UMass · CS685 | Advanced Natural Language Processing (2020)

CS685 (2020)· 课程资料包 @ShowMeAI

视频
中英双语字幕

课件
一键打包下载

笔记
官方笔记翻译

代码
作业项目解析

视频 · B 站 [扫码或点击链接]
https://www.bilibili.com/video/BV1BL411t7RV

课件 & 代码 · 博客 [扫码或点击链接]
http://blog.showmeai.tech/umass-cs685

NLP 语言模型 问答系统 文本生成 BERT 语义解析 GPT-3 知识推理 模型蒸馏 迁移学习 注意力机制 transformer

Awesome AI Courses Notes Cheatsheets 是 ShowMeAI 资料库的分支系列，覆盖最具知名度的 **TOP50+** 门 AI 课程，旨在为读者和学习者提供一整套高品质中文学习笔记和速查表。

点击课程名称，跳转至课程**资料包**页面，**一键下载**课程全部资料！

机器学习	深度学习	自然语言处理	计算机视觉
Stanford · CS229	Stanford · CS230	Stanford · CS224n	Stanford · CS231n
# Awesome AI Courses Notes Cheatsheets· 持续更新中			
知识图谱	**图机器学习**	**深度强化学习**	**自动驾驶**
Stanford · CS520	Stanford · CS224W	UCBerkeley · CS285	MIT · 6.S094

微信公众号

资料下载方式 2：扫码点击底部菜单栏

称为 **AI 内容创作者?** 回复 [添砖加瓦]

Transfer learning with neural language models

CS 685, Spring 2020
Advanced Natural Language Processing

Mohit Iyyer
College of Information and Computer Sciences
University of Massachusetts Amherst

many slides from Jacob Devlin & Matt Peters

Stuff from last time…

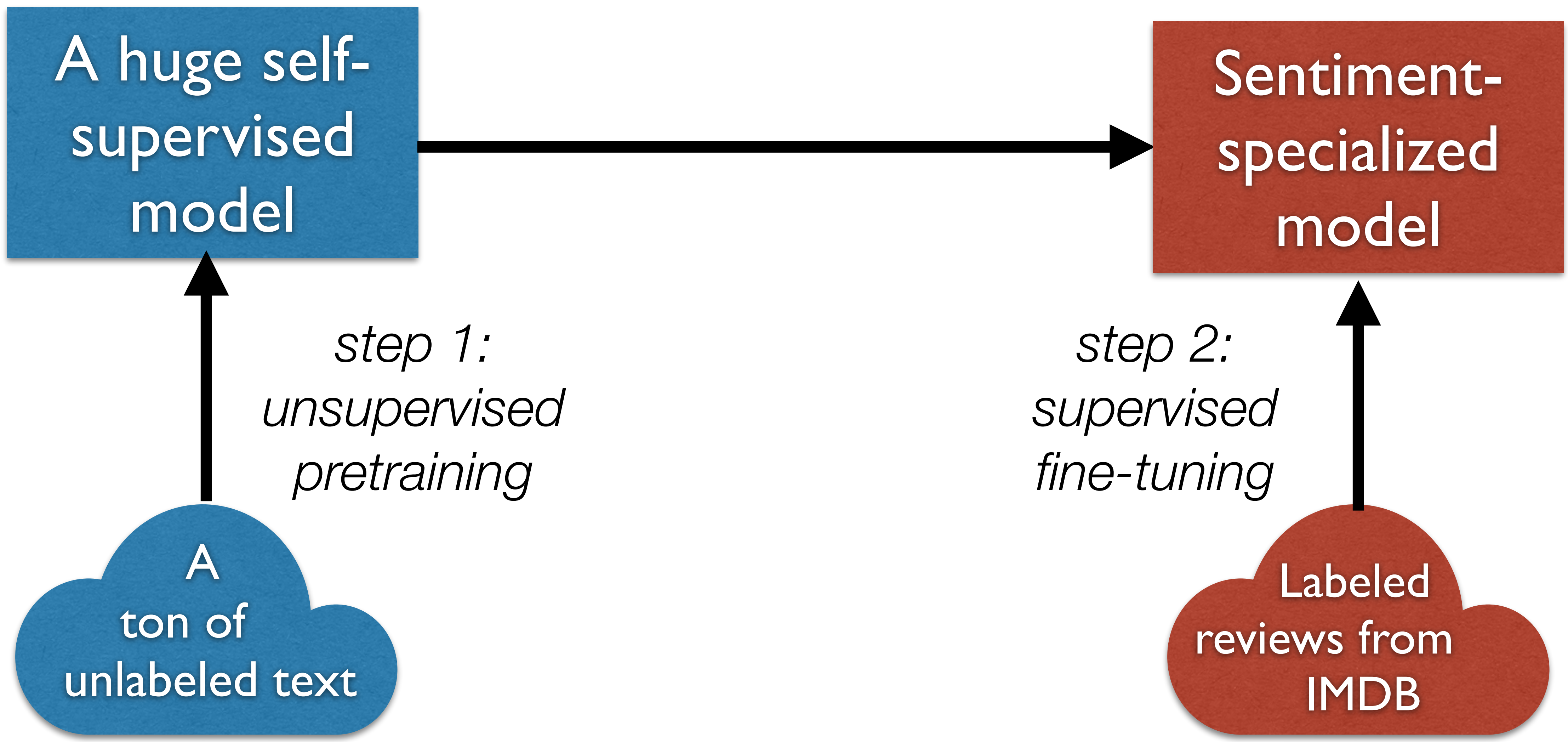
- Project proposals due 9/21, please use Overleaf template
- Still working on making the next homework computationally feasible on Colab, look out for it next week
- Please ask other questions (about logistics / material / etc) in the chatbox!

Do NNs really need millions of labeled examples?

- Can we leverage *unlabeled* data to cut down on the number of labeled examples we need?

What is transfer learning?

- **In our context:** take a network trained on a task for which it is easy to generate labels, and adapt it to a different task for which it is harder.
  - **In computer vision:** train a CNN on ImageNet, transfer its representations to every other CV task
  - **In NLP:** train a really big language model on billions of words, transfer to every NLP task!


A huge self-supervised model
Sentiment-specialized model
step 1: unsupervised pretraining
step 2: supervised fine-tuning
A ton of unlabeled text
Labeled reviews from IMDB

language models for transfer learning

Deep contextualized word representations. Peters et al., NAACL 2018

Previous methods (e.g., word2vec) represent each word type with a **single vector**

play = [0.2, -0.1, 0.5, ...]
bank = [-0.3, 1.4, 0.7, ...]
run = [-0.5, -0.3, -0.1, ...]

NNs are then used to compose those vectors over longer sequences

Single vector per word

The new-look ***play*** area is due to be completed by early spring 2010 .

Single vector per word

Gerrymandered congressional districts favor representatives who ***play*** to the party base .

Single vector per word

The freshman then completed the three-point ***play*** for a 66-63 lead .

Nearest neighbors

play = `[0.2, -0.1, 0.5, ...]`

Nearest Neighbors

playing	plays
game	player
games	Play
played	football
players	multiplayer

Multiple senses entangled

play = [0.2, -0.1, 0.5, ...]

Nearest Neighbors

VERB

playing	plays
game	player
games	Play
played	football
players	multiplayer

Multiple senses entangled

play = [0.2, -0.1, 0.5, ...]

Nearest Neighbors

playing	**VERB**	plays
game	**NOUN**	player
games		Play
played		football
players		multiplayer

Multiple senses entangled

play = [0.2, -0.1, 0.5, ...]

Nearest Neighbors

playing	VERB	plays
game	NOUN	player
games		Play
played	ADJ	football
players		multiplayer

Contextual Representations

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

  `open a bank account`   `on the river bank`

  `[0.3, 0.2, -0.8, …]`

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

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

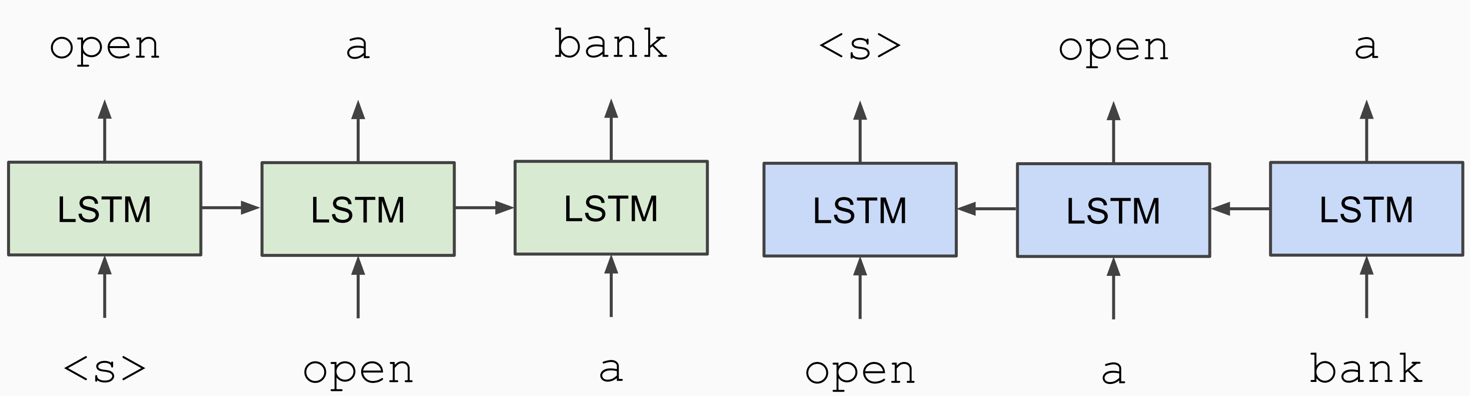
  `open a bank account`   `on the river bank`

