Neural Language Modeling Cont. and Pretrained LMs

CSCI 1460: Computational Linguistics

Lecture 12

Ellie Pavlick

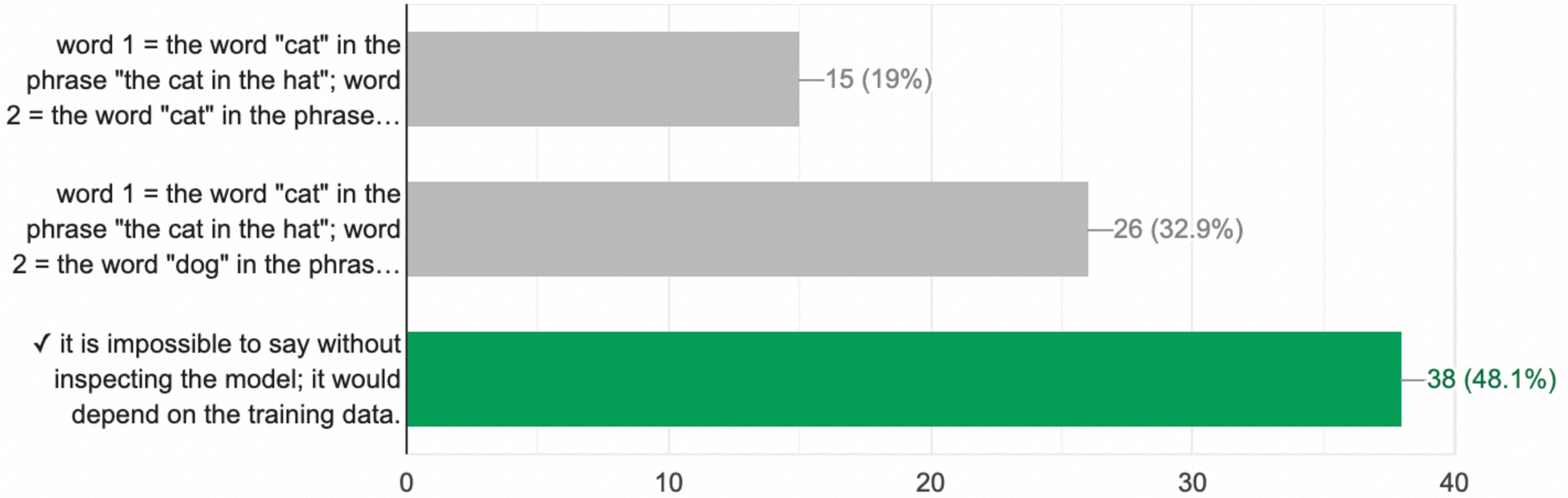
Fall 2022

Quiz Recap

Consider a Transformer language model that is initialized randomly and then trained on a language modeling task. Both the word embeddings and the positional encodings are randomly initialized and then trained via backprop. Which of the following pairs would be the most similar according to the final trained model?

Copy

38 / 79 correct responses



Quiz Recap

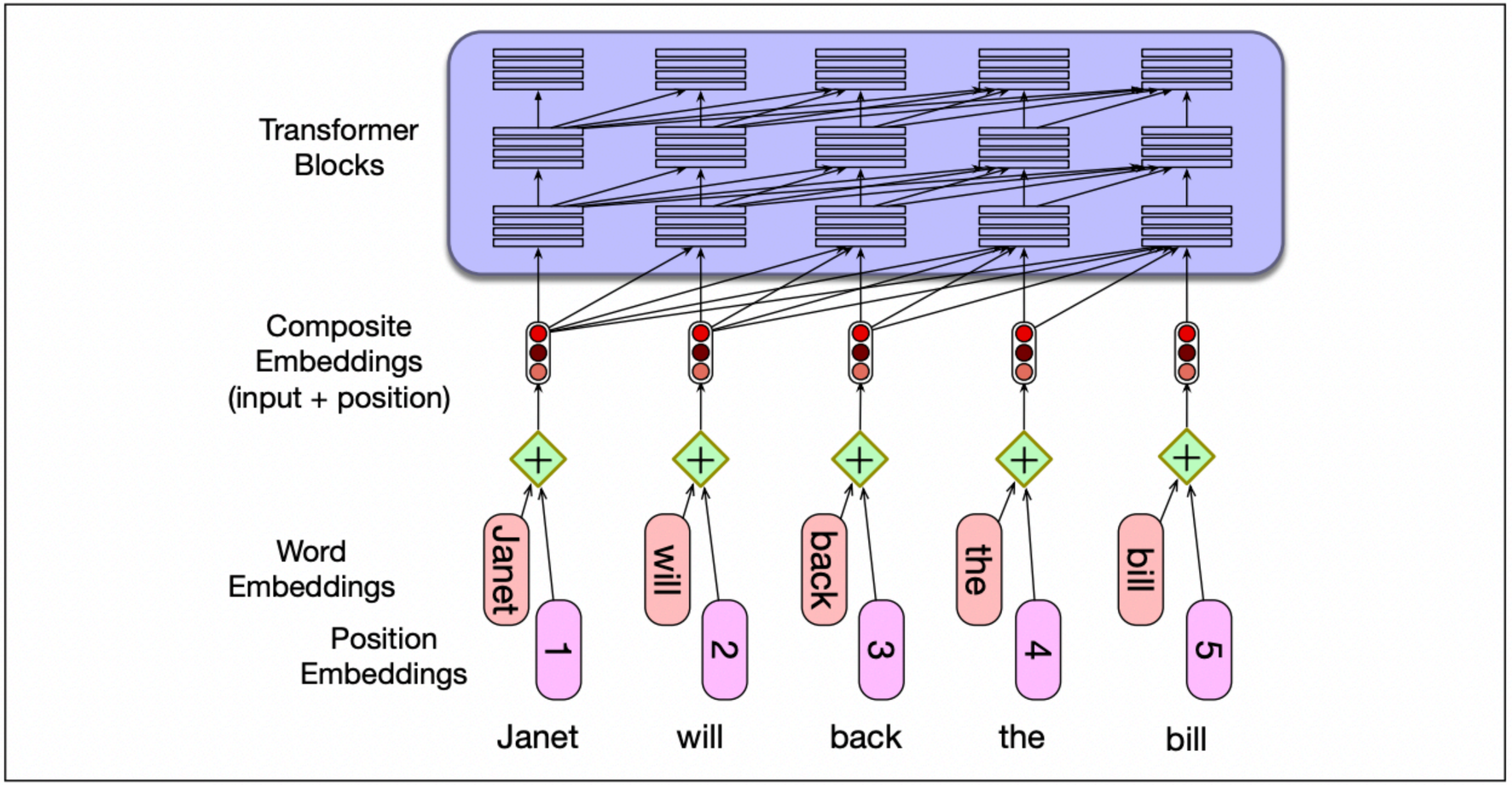


Figure 9.20 A simple way to model position: simply adding an embedding representation of the absolute position to the input word embedding.

Topics

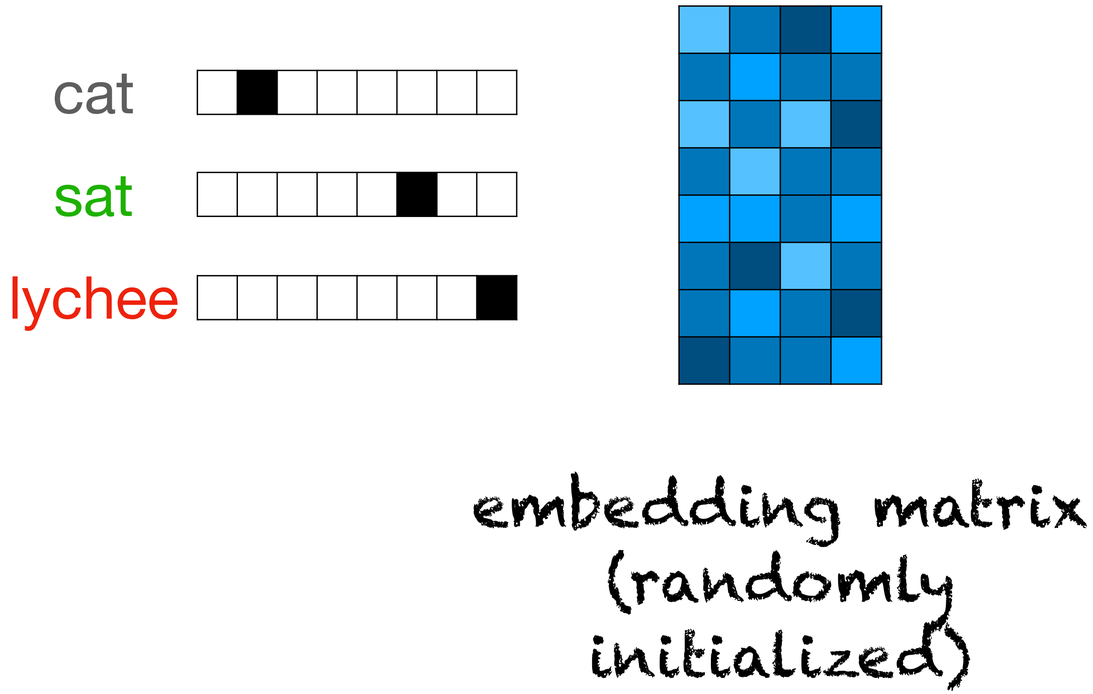
- Contextualized Word Representations:
  - ELMo
  - BERT
  - GPT
- Finetuning and Other Transfer Methods

Topics

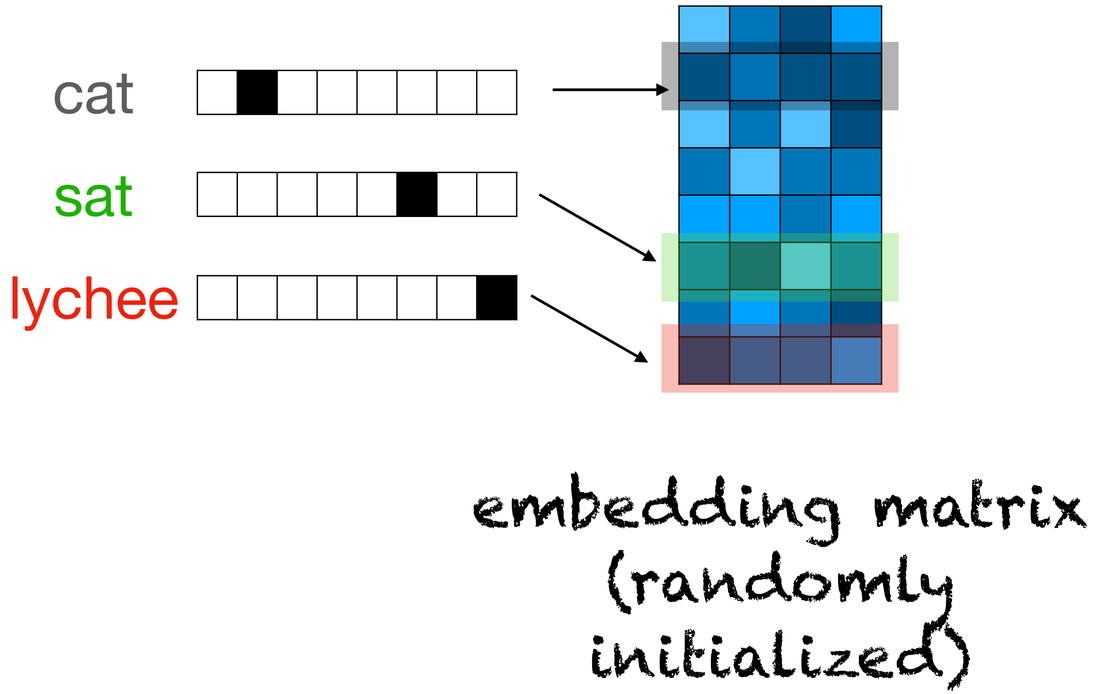
- **Contextualized Word Representations:**
  - ELMo
  - BERT
  - GPT
- Finetuning and Other Transfer Methods

(Regular/“Static”) Word Embeddings

(Regular/“Static”) Word Embeddings

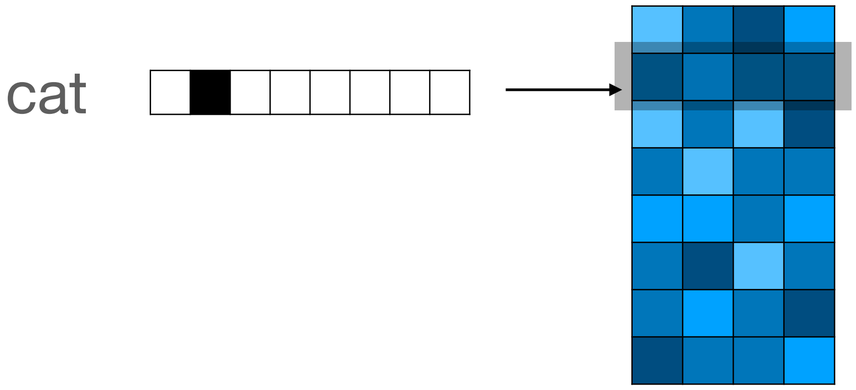


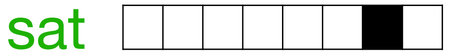
(Regular/“Static”) Word Embeddings



(Regular/“Static”) Word Embeddings

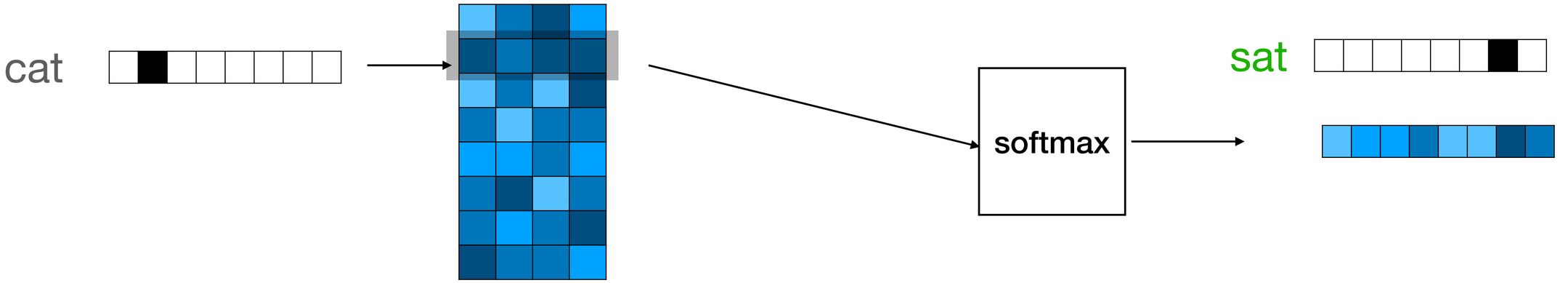
the **cat** sat





(Regular/“Static”) Word Embeddings

the **cat** sat

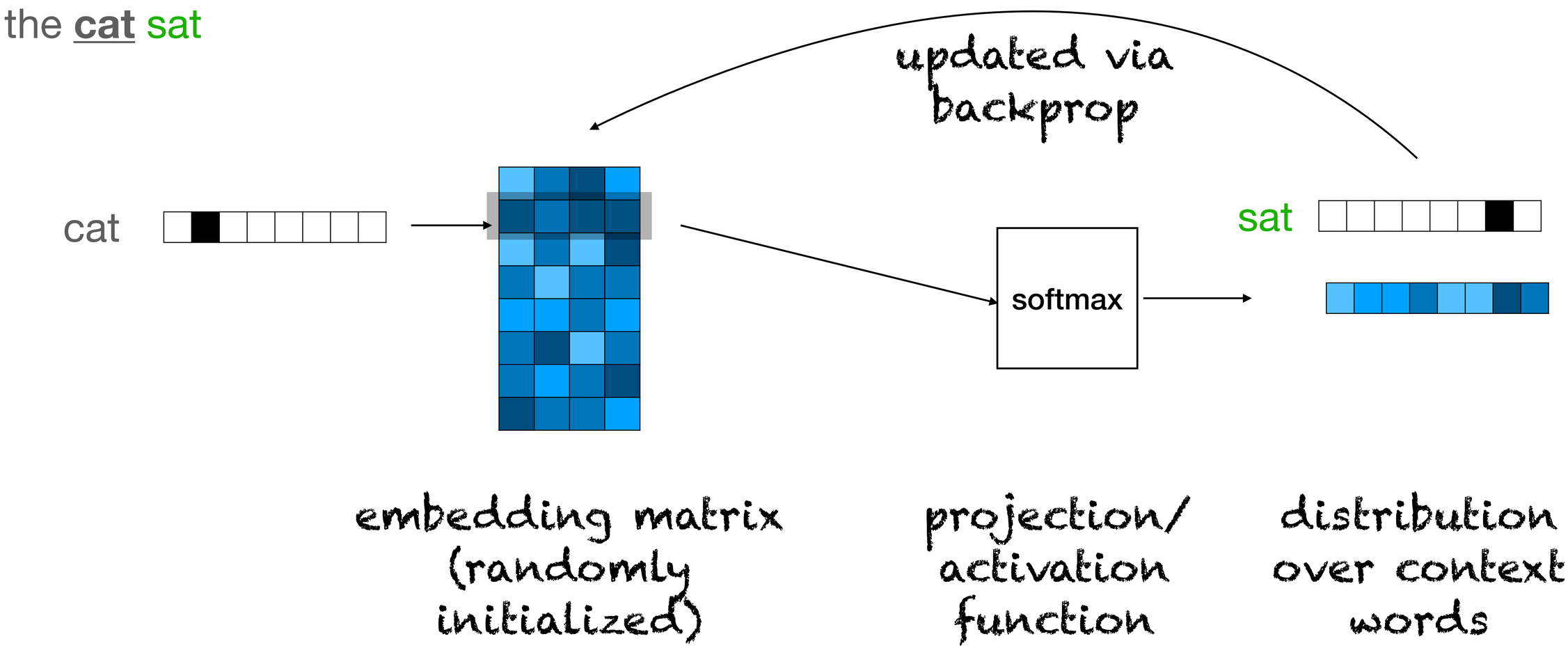


embedding matrix (randomly initialized)

