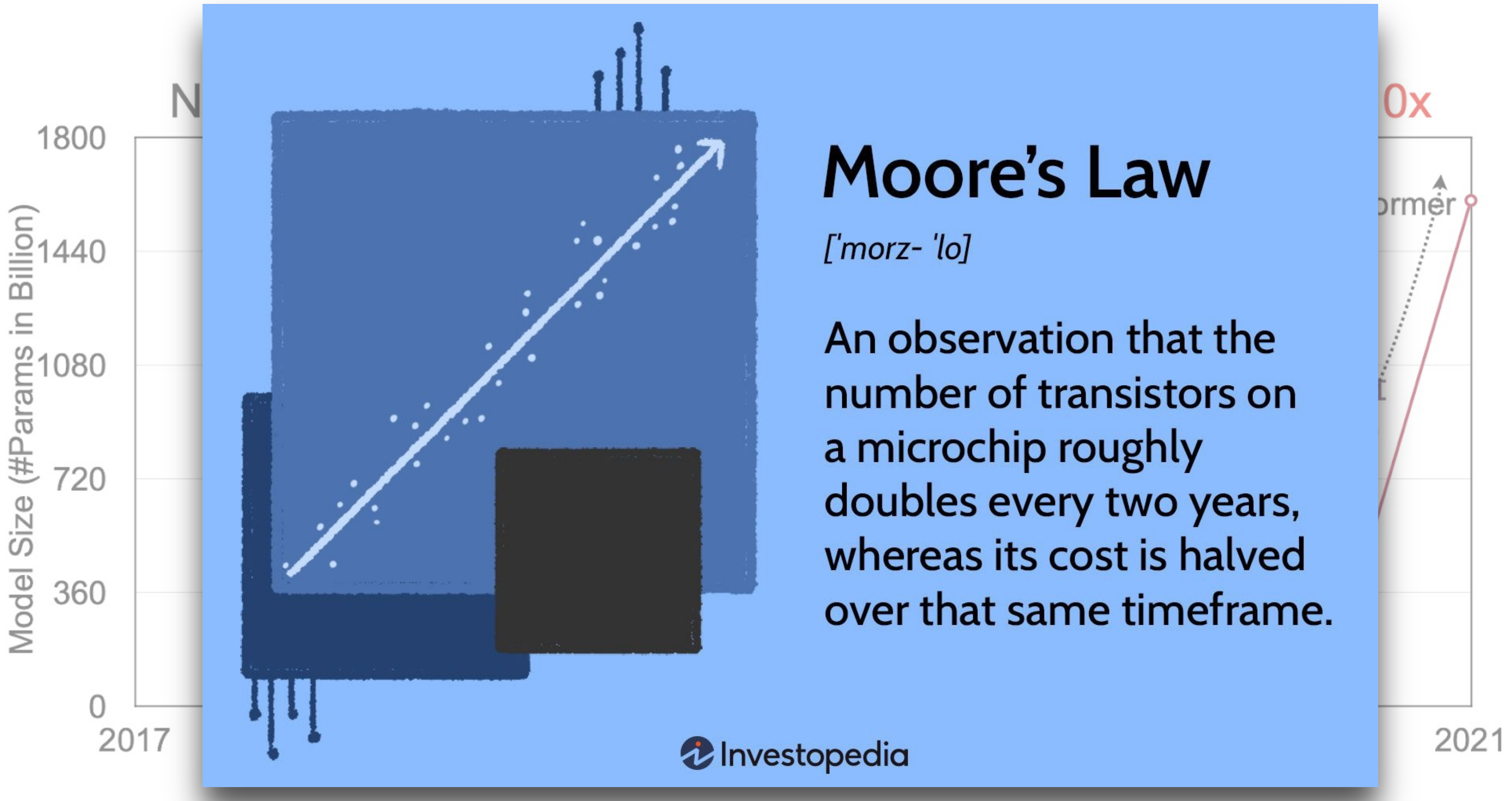
16 Transformers

→ 2023

100B

Figure credit: https://indiaai.gov.in/article/the-future-of-large-language-models-llms-strategy-opportunities-and-challenges

NLP breakthrough with large language models, since 2017



microchip industry

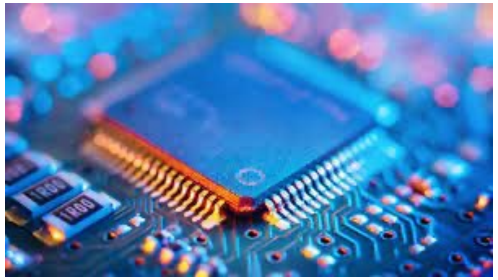


Figure credit: Investopedia

NLP breakthrough with large language models, since 2017

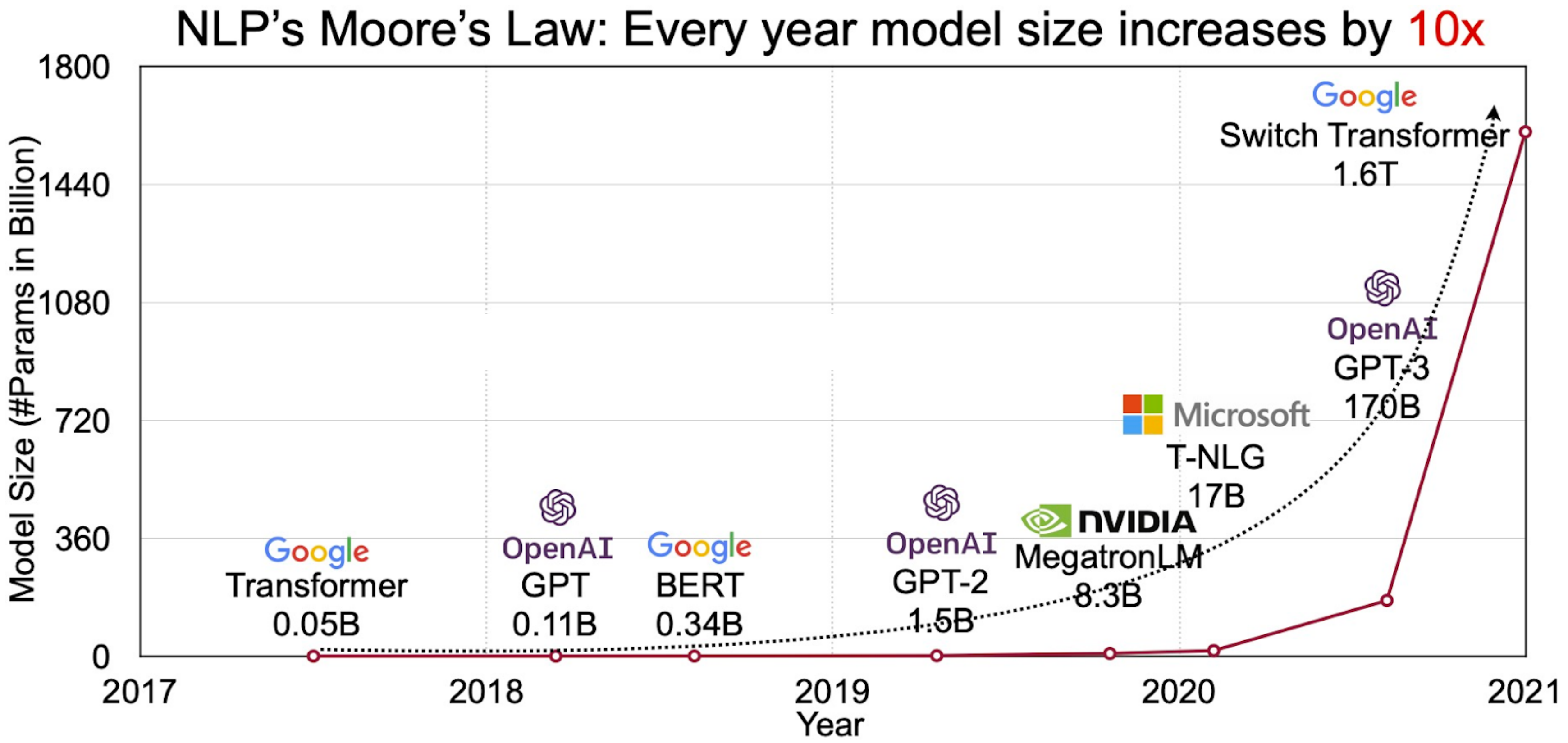


Figure credit: https://indiaai.gov.in/article/the-future-of-large-language-models-llms-strategy-opportunities-and-challenges

NLP breakthrough with large language models, since 2017

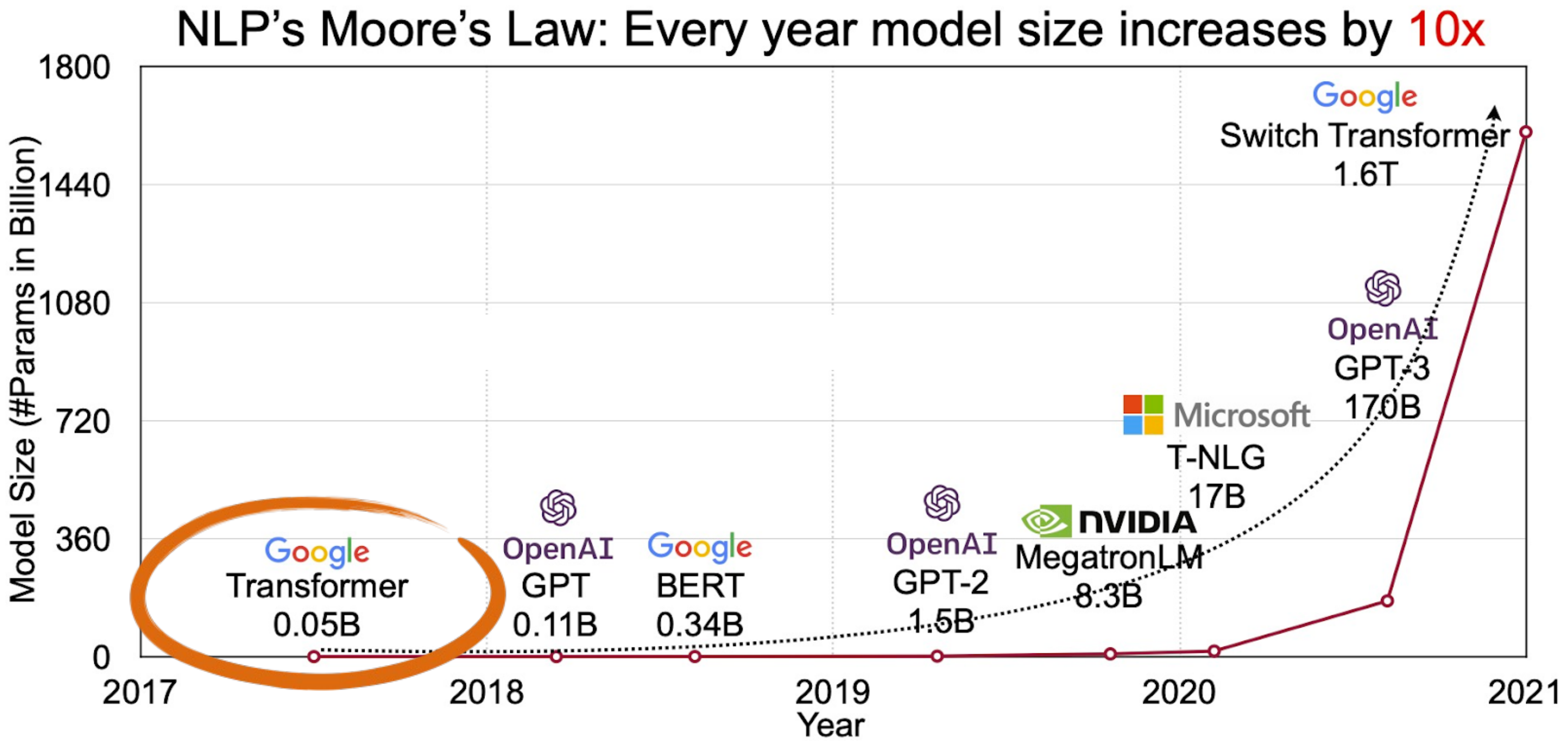
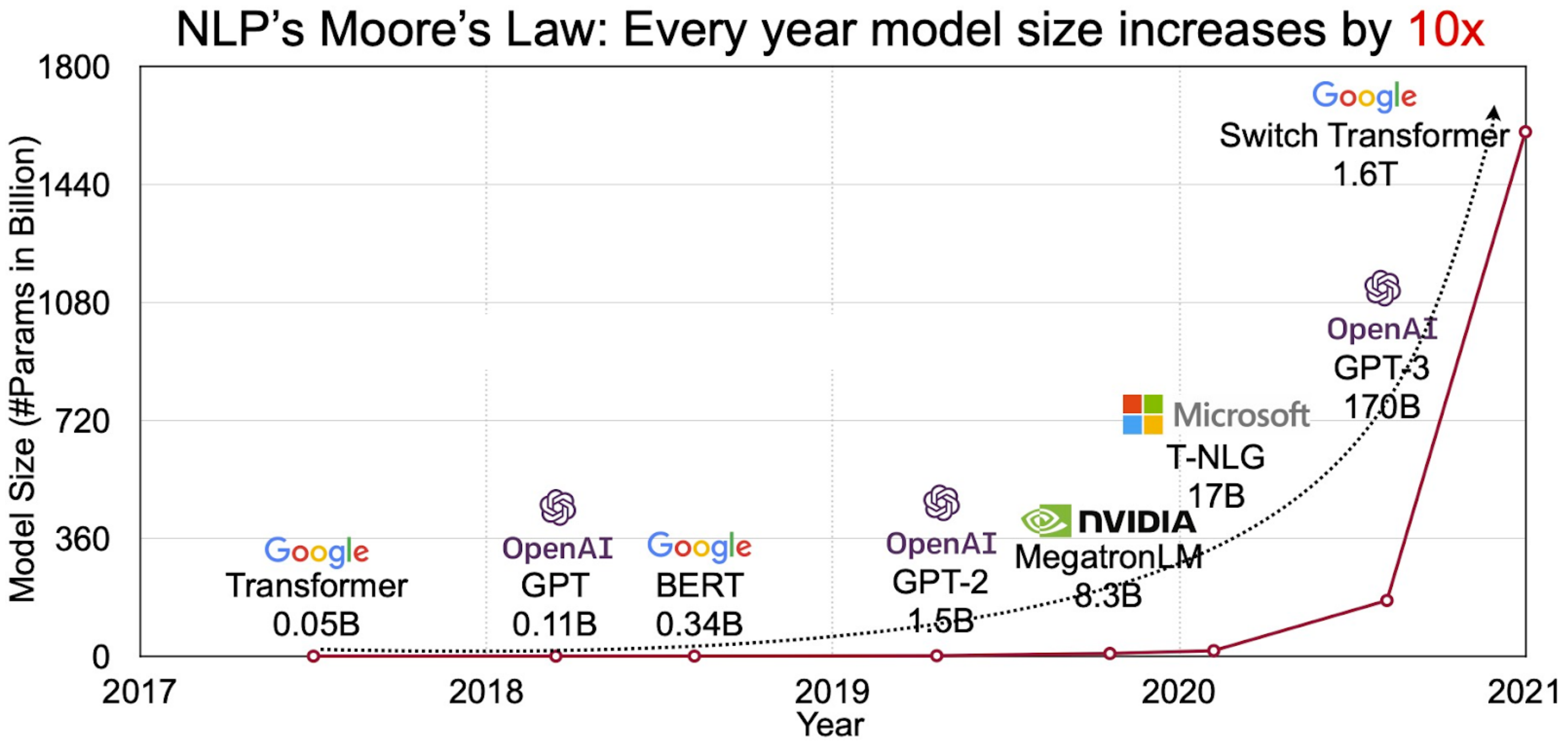


