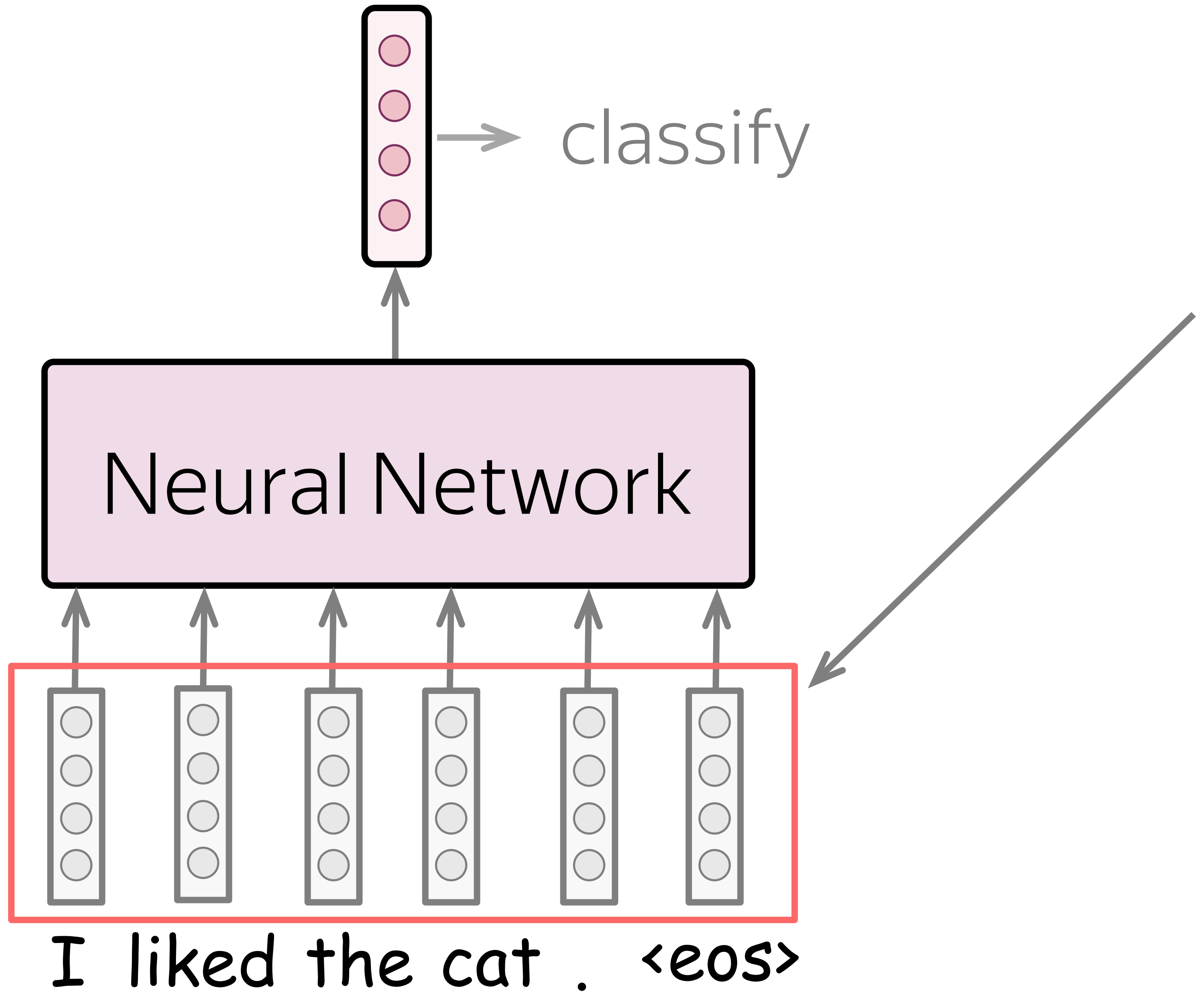
Transfer Learning

Lena Voita

What is going to happen:

- Transfer Learning Idea
- Pretrained Models
-  Analysis and Interpretability

Recap from Text Classification: Word Embeddings

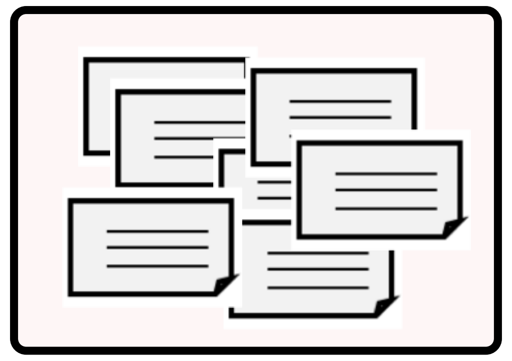


Input word embeddings:

- Train from scratch
- Take pretrained (Word2Vec, GloVe)
- Initialize with pretrained, then fine-tune

Which data do we have?

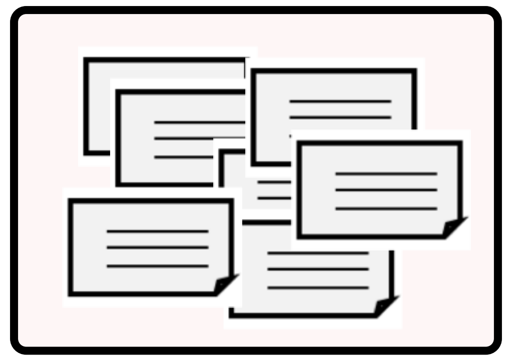
Training data for text classification (labeled)



- Not huge, or not diverse, or both
- Domain: task-specific

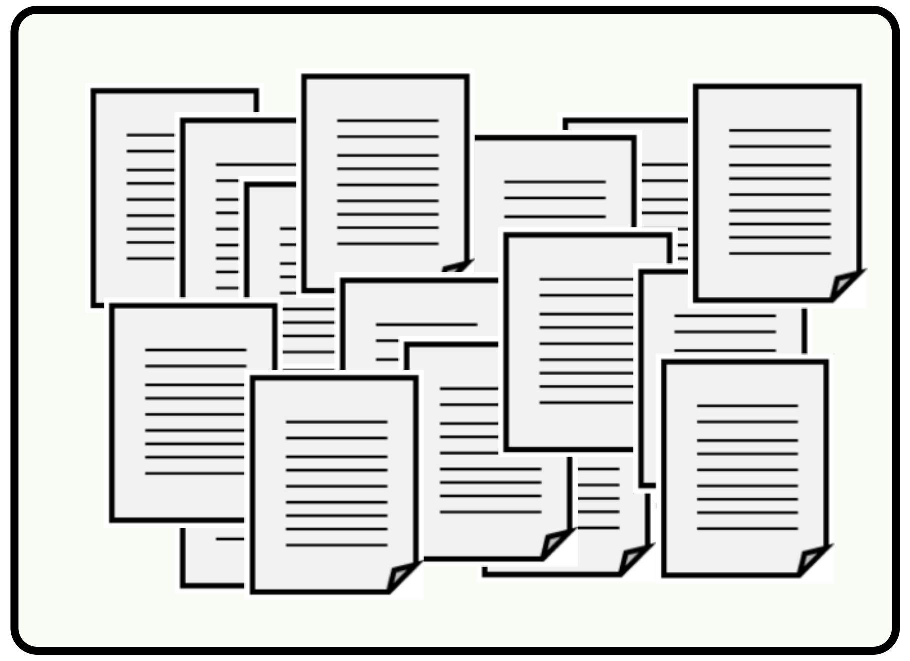
Which data do we have?

Training data for text classification (labeled)



- Not huge, or not diverse, or both
- Domain: task-specific

Training data for word embeddings (unlabeled)



- Huge diverse corpus (e.g., Wikipedia)
- Domain: general

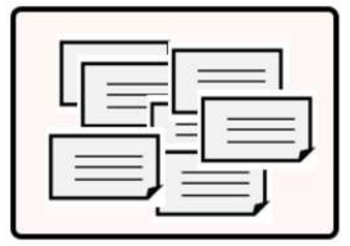
Recap from Text Classification: Word Embeddings

- Train from scratch
- Take pretrained (Word2Vec, GloVe)
- Initialize with pretrained, then fine-tune

Recap from Text Classification: Word Embeddings

- Train from scratch

  What they will know:

  May be not enough to learn relationships between words

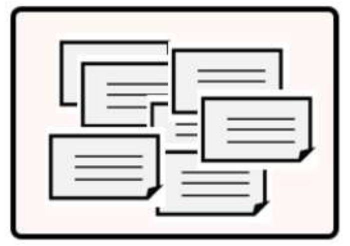
- Take pretrained (Word2Vec, GloVe)

- Initialize with pretrained, then fine-tune

Recap from Text Classification: Word Embeddings

- Train from scratch

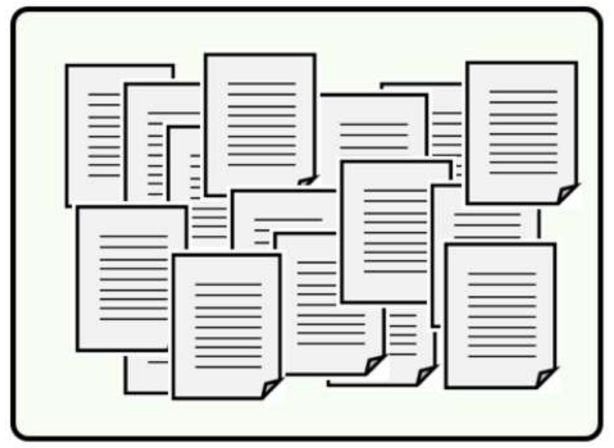
  What they will know:

  May be not enough to learn relationships between words

- Take pretrained (Word2Vec, GloVe)

  What they will know:

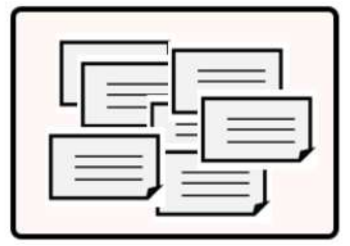
  Know relationships between words, but are **not** specific to the task

- Initialize with pretrained, then fine-tune

Recap from Text Classification: Word Embeddings

- Train from scratch

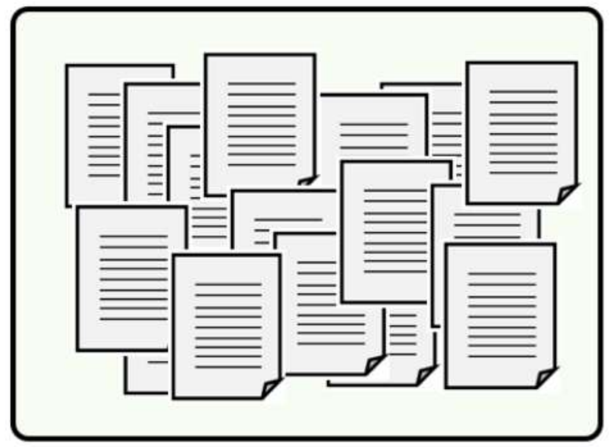
What they will know:



May be not enough to learn relationships between words

- Take pretrained (Word2Vec, GloVe)

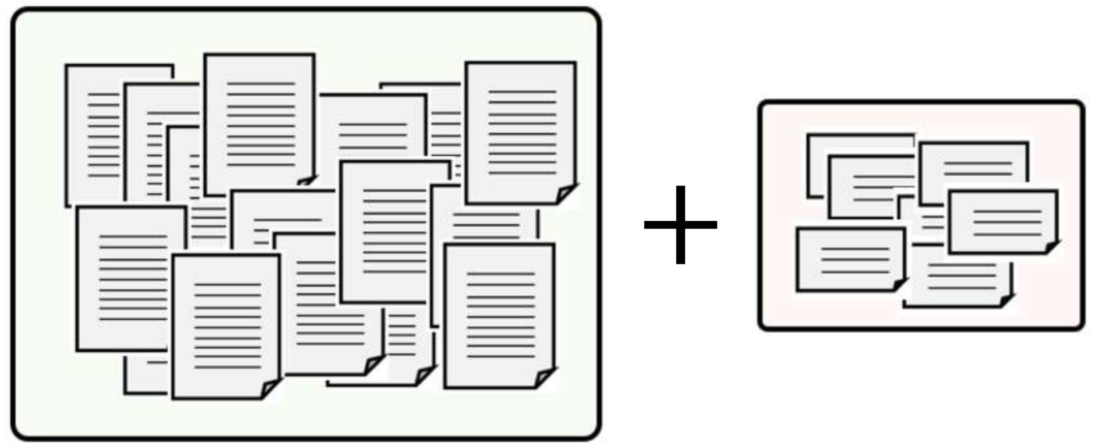
What they will know:



Know relationships between words, but are **not** specific to the task

- **Initialize with pretrained, then fine-tune**

What they will know:

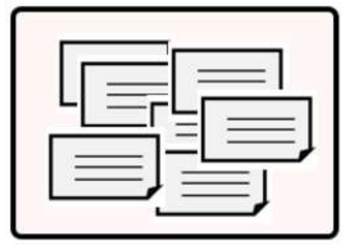


Know relationships between words and adapted for the task

Recap from Text Classification: Word Embeddings

- Train from scratch

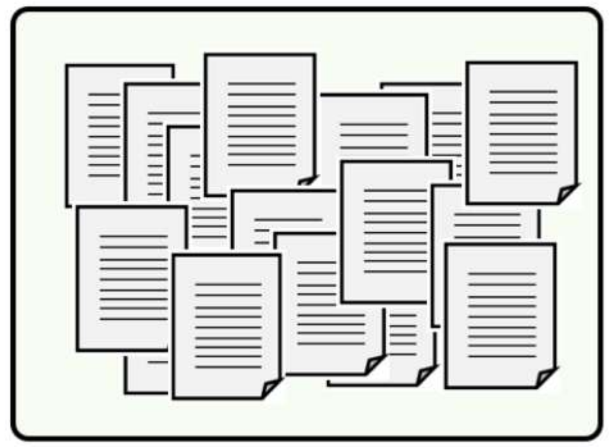
What they will know:



May be not enough to learn relationships between words

- Take pretrained (Word2Vec, GloVe)

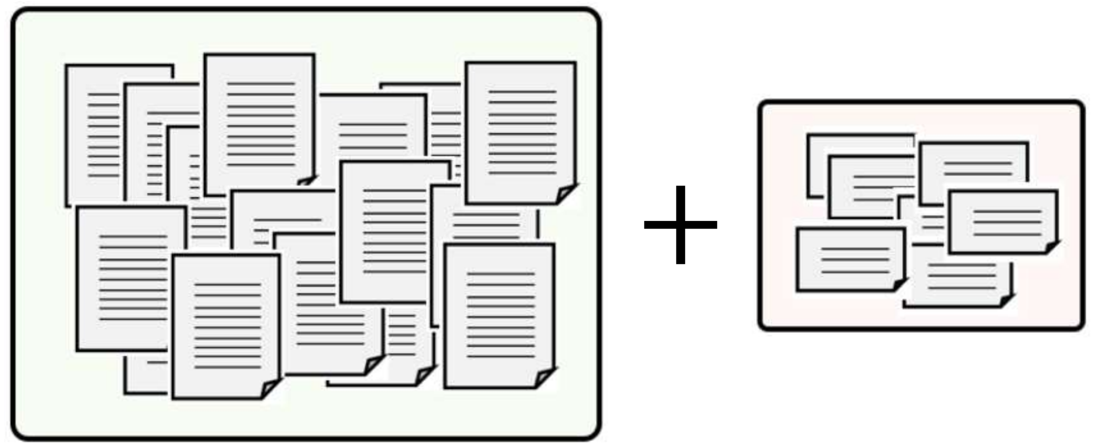
What they will know:



Know relationships between words, but are **not** specific to the task

- **Initialize with pretrained, then fine-tune**

What they will know:



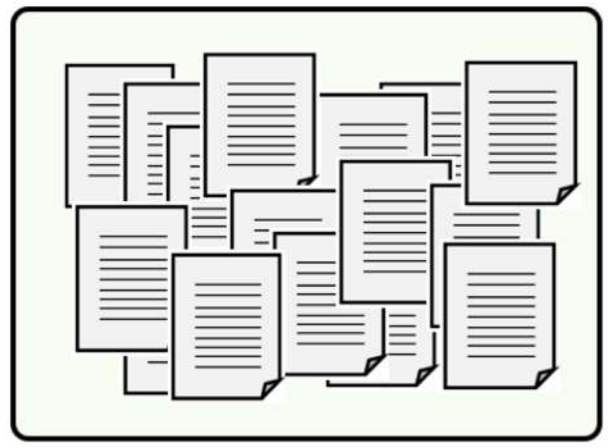
Know relationships between words and adapted for the task

"**Transfer**" knowledge from a huge unlabeled corpus to your task-specific model

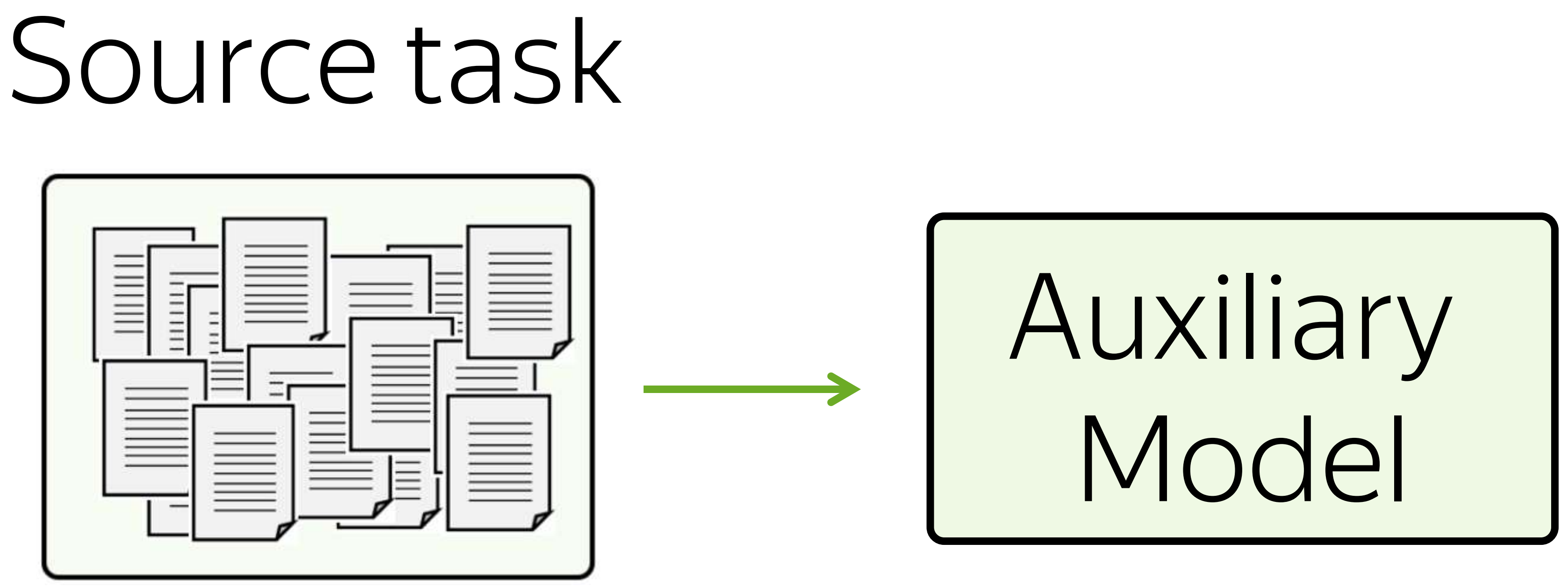
We'll learn more about this later in the course!

Transfer Learning Idea

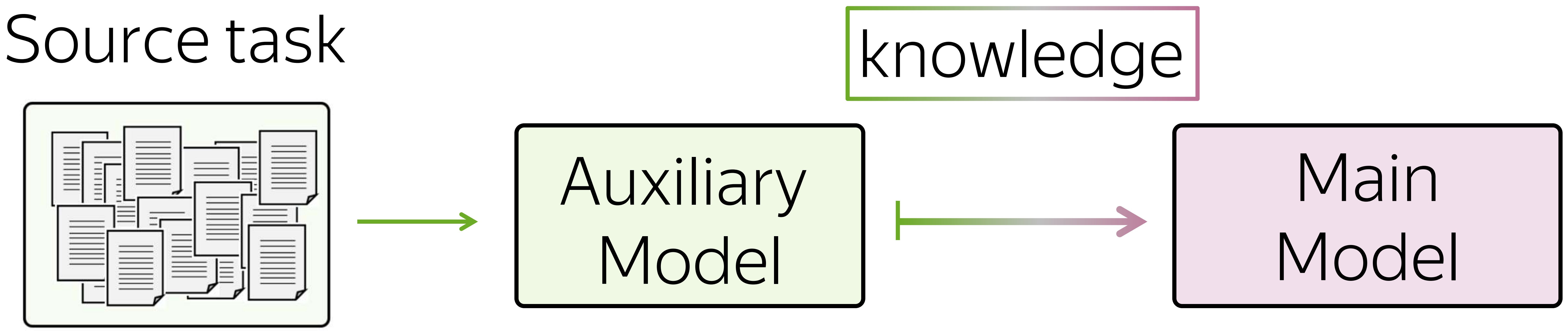
Source task



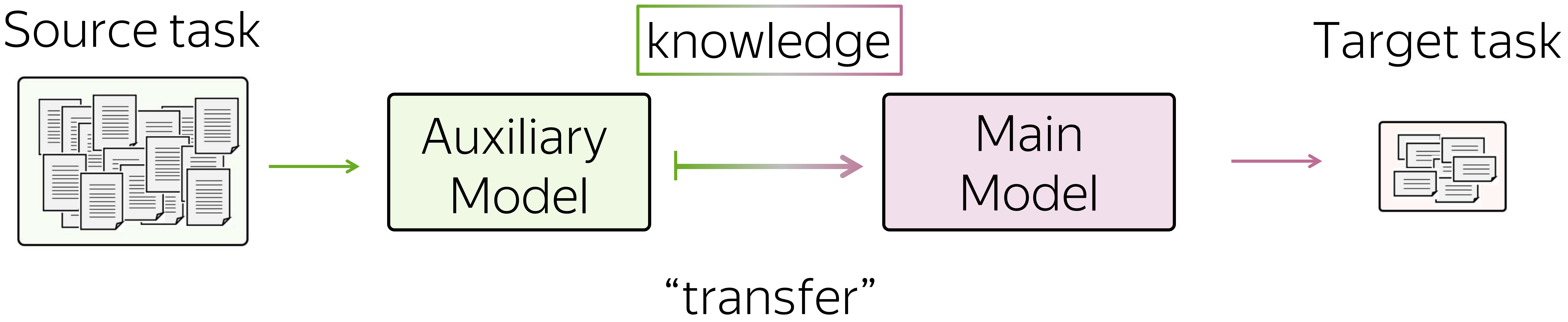
Transfer Learning Idea



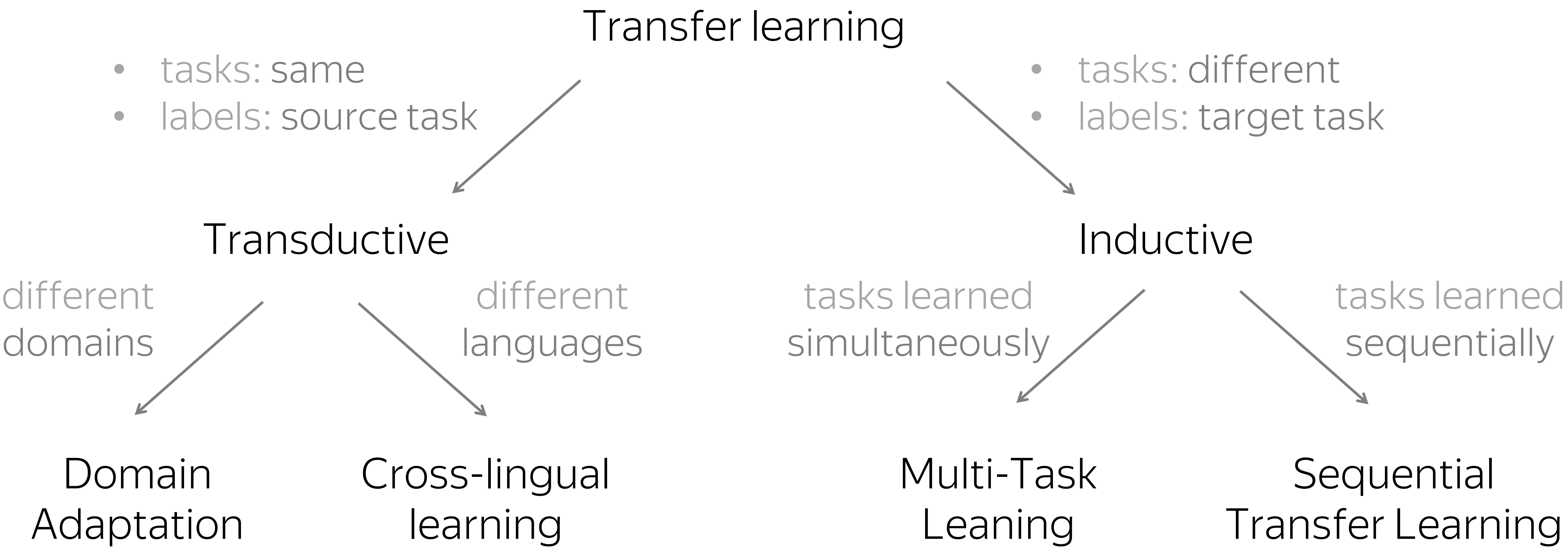
Transfer Learning Idea



Transfer Learning Idea

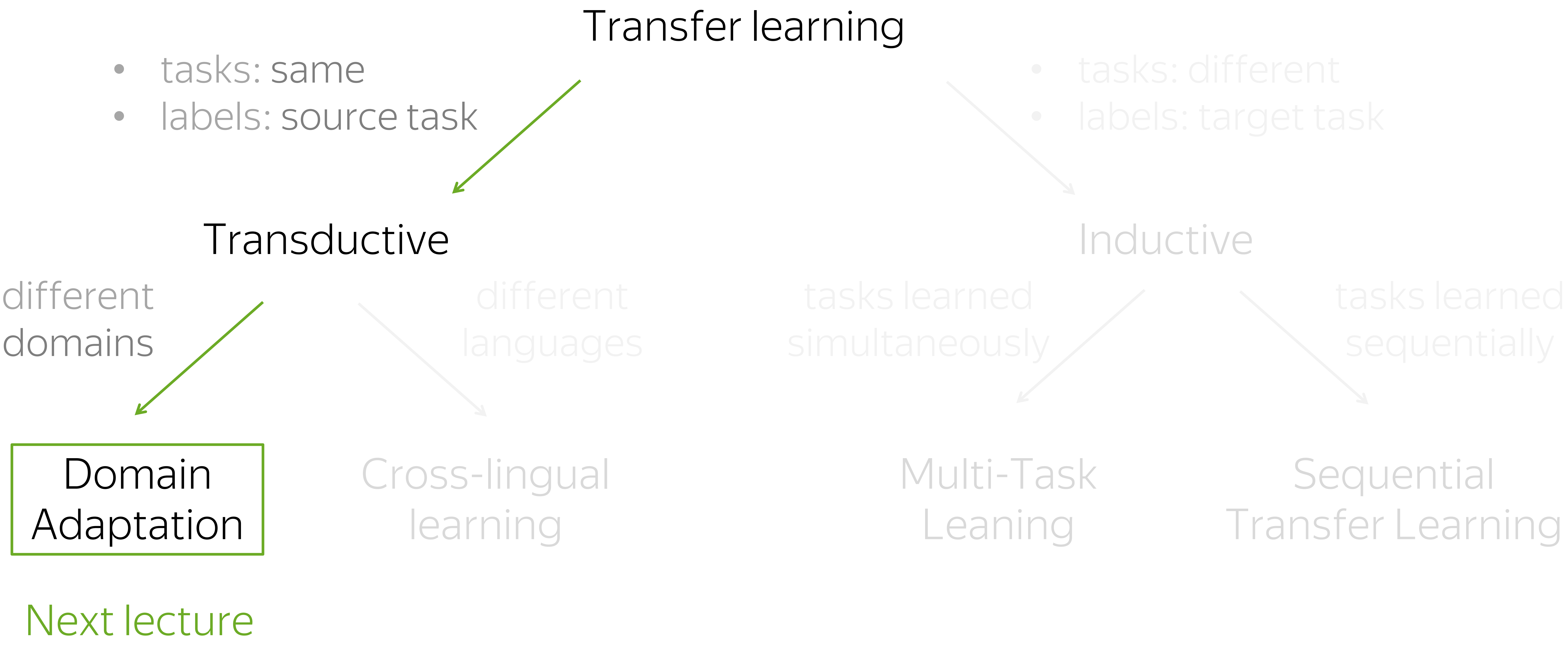


A Taxonomy of Transfer Learning in NLP



This taxonomy is from Ruder, 2019

A Taxonomy of Transfer Learning in NLP



This taxonomy is from Ruder, 2019

A Taxonomy of Transfer Learning in NLP

Transfer learning

- tasks: same
- labels: source task

Transductive

different domains

different languages

Domain Adaptation

Cross-lingual learning

- tasks: different
- labels: target task

Inductive

tasks learned simultaneously

tasks learned sequentially

Multi-Task Leaning

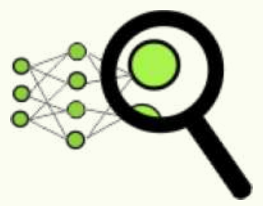
Sequential Transfer Learning

(the most popular)

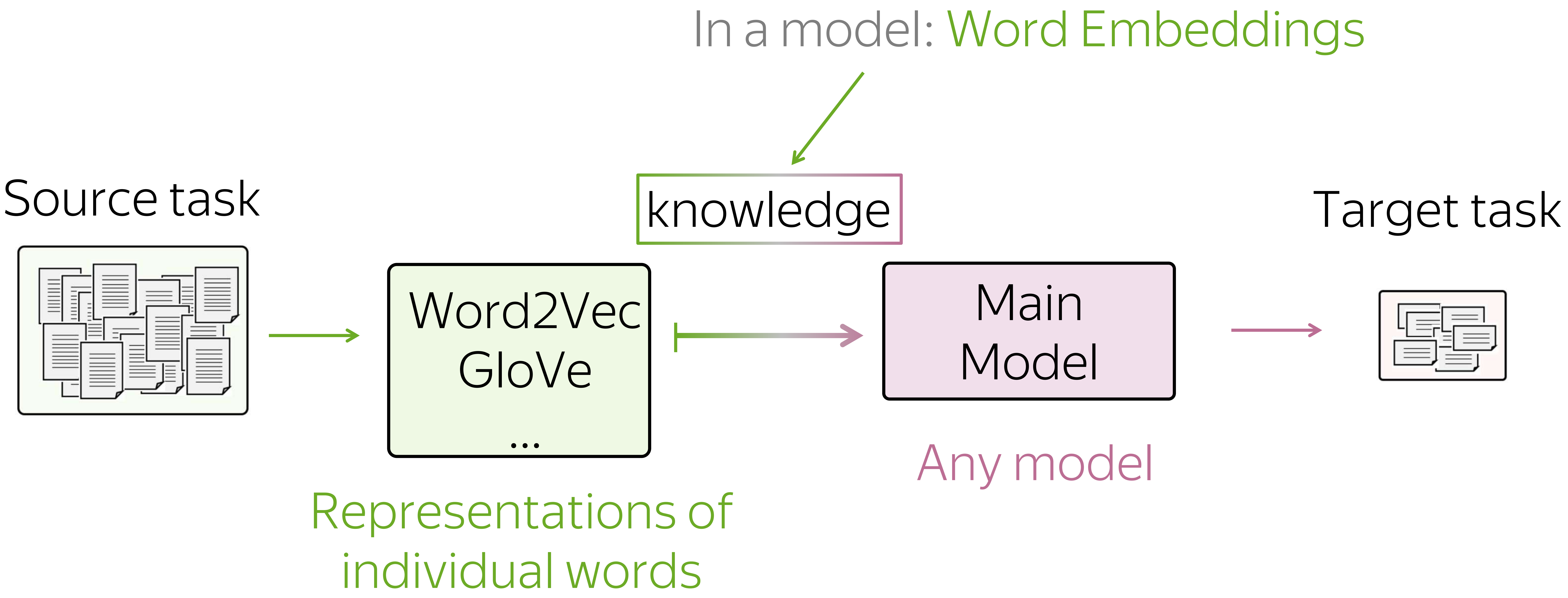
Today!