Examples on iPad

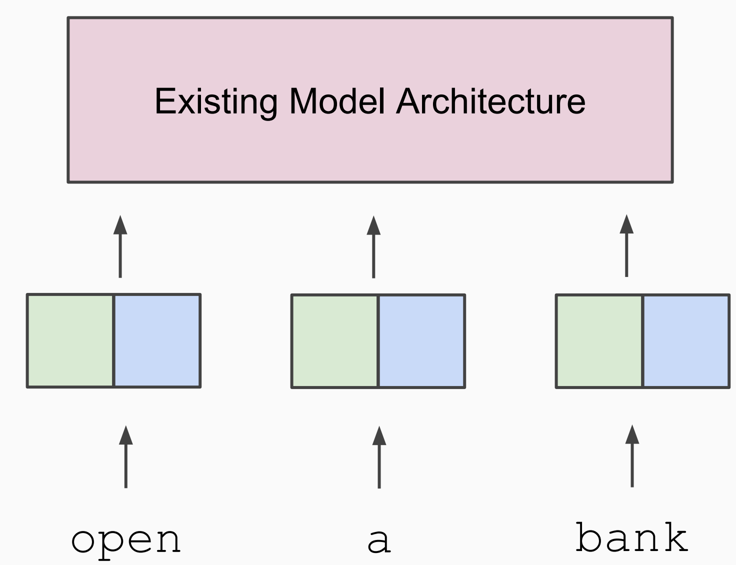
History of Contextual Representations

- *ELMo: Deep Contextual Word Embeddings*, AI2 & University of Washington, 2017

Train Separate Left-to-Right and Right-to-Left LMs



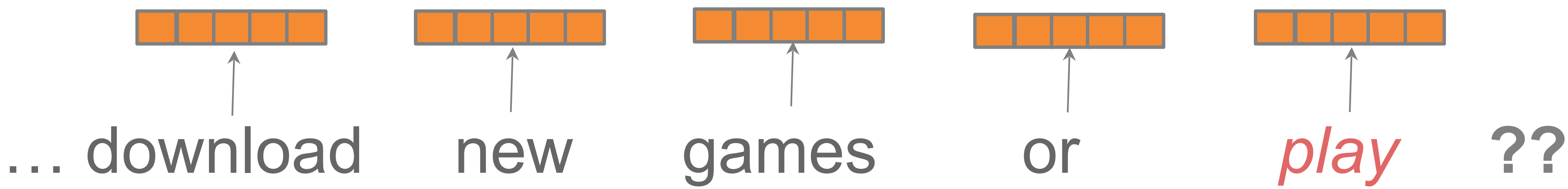
Apply as "Pre-trained Embeddings"



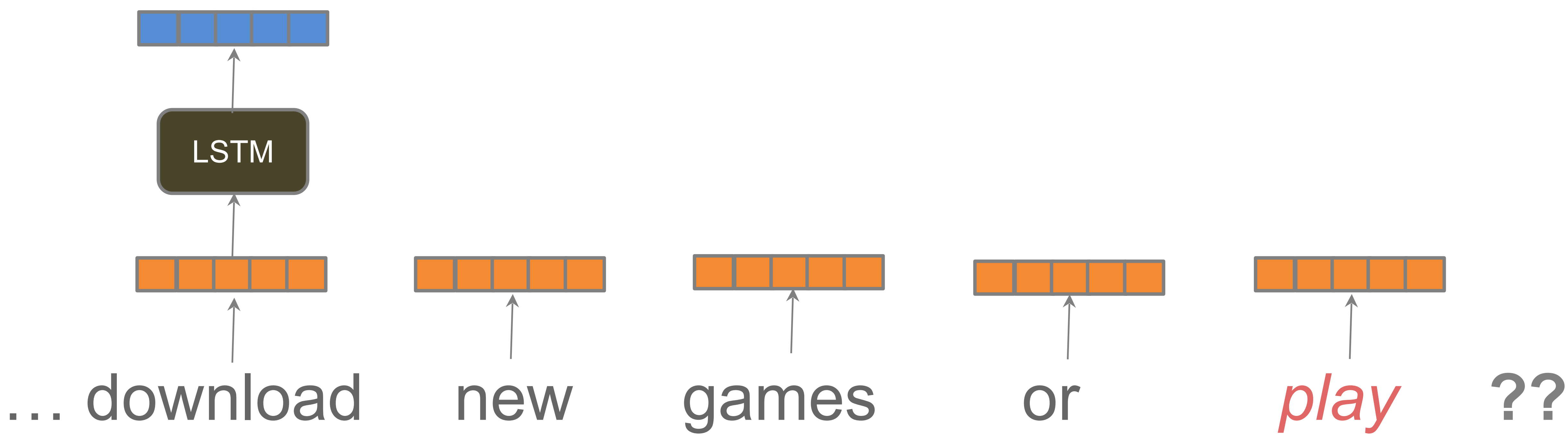
Deep bidirectional language model

… download new games or *play* ??

