CSCE 638 Natural Language Processing Foundation and Techniques

Lecture 9: Contextualized Representations and Pre-Training

Kuan-Hao Huang

Spring 2025

(Some slides adapted from Chris Manning, Karthik Narasimhan, and Danqi Chen)

Quiz 1

- Date: 2/17
  - 10 minutes before the end of the lecture
  - 5 questions focusing on high-level concepts

Week	Date		Topic
W1	1/13	L1	Course Overview [slides]
	1/15	L2	Text Classification [slides]
W2	1/20		Martin Luther King, Jr. Day (No Class)
	1/22	L3	Word Representations [slides]
W3	1/27	L4	Word Representations, Tokenization, Language Modeling [slides]
	1/29	L5	Convolutional Neural Network, Recurrent Neural Network [slides]
W4	2/3	L6	Sequential Labeling, Sequence-to-Sequence, Attention
	2/5	L7	Transformers

Assignment 1

- Due: 2/17 11:59pm
- Small modification
  - Problem 5.7

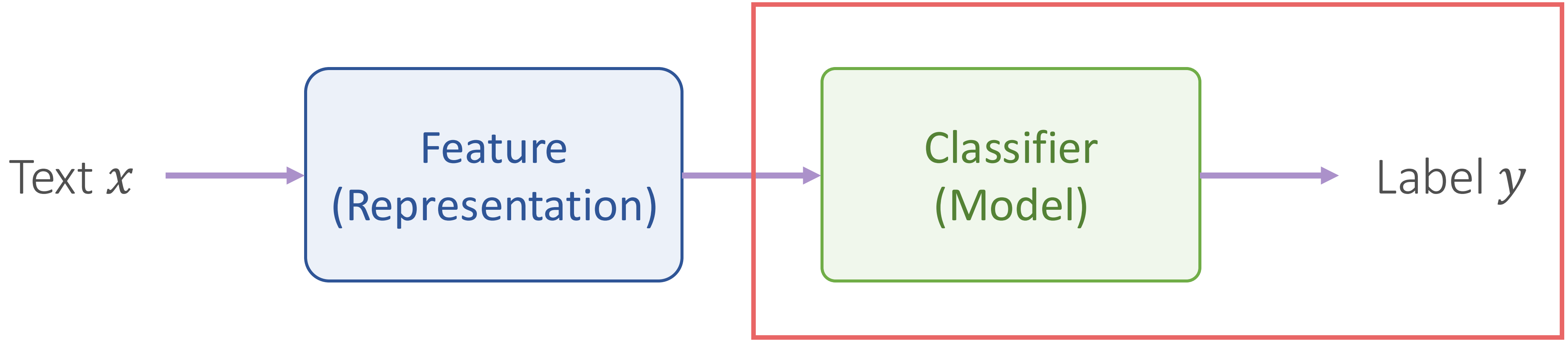
```python
epochs = 100
best_valid_acc = 0.0
for epoch in range(epochs):
    model.train()
    total_loss = 0
    for texts, labels in loader_train:

        ### ========= TODO : START ========= ###

        ### ========= TODO :  END  ========= ###

        valid_acc = evaluate_acc(model, loader_valid)
        if valid_acc > best_valid_acc:
            best_valid_acc = valid_acc
            torch.save(model.state_dict(), model_path)

    print(f"Epoch [{epoch+1}/{epochs}], Loss: {total_loss / len(loader_train)}, Valid Acc: {valid_acc}")
```

```python
epochs = 100
best_valid_acc = 0.0
for epoch in range(epochs):
    model.train()
    total_loss = 0
    for texts, labels in loader_train:

        ### ========= TODO : START ========= ###

        ### ========= TODO :  END  ========= ###

    valid_acc = evaluate_acc(model, loader_valid)
    if valid_acc > best_valid_acc:
        best_valid_acc = valid_acc
        torch.save(model.state_dict(), model_path)

    print(f"Epoch [{epoch+1}/{epochs}], Loss: {total_loss / len(loader_train)}, Valid Acc: {valid_acc}")
```

Lecture Plan

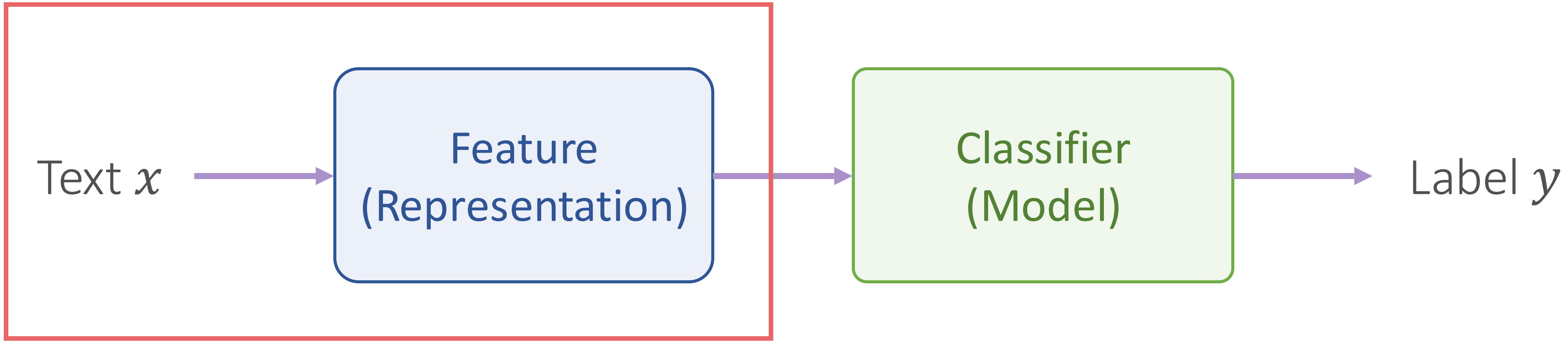
- Contextualized Representations
  - ELMo
- Pre-Training
  - Encoder-Only Pre-Training
  - Encoder-Decoder Pre-Training
  - Decoder-Only Pre-Training
- Model Distillation

A General Framework for Text Classification



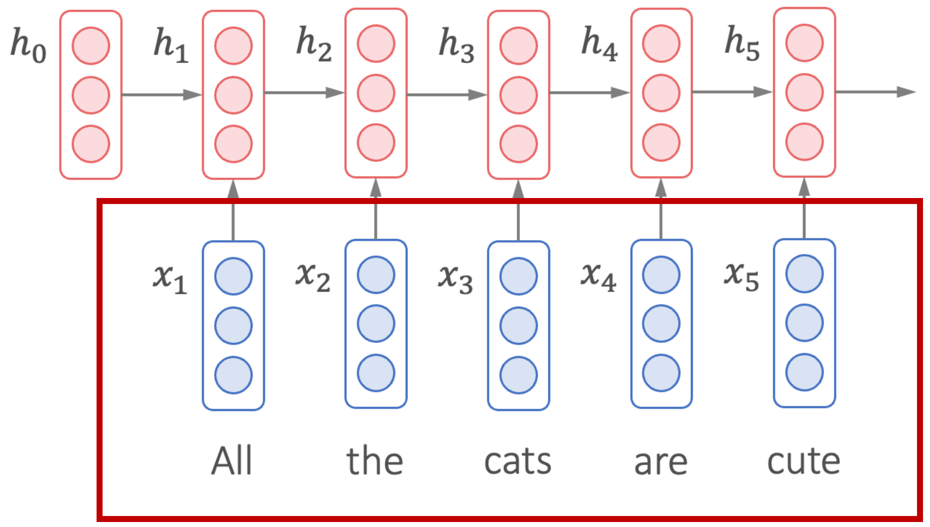
- Teach the model how to make prediction y
- Logistic regression, neural networks, CNN, RNN, LSTM, Transformers

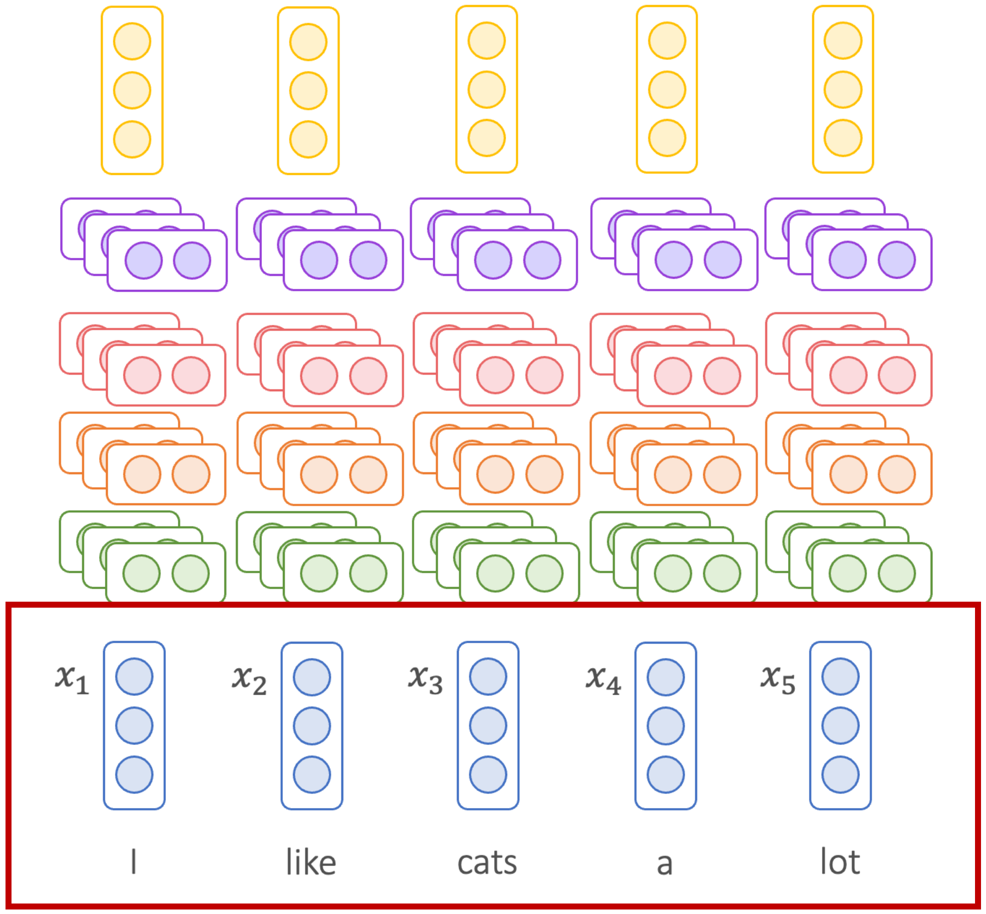
A General Framework for Text Classification



- Teach the model how to understand example x
- Rule-based representations
  - Bag-of-words, n-grams
- Learnable representations
  - Word2Vec (Skip-Gram and CBOW), GloVe, FastText

Static Word Embeddings





Static Word Embeddings

- One vector for each word type
- How about words with multiple meanings?

mouse^1 : .... a *mouse* controlling a computer system in 1968.
mouse^2 : .... a quiet animal like a *mouse*
bank^1 : ...a *bank* can hold the investments in a custodial account ...
bank^2 : ...as agriculture burgeons on the east *bank*, the river ...