This taxonomy is from Ruder, 2019

What is going to happen:

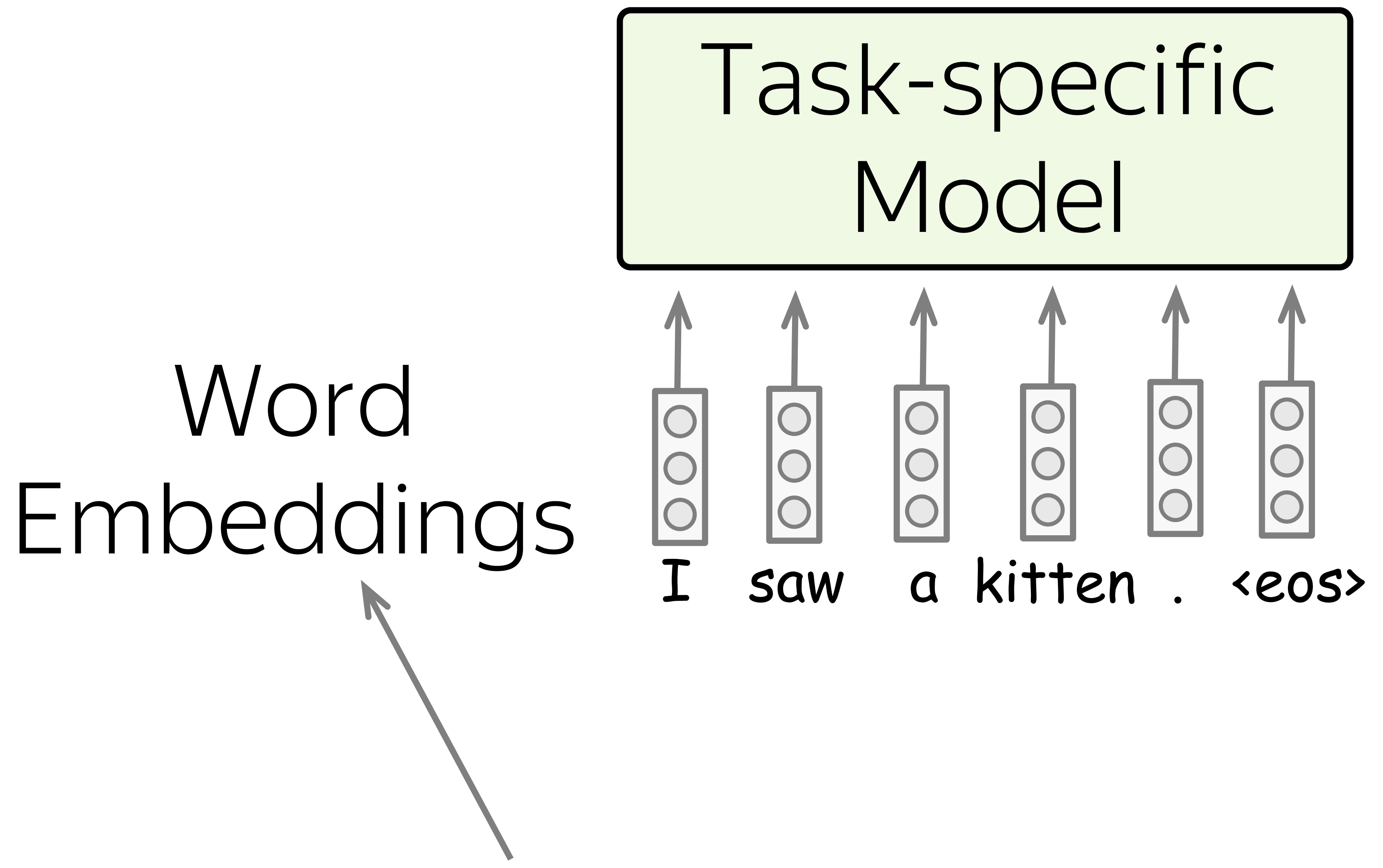
- Transfer Learning Idea
- Pretrained Models
    - (recap) Word Embeddings
    - ELMo
    - BERT
    - (a note on) GPT
    - (a note on) Adaptors
-  Analysis and Interpretability

Simplest (recap once again):
Word Embeddings (Word2Vec, GloVe)



Transfer Through Word Embedding

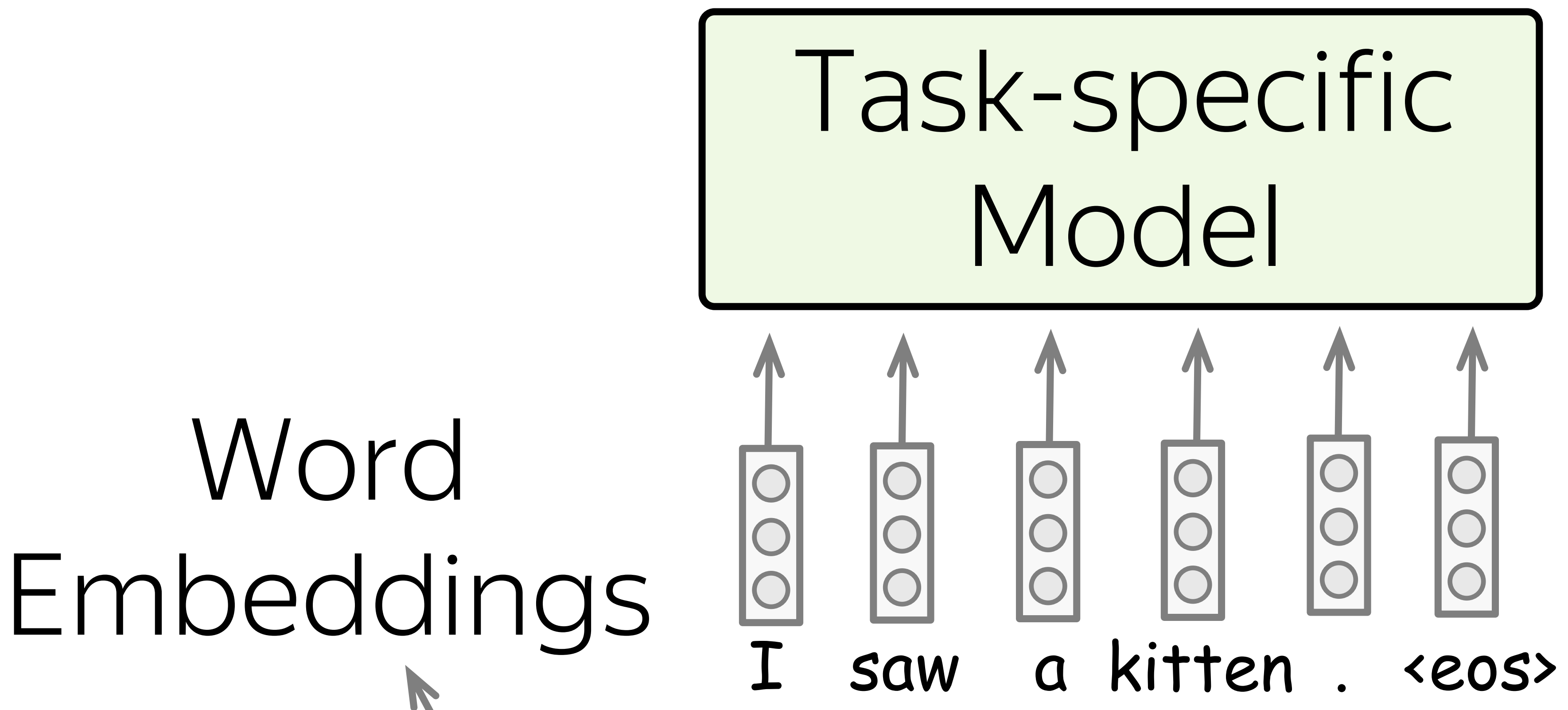
Before



Know nothing, trained from scratch with the model

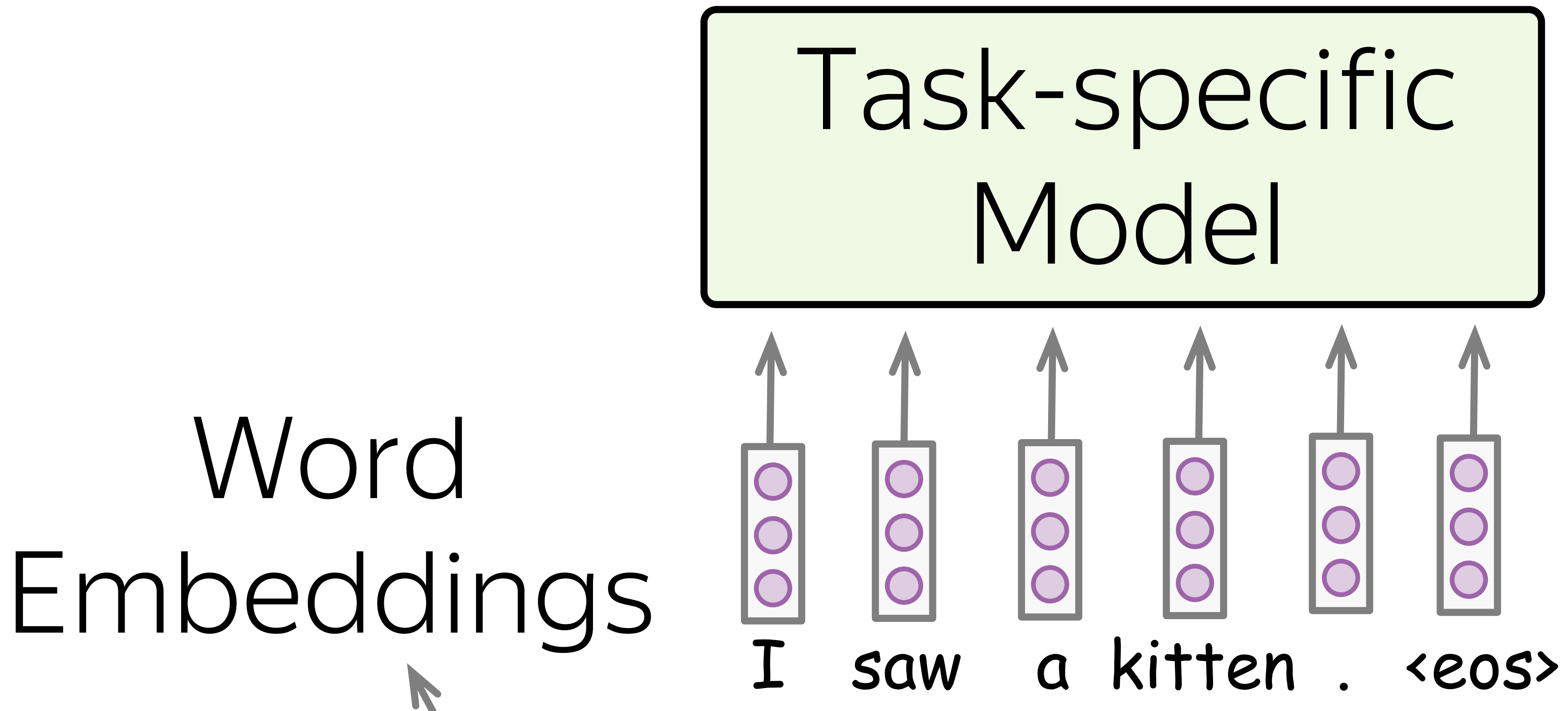
Transfer Through Word Embedding

Before



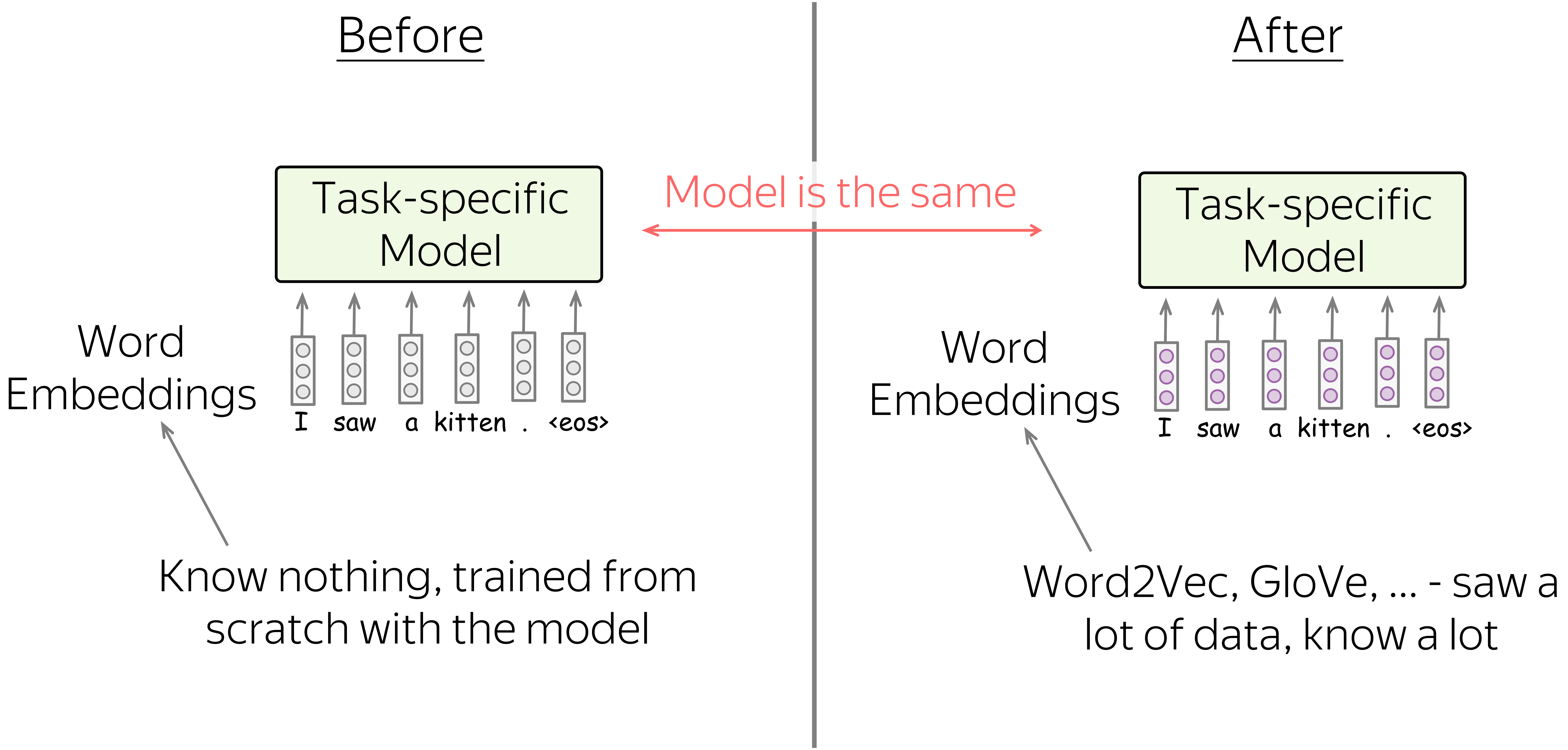
Know nothing, trained from scratch with the model

After

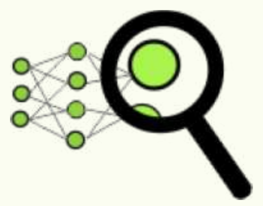


Word2Vec, GloVe, ... - saw a lot of data, know a lot

Transfer Through Word Embedding



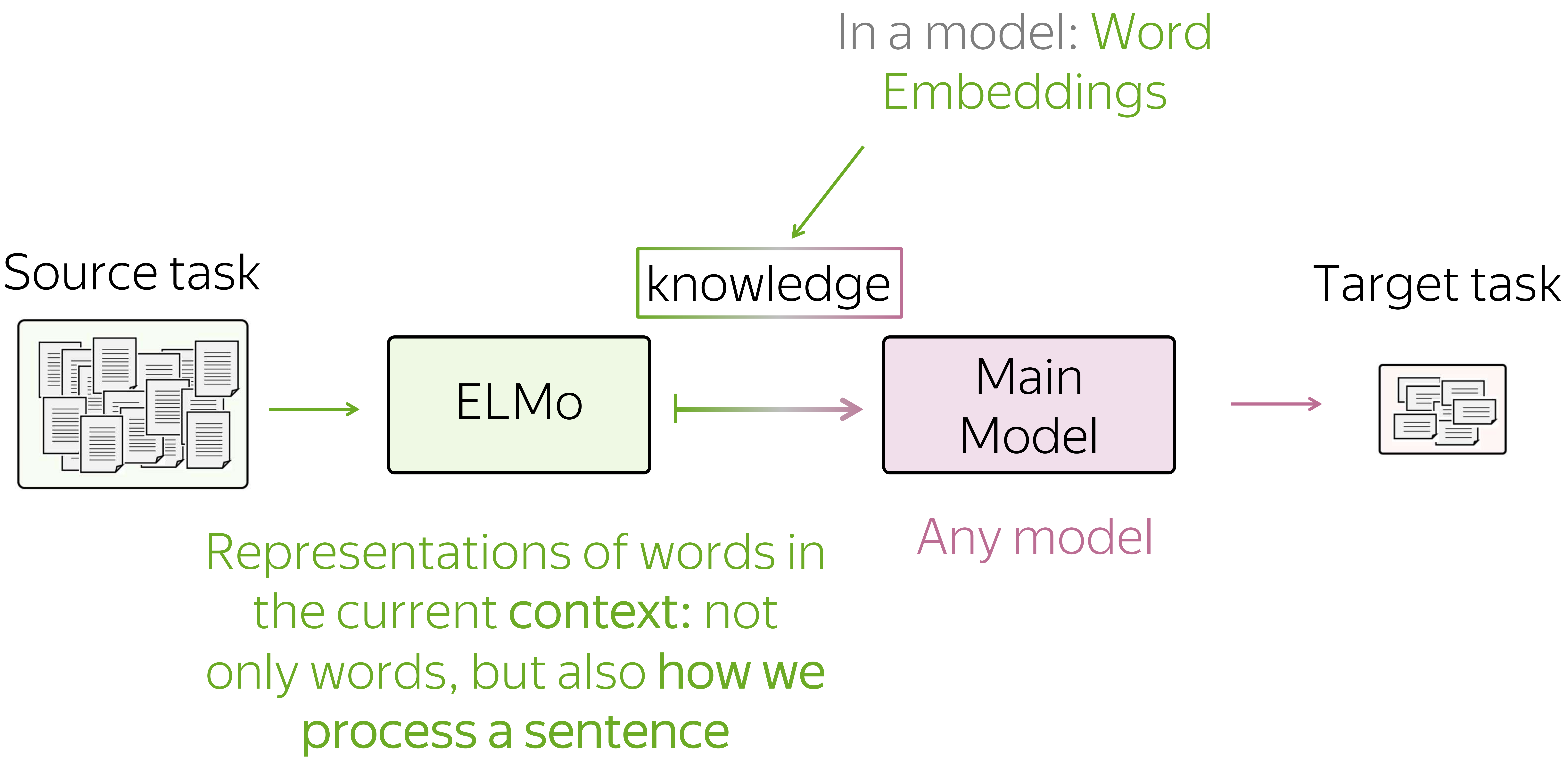
What is going to happen:

- Transfer Learning Idea
- Pretrained Models
    - (recap) Word Embeddings
    - ELMo
    - BERT
    - (a note on) GPT
    - (a note on) Adaptors
-  Analysis and Interpretability

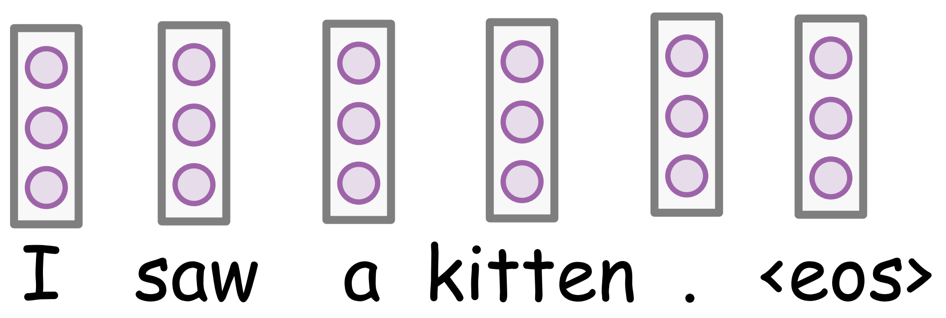
What is going to happen:

- Transfer Learning Idea
- Pretrained Models
  - (recap) Word Embeddings
  - ELMo
  - BERT
  - (a note on) GPT
  - (a note on) Adaptors
-  Analysis and Interpretability

