Topics:

- Masked Language Models **(dropbox M3L12)**
- Embeddings **(dropbox M3L13)**
- Reinforcement Learning introduction

CS 4644-DL / 7643-A
ZSOLT KIRA

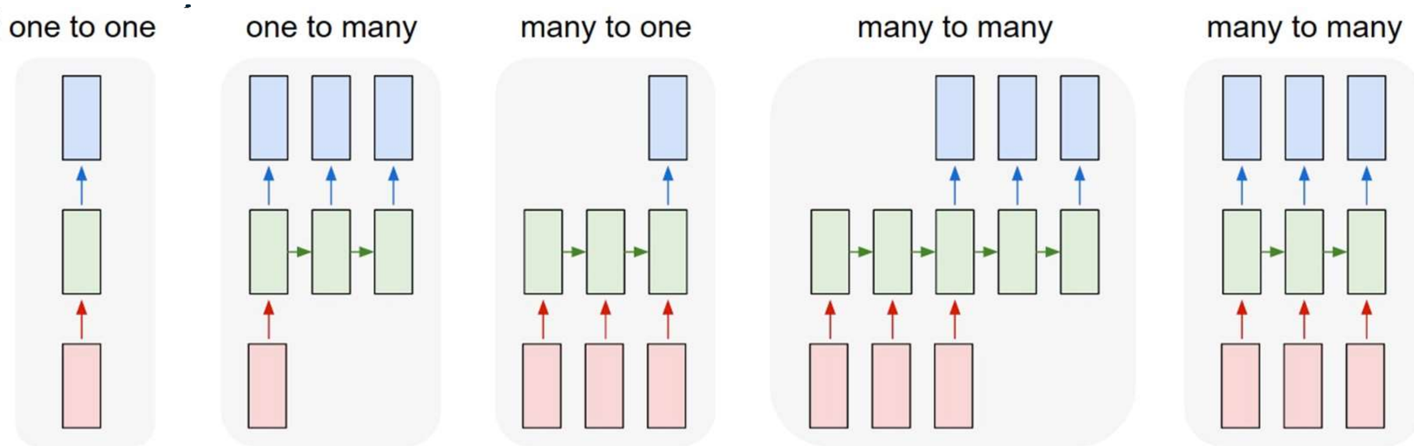
- **Assignment 4 out**
  - Due **April 4th 11:59pm EST (grace April 6th)**
  - Do not submit first version last-minute on 6th!
    - Please submit *something* by deadline (Apr 4th) to avoid last-minute hiccups and zero!

- **Projects**
  - Project due **May 1st 11:59pm EST**

- Outline of rest of course:
  - Today we start (deep) reinforcement learning
  - Guest lectures/other topics (e.g. self-supervised learning)
  - Generative models (VAEs / GANs)

Sequences in Input or Output?

- It's



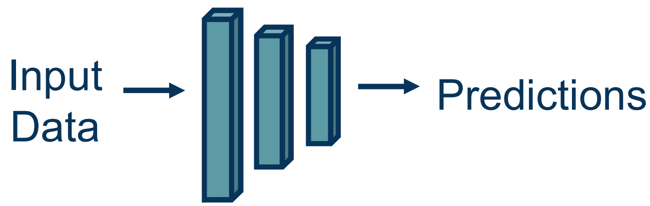
Input: No sequence
Output: No sequence
Example: "standard" classification / regression problems

Input: No sequence
Output: Sequence
Example: Im2Caption

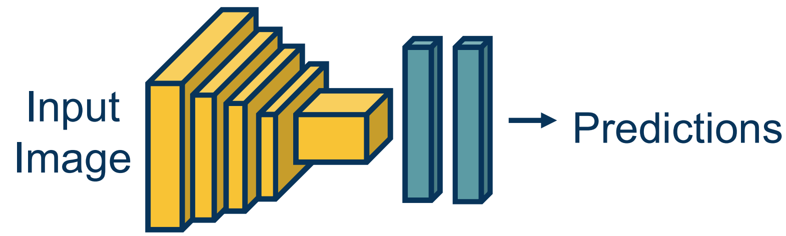
Input: Sequence
Output: No sequence
Example: sentence classification, multiple-choice question answering

Input: Sequence
Output: Sequence
Example: machine translation, video classification, video captioning, open-ended question answering

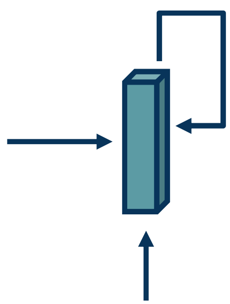
Image Credit: Andrej Karpathy



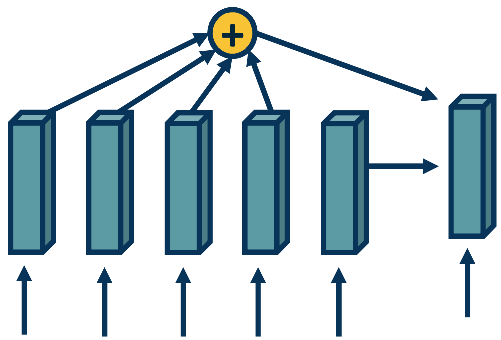
Fully Connected Neural Networks



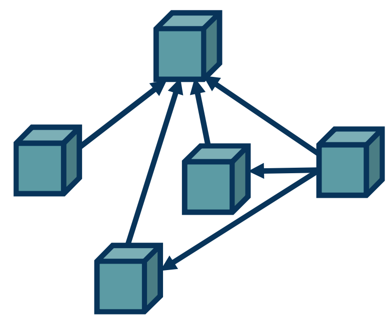
Convolutional Neural Networks



Recurrent Neural Networks

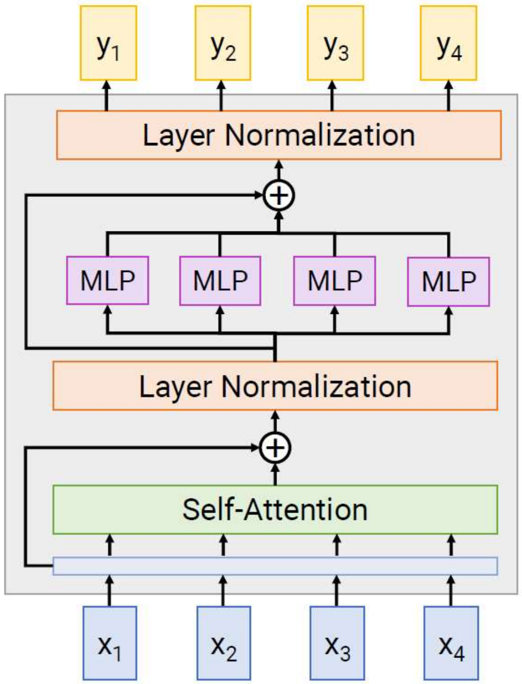


Attention-Based Networks

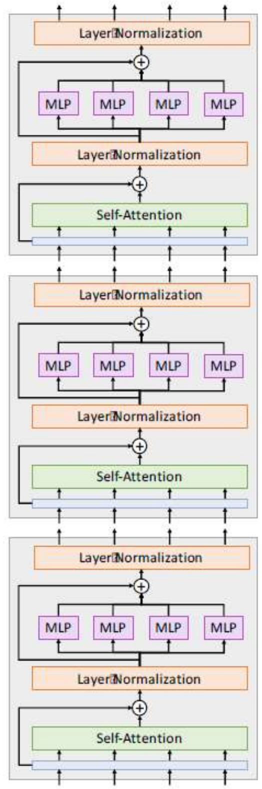


Graph-Based Networks

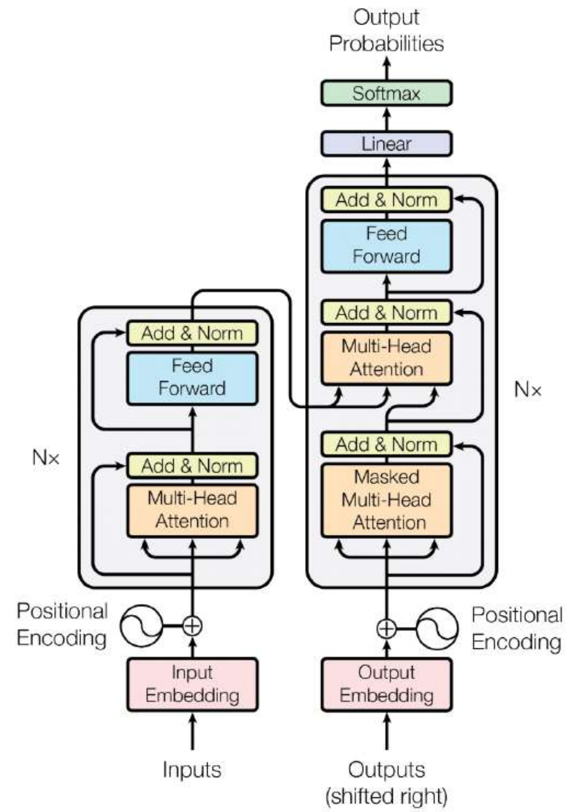
The Space of Architectures



Transformer Block



Multi-Layered



Encoder/Decoder

Recall: Transformers

Georgia Tech

Masked Language Models

FACEBOOK AI Georgia Tech

Jean Maillard

Jean Maillard is a Research Scientist on the Language And Translation Technologies Team (LATTE) at Facebook AI. His research interests within NLP include word- and sentence-level semantics, structured prediction, and low-resource languages. Prior to joining Facebook in 2019, he was a doctoral student with the NLP group at the University of Cambridge, where he researched compositional semantic methods. He received his BSc in Theoretical Physics from Imperial College London.

Lecturer Introduction

- **Recall:** language models estimate the probability of sequences of words:

$$\mathrm{p}(\boldsymbol{s}) = \mathrm{p}(w_1, w_2, \ldots, w_n)$$

- ***Masked language modeling*** is a related ***pre-training task*** – an auxiliary task, different from the final task we're really interested in, but which can help us achieve better performance by finding good initial parameters for the model.

- By pre-training on masked language modeling before training on our final task, it is usually possible to obtain higher performance than by simply training on the final task.