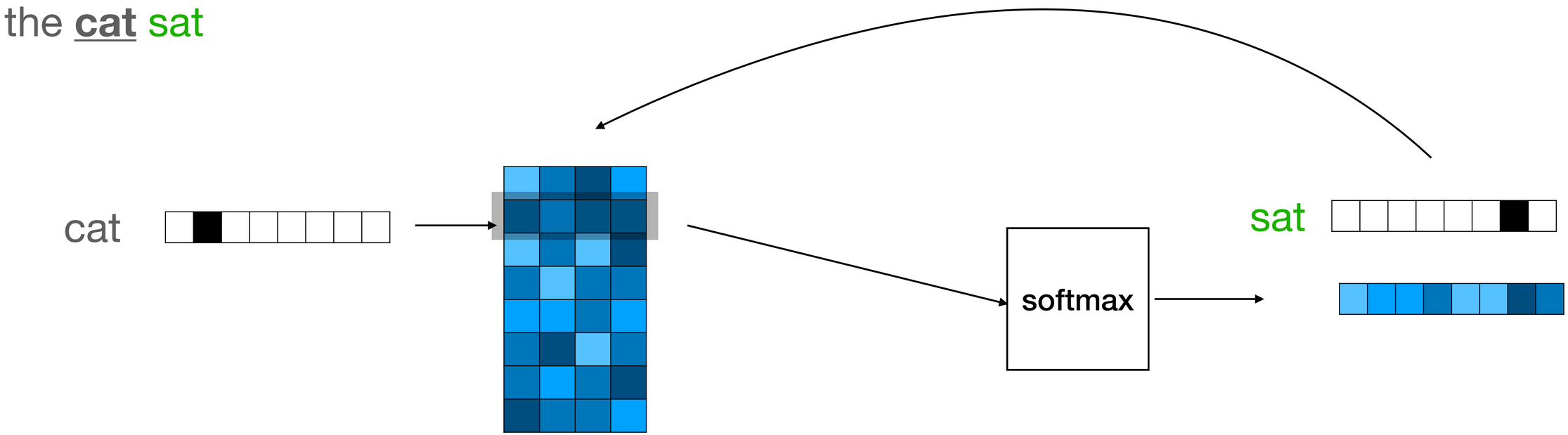
projection/ activation function

distribution over context words

(Regular/“Static”) Word Embeddings



(Regular/“Static”) Word Embeddings

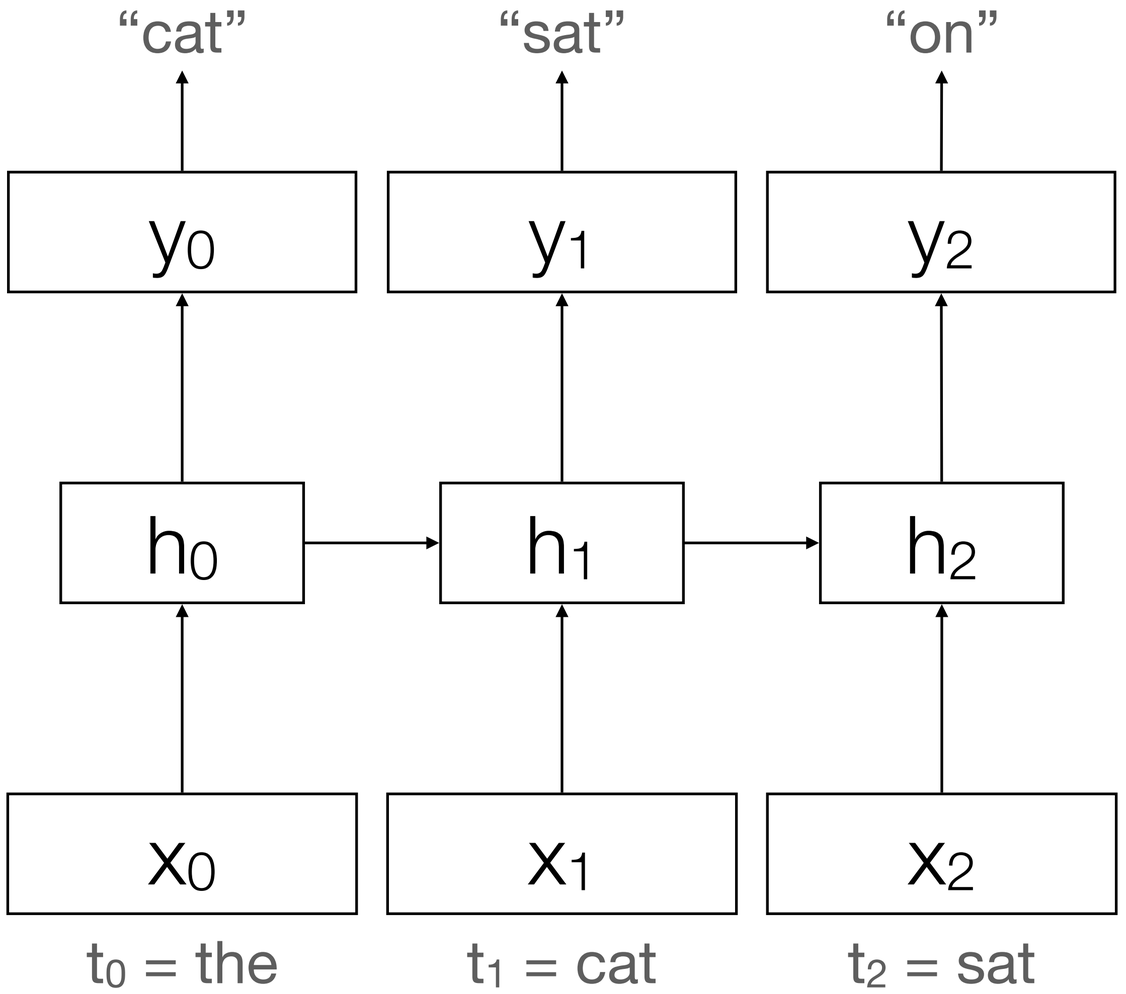


one embedding for each word (“cat”), updated each time the word is seen in training

Contextualized Word Representations

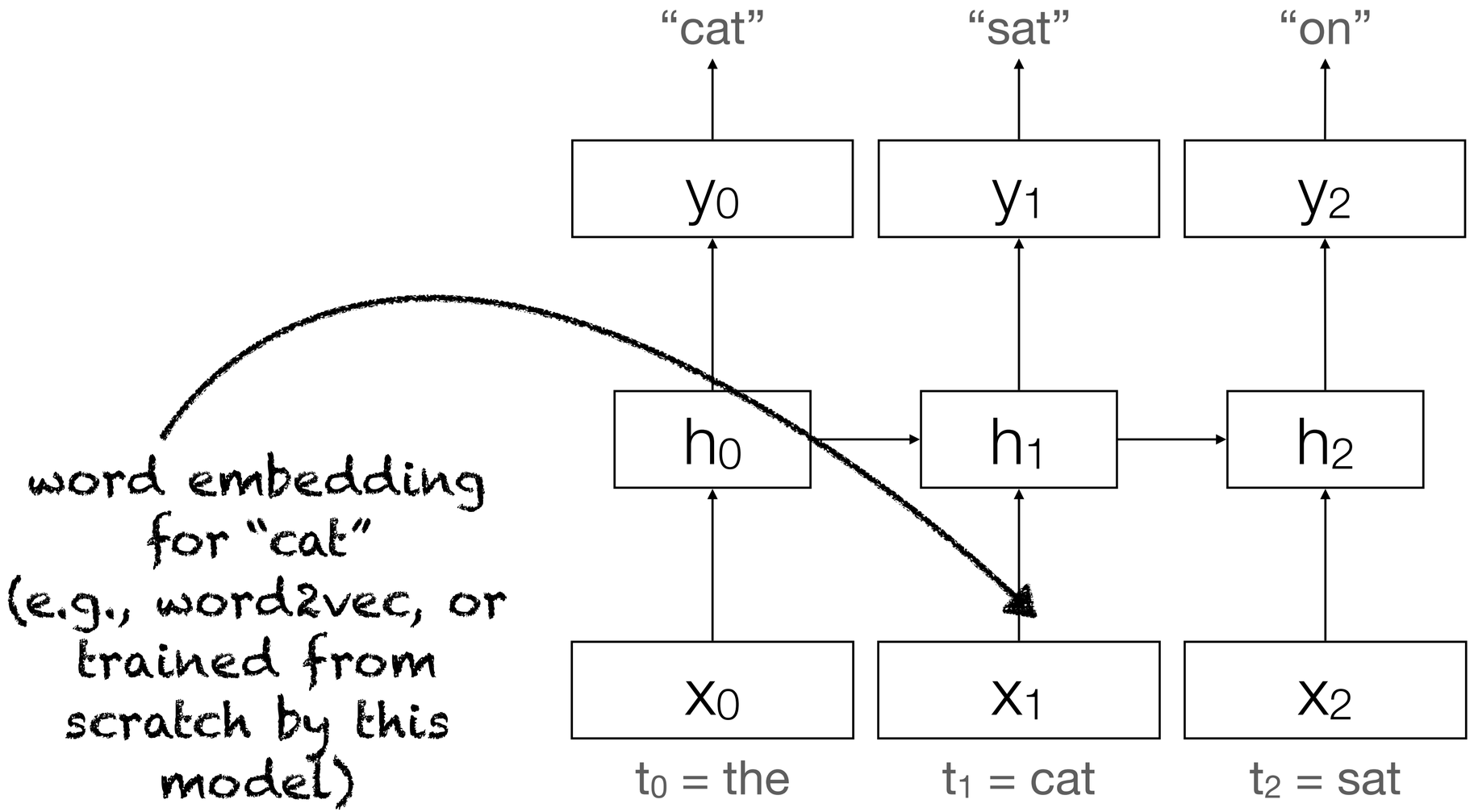
Basic Idea

Neural Language Model (e.g., an RNN)



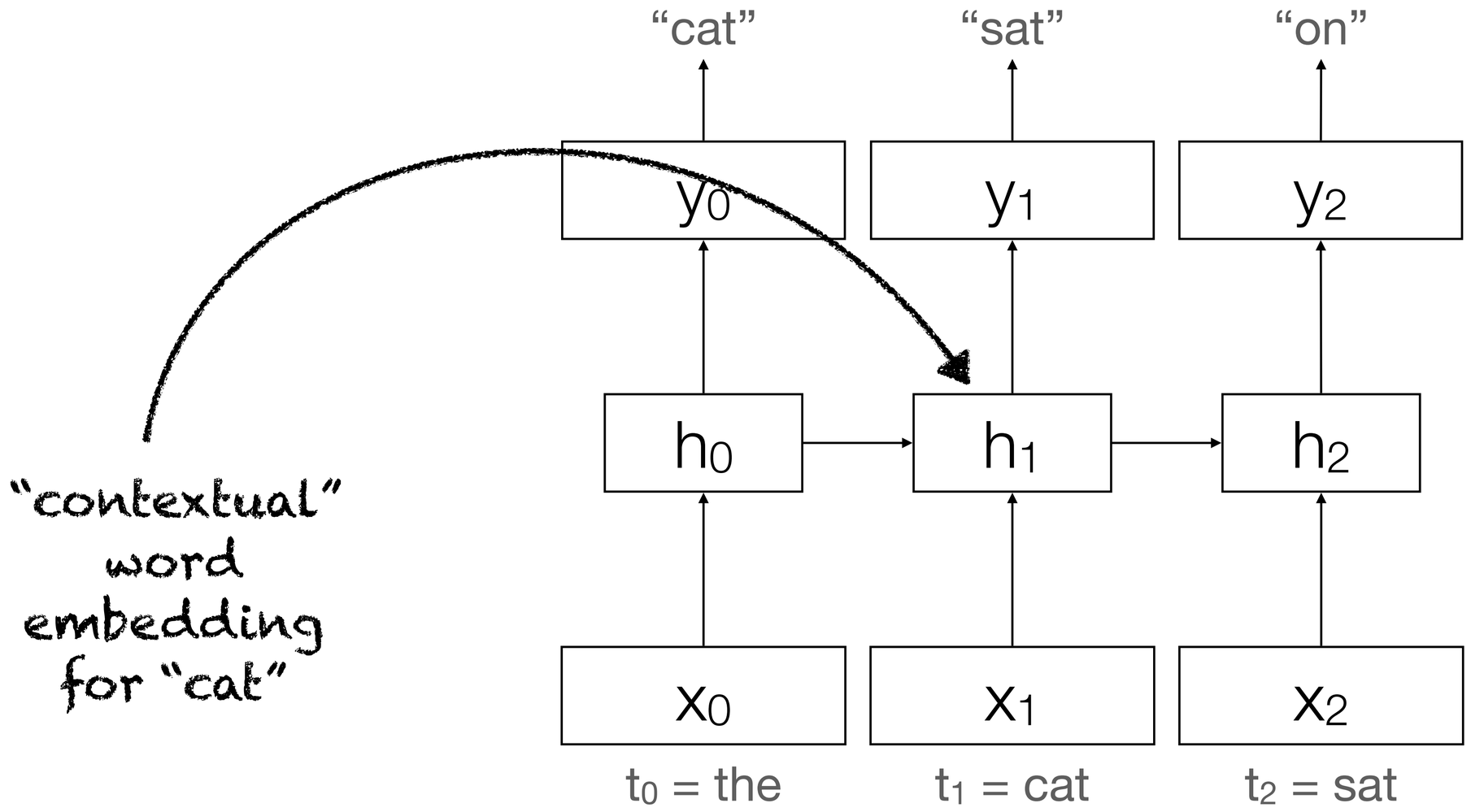
Contextualized Word Representations

Basic Idea



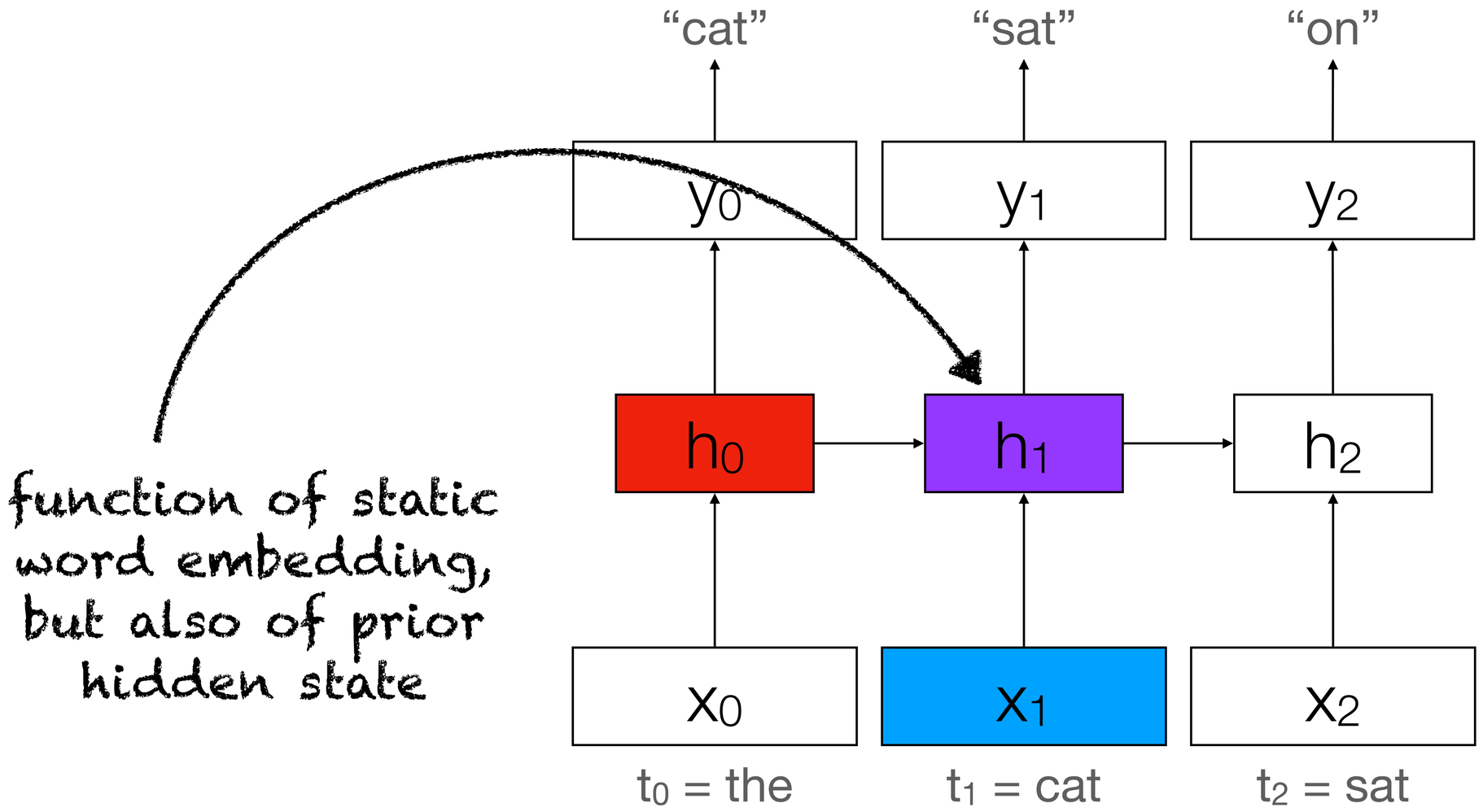
Contextualized Word Representations

Basic Idea



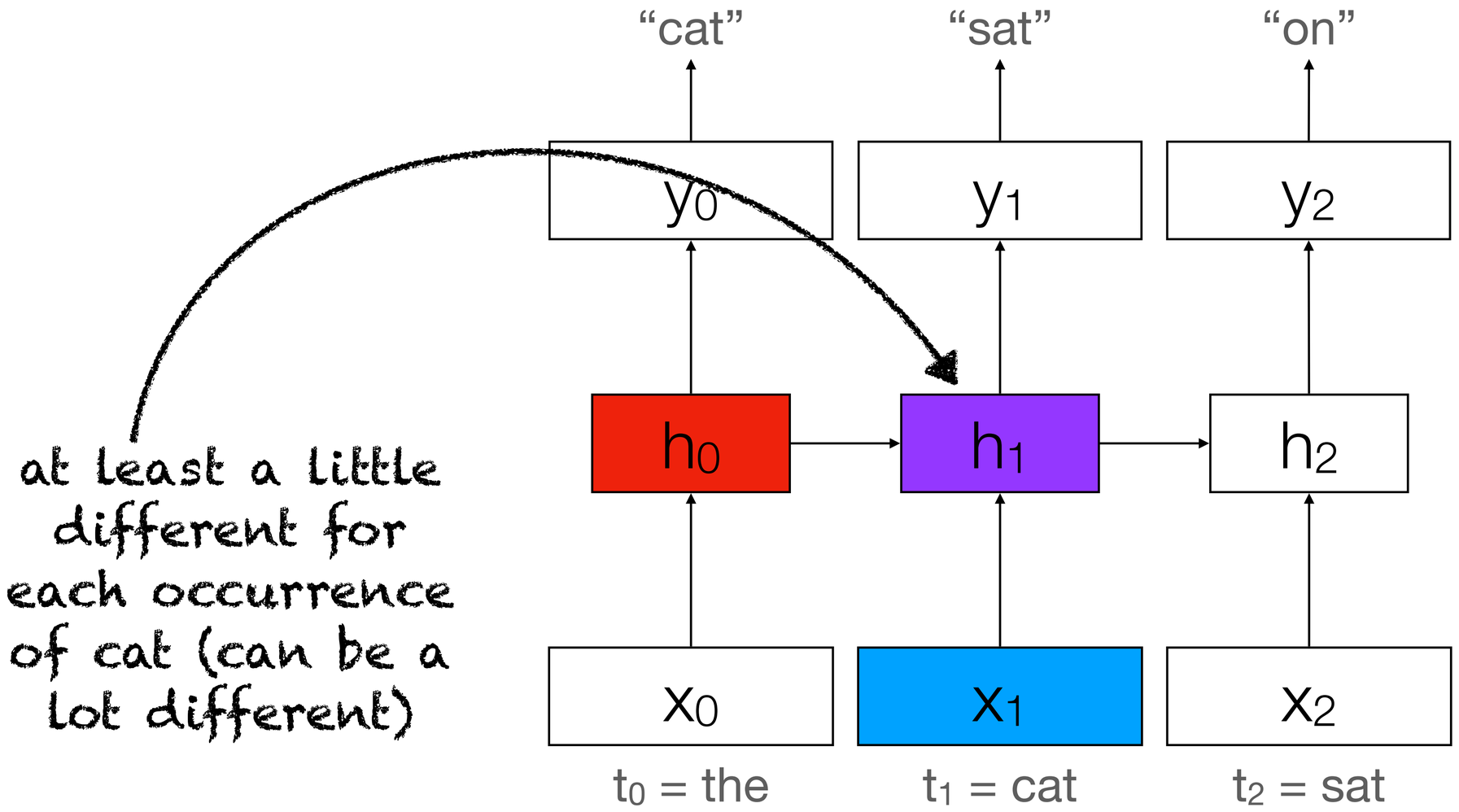
Contextualized Word Representations

Basic Idea



Contextualized Word Representations

Basic Idea



Topics

- Transformers Cont. (+ any questions you have!)
- What is pretraining?
- **Contextualized Word Representations:**
  - **ELMo**
  - BERT
  - GPT
- Finetuning and Other Transfer Methods

ELMo

Deep contextualized word representations

Matthew E. Peters[†], **Mark Neumann**[†], **Mohit Iyyer**[†], **Matt Gardner**[†],
{matthewp,markn,mohiti,mattg}@allenai.org

Christopher Clark[*], **Kenton Lee**[*], **Luke Zettlemoyer**[†*]
{csquared,kentonl,lsz}@cs.washington.edu

[†]Allen Institute for Artificial Intelligence
[*]Paul G. Allen School of Computer Science & Engineering, University of Washington