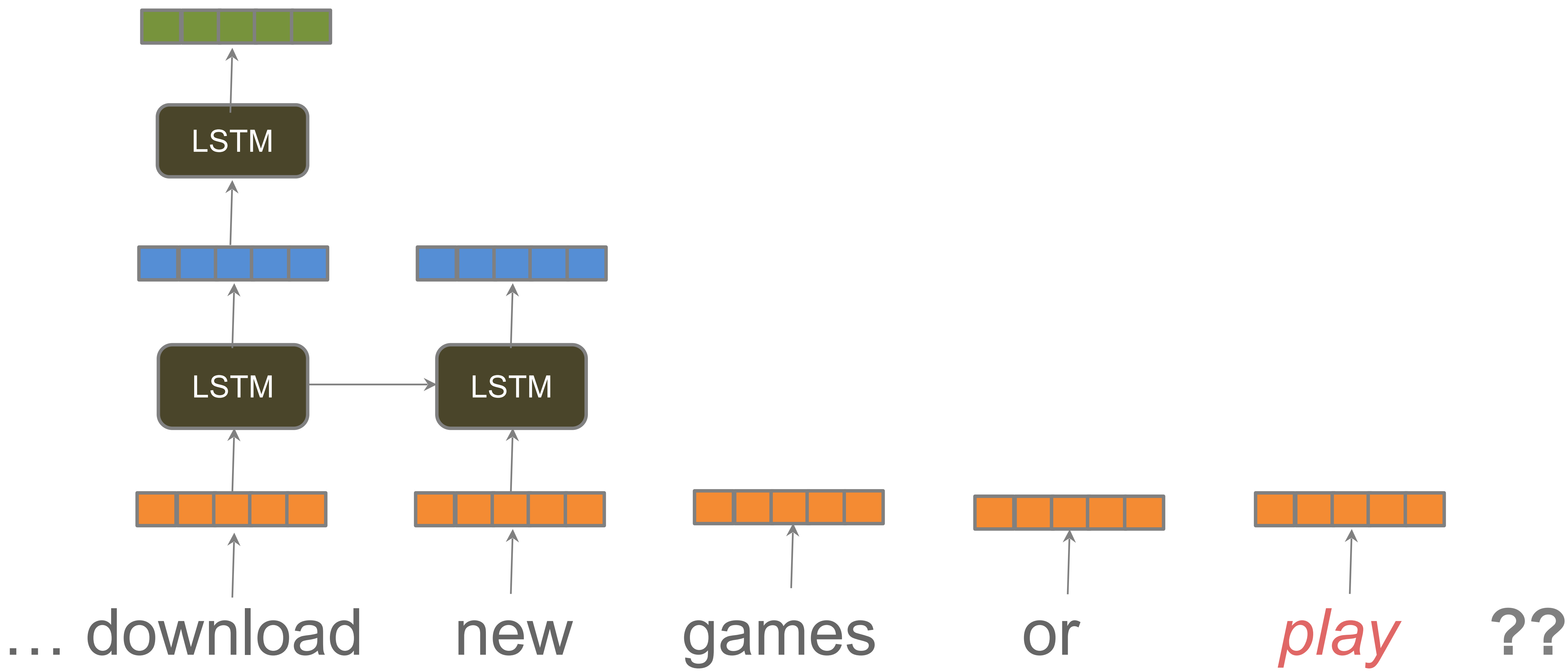
Deep bidirectional language model



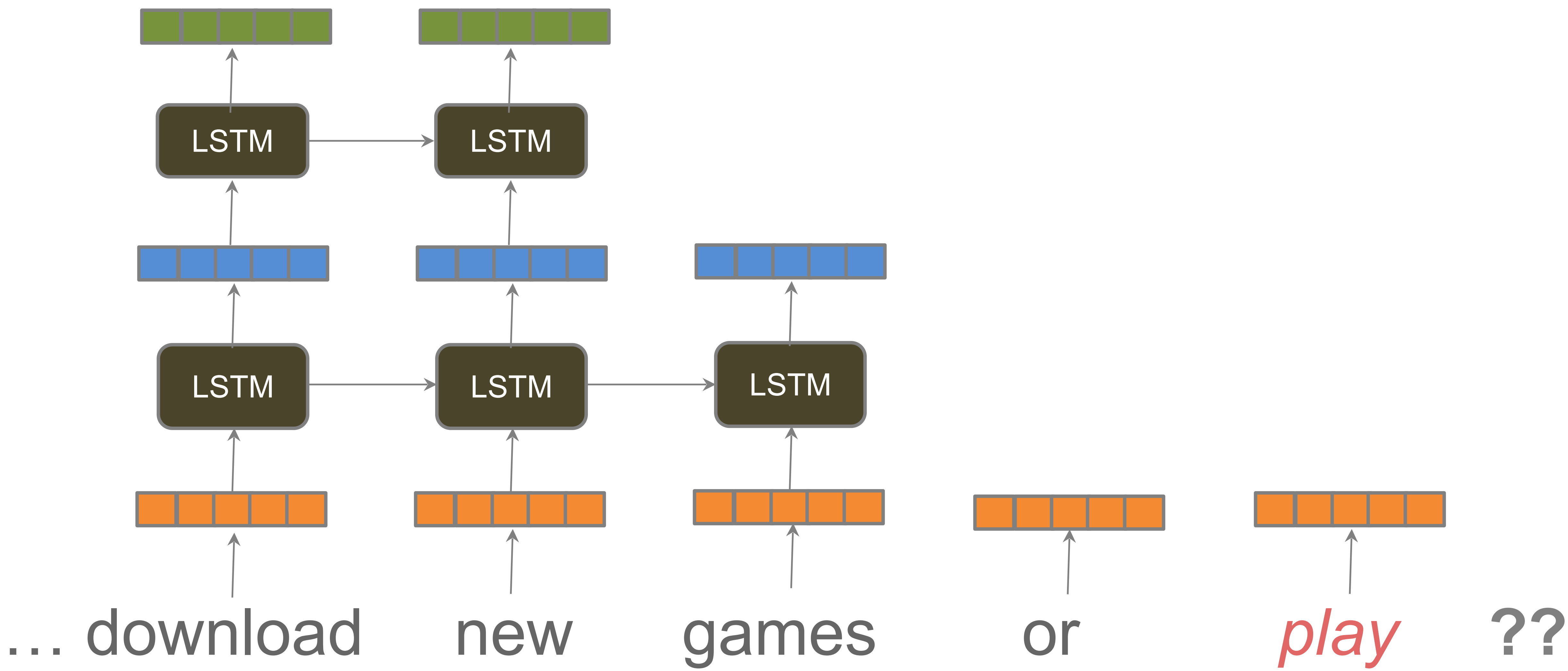
Deep bidirectional language model



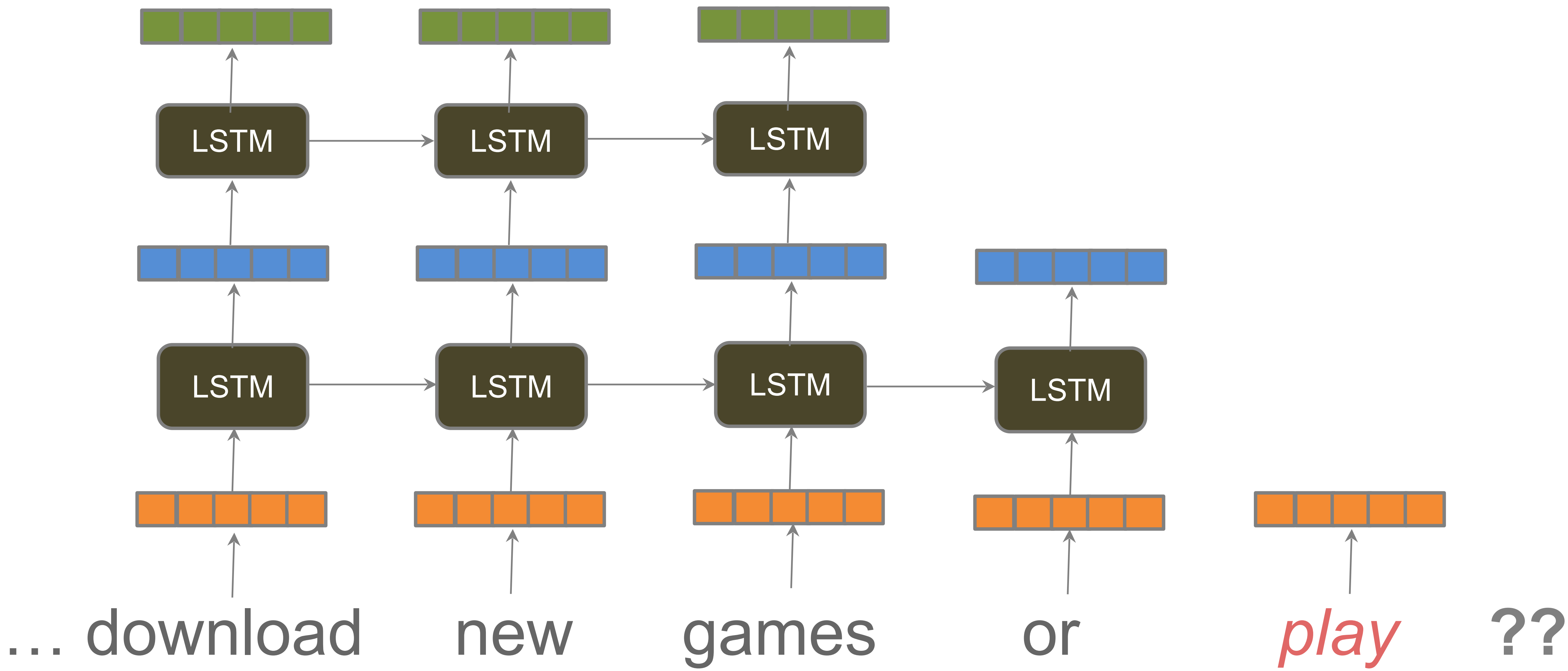
Deep bidirectional language model



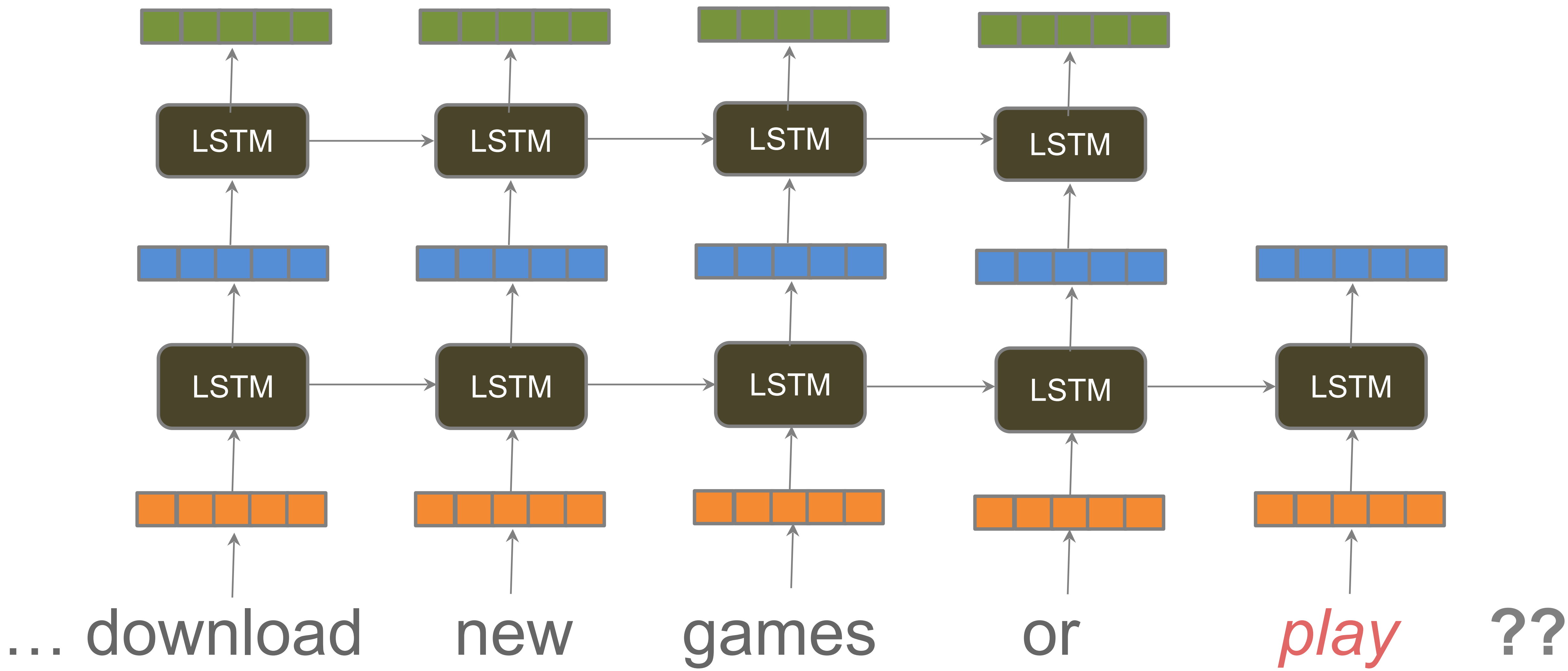
Deep bidirectional language model



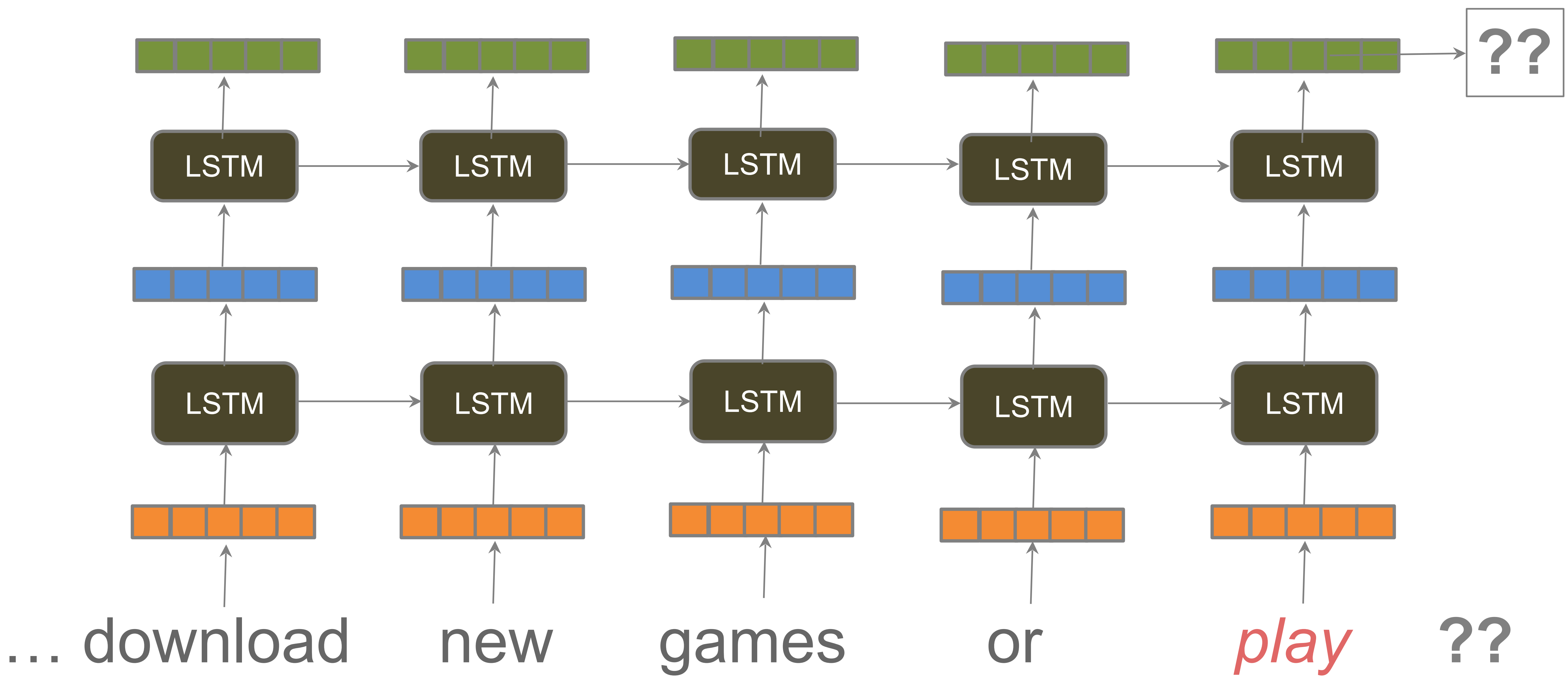
Deep bidirectional language model



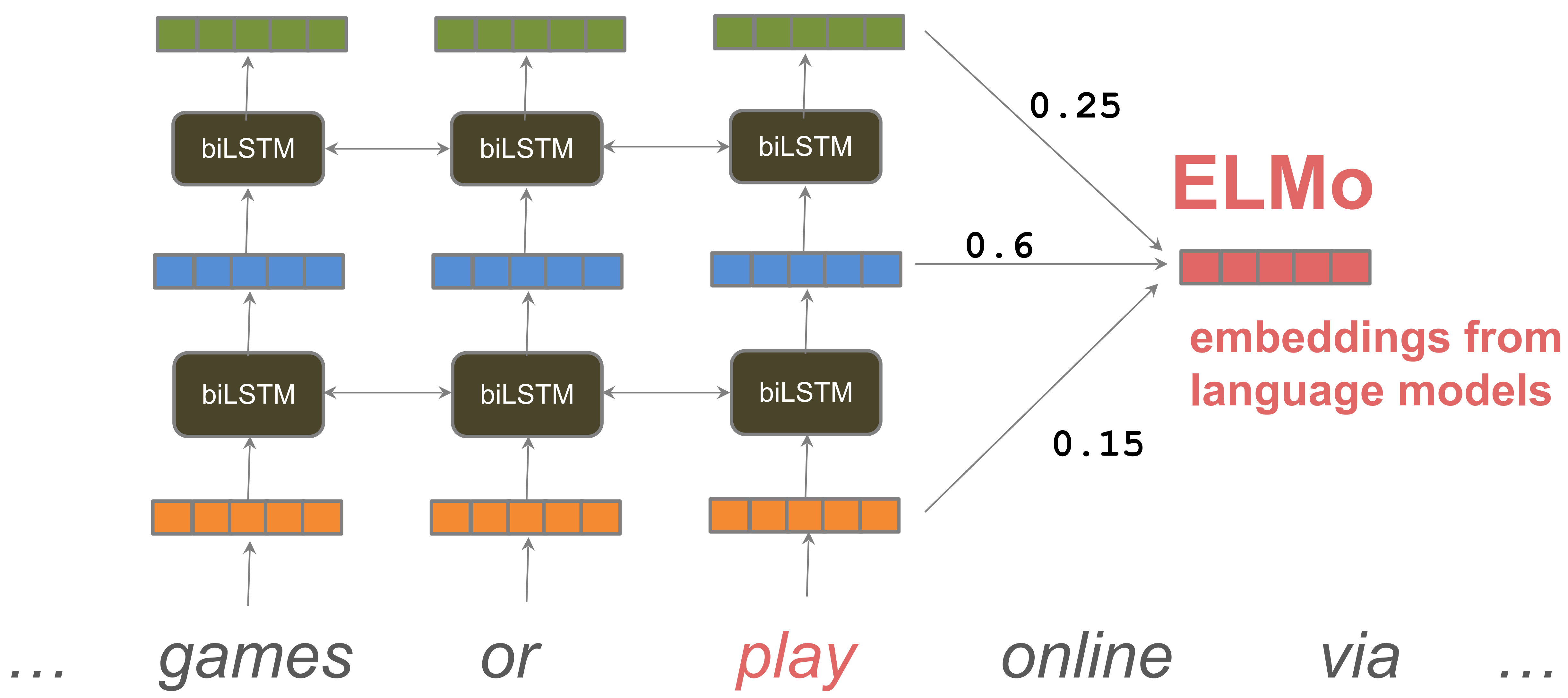
Deep bidirectional language model



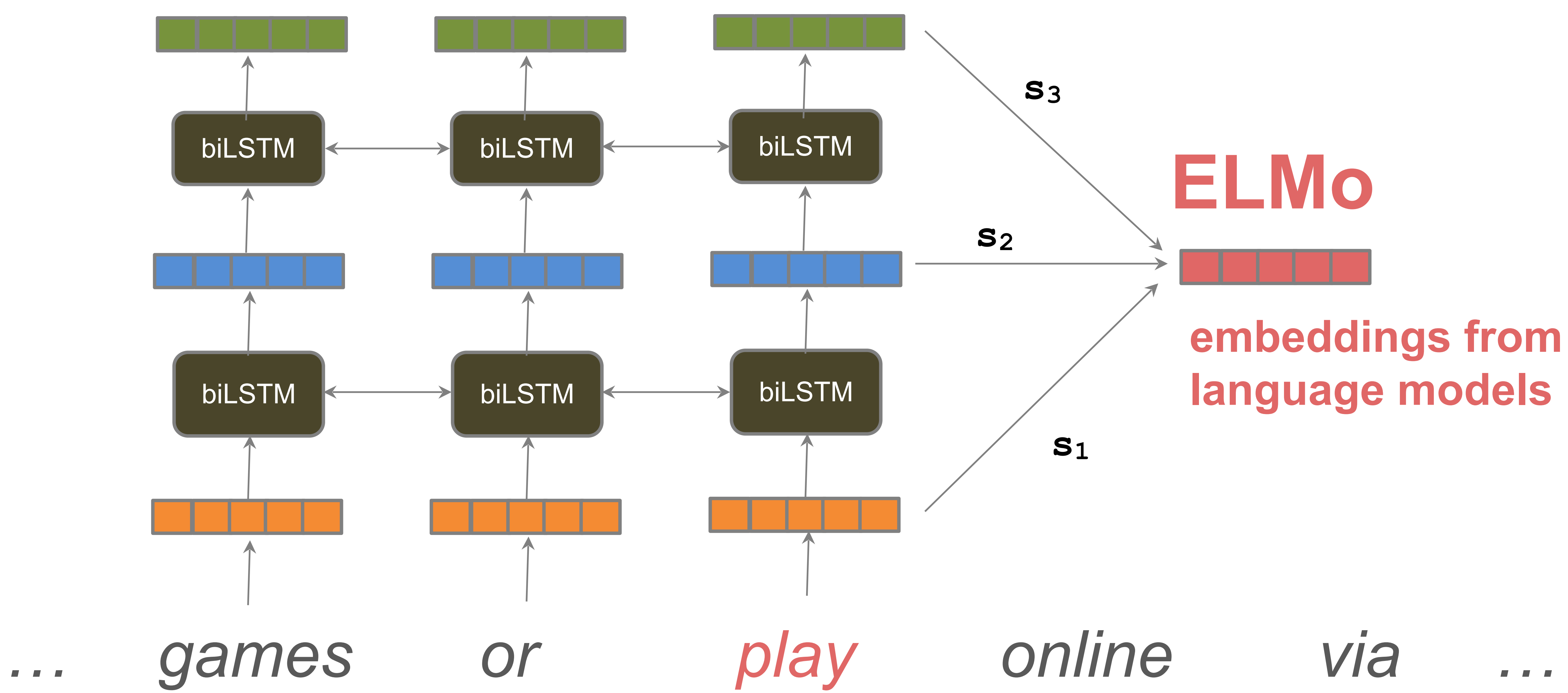
Deep bidirectional language model



Use all layers of language model



Learned task-specific combination of layers



Contextual representations

ELMo representations are **contextual** – they depend on the entire sentence in which a word is used.

how many different embeddings does ELMo compute for a given word?

ELMo improves NLP tasks

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 ± 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 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Large-scale recurrent neural language models learn contextual representations that capture basic elements of semantics and syntax