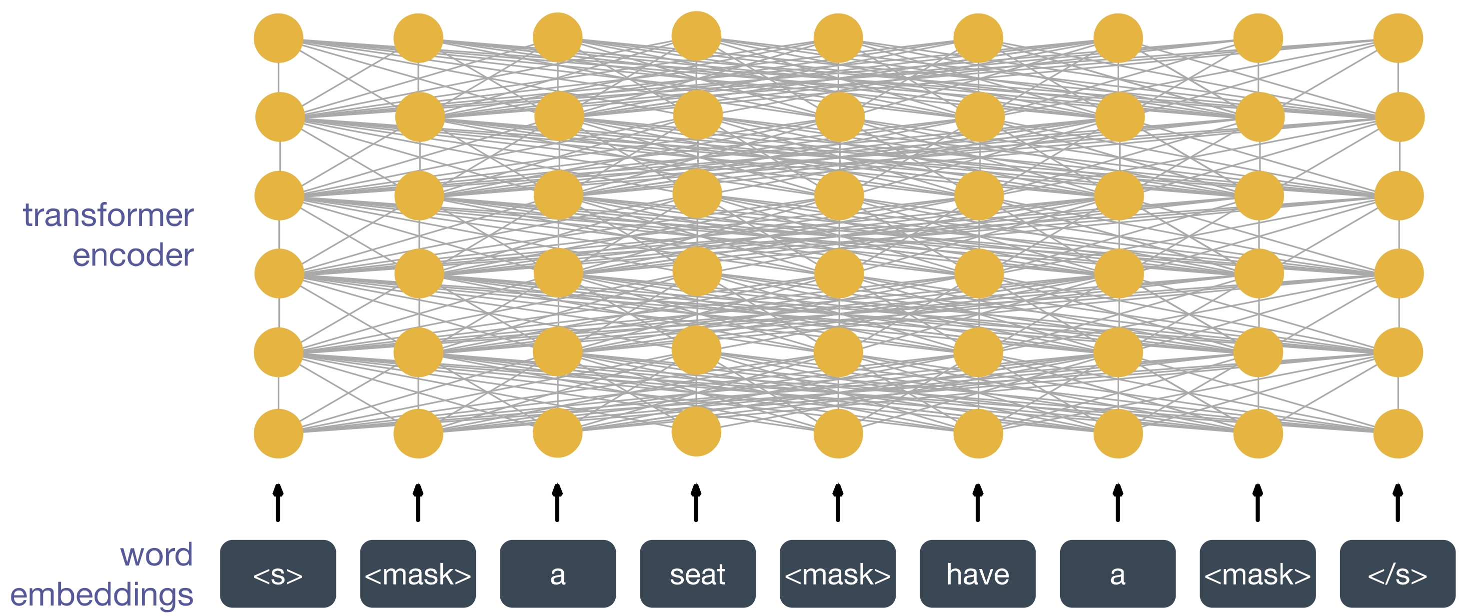
take a seat , have a drink

Masked Language Models

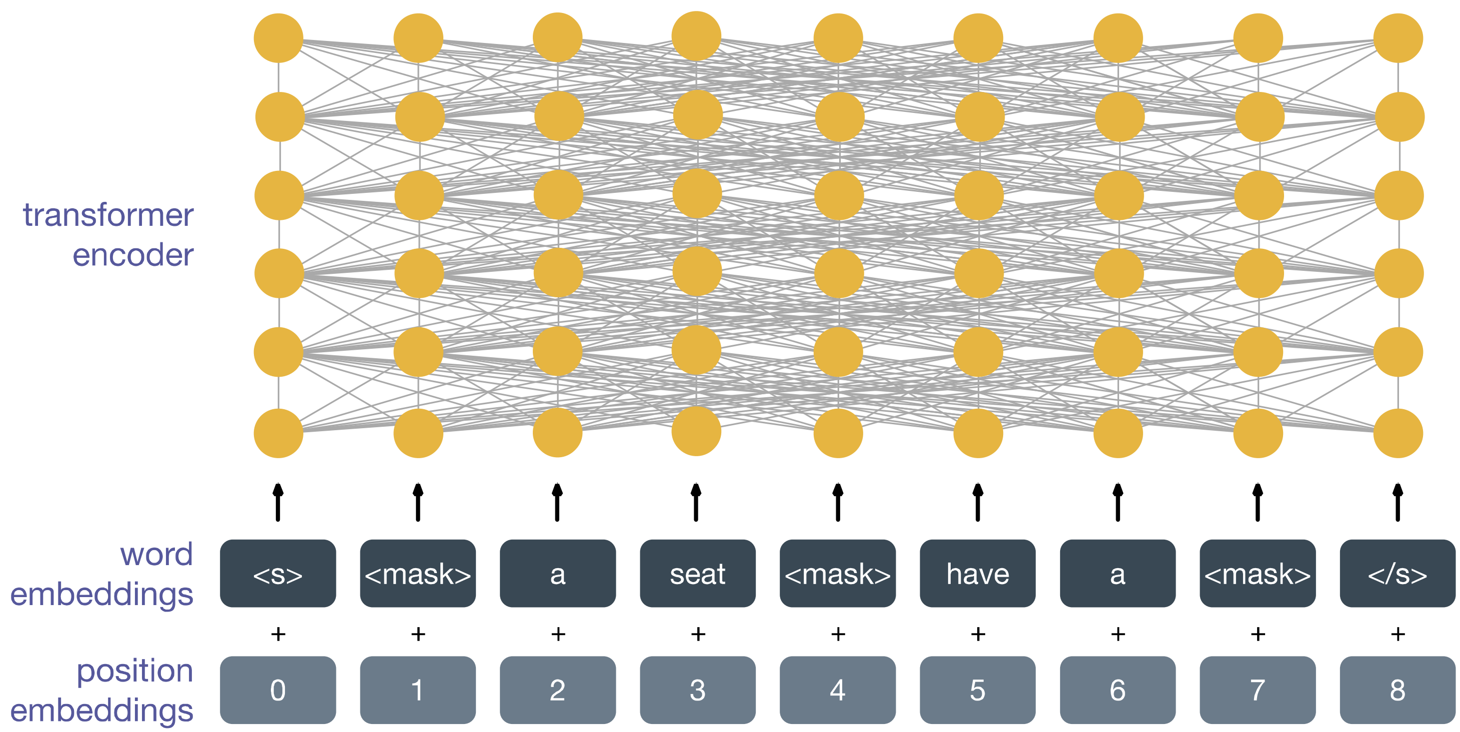
FACEBOOK AI Georgia Tech

<s> <mask> a seat <mask> have a <mask> </s>

Masked Language Models

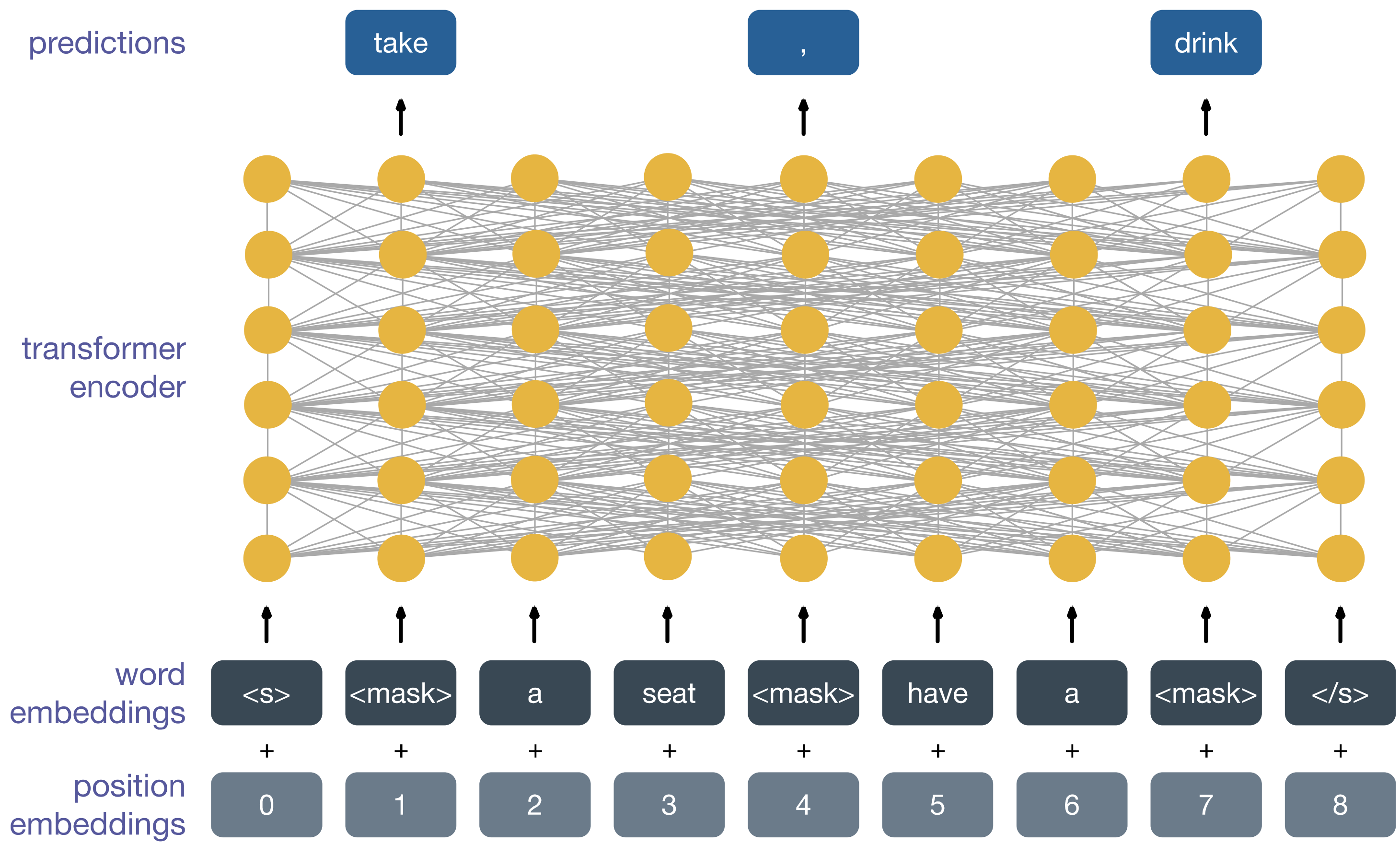


Masked Language Models

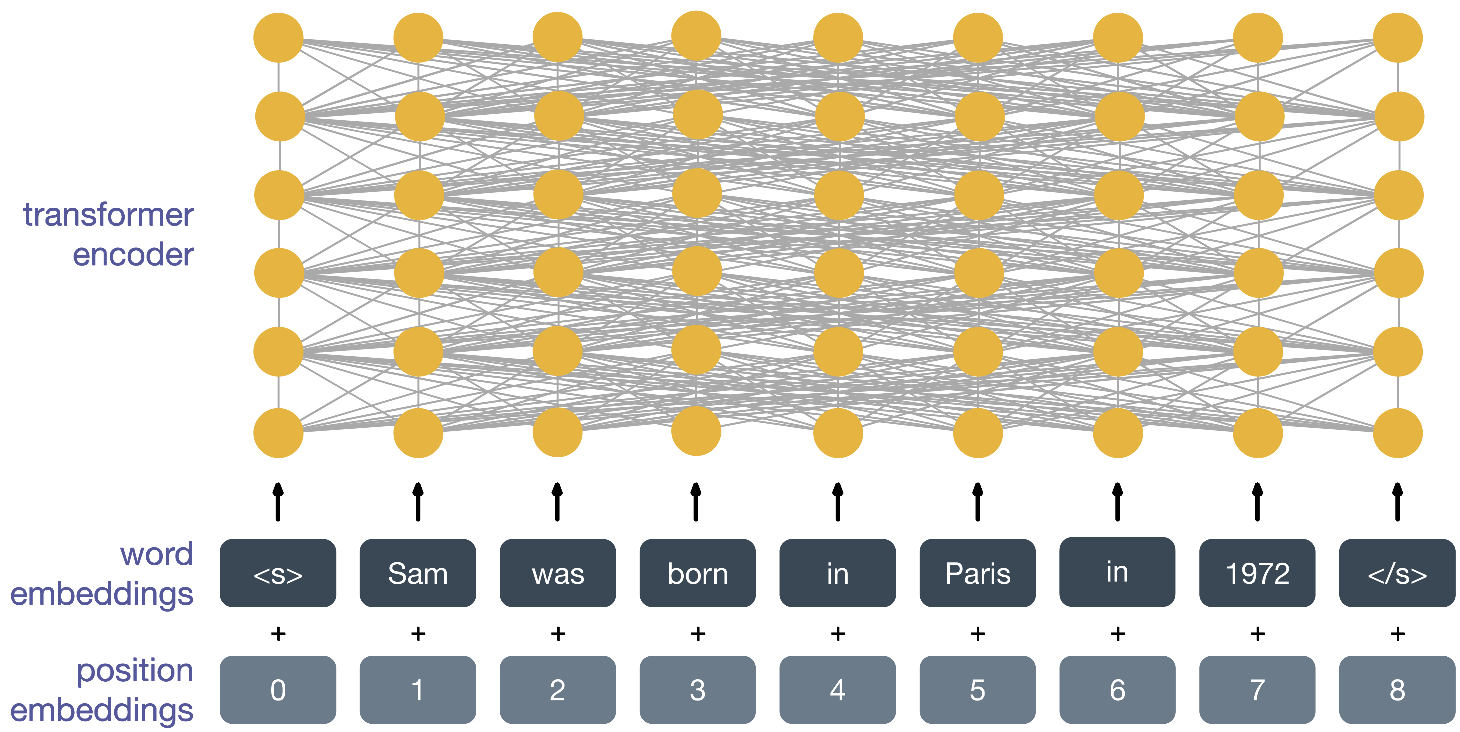


Masked Language Models

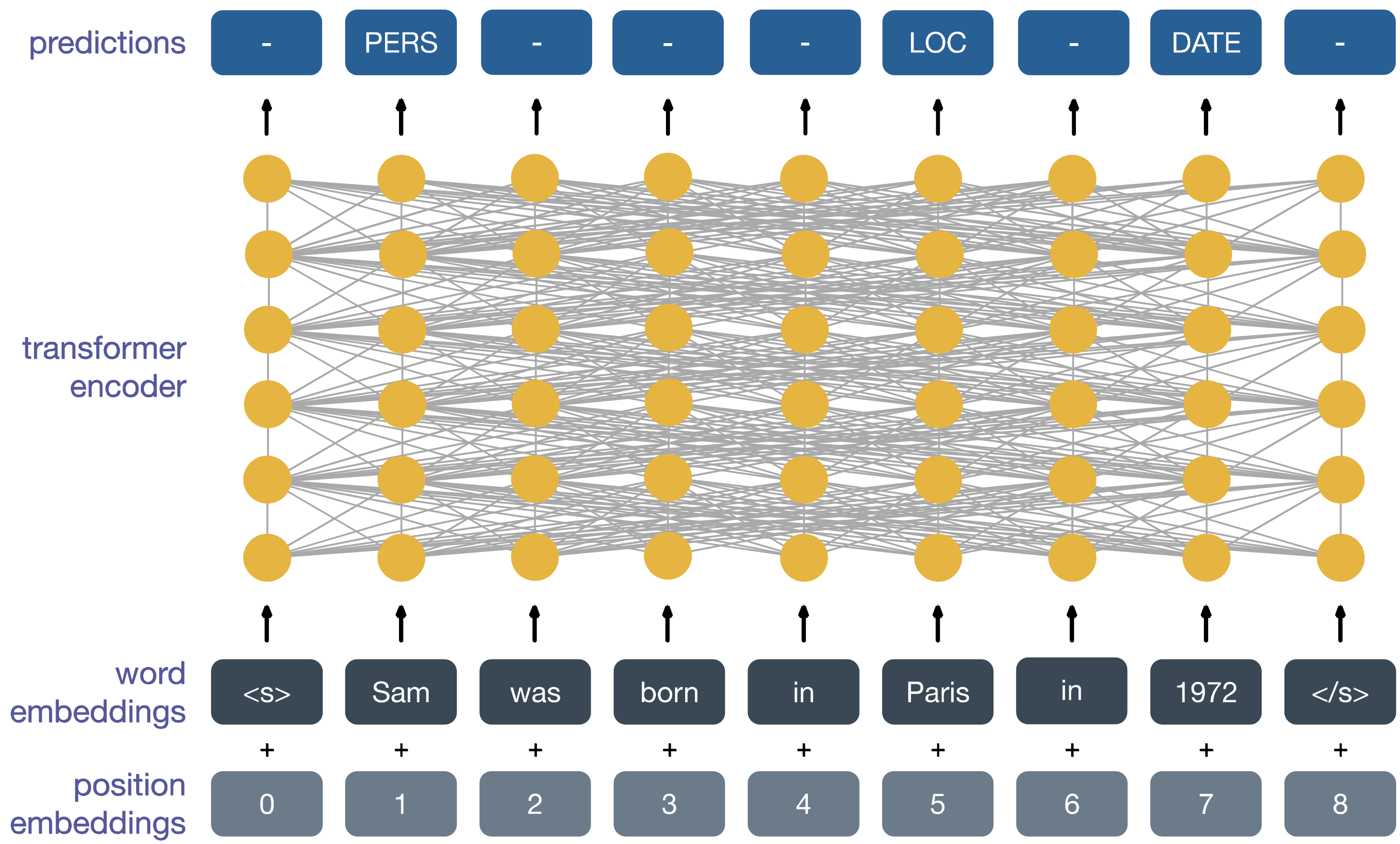
FACEBOOK AI Georgia Tech



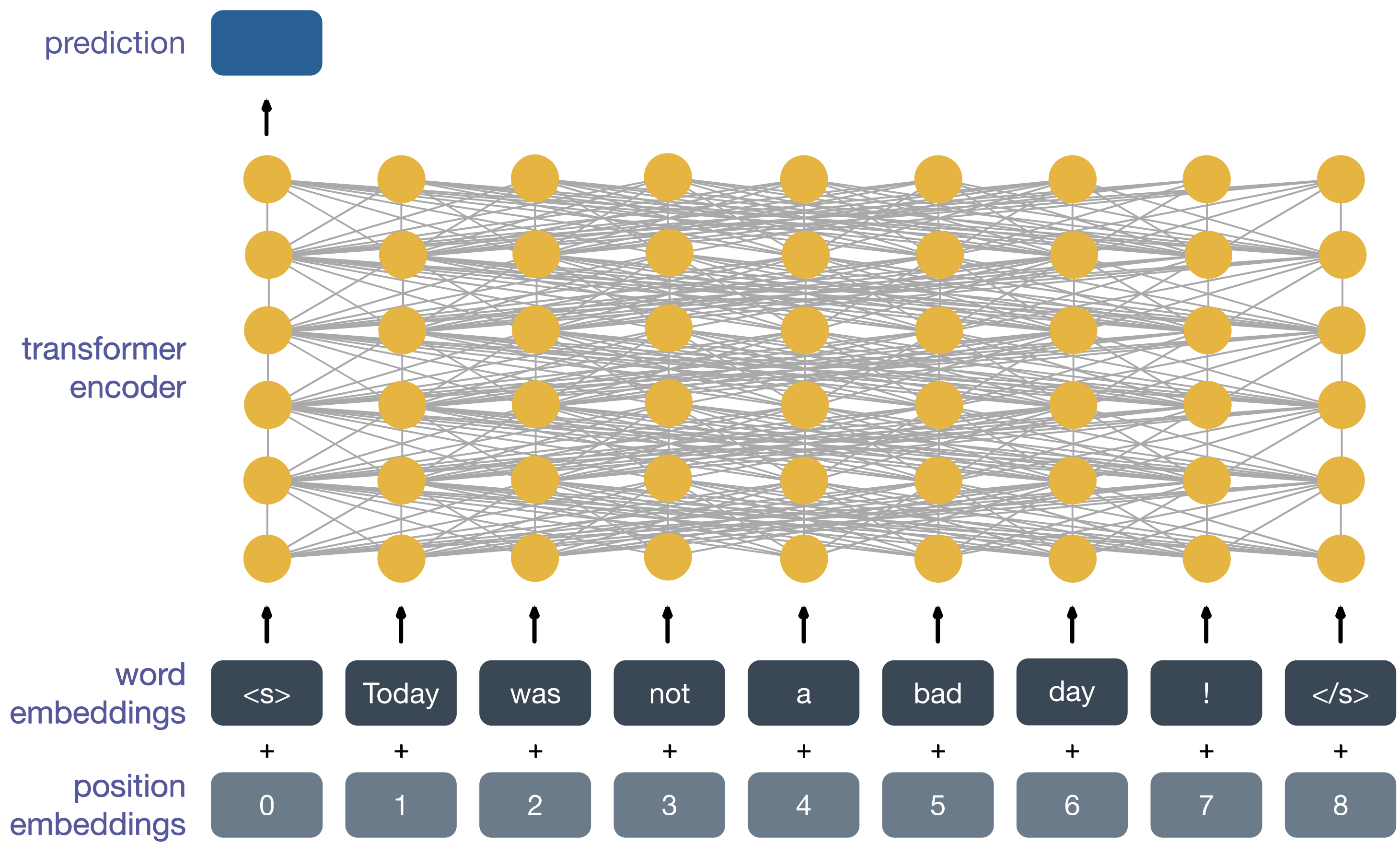
Masked Language Models



Token-level Tasks

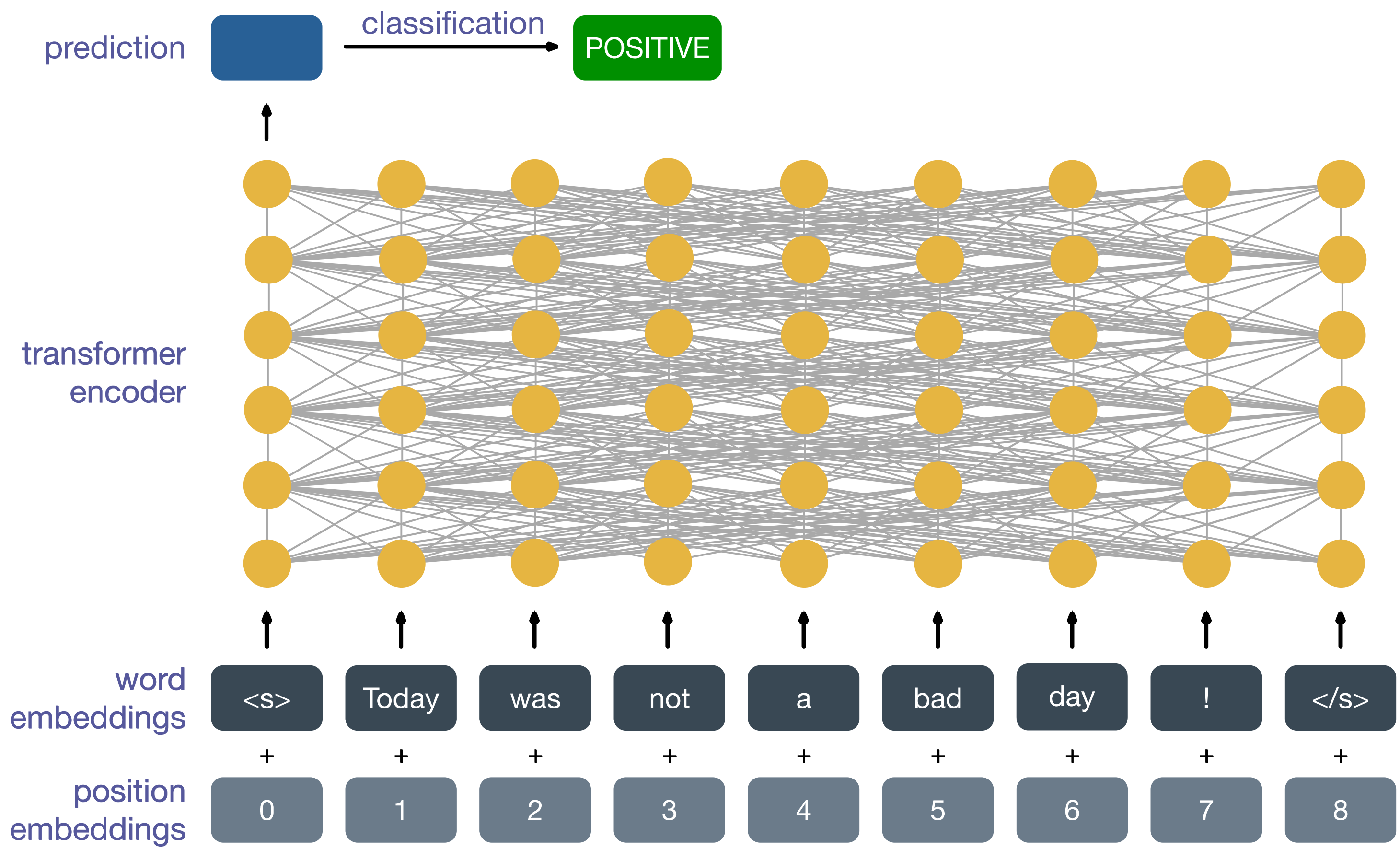


Token-level Tasks



Sentence-level Tasks

FACEBOOK AI Georgia Tech



Sentence-level Tasks

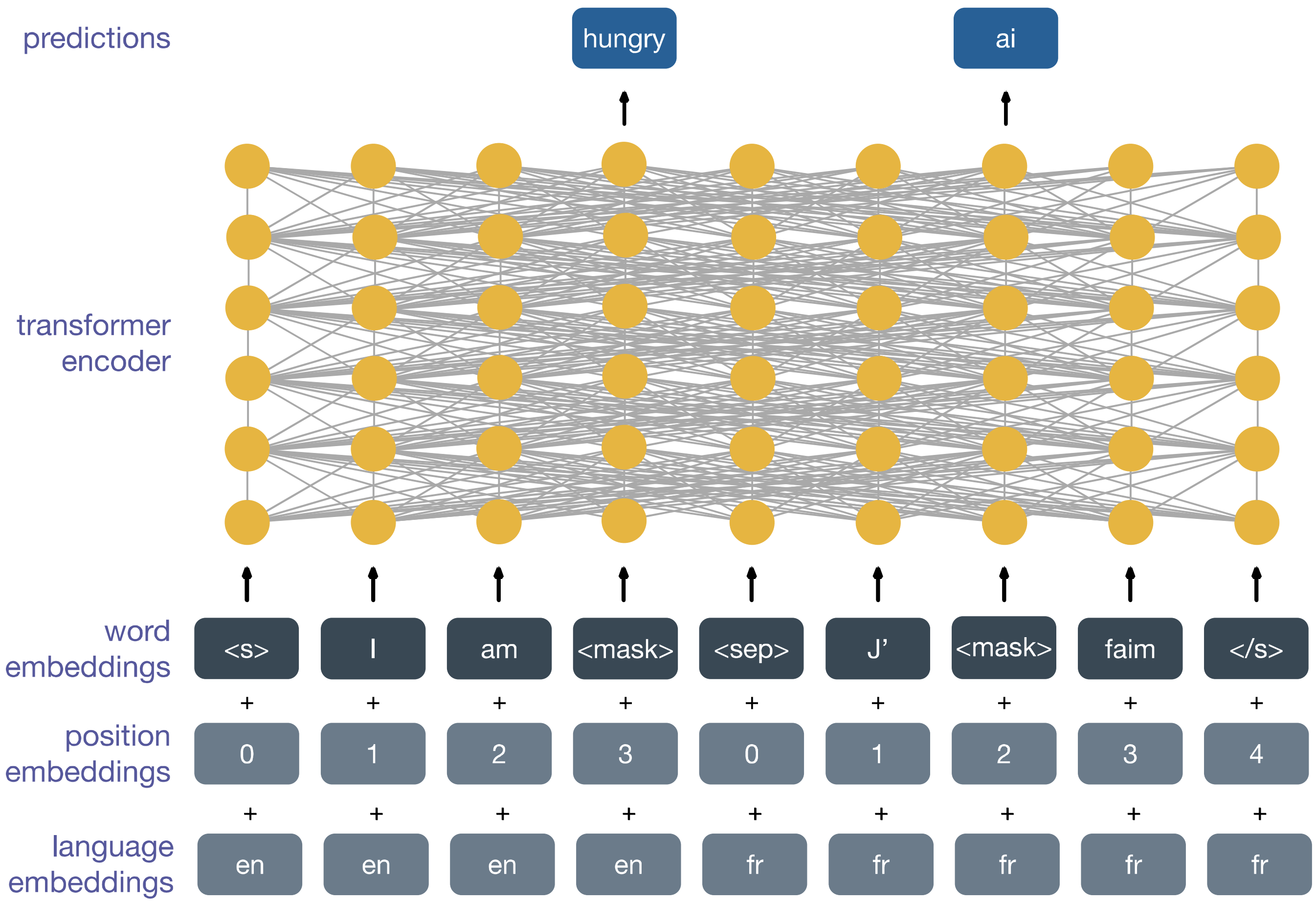
I am hungry J’ ai faim

Cross-lingual Masked Language Modeling

<s> I am <mask> <sep> J’ <mask> faim </s>

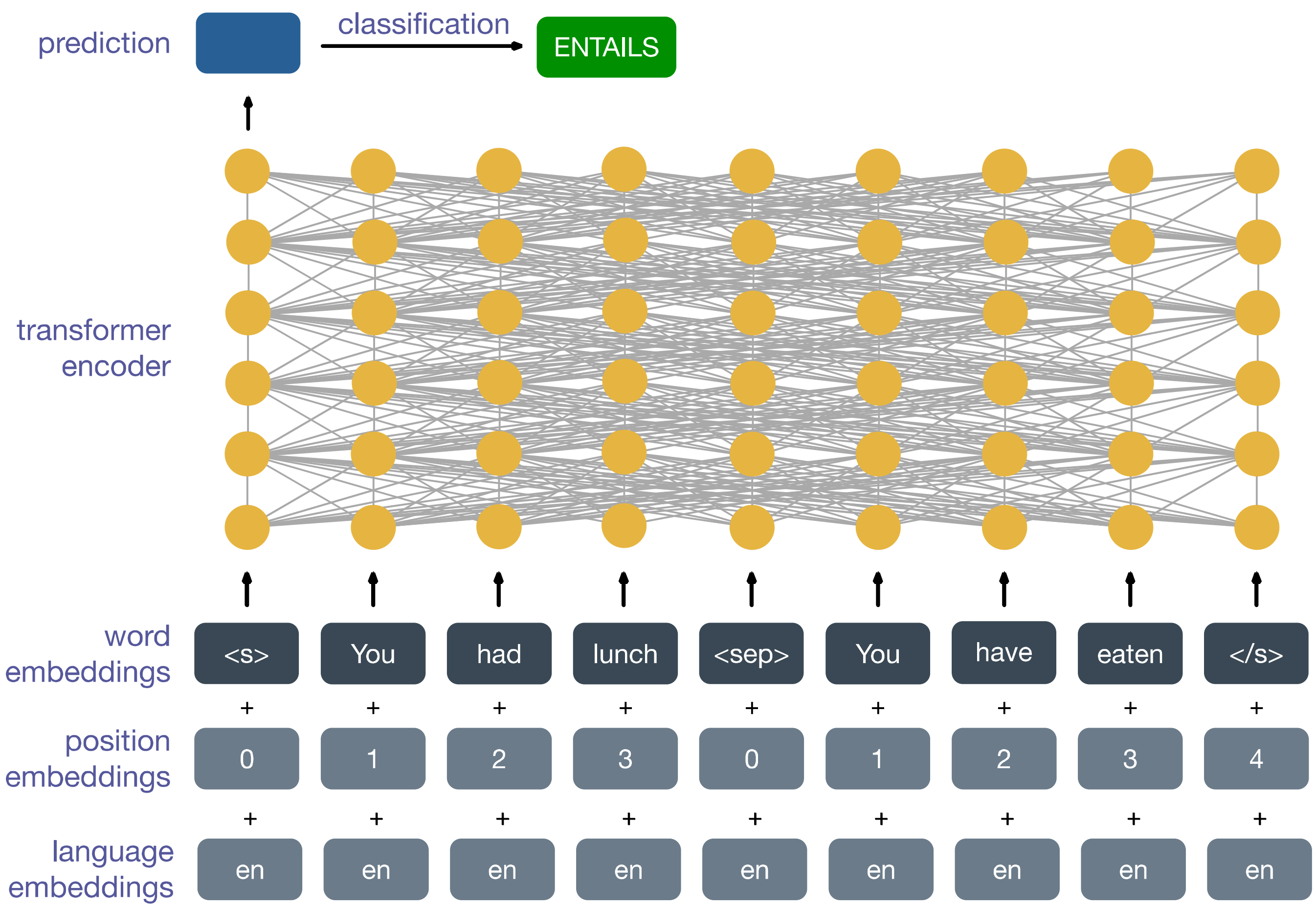
Cross-lingual Masked Language Modeling

FACEBOOK AI Georgia Tech

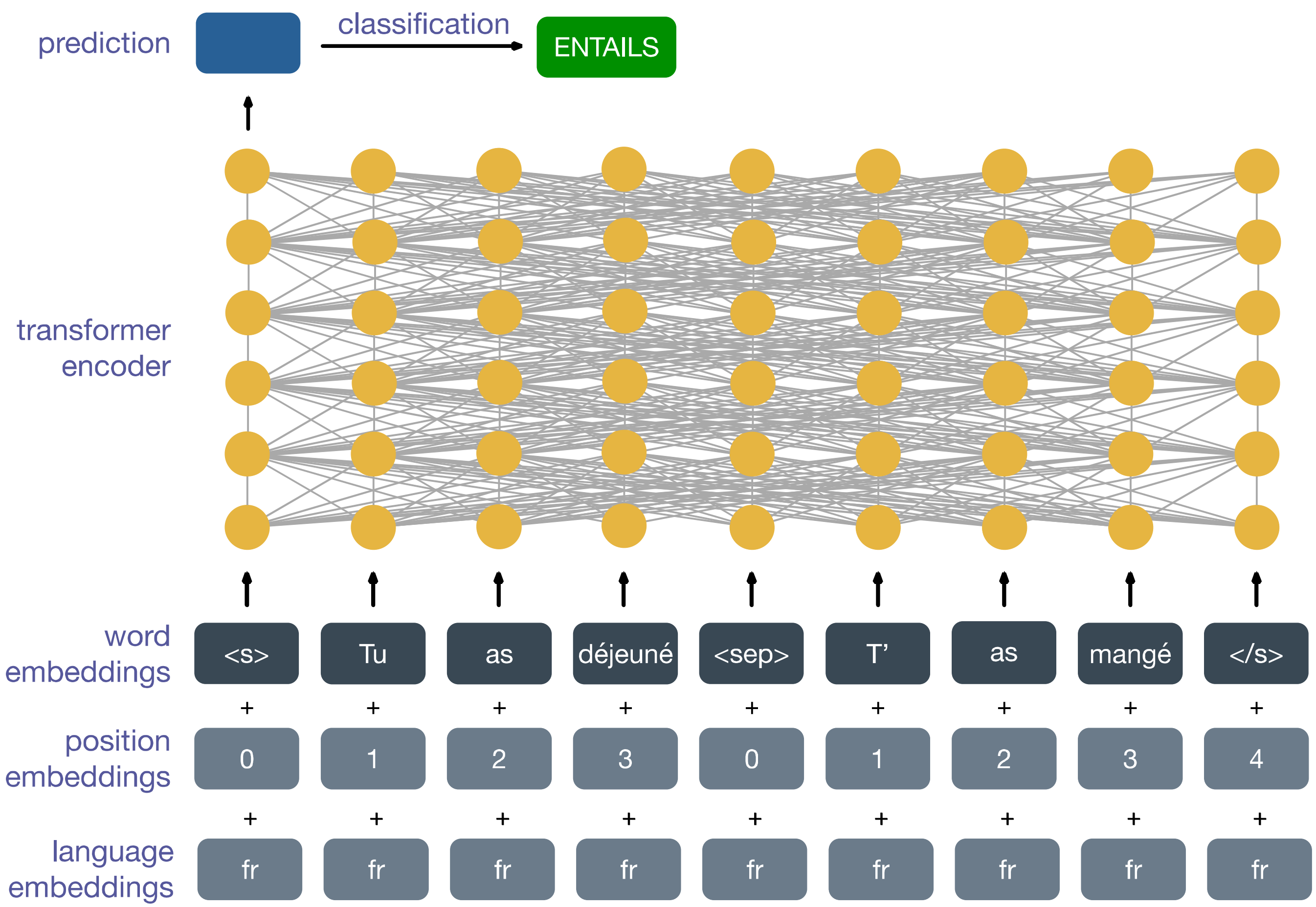


Cross-lingual Masked Language Modeling