Figure credit: https://indiaai.gov.in/article/the-future-of-large-language-models-llms-strategy-opportunities-and-challenges

NLP breakthrough with large language models, since 2017



ChatGPT

2022

Figure credit: https://indiaai.gov.in/article/the-future-of-large-language-models-llms-strategy-opportunities-and-challenges

What is a language model?

What is a language model?

Sure, here's an example that uses an analogy to explain what a language model is:

Imagine you're playing a game of Mad Libs. You know the general story, but some of the words are missing and you need to fill them in to make it complete. A language model is kind of like the person who's playing Mad Libs with you, but they're really good at guessing what words you're going to pick based on the context and the words you've already chosen.

Language Model 101

S = Imagine you're playing a game of Mad Libs.

Language Model 101

S = Imagine you're playing a game of Mad $\boxed{\text{Libs.}}$

Previous words
(Context)

Word being
predicted

Next word prediction

Language Model 101

S = Imagine you're playing a game of Mad $\boxed{\text{Libs.}}$

Previous words (Context)

Word being predicted

Next word prediction

$$P(w_i | w_1, \ldots, w_{i-1})$$

Language Model 101

inference

Variational inference

S = Imagine you're playing a game of Mad $\boxed{\text{Libs.}}$

Previous words (Context)

Word being predicted

Next word prediction

$$P(w_i|w_1, \ldots, w_{i-1})$$

generate

I ___

decoding

sampled decoding

greedy

beam search

top-k

top-p

Figure credit: https://lena-voita.github.io/nlp_course/language_modeling.html

Language Model 101

$$P(w_i | w_1, \ldots, w_{i-1})$$

Implementations (model architecture):

N-grams

LSTM

Recurrent Neural Networks (RNNs)

Transformer

…

Language Model 101

$$P(w_i | w_1, \ldots, w_{i-1})$$

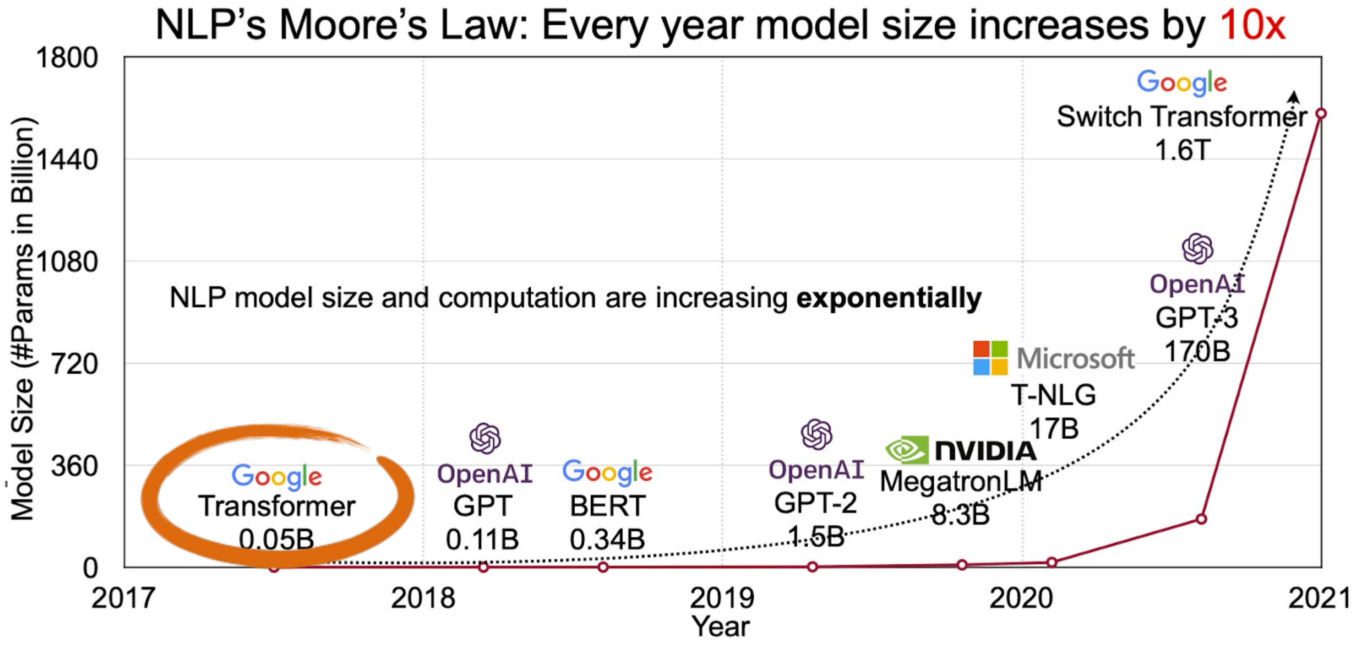
Implementations (model archi

N-grams

Recurrent Neural Networks (RNNs)