Adding ELMo to existing state-of-the-art models provides significant performance improvement on all NLP tasks.

TensorFlow™

```
elmo = hub.Module("https://tfhub.dev/google/elmo/1", trainable=True)
embeddings = elmo(
    ["the cat is on the mat", "dogs are in the fog"],
    signature="default",
    as_dict=True)["elmo"]
```

AllenNLP

FROM  TO 

Problem with Previous Methods

- **Problem**: Language models only use left context *or* right context, but language understanding is bidirectional.
- Why are LMs unidirectional?

Problem with Previous Methods

- **Problem**: Language models only use left context *or* right context, but language understanding is bidirectional.
- Why are LMs unidirectional?
- Reason 1: Directionality is needed to generate a well-formed probability distribution.
  - We don't care about this.

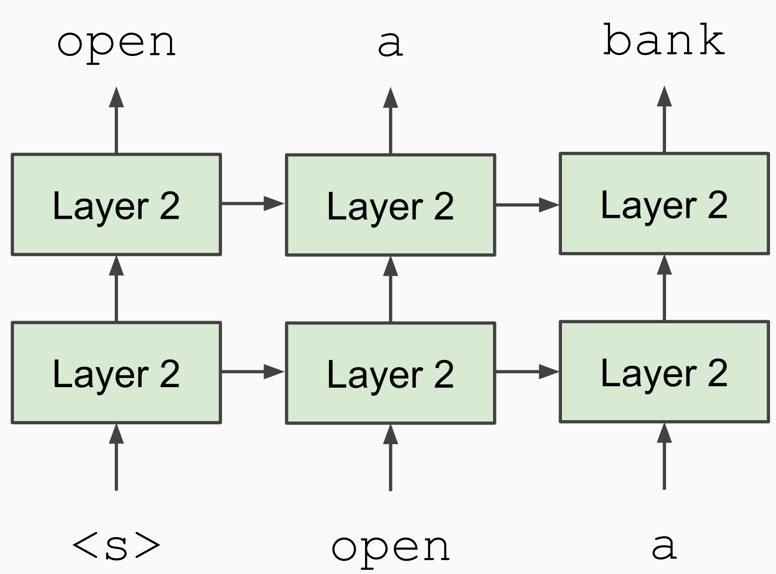
Why not?

Problem with Previous Methods

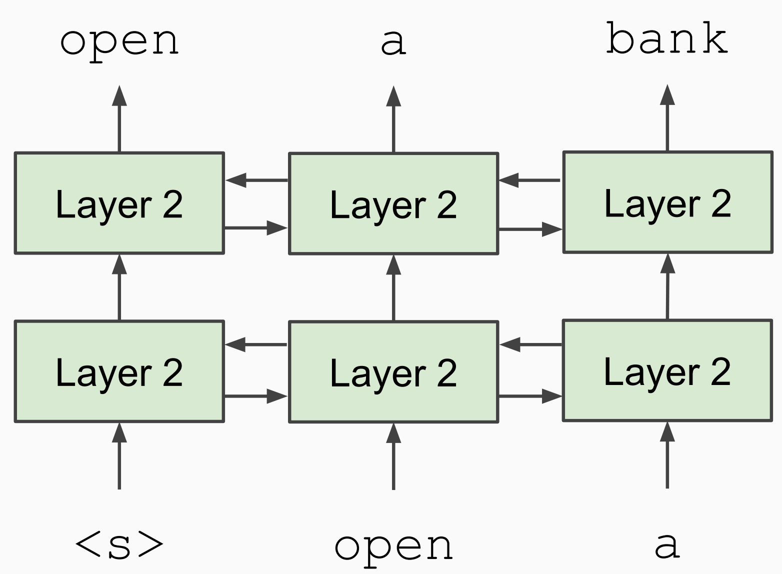
- **Problem**: Language models only use left context *or* right context, but language understanding is bidirectional.
- Why are LMs unidirectional?
- <u>Reason 1</u>: Directionality is needed to generate a well-formed probability distribution.
  - We don't care about this.
- <u>Reason 2</u>: Words can "see themselves" in a bidirectional encoder.

Unidirectional vs. Bidirectional Models

Unidirectional context
Build representation incrementally



Bidirectional context
Words can “see themselves”



Masked LM

- **Solution:** Mask out *k*% of the input words, and then predict the masked words
  - We always use *k* = 15%

```
                      store            gallon
                        ↑                ↑
the man went to the [MASK] to buy a [MASK] of milk
```

What are the pros and cons of increasing *k*?

Masked LM

- Problem: Mask token never seen at fine-tuning
- Solution: 15% of the words to predict, but don't replace with `[MASK]` 100% of the time. Instead:
- 80% of the time, replace with `[MASK]`

  `went to the store` → `went to the [MASK]`
- 10% of the time, replace random word