FACEBOOK AI Georgia Tech



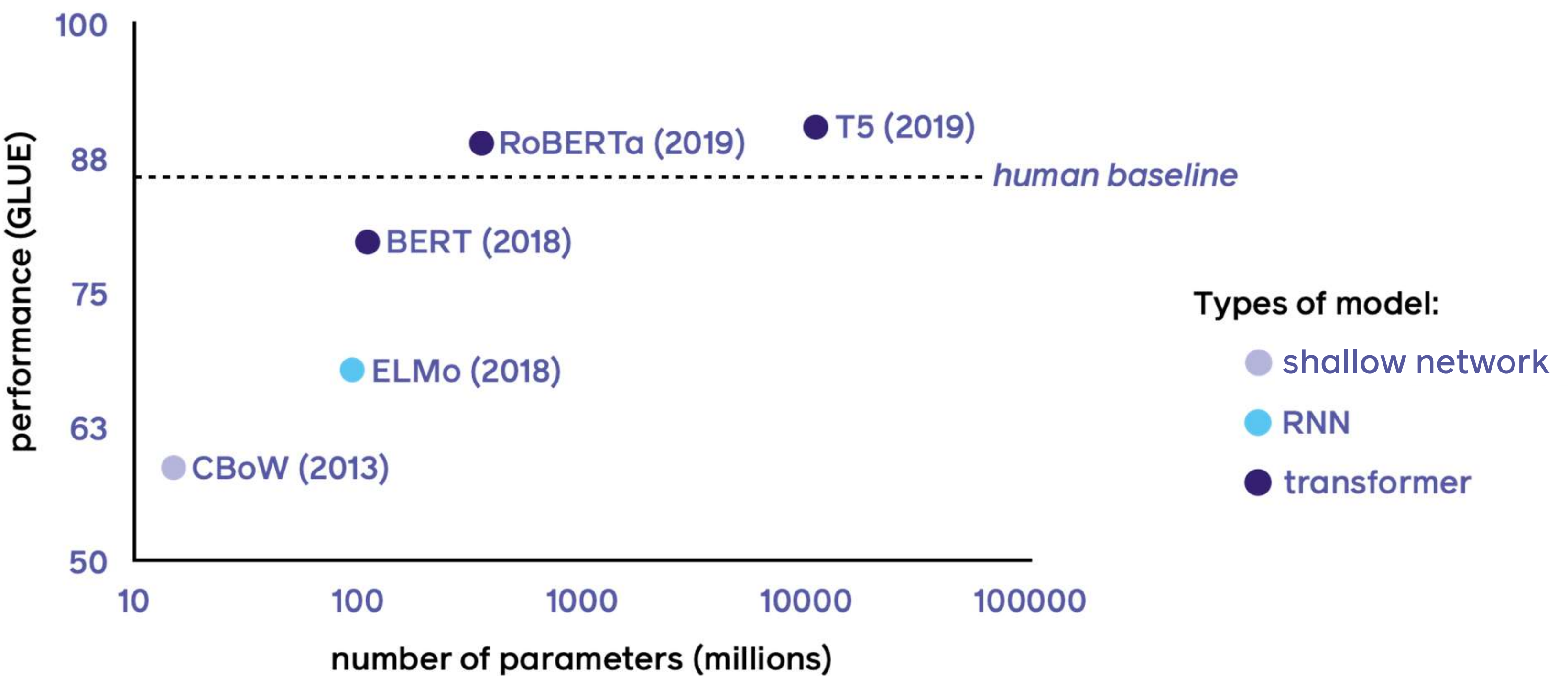
Cross-lingual Task: Natural Language Inference



Cross-lingual Task: Natural Language Inference

FACEBOOK AI

Georgia Tech



Model Size in Perspective

FACEBOOK AI Georgia Tech

Preprint. Under review.

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

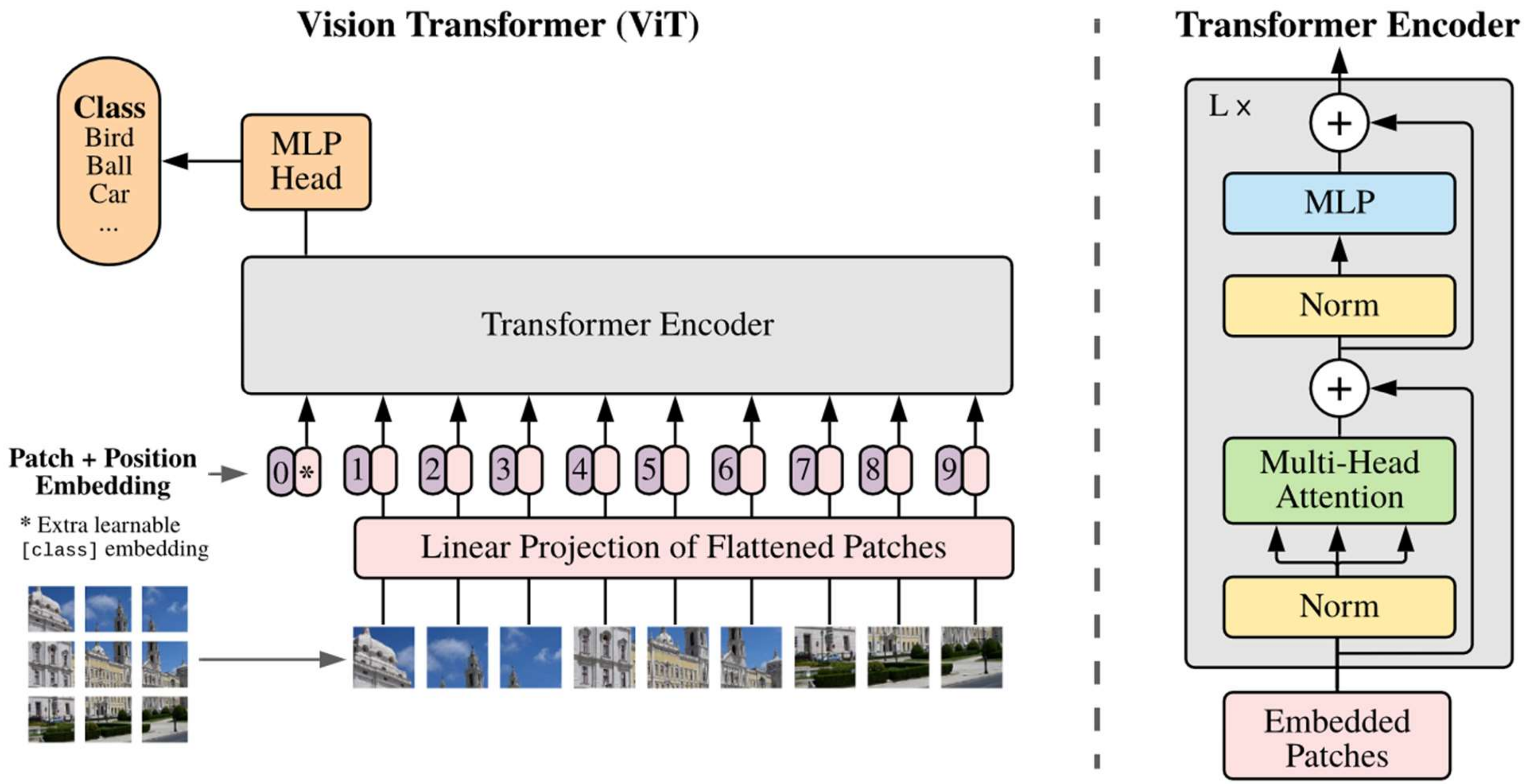
Alexey Dosovitskiy[*,†], Lucas Beyer[*], Alexander Kolesnikov[*], Dirk Weissenborn[*], Xiaohua Zhai[*], Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby[*,†]
[*]equal technical contribution, [†]equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com

[cs.CV] 22 Oct 2020

Abstract

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train[1]

What About Vision?



Vision Transformer (ViT)

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21K (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	$\mathbf{88.55} \pm 0.04$	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	$88.4/88.5^*$