Contextualized Word Embeddings

- The embeddings of a word should be conditioned on its context

Distributional hypothesis: words that occur in similar contexts tend to have similar meanings

J.R.Firth 1957

- "You shall know a word by the company it keeps"
- One of the most successful ideas of modern statistical NLP!

…government debt problems turning into	***banking***	*crises as happened in 2009…*
…saying that Europe needs unified	***banking***	*regulation to replace the hodgepodge…*
…India has just given its	***banking***	*system a shot in the arm…*

Contextualized Word Embeddings

- Chico Ruiz made a spectacular play on Alusik's grounder …
- Olivia De Havilland signed to do a Broadway play for Garson …
- Kieffer was commended for his ability to hit in the clutch , as well as his all-round excellent play …
- … they were actors who had been handed fat roles in a successful play …
- Concepts play an important role in all aspects of cognition …

ELMo: Embeddings from Language Models

Deep contextualized word representations

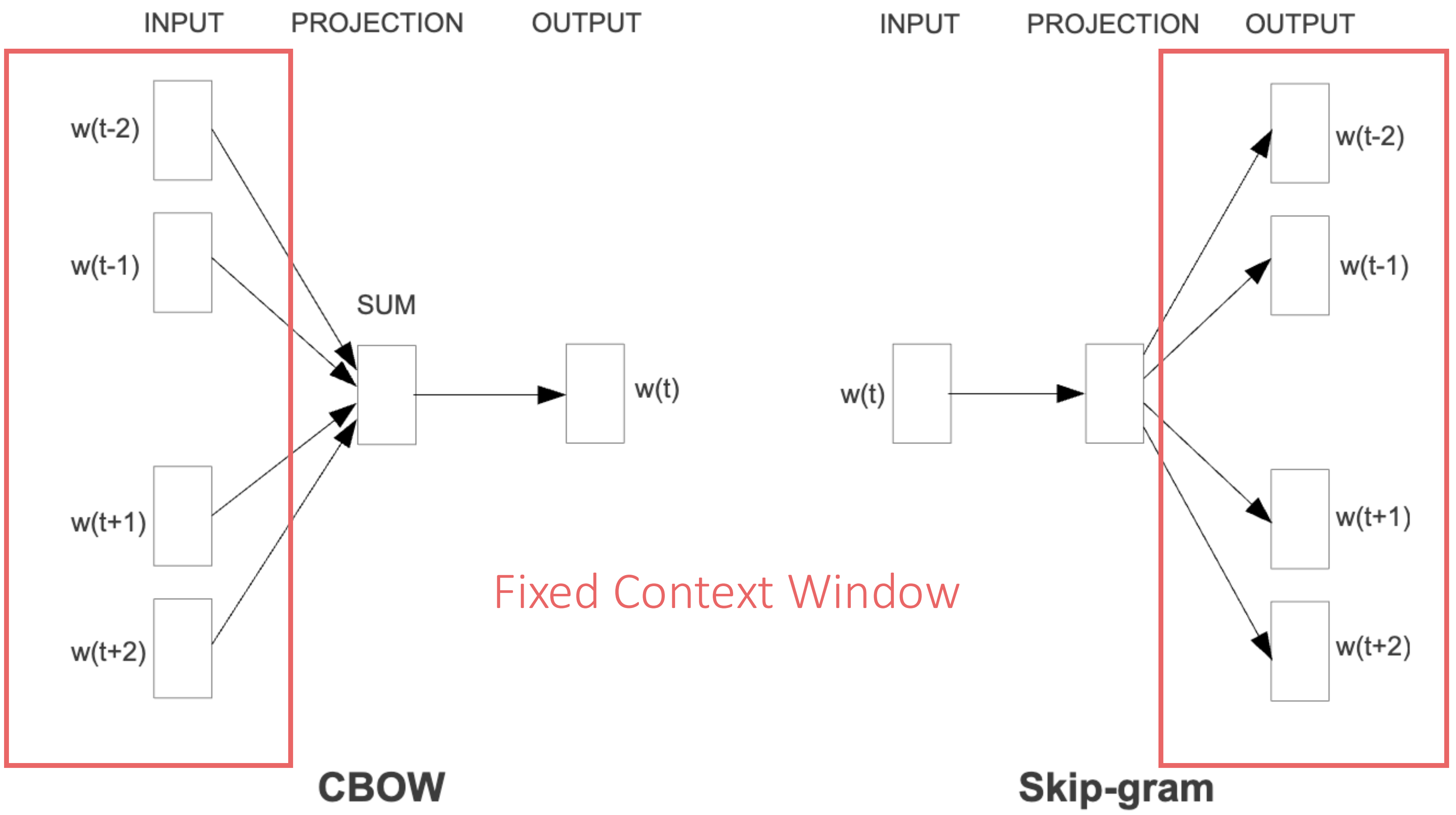
Matthew E. Peters[†], **Mark Neumann**[†], **Mohit Iyyer**[†], **Matt Gardner**[†],
{matthewp,markn,mohiti,mattg}@allenai.org

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

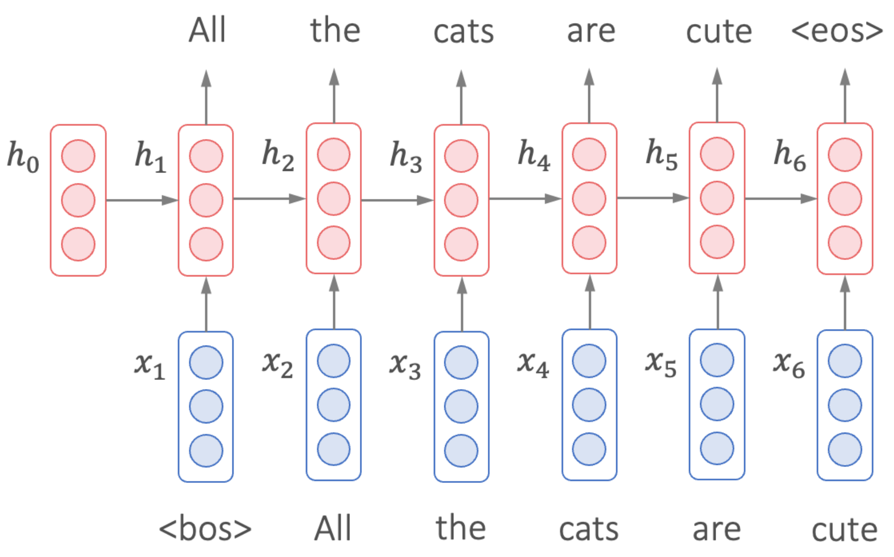
[†]Allen Institute for Artificial Intelligence
[*]Paul G. Allen School of Computer Science & Engineering, University of Washington

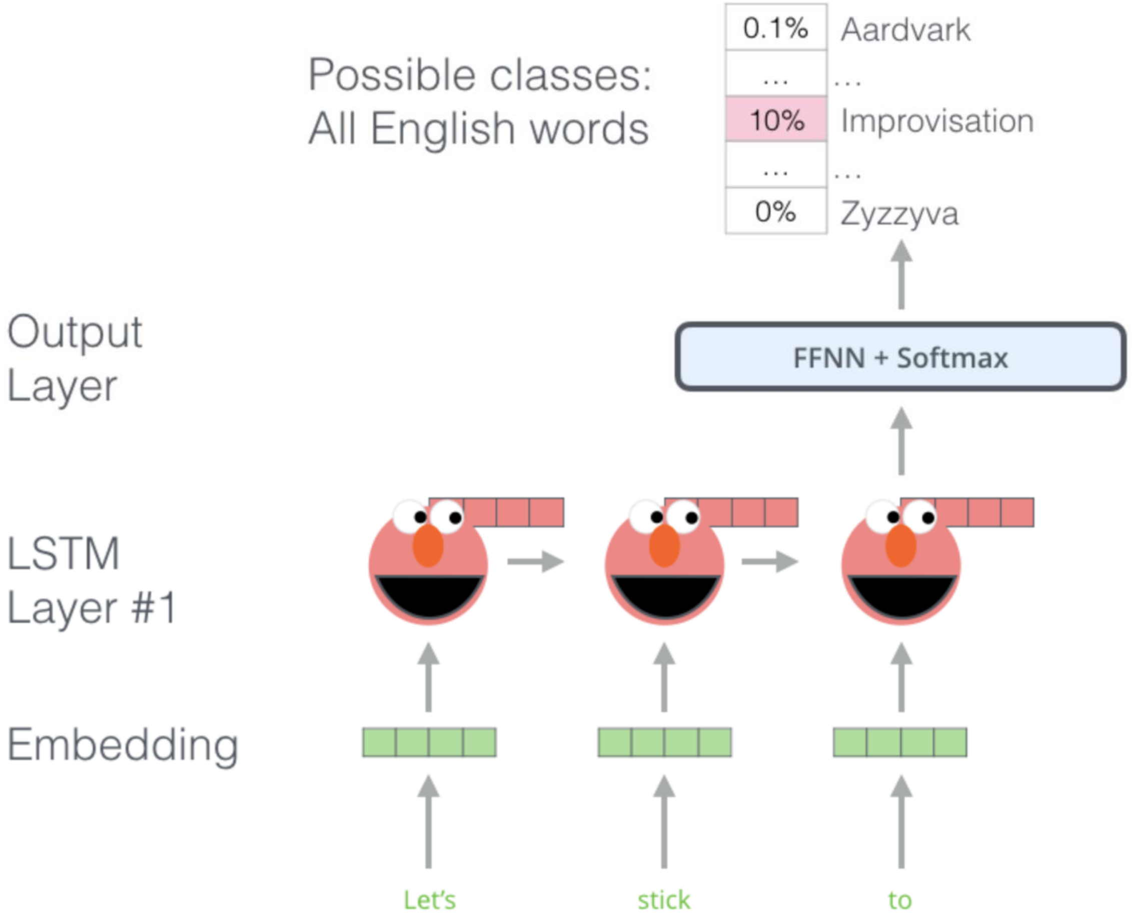
Deep contextualized word representations, 2018