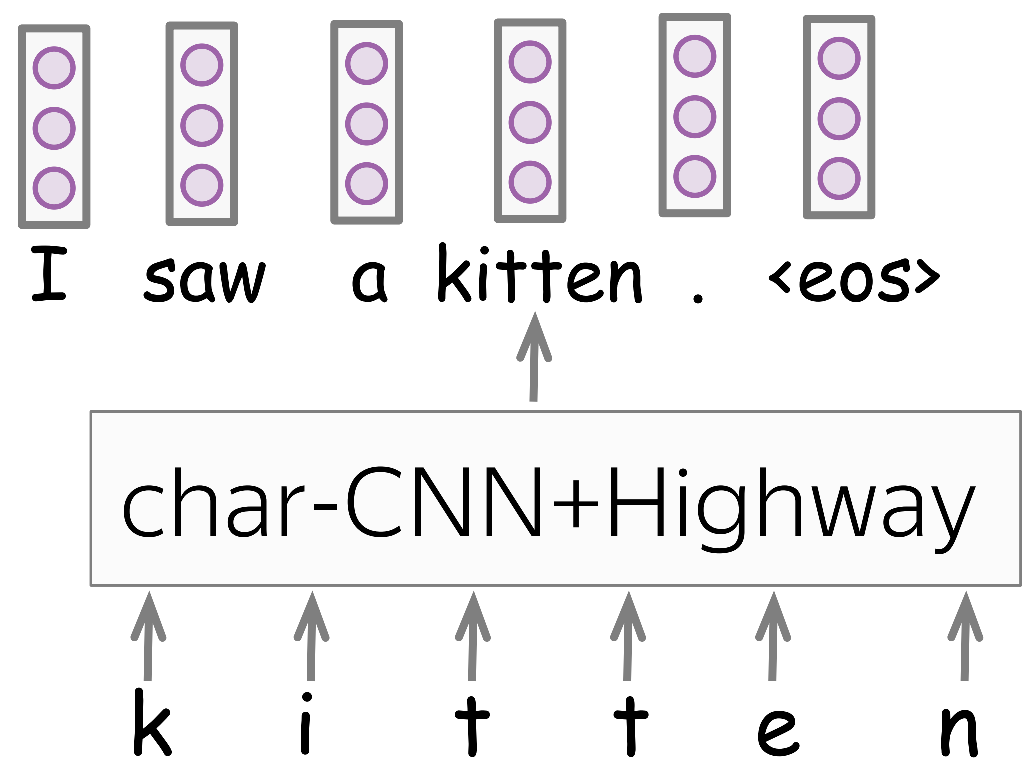
ELMo: From Words to Words-in-Context



ELMo: From Words to Words-in-Context



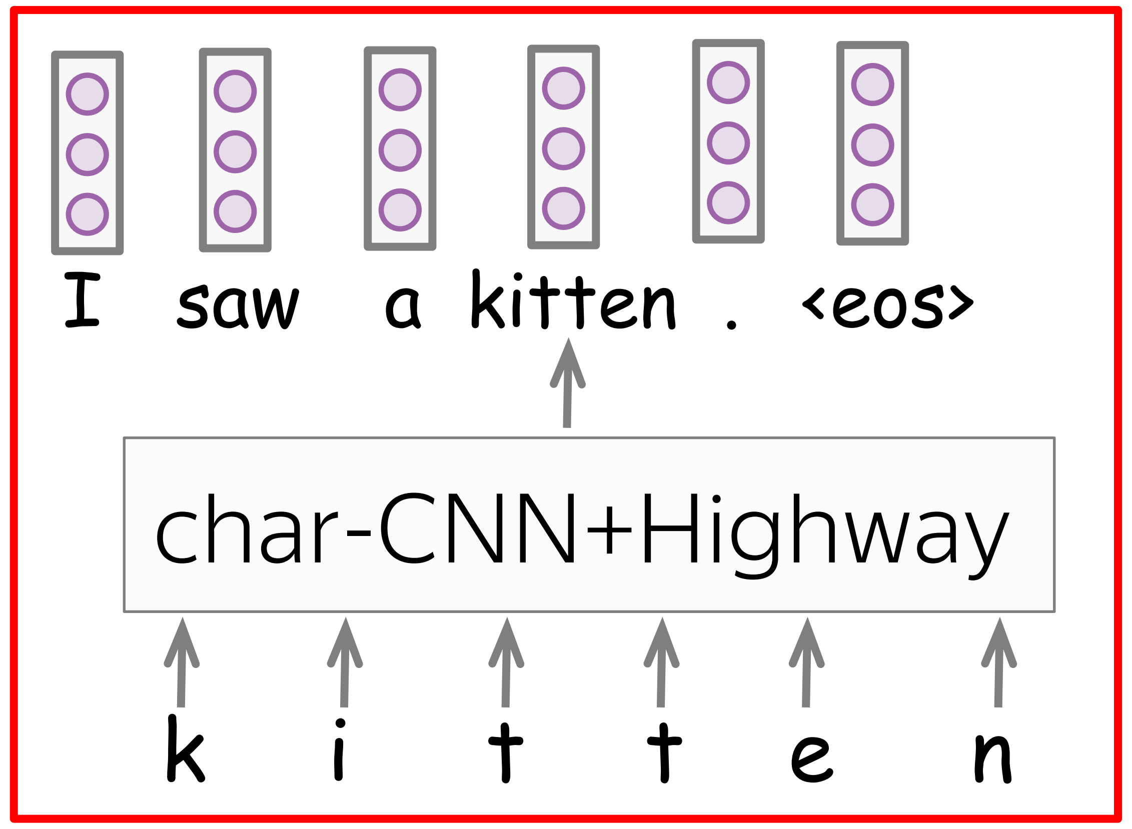
ELMo: From Words to Words-in-Context

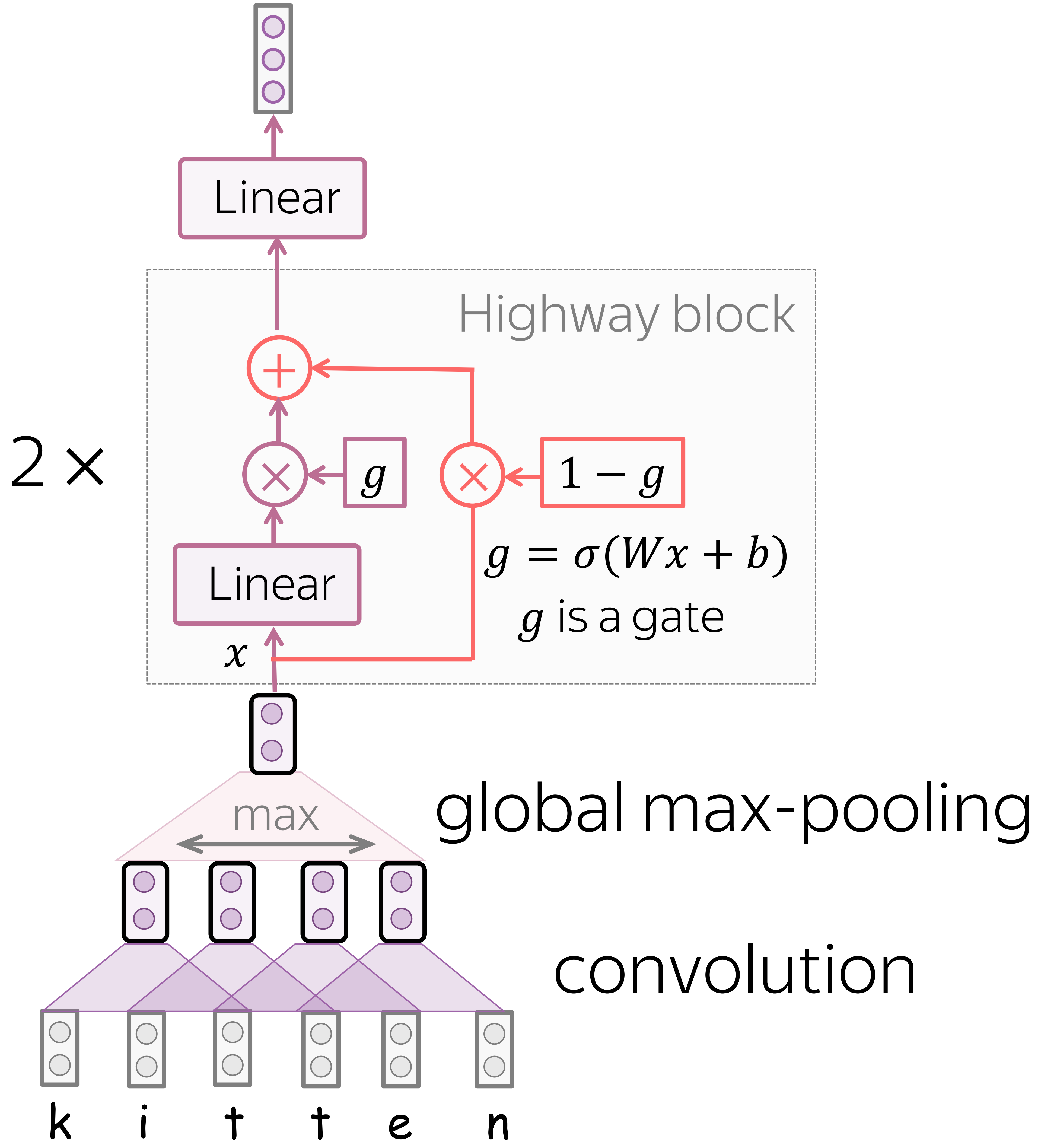


Character-level CNN:

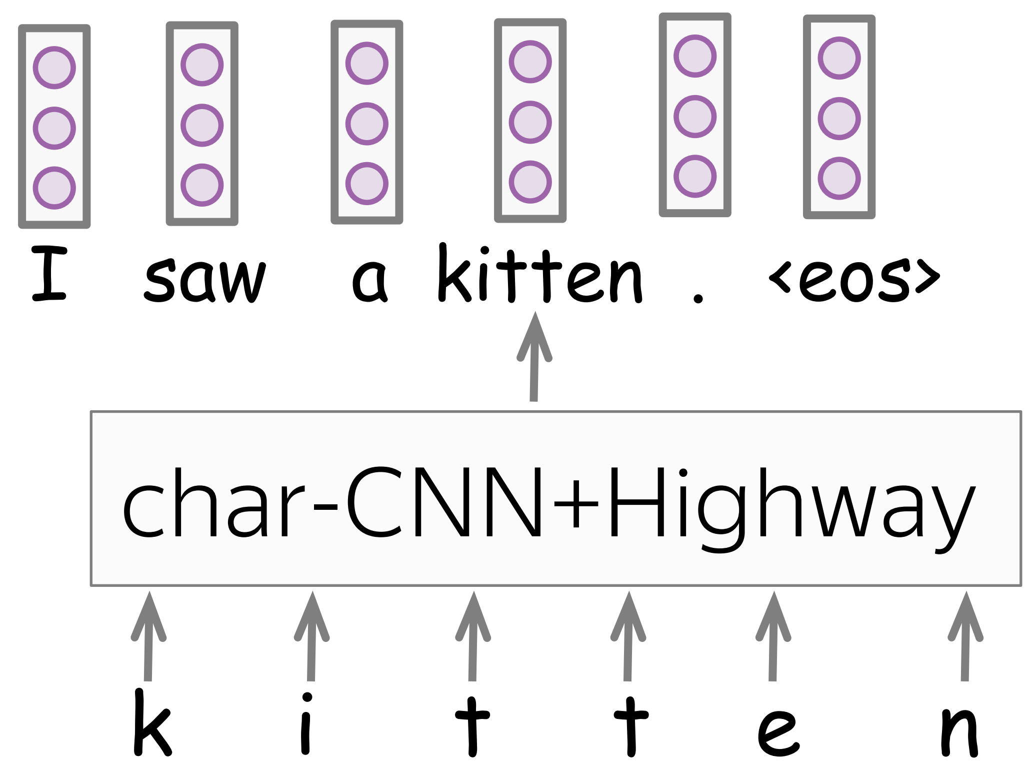
- makes it possible to represent even unknown words
- tells a network which words are written similarly