ImageNet ReaL	$\mathbf{90.72} \pm 0.05$	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	$\mathbf{99.50} \pm 0.06$	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	$\mathbf{94.55} \pm 0.04$	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	$\mathbf{97.56} \pm 0.03$	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	$\mathbf{99.74} \pm 0.00$	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	$\mathbf{77.63} \pm 0.23$	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

ViT Results

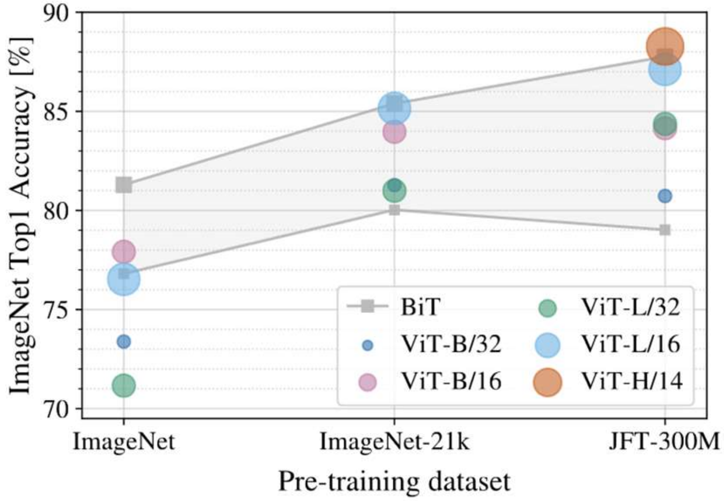


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

When trained on mid-sized datasets such as ImageNet, such models yield modest accuracies of a few percentage points below ResNets of comparable size. This seemingly discouraging outcome maybe expected: Transformers lack some of the inductive biases inherent to CNNs, such as translation equivariance and locality, and therefore do not generalize well when trained on insufficient amounts of data.