Recap: Continuous Bag of Words (CBOW) and Skip-Grams



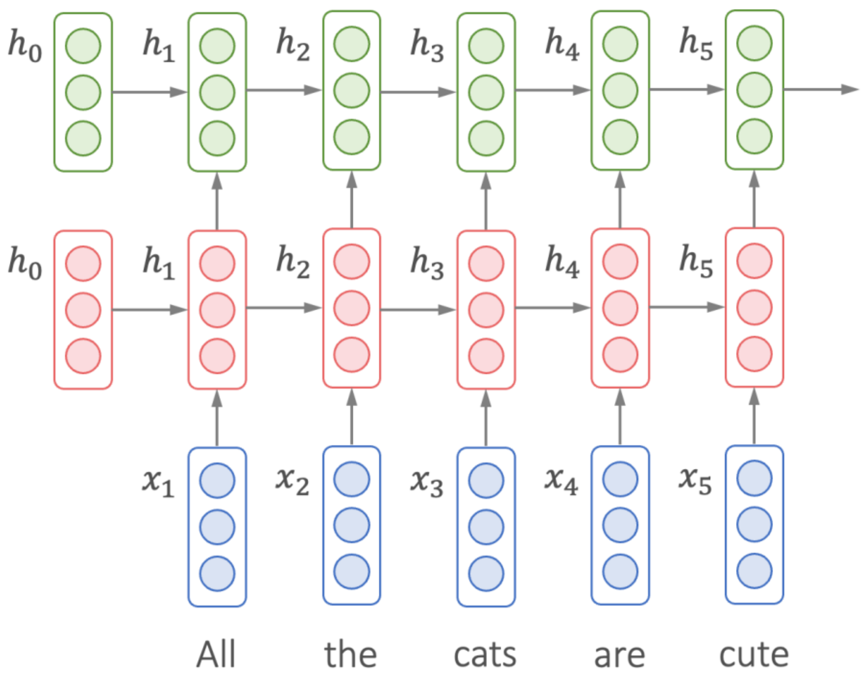
ELMo: Language Modeling

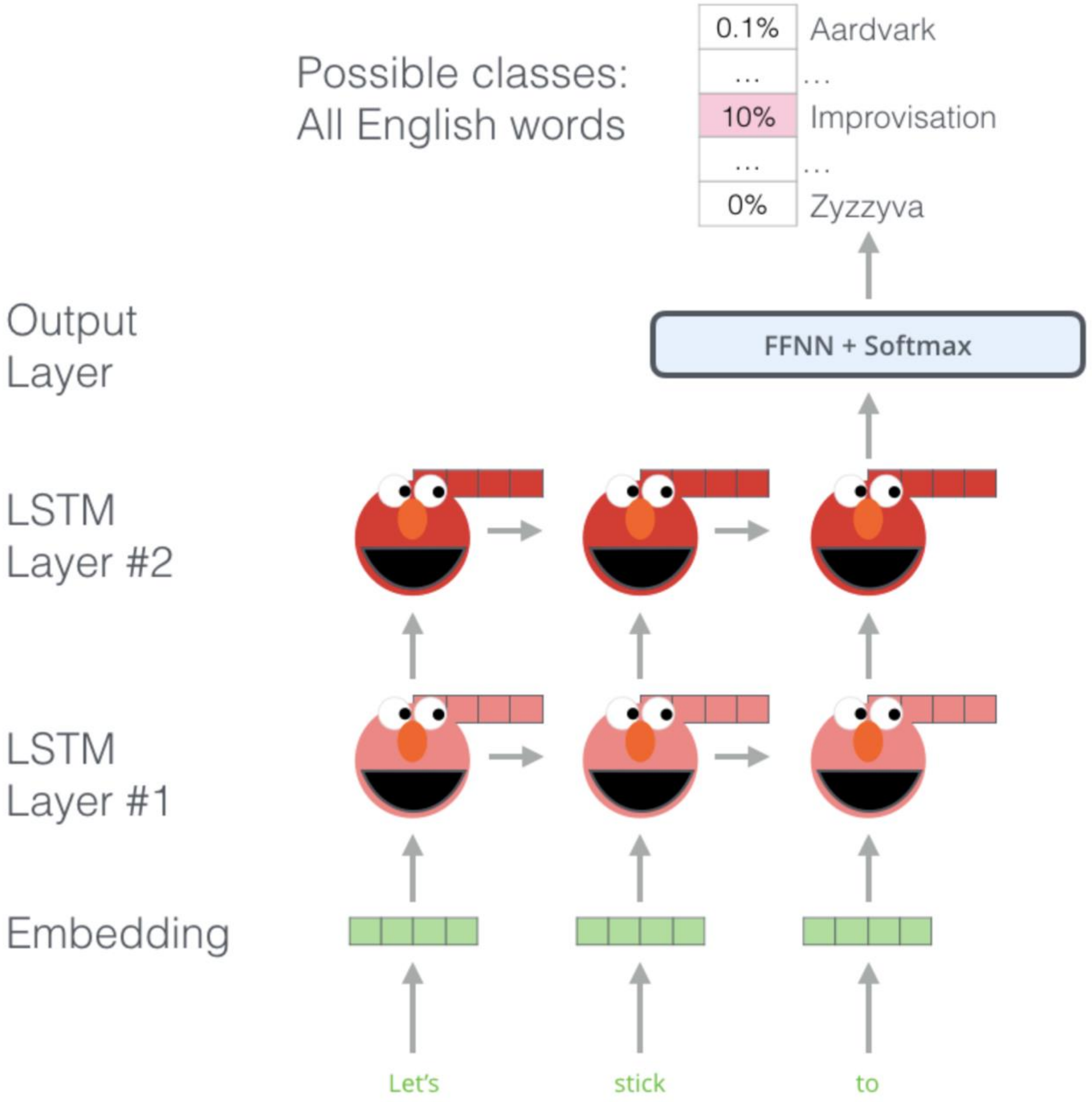




http://jalammar.github.io/illustrated-bert

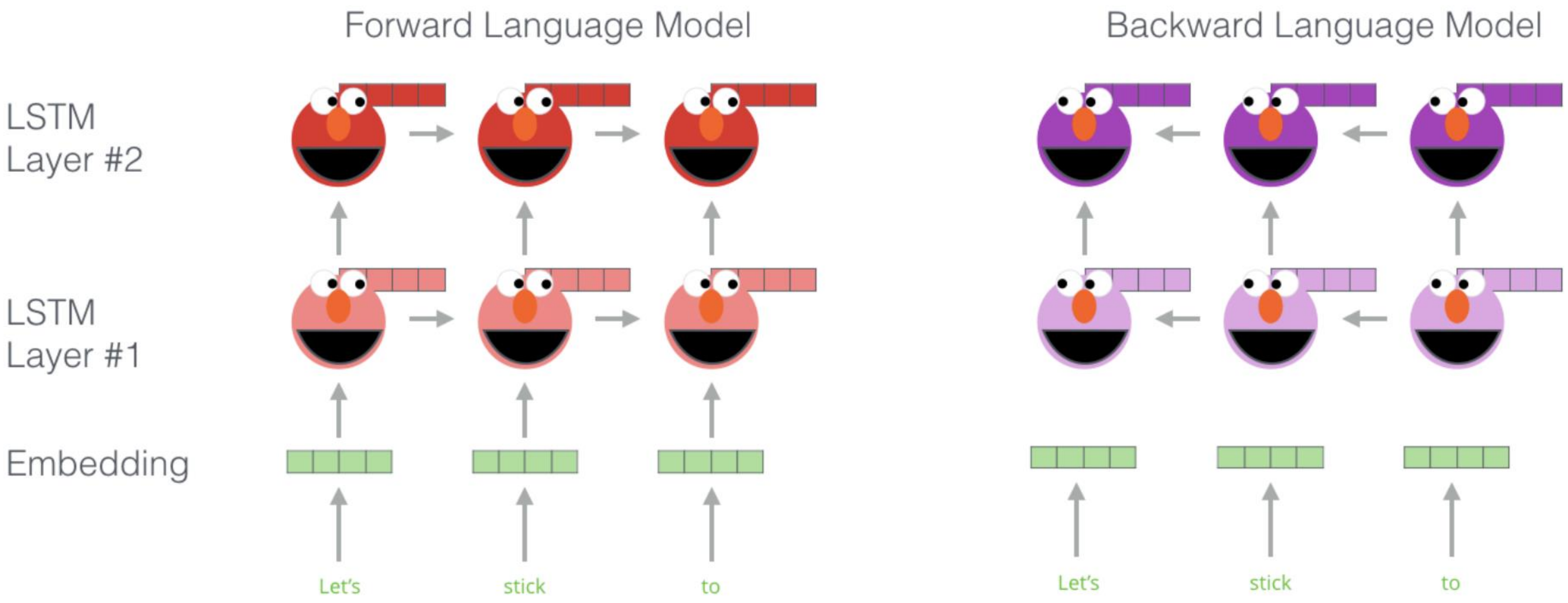
ELMo: Language Modeling with Stacked LSTM