Transformer

…



Language Model 101: Transformer

$$P(w_i | w_1, \ldots, w_{i-1})$$

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

2017

Language Model 101: Transformer

$$P(w_i | \underline{w_1, \ldots, w_{i-1}})$$

Attention Is All You Nee

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

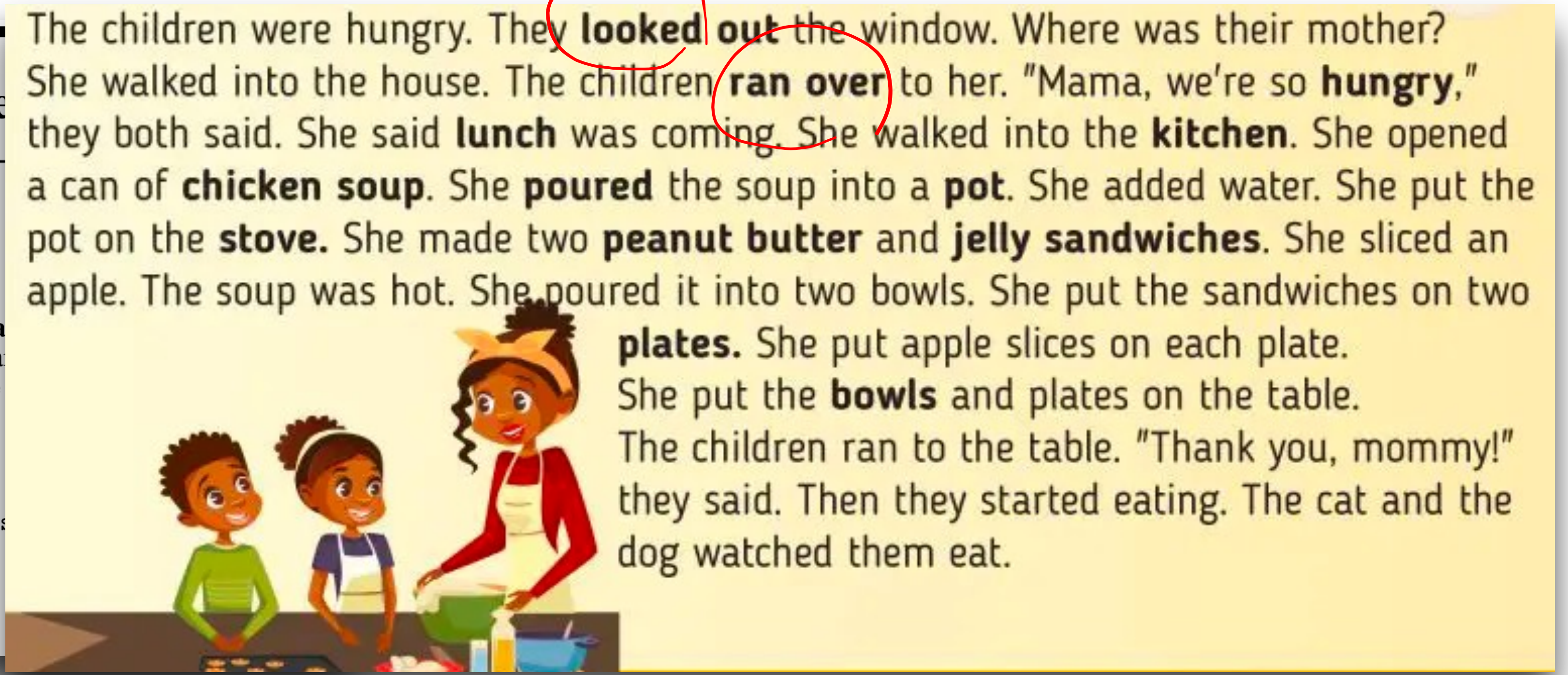
Niki Parma
Google Resea
nikip@google

Llion Jones*
Google Research
llion@google.com

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

lukas

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

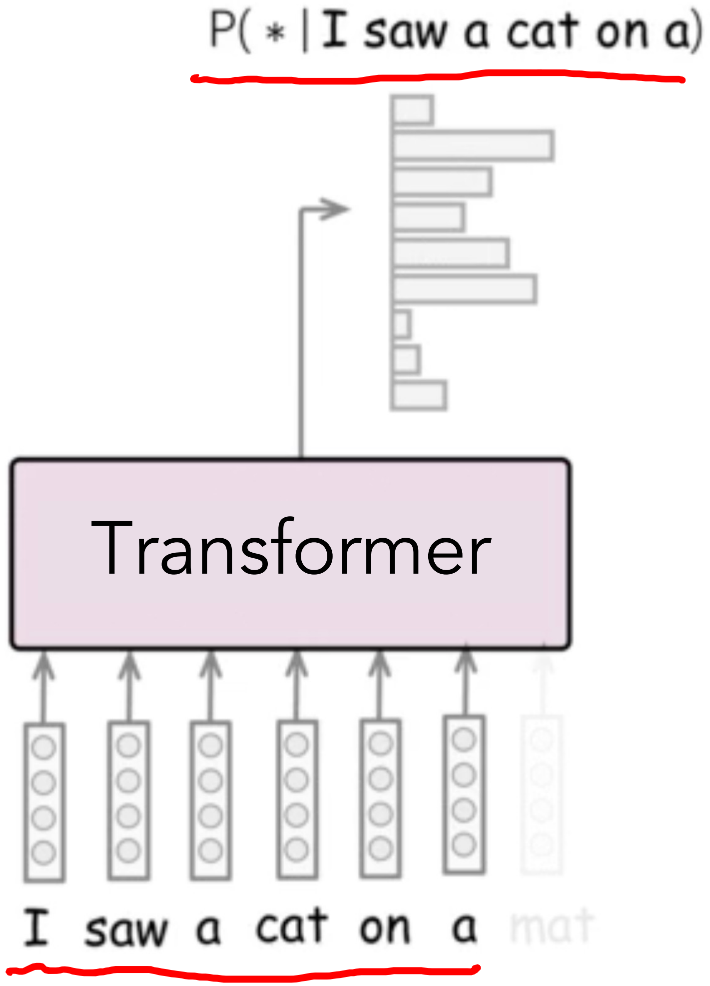


The children were hungry. They **looked out** the window. Where was their mother? She walked into the house. The children **ran over** to her. "Mama, we're so **hungry**," they both said. She said **lunch** was coming. She walked into the **kitchen**. She opened a can of **chicken soup**. She **poured** the soup into a **pot**. She added water. She put the pot on the **stove.** She made two **peanut butter** and **jelly sandwiches**. She sliced an apple. The soup was hot. She poured it into two bowls. She put the sandwiches on two **plates.** She put apple slices on each plate. She put the **bowls** and plates on the table. The children ran to the table. "Thank you, mommy!" they said. Then they started eating. The cat and the dog watched them eat.

2017

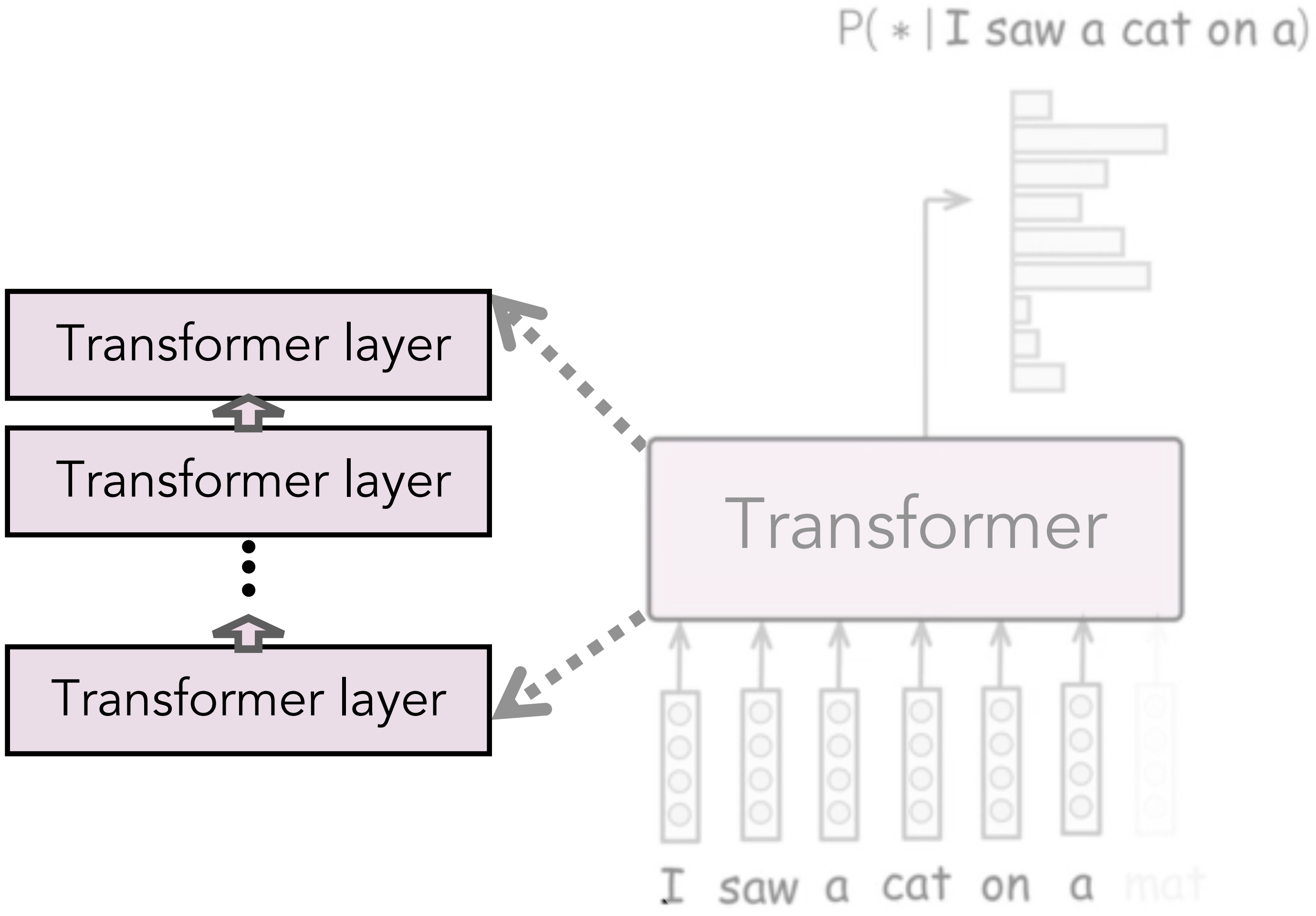
Language Model 101: Transformer

$$P(w_i | w_1, \ldots, w_{i-1})$$



Language Model 101: Transformer

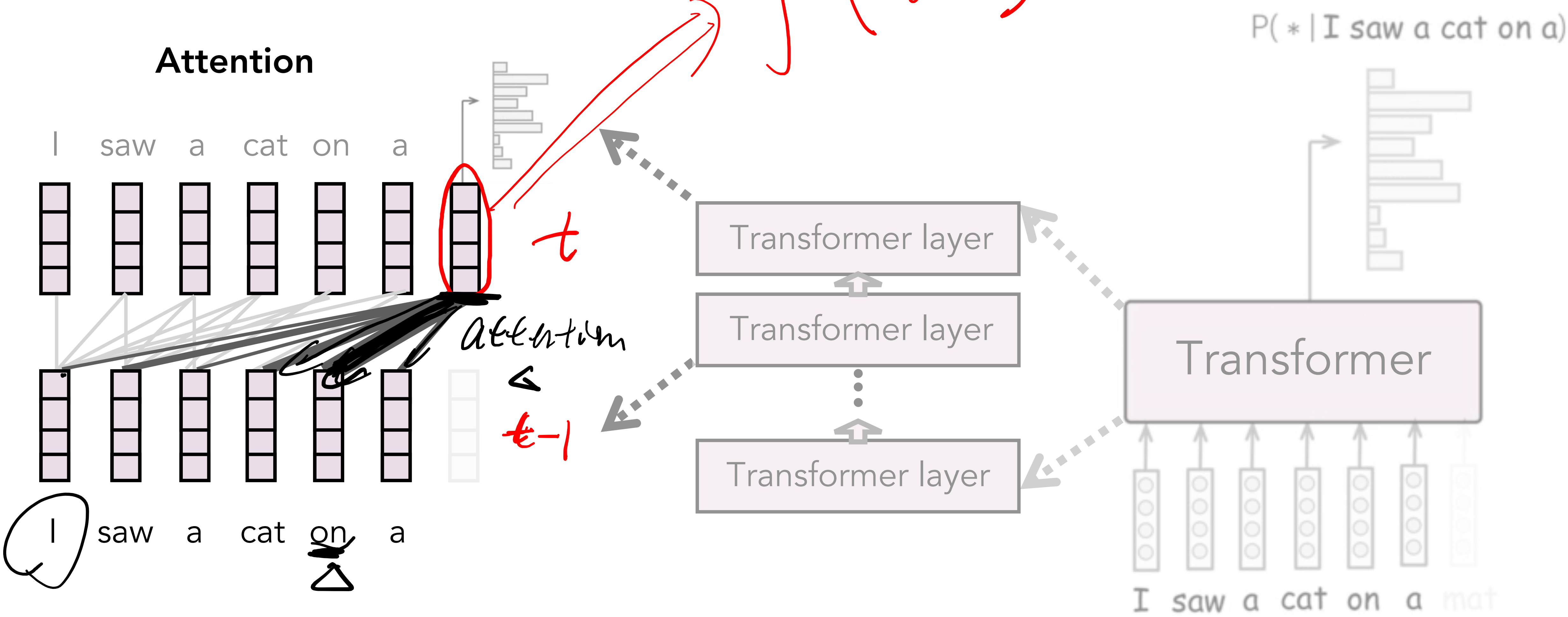
$$P(w_i | w_1, \ldots, w_{i-1})$$



Language Model 101: Transformer

$$P(w_i | w_1, \ldots, w_{i-1})$$

$f(z_i^{(t-1)})$



Language models: Summary so far

- Which components of LMs have we talked about so far?

ML solution:

$$\min_{\theta} \mathcal{L}(\theta, \mathcal{E})$$

Language models: Summary so far

- So far, we've talked about the model architectures and inference of LMs
  - Model architecture: Transformers
  - Inference: next word prediction (sampling tokens at each step)
- Next: training of LMs

$P(w_i \mid w_1, \ldots, w_{i-1})$

ML solution:

$$\min_\theta \mathcal{L}(\theta, \mathcal{E})$$

Self-Supervised Learning

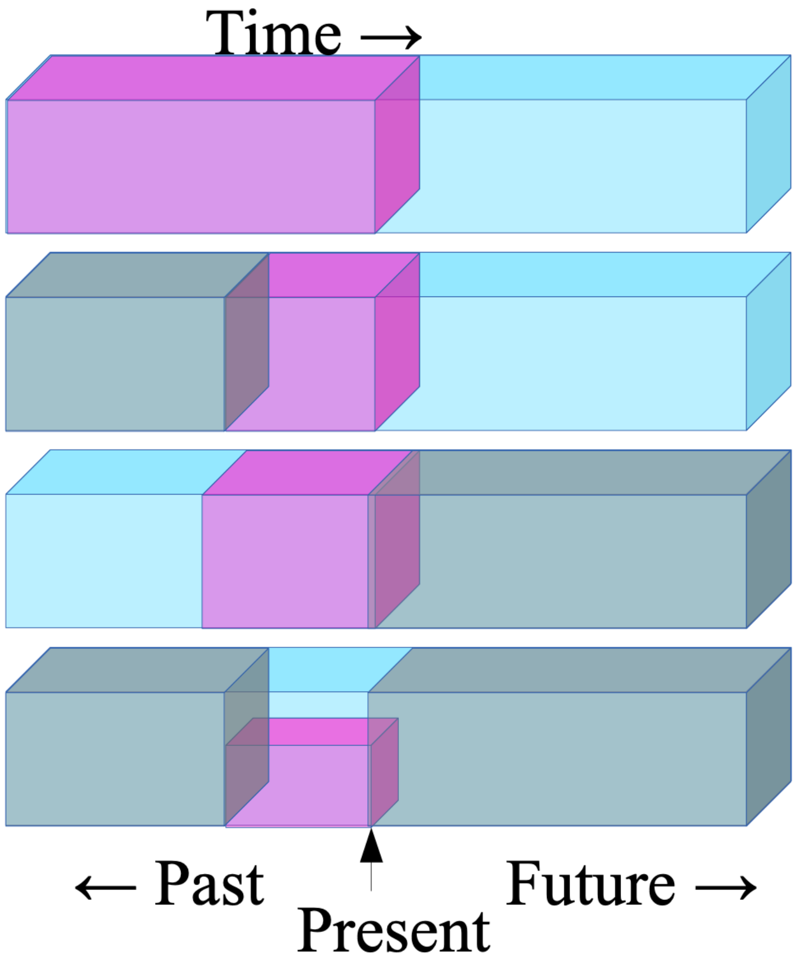
Terminology

- Supervised Learning
- Semi-supervised Learning
- Weakly-supervised Learning
- Self-supervised Learning
- Unsupervised Learning