  `went to the store` → `went to the running`
- 10% of the time, keep same

  `went to the store` → `went to the store`

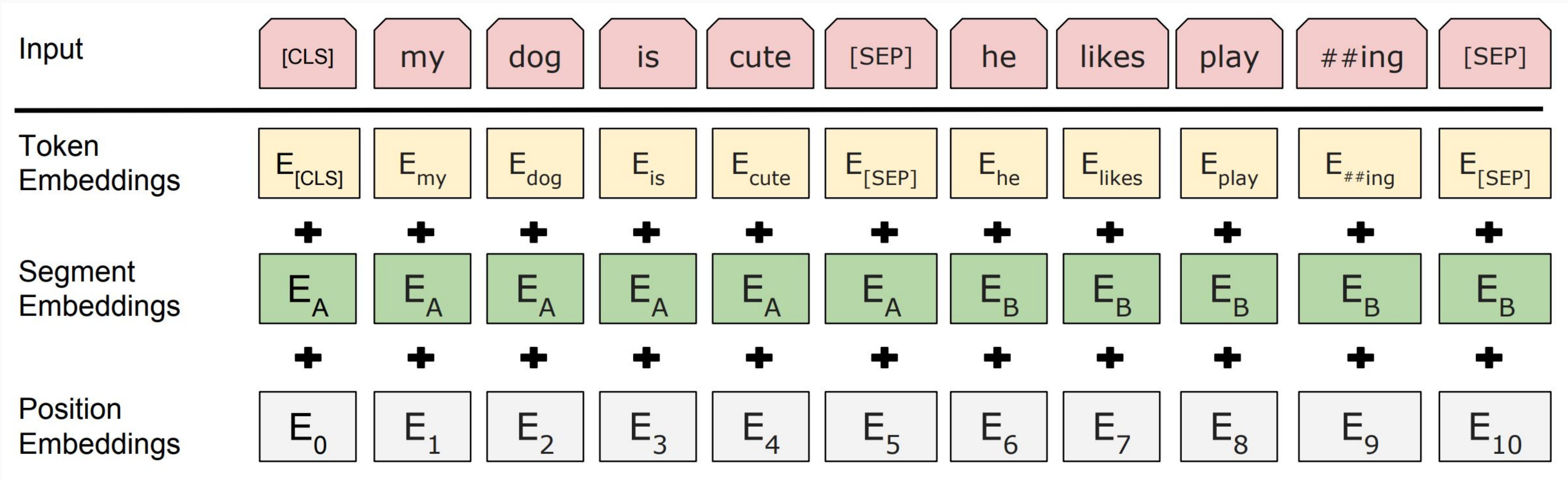
Next Sentence Prediction

- To learn *relationships* between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence

Sentence A = `The man went to the store.` **Sentence B** = `He bought a gallon of milk.` **Label** = `IsNextSentence`	**Sentence A** = `The man went to the store.` **Sentence B** = `Penguins are flightless.` **Label** = `NotNextSentence`

This has since been shown to be unimportant (and can be removed e.g., in RoBERTa)

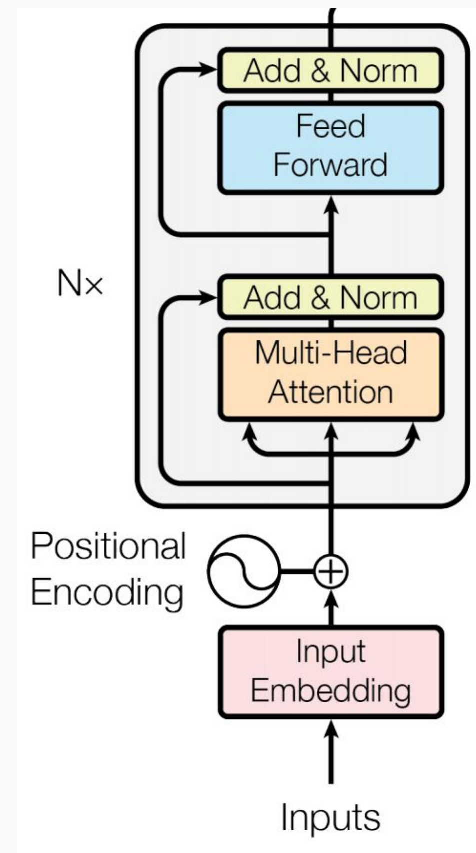
Input Representation

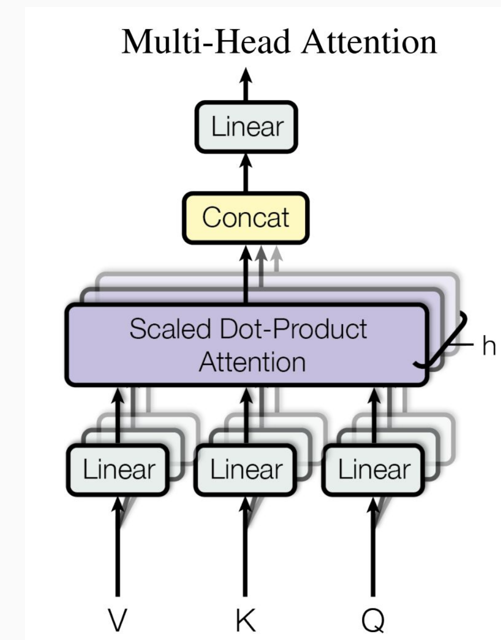


- Use 30,000 WordPiece vocabulary on input.
- Each token is sum of three embeddings
- Single sequence is much more efficient.

Transformer encoder

- Multi-headed self attention
  - Models context
- Feed-forward layers
  - Computes non-linear hierarchical features
- Layer norm and residuals
  - Makes training deep networks healthy
- Positional embeddings
  - Allows model to learn relative positioning