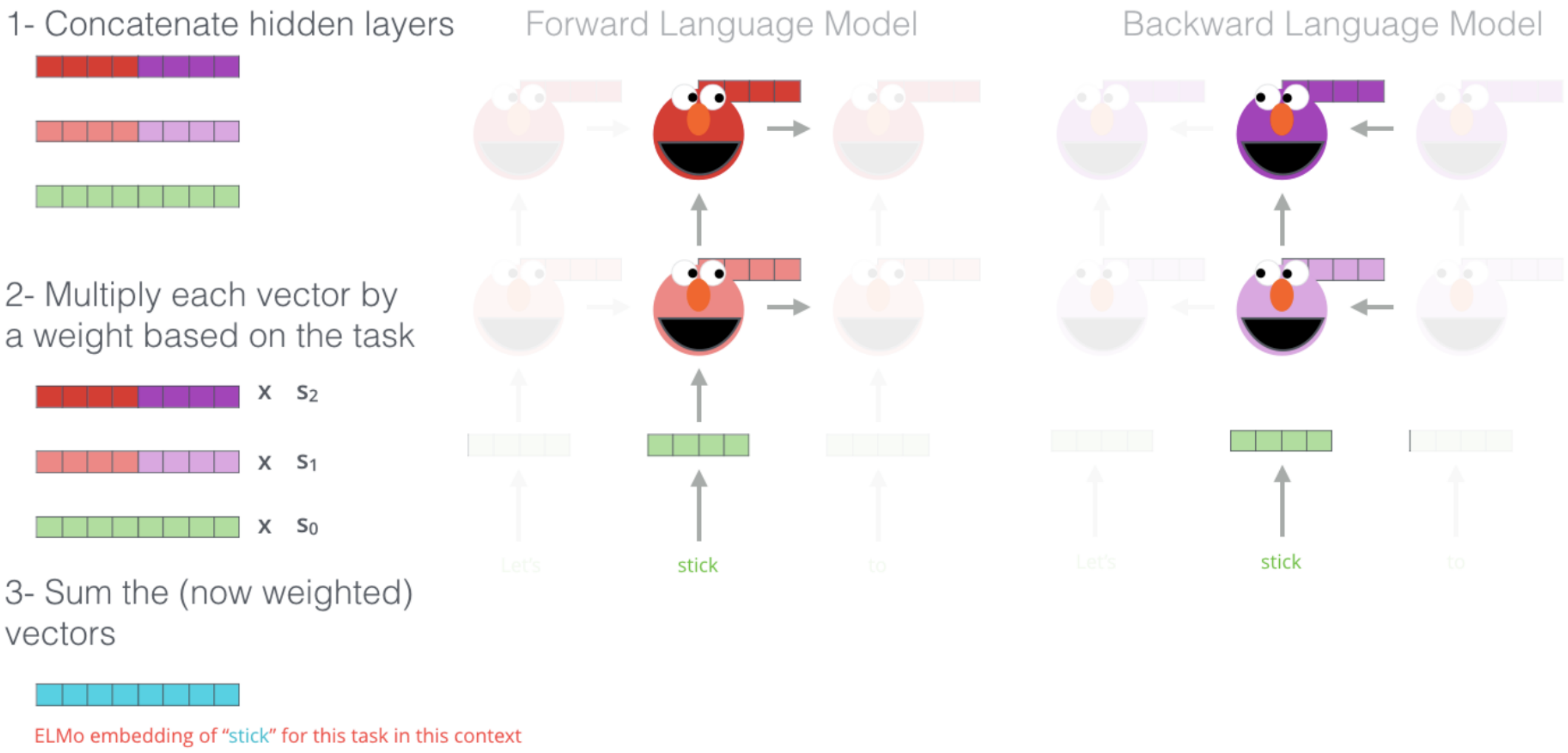
http://jalammar.github.io/illustrated-bert

ELMo: Bi-Directional Language Modeling



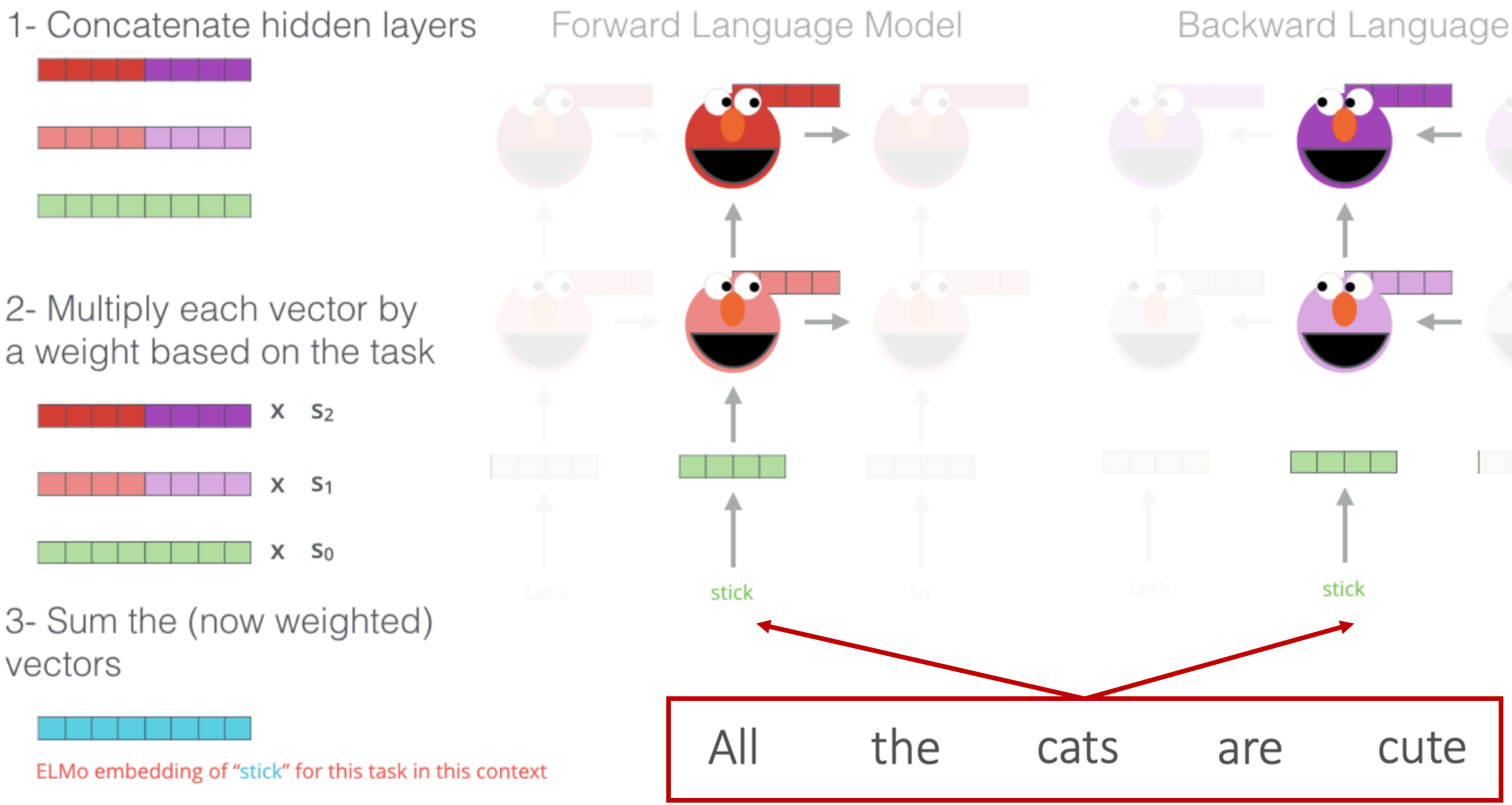
http://jalammar.github.io/illustrated-bert

ELMo: Contextualized Word Embeddings



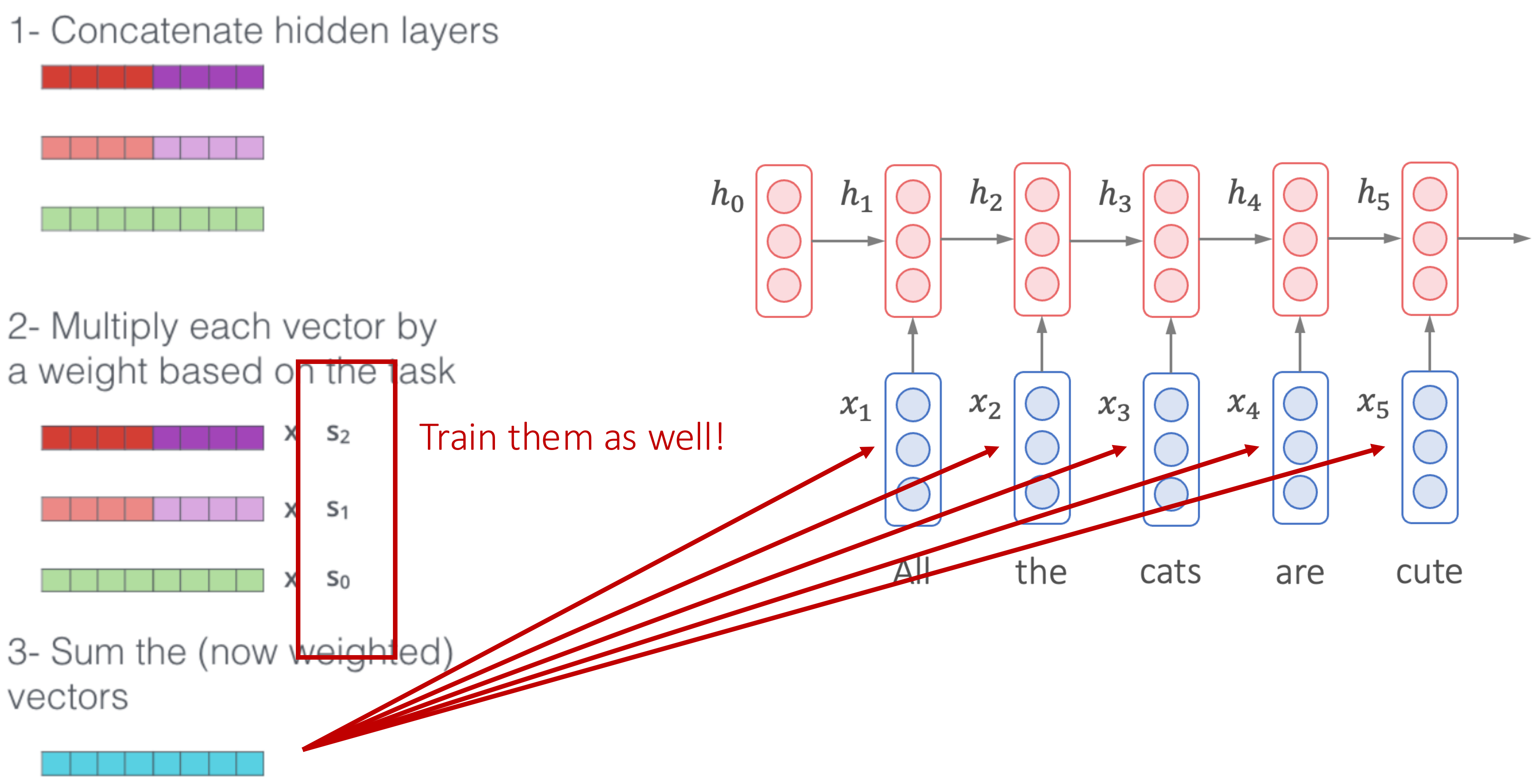
http://jalammar.github.io/illustrated-bert

How to Use ELMo?



http://jalammar.github.io/illustrated-bert

How to Use ELMo?