ELMo: From Words to Words-in-Context

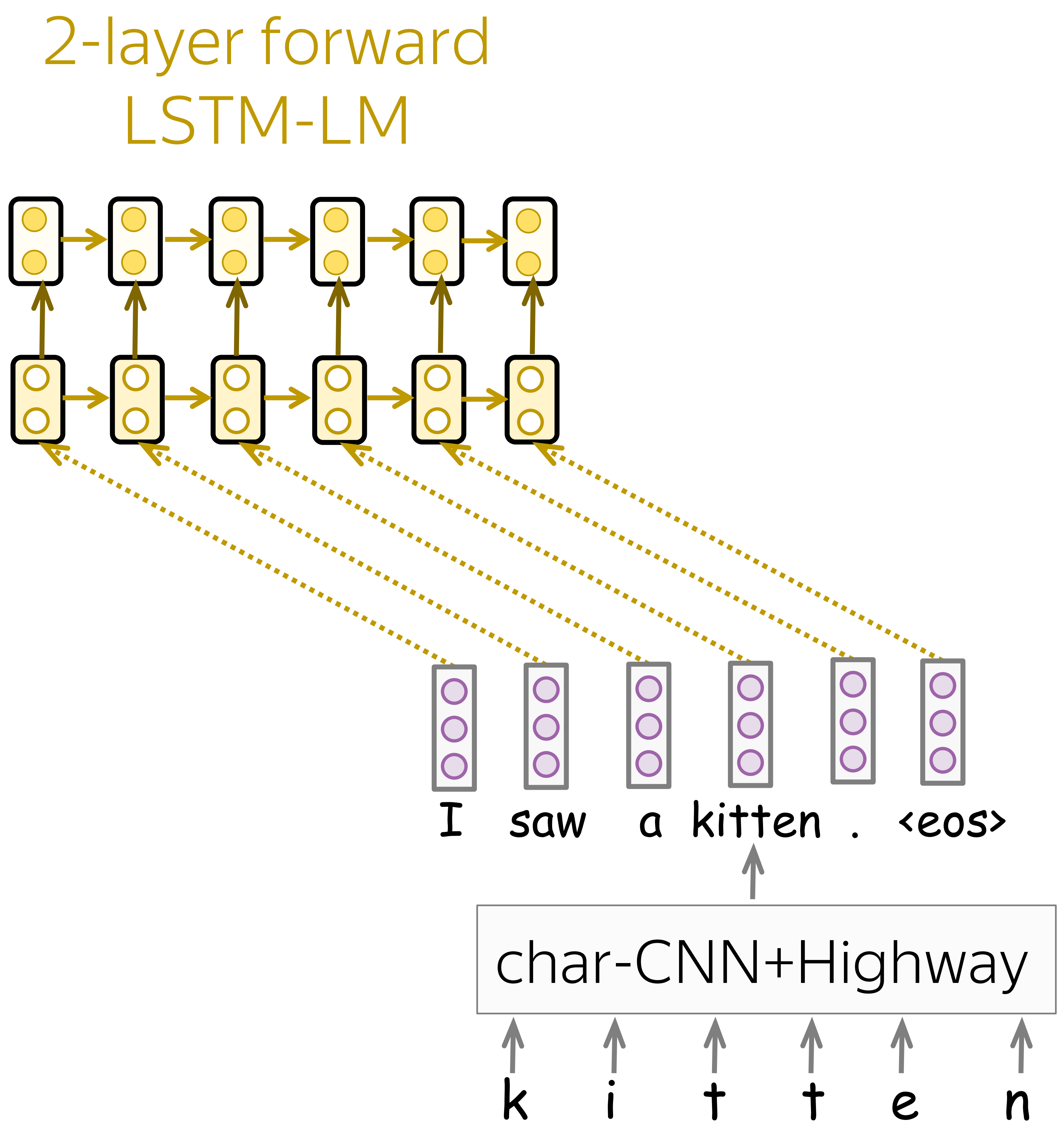




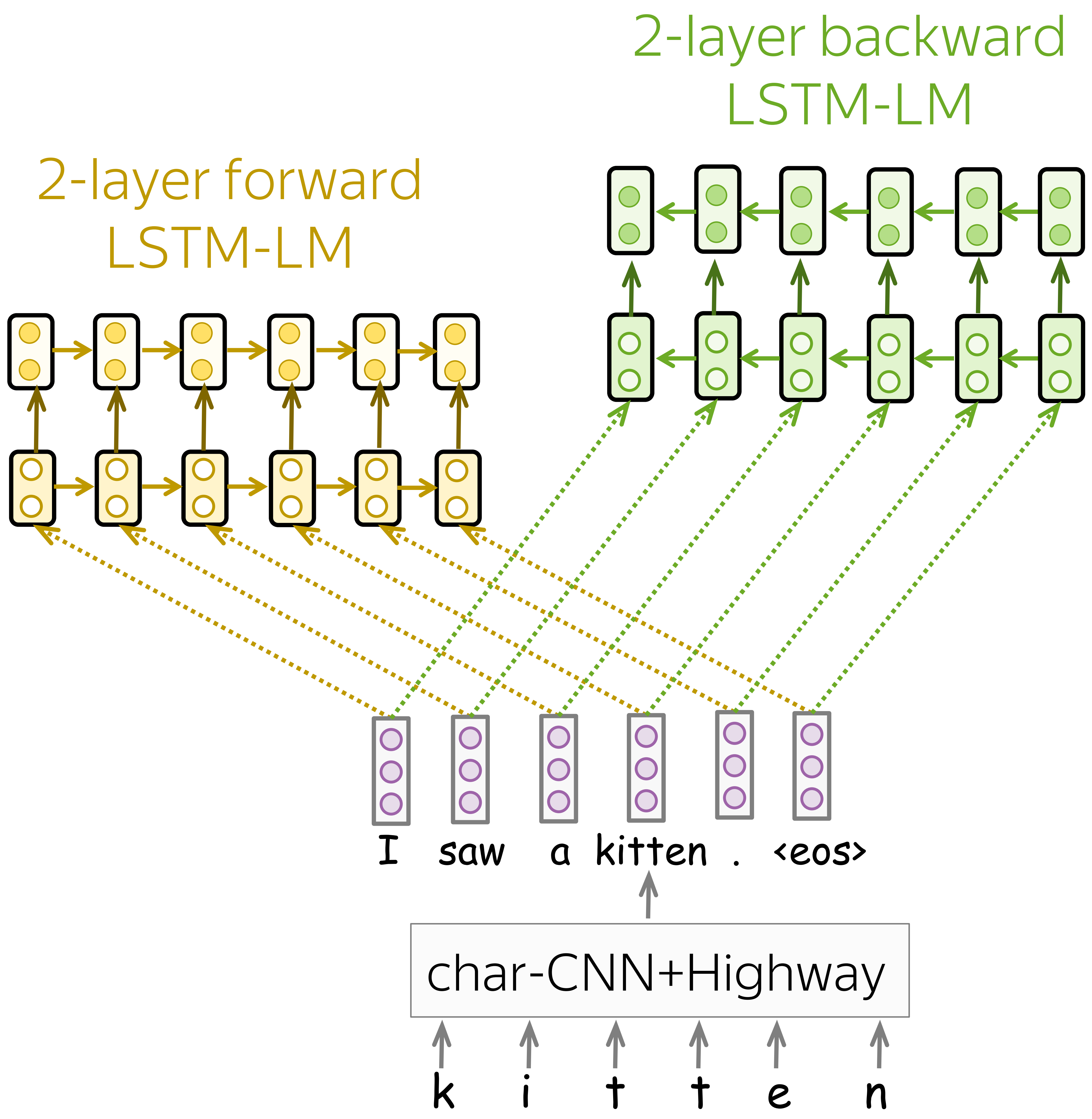
ELMo: From Words to Words-in-Context



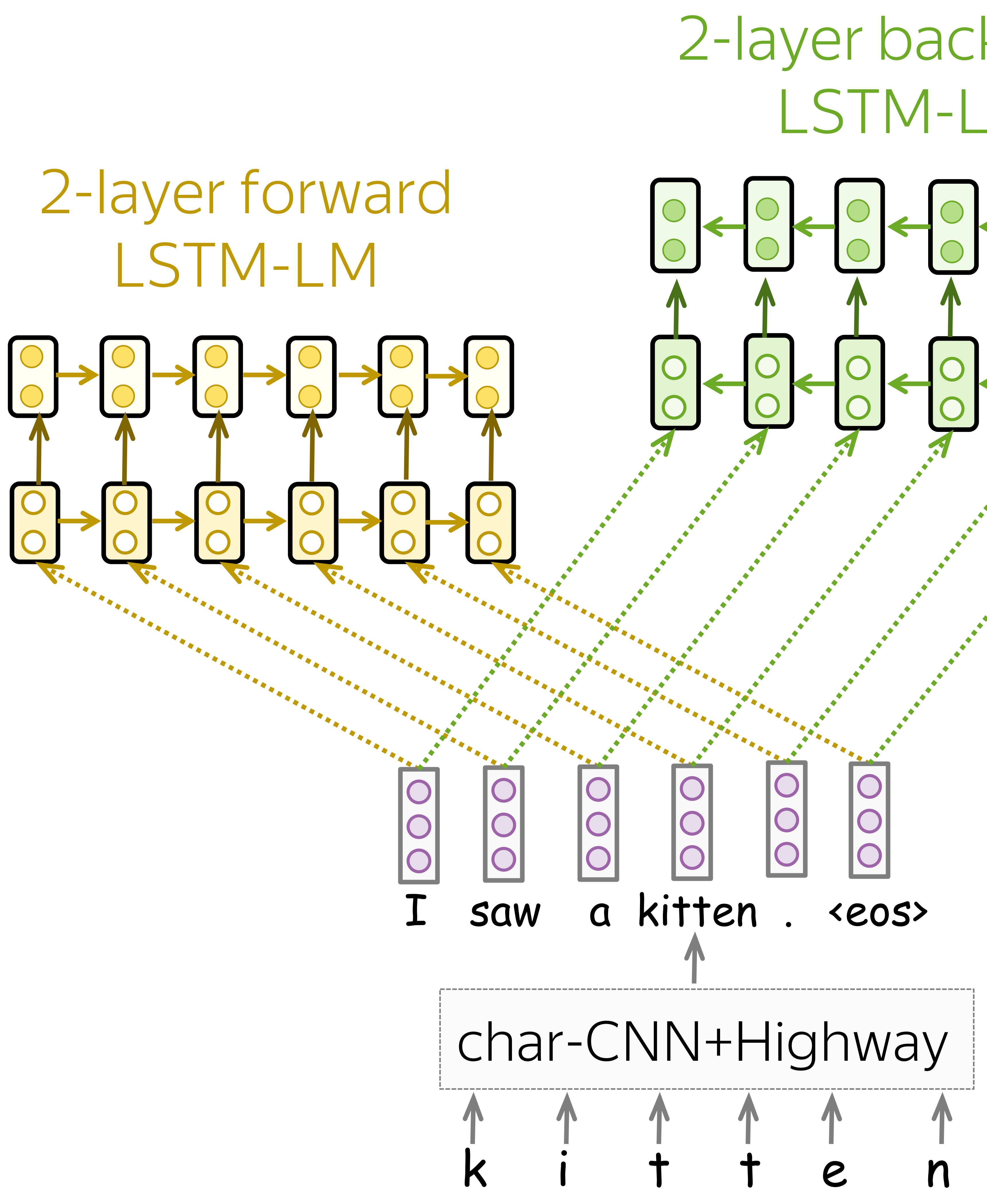
ELMo: From Words to Words-in-Context

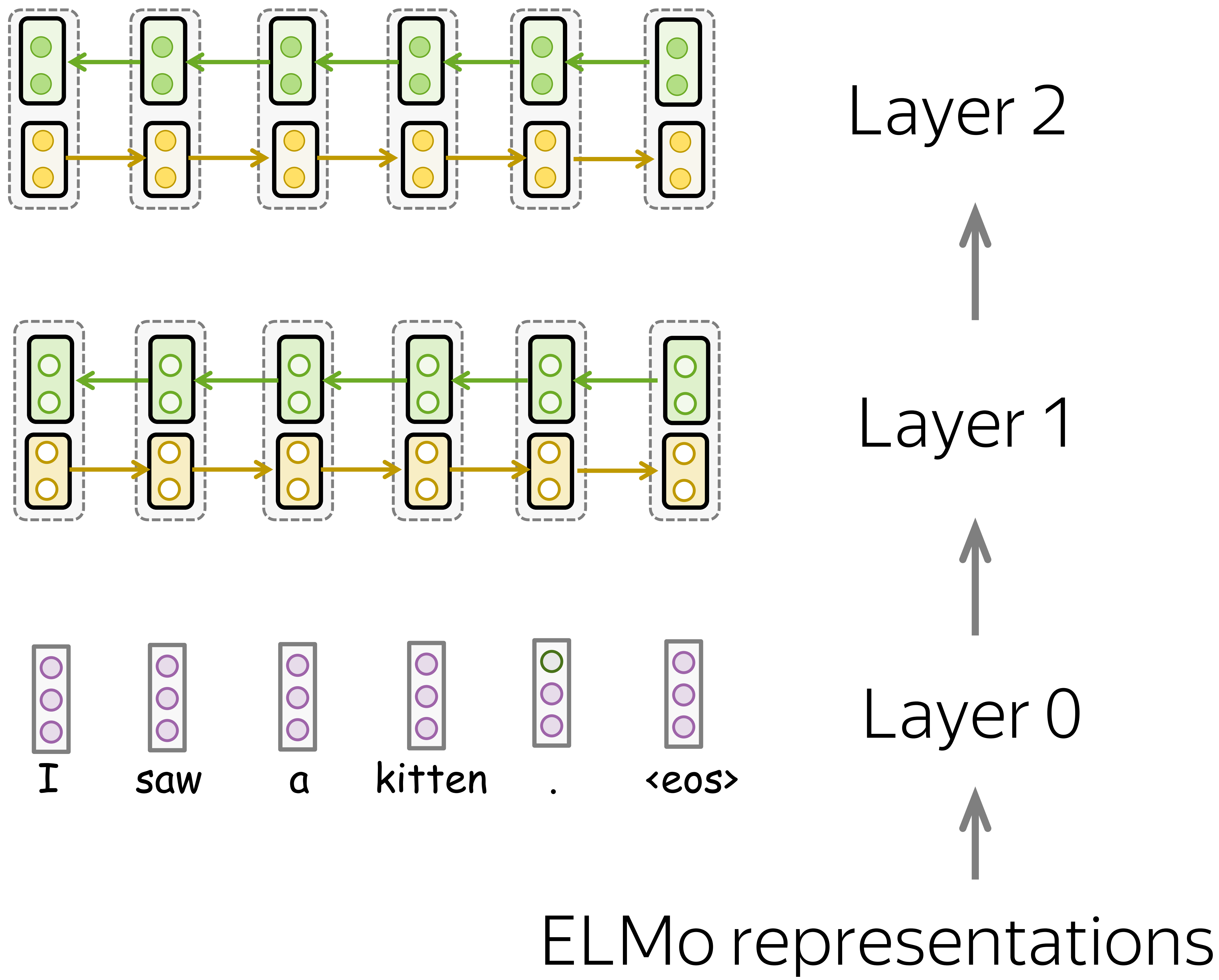


ELMo: From Words to Words-in-Context

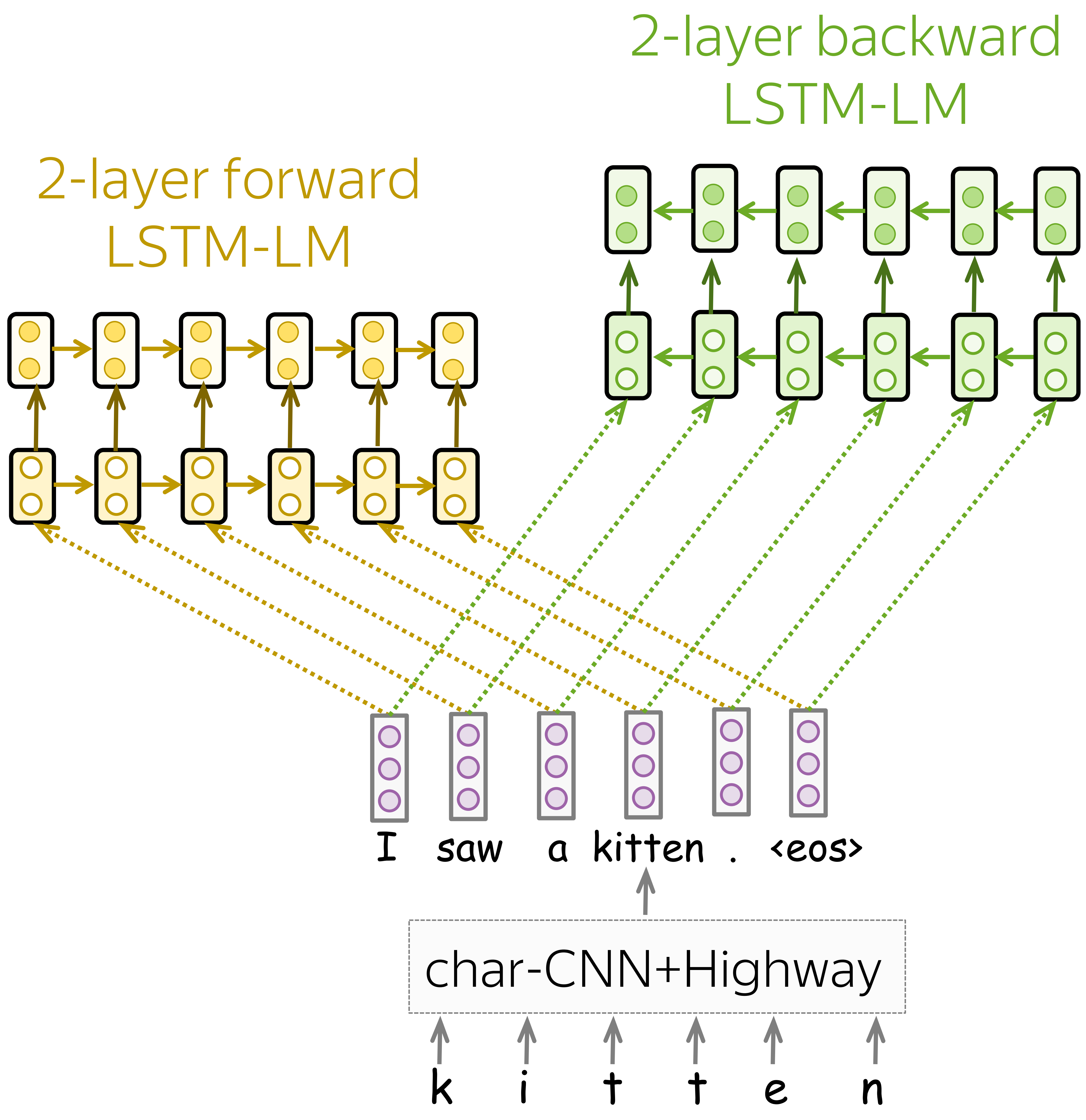


ELMo: From Words to Words-in-Context

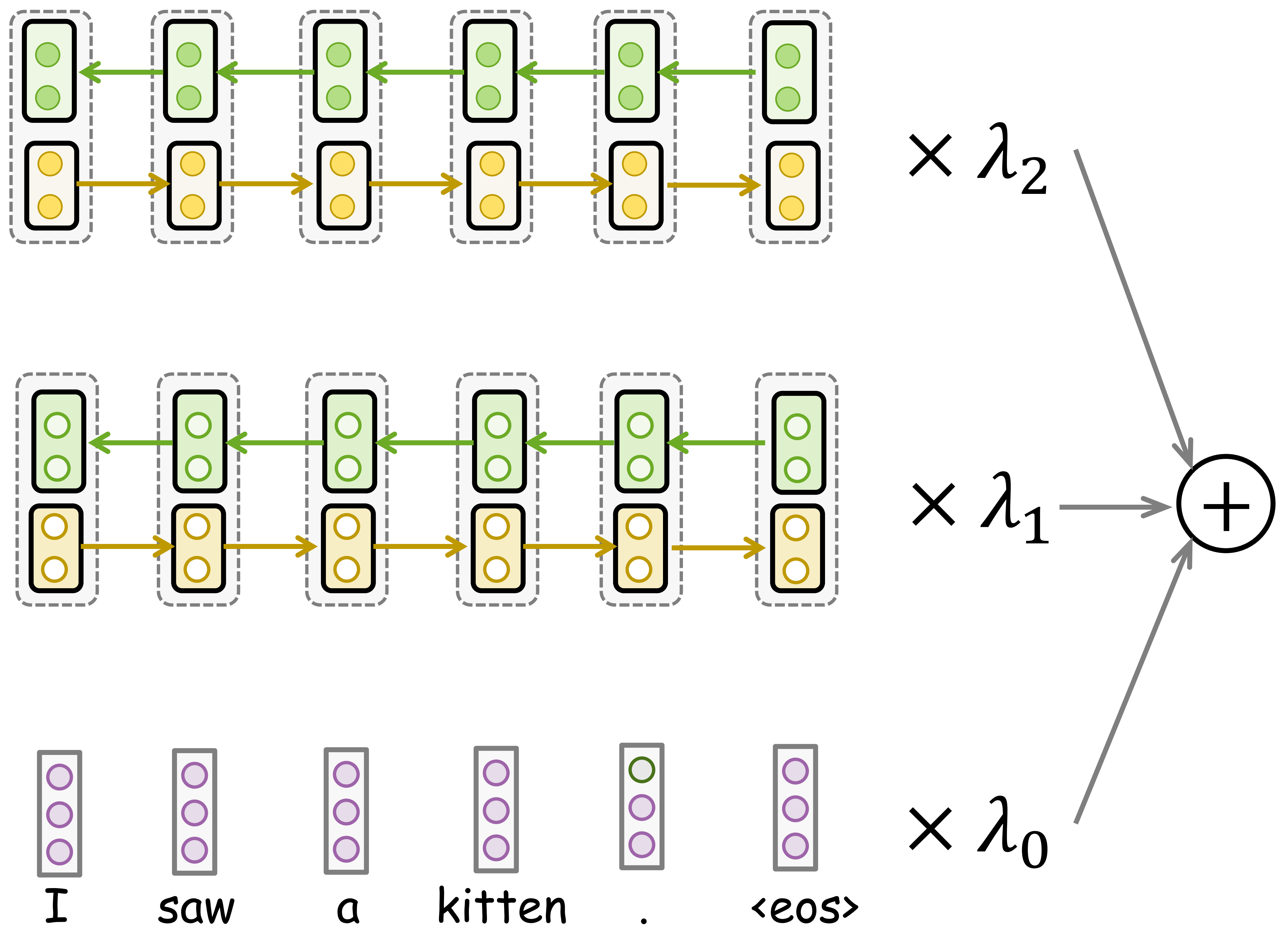




ELMo: From Words to Words-in-Context

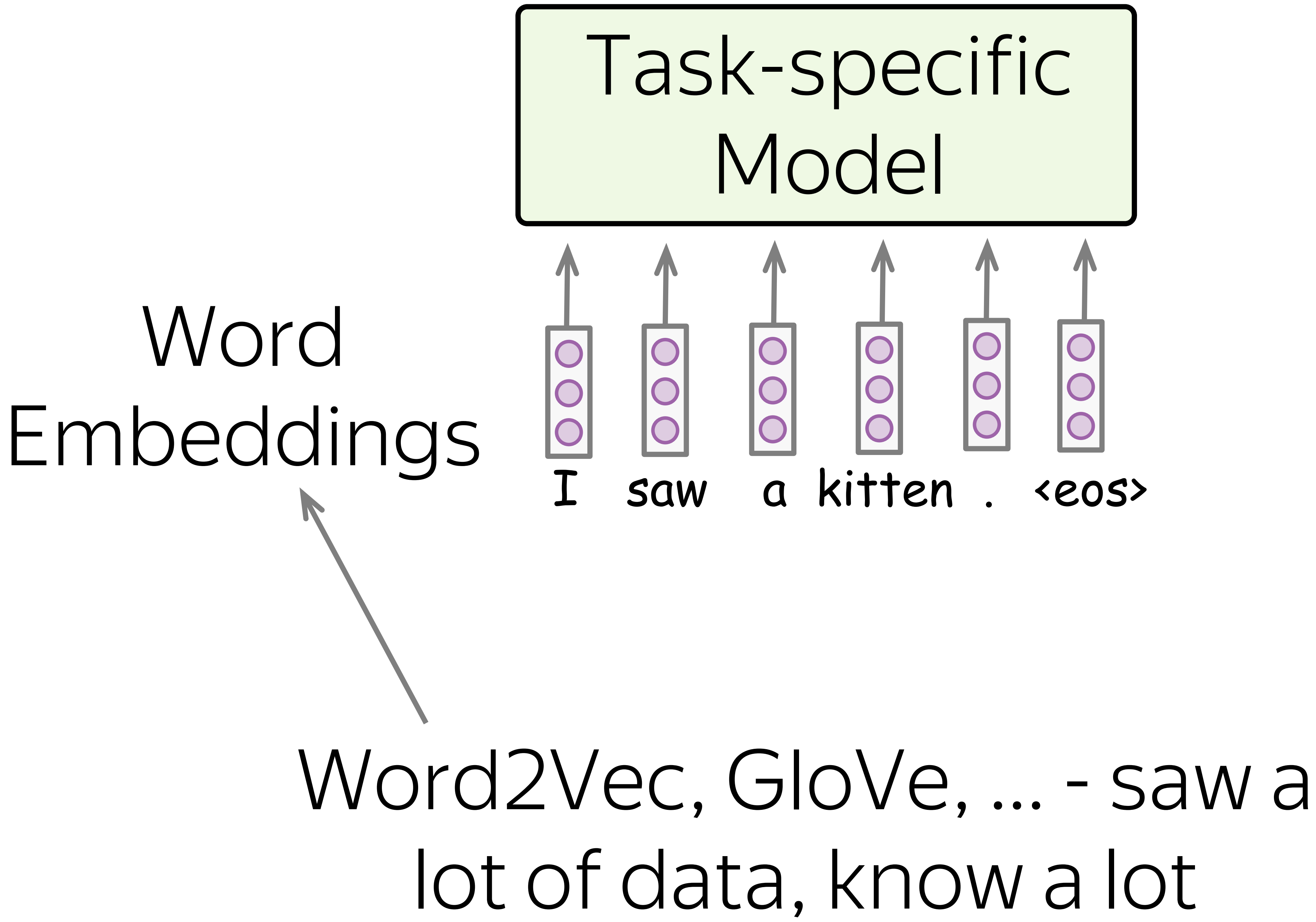


Learn specific $\lambda_0, \lambda_1, \lambda_2$ for each task



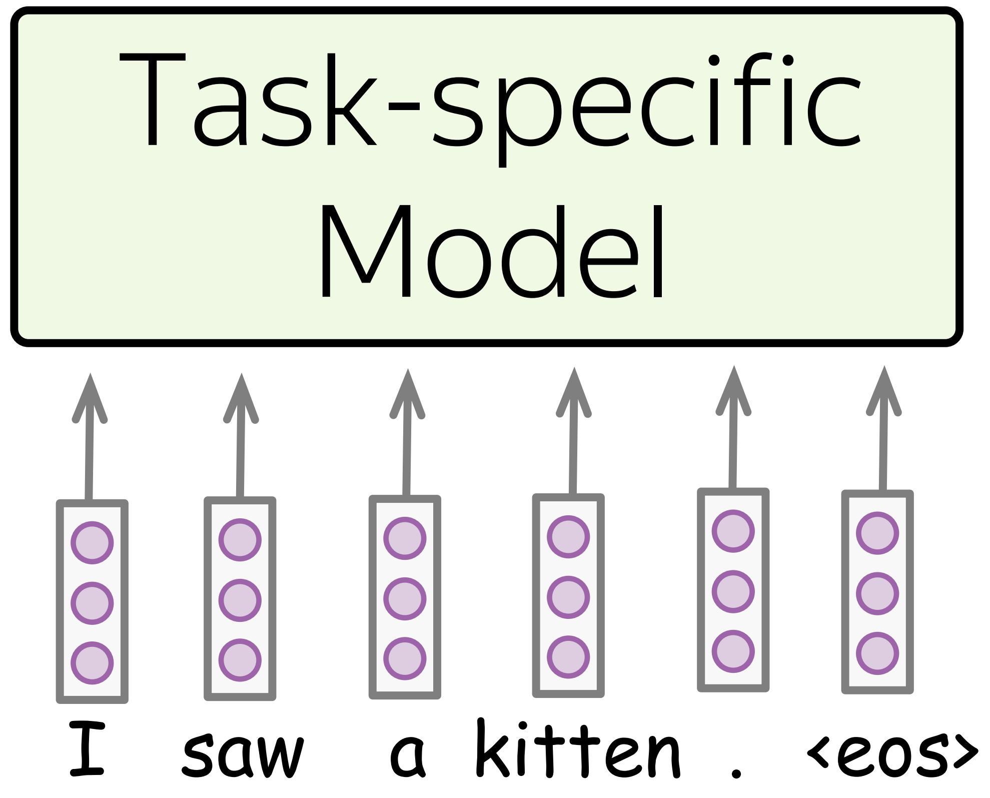
ELMo: How to Use?

Before



ELMo: How to Use?

Before

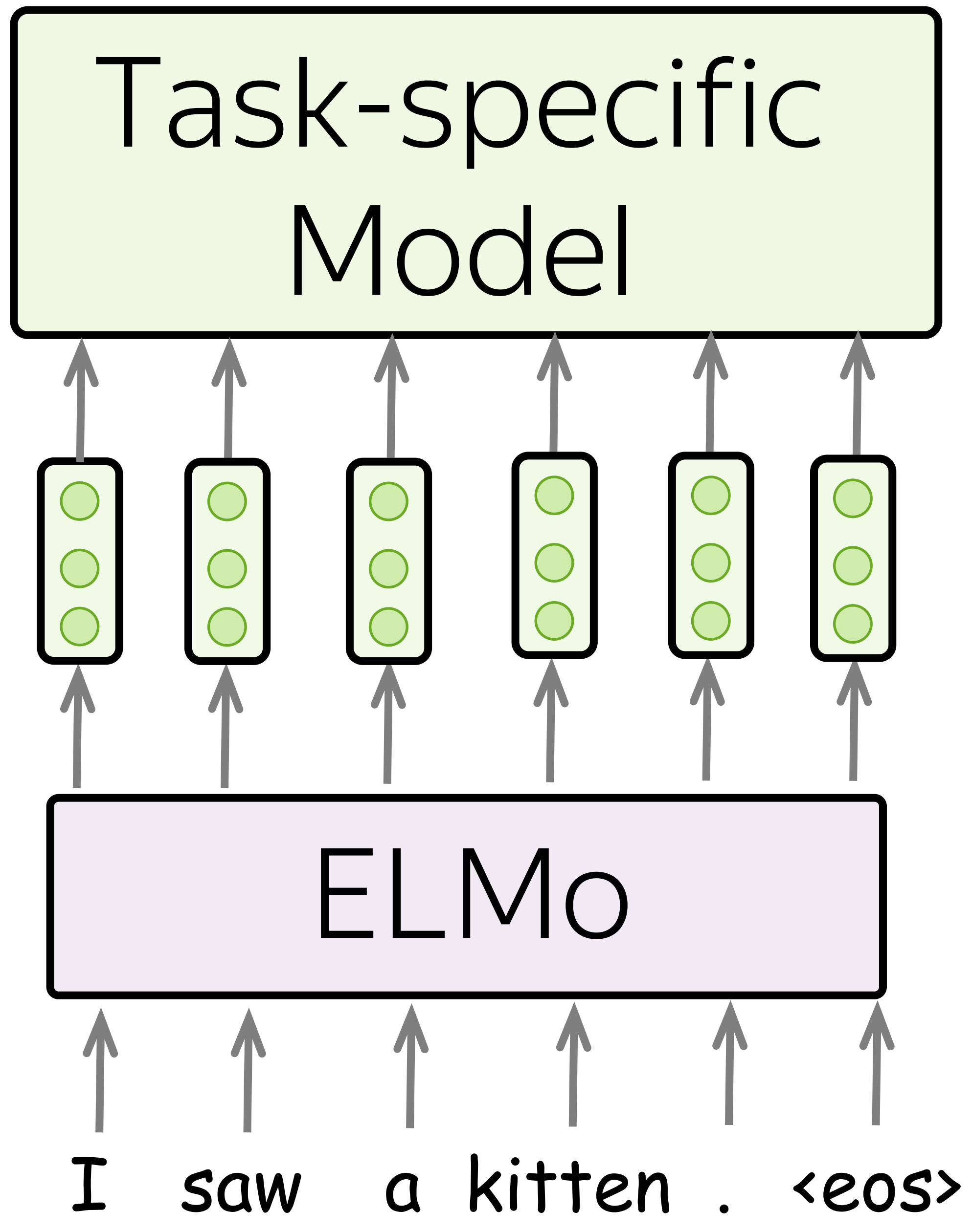


Word Embeddings

Word2Vec, GloVe, ... - saw a lot of data, know a lot

After

Representations of words **in context**



We used huge unlabeled dataset to learn not only words, but **how to process entire texts**

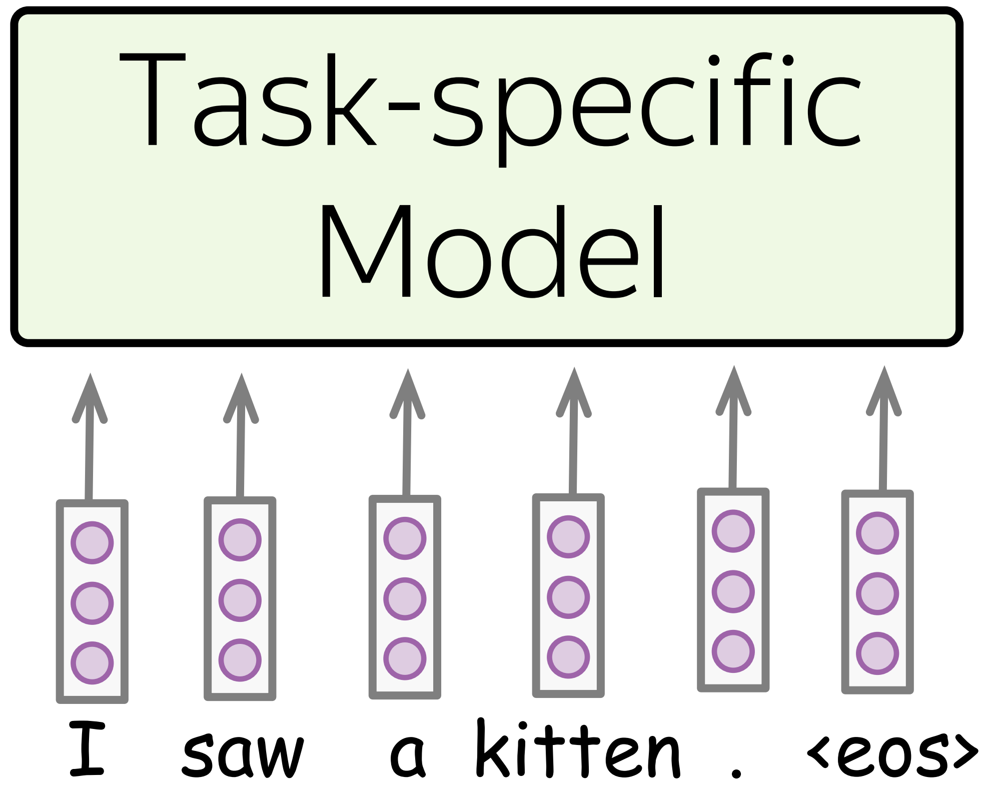
ELMo: How to Use?

Before

After

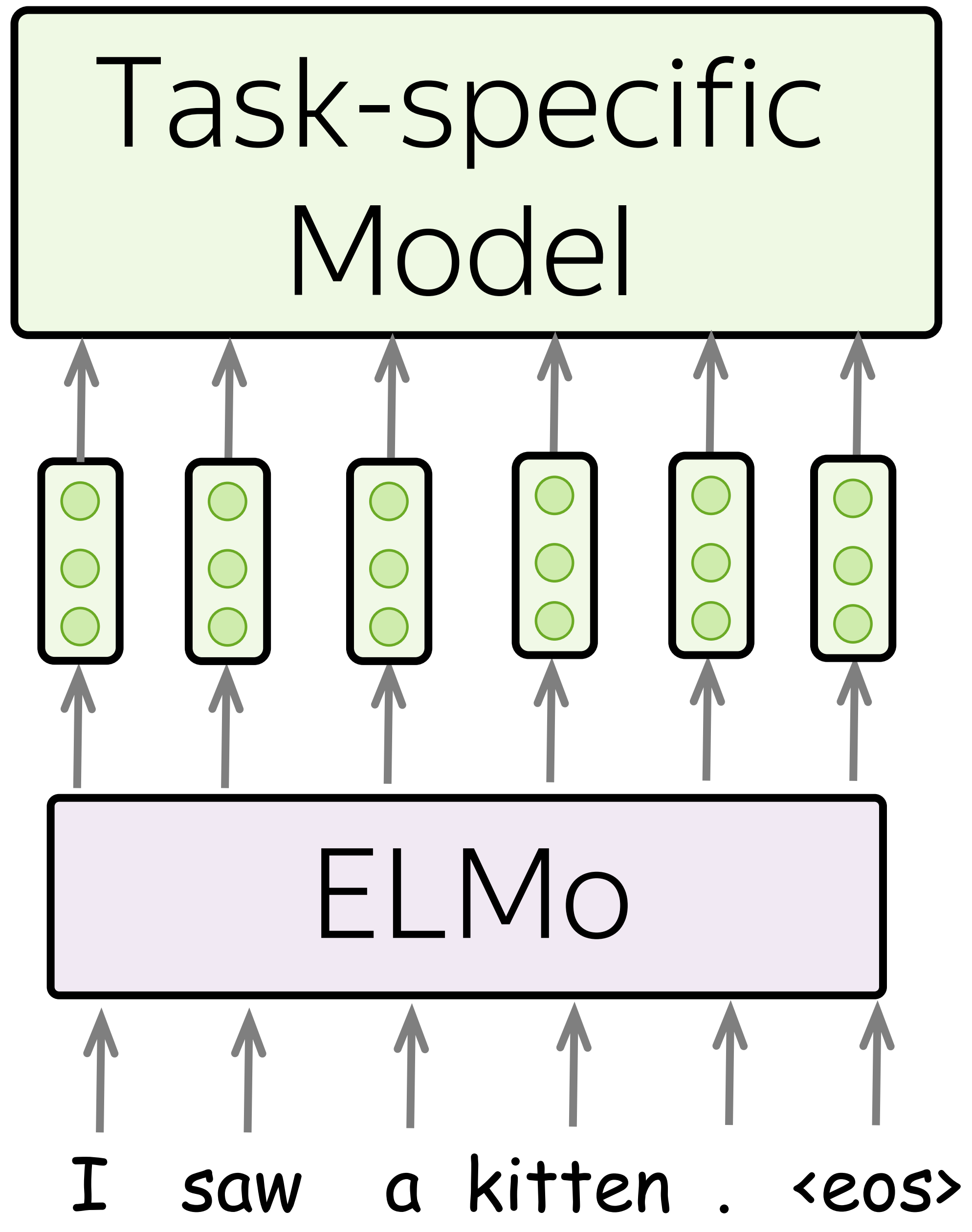
Model is (almost) the same

Word
Embeddings



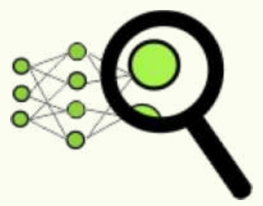
Word2Vec, GloVe, ... - saw a
lot of data, know a lot

Representations of
words **in context**



We used huge unlabeled
dataset to learn not only words,
but **how to process entire texts**

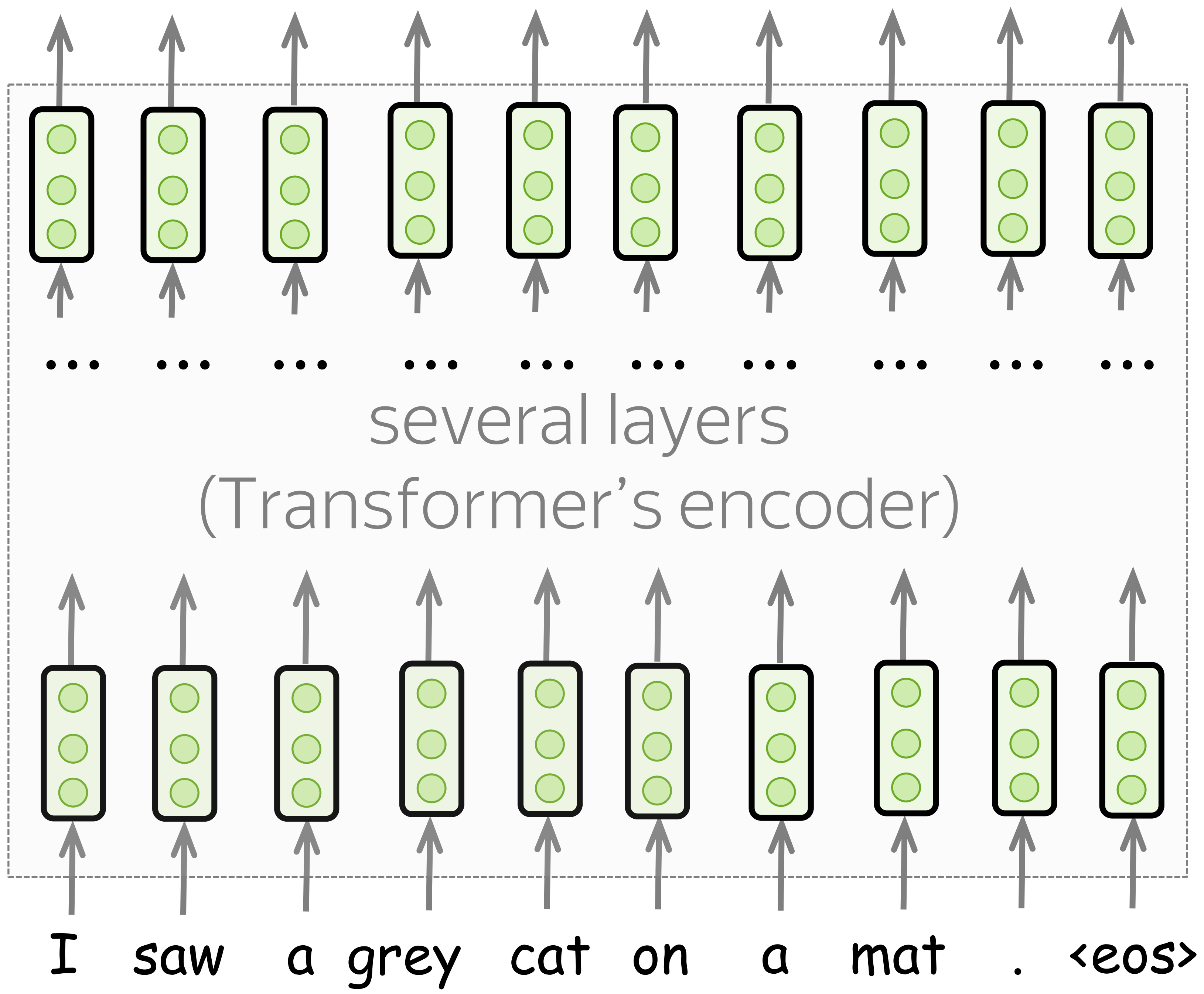
What is going to happen:

- Transfer Learning Idea
- Pretrained Models
  - (recap) Word Embeddings
  - ELMo
  - BERT
  - (a note on) GPT
  - (a note on) Adaptors
-  Analysis and Interpretability

What is going to happen:

- Transfer Learning Idea
- Pretrained Models
    - (recap) Word Embeddings
    - ELMo
    - BERT
    - (a note on) GPT
    - (a note on) Adaptors
-  Analysis and Interpretability

BERT: Transformer Encoder with Fancy Training



Model architecture:

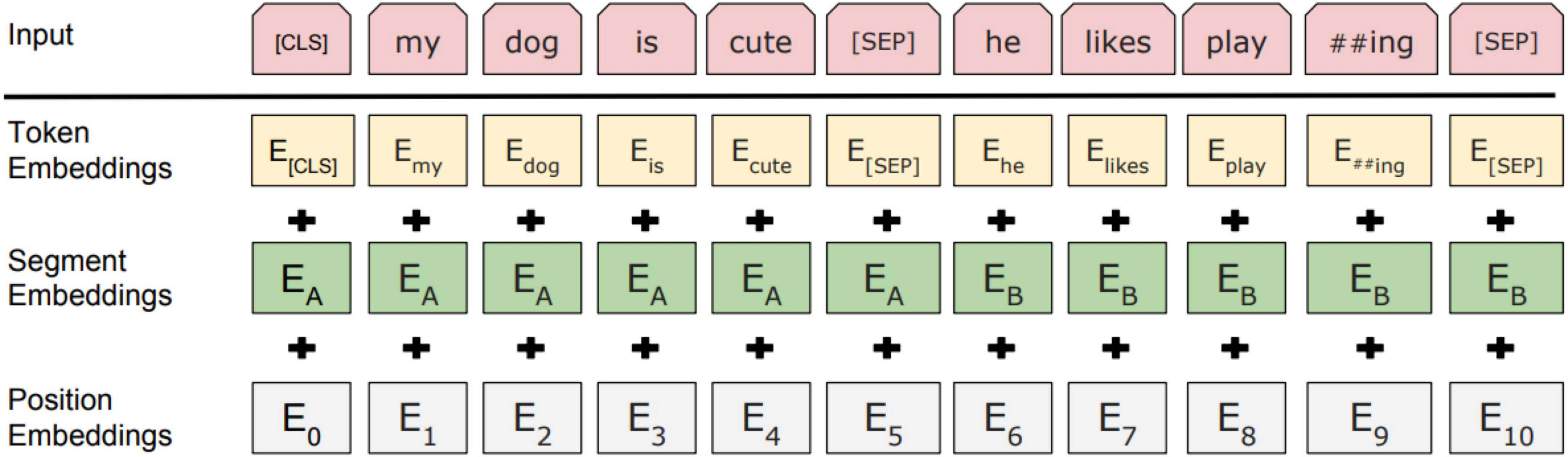
- Transformer encoder

What is special about it:

- Training objectives
  - MLM: Masked language modeling
  - NSP: Next sentence prediction