- All need some forms of supervision, or experience

Self-Supervised Learning: Examples

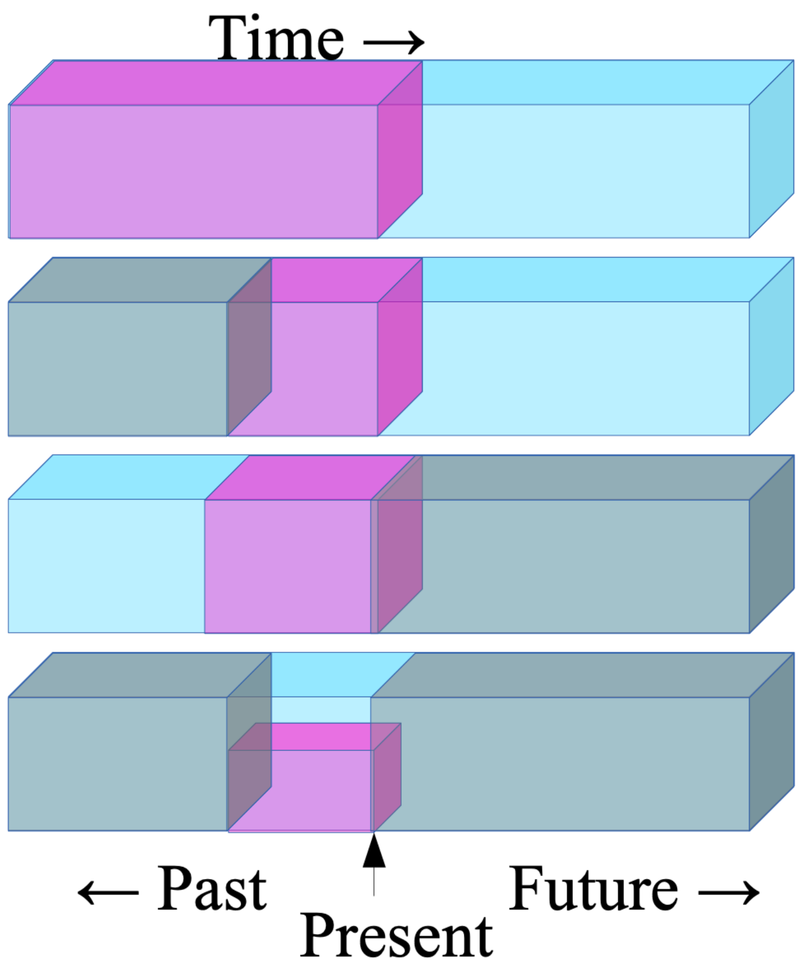
- **Predict any part of the input from any other part.**
- **Predict the future from the past.**
- **Predict the future from the recent past.**
- **Predict the past from the present.**
- **Predict the top from the bottom.**



[Courtesy: Lecun "Self-supervised Learning"]

Self-Supervised Learning: Examples

- **Predict any part of the input from any other part.**
- **Predict the future from the past.**
- **Predict the future from the recent past.**
- **Predict the past from the present.**
- **Predict the top from the bottom.**
- **Predict the occluded from the visible**
- **Pretend there is a part of the input you don't know and predict that.**



[Courtesy: Lecun "Self-supervised Learning"]

Self-Supervised Learning: Motivation (I)

- **Our brains do this all the time**

- **Filling in the visual field at the retinal blind spot**
- **Filling in occluded images, missing segments in speech**
- **Predicting the state of the world from partial (textual) descriptions**
- **Predicting the consequences of our actions**
- **Predicting the sequence of actions leading to a result**

- **Predicting any part of the past, present or future percepts from whatever information is available.**

[Courtesy: Lecun “Self-supervised Learning”]

Self-Supervised Learning: Motivation (I)

- Successfully learning to predict everything from everything else would result in the accumulation of lots of background knowledge about how the world works

- The model is forced to learn what we really care about, e.g. a semantic representation, in order to solve the prediction problem

[Courtesy: Lecun “Self-supervised Learning”]

[Courtesy: Zisserman “Self-supervised Learning”]

Self-Supervised Learning: Motivation (II)

- The machine predicts any part of its input from any observed part
  - **A lot of** supervision signals in each data instance

- Untapped/availability of vast numbers of unlabeled text/images/videos..
  - Facebook: one billion images uploaded per day
  - 300 hours of video are uploaded to YouTube every minute

[Courtesy: Zisserman "Self-supervised Learning"]

SSL in Language Models

- Calculates the probability of a sentence:
  - Sentence:

$$\boldsymbol{y} = (y_1, y_2, \dots, y_T)$$

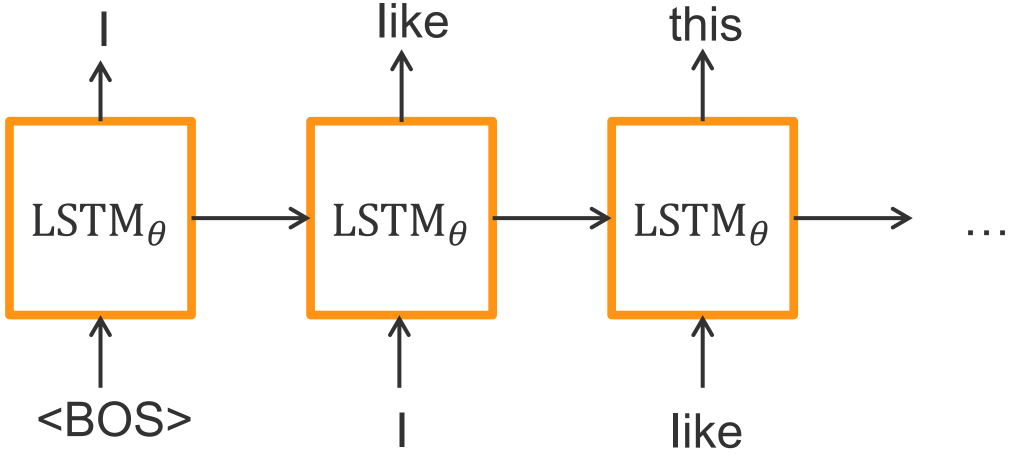
$$p_\theta(\boldsymbol{y}) = \prod_{t=1}^{T} p_\theta(y_t \mid \boldsymbol{y}_{1:t-1})$$

Example:

(I, like, this, …)

$$\cdots p_\theta\,(like \mid I)\, p_\theta(this \mid I, like) \cdots$$

Model: LSTM RNN



SSL in Language Models

- Calculates the probability of a sentence:
  - Sentence:

Example:

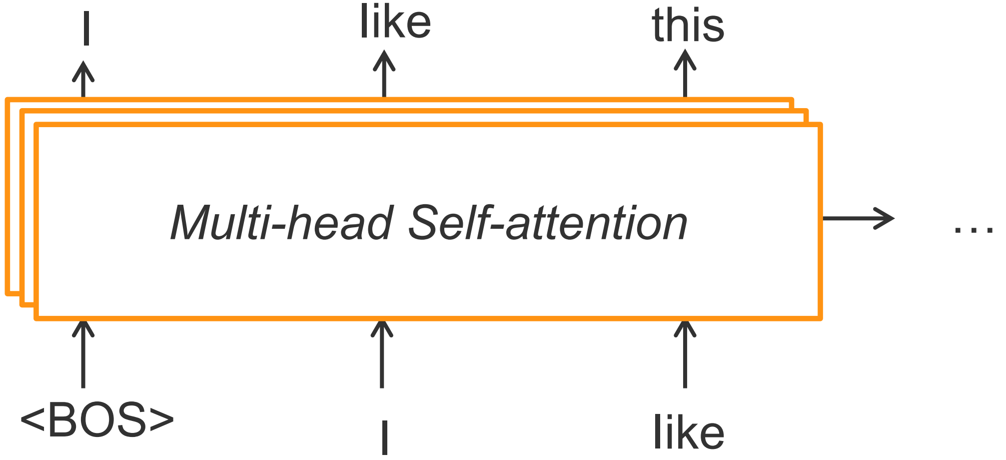
$$\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$$

(I, like, this, …)

$$p_\theta(\boldsymbol{y}) = \prod_{t=1}^{T} p_\theta(y_t \mid \boldsymbol{y}_{1:t-1})$$

$$\cdots p_\theta\,(like \mid I)\, p_\theta(this \mid I, like) \cdots$$

Model: Transformer



SSL in Language Models: Training

- Given data example $\boldsymbol{y}^*$
- Minimizes negative log-likelihood of the data

$$\min_\theta \mathcal{L}_{\mathrm{MLE}} = -\log p_\theta(\boldsymbol{y}^*) = -\prod_{t=1}^{T} p_\theta(y_t^* \mid \boldsymbol{y}_{1:t-1}^*)$$

SSL in Language Models: GPT3

- A Transformer-based LM with 125M to 175B parameters
- Trained on massive text data

Dataset	# Tokens (Billions)
Total	499
Common Crawl (filtered by quality)	410
WebText2	19
Books1	12
Books2	55
Wikipedia	3

Brown et al., 2020 "Language Models Are Few-Shot Learners”

[Table from https://lambdalabs.com/blog/demystifying-gpt-3/]

Other examples of self-supervised learning

- Learning contextual text representations
- Learning image / video representations

Word Embedding

- Conventional word embedding:
  - Word2vec, Glove
  - A pre-trained **matrix**, each row is an embedding vector of a word

	0	1	2	3	4	5	6	7	8	9	...
fox	-0.348680	-0.077720	0.177750	-0.094953	-0.452890	0.237790	0.209440	0.037886	0.035064	0.899010	...
ham	-0.773320	-0.282540	0.580760	0.841480	0.258540	0.585210	-0.021890	-0.463680	0.139070	0.658720	...
brown	-0.374120	-0.076264	0.109260	0.186620	0.029943	0.182700	-0.631980	0.133060	-0.128980	0.603430	...
beautiful	0.171200	0.534390	-0.348540	-0.097234	0.101800	-0.170860	0.295650	-0.041816	-0.516550	2.117200	...
jumps	-0.334840	0.215990	-0.350440	-0.260020	0.411070	0.154010	-0.386110	0.206380	0.386700	1.460500	...
eggs	-0.417810	-0.035192	-0.126150	-0.215930	-0.669740	0.513250	-0.797090	-0.068611	0.634660	1.256300	...
beans	-0.423290	-0.264500	0.200870	0.082187	0.066944	1.027600	-0.989140	-0.259950	0.145960	0.766450	...
sky	0.312550	-0.303080	0.019587	-0.354940	0.100180	-0.141530	-0.514270	0.886110	-0.530540	1.556600	...
bacon	-0.430730	-0.016025	0.484620	0.101390	-0.299200	0.761820	-0.353130	-0.325290	0.156730	0.873210	...
breakfast	0.073378	0.227670	0.208420	-0.456790	-0.078219	0.601960	-0.024494	-0.467980	0.054627	2.283700	...
toast	0.130740	-0.193730	0.253270	0.090102	-0.272580	-0.030571	0.096945	-0.115060	0.484000	0.848380	...
today	-0.156570	0.594890	-0.031445	-0.077586	0.278630	-0.509210	-0.066350	-0.081890	-0.047986	2.803600	...
blue	0.129450	0.036518	0.032298	-0.060034	0.399840	-0.103020	-0.507880	0.076630	-0.422920	0.815730	...
green	-0.072368	0.233200	0.137260	-0.156630	0.248440	0.349870	-0.241700	-0.091426	-0.530150	1.341300	...
kings	0.259230	-0.854690	0.360010	-0.642000	0.568530	-0.321420	0.173250	0.133030	-0.089720	1.528600	...
dog	-0.057120	0.052685	0.003026	-0.048517	0.007043	0.041856	-0.024704	-0.039783	0.009614	0.308416	...
sausages	-0.174290	-0.064869	-0.046976	0.287420	-0.128150	0.647630	0.056315	-0.240440	-0.025094	0.502220	...
lazy	-0.353320	-0.299710	-0.176230	-0.321940	-0.385640	0.586110	0.411160	-0.418680	0.073093	1.486500	...
love	0.139490	0.534530	-0.252470	-0.125650	0.048748	0.152440	0.199060	-0.065970	0.128830	2.055900	...
quick	-0.445630	0.191510	-0.249210	0.465900	0.161950	0.212780	-0.046480	0.021170	0.417660	1.686900	...