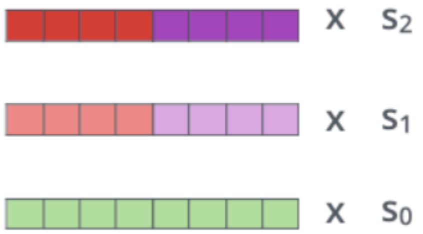


http://jalammar.github.io/illustrated-bert

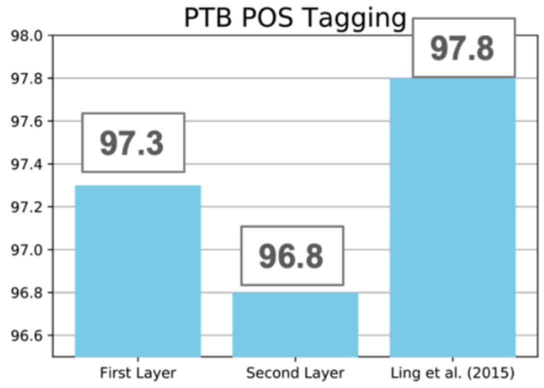
Task-Specific Weights

1- Concatenate hidden layers

2- Multiply each vector by a weight based on the task

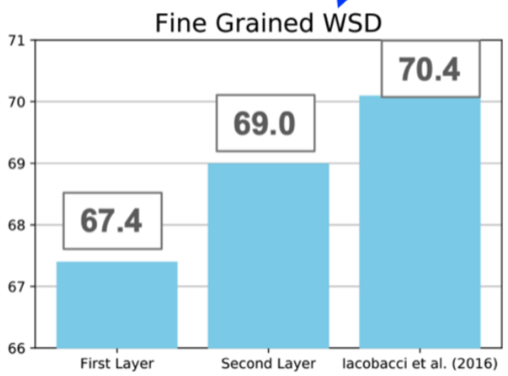


3- Sum the (now weighted) vectors



first layer > second layer

WSD = word sense disambiguation



second layer > first layer

http://jalammar.github.io/illustrated-bert

Nearest Neighbor in Embedding Space

	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 .

ELMo Performance

Task	Previous SOTA		Our baseline	ELMo + baseline
SQuAD	Liu et al. (2017)	84.4	81.1	85.8
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17
SRL	He et al. (2017)	81.7	81.4	84.6
Coref	Lee et al. (2017)	67.2	67.2	70.4
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5

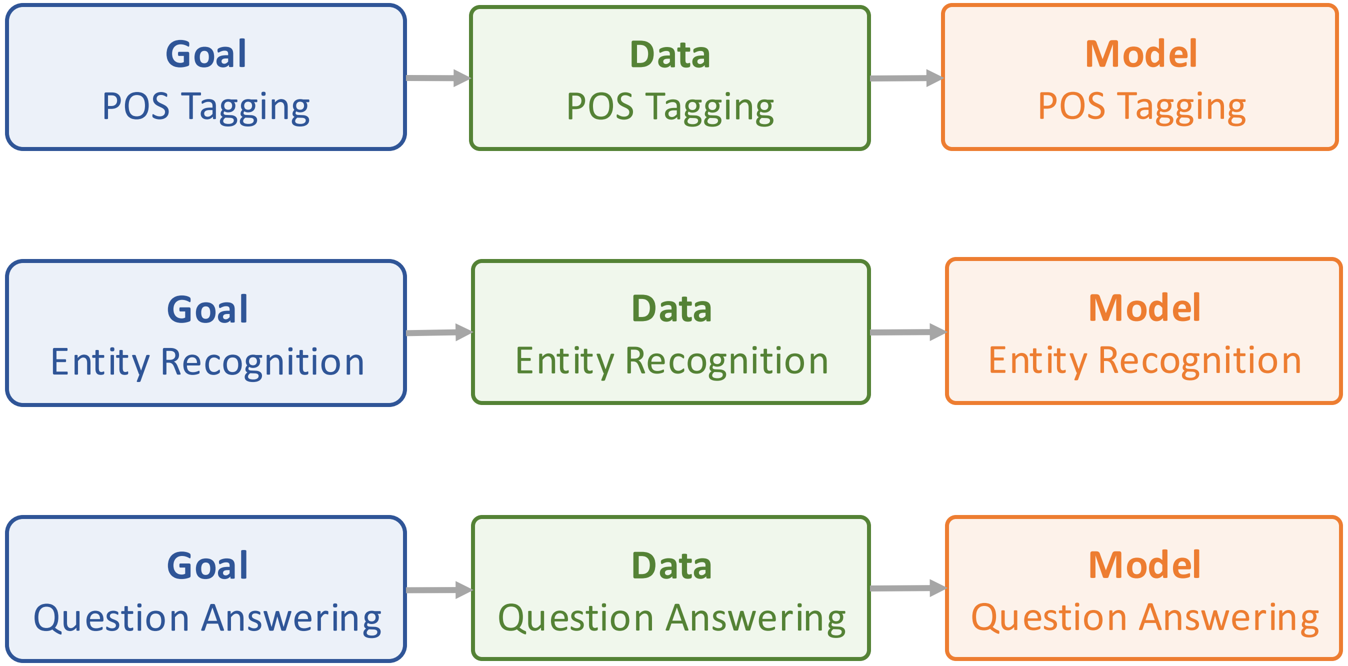
Lecture Plan

- Contextualized Representations
  - ELMo
- Pre-Training
  - Encoder-Only Pre-Training
  - Encoder-Decoder Pre-Training
  - Decoder-Only Pre-Training
- Model Distillation

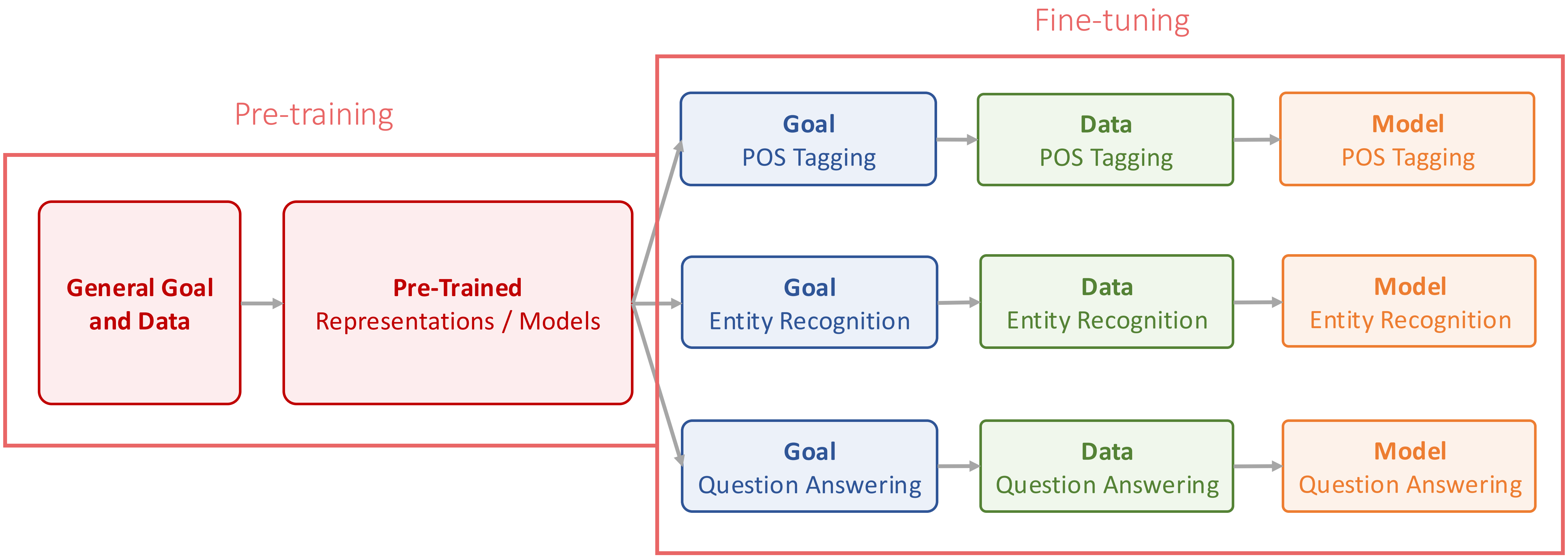
Pre-Training

- Pre-training and fine-tuning
  - First, pre-train a model on a large dataset for task X
  - Them, fine-tune the same on a dataset for task Y
- If task X is somewhat related to task Y
  - Good performance on task X → It is helpful for task Y
- The objective of task X is typically self-supervised
- Word2Vec and ELMo are one kind of pre-training
  - Task X: Predicting context words / Language modeling
  - Task Y: Any downstream tasks