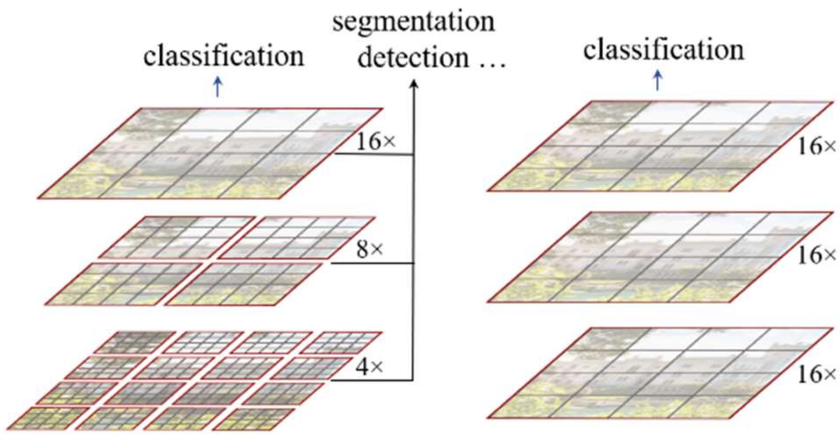
However, the picture changes if the models are trained on larger datasets (14M-300M images). We find that large scale training trumps inductive bias.

Dosovitskiy et al.

ViTs and Transfer Learning

https://paperswithcode.com/sota/image-classification-on-imagenet

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows
Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo



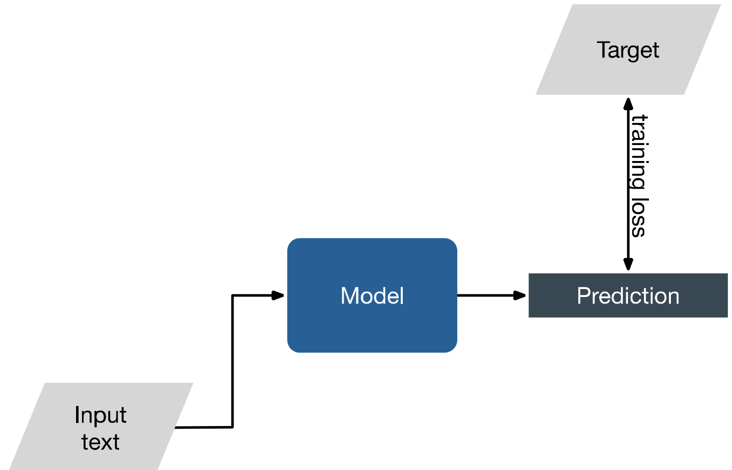
(a) Swin Transformer (ours) (b) ViT

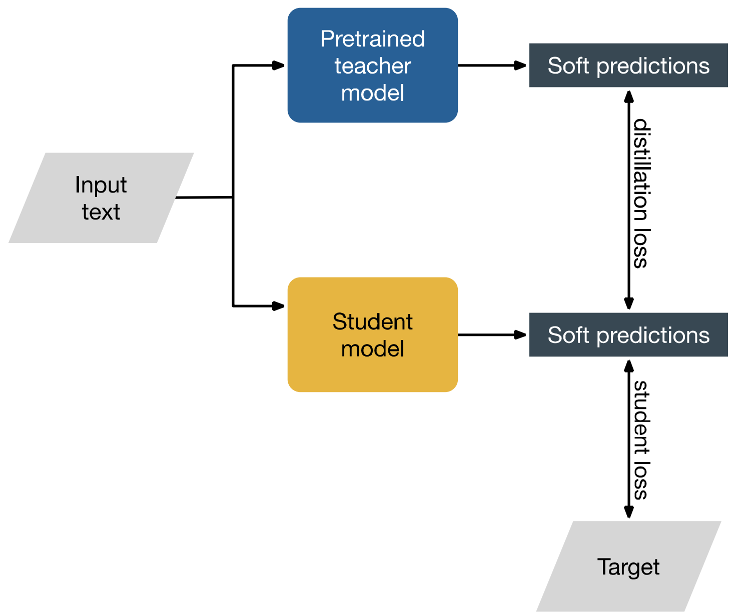
Swin Transformers

https://paperswithcode.com/sota/instance-segmentation-on-coco

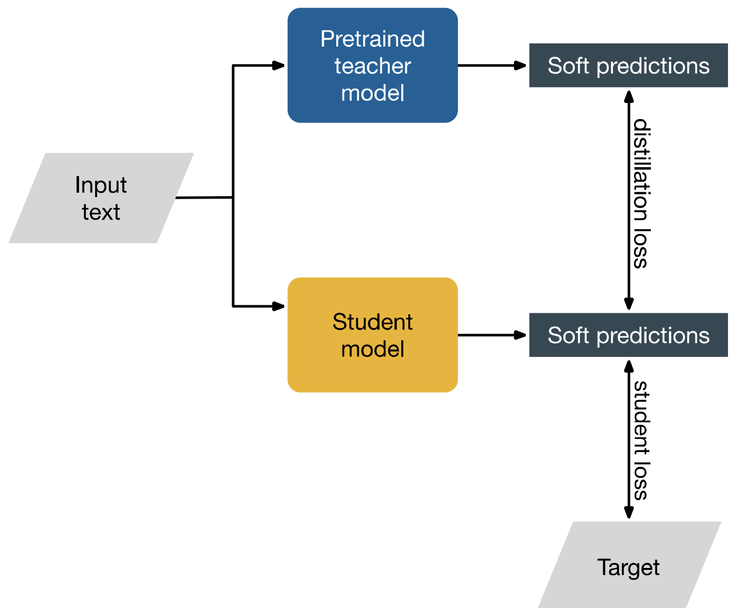
Summary

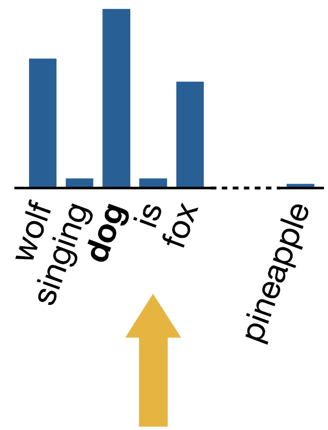
- "Attention" models outperform recurrent models and convolutional models for sequence processing. They allow long range interactions.
- These models do best with LOTS of training data
- Surprisingly, they seem to outperform convolutional networks for image processing tasks. Again, long range interactions might be more important than we realized.
- Naïve attention mechanisms have quadratic complexity with the number of input tokens, but there are often workarounds for this.


Target
training loss
Model
Prediction
Input
text



Knowledge Distillation to Reduce Model Sizes





the <mask> licked its fur and howled

Knowledge Distillation to Reduce Model Sizes

FACEBOOK AI Georgia Tech

cross-entropy $$\mathrm{H}(p^*, p) = -\sum_{x \in \mathcal{X}} p^*(x) \log p(x)$$

reference distribution

$$\mathcal{L}_{\text{dist}} = \mathrm{H}(t, s) = -\sum_i t_i \log s_i \quad \text{or} \quad \mathrm{D}_{\mathrm{KL}}(t \| s)$$

$$\mathcal{L}_{\text{student}} = \mathrm{H}(y, s) = -\sum_i y_i \log s_i$$

$$\mathcal{L} = \alpha \mathcal{L}_{\text{dist}} + \beta \mathcal{L}_{\text{student}}$$

- Vaswani et al. (2017). "Attention is all you need", in *NIPS 2017*.
- Devlin et al. (2018). "BERT: pre-training of deep bidirectional transformers for language understanding".
- Liu, Ott, Goyal, Du, et al. (2019). "RoBERTa: a robustly optimized BERT pretraining approach".
- Lample & Conneau (2019). "Cross-lingual language model pretraining", in NeurIPS 2019.
- Conneau, Khandelwal, et al. (2020). "Unsupervised cross-lingual representation learning at scale", in *ACL 2020*.
- Lewis, Liu, Goyal, et al. (2019). "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension", in *ACL 2020*.
- Raffel, Shazeer, Roberts, Lee, et al. (2020), "Exploring the limits of transfer learning with a unified text-to-text transformer", in *JMLR* 21(2020): 1-67.
- Hinton, Vinyals, Dean (2015). "Distilling the knowledge in a neural network", in *NIPS 2014 deep learning workshop*.

References

Ledell Wu

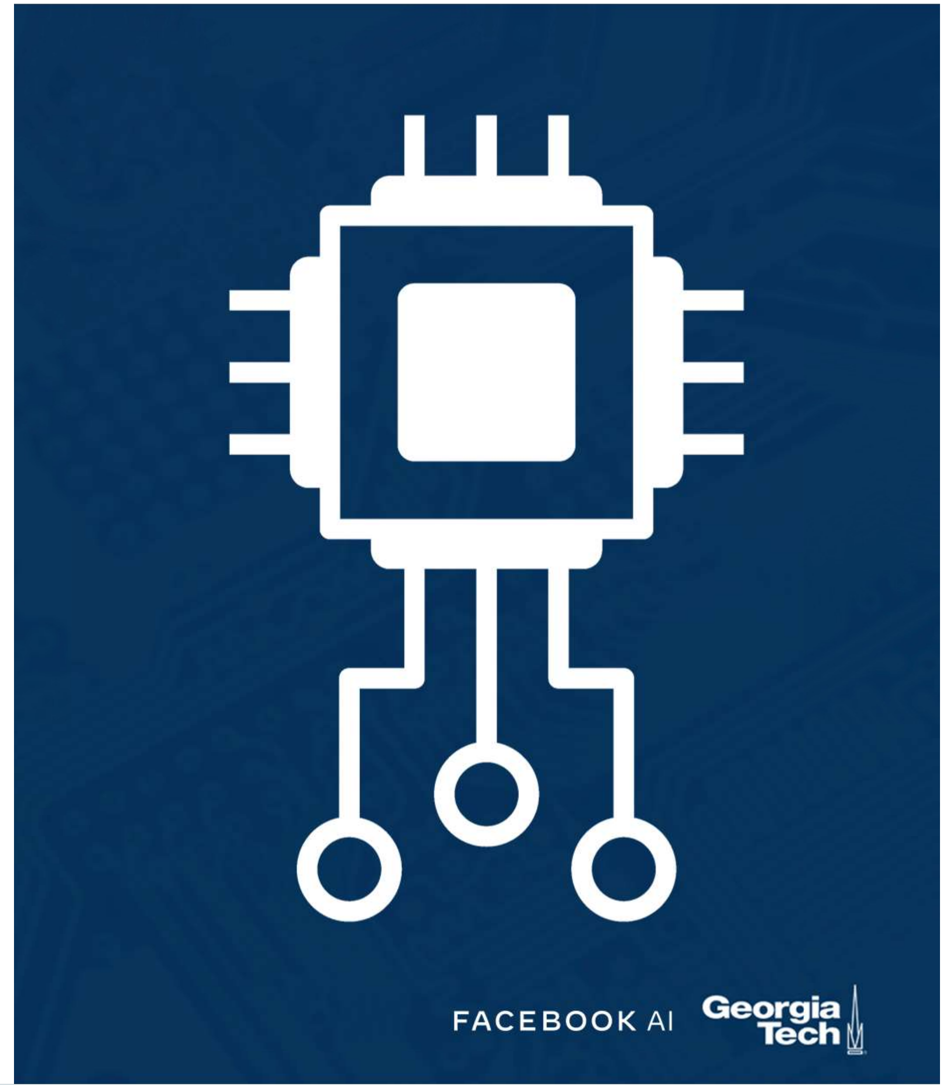
Ledell Wu is a research engineer at Facebook AI Research. Ledell joined Facebook in 2013 after graduating from University of Toronto. She worked on Newsfeed ranking as a machine learning engineer. After joining Facebook AI, Ledell worked on general purpose and large-scale embedding systems. She collaborated with teams including page recommendations, video recommendations, ads interest suggestion, people search and feed integrity, to use embeddings to better serve products. She is one of the main contributors in open source projects including StarSpace (general purpose embedding system), PyTorch Big-Graph (large-scale graph embedding system) and BLINK (entity linking). Ledell also studies fairness and biases in machine learning models.