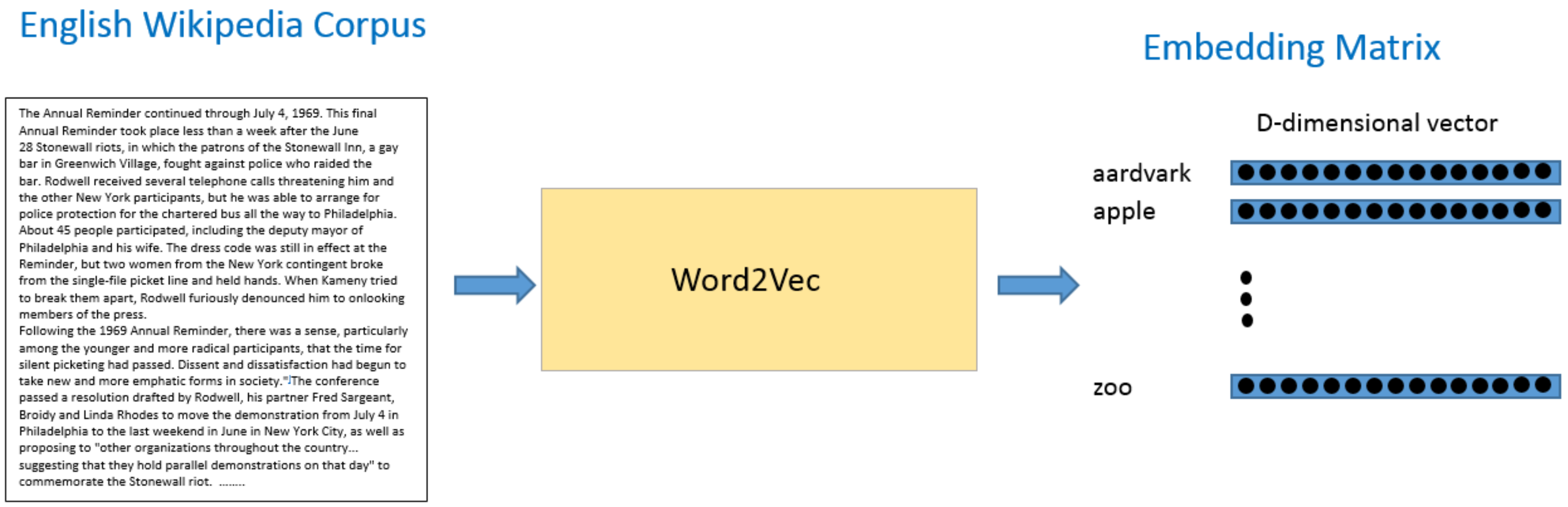
20 rows × 300 columns

[Courtesy: Vaswani, et al., 2017]

Word Embedding

- Conventional word embedding:
  - Word2vec, Glove
  - A pre-trained **matrix**, each row is an embedding vector of a word

	0	1	2	3	4	5	6	7	8	9	...
fox	-0.348680	-0.077720	0.177750	-0.094953	-0.452890	0.237790	0.209440	0.037886	0.035064	0.899010	...
ham	-0.773320	-0.282540	0.580760	0.841480	0.258540	0.585210	-0.021890	-0.463680	0.139070	0.658720	...
brown	-0.374120	-0.076264	0.109260	0.186620	0.029943	0.182700	-0.631980	0.133060	-0.128980	0.603430	...
beautiful	0.171200	0.534390	-0.348540	-0.097234	0.101800	-0.170860	0.295650	-0.041816	-0.516550	2.117200	...
jumps	-0.334840	0.215990	-0.350440	-0.260020	0.411070	0.154010	-0.386110	0.206380	0.386700	1.460500	...
eggs	-0.417810	-0.035192	-0.126150	-0.215930	-0.669740	0.513250	-0.797090	-0.068611	0.634660	1.256300	...
beans	-0.423290	-0.264500	0.200870	0.082187	0.066944	1.027600	-0.989140	-0.259950	0.145960	0.766450	...
sky	0.312550	-0.303080	0.019587	-0.354940	0.100180	-0.141530	-0.514270	0.886110	-0.530540	1.556600	...
bacon	-0.430730	-0.016025	0.484620	0.101390	-0.299200	0.761820	-0.353130	-0.325290	0.156730	0.873210	...
breakfast	0.073378	0.227670	0.208420	-0.456790	-0.078219	0.601960	-0.024494	-0.467980	0.054627	2.283700	...
							945	-0.115060	0.484000	0.848380	...
							350	-0.081890	-0.047986	2.803600	...
							880	0.076630	-0.422920	0.815730	...
							700	-0.091426	-0.530150	1.341300	...
							250	0.133030	-0.089720	1.528600	...
							704	-0.039783	0.009614	0.308416	...
							315	-0.240440	-0.025094	0.502220	...
							160	-0.418680	0.073093	1.486500	...
							060	-0.065970	0.128830	2.055900	...
							480	0.021170	0.417660	1.686900	...



[Image source: Va

Word Embedding

- **Problem**: word embeddings are applied in a context free manner

```
open a bank account        on the river bank

          [0.3, 0.2, -0.8, …]
```

Courtesy: Devlin 2019

Word Embedding

- **Problem**: word embeddings are applied in a context free manner

```
open a bank account        on the river bank

          [0.3, 0.2, -0.8, …]
```

- **Solution**: Train contextual representations on text corpus

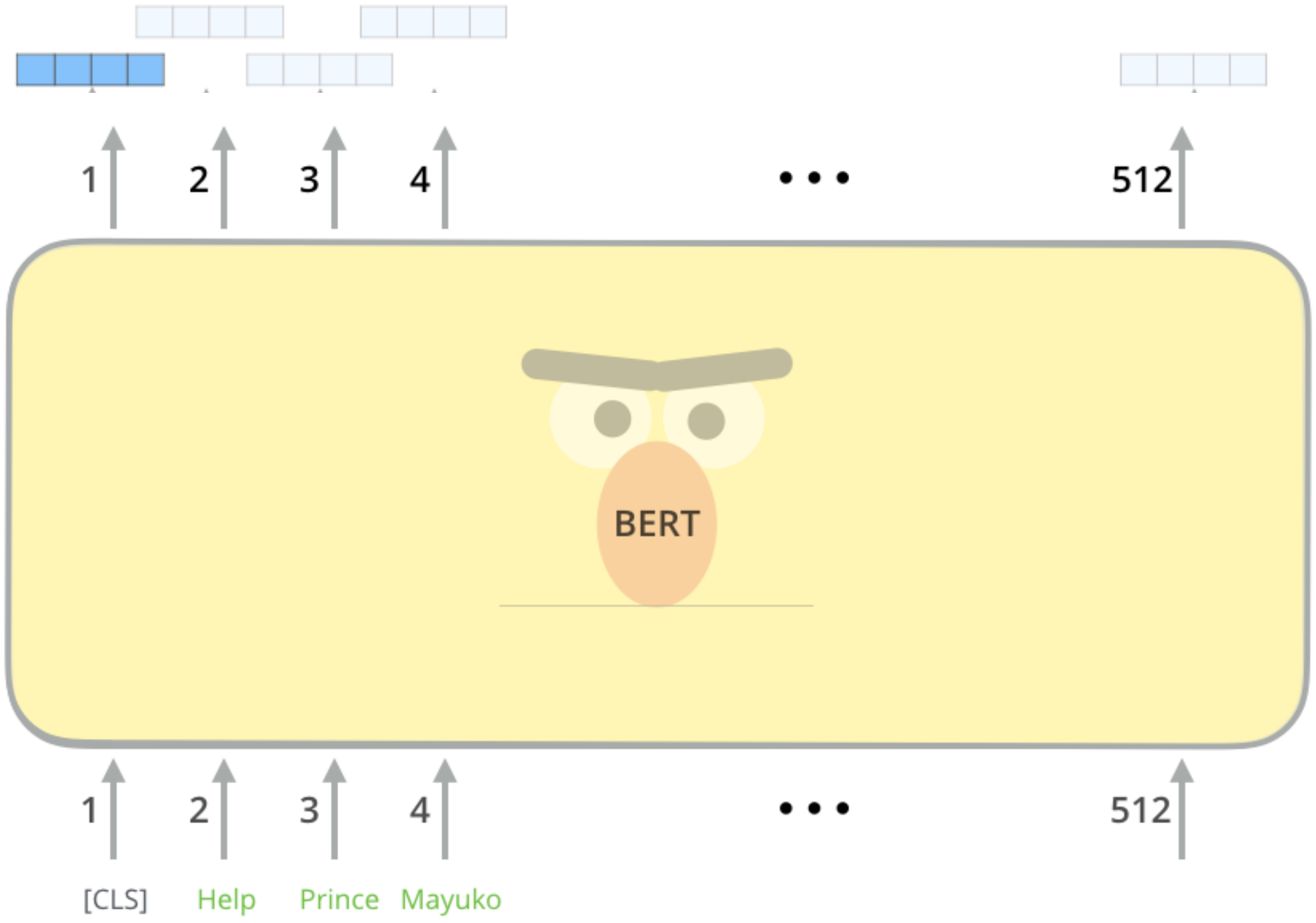
```
[0.9, -0.2, 1.6, …]        [-1.9, -0.4, 0.1, …]

open a bank account         on the river bank
```

Courtesy: Devlin 2019

BERT

- BERT: A bidirectional model to extract contextual word embedding



BERT: Pre-training Procedure

- Dataset:
  - Wikipedia (2.5B words) + a collection of free ebooks (800M words)

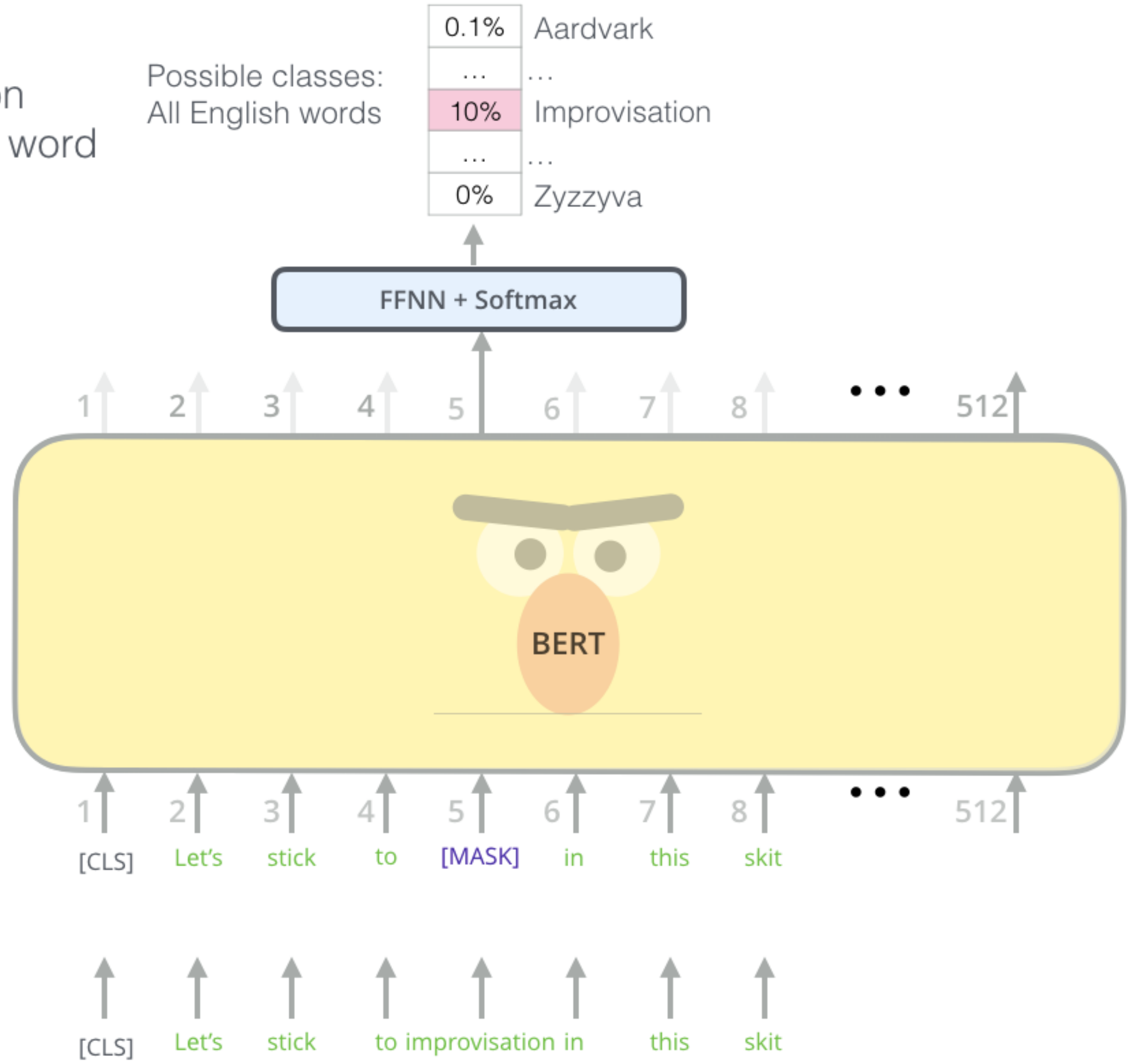
BERT: Pre-training Procedure

- Dataset:
  - Wikipedia (2.5B words) + a collection of free ebooks (800M words)
- Training procedure
  - **masked language model** (masked LM)
    - Masks some percent of words from the input and has to reconstruct those words from context