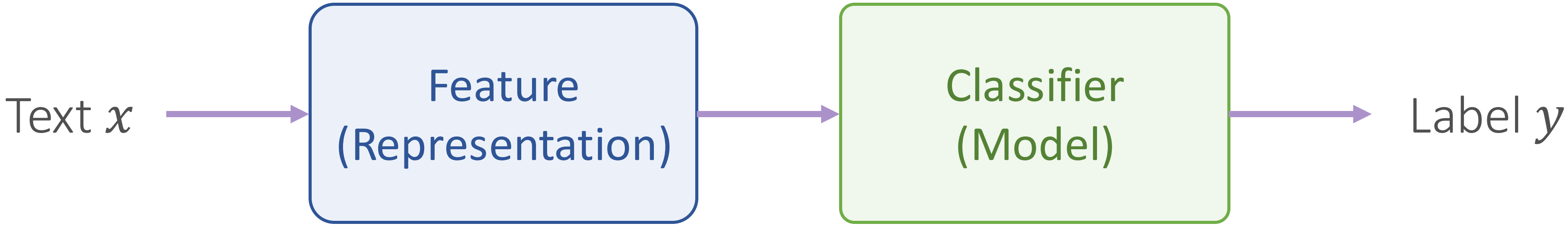
Training from Scratch



Fine-Tuning with Pre-Training

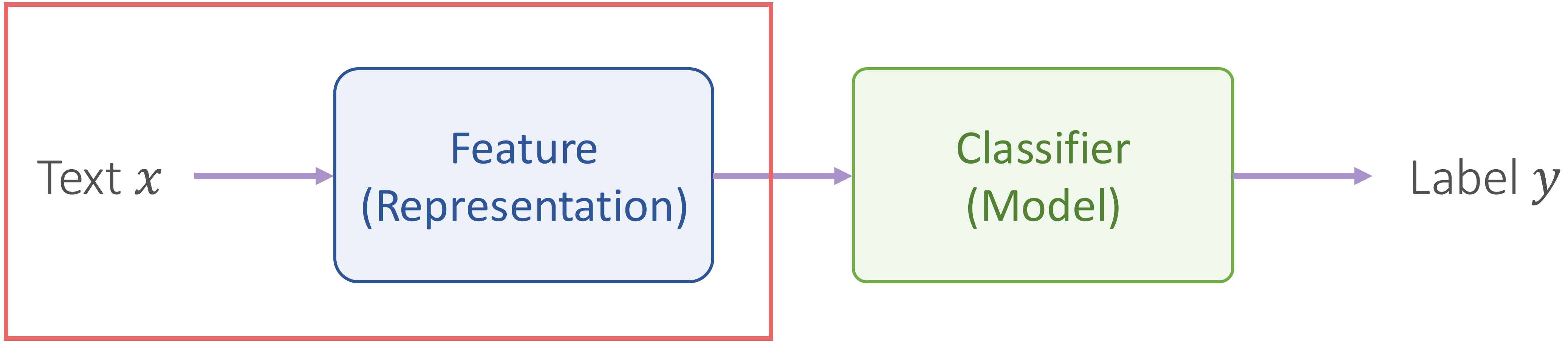


A General Framework for Text Classification



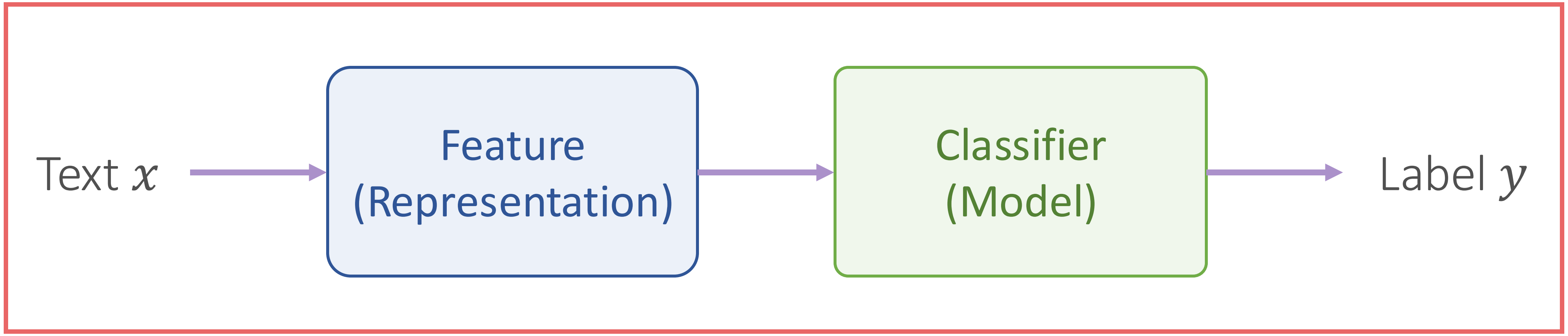
- Task-specific feature: N-gram features, TF-IDF
- Task-specific classifier: Logistic Regression, CNN, RNN, Transformers
- No pre-training

A General Framework for Text Classification



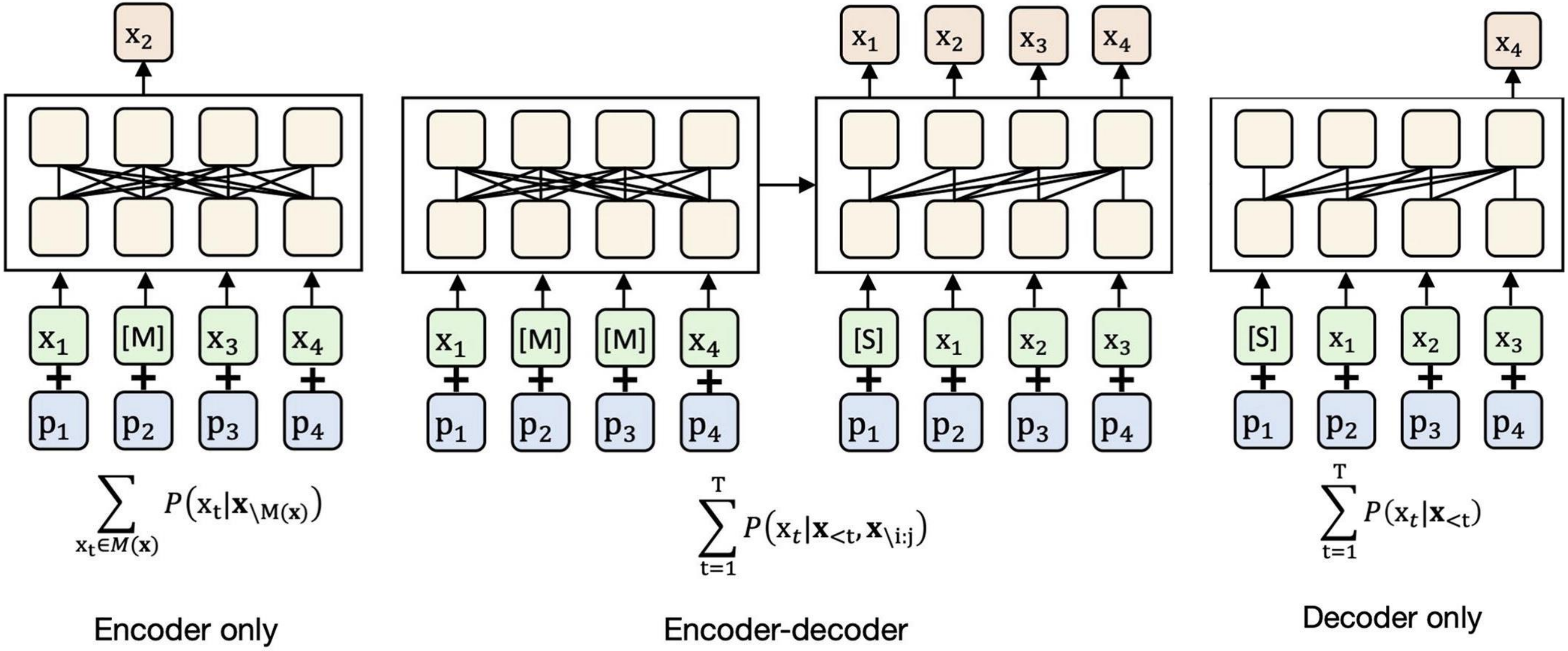
- Pre-trained feature: Word2Vec, Glove, ELMo
- Task-specific classifier: Logistic Regression, CNN, RNN, Transformers
- Pre-training on features/representations only

A General Framework for Text Classification



- Pre-training the whole pipeline
  - Pre-trained representations + pre-trained model weights
  - We only cover Transformer-based pre-training

Types of Pre-Training



https://www.sciencedirect.com/science/article/pii/S2095809922006324

Encoder-Only: BERT

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (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

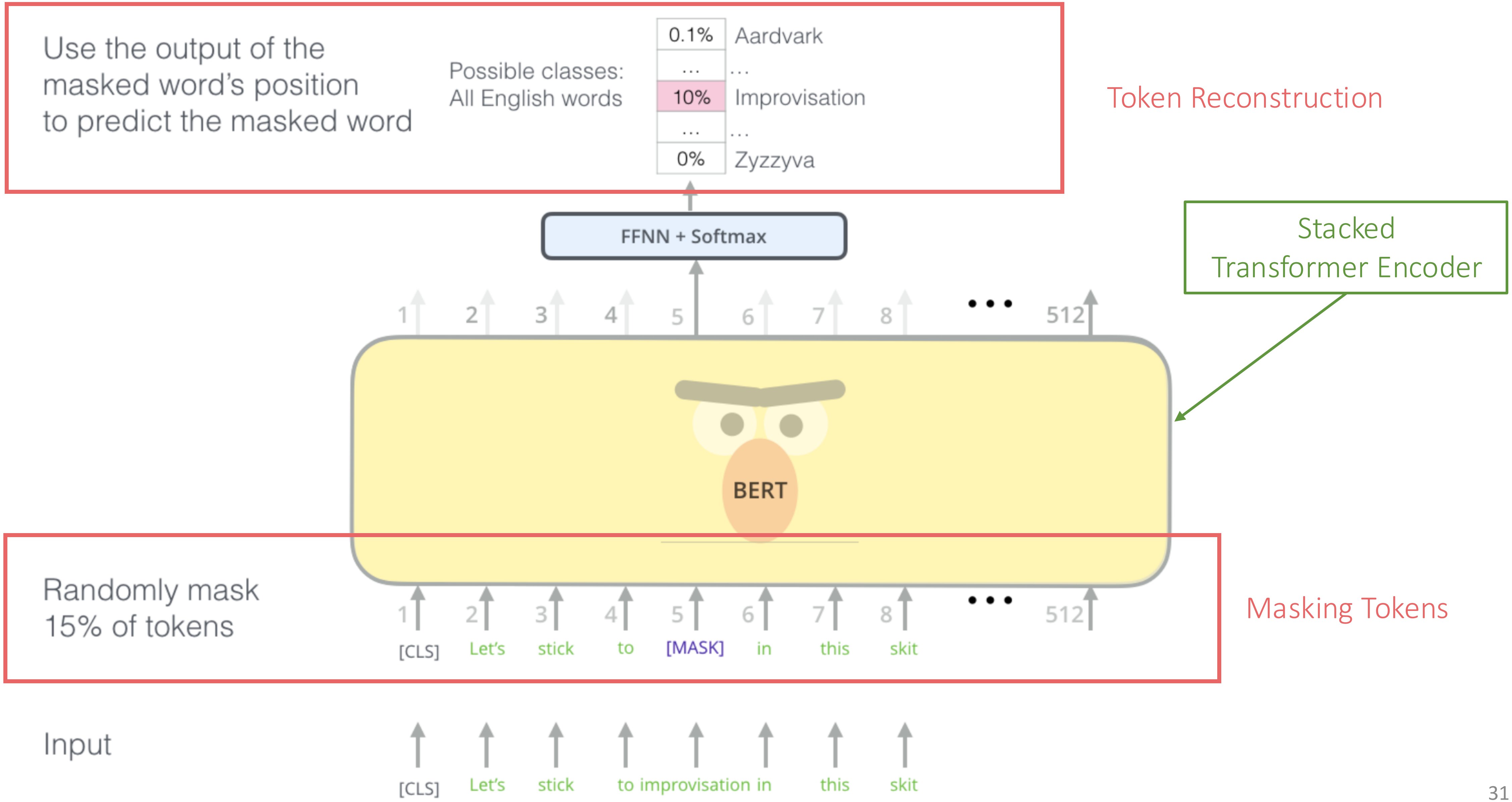
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018