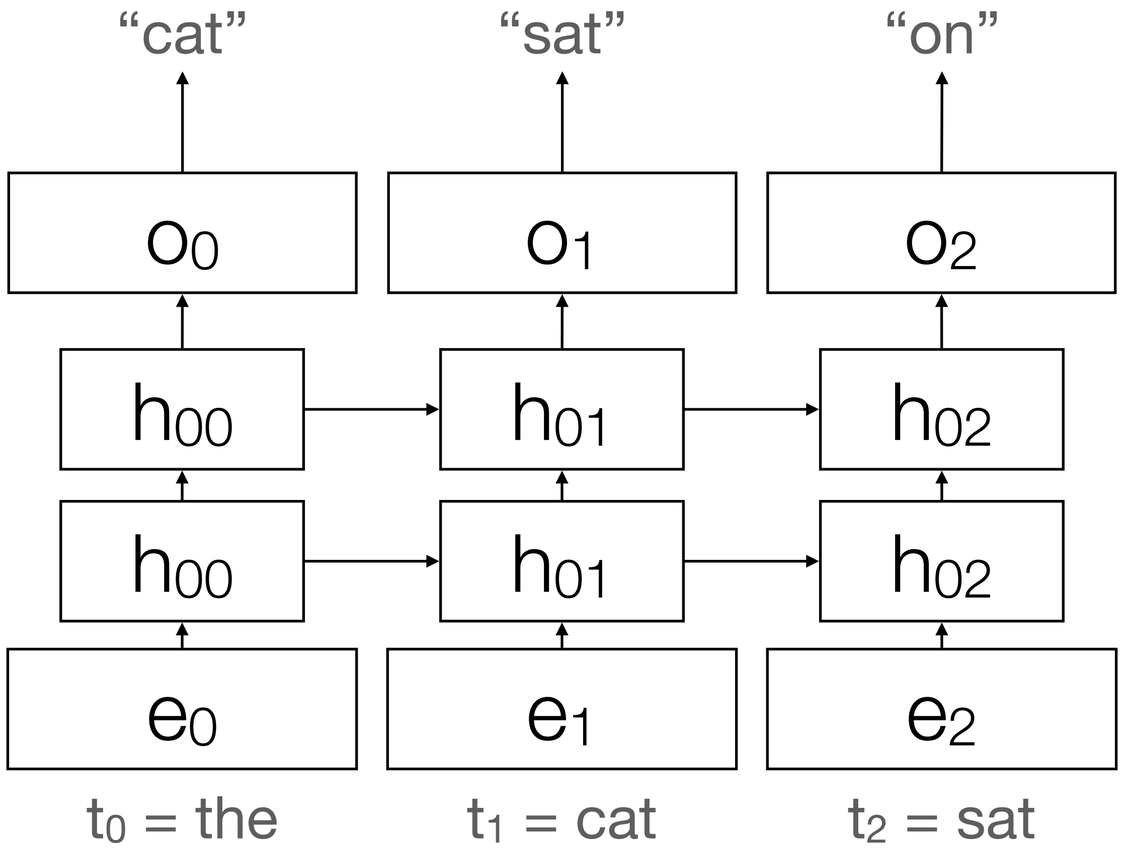
https://allenai.org/allennlp/software/elmo

ELMo

- Architecture: Two-layer BiLSTM
  - “bi” = bidirectional, i.e., trained left-to-right and also right-to-left
- Layer 0 = pretrained static embeddings (Glove)
- Trained on vanilla language modeling task
- “Finetuned” by learning a simple linear combination of the learned embeddings

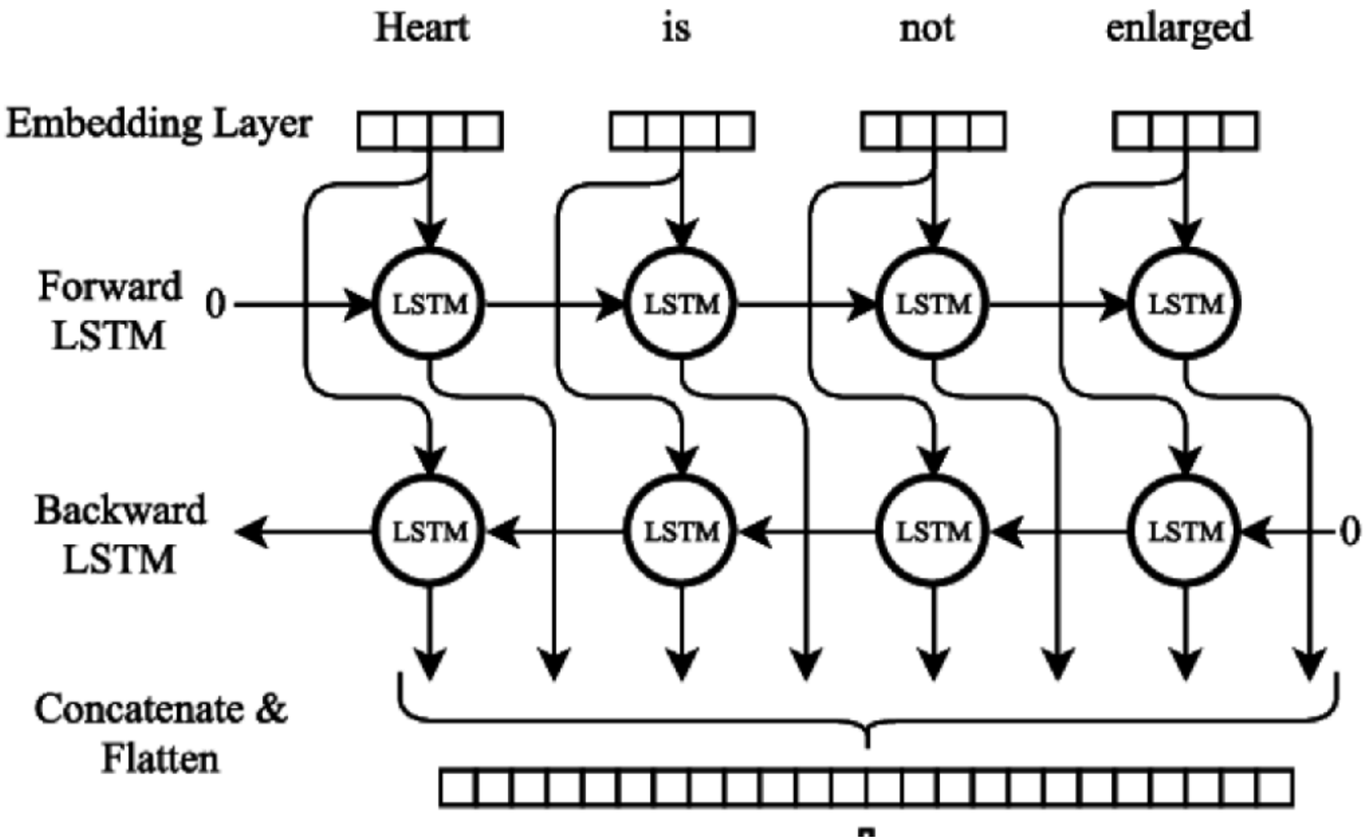
ELMo

Architecture



ELMo

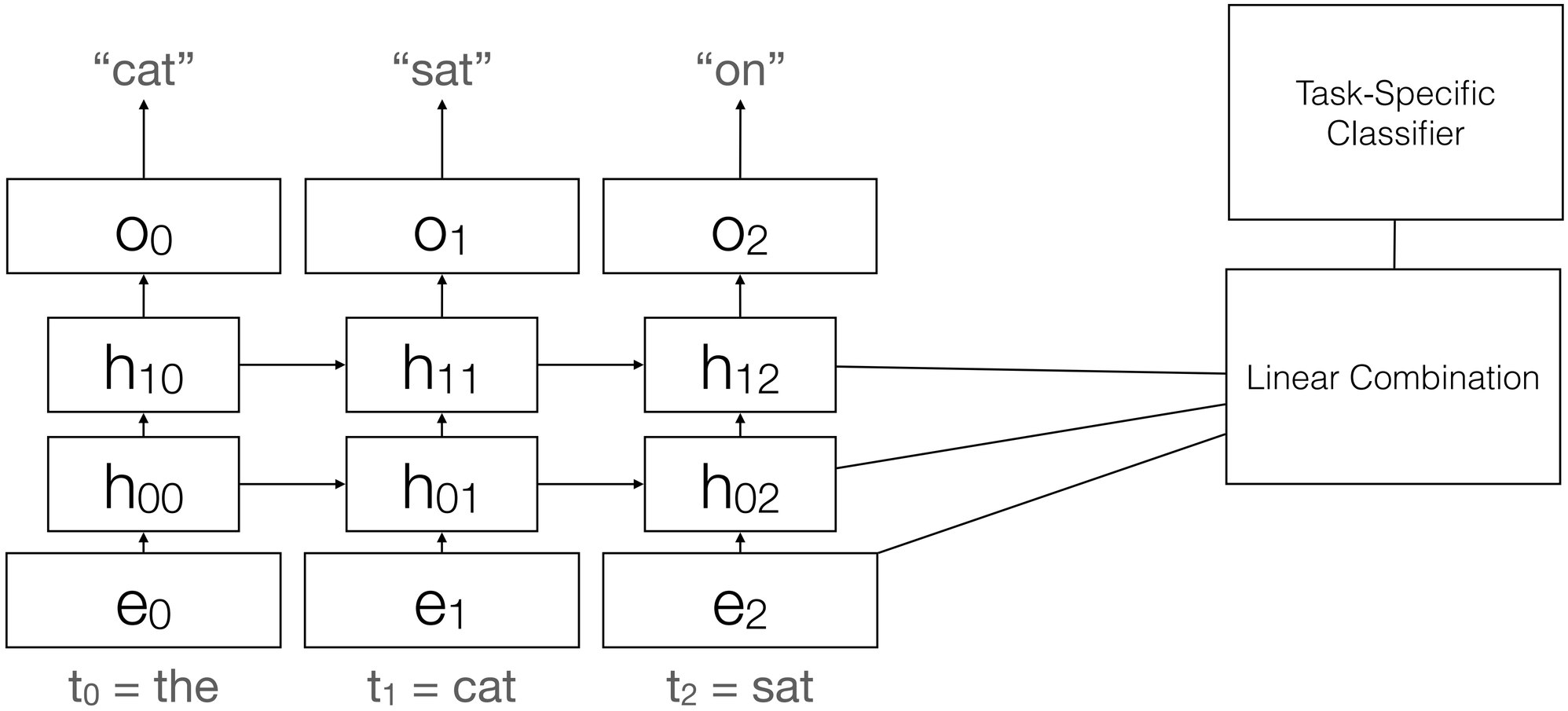
Architecture



https://paperswithcode.com/method/bilstm

ELMo

Transfer



ELMo

Evaluation

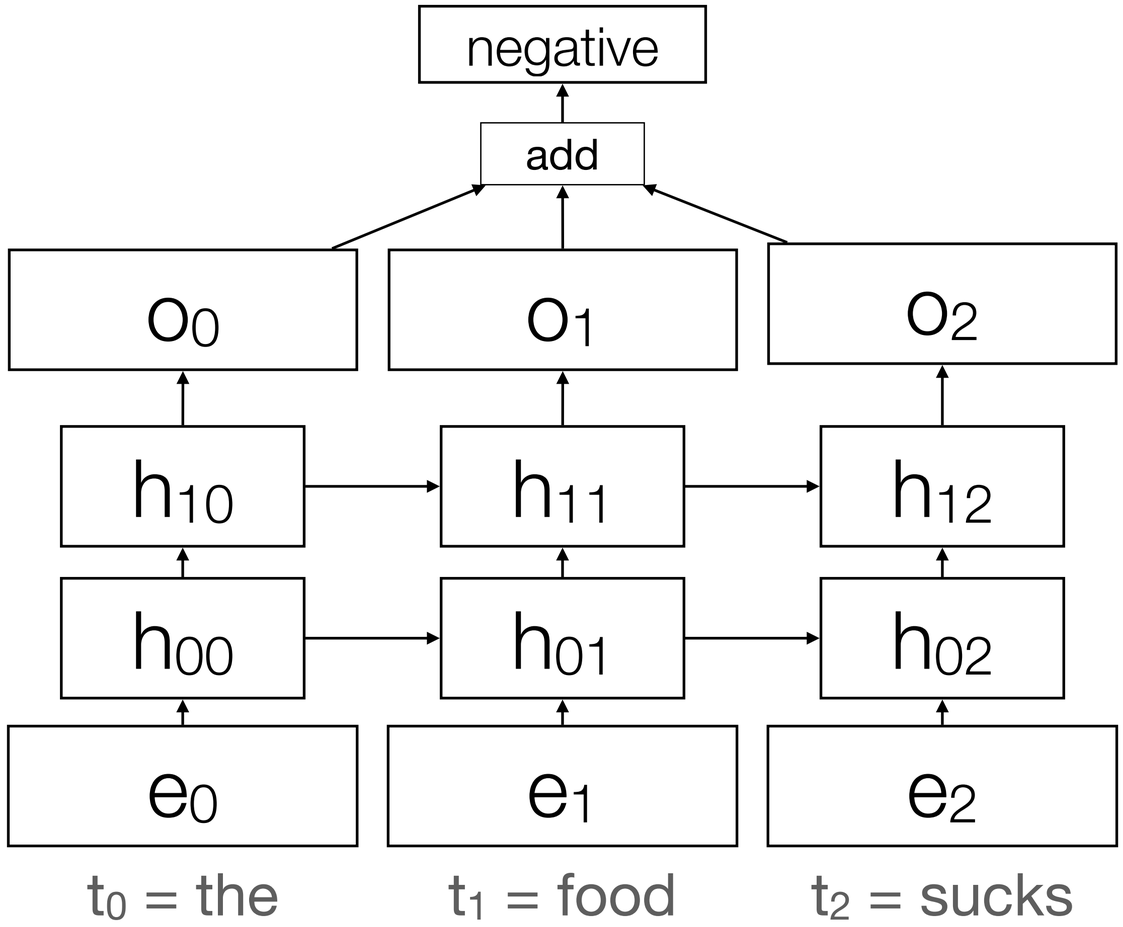
Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 $\pm$ 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 $\pm$ 0.19	90.15	92.22 $\pm$ 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 $\pm$ 0.5	3.3 / 6.8%

ELMo

Evaluation

"Our Baseline"

Each task has a custom neural network model as a starting point



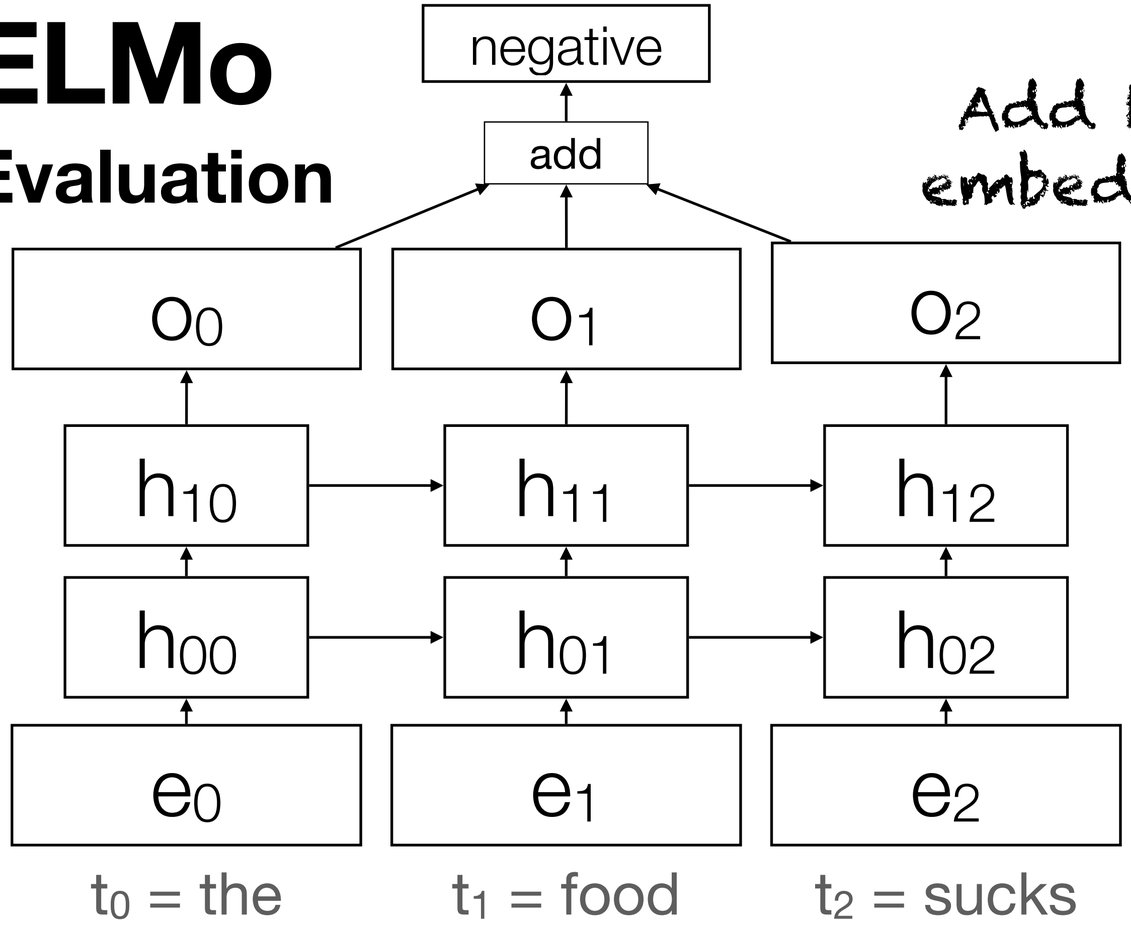
E.g., sentiment analysis model might be an RNN

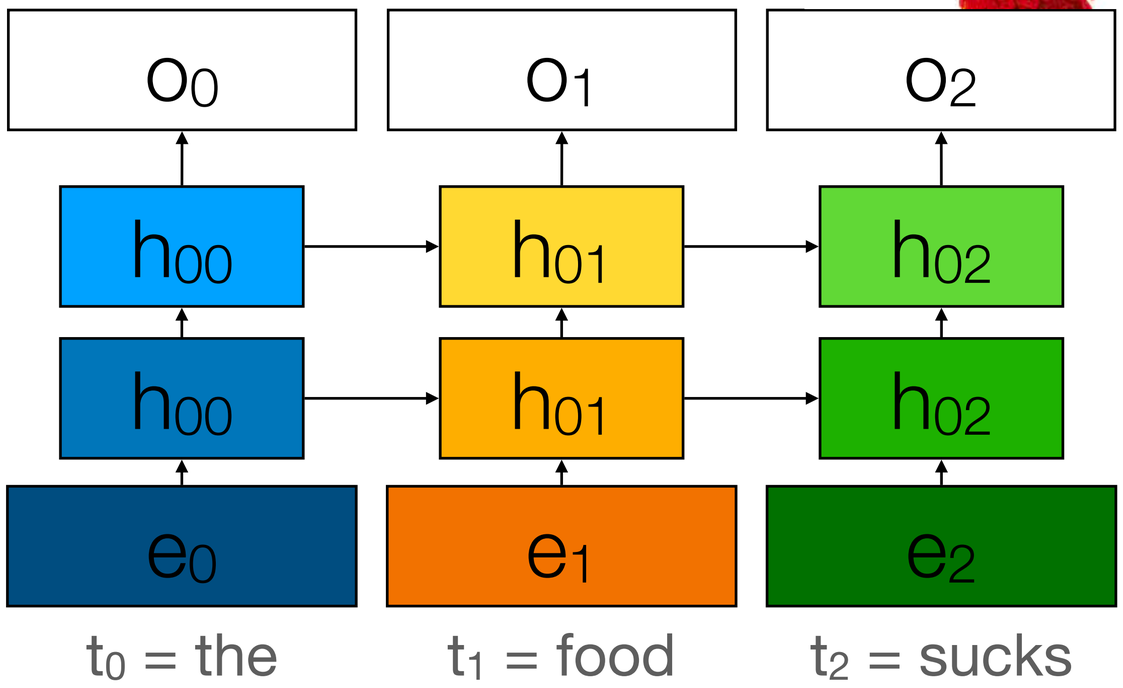
ELMo

Evaluation

"Baseline+ELMo"