Lecturer Introduction

FACEBOOK AI Georgia Tech

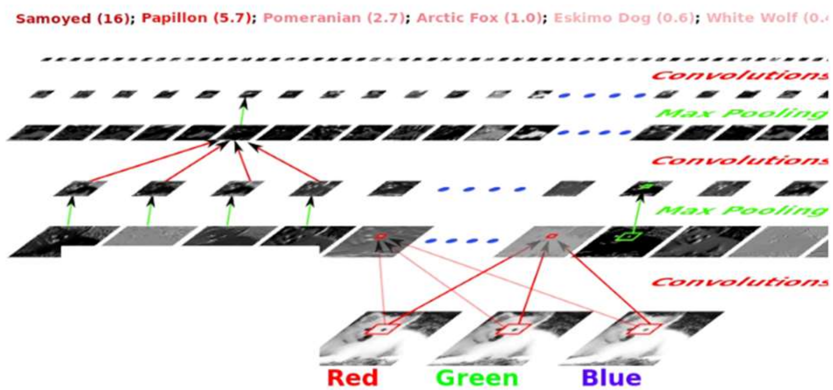
Embeddings

- Word Embeddings
- Graph Embeddings
- Applications, world2vec
- Additional Topics



- **Mapping Objects to Vectors through a trainable function**

[0.4, -1.3, 2.5, -0.7, …]

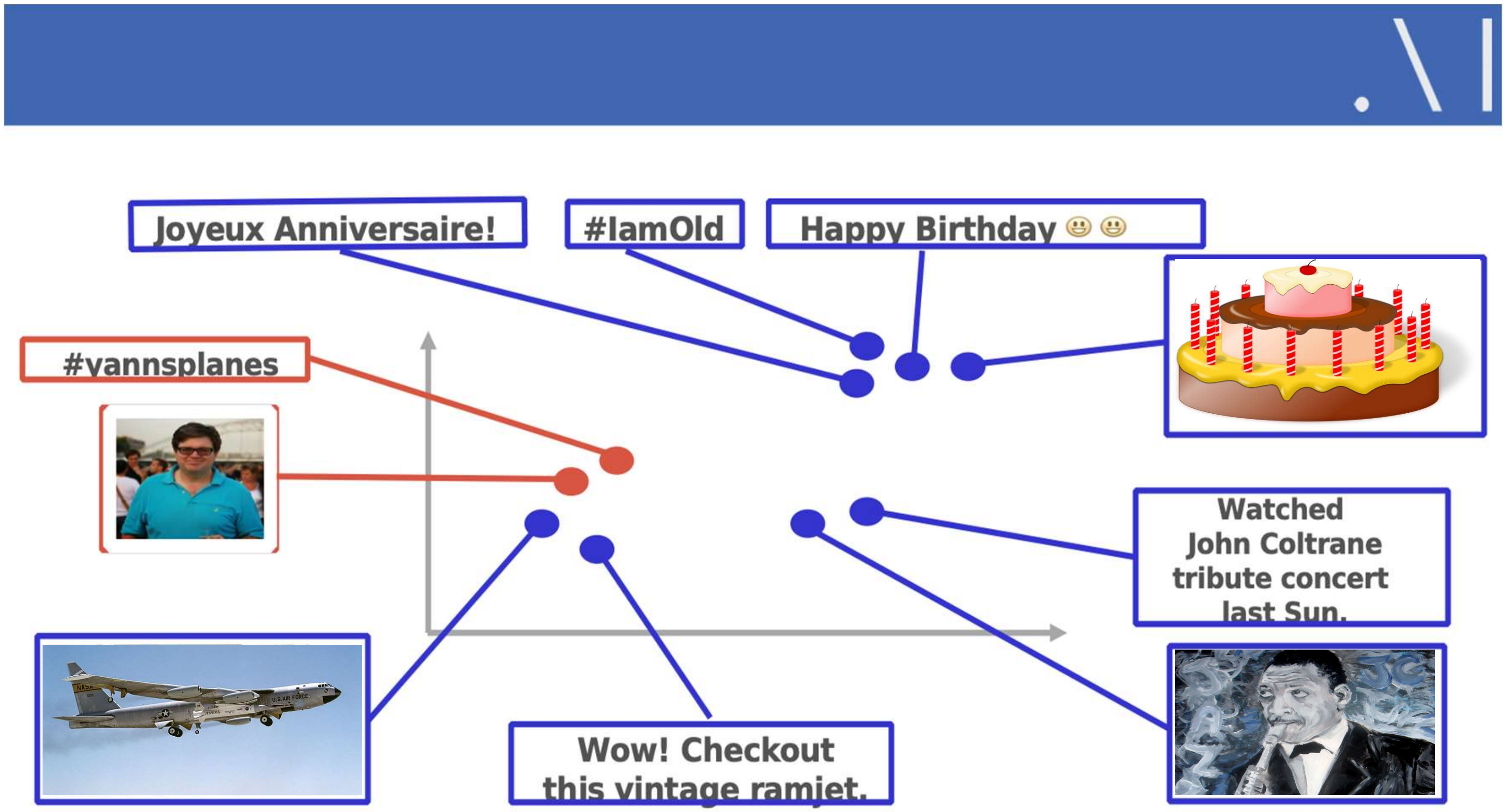


[0.2, -2.1, 0.4, -0.5, …]

Neural Net

“The neighbors' dog was a Samoyed, which looks a lot like a Siberian husky”

Slide Credit: Yann LeCun

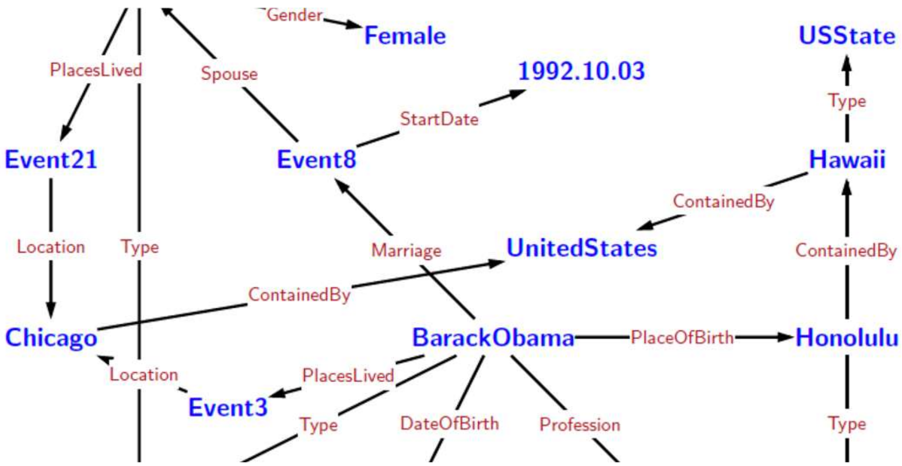


Slide Credit: Yann LeCun

(Big) Graph Data is Everywhere

Knowledge Graphs

Standard domain for studying graph embeddings *(Freebase, …)*



Wang, Zhenghao & Yan, Shengquan & Wang, Huaming & Huang, Xuedong. (2014). An Overview of Microsoft Deep QA System on Stanford WebQuestions Benchmark.

Recommender Systems

Deals with graph-like data, but supervised

	user_id	movie_id	rating
0	196	242	3

Social Graphs

Predict attributes based on homophily or structural similarity *(Twitter, Yelp, …)*

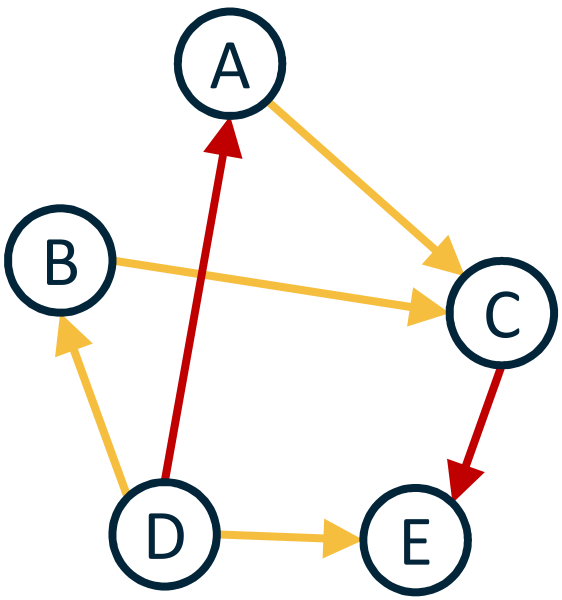
Slide Credit: Adam Lerer

Graph Embedding & Matrix Completion

	item1	item2	…	itemN
person1	-	+		+
person2	+	?		
…				
personP	+	-		?

- Relations between items (and people)
- Items in {people, movies, page, articles, products, word sequences…}
- Predict if someone will like an item, if a word will follow a word sequence

Slide Credit: Yann LeCun



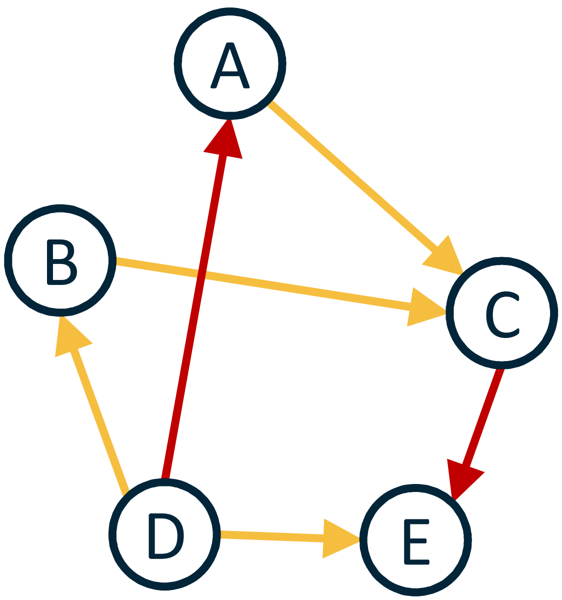
A multi-relation graph

Embedding: A learned map from entities to vectors of numbers that encodes similarity

- Word embeddings: word → vector
- Graph embeddings: node → vector

Graph Embedding: Optimize the objective that **connected nodes have more similar embeddings** than unconnected nodes via gradient descent.

Slide Credit: Adam Lerer



A multi-relation graph

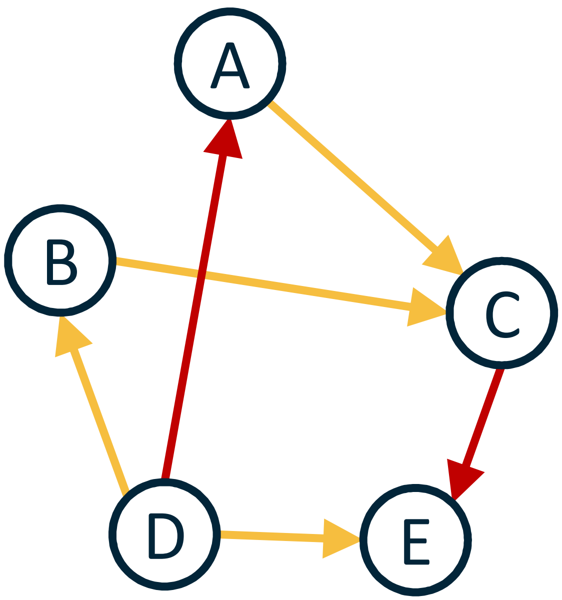
Why Graph Embeddings?

Graph embeddings are a form of **unsupervised learning** on graphs.

- **Task-agnostic** entity representations
- Features are useful on downstream tasks without much data
- Nearest neighbors are semantically meaningful

Slide Credit: Adam Lerer

PyTorch BigGraph



A multi-relation graph

Margin loss between the score for an edge $f(e)$ and a negative sampled edge $f(e')$

$$\mathcal{L} = \sum_{e \in G} \sum_{e' \in S'_e} \max(f(e) - f(e') + \lambda, 0))$$

The score for an edge is a similarity (e.g. dot product) between the source embedding and a transformed version of the destination embedding, e.g.

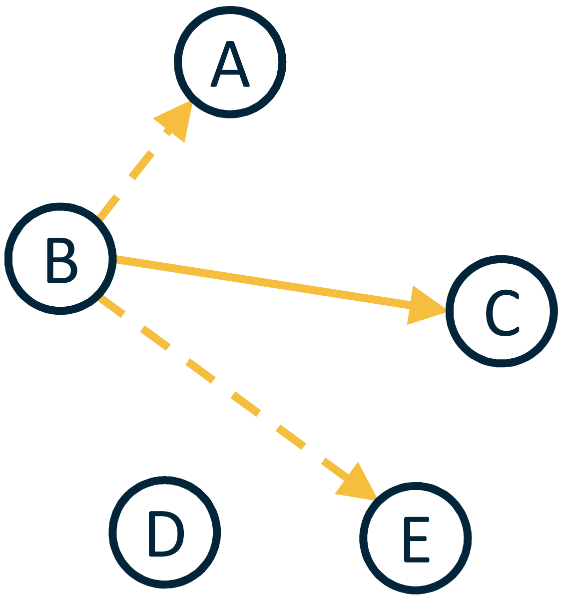
$$f(e) = \cos(\theta_s, \theta_r + \theta_d)$$

Negative samples are constructed by taking a real edge and replacing the source or destination with a random node.

$$S'_e = \{(s', r, d) | s' \in V\} \cup \{(s, r, d' | d' \in V\}$$

Slide Credit: Adam Lerer

PyTorch BigGraph



A multi-relation graph

Margin loss between the score for an edge $f(e)$ and a negative sampled edge $f(e')$

$$\mathcal{L} = \sum_{e \in G} \sum_{e' \in S'_e} \max(f(e) - f(e') + \lambda, 0))$$

The score for an edge is a similarity (e.g. dot product) between the source embedding and a transformed version of the destination embedding, e.g.