Encoder-Only: BERT

- Transformer architecture
- Encoder-only
  - More about representations
  - Bi-directional
- Pre-training corpus
  - Wikipedia (2.5B tokens) + BookCorpus (0.8B tokens)
- Two self-supervised objectives
  - Masked language modeling
  - Next sentence prediction

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018

Pre-Training Task: Masked Language Modeling



http://jalammar.github.io/illustrated-bert/

Pre-Training Task: Masked Language Modeling

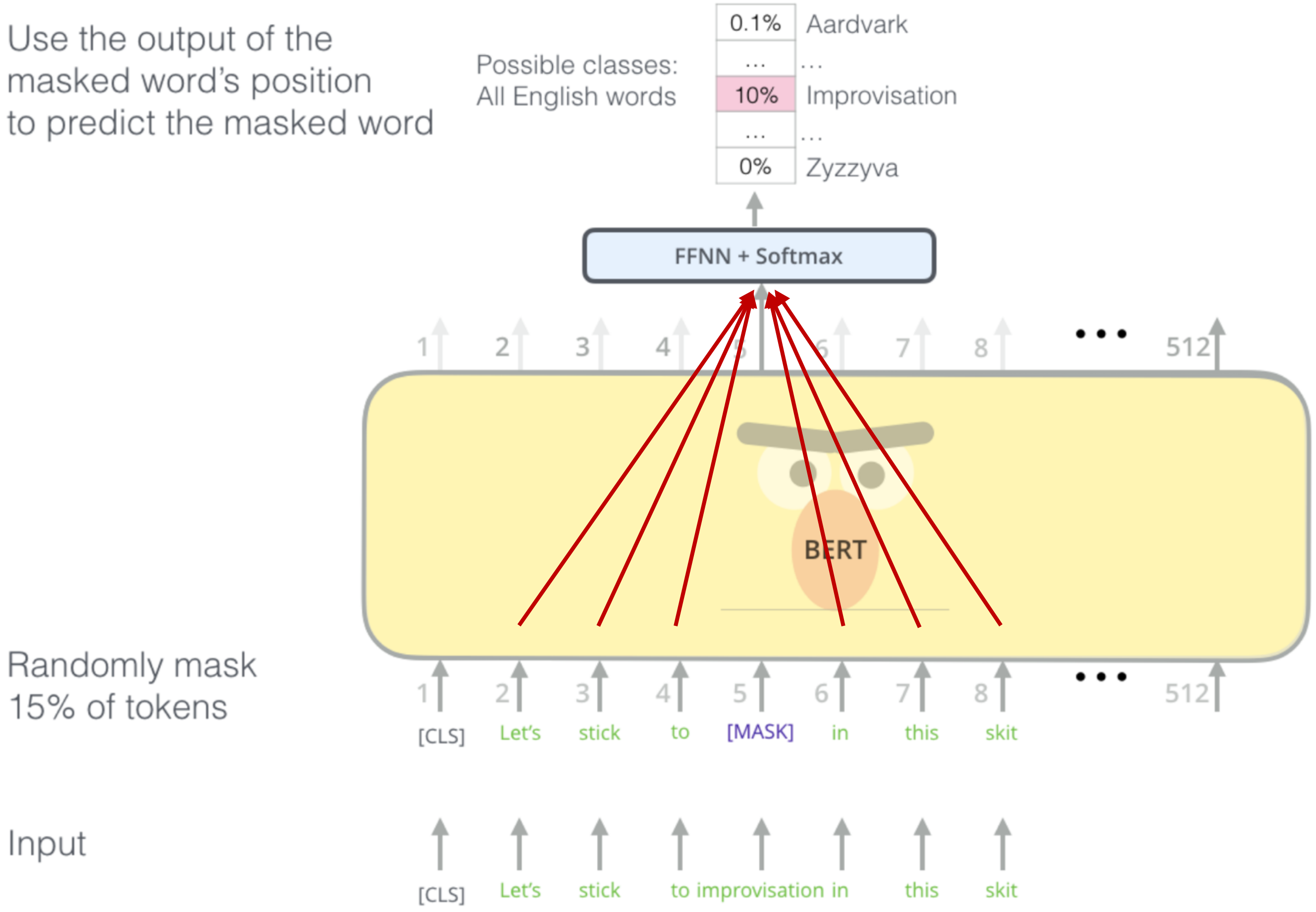
- Why is it useful?
  - Learn to aggregate information from context

Distributional hypothesis: words that occur in similar contexts tend to have similar meanings

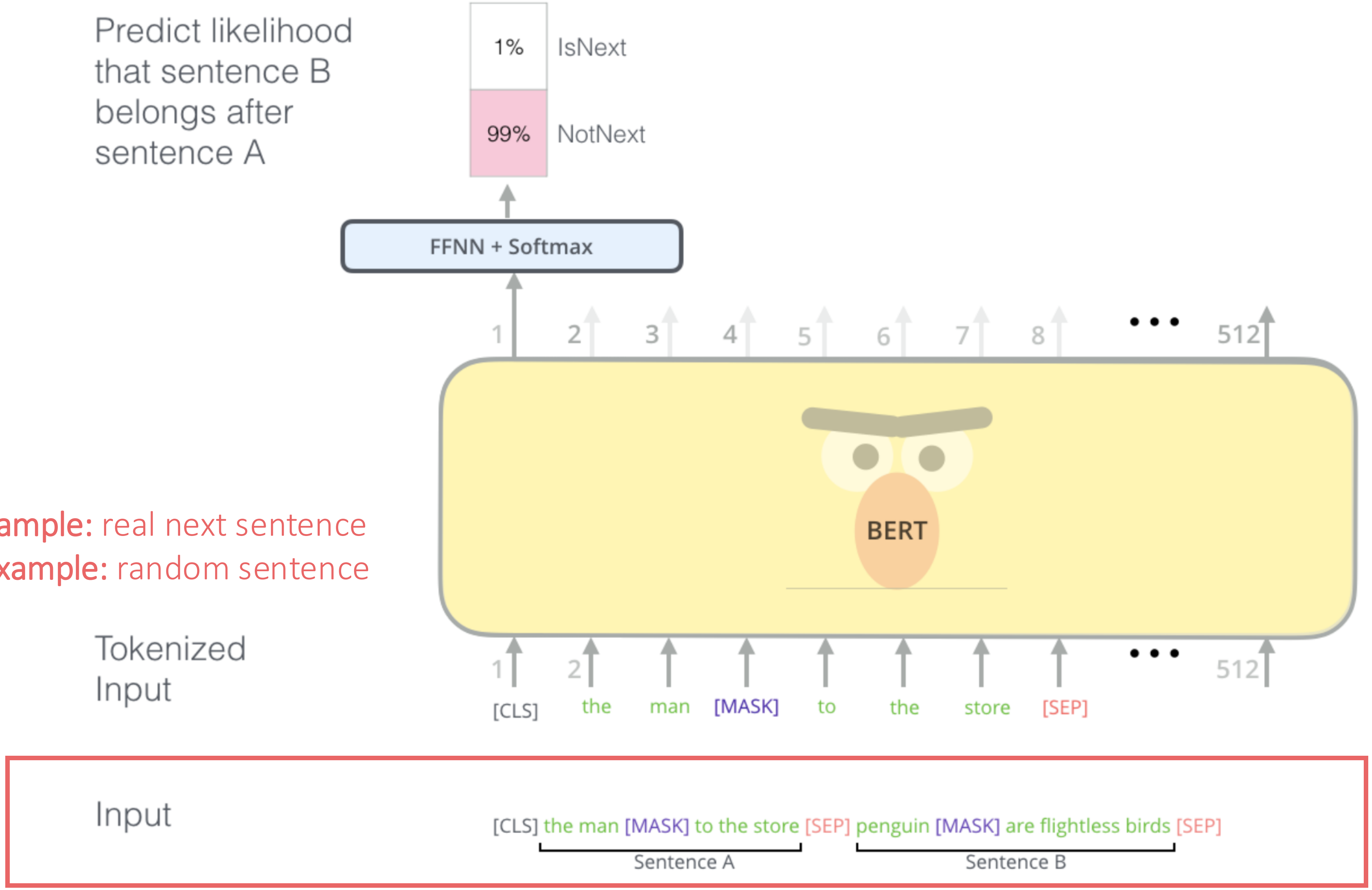
J.R.Firth 1957

- "You shall know a word by the company it keeps"
- One of the most successful ideas of modern statistical NLP!