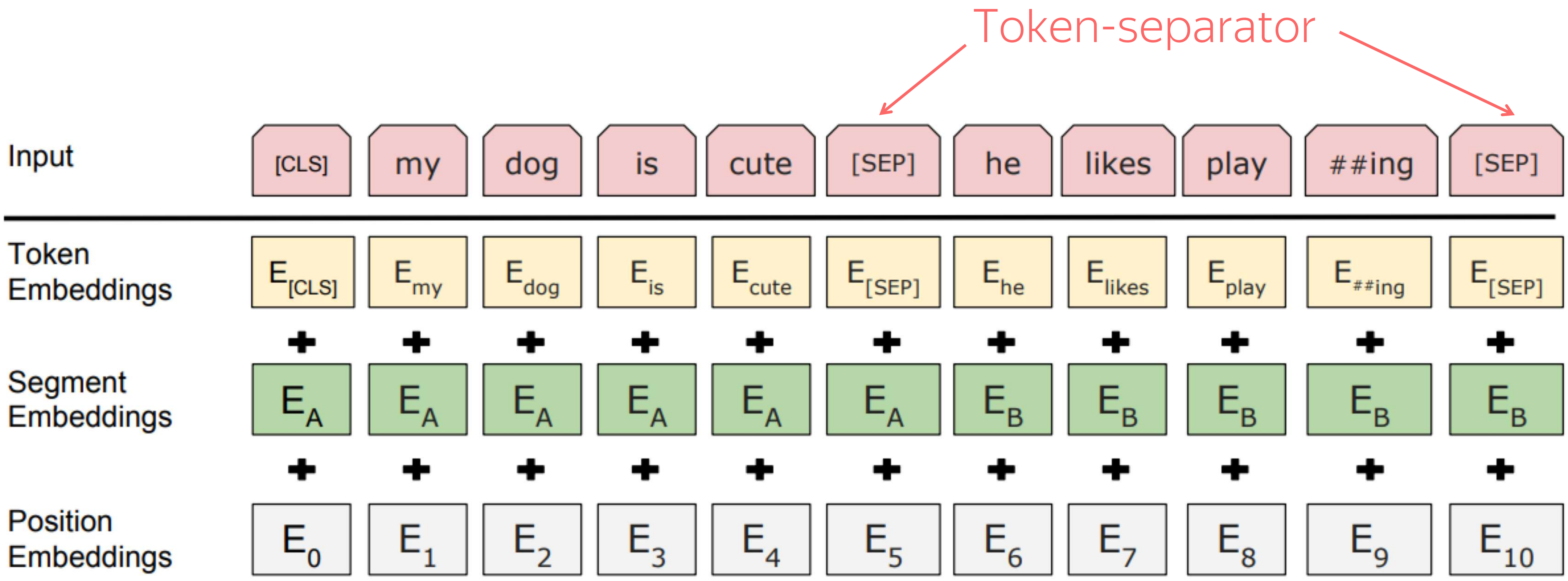
- Lots of data

BERT: Input



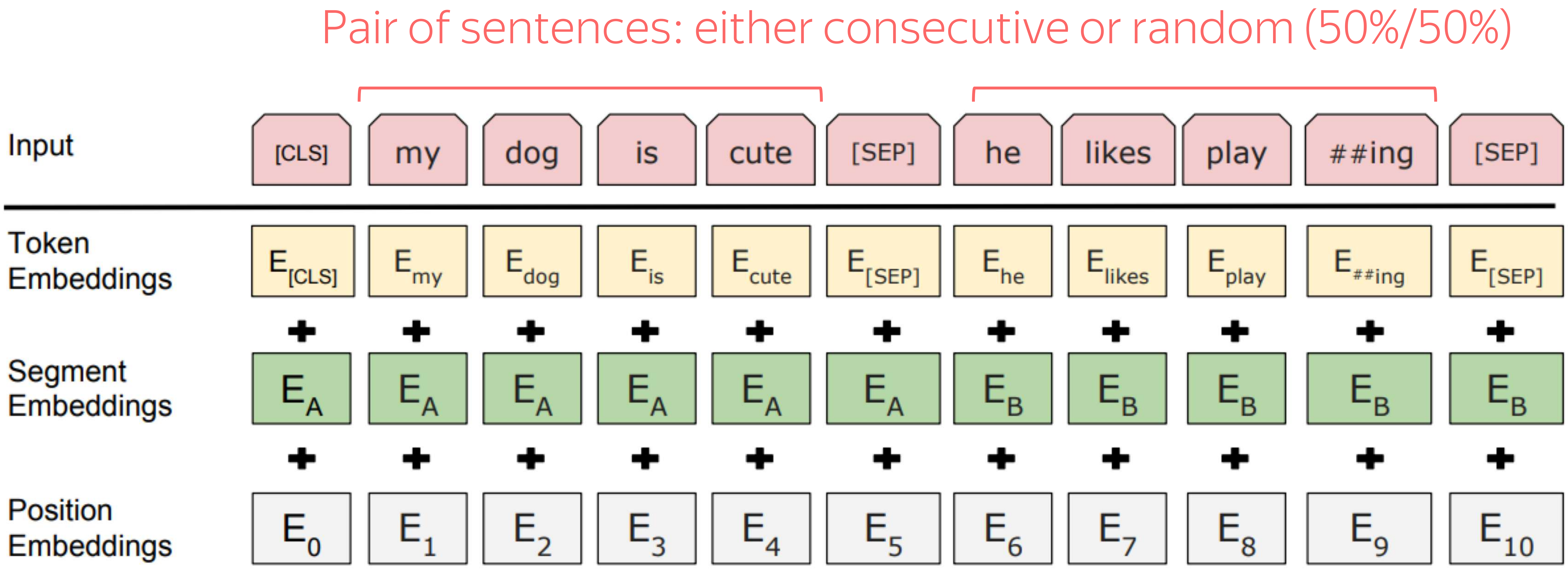
The figure is from the original BERT paper

BERT: Input



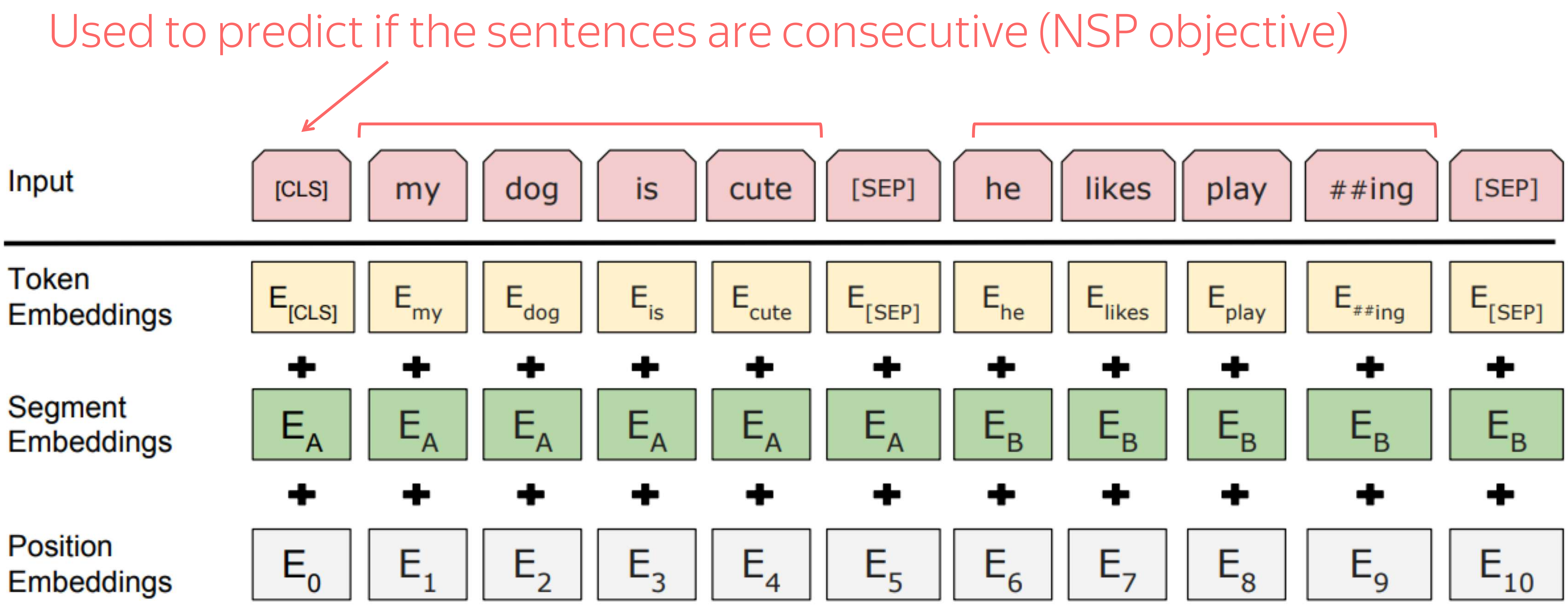
The figure is from the original BERT paper

BERT: Input



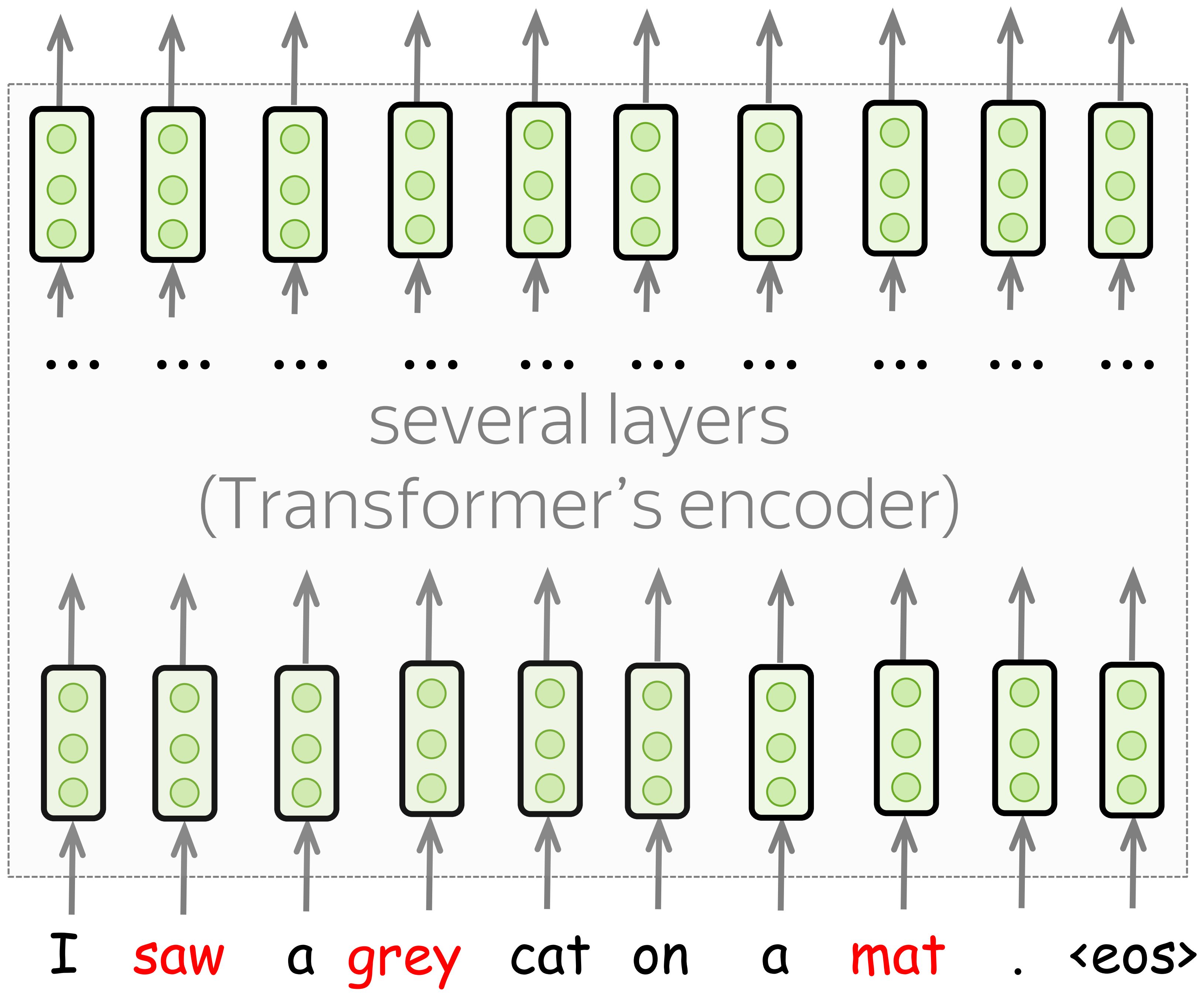
The figure is from the original BERT paper

BERT: Input



The figure is from the original BERT paper

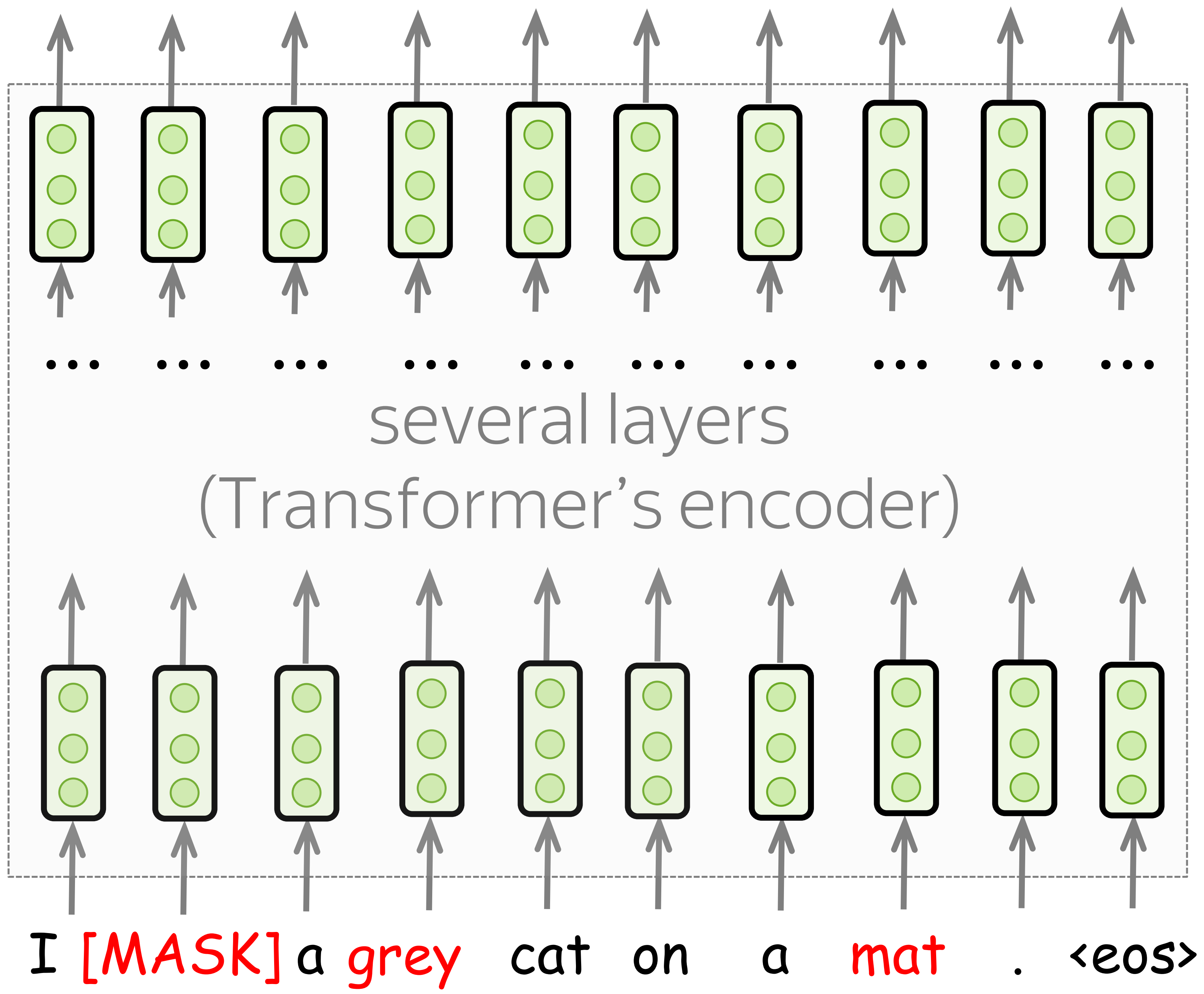
BERT: Masked Language Modeling Objective



At each training step:

- pick randomly 15% of tokens

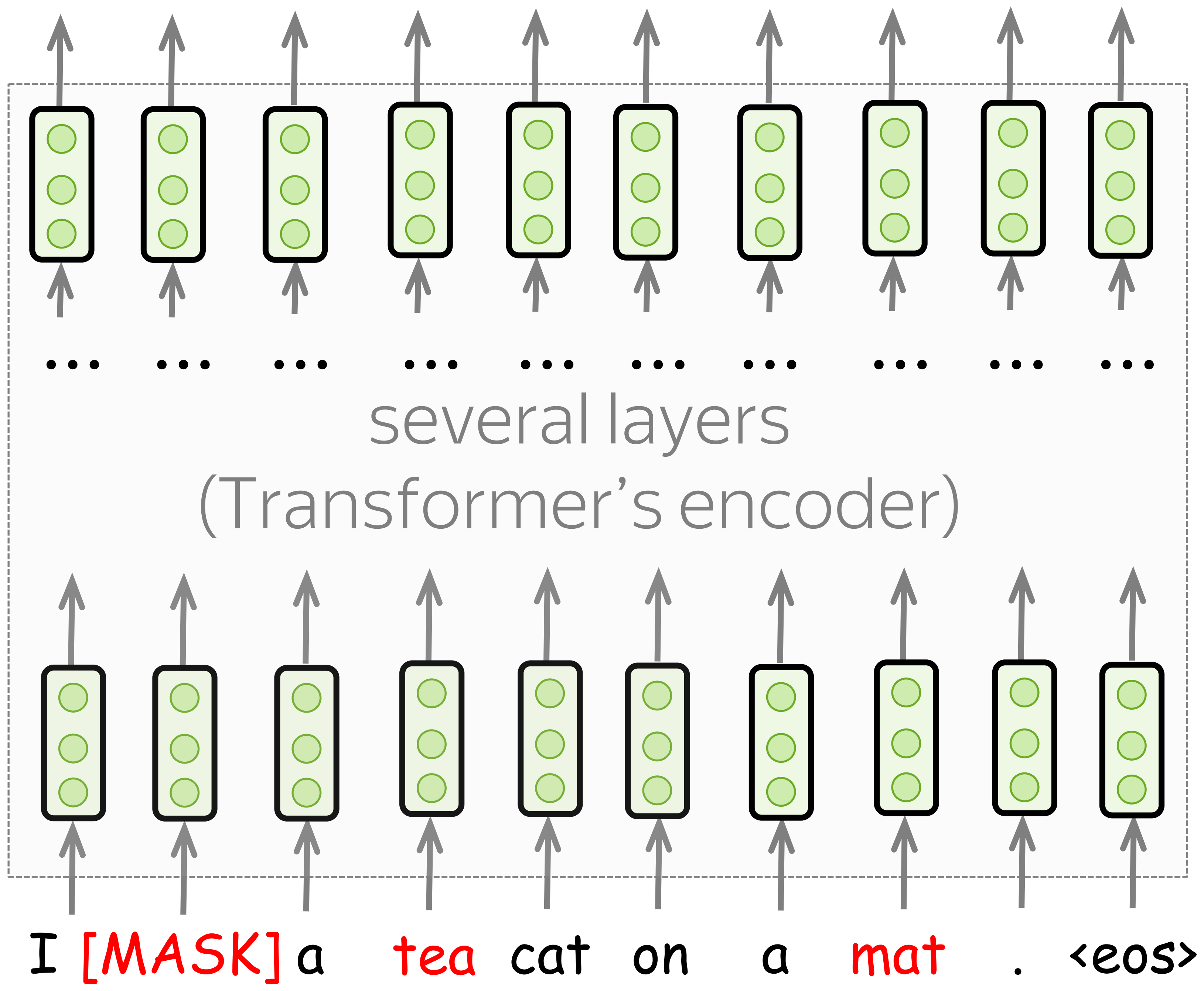
BERT: Masked Language Modeling Objective



At each training step:

- pick randomly 15% of tokens
- replace each of the chosen tokens with
  - [MASK] with prob. 80%

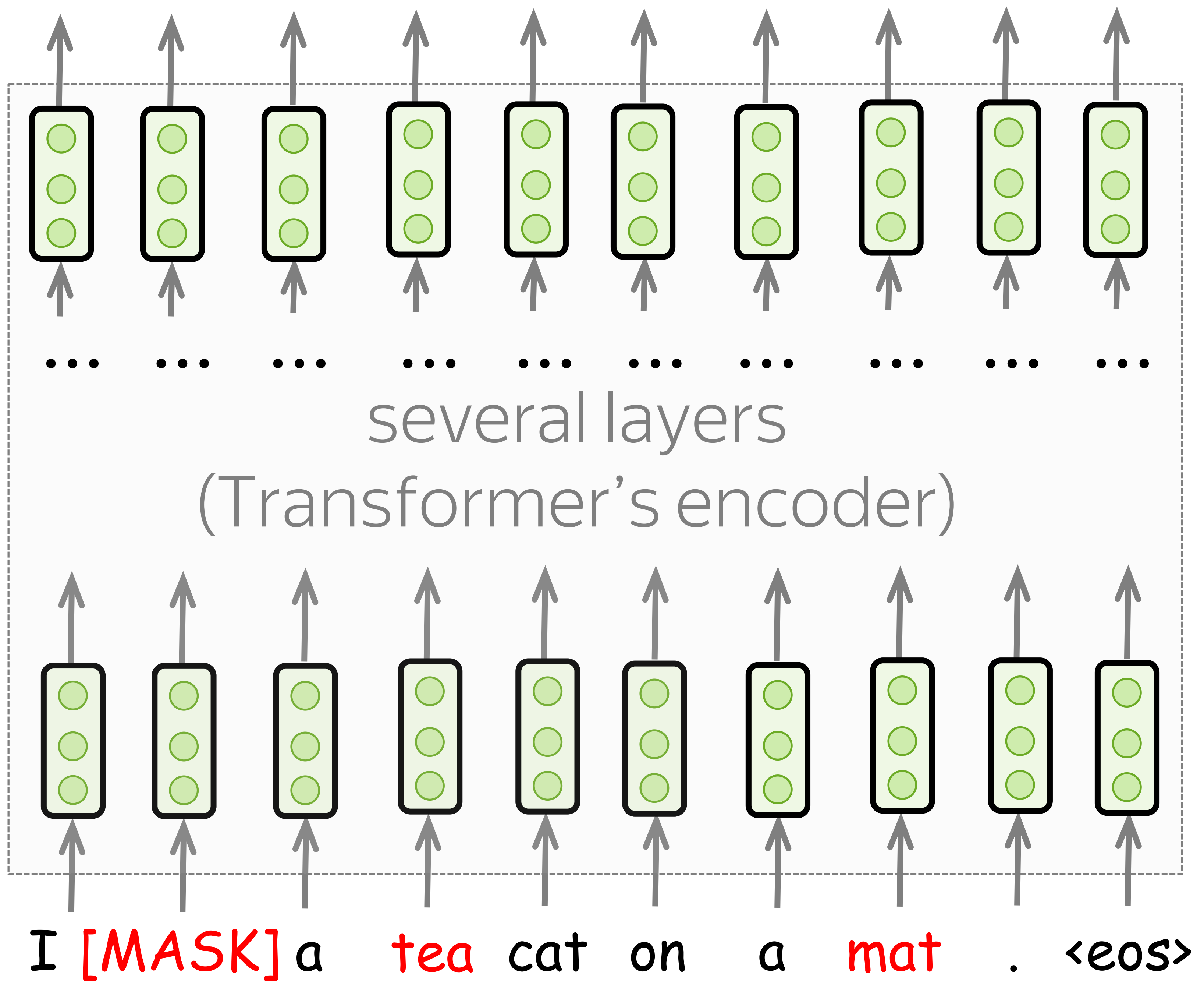
BERT: Masked Language Modeling Objective



At each training step:

- pick randomly 15% of tokens
- replace each of the chosen tokens with
  - [MASK] with prob. 80%
  - **random token** with prob. 10%

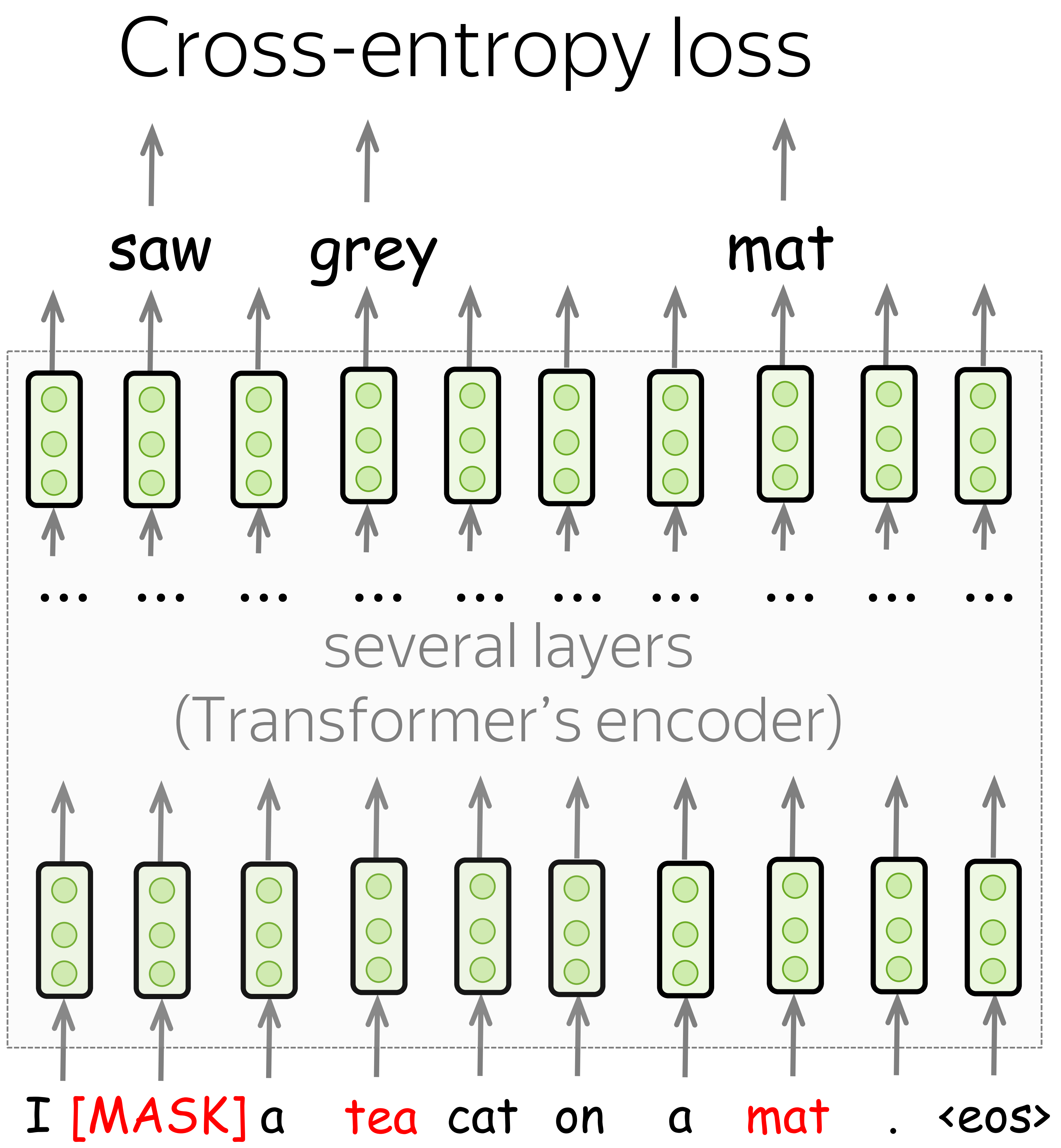
BERT: Masked Language Modeling Objective



At each training step:

- pick randomly 15% of tokens
- replace each of the chosen tokens with
  - [MASK] with prob. 80%
  - **random token** with prob. 10%
  - **self** with prob. 10%

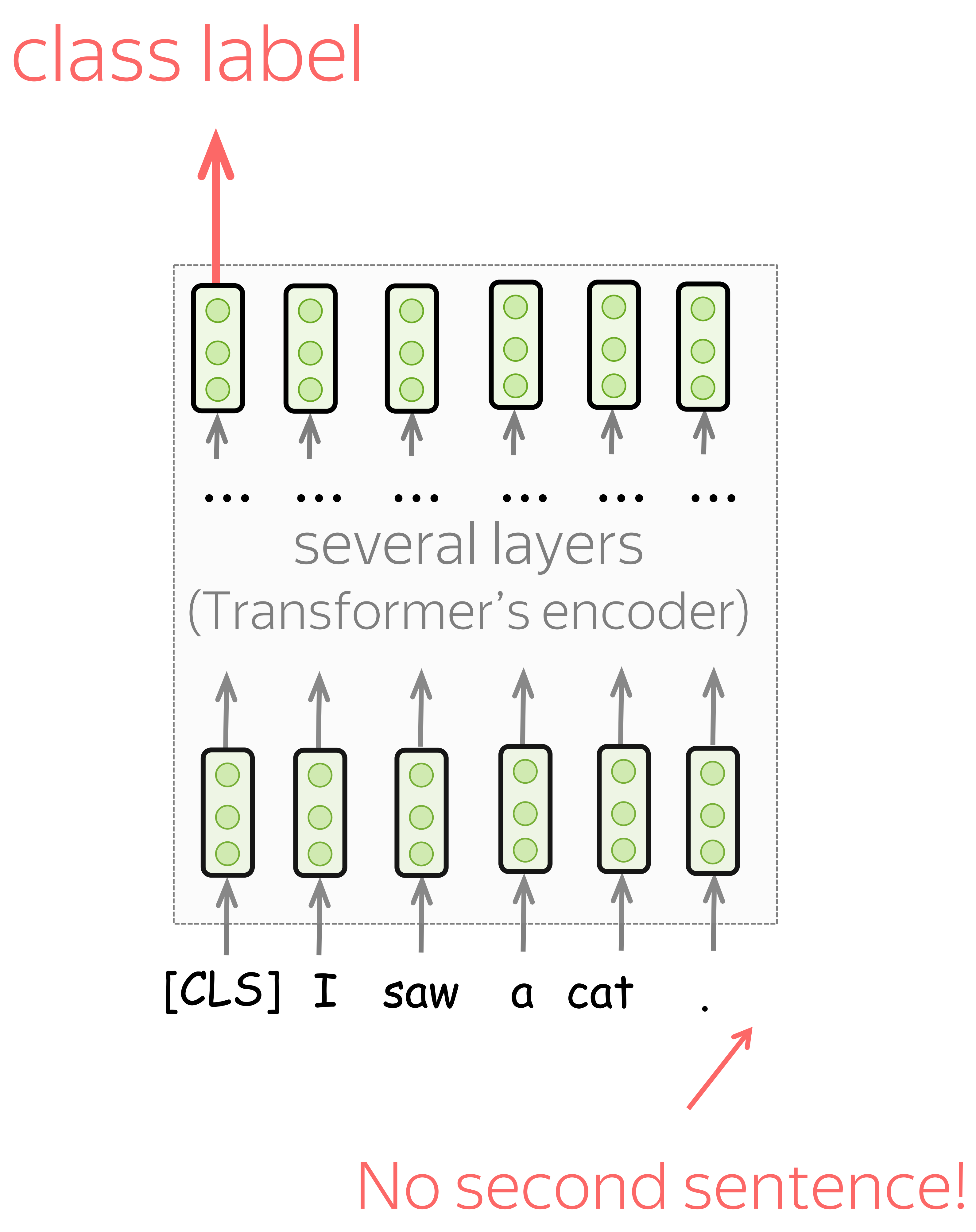
BERT: Masked Language Modeling Objective



At each training step:

- pick randomly 15% of tokens
- replace each of the chosen tokens with
  - [MASK] with prob. 80%
  - **random token** with prob. 10%
  - **self** with prob. 10%
- predict original tokens
  (only chosen ones!)