Add linear combo of ELMo embeddings as starting point



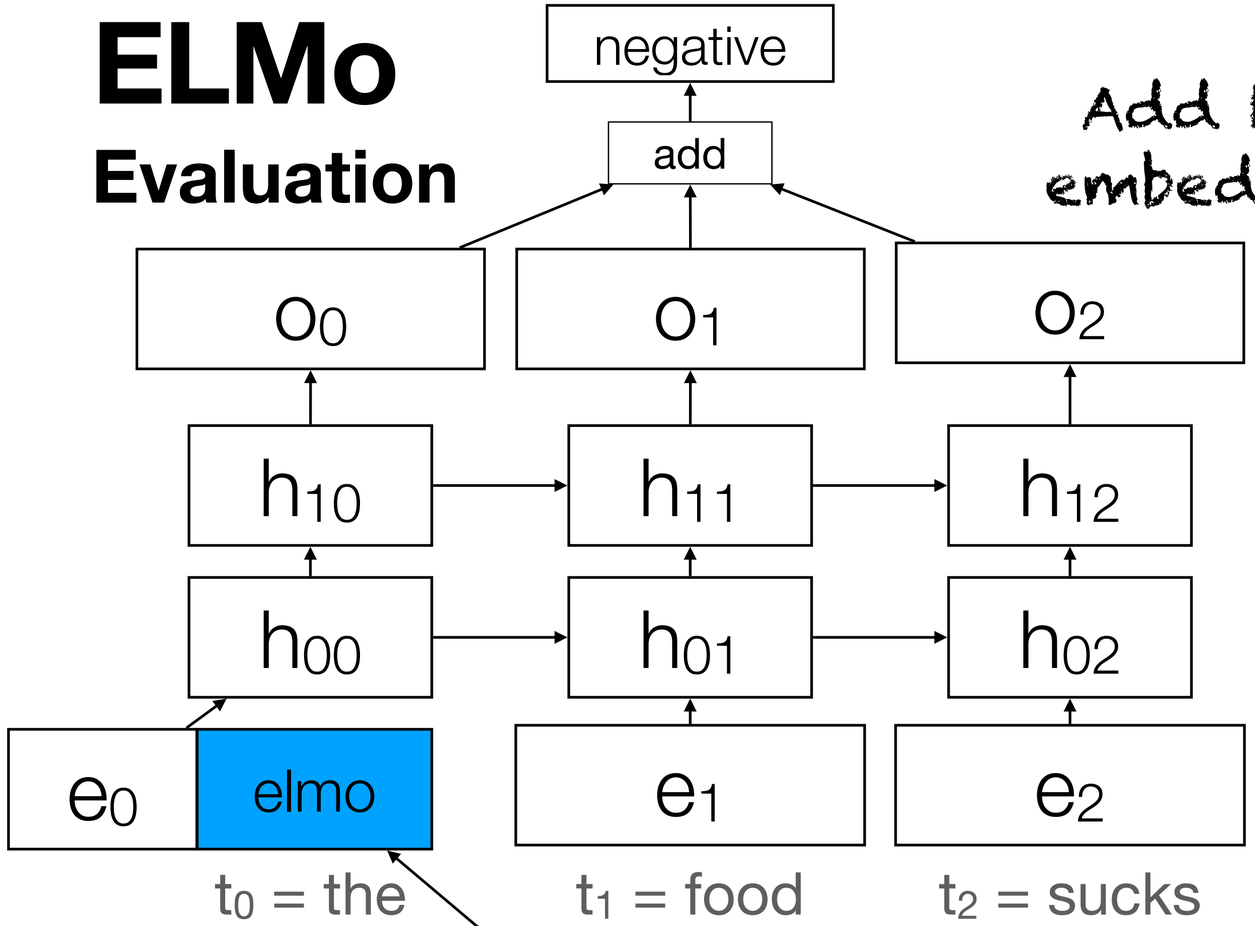


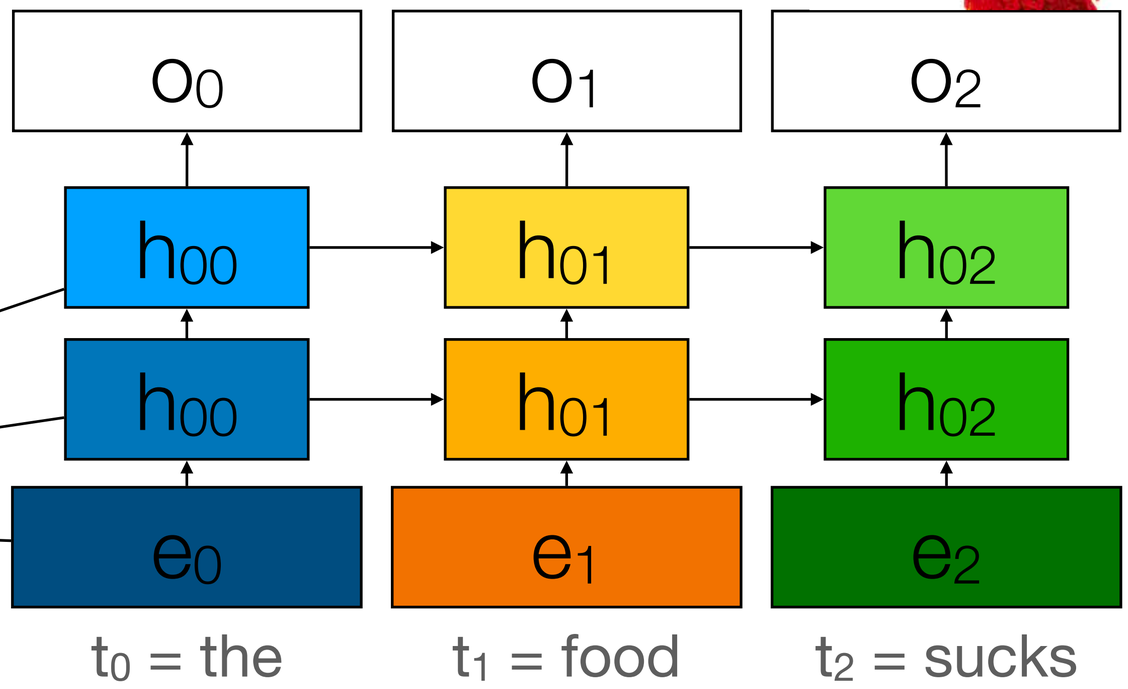
ELMo

Evaluation

"Baseline+ELMo"

Add linear combo of ELMo embeddings as starting point





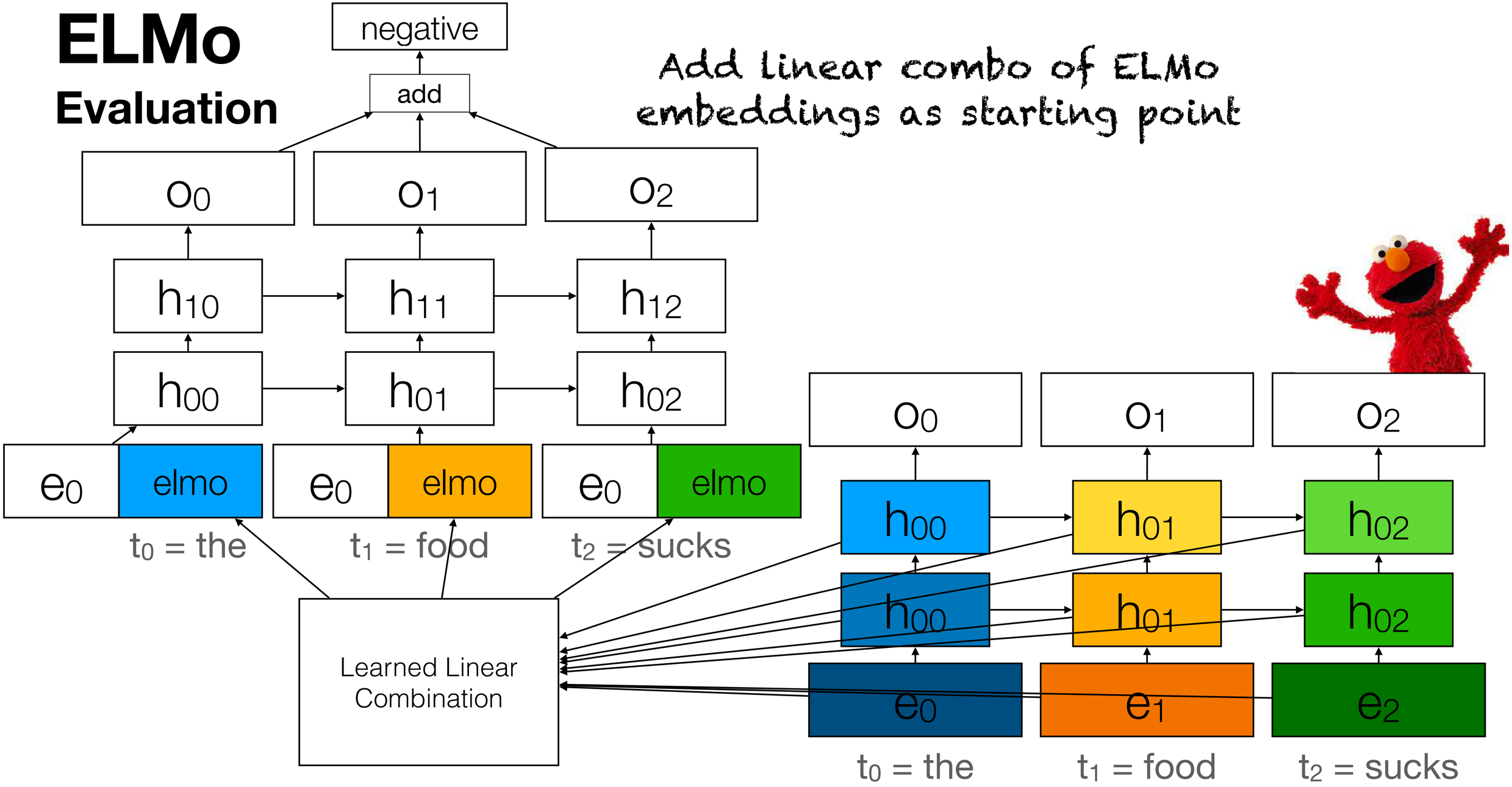
Learned Linear Combination

ELMo

Evaluation

"Baseline+ELMo"

Add linear combo of ELMo embeddings as starting point



ELMo

Evaluation

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

Table 4: Nearest neighbors to "play" using GloVe and the context embeddings from a biLM.

Topics

- **Contextualized Word Representations:**
  - ELMo
  - **BERT**
  - GPT
- Finetuning and Other Transfer Methods

BERT

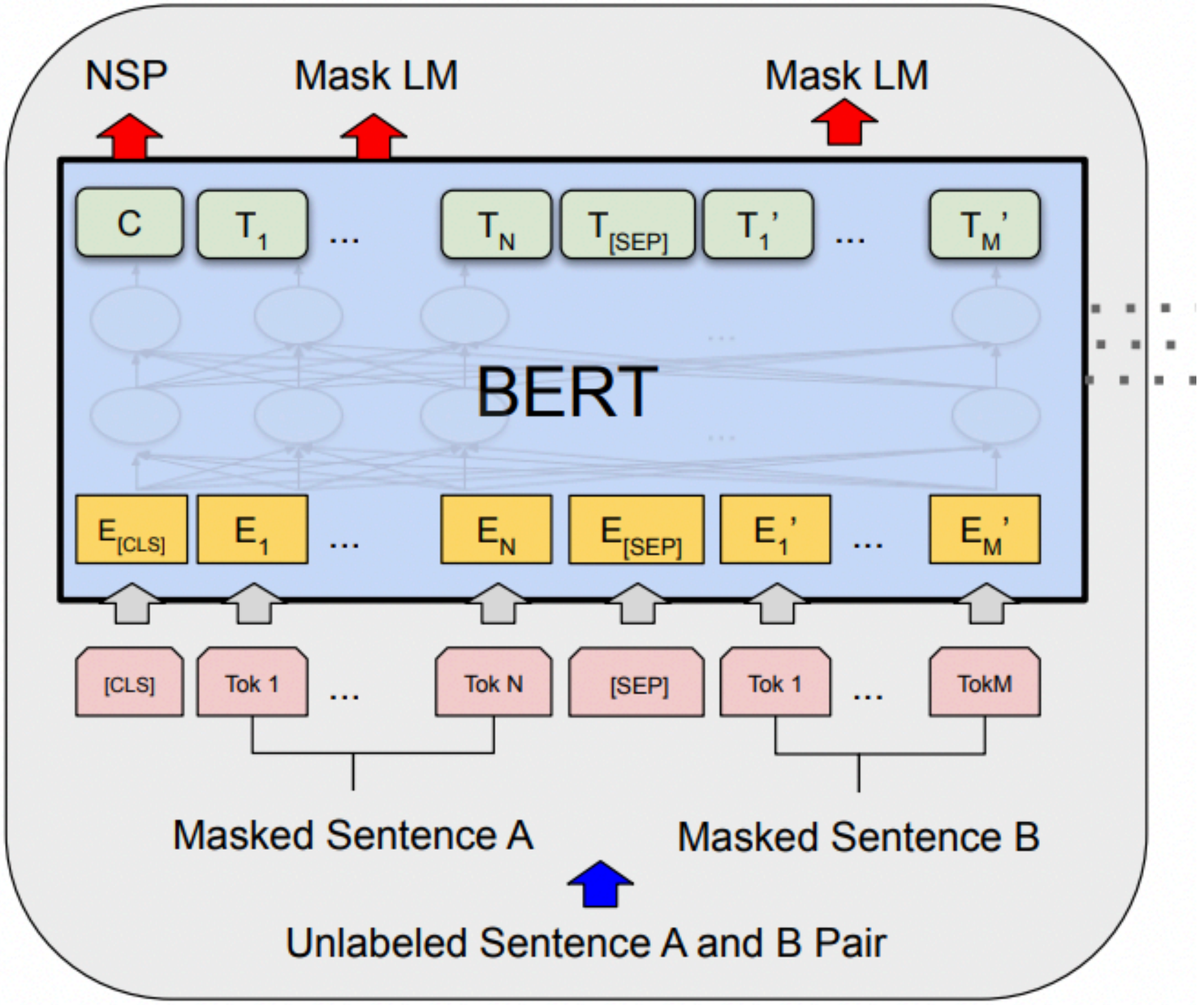
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

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

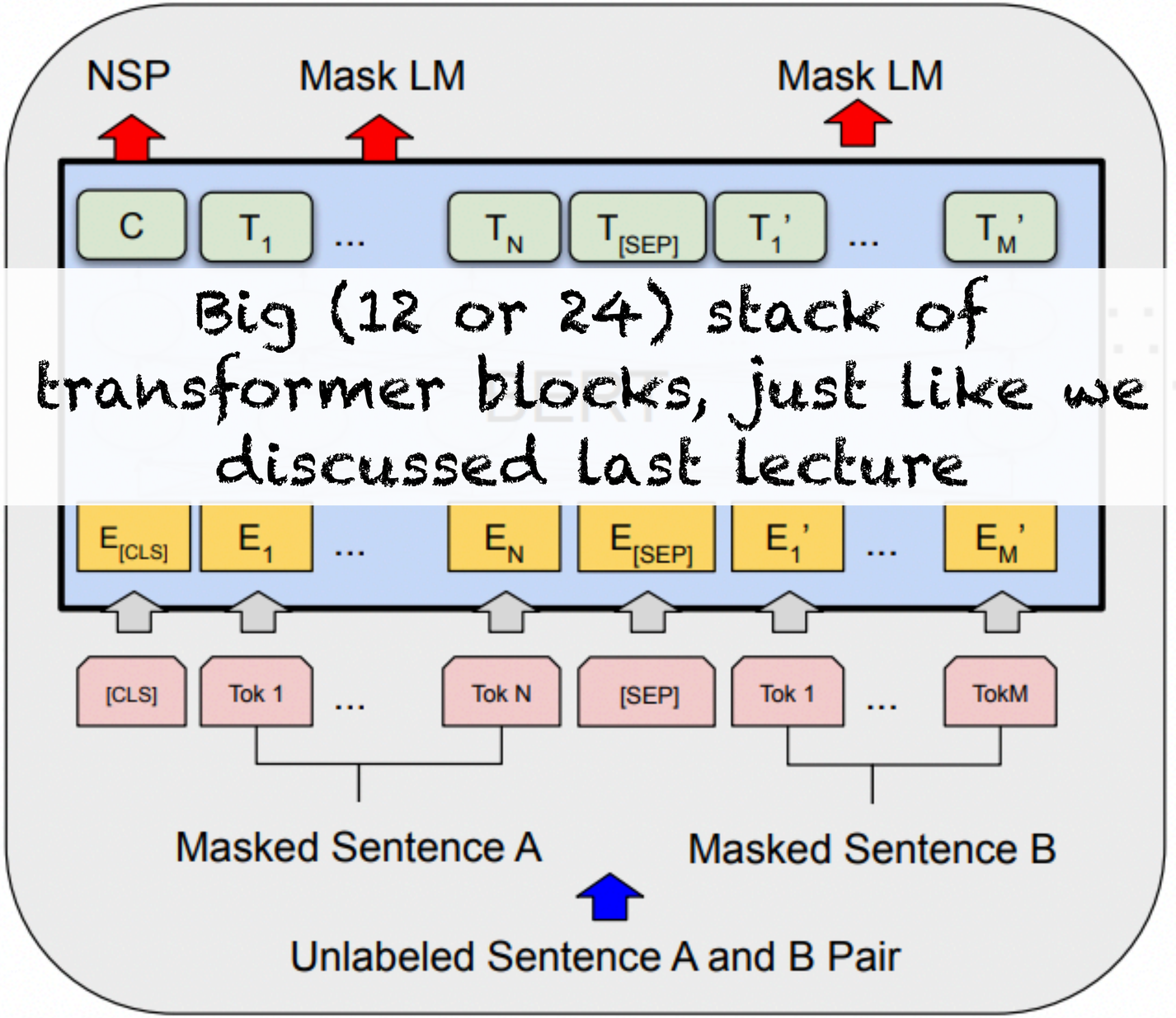
BERT

- Architecture: Deep Transformer (small = 12, large = 24)
- Layer 0 = wordpiece embeddings
- Trained on masked language modeling + next-sentence prediction
- Typically finetuned by updating all parameters (though there are other strategies)

BERT


NSP
Mask LM
Mask LM
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair

BERT


NSP
Mask LM
Mask LM
C
T_1
...
T_N
$T_{[SEP]}$
T_1'
...
T_M'
Big (12 or 24) stack of transformer blocks, just like we discussed last lecture
$E_{[CLS]}$
E_1
...
E_N
$E_{[SEP]}$
E_1'
...
E_M'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair

BERT

- ~~Architecture: Deep Transformer (small = 12, large = 24)~~
- Layer 0 = wordpiece embeddings
- **Trained on masked language modeling + next-sentence prediction**
- Typically finetuned by updating all parameters (though there are other strategies)

BERT

Masked Language Modeling

- Traditional language modeling:
  - The cat sat on ???
  - The cat sat on the ???
  - etc.
- Masked Language Modeling (MLM)
  - The cat sat on the [MASK]
  - The [MASK] sat on the mat
  - etc.

BERT

Masked Language Modeling

The cat sat on the mat.
He stretched and yawned and went to sleep.

[CLS] The [MASK] sat on the mat [SEP] He stretched [MASK] yawned and [MASK] to sleep.

BERT

Masked Language Modeling

The cat sat on the mat.
He stretched and yawned and went to sleep.

[CLS] The [MASK] sat on the mat [SEP] He stretched [MASK] yawned and [MASK] to sleep.

BERT

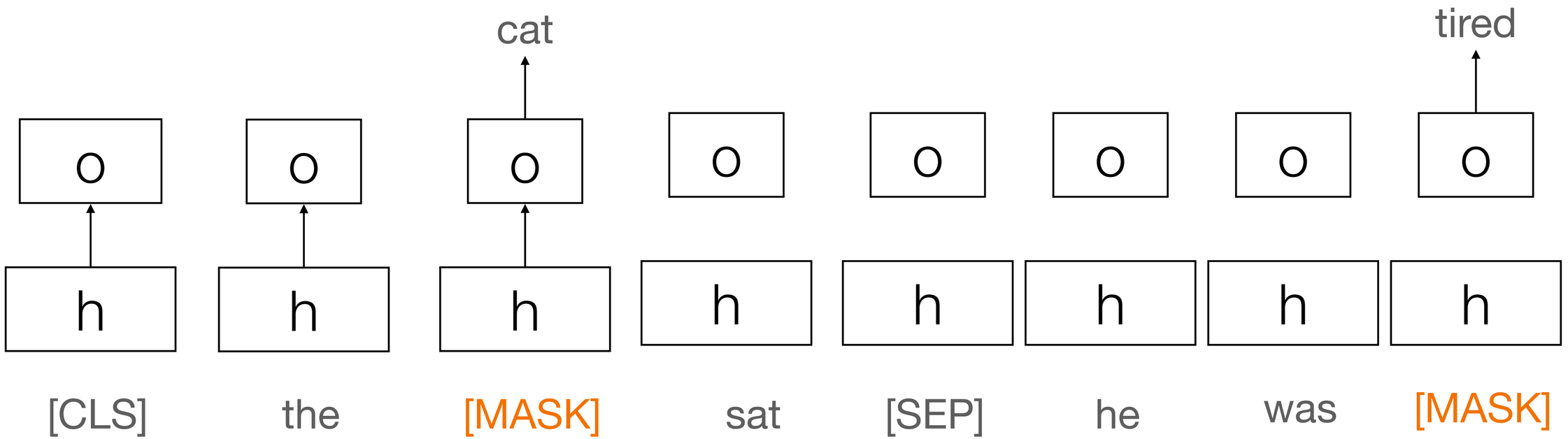
Masked Language Modeling

The cat sat on the mat.
He stretched and yawned and went to sleep.