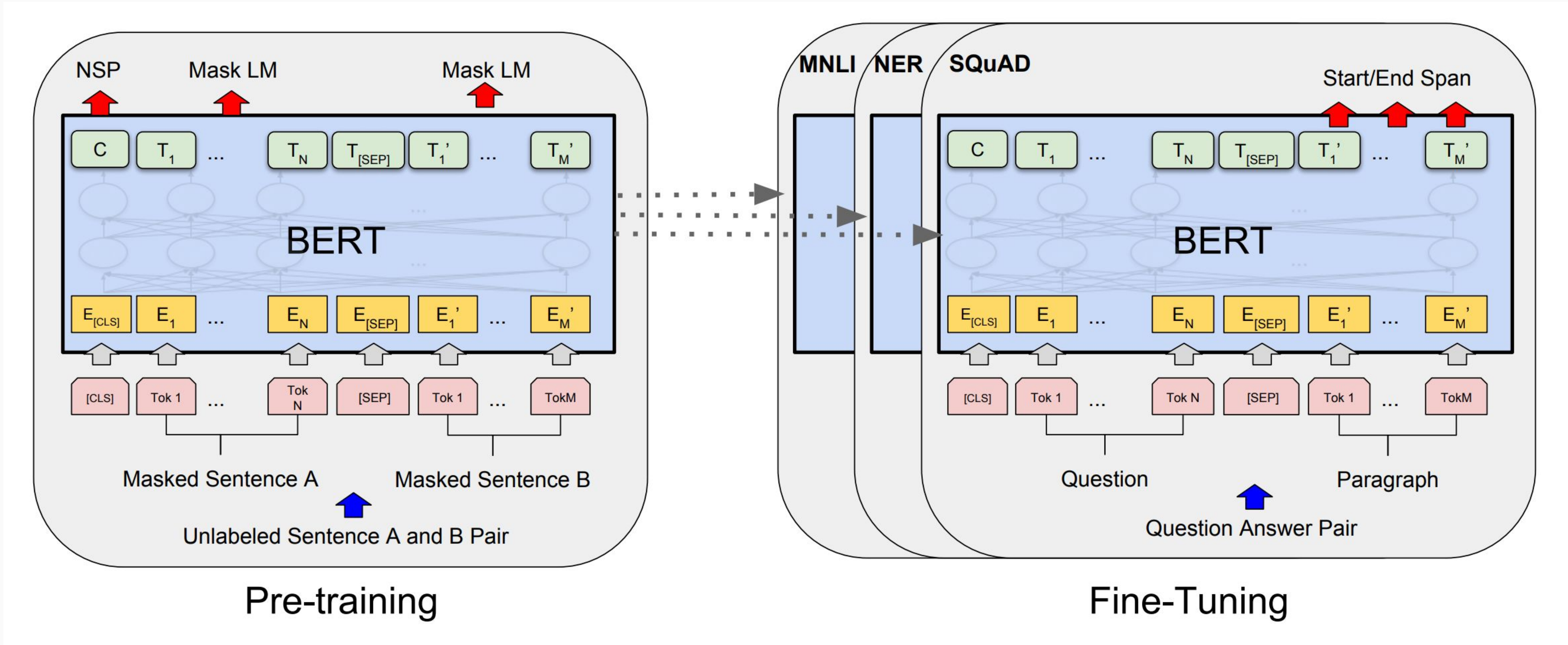




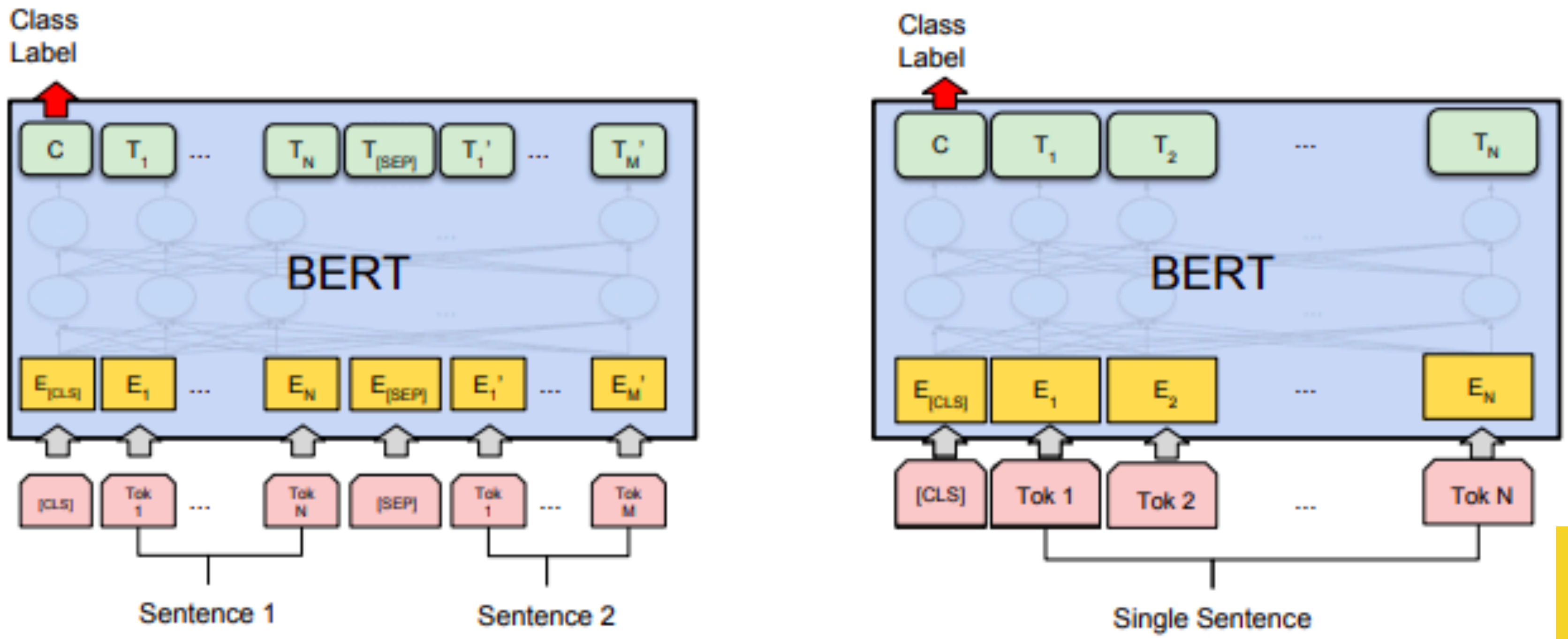
Model Details

- Data: Wikipedia (2.5B words) + BookCorpus (800M words)
- Batch Size: 131,072 words (1024 sequences * 128 length or 256 sequences * 512 length)
- Training Time: 1M steps (~40 epochs)
- Optimizer: AdamW, 1e-4 learning rate, linear decay
- `BERT-Base`: 12-layer, 768-hidden, 12-head
- `BERT-Large`: 24-layer, 1024-hidden, 16-head
- Trained on 4x4 or 8x8 TPU slice for 4 days

Fine-Tuning Procedure



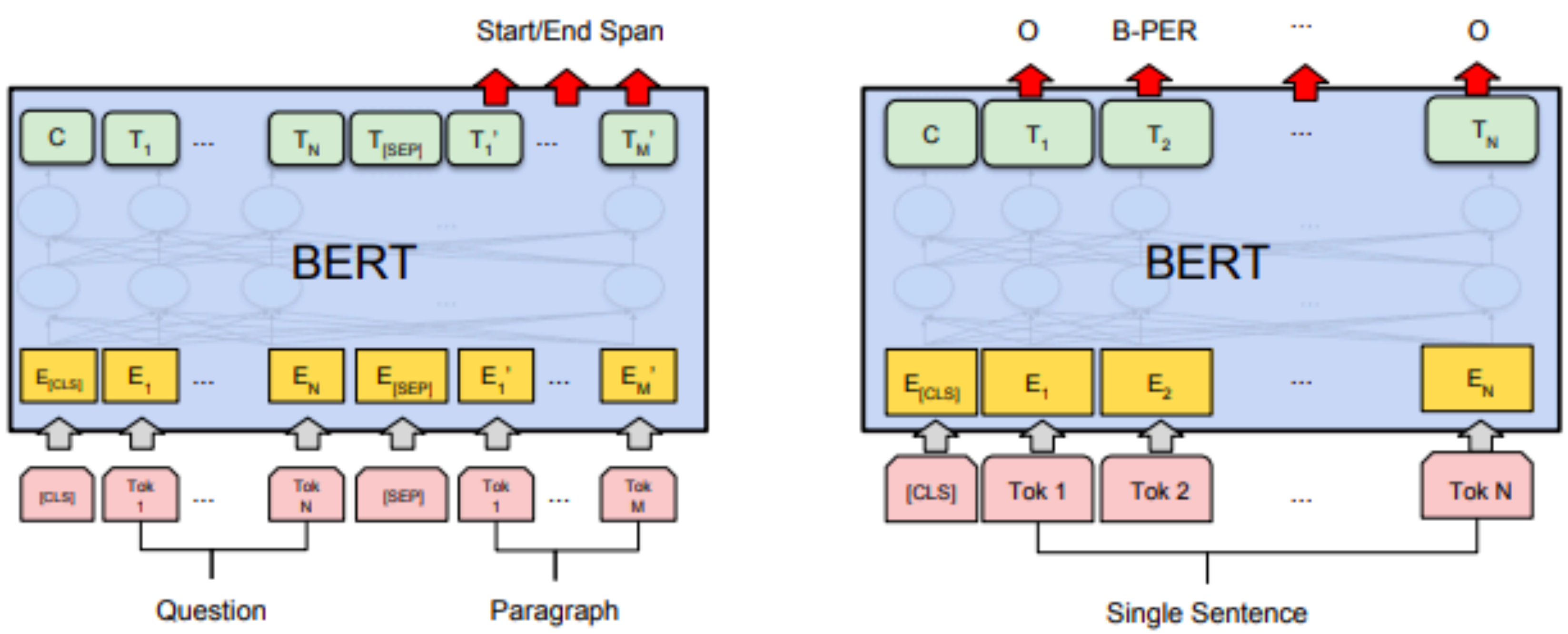
Fine-Tuning Procedure



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

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

More details next week!



(c) Question Answering Tasks: SQuAD v1.1

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

GLUE Results

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

MultiNLI

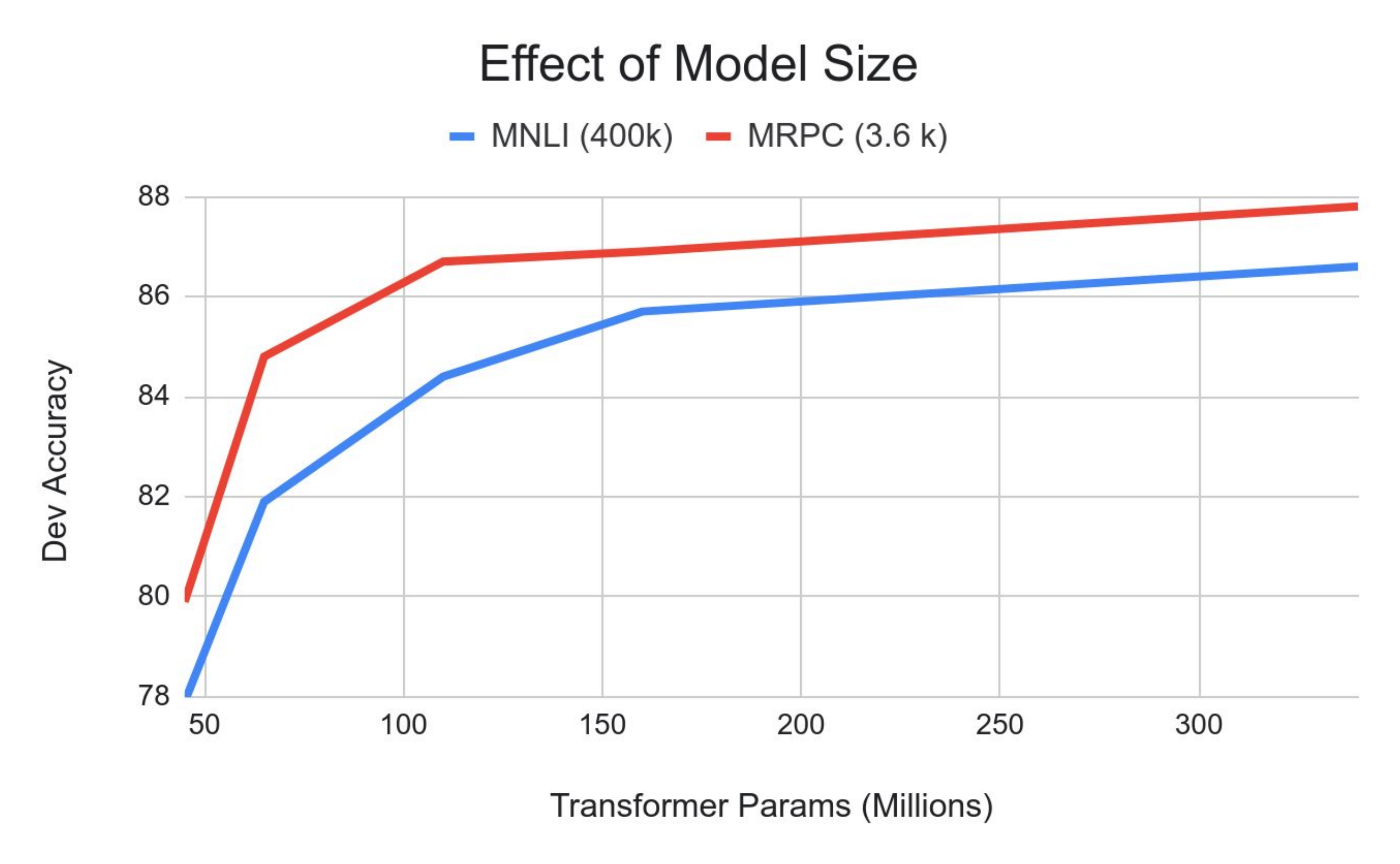
Premise: Hills and mountains are especially sanctified in Jainism.
Hypothesis: Jainism hates nature.
Label: Contradiction