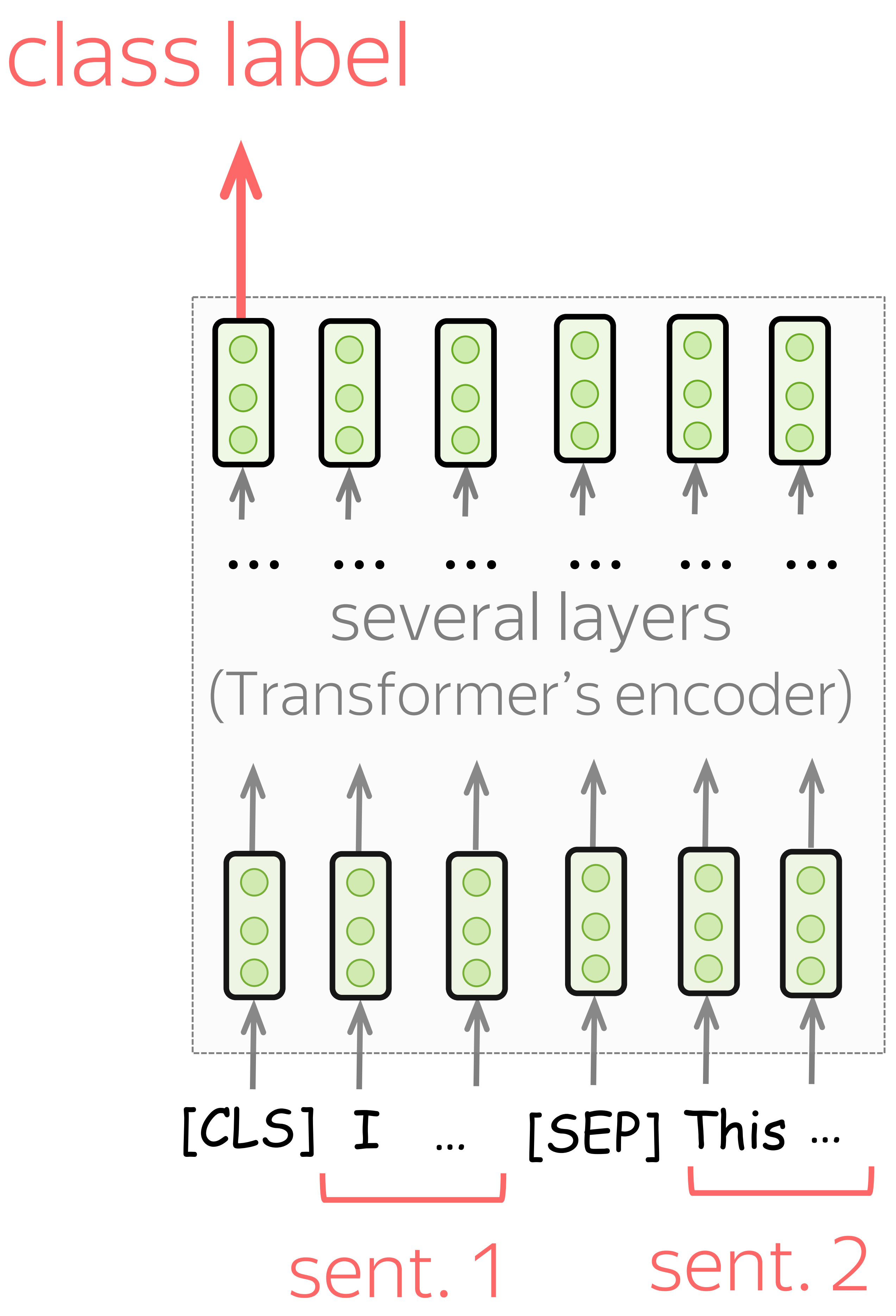
Finetuning BERT: Single-Sentence Classification



Examples of tasks:

- SST-2 – binary sentiment classification (we saw it in the text classification lecture)
- CoLA (Corpus of Linguistic Acceptability) – say whether a sentence is linguistically acceptable

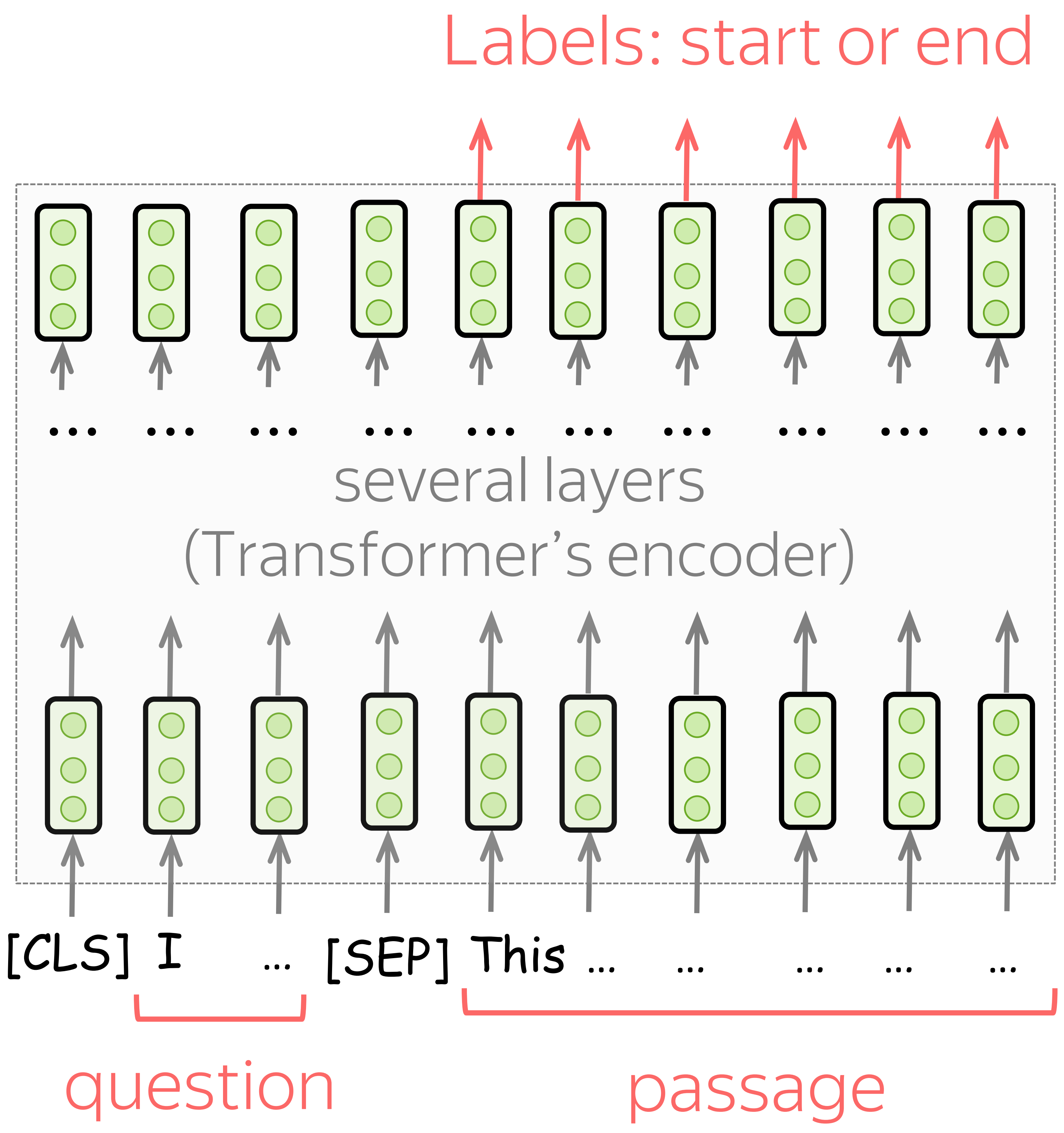
Finetuning BERT: Sentence Pair Classification



Examples of tasks:

- MLNI – entailment classification. Given a pair of sentences, say of the second is an **entailment**, **contradiction** or **neutral**
- QQP (Quora Question Pairs) – given two questions say if they are semantically equivalent
- STS-B – given two sentences return a similarity score from 1 to 5

Finetuning BERT: Question Answering



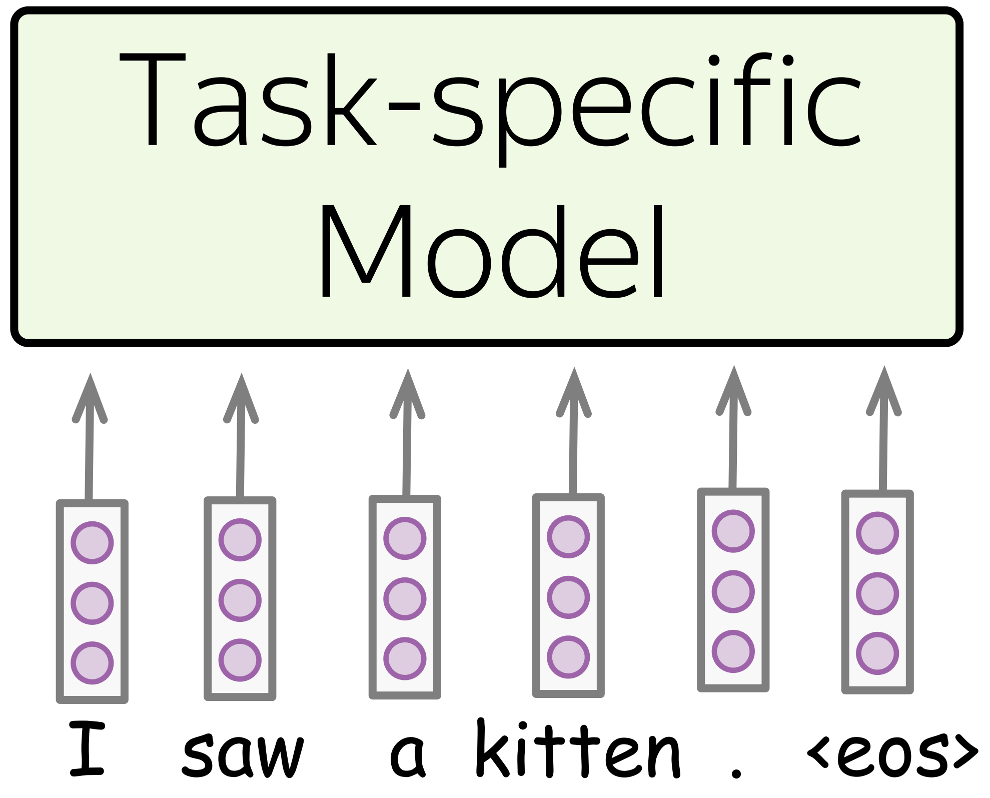
Examples of tasks:

- SQUAD – dataset with pairs of question-passage; the passage contains the answer – need to indicate where

ELMo: What’s changed?

Before

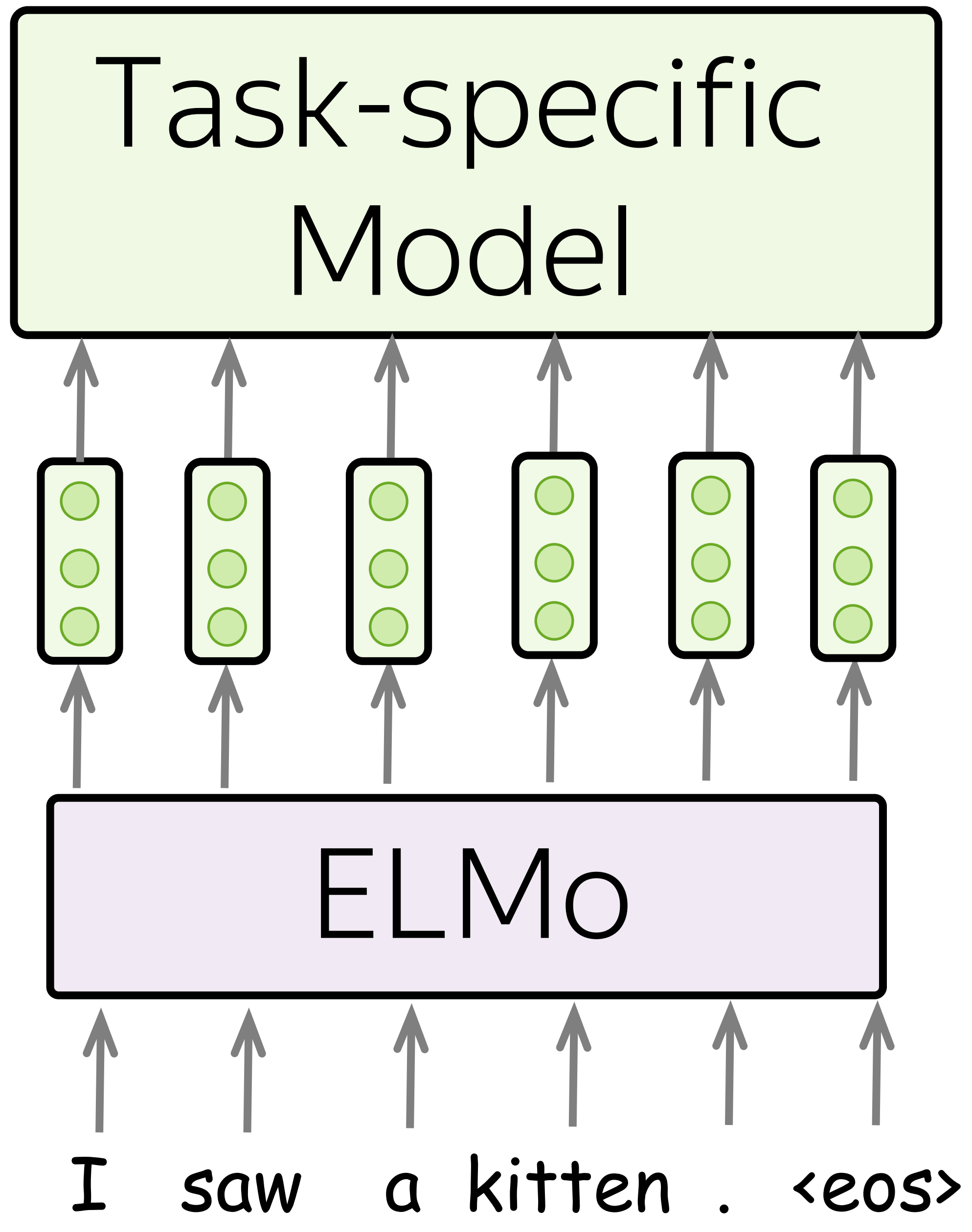
Word Embeddings



Word2Vec, GloVe, ... - saw a lot of data, know a lot

Model is the same

After



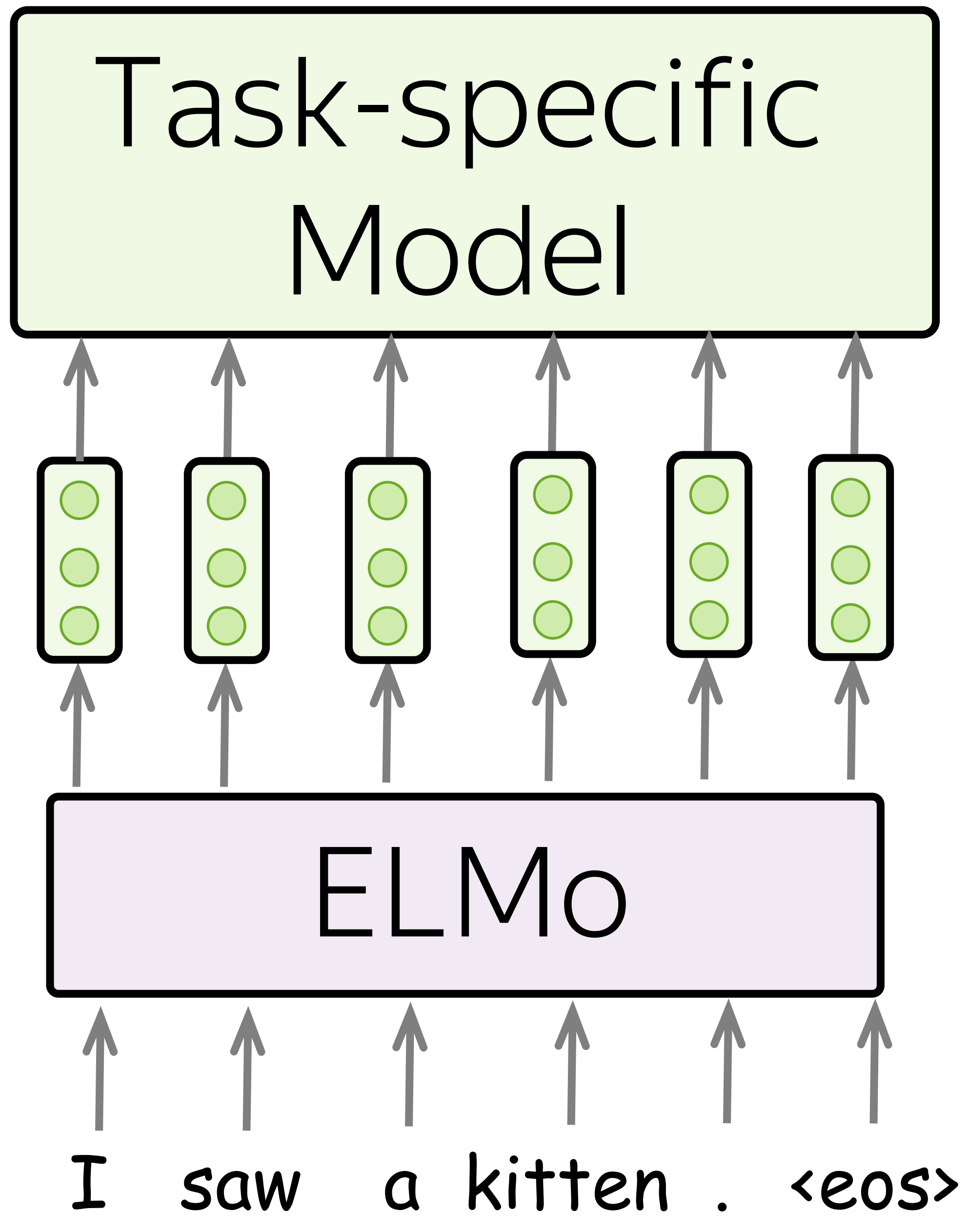
Representations of words **in context**

We used huge unlabeled dataset to learn not only words, but **how to process entire texts**

BERT: What's changed?

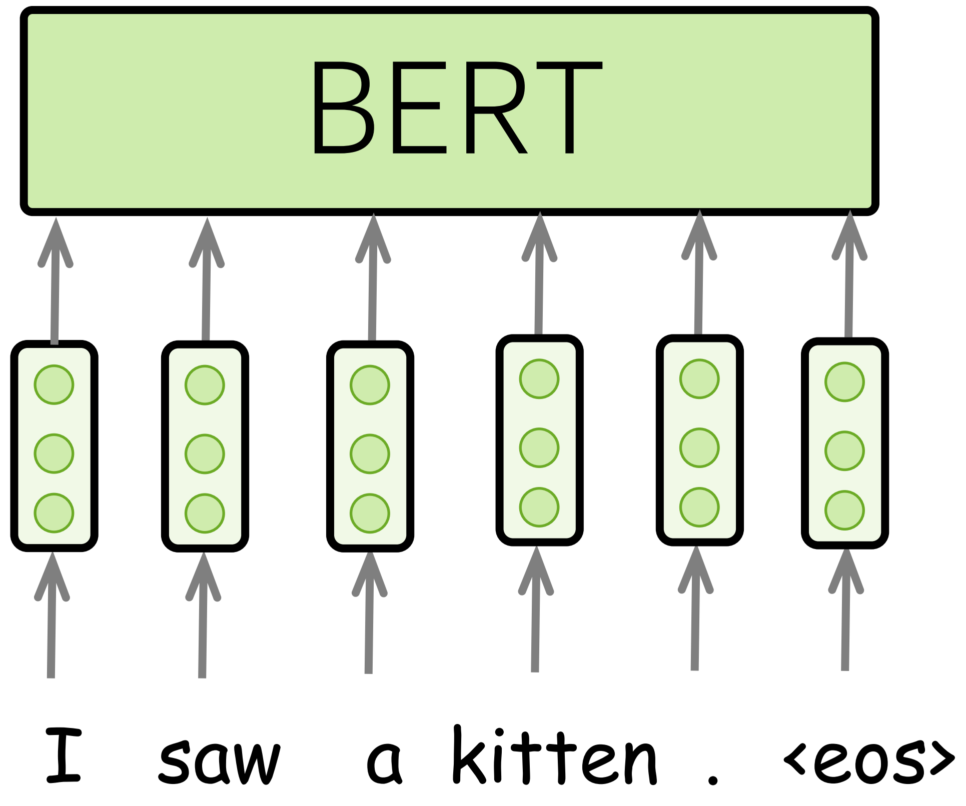
Before

Representations of words in context



After

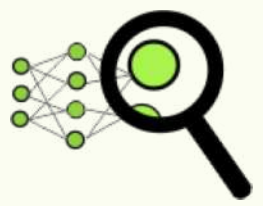
No task-specific model at all!



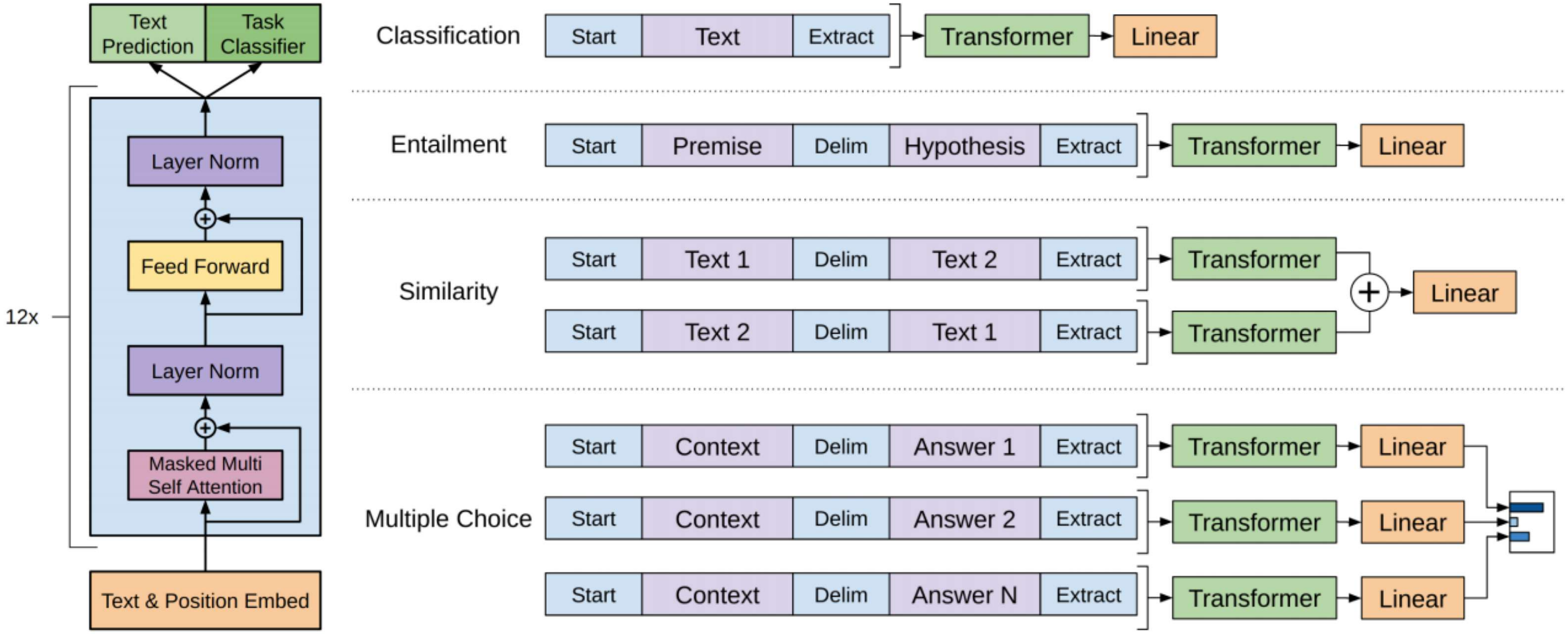
What is going to happen:

- Transfer Learning Idea
- Pretrained Models
  - (recap) Word Embeddings
  - ELMo
  - BERT
  - (a note on) GPT
  - (a note on) Adaptors
- Analysis and Interpretability

What is going to happen:

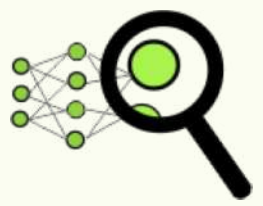
- Transfer Learning Idea
- Pretrained Models
    - (recap) Word Embeddings
    - ELMo
    - BERT
    - (a note on) GPT
    - (a note on) Adaptors
-  Analysis and Interpretability

GPT(1-2-3): Transformer Decoder



The figure is from the paper Improving Language Understanding by Generative Pretraining

What is going to happen:

- Transfer Learning Idea
- Pretrained Models
    - (recap) Word Embeddings
    - ELMo
    - BERT
    - (a note on) GPT
    - (a note on) Adaptors
-  Analysis and Interpretability

What is going to happen:

- Transfer Learning Idea
- Pretrained Models
    - (recap) Word Embeddings
    - ELMo
    - BERT
    - (a note on) GPT
    - (a note on) Adaptors
-  Analysis and Interpretability

Adaptors: Parameter-Efficient Adaptation

Finetuning:

- need a new (huge!) model for each task

Parameters updated: 100%

Adaptors: Parameter-Efficient Adaptation

Finetuning:

- need a new (huge!) model for each task

Parameters updated: 100%

Adaptors:

- model is fixed, train only small adaptors

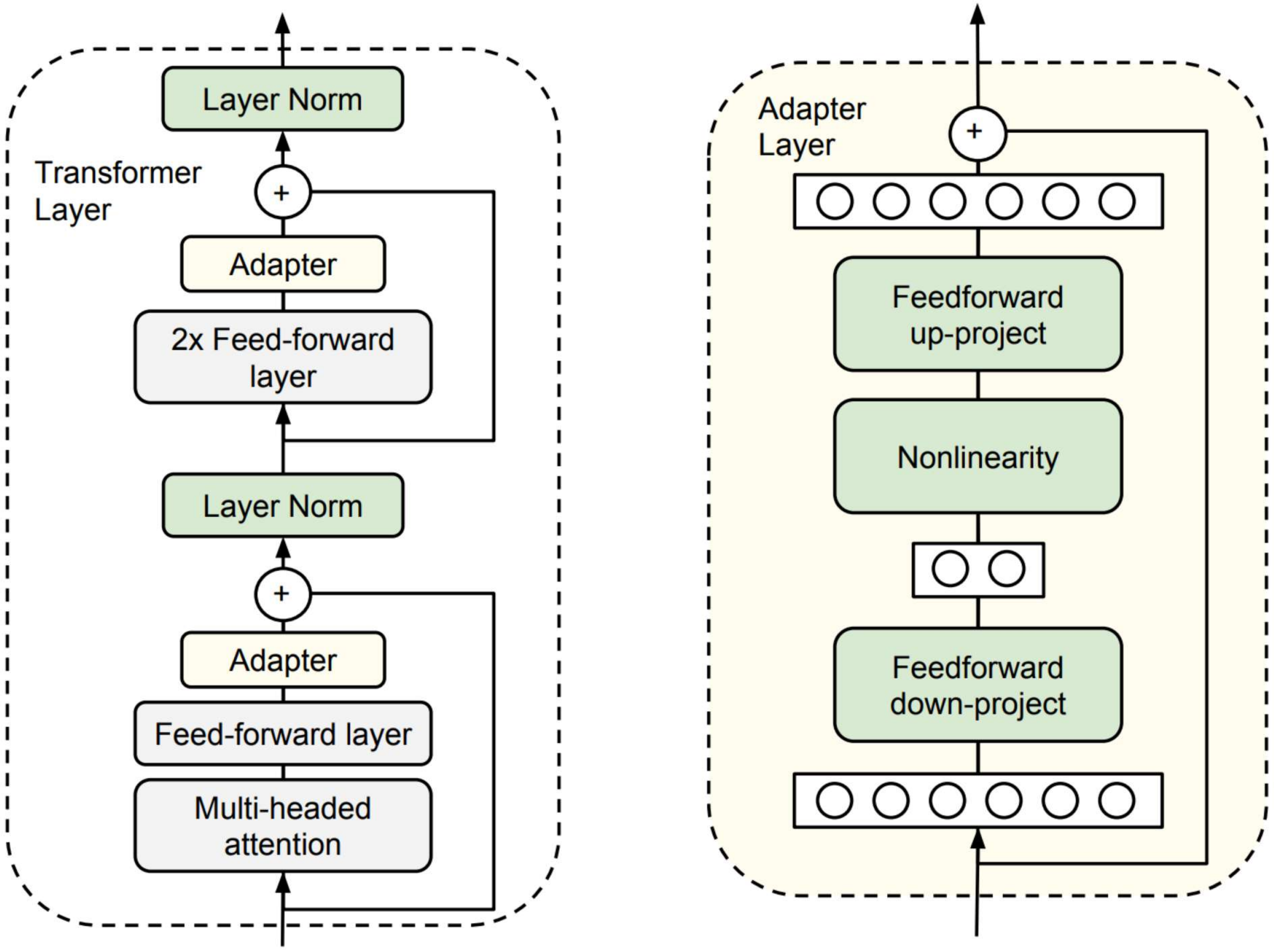
Adaptors: Parameter-Efficient Adaptation

Finetuning:
- need a new (huge!) model for each task

Parameters updated: 100%

Adaptors:
- model is fixed, train only small adaptors

Parameters updated: ≈1%



The figure is from the paper Parameter-Efficient Transfer Learning for NLP

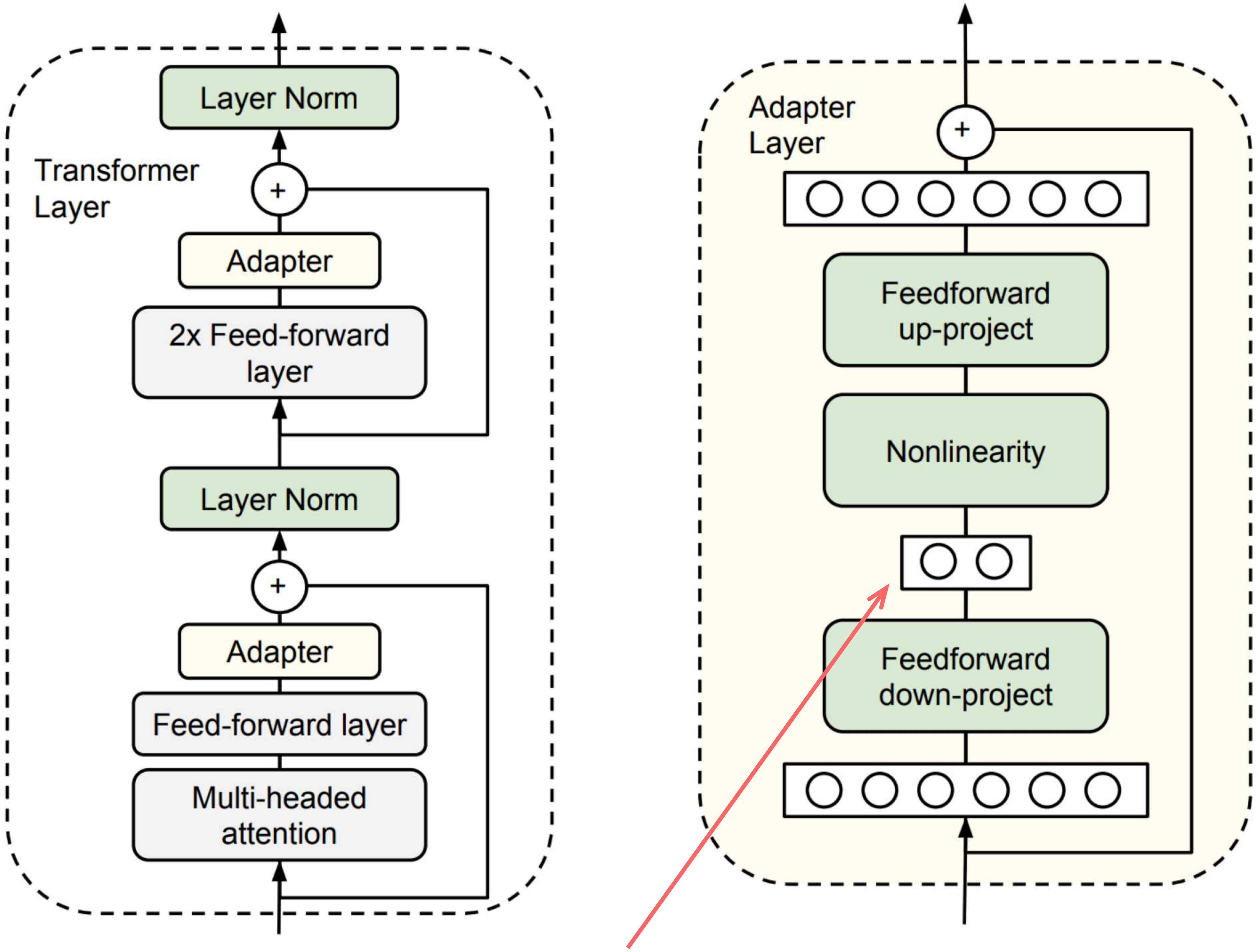
Adaptors: Parameter-Efficient Adaptation

Finetuning:
- need a new (huge!) model for each task

Parameters updated: 100%

Adaptors:
- model is fixed, train only small adaptors

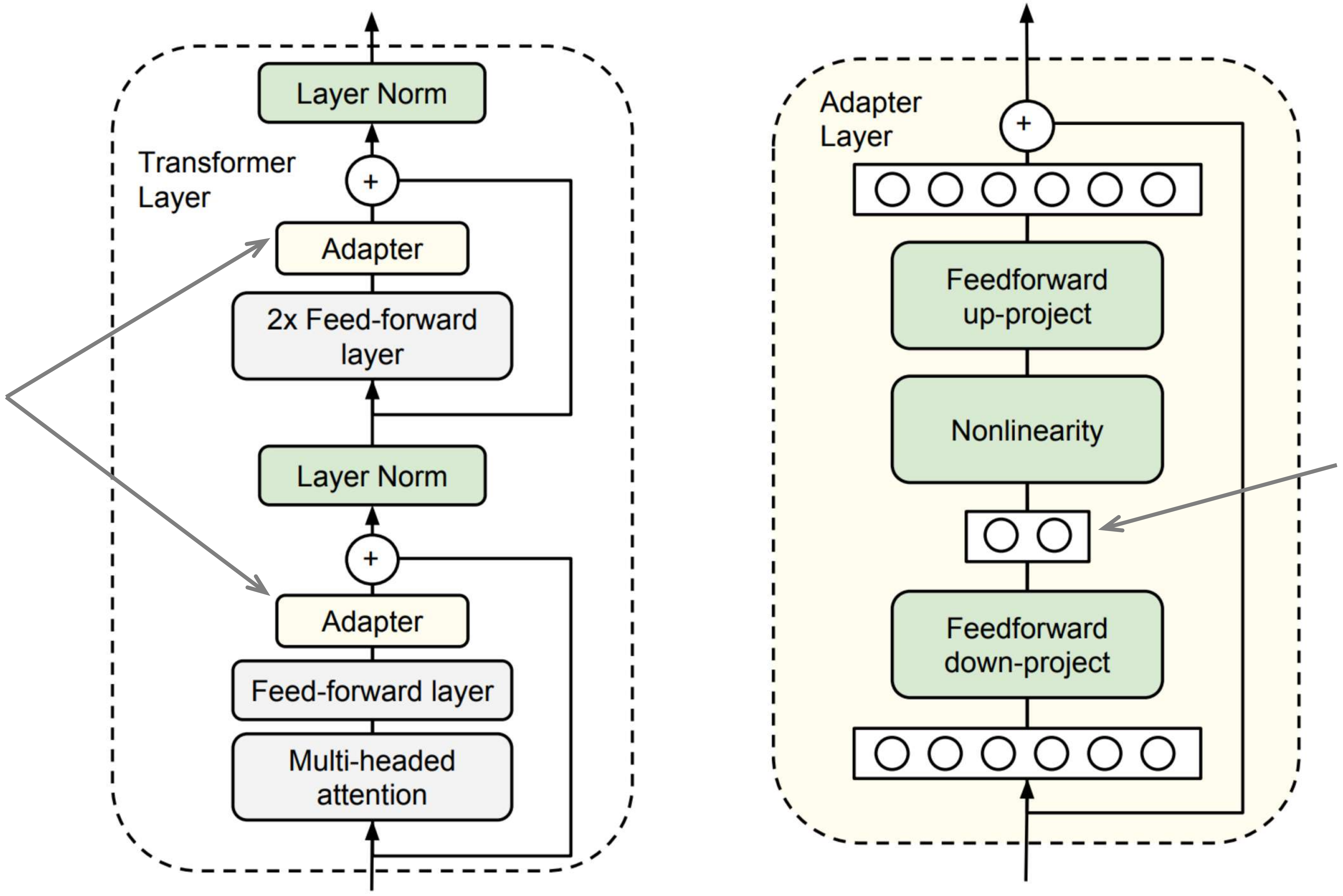
Parameters updated: ≈1%



This is small ⟹ only a few new parameters

The figure is from the paper Parameter-Efficient Transfer Learning for NLP

Only these are trained, everything else is fixed and is the same for all tasks



Small hidden size, i.e. an adaptor has only a few parameters (which is good!)

Other Research Directions

- Pretraining Objectives
- How to fine-tune
- How to Adapt
- How to modify for a new task (e.g. image-to-text, multilingual)

LoRA: Low-Rank Adaptation

COMS4054A/COMS7066A
Natural Language Processing

Mohammad Zaid
Moonsamy

Background

-> Large Language Models, including GPT, LLaMa, Claude, and others, have demonstrated remarkable abilities across a wide range of tasks, from text generation to deep language comprehension.

-> These LLMs are huge and great but very generic. To use these models for specific in-domain tasks, one has to fine-tune these models.

-> Fine-tuning involves training a pre-trained model on a smaller, task-specific dataset to enhance its performance within a particular domain or for a specific task.

Problems with Traditional Fine-Tuning

Traditional fine-tuning requires training all of the model's parameters, which present several challenges

Time: Training a large number of parameters significantly increases the time required to fine-tune the model.

Computational Resources: A higher number of trainable parameters increases the demand for computational power, making it more expensive and time-consuming to fine-tune large models.

Memory Usage: More trainable parameters require greater memory capacity, leading to higher reliance on disk reads, which can slow down training and reduce efficiency.

Catastrophic Forgetting: When fine-tuning on a specific task, the model may lose its ability to generalise to previously learned tasks, as the new training can overwrite essential knowledge acquired from the original dataset.

Prior Solutions

Adapter Layers

- Introduces inference latency

Prefix Tuning

- Difficult to optimise and can reduce the sequence length available for downstream tasks

Batch Size Sequence Length $\|\Theta\|$	32 512 0.5M	16 256 11M	1 128 11M
Fine-Tune/LoRA	1449.4±0.8	338.0±0.6	19.8±2.7
Adapter^L	1482.0±1.0 (+2.2%)	354.8±0.5 (+5.0%)	23.9±2.1 (+20.7%)
Adapter^H	1492.2±1.0 (+3.0%)	366.3±0.5 (+8.4%)	25.8±2.2 (+30.3%)

Solution: LoRA (Low-Rank Adaptation)

-> LoRA (Low Rank Adaptation) is a parameter efficient fine-tuning technique that reduces the number of trainable parameters of a model

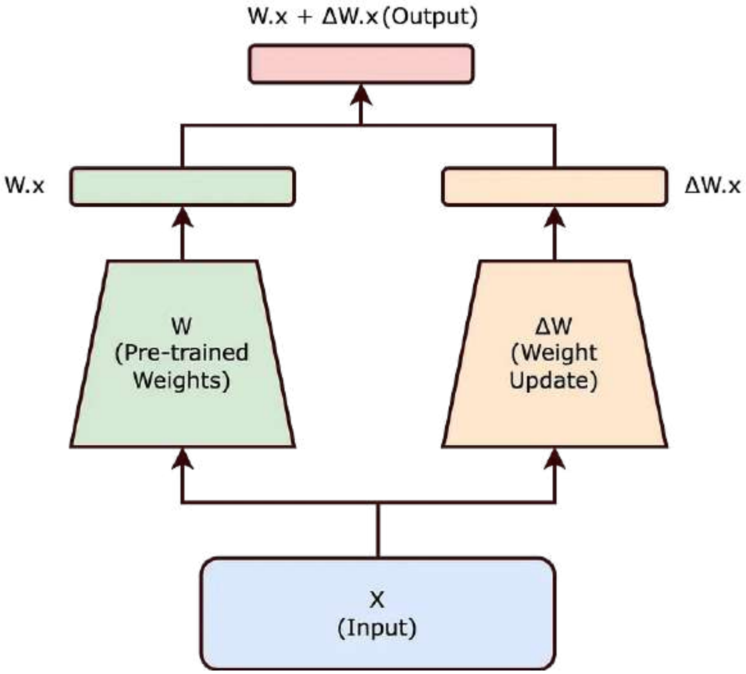
-> LoRA freezes the pretrained model weights and adds trainable rank decomposition matrices to each layer of the model

-> LoRA enables a fast, cost-effective, and efficient solution to the problems encountered by the full fine-tuning method

-> “LoRA can reduce the number of trainable parameters by 10,000 times and the GPU memory requirement by 3 times” for GPT3 175B

LoRA: How it works

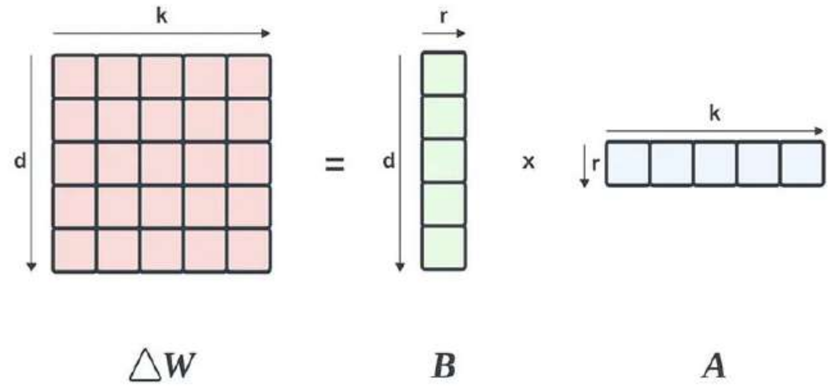
- Suppose our model has initial pretrained weights $\mathbf{W_o}$
- The learned weights can be represented as $\mathbf{\Delta W}$.
- $\mathbf{W'} = \mathbf{W_o} + \mathbf{\Delta W}$.
- if the pretrained weight matrix $\mathbf{W_o}$ was of size d x d then $\mathbf{\Delta W}$ is also d x d.
- Problem: computing the matrix $\mathbf{\Delta W}$ can be very compute and memory intensive.



LoRA: How it works

Intrinsic Rank Hypothesis

The intrinsic rank hypothesis suggests that significant changes to the neural network can be captured using a lower-dimensional representation.



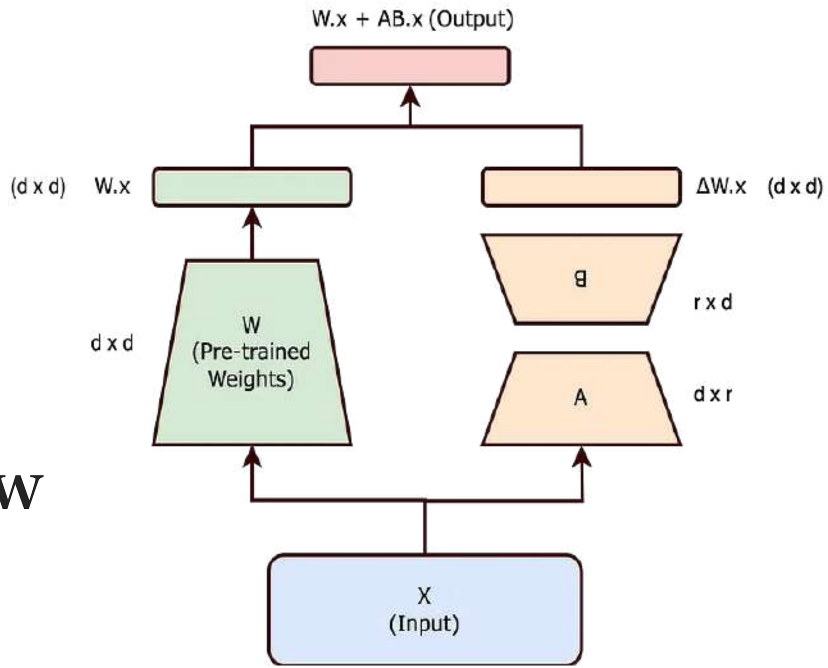
LoRA: How it works

Using this hypothesis, we can now represent $\mathbf{\Delta W}$ using smaller matrices $\mathbf{A}$ and $\mathbf{B}$:

$$\mathbf{W'} = \mathbf{Wo} + \mathbf{BA}.$$

The matrices $\mathbf{A}$ and $\mathbf{B}$ are of lower dimensionality, with their product $\mathbf{BA}$ representing a low-rank approximation of $\mathbf{\Delta W}$

We reduce the no. of trainable parameters significantly

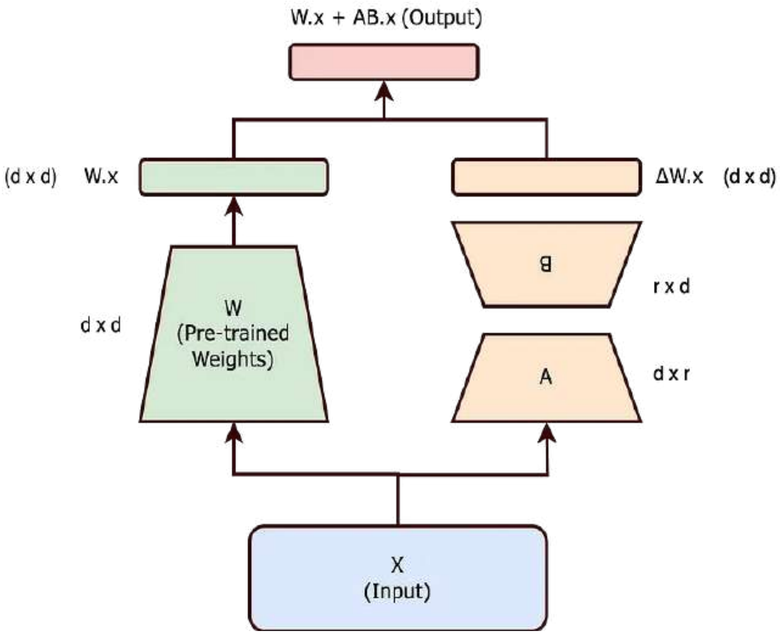


LoRA: How it works

Updating the initial weight matrix, $\mathbf{W_0}$ (d x d), would involve $\mathbf{d^2}$ parameters.

However with LoRA, we have smaller matrices $\mathbf{A}$ and $\mathbf{B}$, which are of sizes (d x r) and (r x d).

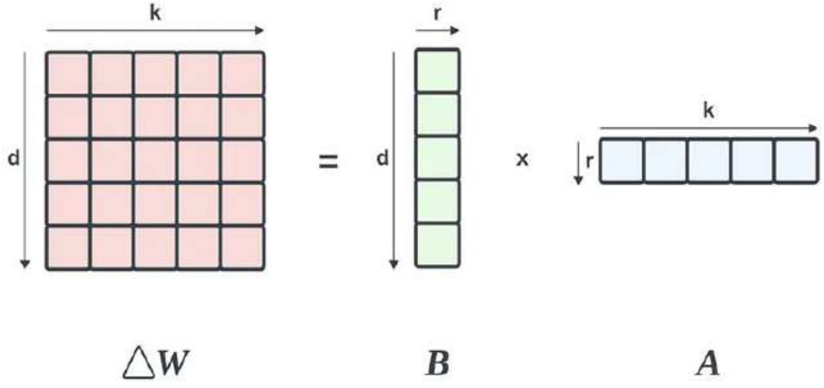
Thus, the total number of parameters we have to update reduces to (**2dr**), which is much smaller when **r << d**.



LoRA: Example

If $\mathbf{Wo}$ is of size d x k, where d = 30 and k = 10 then the number of parameters to be updated will be 30 x 10 = **300 parameters**

LoRA: Using rank **r = 2**, Then the number of parameters reduces to (d x r) + (r x k) = (30 x 2) + (2 x 10) = 60 + 20 = **80 parameters**



Note: While matrices $\mathbf{A}$ and $\mathbf{B}$ do not capture all information from $\mathbf{\Delta W}$, the LoRA method is effective due to the intrinsic rank hypothesis, meaning a lower rank can still capture the key information needed for adaptation.

GPT-3 LoRA performance

Model&Method	# Trainable Parameters	WikiSQL	MNLI-m	SAMSum
		Acc. (%)	Acc. (%)	R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 ($\text{Adapter}^{\text{H}}$)	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 ($\text{Adapter}^{\text{H}}$)	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

Advantages of LoRA

- Fast, cost-effective and efficient solution to the traditional fine-tuning problem
- Save computational resources as only the lower rank matrices are optimised
- Less trainable parameters mean less training time
- LoRA can be combined with other prior methods such as prefix tuning
- No inference latency
- Reduced checkpoint sizes (e.g. GPT-3: 1 TB to around 25 MB per checkpoint)

How to choose rank r?

- A lower rank would mean less number of trainable parameters while a higher rank would mean high number of parameters, eventually, converge to full fine-tuning
- LoRA authors show that a low rank value of 1 or 2 is sufficient even when the highest rank value can go upto 12288. This proves LORA's efficiency
- r=8 is the standard default value for rank

Additional Insights

-> LoRA can be applied to any model that makes use of matrix multiplications, even SVMs

-> If LoRA underperforms, we can always adapt more parameters by increasing the rank

-> Alpha hyperparameter

QLoRA: Quantized LoRA

-> Quantized LoRA (**QLoRA**) combines the efficiency of LoRA with the benefits of quantization.