[MASK] hides words so that they have to be predicted based on context
[SEP] = tells the model where the sentence boundary is
[CLS] = Used for the “next sentence prediction” training objective (more soon)

BERT

Masked Language Modeling



BERT

Masked Language Modeling

- Choose 15% of words to me “masked”. If a word is “masked”
  - 80% of the time, replace with the token [MASK]
  - 10% of the time, replace with a random token
  - 10% of the time, do nothing
- At training time, model has to predict the masked words
- The above weirdness is so that the model learns to encode all words, since [MASK] won’t occur at test time

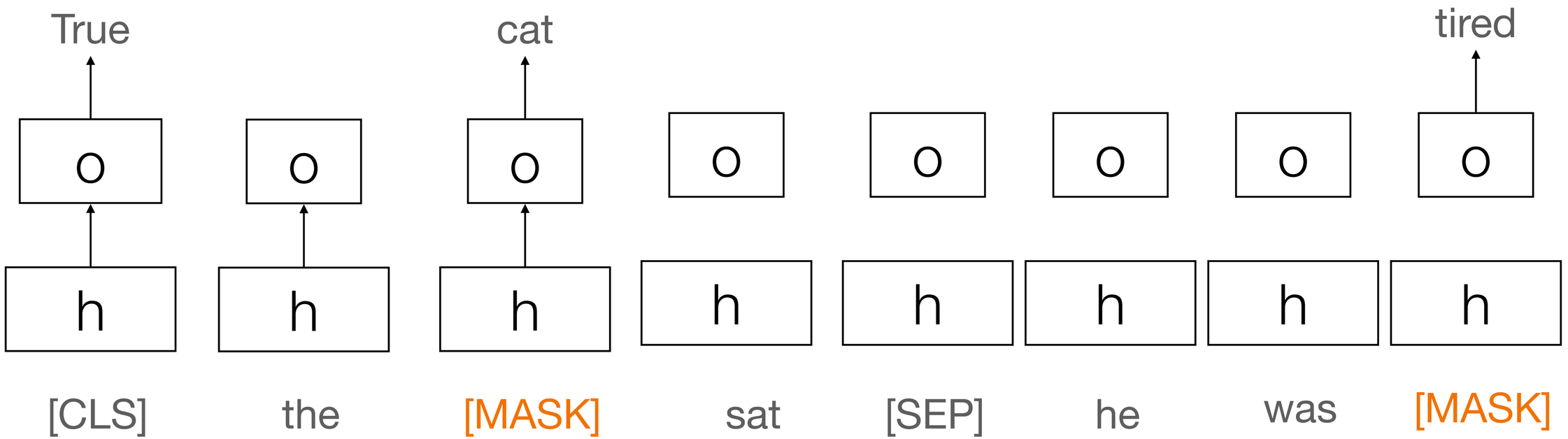
BERT

Next Sentence Prediction

- In addition to MLM, model has to predict whether or not the two sentences are consecutive
- Like SkipGram but for sentences
- Intended to encourage discourse structure
- 50% of time, sentence B actually follows sentence A in the corpus, 50% of the time, sentence B is a random sentence

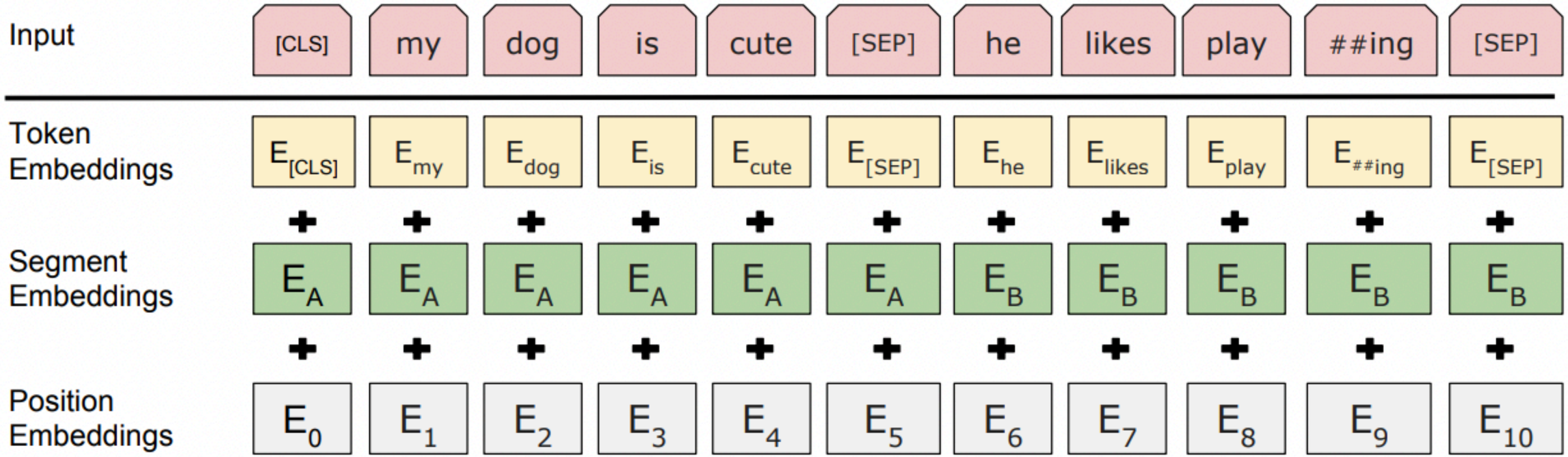
BERT

Masked Language Modeling



BERT

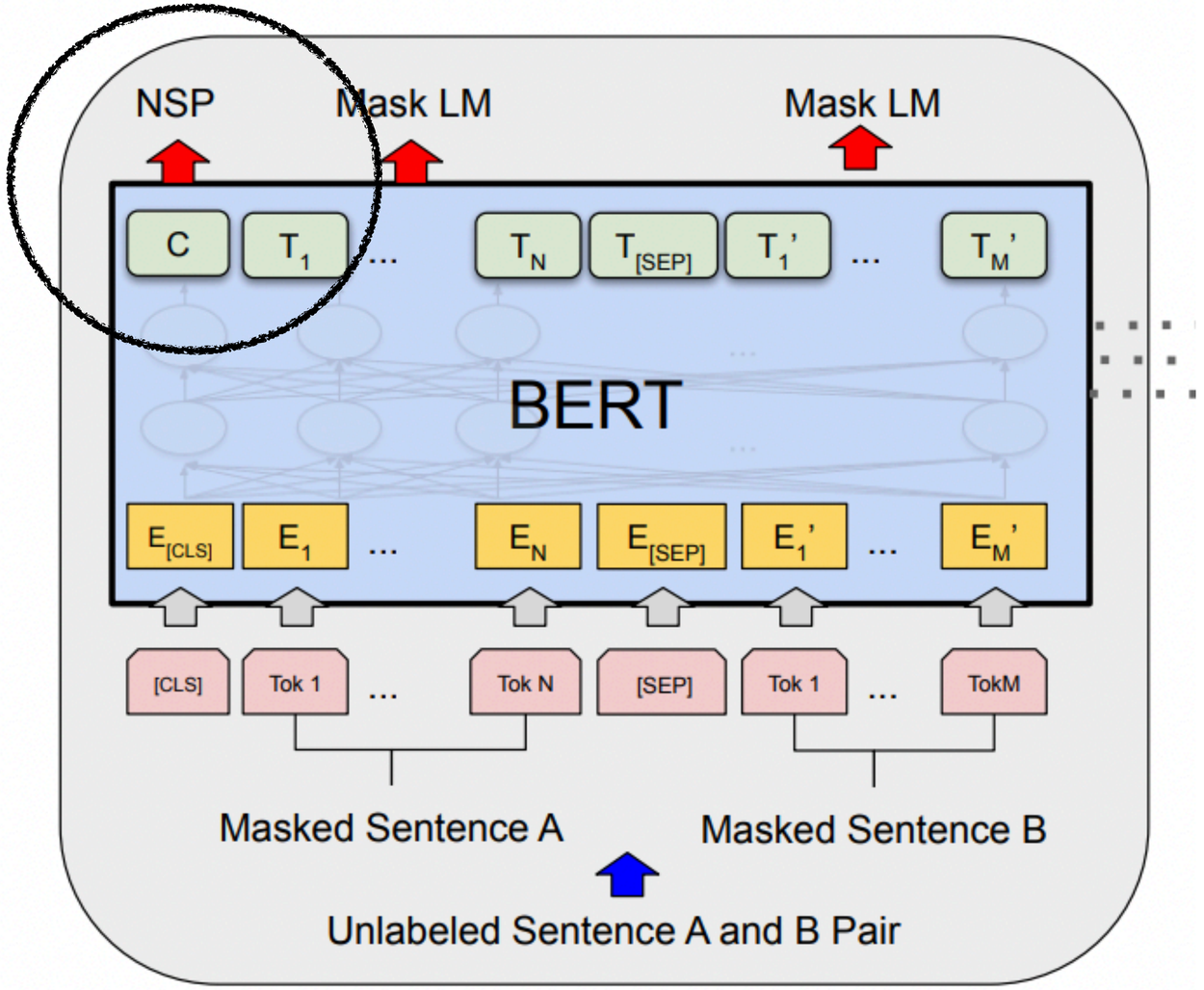
Next Sentence Prediction



BERT

Next Sentence Prediction

CLS token used for binary prediction in NSP task.

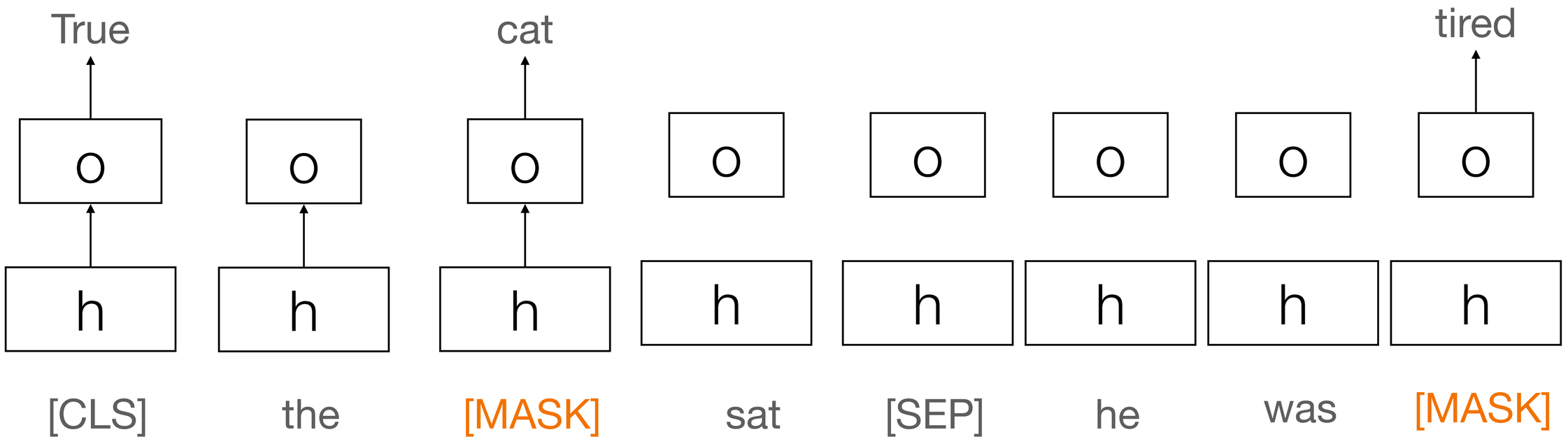


BERT

- ~~Architecture: Deep Transformer (small ≡ 12, large ≡ 24)~~
- ~~Layer 0 ≡ wordpiece embeddings~~
- ~~Trained on masked language modeling + next-sentence prediction~~
- **Typically finetuned by updating all parameters (though there are other strategies)**

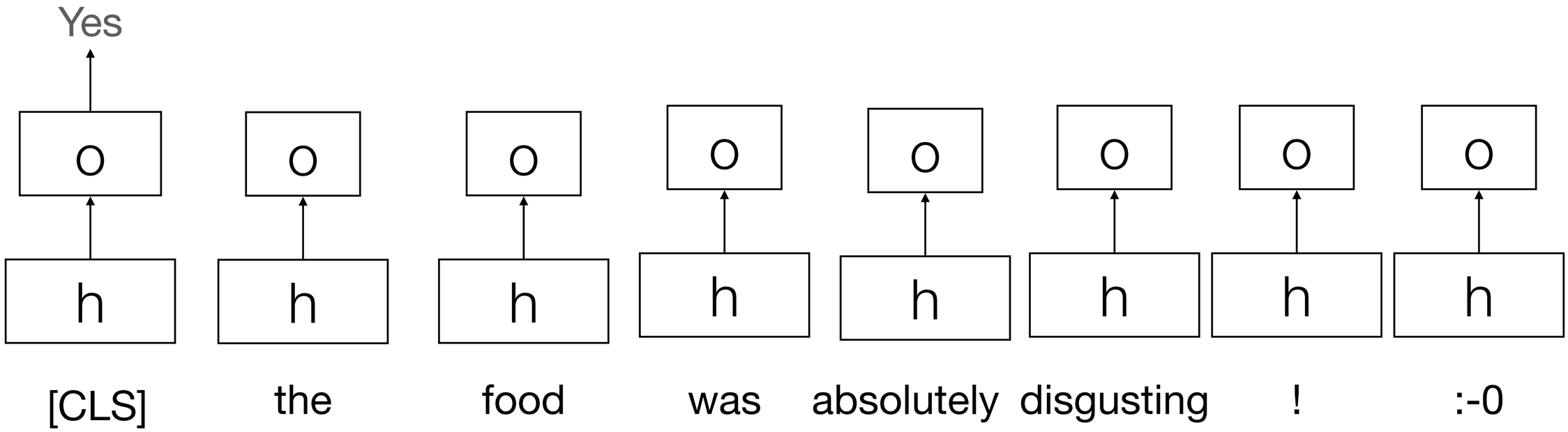
BERT

Masked Language Modeling



BERT

Finetuning on Sentiment Analysis



BERT

Finetuning

SQuAD: 100,000+ Questions for Machine Comprehension of Text

Pranav Rajpurkar and **Jian Zhang** and **Konstantin Lopyrev** and **Percy Liang**
{pranavsr,zjian,klopyrev,pliang}@cs.stanford.edu
Computer Science Department
Stanford University

Know What You Don't Know: Unanswerable Questions for SQuAD

Pranav Rajpurkar* **Robin Jia*** **Percy Liang**
Computer Science Department, Stanford University
{pranavsr,robinjia,pliang}@cs.stanford.edu

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

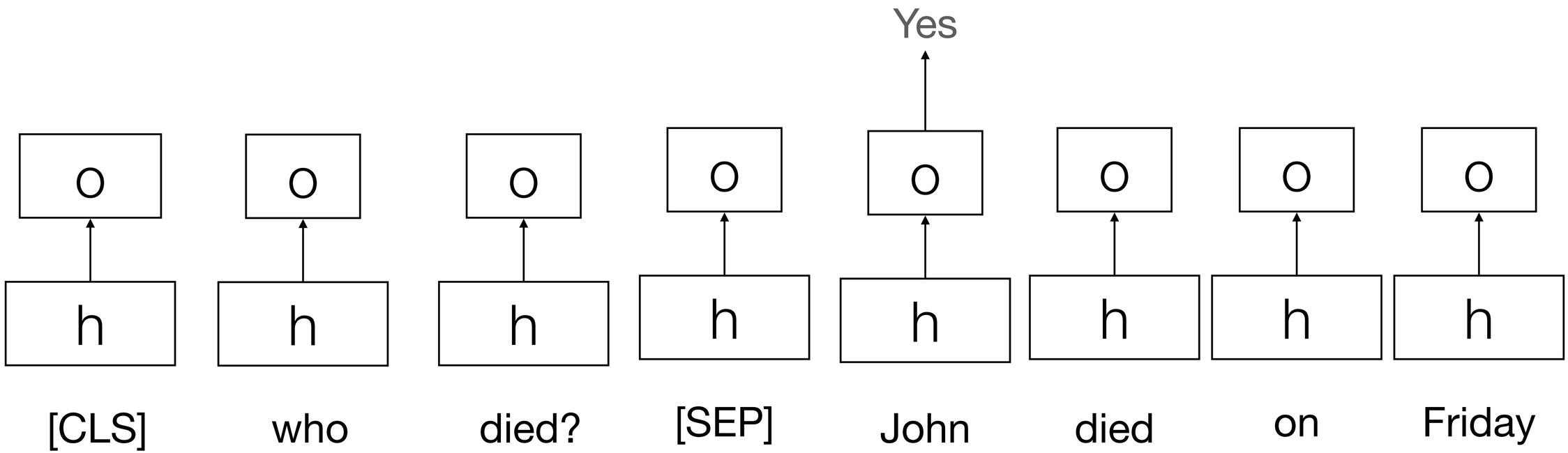
What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

BERT

Finetuning on SQUAS



BERT

Evaluation

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
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	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Topics

- Transformers Cont. (+ any questions you have!)
- What is pretraining?
- **Contextualized Word Representations:**
  - ELMo
  - BERT
  - **GPT**
- Finetuning and Other Transfer Methods

Improving Language Understanding by Generative Pre-Training

Alec Radford OpenAI alec@opena

Karthik Narasimhan OpenAI

Tim Salimans OpenAI

Ilya Sutskever OpenAI

Language Models are Unsupervised Multitask Learners

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

Language Models are Few-Shot Learners

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

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

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

GPT

- Architecture: Deep Transformer
- Layer 0 = BPE
- Trained on vanilla language modeling
- Notable because the recent versions (GPT-3) are HUGE, and very impressive
- Typically finetuned by updating all parameters (for the small models) or (for the large models) “prompting” (more soon)

GPT

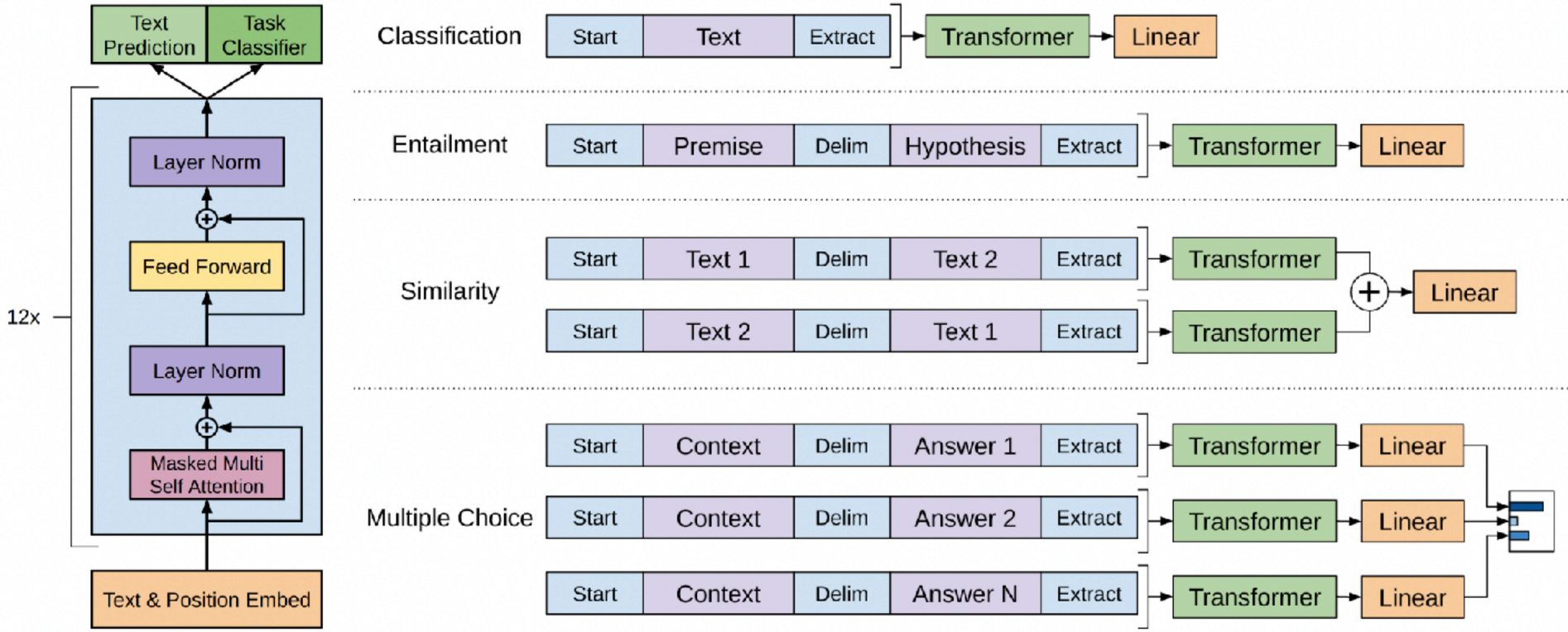


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Topics

- Transformers Cont. (+ any questions you have!)
- What is pretraining?
- **Contextualized Word Representations:**
  - ELMo
  - BERT
  - GPT
- **Finetuning and Other Transfer Methods**