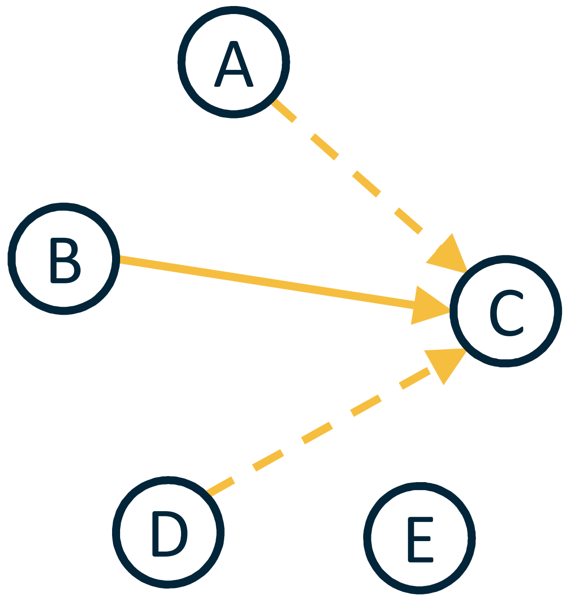
$$f(e) = \cos(\theta_s, \theta_r + \theta_d)$$

Negative samples are constructed by taking a real edge and replacing the source or destination with a random node.

$$S'_e = \{(s', r, d) | s' \in V\} \cup \{(s, r, d' | d' \in V\}$$

Slide Credit: Adam Lerer

PyTorch BigGraph



A multi-relation graph

Margin loss between the score for an edge $f(e)$ and a negative sampled edge $f(e')$

$$\mathcal{L} = \sum_{e \in G} \sum_{e' \in S'_e} \max(\boxed{f(e)} - f(e') + \lambda, 0))$$

The score for an edge is a similarity (e.g. dot product) between the source embedding and a transformed version of the destination embedding, e.g.

$$f(e) = \cos(\theta_s, \theta_r + \theta_d)$$

Negative samples are constructed by taking a real edge and replacing the source or destination with a random node.

$$S'_e = \{(s', r, d) | s' \in V\} \cup \{(s, r, d' | d' \in V\}$$

Slide Credit: Adam Lerer

Multiple Relations in Graphs

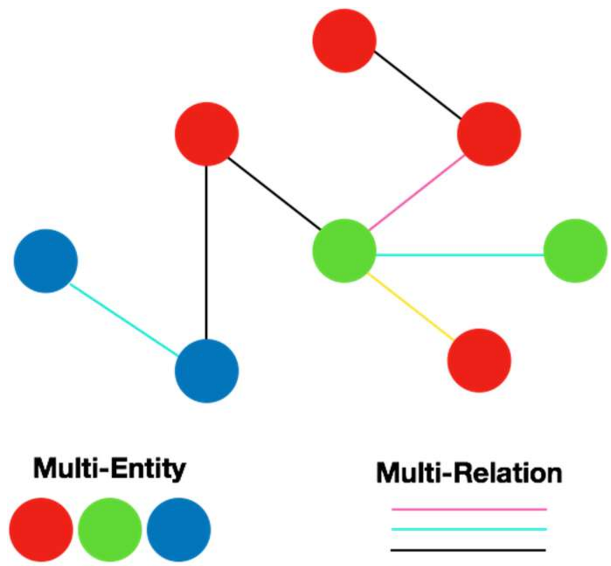


Figure Credit: Alex Peysakhovich

- **Identity:** $g(x) = x$
- **Translator:** $g(x|\Delta) = x + \Delta$
  [Bordes et al. 13']
- **Affine:** $g(x|A, \Delta) = Ax + \Delta$
  [Nickel et al., 11']
- **Diagonal:** $g(x|b) = b \odot x$
  [Yang et al., 15']

TagSpace

Input: restaurant has great food

Label: #yum, #restaurant

Use-cases:

- Labeling posts
- Clustering of hashtags

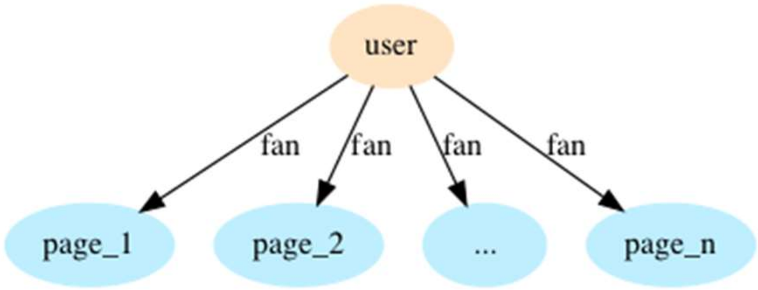
Reference: [Weston et al. 14'], [Wu et al. 18']
https://github.com/facebookresearch/StarSpace

PageSpace

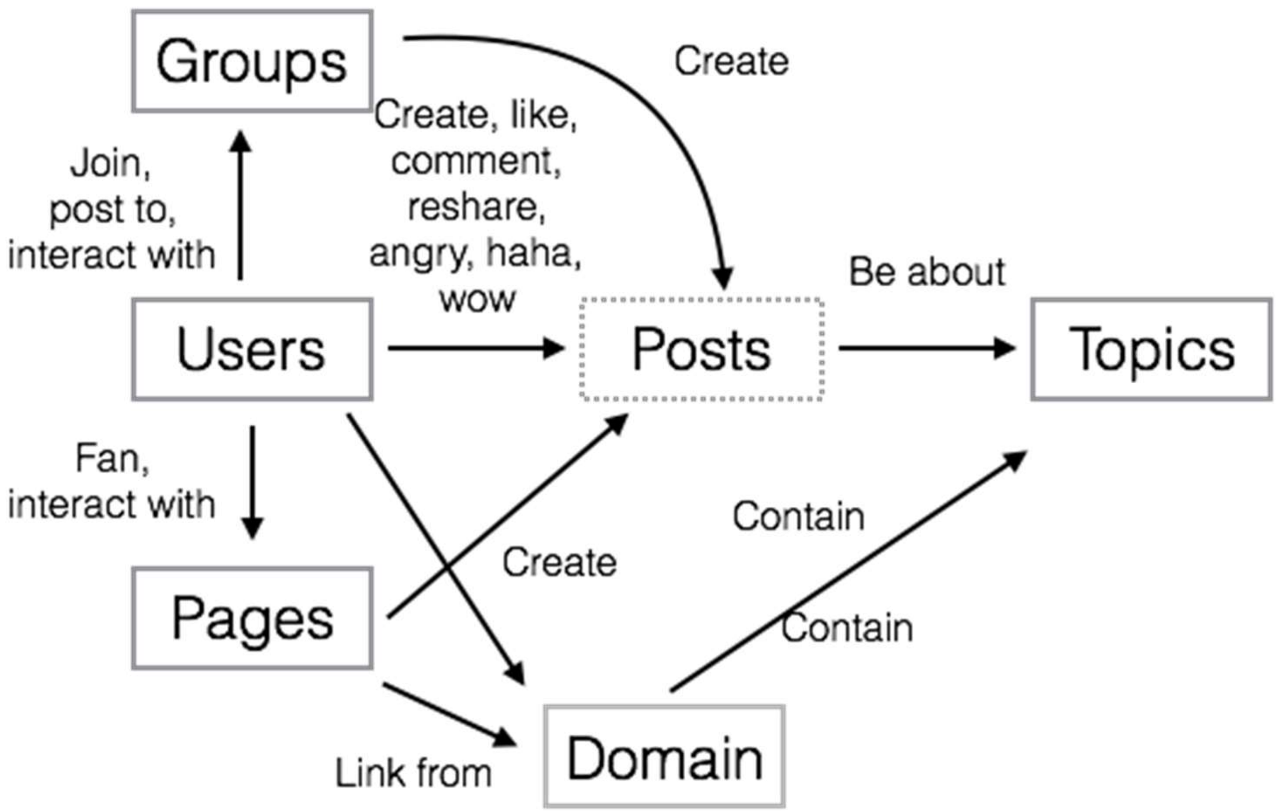
Input: (user, page) pairs

Use-cases:

- Clustering of pages
- Recommending pages to users



Application: TagSpace, PageSpace



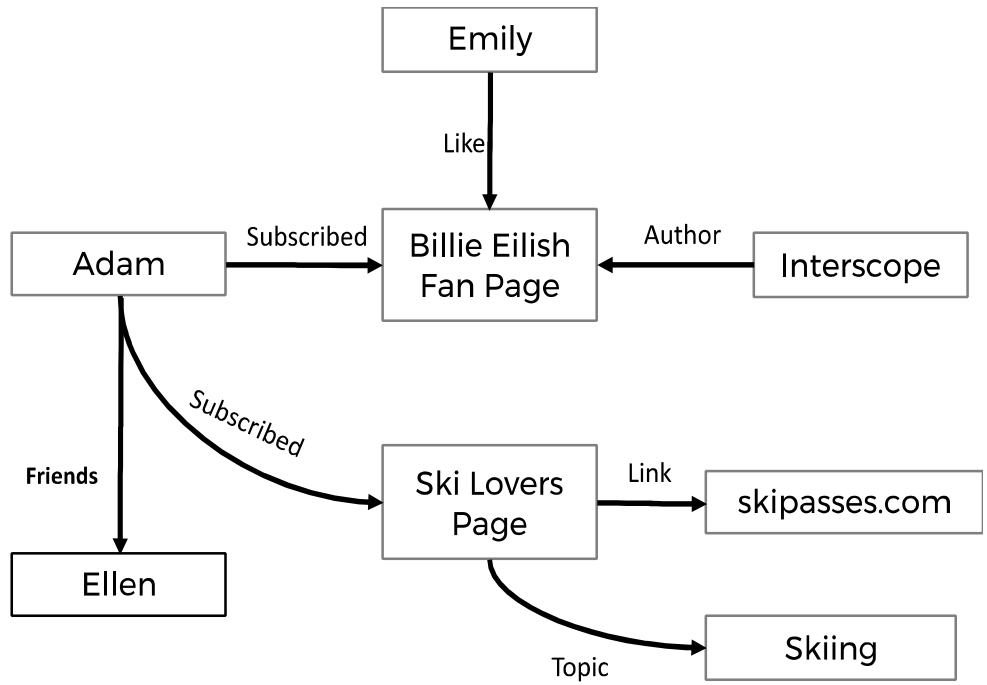
Slide Credit: Alex Peysakhovich

Application: world2vec

FACEBOOK AI Georgia Tech

The Power of Universal Behavioral Features

- What pages or topics might you be interested in?
- Which posts contain misinformation, hate speech, election interference, …?
- Is a person's account fake / hijacked?
- What songs might you like? (even if you've never provided any song info)

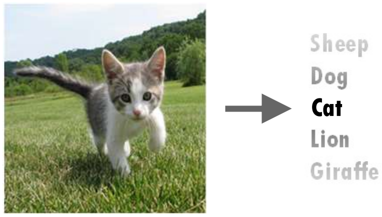


Slide Credit: Adam Lerer

Reinforcement Learning Introduction

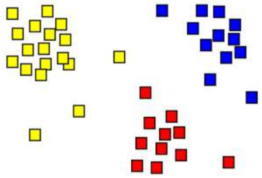
Supervised Learning

- Train Input: $\{X, Y\}$
- Learning output: $f : X \rightarrow Y$, $P(y|x)$
- e.g. classification



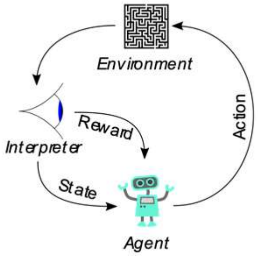
Unsupervised Learning

- Input: $\{X\}$
- Learning output: $P(x)$
- Example: Clustering, density estimation, etc.



Reinforcement Learning

- Evaluative feedback in the form of **reward**
- No supervision on the right action



Types of Machine Learning

RL: Sequential decision making in an environment with evaluative feedback.

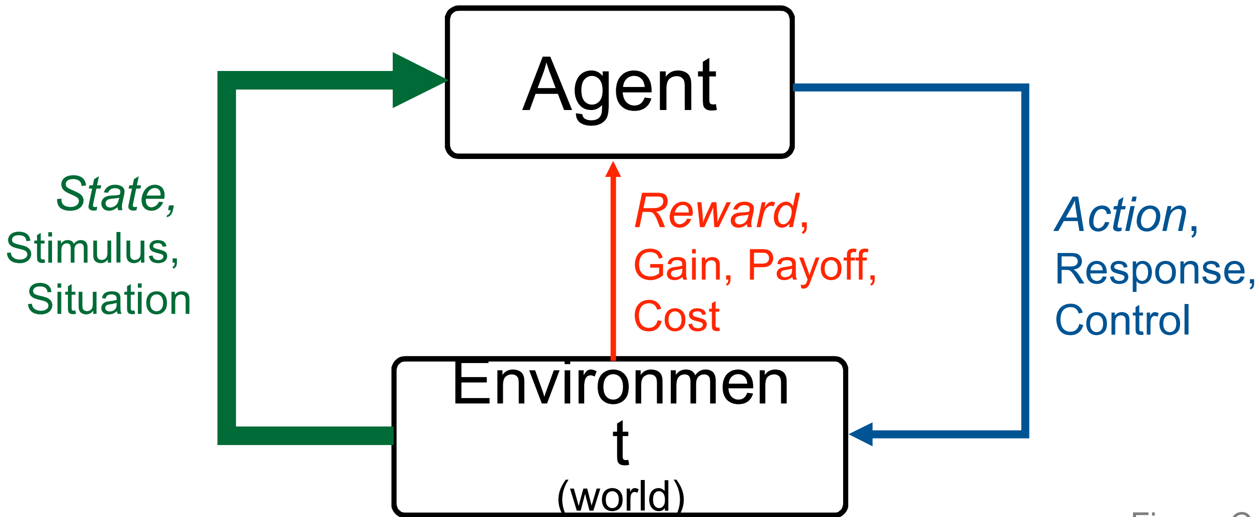


Figure Credit: Rich Sutton

- **Environment** may be unknown, non-linear, stochastic and complex.
- **Agent** learns a **policy** to map states of the environments to actions.
  - Seeking to maximize cumulative reward in the long run.

RL: Sequential decision making in an environment with evaluative feedback.

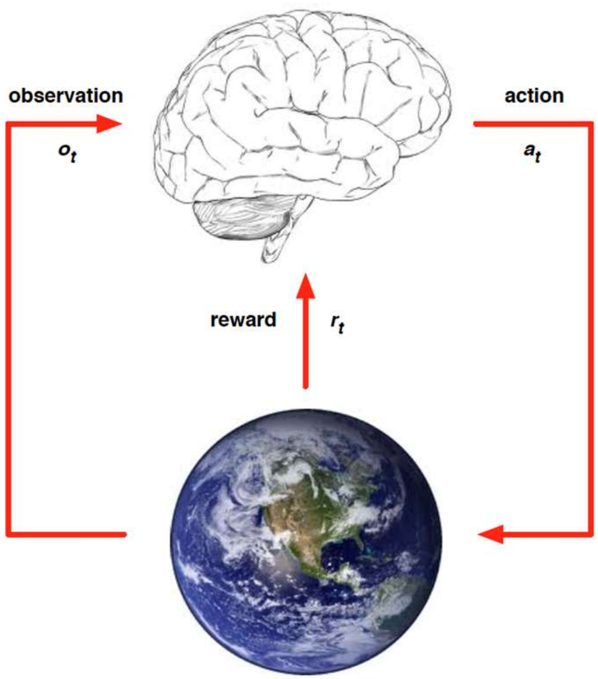
Evaluative Feedback

- Pick an action, receive a reward (positive or negative)
- No supervision for what the “correct” action is or would have been, unlike supervised learning

Sequential Decisions

- Plan and execute actions over a sequence of states
- Reward may be delayed, requiring optimization of future rewards (long-term planning).