BERT: Pre-training Procedure

- Masked LM

Use the output of the
masked word's position
to predict the masked word



Randomly mask
15% of tokens

Input

BERT: Pre-training Procedure

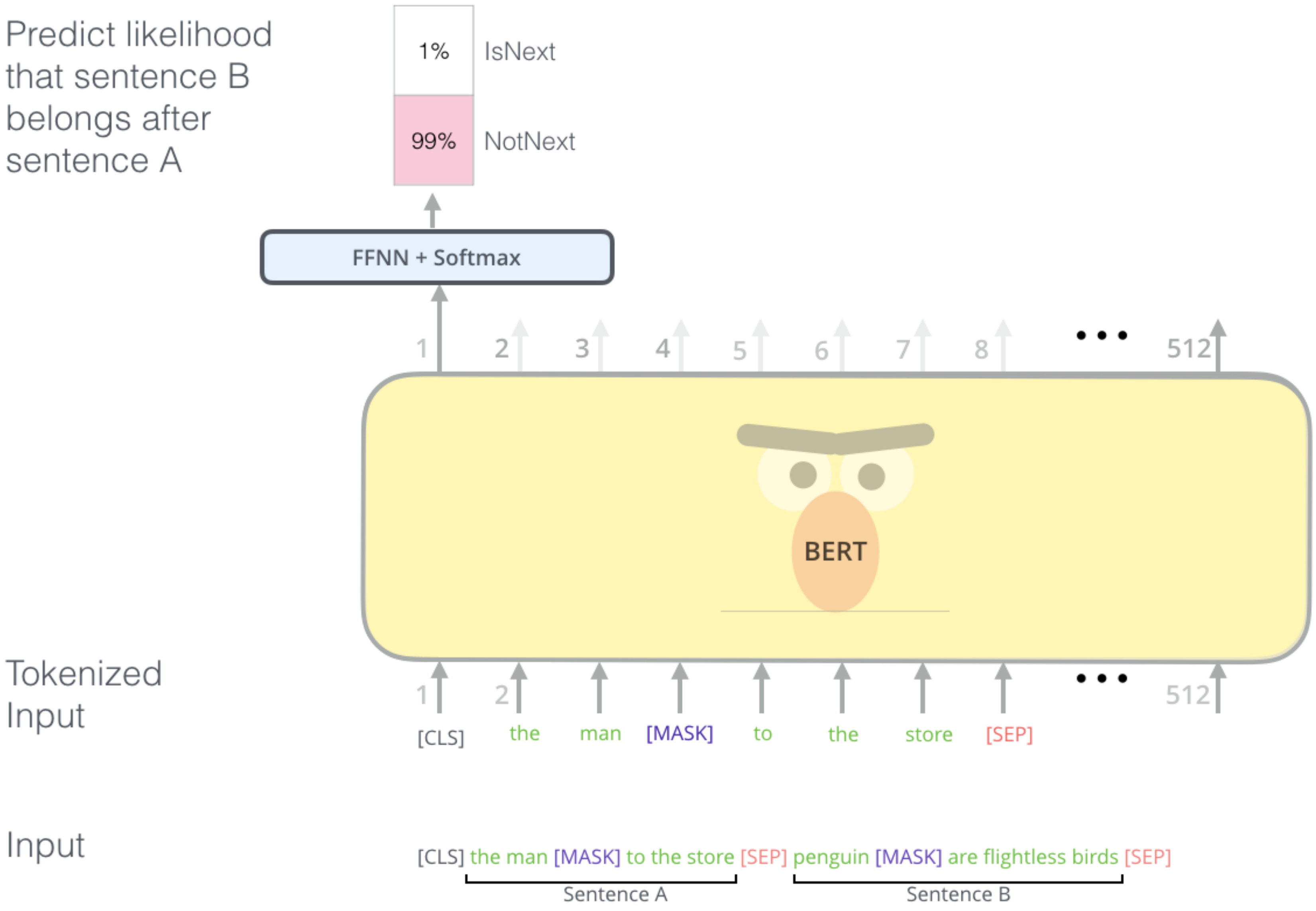
- Masked LM
- 15% masking:
  - Too little masking: Too expensive to train (few supervision signals per example)
  - Too much masking: Not enough context
- Problem: Mask token never seen at fine-tuning
- Solution: don't replace with `[MASK]` 100% of the time. Instead:
- 80% of the time, replace with `[MASK]`
  - `went to the store → went to the [MASK]`
- 10% of the time, replace random word
  - `went to the store → went to the running`
- 10% of the time, keep same
  - `went to the store → went to the store`

BERT: Pre-training Procedure

- Dataset:
  - Wikipedia (2.5B words) + a collection of free ebooks (800M words)
- Training procedure
  - **masked language model** (masked LM)
    - Masks some percent of words from the input and has to reconstruct those words from context
  - **Two-sentence task**
    - To understand relationships between sentences
    - Concatenate two sentences A and B and predict whether B actually comes after A in the original text