Pretraining

Transferring pretrained representations

- Frozen: Just pool representations and train a new classifier on top
- Finetuning: Just treat pretraining as a good initialization. On new task, continue to update all parameters
- Parameter-Efficient or “Prompt” Tuning: Just update a small number of parameters at the bottom of the network
- Adapters: Just update a small number of parameters throughout the network
- Zero-Shot or “In-Context” Learning: Cast all tasks as an instance of the task the model was trained on (e.g., language modeling)

Pretraining

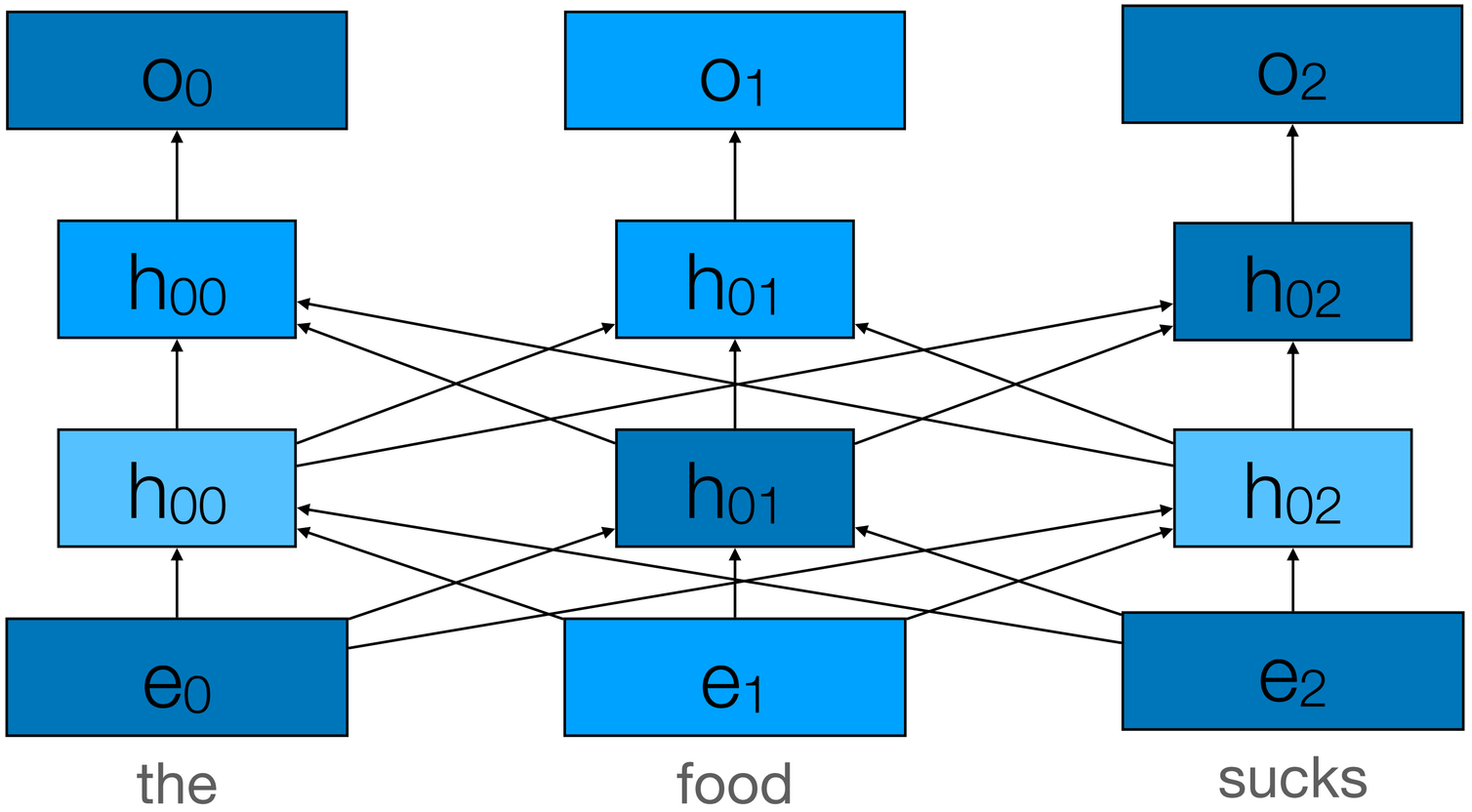
Transferring pretrained representations

- Frozen: Just pool representations and train a new classifier on top

Pretraining

Transferring frozen pretrained representations

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative

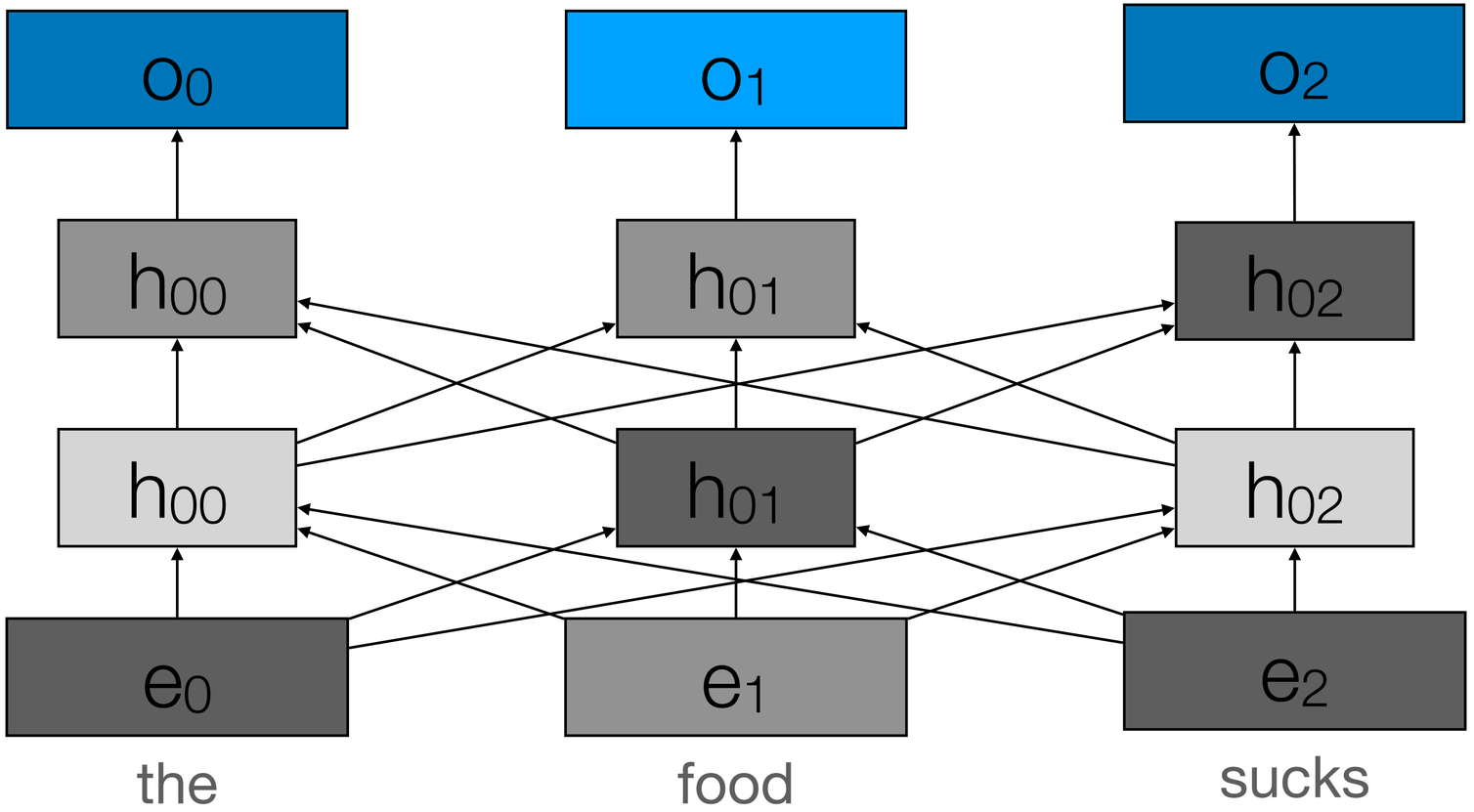


Pretraining

Transferring frozen pretrained representations

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative

Freeze the parameters (no more updating).

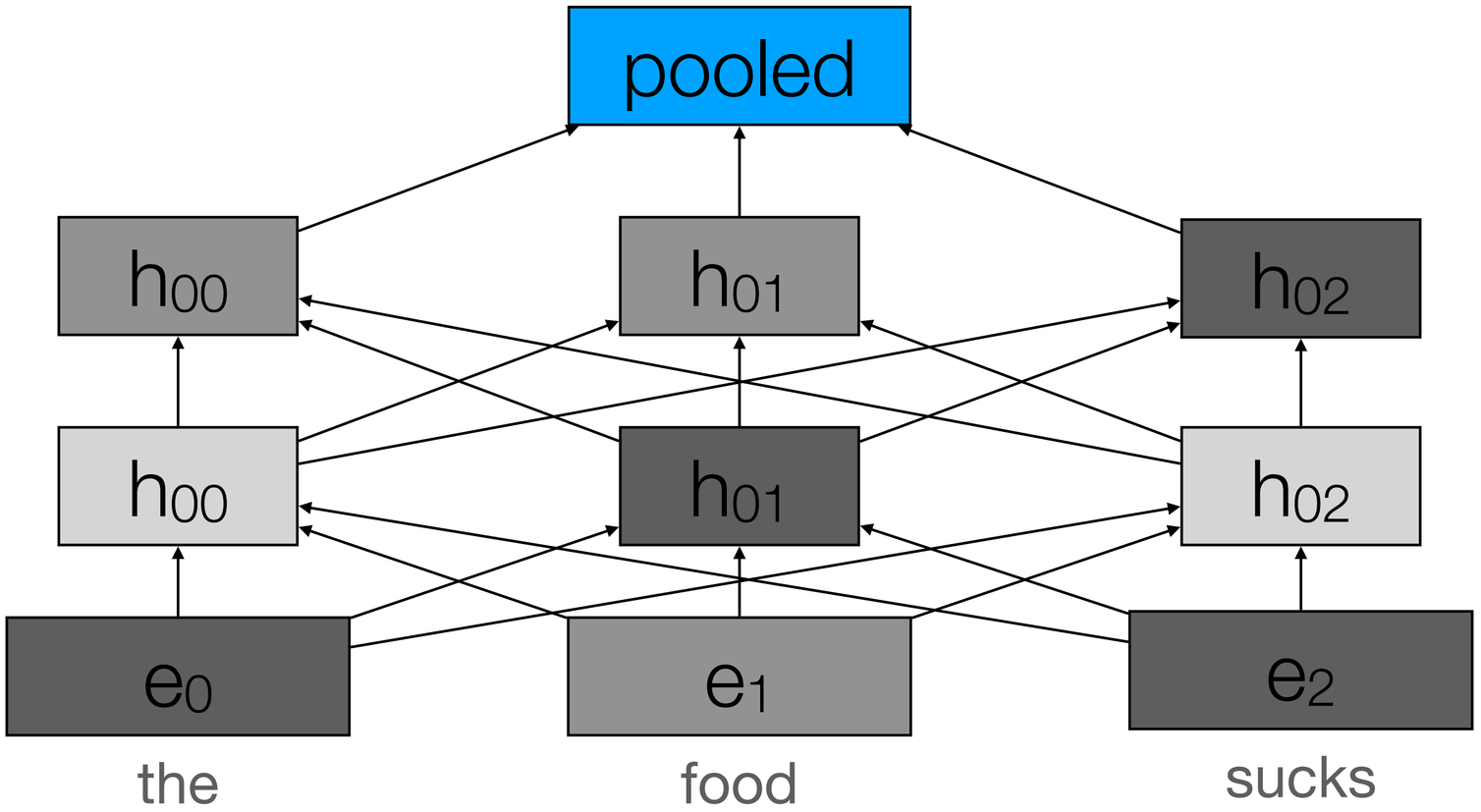


Pretraining

Transferring frozen pretrained representations

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative

Pool some representations (e.g., final layer or CLS)

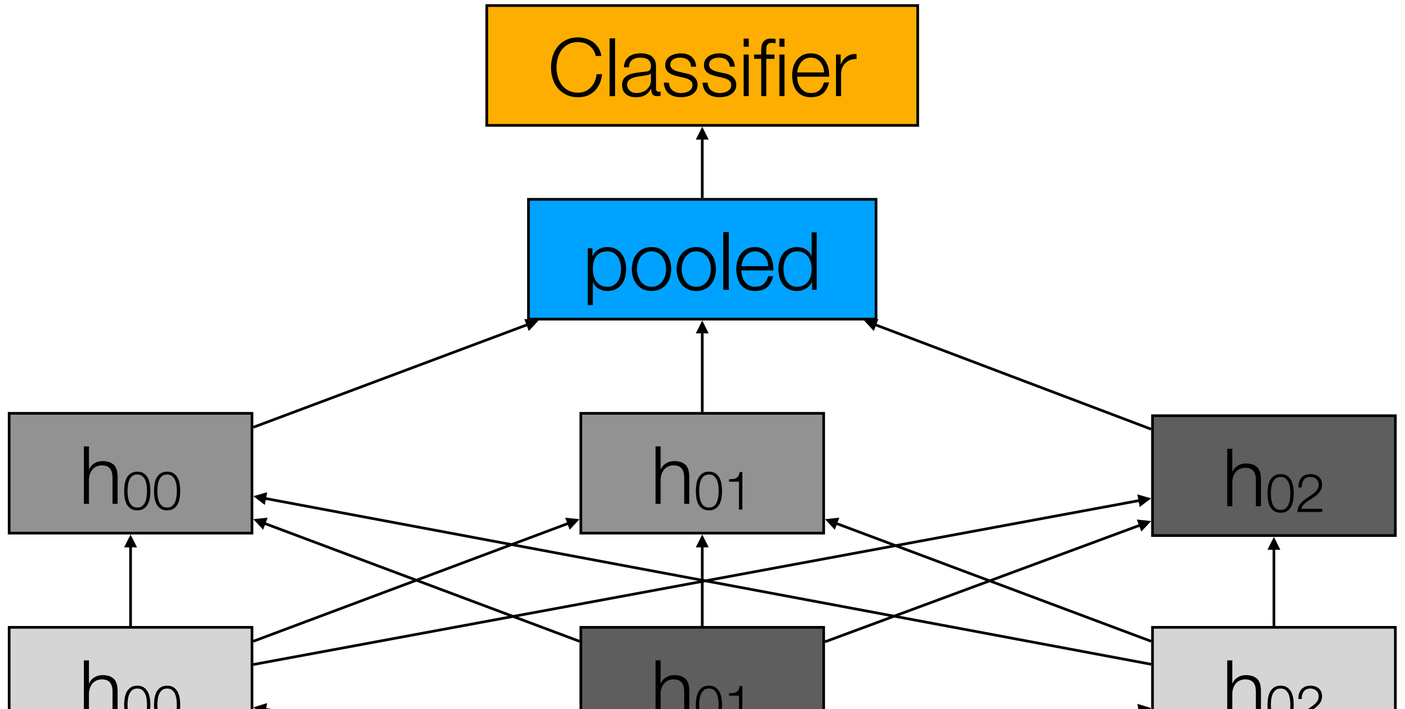


Pretraining

Transferring frozen pretrained representations

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative

Train a new network on top of the pooled representation (optionally, backprop through the pooled representation).



Pretraining

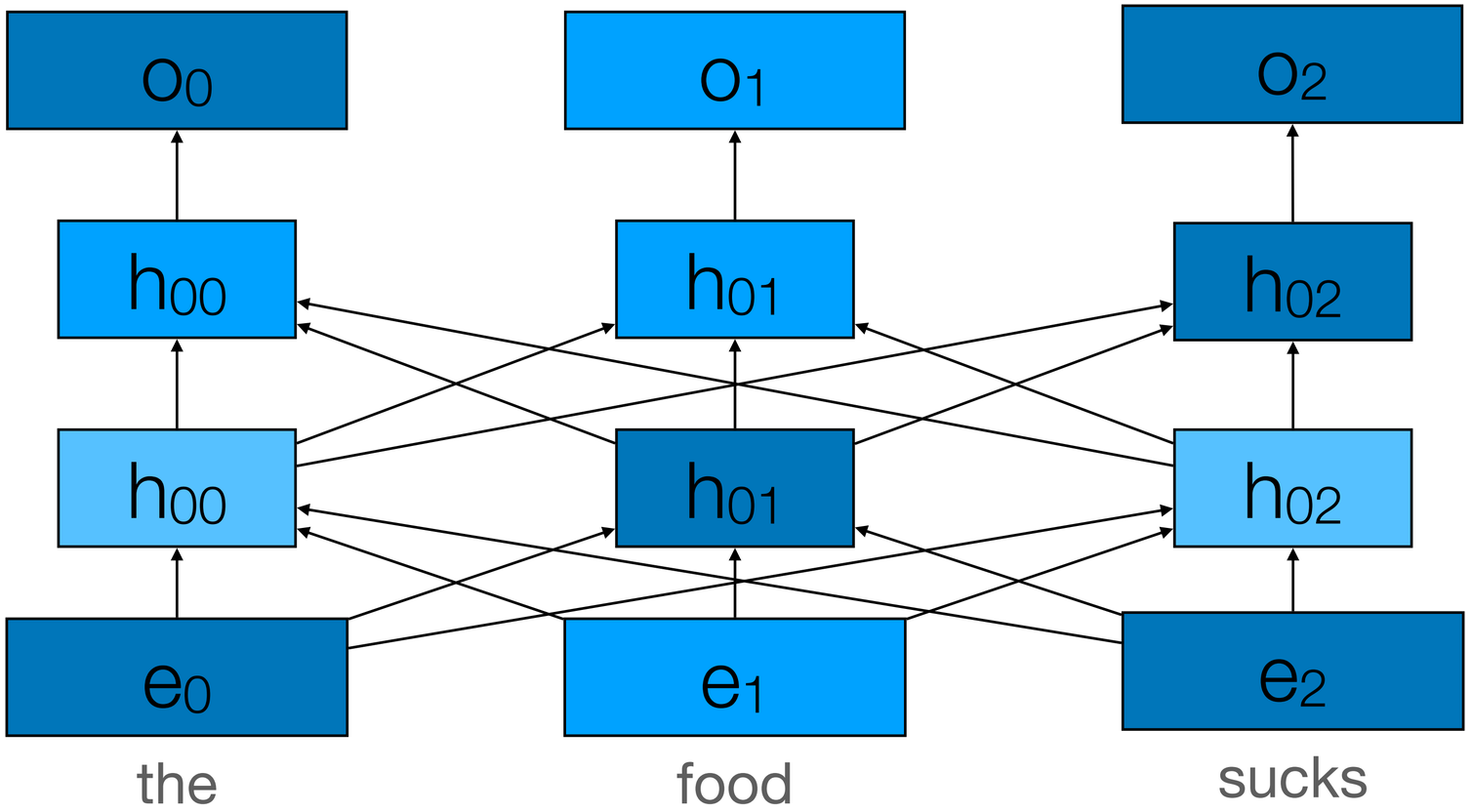
Transferring pretrained representations

- Frozen: Just pool representations and train a new classifier on top
- Finetuning: Just treat pretraining as a good initialization. On new task, continue to update all parameters