RL: Environment Interaction API



- At each time step t, the agent:
  - Receives observation o_t
  - Executes action a_t
- At each time step t, the environment:
  - Receives action a_t
  - Emits observation o_{t+1}
  - Emits scalar reward r_{t+1}

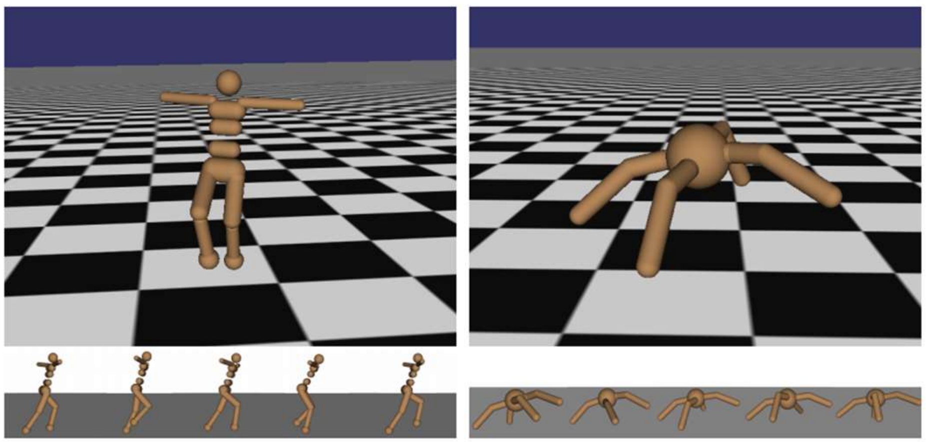
Slide credit: David Silver

Signature Challenges in Reinforcement Learning

- Evaluative feedback: Need trial and error to find the right action
- Delayed feedback: Actions may not lead to immediate reward
- Non-stationarity: Data distribution of visited states changes when the policy changes
- Fleeting nature of time and online data

Slide adapted from: Richard Sutton

Robot Locomotion

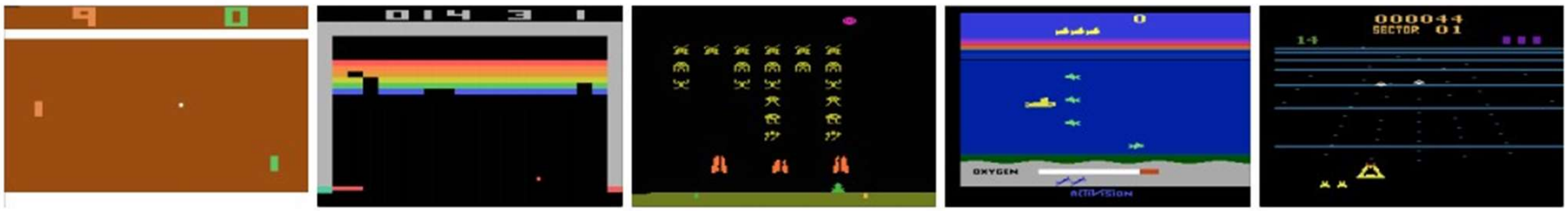


Figures copyright John Schulman et al., 2016. Reproduced with permission.

- **Objective**: Make the robot move forward
- **State**: Angle and position of the joints
- **Action**: Torques applied on joints
- **Reward**: +1 at each time step upright and moving forward

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

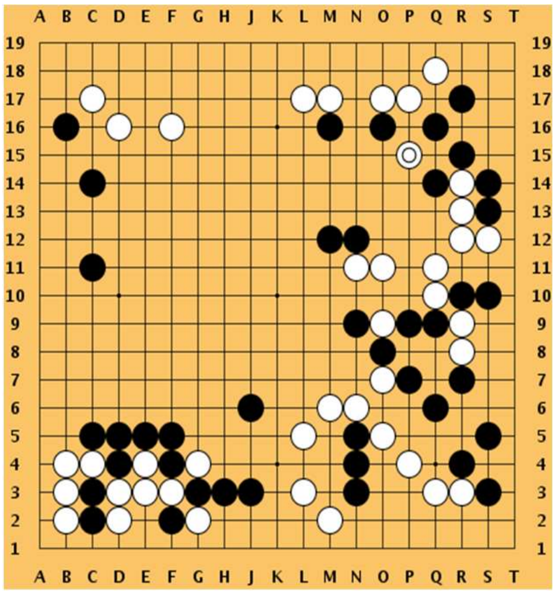
Atari Games



- **Objective**: Complete the game with the highest score
- **State**: Raw pixel inputs of the game state
- **Action**: Game controls e.g. Left, Right, Up, Down
- **Reward**: Score increase/decrease at each time step

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Go



- **Objective**: Defeat opponent
- **State**: Board pieces
- **Action**: Where to put next piece down
- **Reward**: +1 if win at the end of game, 0 otherwise

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Markov Decision Processes

- **MDPs**: Theoretical framework underlying RL

Markov Decision Processes (MDPs)

- **MDPs**: Theoretical framework underlying RL
- An MDP is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbb{T}, \gamma)$
  - $\mathcal{S}$: Set of possible states
  - $\mathcal{A}$: Set of possible actions
  - $\mathcal{R}(s, a, s')$: Distribution of reward
  - $\mathbb{T}(s, a, s')$: Transition probability distribution, also written as p(s'|s,a)
  - γ : Discount factor

Markov Decision Processes (MDPs)

Georgia Tech

- **MDPs**: Theoretical framework underlying RL
- An MDP is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbb{T}, \gamma)$
  - $\mathcal{S}$: Set of possible states
  - $\mathcal{A}$: Set of possible actions
  - $\mathcal{R}(s, a, s')$: Distribution of reward
  - $\mathbb{T}(s, a, s')$: Transition probability distribution, also written as p(s'|s,a)
  - γ : Discount factor
- **Interaction trajectory**: $\ldots s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}, r_{t+2}, s_{t+2}, \ldots$

Markov Decision Processes (MDPs)

Georgia Tech

- **MDPs**: Theoretical framework underlying RL
- An MDP is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbb{T}, \gamma)$
  - $\mathcal{S}$: Set of possible states
  - $\mathcal{A}$: Set of possible actions
  - $\mathcal{R}(s, a, s')$: Distribution of reward
  - $\mathbb{T}(s, a, s')$: Transition probability distribution, also written as p(s'|s,a)
  - γ : Discount factor
- **Interaction trajectory**: $\ldots s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}, r_{t+2}, s_{t+2}, \ldots$
- **Markov property**: Current state completely characterizes state of the environment
- **Assumption**: Most recent observation is a sufficient statistic of history

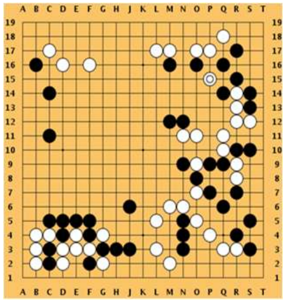
$$p\left(S_{t+1} = s' | S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, \ldots S_0 = s_0\right) = p\left(S_{t+1} = s' | S_t = s_t, A_t = a_t\right)$$

Markov Decision Processes (MDPs)

Georgia Tech

Fully observed MDP

- Agent receives the true state s_t at time t
- Example: Chess, Go



Partially observed MDP

- Agent perceives its own partial observation o_t of the state s_t at time t, using past states e.g. with an RNN
- Example: Poker, First-person games (e.g. Doom)

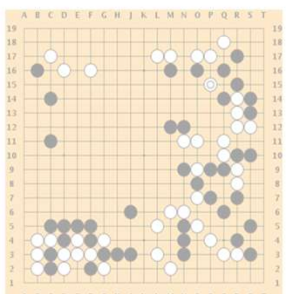
Source: https://github.com/mwydmuch/ViZDoom

MDP Variations

Georgia Tech

Fully observed MDP

- Agent receives the true state s_t at time t
- Example: Chess, Go



Partially observed MDP

- Agent perceives its own partial observation o_t of the state s_t at time t, using past

Source: https://github.com/mwydmuch/ViZDoom

We will assume **fully observed MDPs** for this lecture

MDP Variations

Georgia Tech

- In **Reinforcement Learning**, we assume an underlying **MDP** with unknown:
  - Transition probability distribution $\mathbb{T}$
  - Reward distribution $\mathcal{R}$

MDP
$(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbb{T}, \gamma)$

MDPs in the context of RL

Georgia Tech

MDPs in the context of RL

- In **Reinforcement Learning**, we assume an underlying **MDP** with unknown:
  - Transition probability distribution $\mathbb{T}$
  - Reward distribution $\mathcal{R}$

MDP
$(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbb{T}, \gamma)$

- Evaluative feedback comes into play, trial and error necessary

- In **Reinforcement Learning**, we assume an underlying **MDP** with unknown:
  - Transition probability distribution $\mathbb{T}$
  - Reward distribution $\mathcal{R}$

MDP
$(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbb{T}, \gamma)$

- Evaluative feedback comes into play, trial and error necessary

- For this and next lecture, assume that we know the true reward and transition distribution and look at algorithms for **solving MDPs** i.e. finding the best policy
  - Rewards known everywhere, no evaluative feedback
  - Know how the world works i.e. all transitions

MDPs in the context of RL

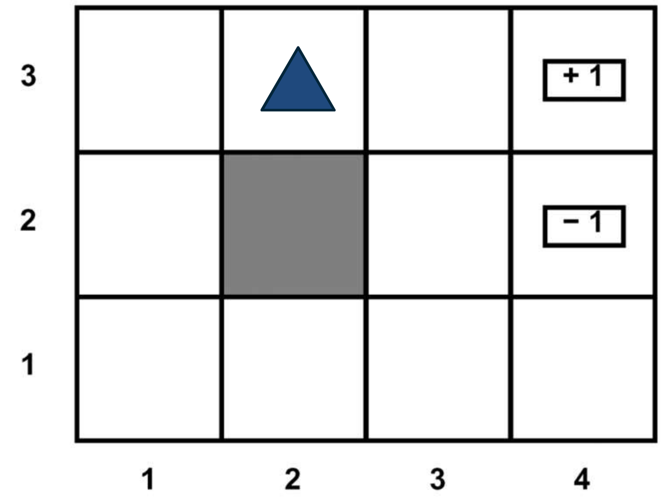


Figure credits: Pieter Abbeel

A Grid World MDP

- Agent lives in a 2D grid environment

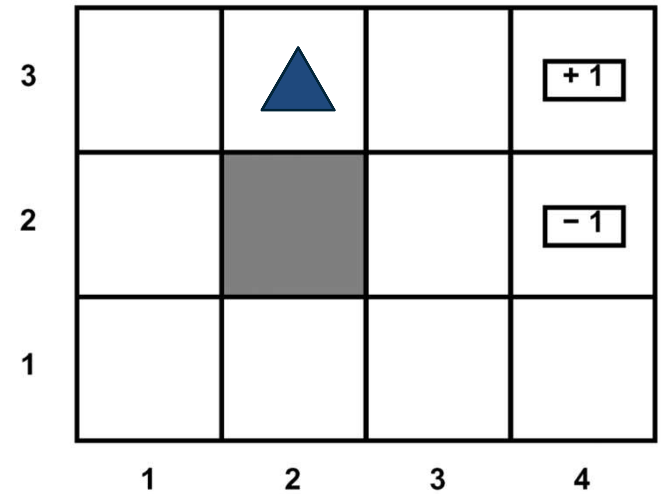


Figure credits: Pieter Abbeel

A Grid World MDP

- Agent lives in a 2D grid environment

- State: Agent's 2D coordinates
- Actions: N, E, S, W
- Rewards: +1/-1 at absorbing states

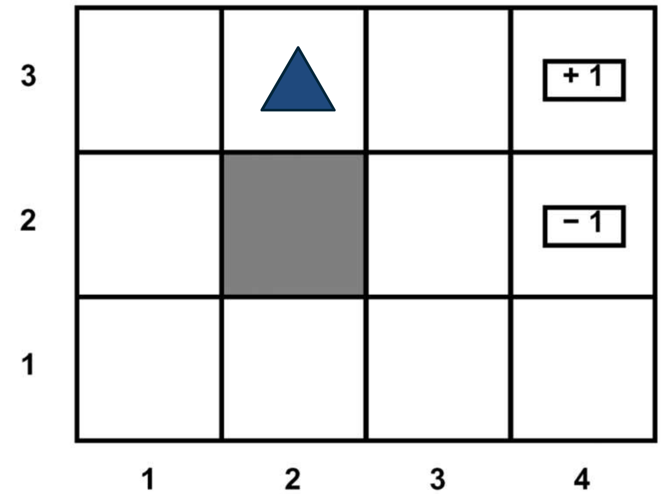
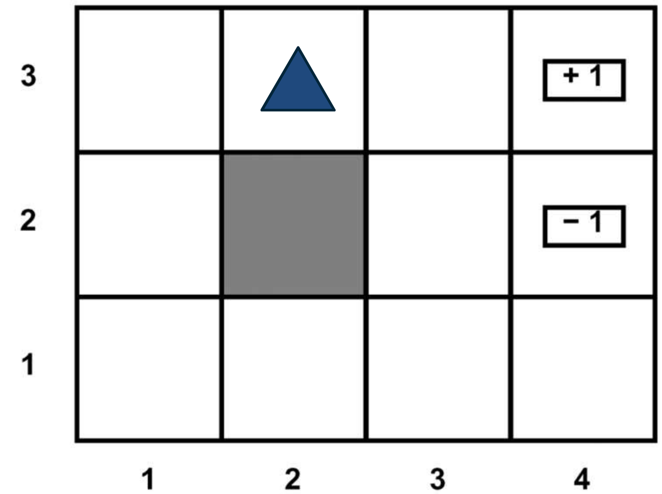


Figure credits: Pieter Abbeel

A Grid World MDP

A Grid World MDP

- Agent lives in a 2D grid environment
- State: Agent's 2D coordinates
- Actions: N, E, S, W
- Rewards: +1/-1 at absorbing states
- Walls block agent's path
- Actions to not always go as planned
  - 20% chance that agent drifts one cell left or right of direction of motion (except when blocked by wall).



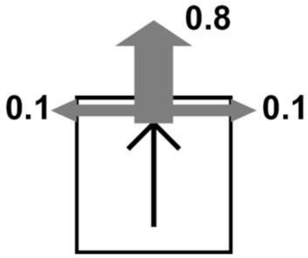


Figure credits: Pieter Abbeel