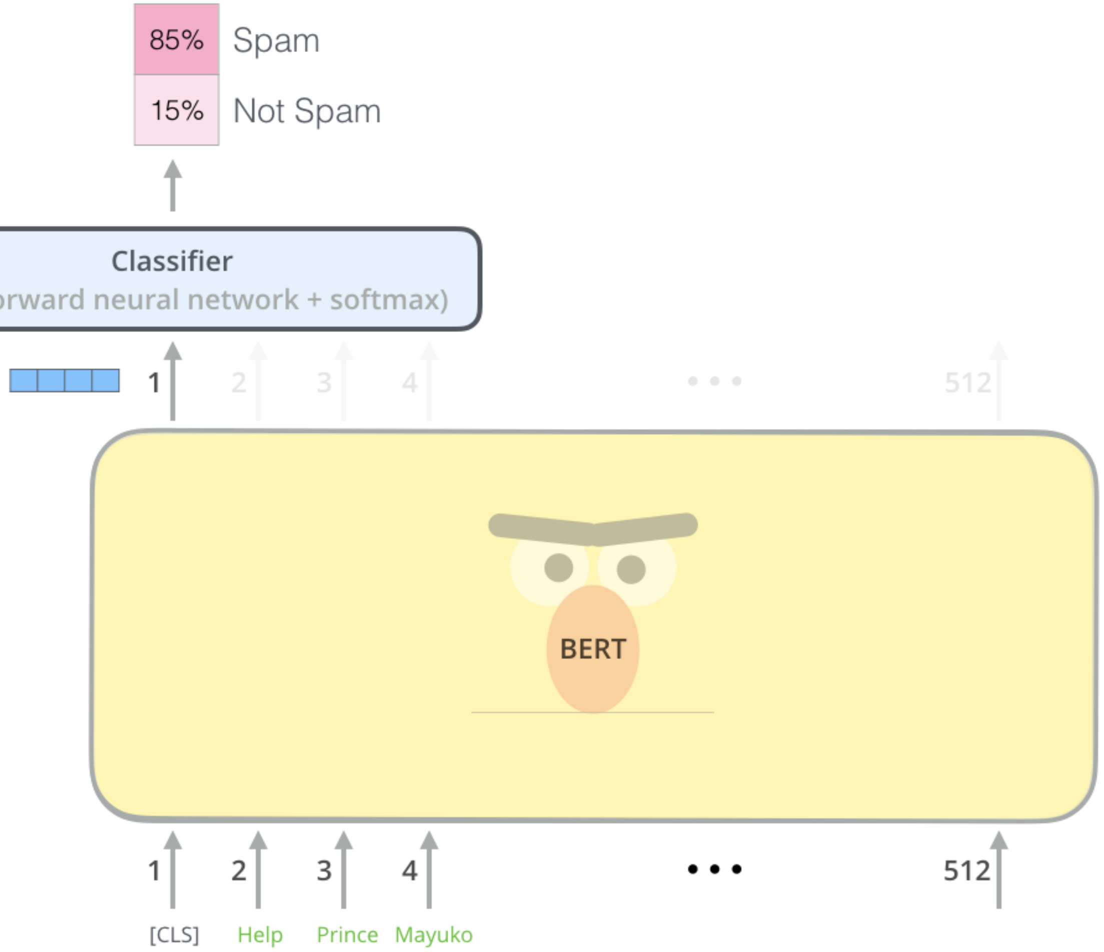
BERT: Pre-training Procedure

- Two sentence task

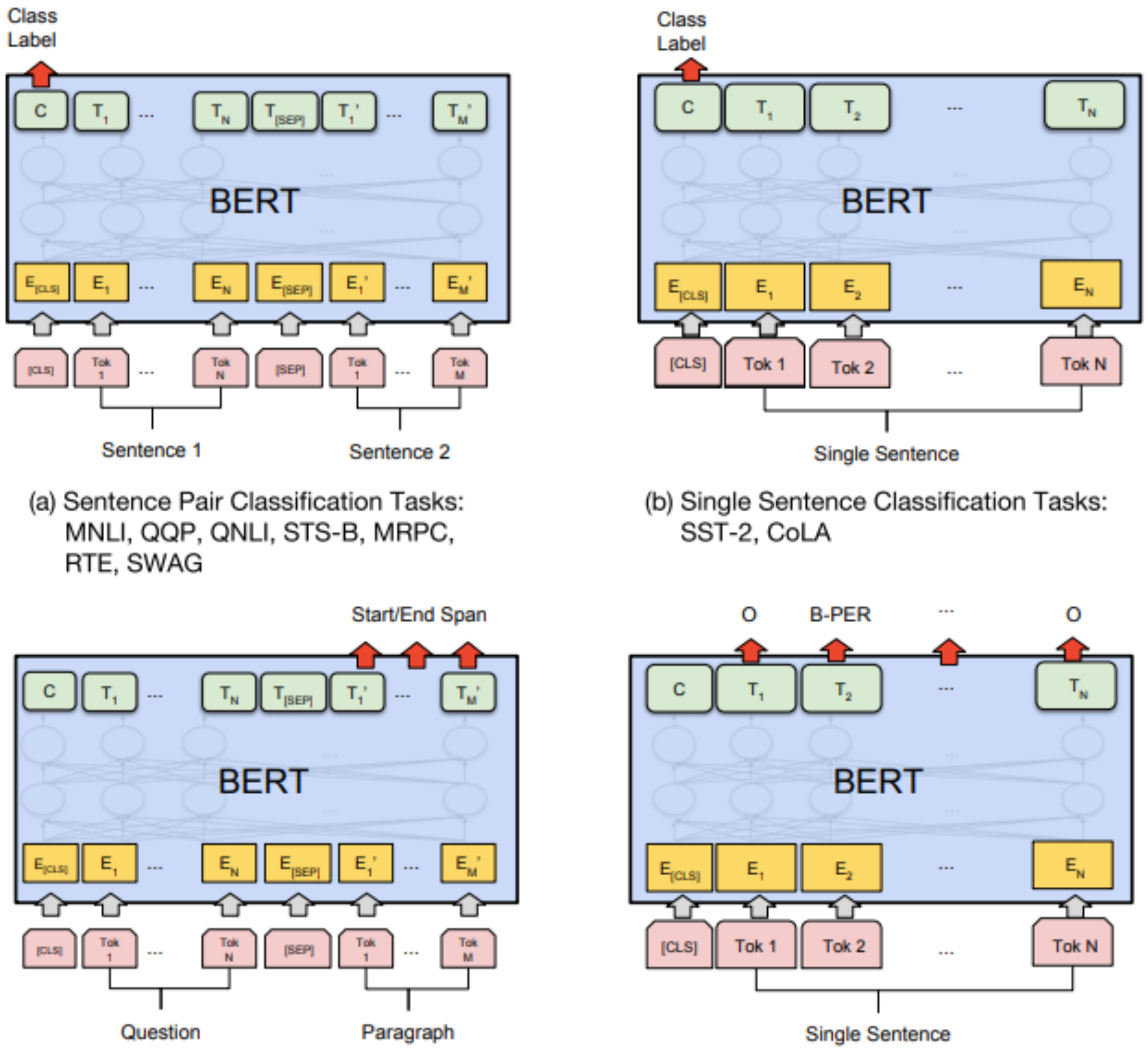


BERT: Downstream Fine-tuning

- Use BERT for sentence classification



BERT: Downstream Fine-tuning



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

BERT Results

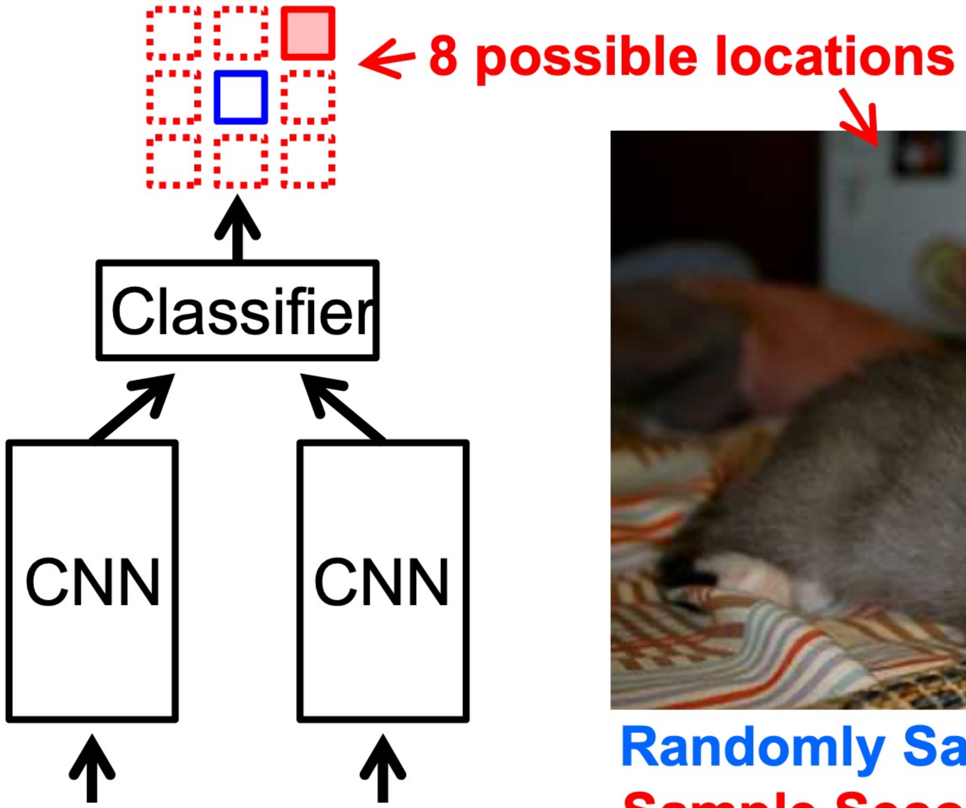
- Huge improvements over SOTA on 12 NLP task

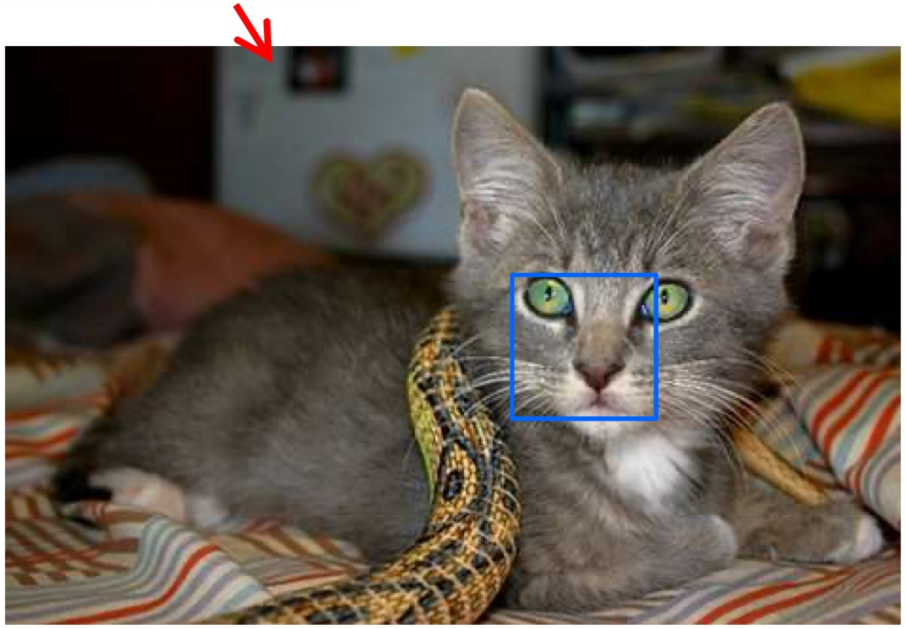
System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

Table 1: GLUE Test results, scored by the GLUE evaluation server. The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set. OpenAI GPT = (L=12, H=768, A=12); $\text{BERT}_{\text{BASE}}$ = (L=12, H=768, A=12); $\text{BERT}_{\text{LARGE}}$ = (L=24, H=1024, A=16). BERT and OpenAI GPT are single-model, single task. All results obtained from https://gluebenchmark.com/leaderboard and https://blog.openai.com/language-unsupervised/.

SSL from Images, EX (I): relative positioning

Train network to predict relative position of two regions in the same image



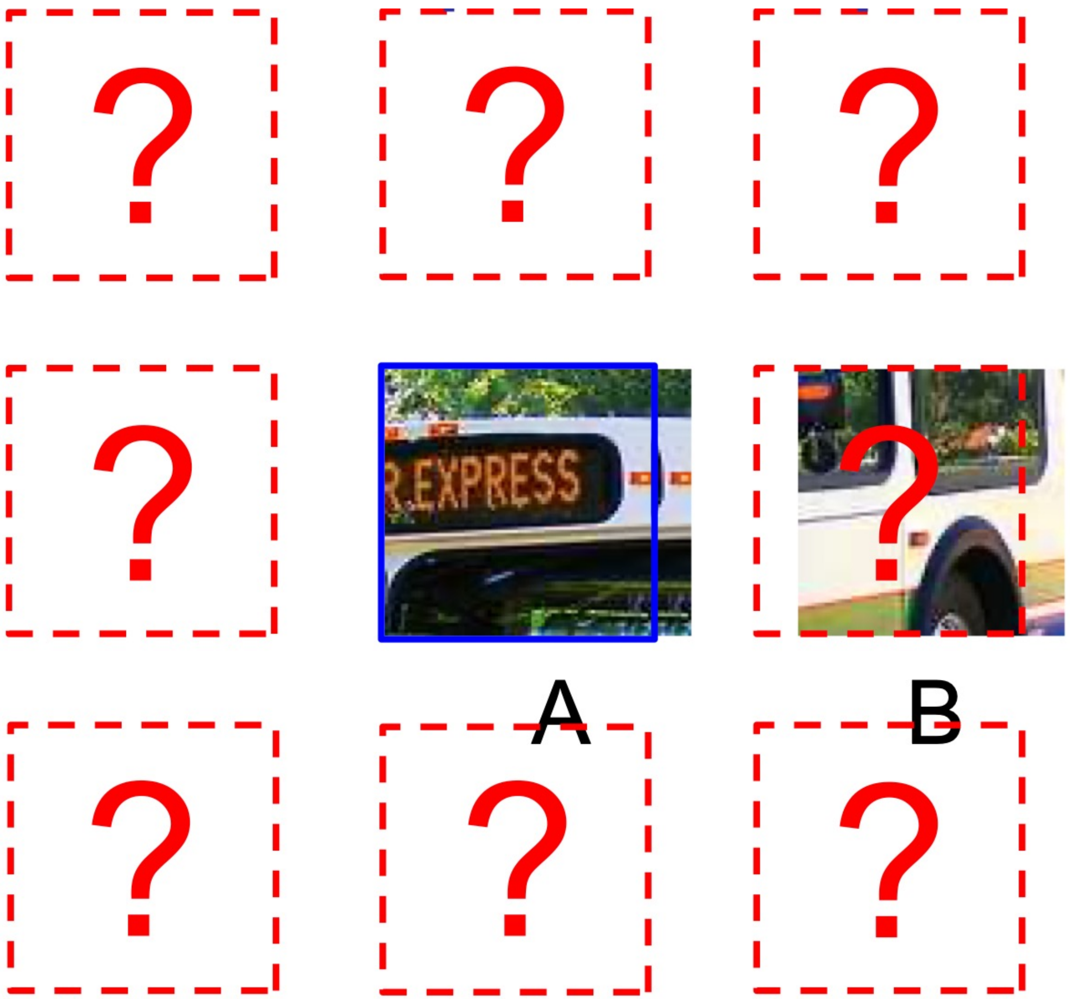


Randomly Sample Patch
Sample Second Patch

Unsupervised visual representation learning by context prediction,
Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

[Courtesy: Zisserman "Self-supervised Learning"]

SSL from Images, EX (I): relative positioning



Unsupervised visual representation learning by context prediction,
Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

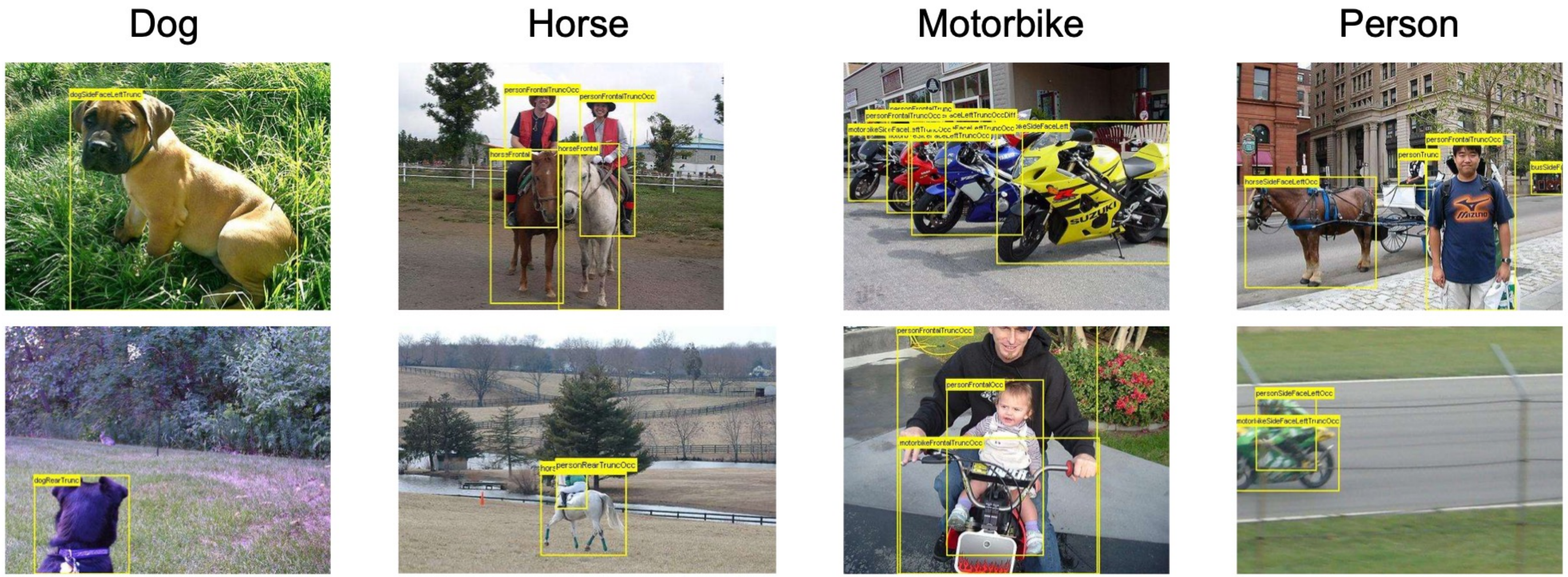
[Courtesy: Zisserman “Self-supervised Learning”]

SSL from Images, EX (I): relative positioning

Evaluation: PASCAL VOC Detection

- 20 object classes (car, bicycle, person, horse …)

- Predict the bounding boxes of all objects of a given class in an image (if any)