Pretraining

Finetuning

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative

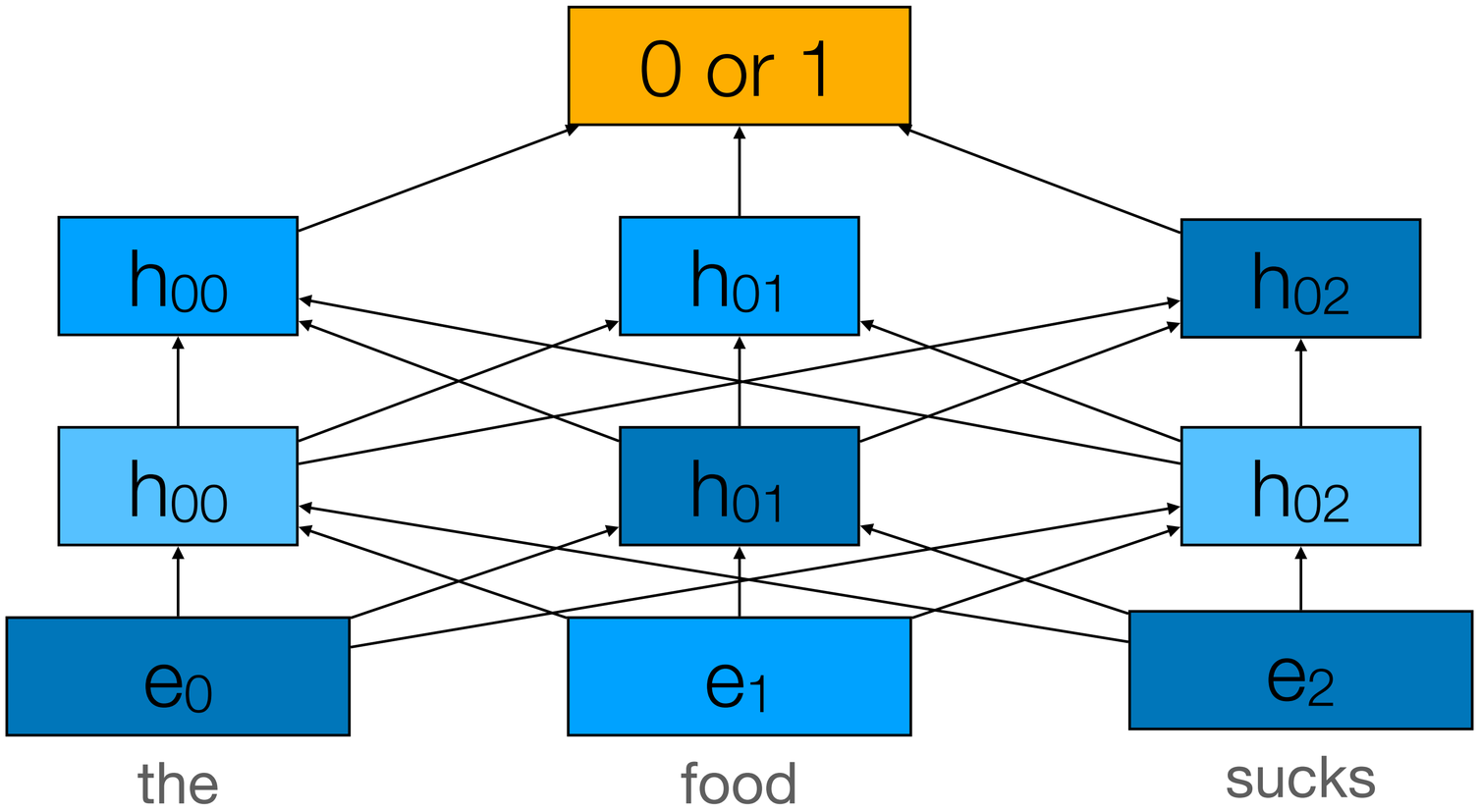


Pretraining

Finetuning

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative

Adjust output layers to reflect target task
(for BERT, can use CLS for this)

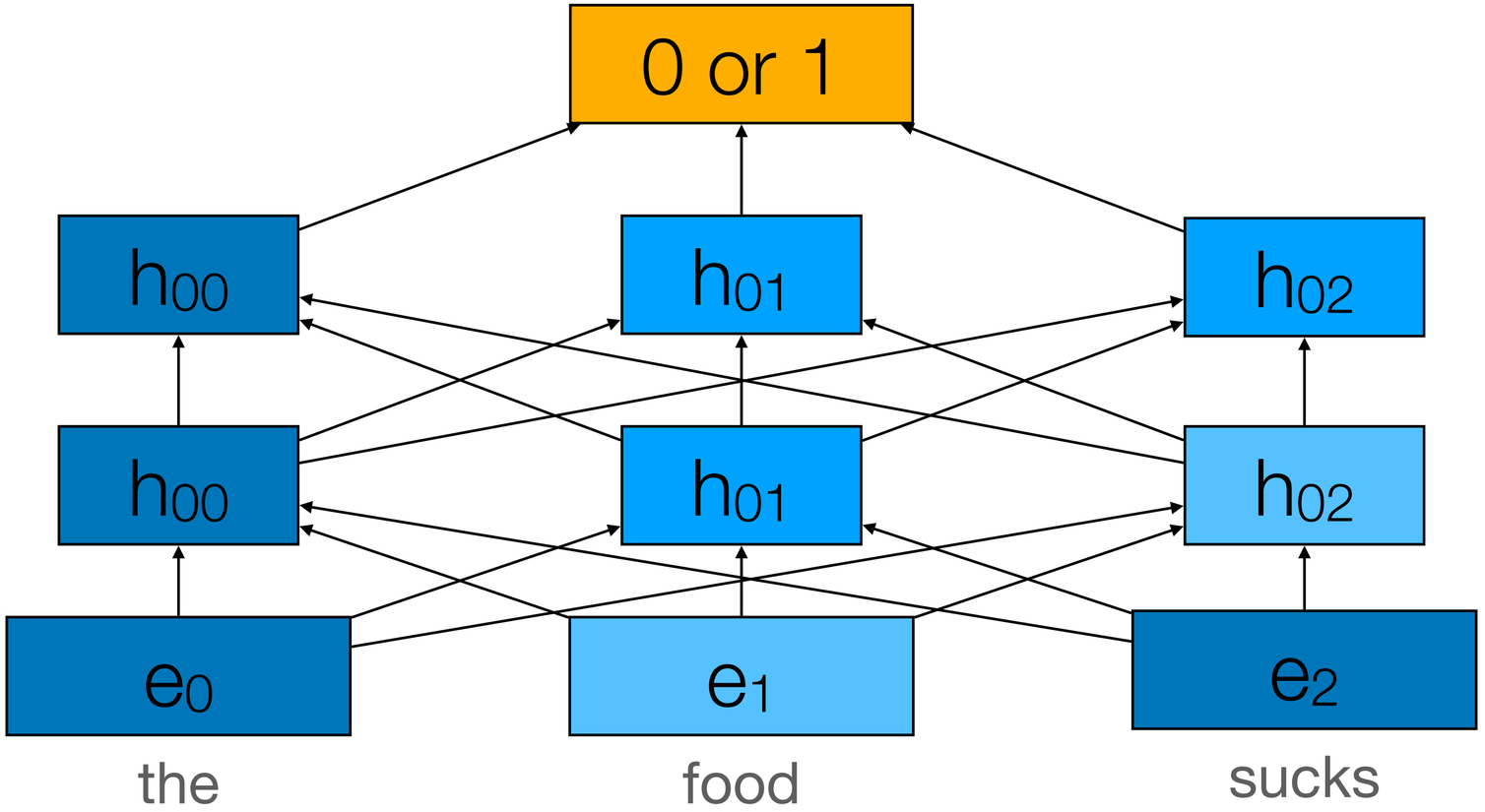


Pretraining

Finetuning

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative

Keep training network, backproping through everything as needed.



Pretraining

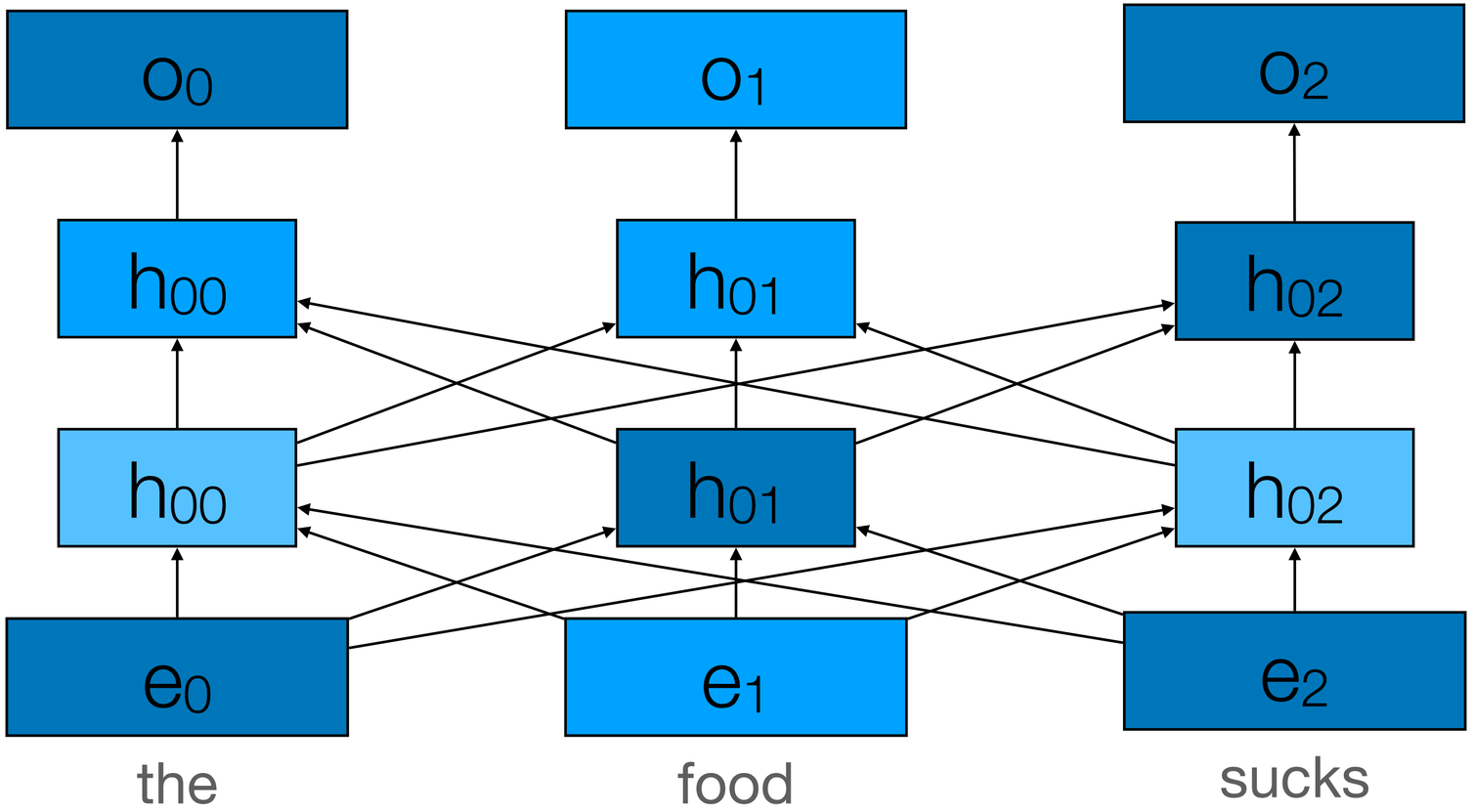
Transferring pretrained representations

- Frozen: Just pool representations and train a new classifier on top
- Finetuning: Just treat pretraining as a good initialization. On new task, continue to update all parameters
- Parameter-Efficient or “Prompt” Tuning: Just update a small number of parameters at the bottom of the network

Pretraining

Parameter-Efficient (“Prompt”) Tuning

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative

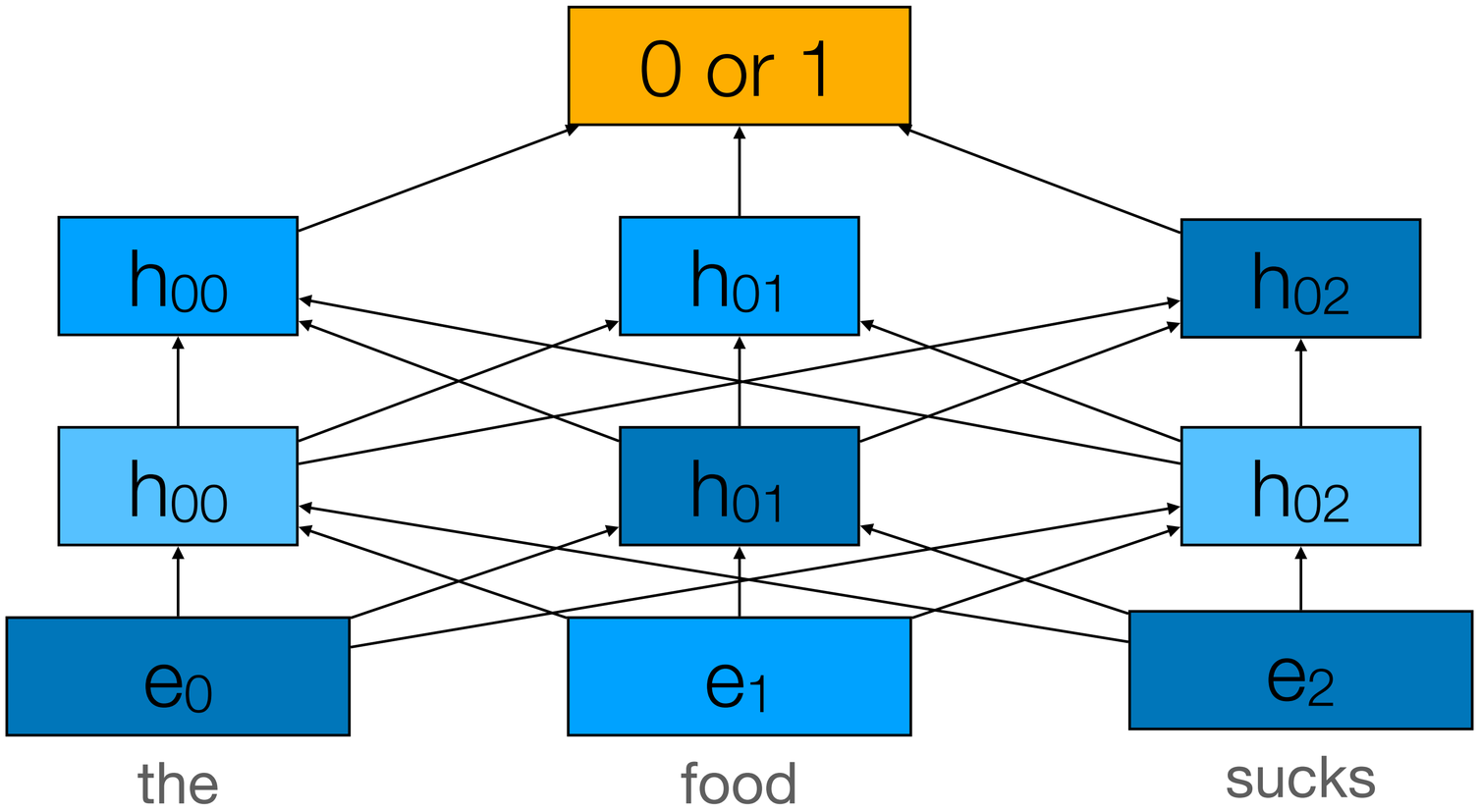


Pretraining

Parameter-Efficient ("Prompt") Tuning

Task: Sentiment Analysis
Input: "the food sucks"
Expected: Negative

Adjust output layers to reflect target task

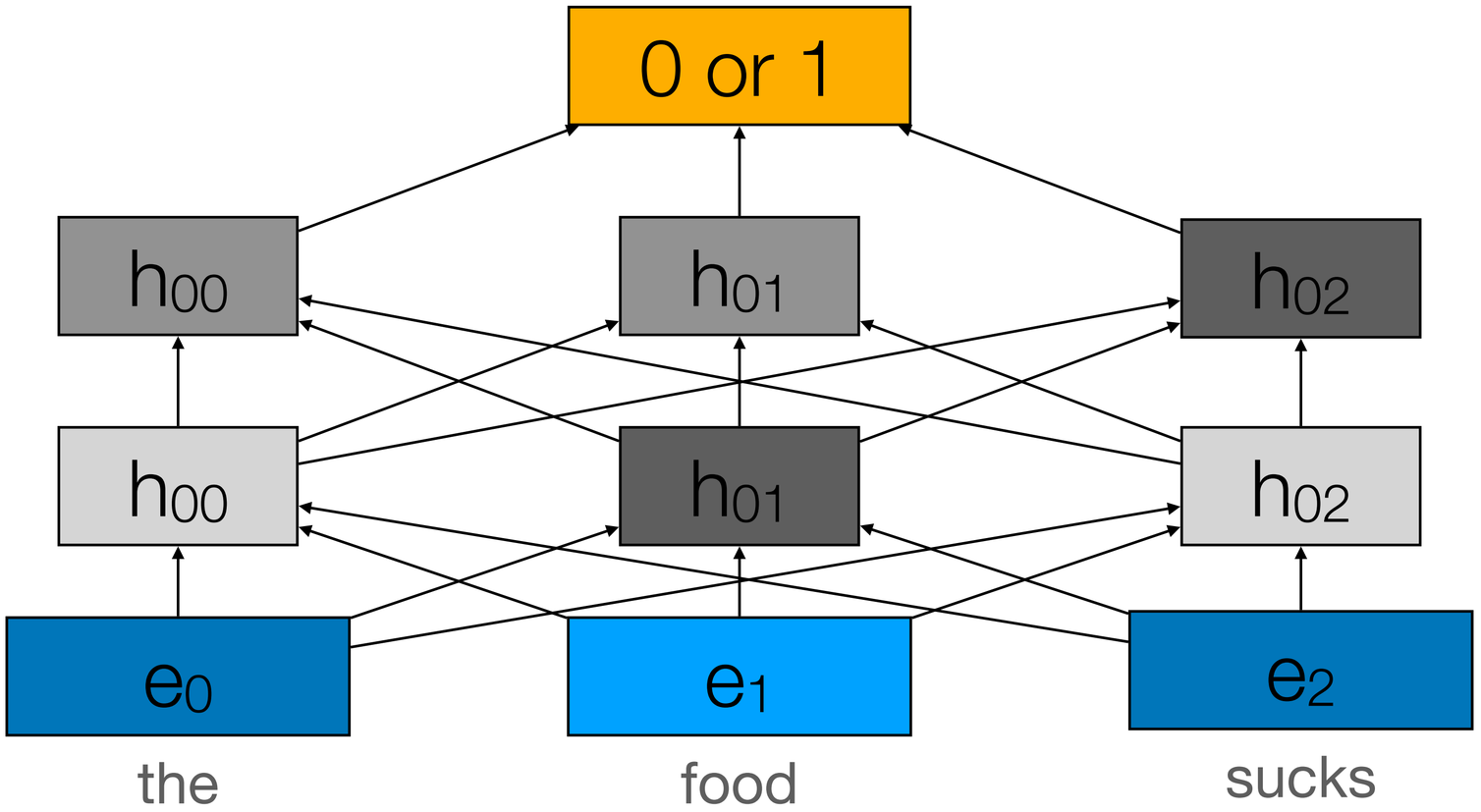


Pretraining

Parameter-Efficient ("Prompt") Tuning

Task: Sentiment Analysis
Input: "the food sucks"
Expected: Negative

Freeze everything except a small number of parameters (e.g., the embeddings)