CoLa

Sentence: The wagon rumbled down the road.
Label: Acceptable

Sentence: The car honked down the road.
Label: Unacceptable

Effect of Model Size



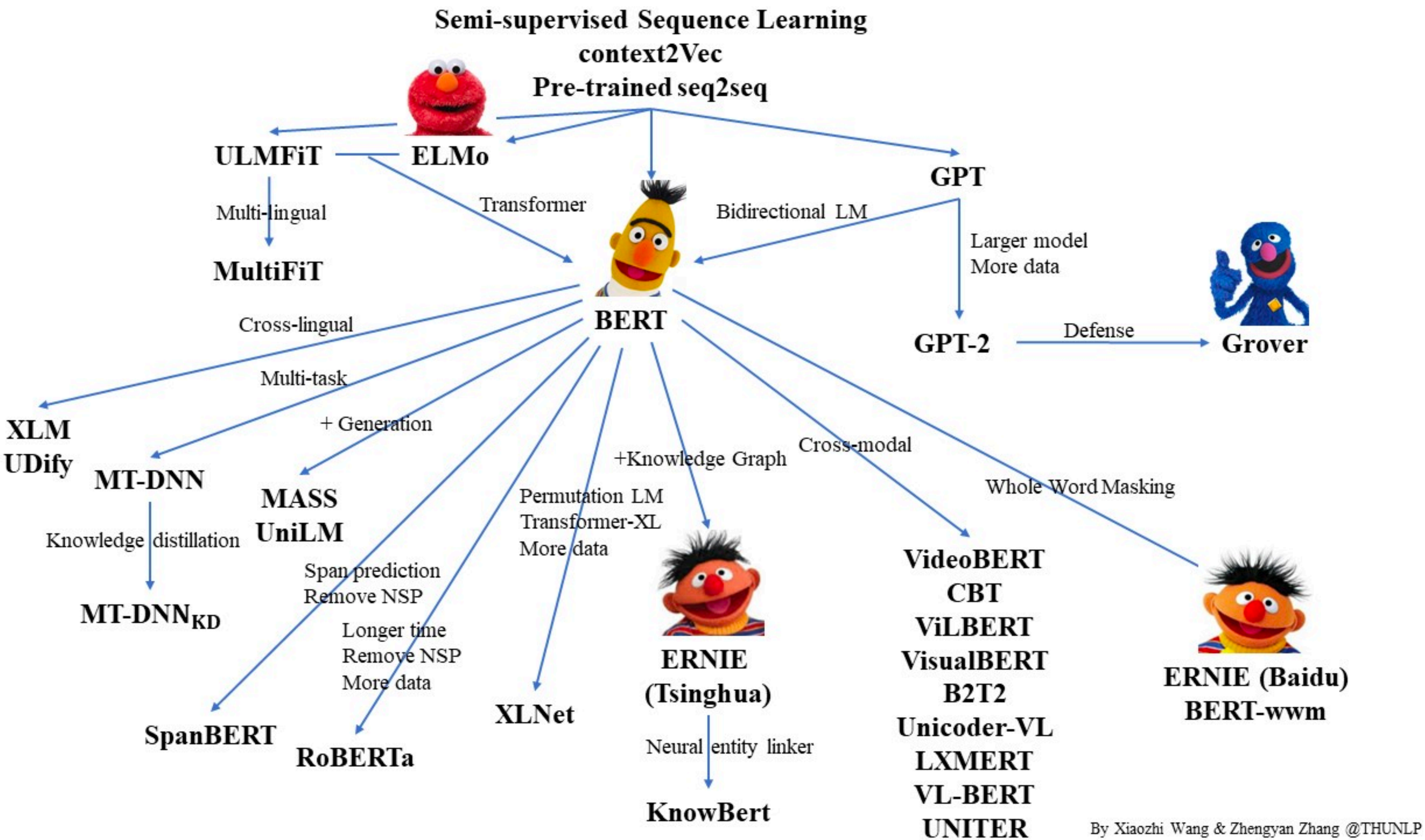
- Big models help *a lot*
- Going from 110M -> 340M params helps even on datasets with 3,600 labeled examples
- Improvements have *not* asymptoted

Multilingual BERT

- Trained single model on 104 languages from Wikipedia. Shared 110k WordPiece vocabulary.

System	English	Chinese	Spanish
XNLI Baseline - Translate Train	73.7	67.0	68.8
XNLI Baseline - Translate Test	73.7	68.4	70.7
BERT - Translate Train	81.9	76.6	77.8
BERT - Translate Test	81.9	70.1	74.9
BERT - Zero Shot	81.9	63.8	74.3

- XNLI is MultiNLI translated into multiple languages.
- Always evaluate on human-translated Test.
- Translate Train: MT English Train into Foreign, then fine-tune.
- Translate Test: MT Foreign Test into English, use English model.
- Zero Shot: Use Foreign test on English model.


Semi-supervised Sequence Learning
context2Vec
Pre-trained seq2seq
ULMFiT
ELMo
GPT
Multi-lingual
MultiFiT
Transformer
Bidirectional LM
Larger model
More data
BERT
GPT-2
Defense
Grover
Cross-lingual
Multi-task
XLM
UDify
MT-DNN
+ Generation
MASS
UniLM
Knowledge distillation
MT-DNN$_{KD}$
Span prediction
Remove NSP
Longer time
Remove NSP
More data
SpanBERT
RoBERTa
Permutation LM
Transformer-XL
More data
XLNet
+Knowledge Graph
ERNIE
(Tsinghua)
Neural entity linker
KnowBert
Cross-modal
VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMERT
VL-BERT
UNITER
Whole Word Masking
ERNIE (Baidu)
BERT-wwm
By Xiaozhi Wang & Zhengyan Zhang @THUNLP

Common Questions

- Why did no one think of this before?
- Better question: Why wasn't contextual pre-training popular before 2018 with ELMo?
- Good results on pre-training is >1,000x to 100,000 more expensive than supervised training.
  - E.g., 10x-100x bigger model trained for 100x-1,000x as many steps.
  - Imagine it's 2013: Well-tuned 2-layer, 512-dim LSTM sentiment analysis gets 80% accuracy, training for 8 hours.
  - Pre-train LM on same architecture for a week, get 80.5%.
  - Conference reviewers: "Who would do something so expensive for such a small gain?"

Common Questions

- The model must be learning more than “contextual embeddings”
- Alternate interpretation: Predicting missing words (or next words) requires learning many types of language understanding features.
  - syntax, semantics, pragmatics, coreference, etc.
- Implication: Pre-trained model is much bigger than it needs to be to solve specific task
- Task-specific model distillation words very well

UMass · CS685 | Advanced Natural Language Processing (2020)

CS685 (2020)· 课程资料包 @ShowMeAI

视频
中英双语字幕

课件
一键打包下载

笔记
官方笔记翻译

代码
作业项目解析

视频 · B 站 [扫码或点击链接]
https://www.bilibili.com/video/BV1BL411t7RV

课件 & 代码 · 博客 [扫码或点击链接]
http://blog.showmeai.tech/umass-cs685

NLP 语言模型 问答系统 文本生成 BERT 语义解析 GPT-3 知识推理 模型蒸馏 迁移学习 transformer 注意力机制

Awesome AI Courses Notes Cheatsheets 是 ShowMeAI 资料库的分支系列，覆盖最具知名度的 **TOP50+** 门 AI 课程，旨在为读者和学习者提供一整套高品质中文学习笔记和速查表。

点击课程名称，跳转至课程**资料包**页面，**一键下载**课程全部资料！

机器学习	深度学习	自然语言处理	计算机视觉
Stanford · CS229	Stanford · CS230	Stanford · CS224n	Stanford · CS231n
# Awesome AI Courses Notes Cheatsheets· 持续更新中			
知识图谱	图机器学习	深度强化学习	自动驾驶
Stanford · CS520	Stanford · CS224W	UCBerkeley · CS285	MIT · 6.S094

微信公众号

资料下载方式 2：扫码点击底部菜单栏

称为 **AI 内容创作者?** 回复 [添砖加瓦]