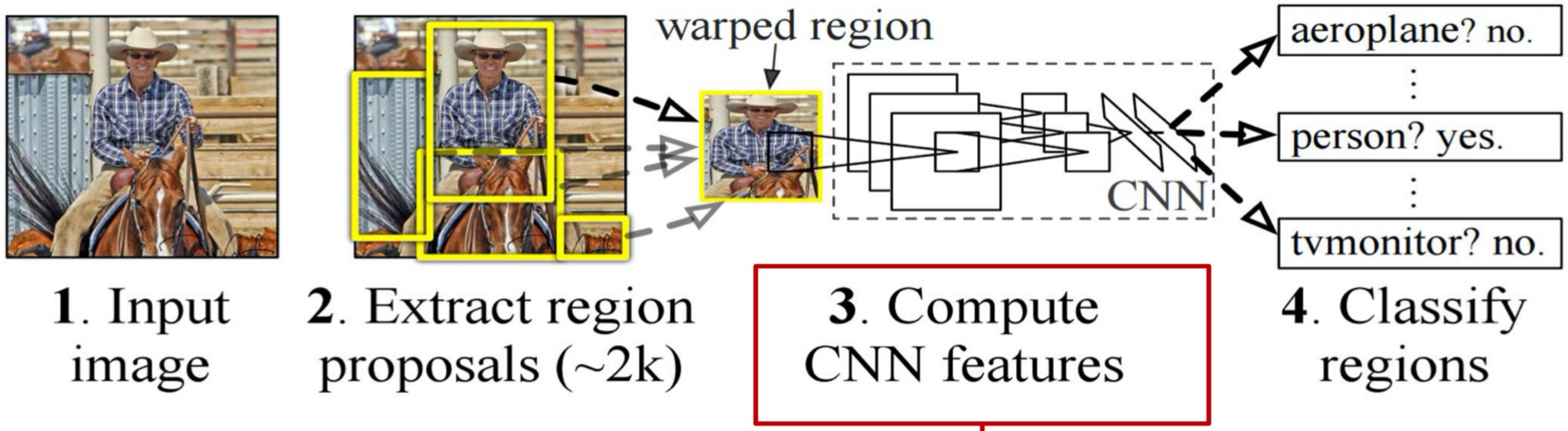
[Courtesy: Zisserman “Self-supervised Learning”]

SSL from Images, EX (I): relative positioning

Evaluation: PASCAL VOC Detection

- Pre-train CNN using self-supervision (no labels)
- Train CNN for detection in R-CNN object category detection pipeline

R-CNN



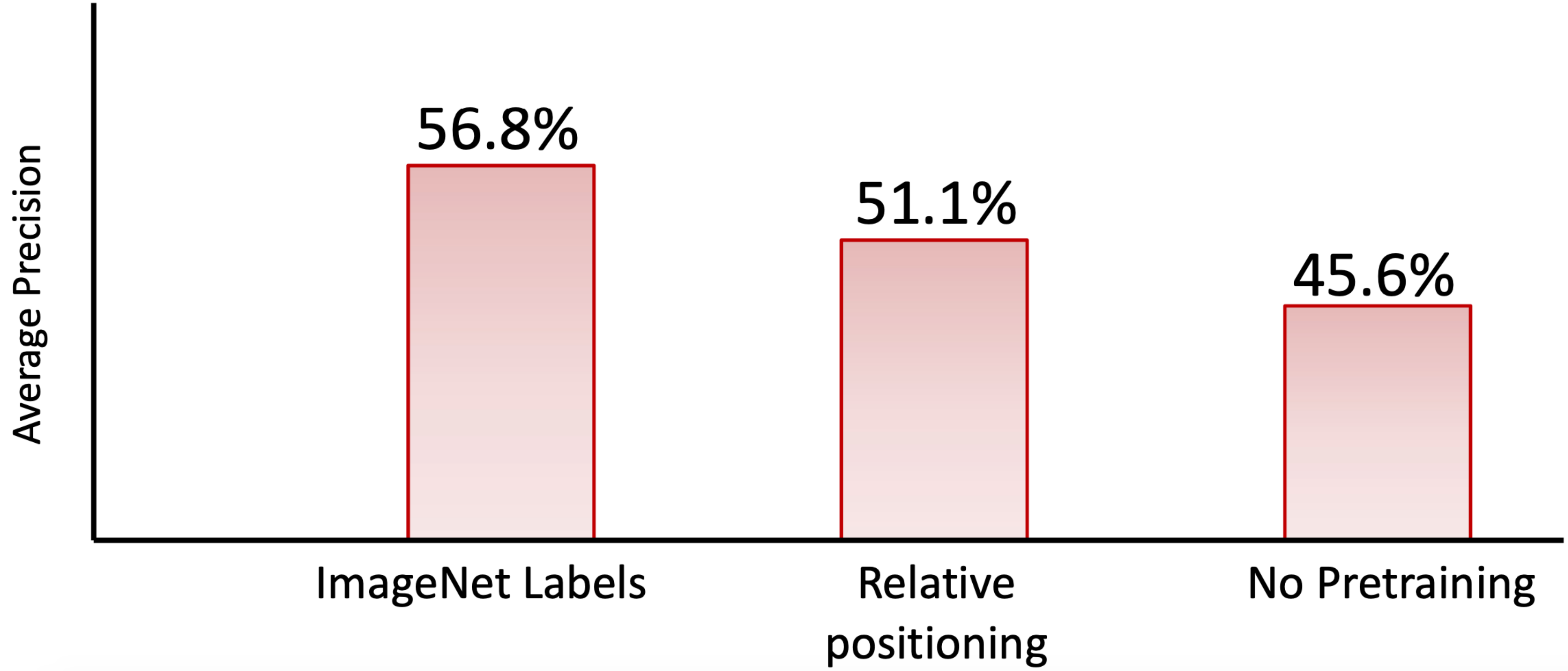
Pre-train on relative-position task, w/o labels

[Girshick et al. 2014]

[Courtesy: Zisserman “Self-supervised Learning”]

SSL from Images, EX (I): relative positioning

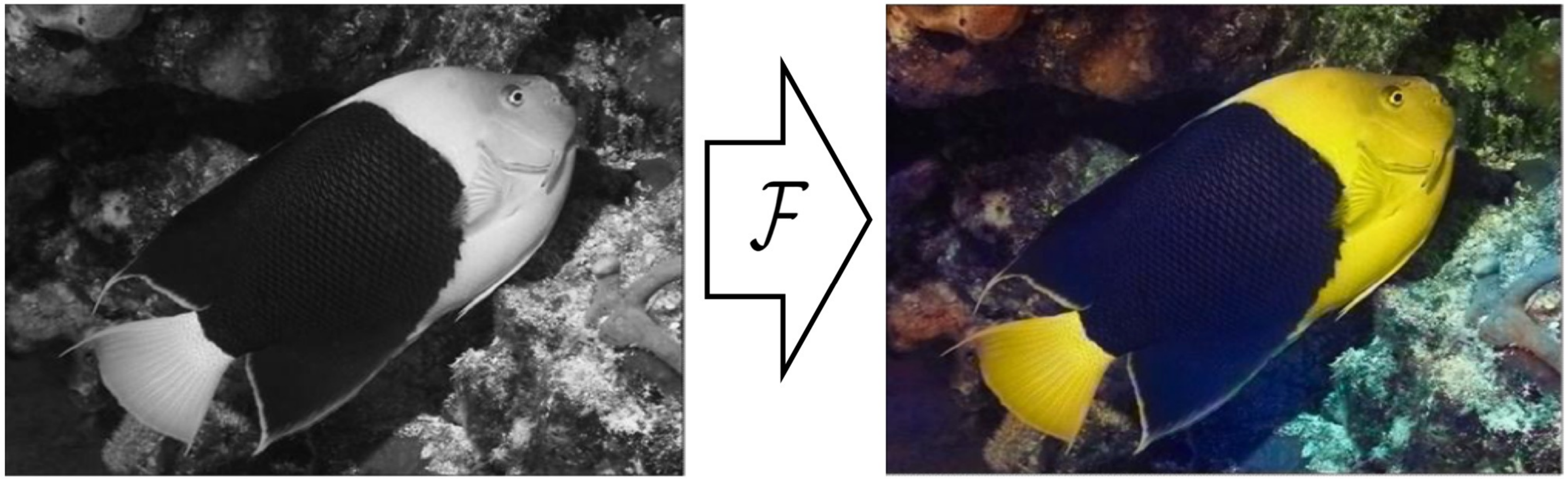
Evaluation: PASCAL VOC Detection



[Courtesy: Zisserman “Self-supervised Learning”]

SSL from Images, EX (II): colorization

Train network to predict pixel colour from a monochrome input

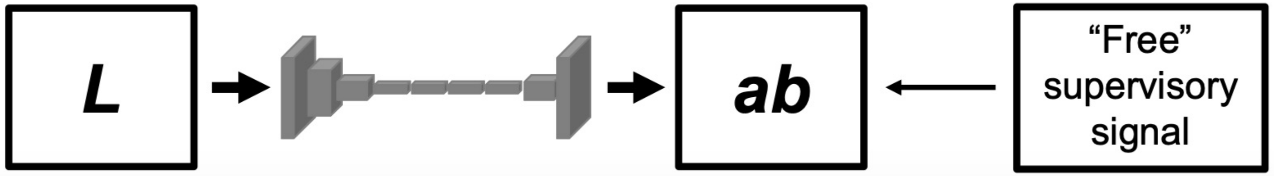


Grayscale image: *L* channel

$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$

Concatenate (L,ab)

$$(\mathbf{X}, \widehat{\mathbf{Y}})$$



[Courtesy: Zisserman "Self-supervised Learning"]

Colorful Image Colorization, Zhang et al., ECCV 2016

SSL from Images, EX (II): colorization

Train network to predict pixel colour from a monochrome input

[Courtesy: Zisserman "Self-supervised Learning"]

Colorful Image Colorization, Zhang et al., ECCV 2016

SSL from Images, EX (III): exemplar networks

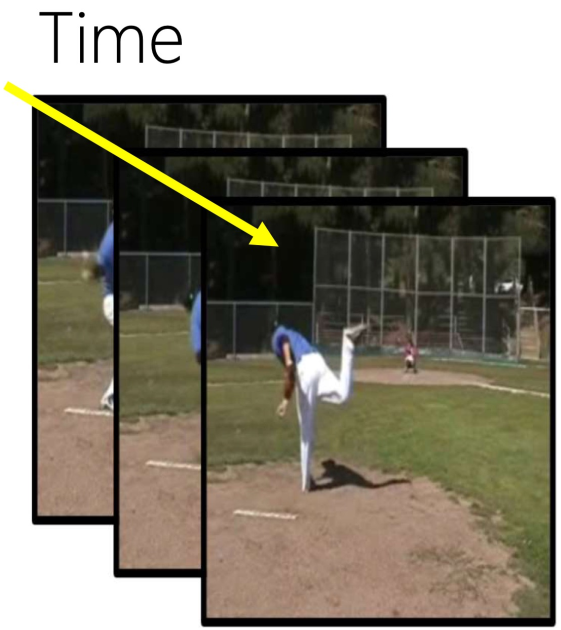
- Exemplar Networks (Dosovitskiy et al., 2014)
- Perturb/distort image patches, e.g. by cropping and affine transformations
- Train to classify these exemplars as same class

[Courtesy: Zisserman "Self-supervised Learning"]

SSL from Videos

Three example tasks:

- Video sequence order
  - Sequential Verification: Is this a valid sequence?



"Sequence" of data

[Courtesy: Zisserman "Self-supervised Learning"]

Wei et al., 2018 Arrow of Time

SSL from Videos

Three example tasks:

- Video sequence order
  - Sequential Verification: Is this a valid sequence?
- Video direction
  - Predict if video playing forwards or backwards

[Courtesy: Zisserman "Self-supervised Learning"]

Wei et al., 2018 Arrow of Time

SSL from Videos

Three example tasks:

- Video sequence order
  - Sequential Verification: Is this a valid sequence?
- Video direction
  - Predict if video playing forwards or backwards
- Video tracking
  - Given a color video, colorize all frames of a gray scale version using a reference frame

[Courtesy: Zisserman “Self-supervised Learning”]

Vondrick et al., 2018

Key Takeaways

- Self supervision learning
  - Predicting any part of the observations given any available information
  - The prediction task forces models to learn semantic representations
  - Massive/unlimited data supervisions
- SSL for text:
  - Language models: next word prediction
  - BERT text representations: masked language model (MLM)
- SSL for images/videos:
  - Various ways of defining the prediction task

Questions?