Pretraining

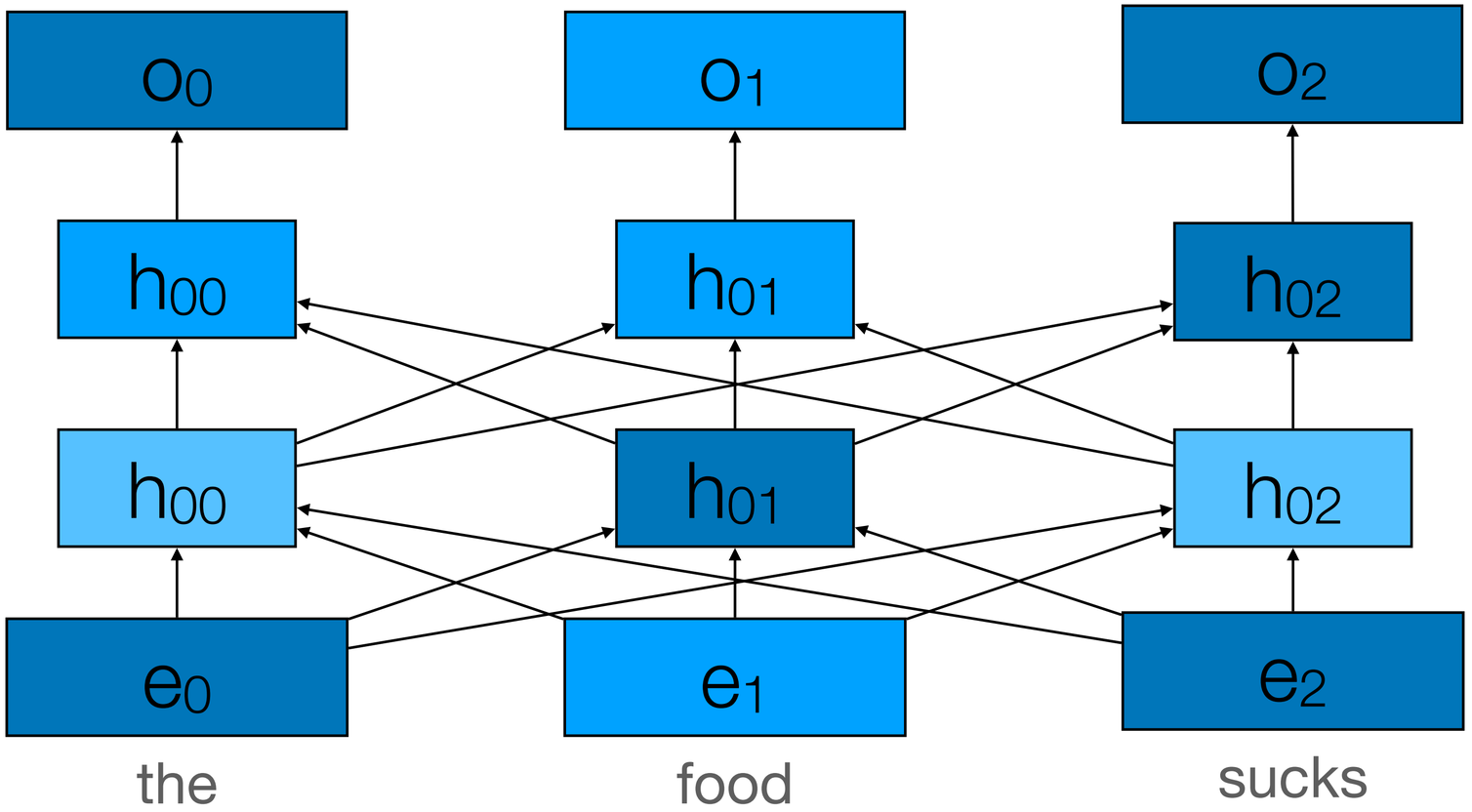
Transferring pretrained representations

- Frozen: Just pool representations and train a new classifier on top
- Finetuning: Just treat pretraining as a good initialization. On new task, continue to update all parameters
- Parameter-Efficient or “Prompt” Tuning: Just update a small number of parameters at the bottom of the network
- Adapters: Just update a small number of parameters throughout the network

Pretraining

Adapters

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative

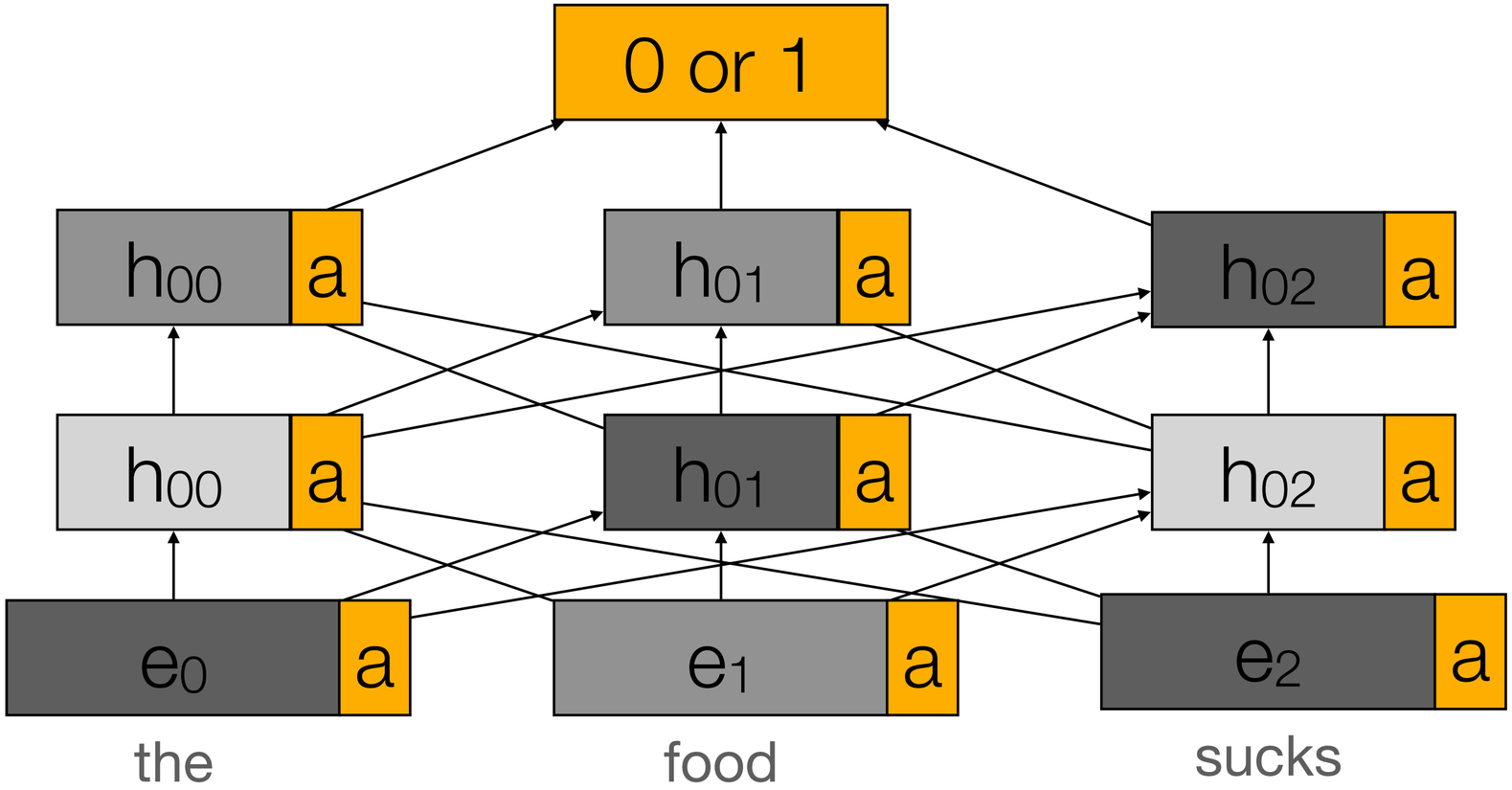


Pretraining

Parameter-Efficient ("Prompt") Tuning

Task: Sentiment Analysis
Input: "the food sucks"
Expected: Negative

Add a few parameters at each layer to tune, freeze the rest.



Pretraining

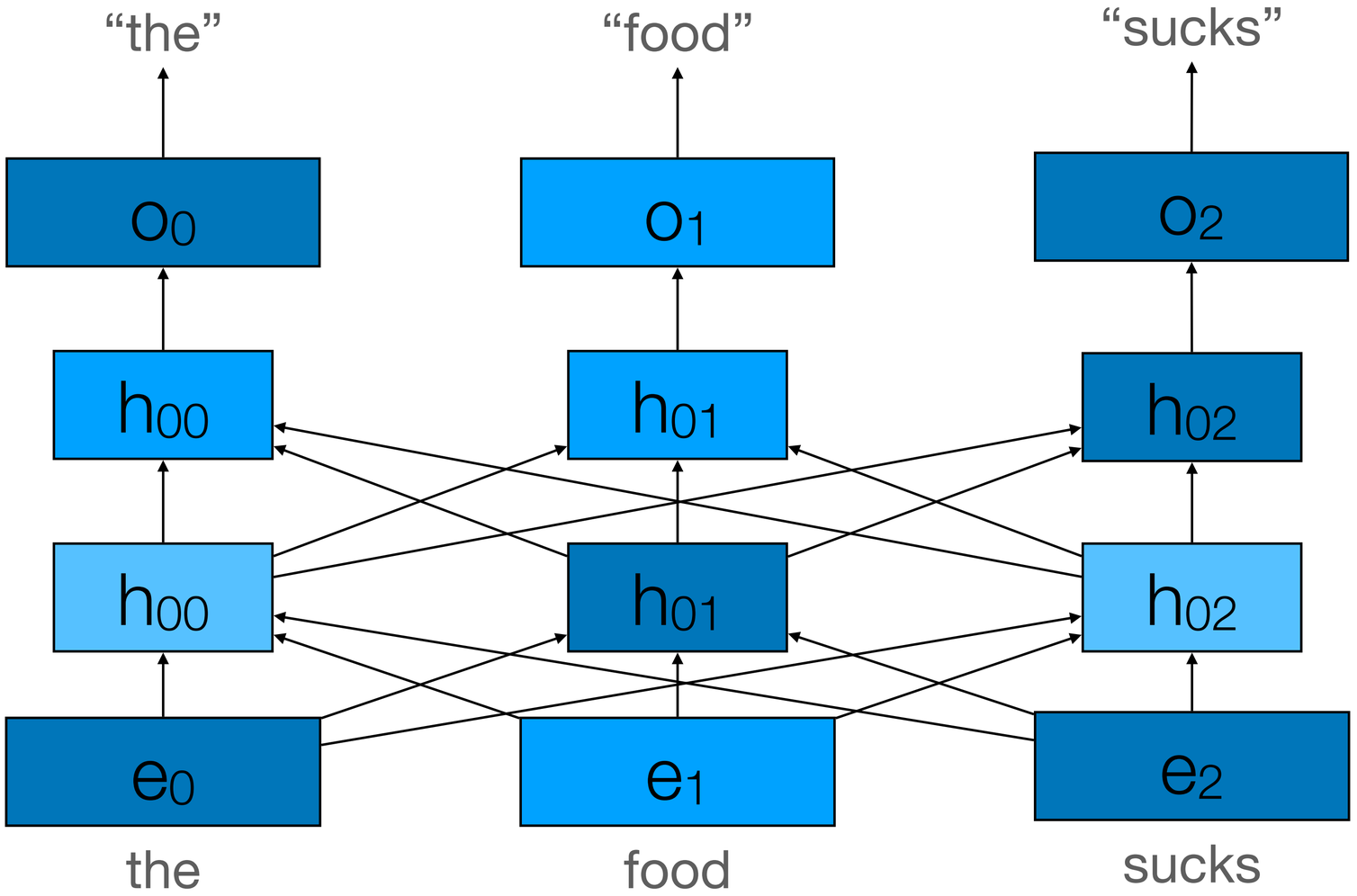
Transferring pretrained representations

- Frozen: Just pool representations and train a new classifier on top
- Finetuning: Just treat pretraining as a good initialization. On new task, continue to update all parameters
- Parameter-Efficient or “Prompt” Tuning: Just update a small number of parameters at the bottom of the network
- Adapters: Just update a small number of parameters throughout the network
- Zero-Shot or “In-Context” Learning: Cast all tasks as an instance of the task the model was trained on (e.g., language modeling)

Pretraining

Zero-Shot Transfer

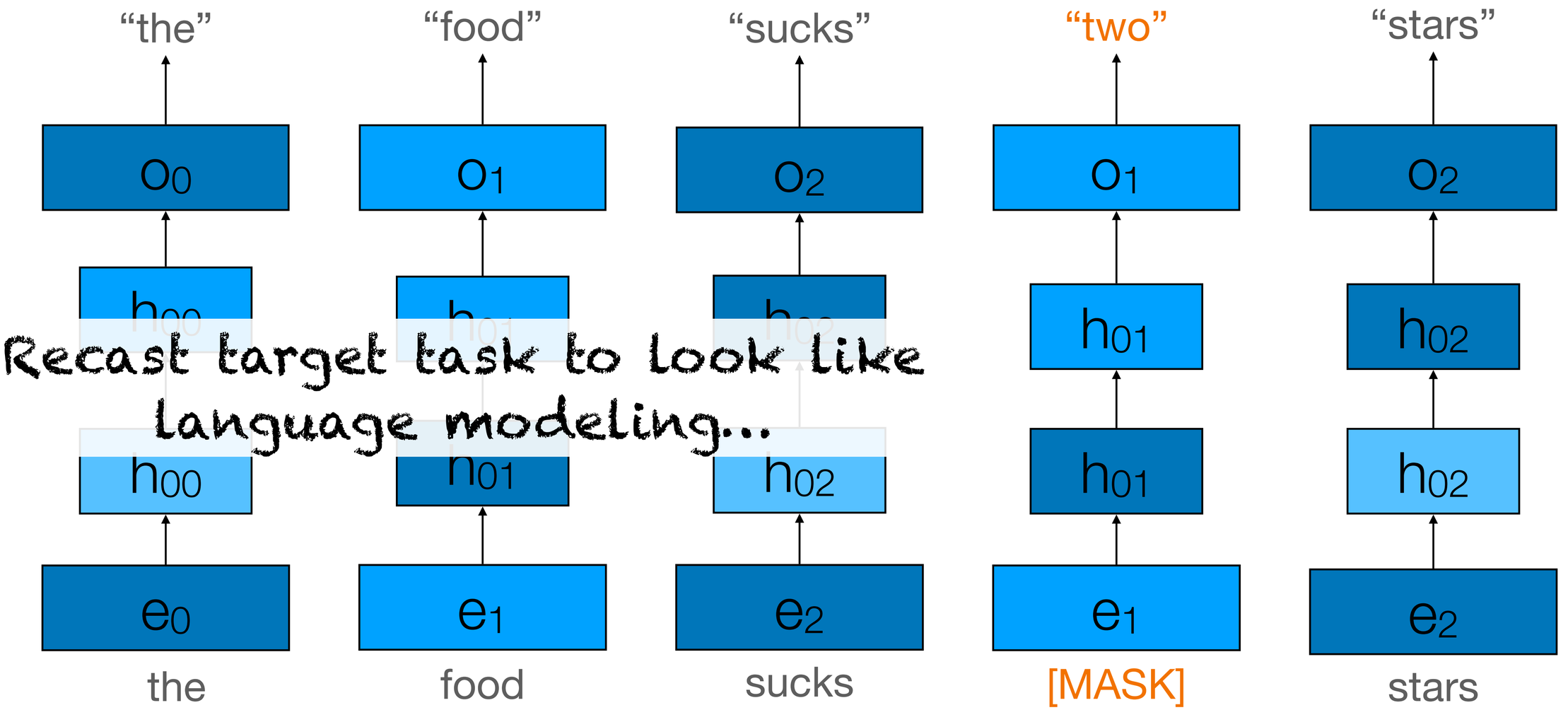
Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative



Pretraining

Zero-Shot Transfer

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative

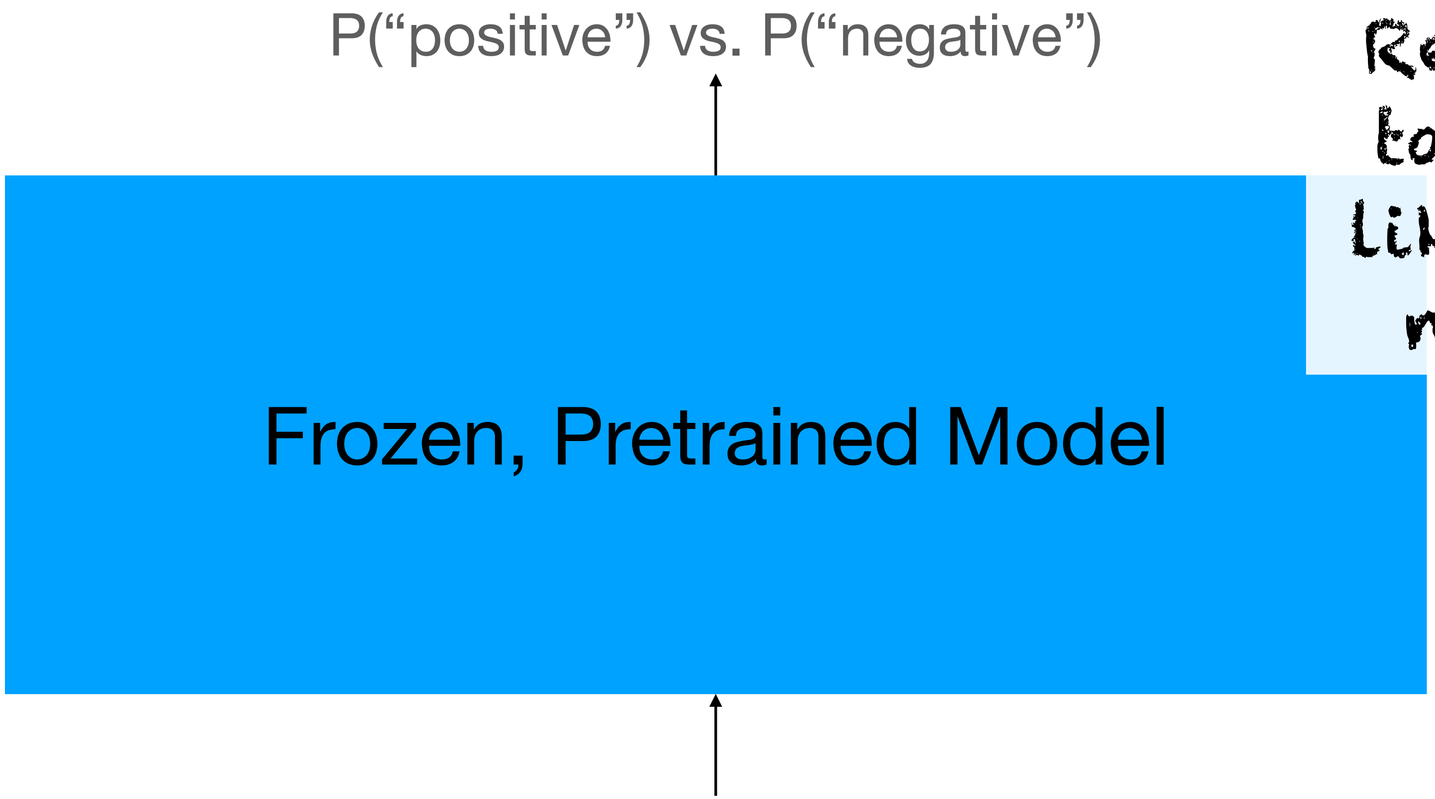


Recast target task to look like language modeling...

Pretraining

Zero-Shot Transfer

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative



Recast target task to look like language modeling...

Pretraining

In-Context Learning

Task: Sentiment Analysis
Input: “the food sucks”
Expected: Negative

P(“positive”) vs. P(“negative”)

↑

Frozen, Pretrained Model

Optionally, offering some examples to guide the model on task and format

↑

Is the following review positive or negative?
Awesome place -> positive
Would not recommend -> negative
My absolute fav -> positive
This food sucks. ->

Pretraining

How well do these methods work?

- In general, very well.
- Most evaluations focus on **task performance**, models do very well on that front
- **Finetuning**: Models perform better and require less data to learn the target task. Still require **100s or 1000s of examples**, typically. And can result in **“catestrophic forgetting”**
- **Parameter-Efficient/Adapters**: Newer, so still being explored. But current results suggest very large models can be adapted in **~15 minutes with O(100) examples**.
- **Zero-Shot/In-Context Learning**: More controversial. Results are **high variance**, depend on exact wording of the prompts. Unclear what was seen during training so hard to know how much they are truly learning and **generalizing vs. parroting**.