-> Quantization reduces the precision of the model’s weights (e.g. 32-bit floating point numbers to 8-bit integers)

-> QLoRA requires less memory compared to LoRA, making it even more efficient

-> It is great for hardware devices with limited resources

Other Uses of LoRA

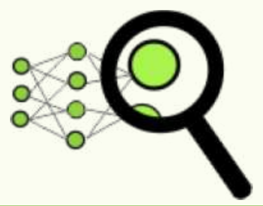
Stable Diffusion LoRA

Combining LoRA with prefix tuning

Benefits

- LoRA reduces parameters; prefix tuning offers task-specific control
- Combined, they allow efficient fine-tuning with less data and compute

What is going to happen:

- Transfer Learning Idea
- Pretrained Models
-  Analysis and Interpretability

Analysis Methods

The methods we used previously:

- (model-specific) looking at model components

Analysis Methods

The methods we used previously:

- (model-specific) looking at model components → Heads in Multi-Head Attention (BERT)

Analysis Methods

The methods we used previously:

- (model-specific) looking at model components → Heads in Multi-Head Attention (BERT)
- (model-agnostic) probing for linguistic structure

Analysis Methods

<u>The methods we used previously:</u>

- (model-specific) looking at model components → Heads in Multi-Head Attention (BERT)
- (model-agnostic) probing for linguistic structure → BERT and the classical NLP pipeline

Analysis Methods

The methods we used previously:

- (model-specific) looking at model components → Heads in Multi-Head Attention (BERT)
- (model-agnostic) probing for linguistic structure → BERT and the classical NLP pipeline
- (model-agnostic) looking at predictions and evaluating specific phenomena

Analysis Methods

The methods we used previously:

- (model-specific) looking at model components → Heads in Multi-Head Attention (BERT)
- (model-agnostic) probing for linguistic structure → BERT and the classical NLP pipeline
- (model-agnostic) looking at predictions and evaluating specific phenomena

Analysis Methods

The methods we used previously:

- (model-specific) looking at model components → Heads in Multi-Head Attention (BERT)
- (model-agnostic) probing for linguistic structure → BERT and the classical NLP pipeline
- (model-agnostic) looking at predictions and evaluating specific phenomena → BERT as knowledge base

What we will see in this lecture:

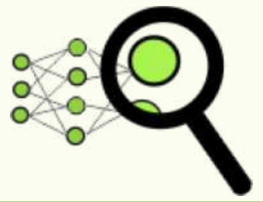
- Heads in Multi-Head Attention (BERT)
- BERT and the classical NLP pipeline
- BERT as knowledge base

Analysis Methods

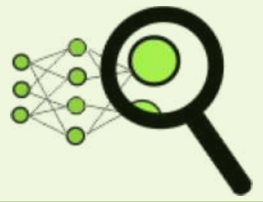
The methods we used previously:

- (model-specific) looking at model components → Heads in Multi-Head Attention (BERT)
- (model-agnostic) probing for linguistic structure → BERT and the classical NLP pipeline
- (model-agnostic) looking at predictions and evaluating specific phenomena → BERT as knowledge base

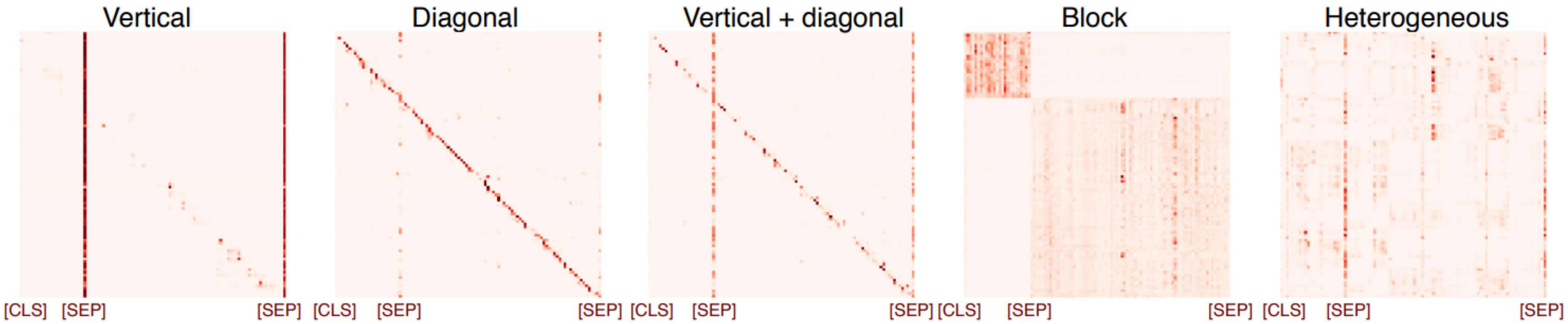
What is going to happen:

- Transfer Learning Idea
- Pretrained Models
-  Analysis and Interpretability

What is going to happen:

- Transfer Learning Idea
- Pretrained Models
-  Analysis and Interpretability
    - Model Components
    - Probing
    - Looking at Predictions

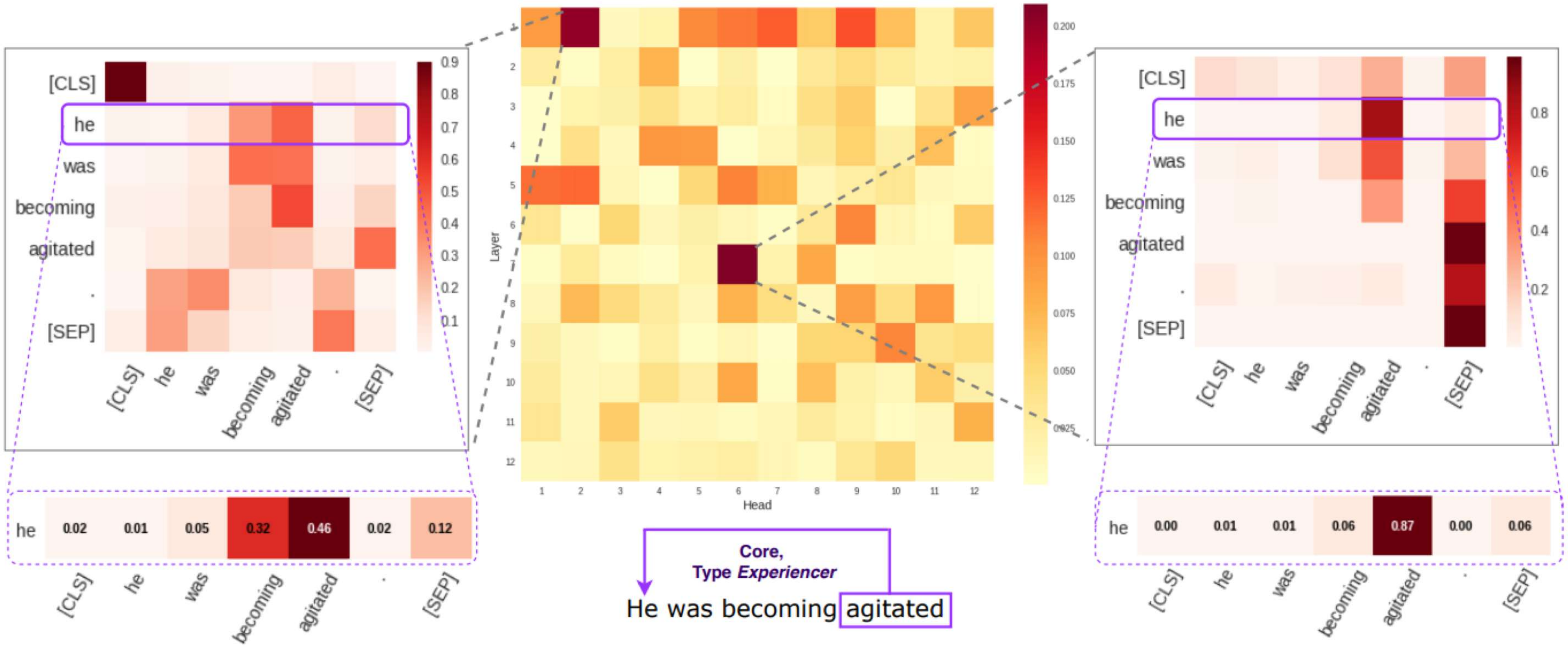
BERT Self-Attention Heads



Typical self-attention patterns

The figure is from the paper Revealing the Dark Secrets of BERT

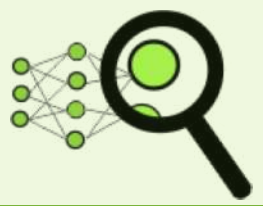
BERT Self-Attention Heads



Heads that encode information correlated to semantic links in the input text

The figure is from the paper Revealing the Dark Secrets of BERT

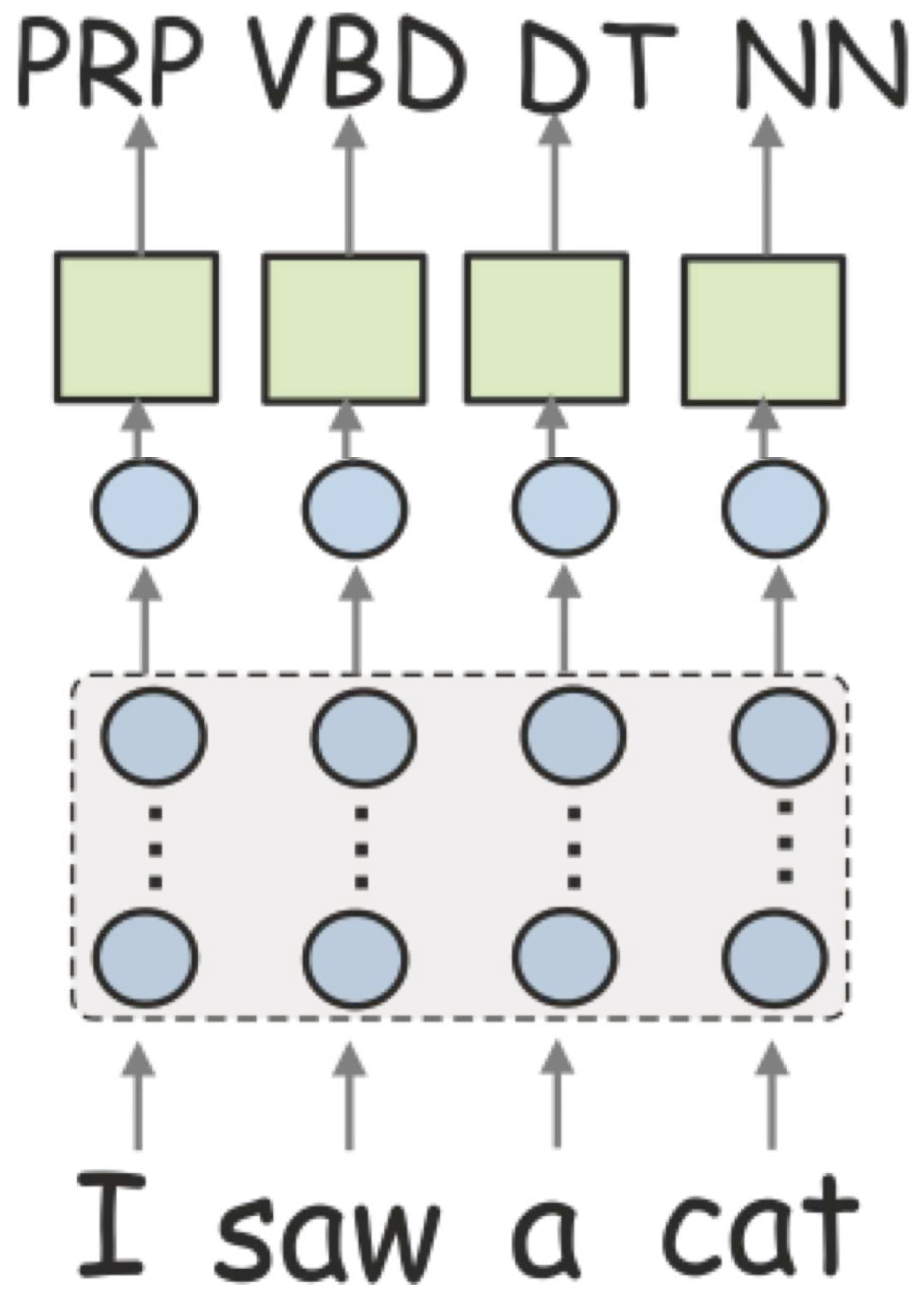
What is going to happen:

- Transfer Learning Idea
- Pretrained Models
-  Analysis and Interpretability
    - Model Components
    - Probing
    - Looking at Predictions

What is going to happen:

- Transfer Learning Idea
- Pretrained Models
- Analysis and Interpretability
    - Model Components
    - Probing
    - Looking at Predictions

RECAP: Probing for linguistic structure



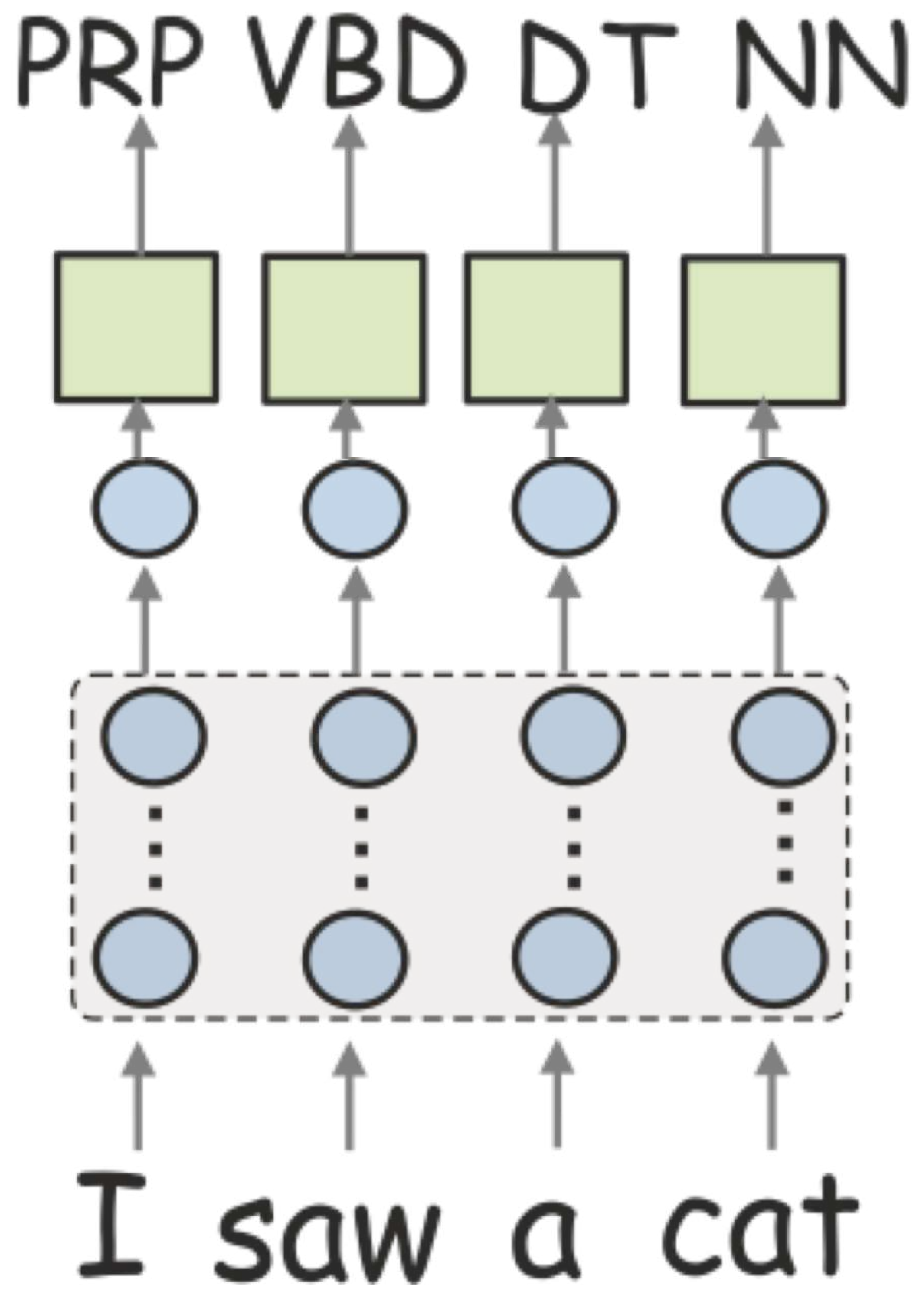
Labels: $y_{1:n}$

Probing classifier

Representations: $x_{1:n}$

Pretrained model

RECAP: Probing for linguistic structure



Labels: $y_{1:n}$

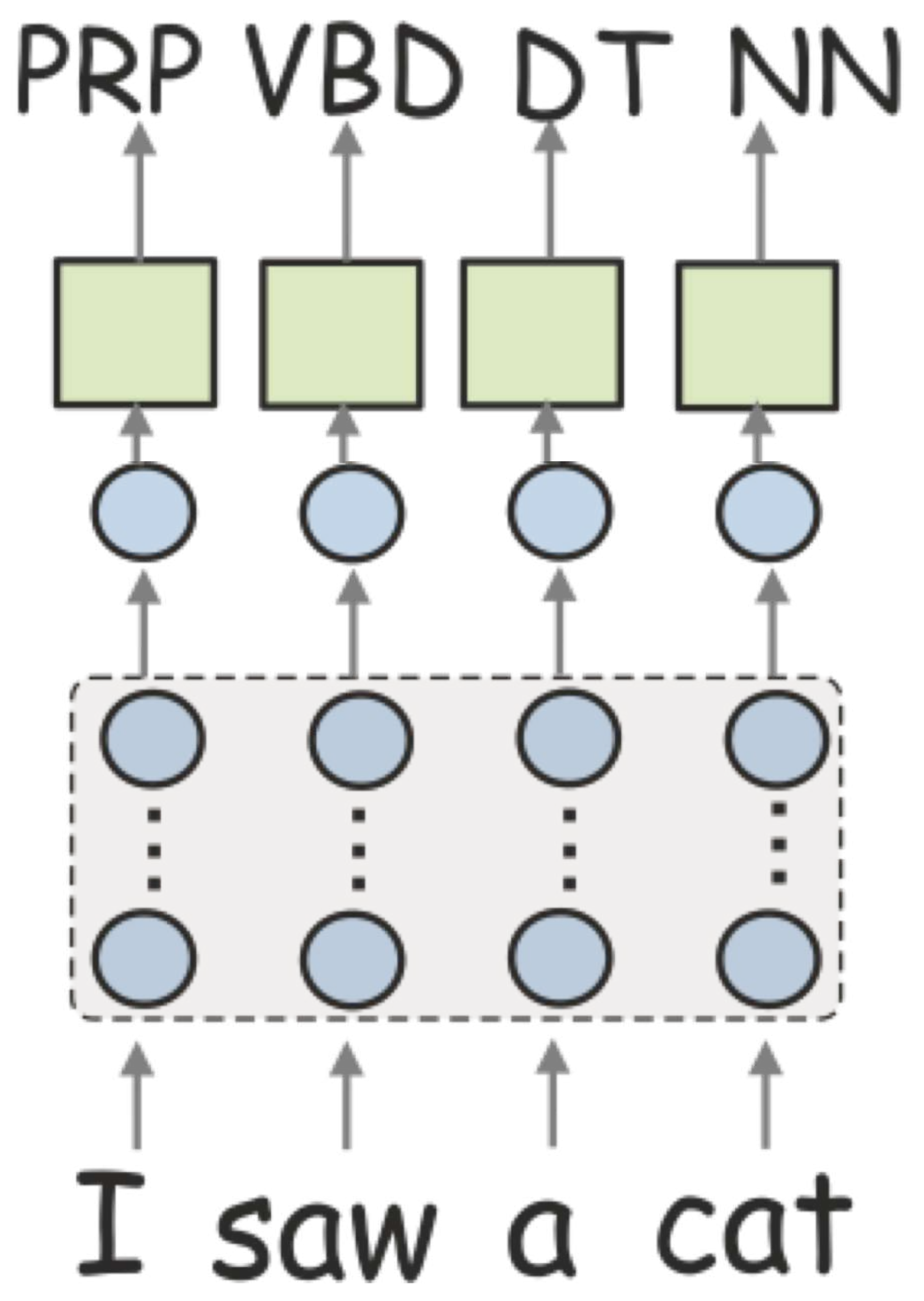
Probing classifier

Representations: $x_{1:n}$

Pretrained model

RECAP: Probing for linguistic structure

After training a probing classifier, look at these weights



Labels: $y_{1:n}$

Probing classifier

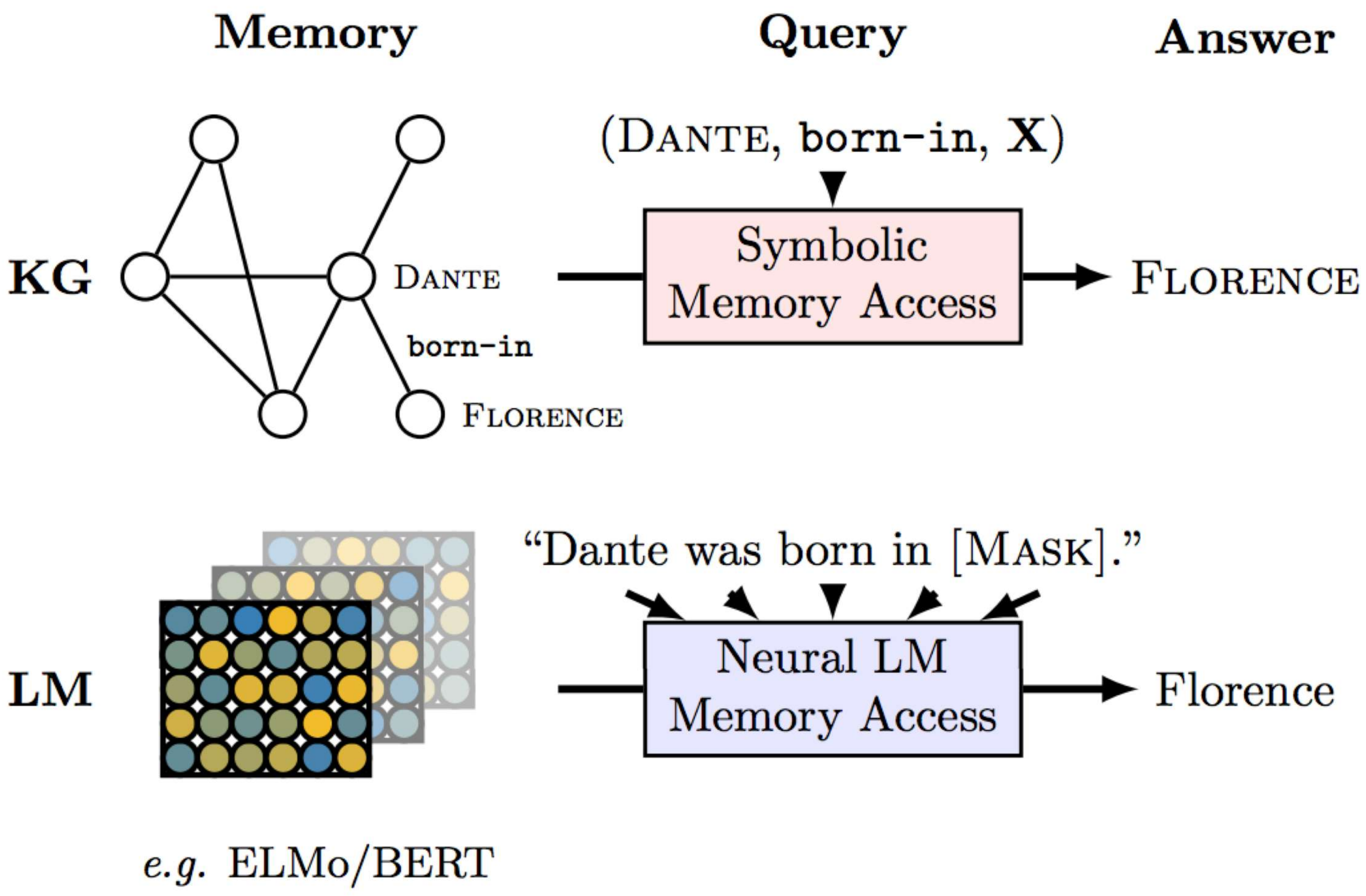
Representations: use weights for layers (like in ELMo)

Pretrained model

What is going to happen:

- Transfer Learning Idea
- Pretrained Models
- Analysis and Interpretability
    - Model Components
    - Probing
    - Looking at Predictions

Language Models as Knowledge Bases?



The figure is from the paper Language Models as Knowledge Bases?

Language Models as Knowledge Bases?

	Relation	Query	Answer	Generation
T-Rex	P19	Francesco Bartolomeo Conti was born in ____.	Florence	Rome [-1.8] , **Florence [-1.8]** , Naples [-1.9] , Milan [-2.4] , Bologna [-2.5]
	P20	Adolphe Adam died in ____.	Paris	**Paris [-0.5]** , London [-3.5] , Vienna [-3.6] , Berlin [-3.8] , Brussels [-4.0]
	P279	English bulldog is a subclass of ____.	dog	dogs [-0.3] , breeds [-2.2] , **dog [-2.4]** , cattle [-4.3] , sheep [-4.5]
	P37	The official language of Mauritius is ____.	English	**English [-0.6]** , French [-0.9] , Arabic [-6.2] , Tamil [-6.7] , Malayalam [-7.0]
	P413	Patrick Oboya plays in ____ position.	midfielder	centre [-2.0] , center [-2.2] , **midfielder [-2.4]** , forward [-2.4] , midfield [-2.7]
	P138	Hamburg Airport is named after ____.	Hamburg	Hess [-7.0] , Hermann [-7.1] , Schmidt [-7.1] , **Hamburg [-7.5]** , Ludwig [-7.5]
	P364	The original language of Mon oncle Benjamin is ____.	French	**French [-0.2]** , Breton [-3.3] , English [-3.8] , Dutch [-4.2] , German [-4.9]
	P54	Dani Alves plays with ____ .	Barcelona	Santos [-2.4] , Porto [-2.5] , Sporting [-3.1] , Brazil [-3.3] , Portugal [-3.7]
	P106	Paul Toungui is a ____ by profession .	politician	lawyer [-1.1] , journalist [-2.4] , teacher [-2.7] , doctor [-3.0] , physician [-3.7]
	P527	Sodium sulfide consists of ____.	sodium	water [-1.2] , sulfur [-1.7] , **sodium [-2.5]** , zinc [-2.8] , salt [-2.9]
	P102	Gordon Scholes is a member of the ____ political party.	Labor	Labour [-1.3] , Conservative [-1.6] , Green [-2.4] , Liberal [-2.9] , **Labor [-2.9]**
	P530	Kenya maintains diplomatic relations with ____.	Uganda	India [-3.0] , **Uganda [-3.2]** , Tanzania [-3.5] , China [-3.6] , Pakistan [-3.6]
	P176	iPod Touch is produced by ____.	Apple	**Apple [-1.6]** , Nokia [-1.7] , Sony [-2.0] , Samsung [-2.6] , Intel [-3.1]
	P30	Bailey Peninsula is located in ____.	Antarctica	**Antarctica [-1.4]** , Bermuda [-2.2] , Newfoundland [-2.5] , Alaska [-2.7] , Canada [-3.1]
	P178	JDK is developed by ____.	Oracle	IBM [-2.0] , Intel [-2.3] , Microsoft [-2.5] , HP [-3.4] , Nokia [-3.5]
	P1412	Carl III used to communicate in ____.	Swedish	German [-1.6] , Latin [-1.9] , French [-2.4] , English [-3.0] , Spanish [-3.0]
	P17	Sunshine Coast, British Columbia is located in ____.	Canada	**Canada [-1.2]** , Alberta [-2.8] , Yukon [-2.9] , Labrador [-3.4] , Victoria [-3.4]
	P39	Pope Clement VII has the position of ____ .	pope	cardinal [-2.4] , Pope [-2.5] , **pope [-2.6]** , President [-3.1] , Chancellor [-3.2]
	P264	Joe Cocker is represented by music label ____.	Capitol	EMI [-2.6] , BMG [-2.6] , Universal [-2.8] , **Capitol [-3.2]** , Columbia [-3.3]
	P276	London Jazz Festival is located in ____.	London	**London [-0.3]** , Greenwich [-3.2] , Chelsea [-4.0] , Camden [-4.6] , Stratford [-4.8]
	P127	Border TV is owned by ____.	ITV	Sky [-3.1] , **ITV [-3.3]** , Global [-3.4] , Frontier [-4.1] , Disney [-4.3]
	P103	The native language of Mammootty is ____.	Malayalam	**Malayalam [-0.2]** , Tamil [-2.1] , Telugu [-4.8] , English [-5.2] , Hindi [-5.6]
	P495	The Sharon Cuneta Show was created in ____.	Philippines	Manila [-3.2] , **Philippines [-3.6]** , February [-3.7] , December [-3.8] , Argentina [-4.0]

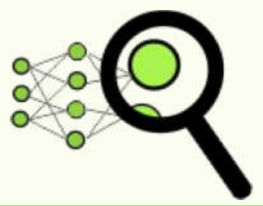
The figure is from the paper Language Models as Knowledge Bases?

Language Models as Knowledge Bases?

ConceptNet	AtLocation	You are likely to find a overflow in a ____.	drain	sewer [-3.1] , canal [-3.2] , toilet [-3.3] , stream [-3.6] , **drain [-3.6]**
	CapableOf	Ravens can ____.	fly	**fly [-1.5]** , fight [-1.8] , kill [-2.2] , die [-3.2] , hunt [-3.4]
	CausesDesire	Joke would make you want to ____.	laugh	cry [-1.7] , die [-1.7] , **laugh [-2.0]** , vomit [-2.6] , scream [-2.6]
	Causes	Sometimes virus causes ____.	infection	disease [-1.2] , cancer [-2.0] , **infection [-2.6]** , plague [-3.3] , fever [-3.4]
	HasA	Birds have ____.	feathers	wings [-1.8] , nests [-3.1] , **feathers [-3.2]** , died [-3.7] , eggs [-3.9]
	HasPrerequisite	Typing requires ____.	speed	patience [-3.5] , precision [-3.6] , registration [-3.8] , accuracy [-4.0] , **speed [-4.1]**
	HasProperty	Time is ____.	finite	short [-1.7] , passing [-1.8] , precious [-2.9] , irrelevant [-3.2] , gone [-4.0]
	MotivatedByGoal	You would celebrate because you are ____.	alive	happy [-2.4] , human [-3.3] , **alive [-3.3]** , young [-3.6] , free [-3.9]
	ReceivesAction	Skills can be ____.	taught	acquired [-2.5] , useful [-2.5] , learned [-2.8] , combined [-3.9] , varied [-3.9]
	UsedFor	A pond is for ____.	fish	swimming [-1.3] , fishing [-1.4] , bathing [-2.0] , **fish [-2.8]** , recreation [-3.1]

The figure is from the paper Language Models as Knowledge Bases?

What is going to happen:

- Transfer Learning Idea
- Pretrained Models
-  Analysis and Interpretability

Human Language Learning

- Learning language takes huge amounts of data
- But children do it relatively quickly – the “Poverty of the stimulus” argument by Chompsky
- One possible conclusion is that the capacity for language is genetic or built into our brains

The Universal Grammar

- This is known as the “Universal Grammar” also by Chompsky.

- Believes that there is a limited set of possible languages learnable by the brain.

- The brain then is very well tuned to learning structures in nature – compositional generalisation for example.

An Alternate Theory

- Instead, what if language is well tuned to our brains?
- Iterated Learning is one theory which puts forward this idea – by Kirby
- The debate around human's capacity for language is at the very heart of the origins of neural networks[1].

1. https://blogs.umass.edu/brain-wars/the-debates/pinker-and-prince-vs-rumelhart-and-mcclelland/

Iterated Learning

- Language structure is the result of multiple learners attempting to communicate.
- Each generation learns language from the one before.
- The parts of language which are easiest to learn will be remembered and dominate.
- The “communication bottleneck” results is a refinement of language.

Iterated Learning

- Procedure:
  - Initialize a network and obtain some data.
  - Train the network but stop early.
  - Relabel the data with the outputs (logits) of the network.
  - Train a new network on the modified labels.
  - Repeat for k generations.

Iterated Learning