…government debt problems turning into	***banking***	*crises as happened in 2009…*
…saying that Europe needs unified	***banking***	*regulation to replace the hodgepodge…*
…India has just given its	***banking***	*system a shot in the arm…*



Pre-Training Task: Next Sentence Prediction

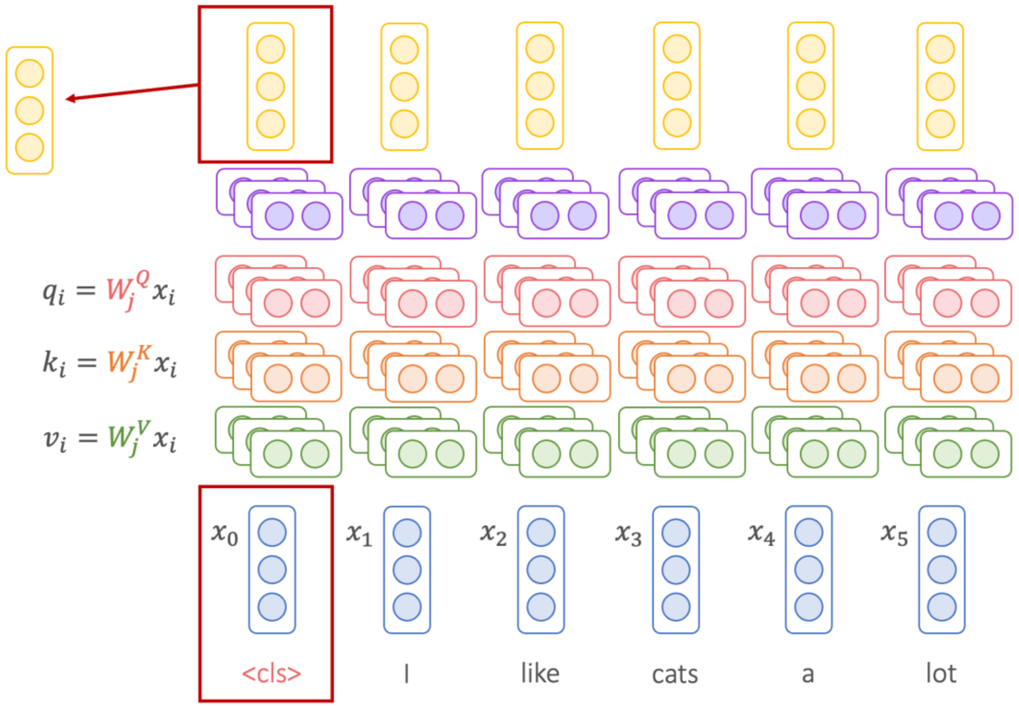


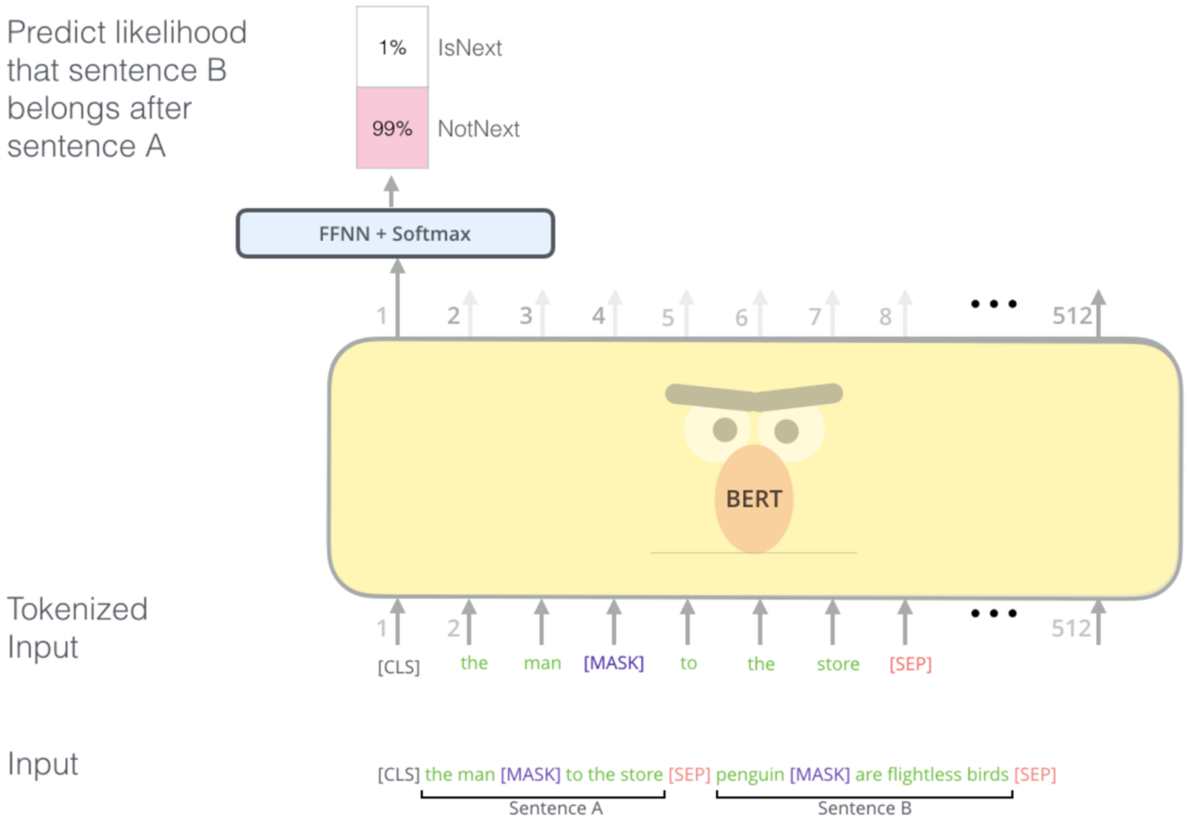
Positive example: real next sentence
Negative example: random sentence

http://jalammar.github.io/illustrated-bert/

Pre-Training Task: Next Sentence Prediction

- Why do we need this?

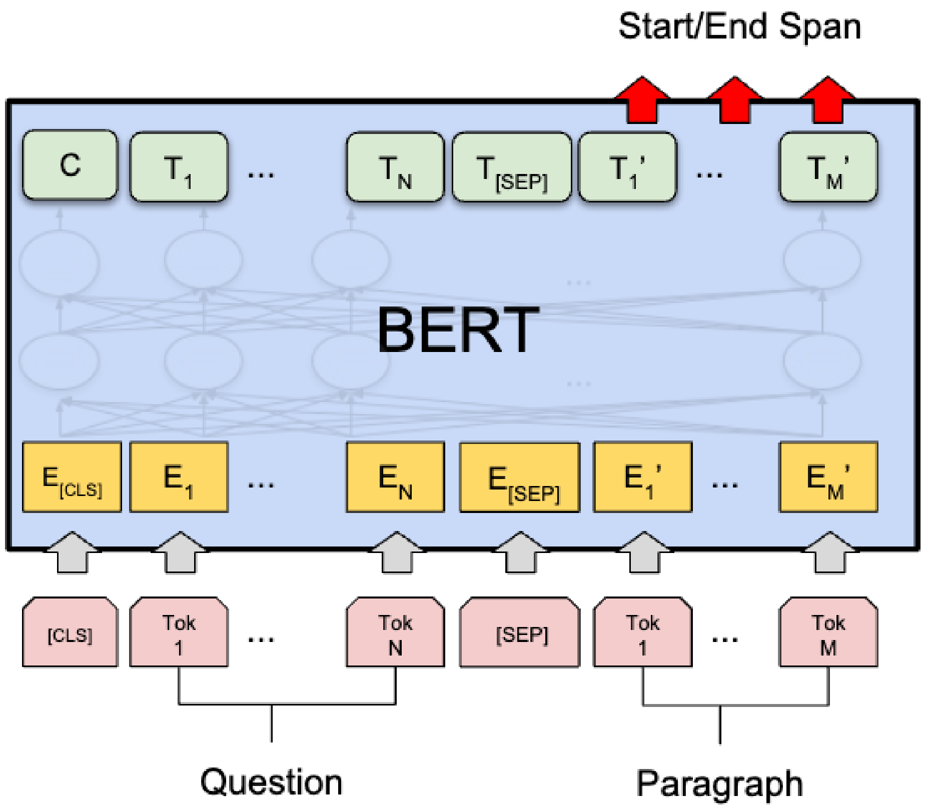


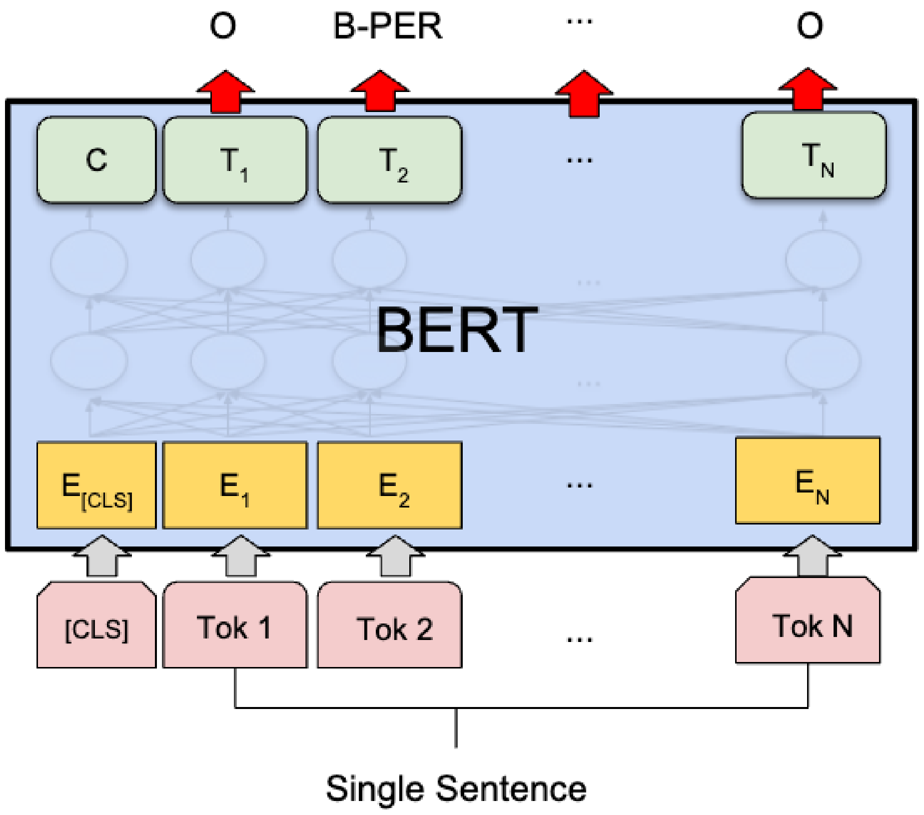


Do we really need this (?)

Fine-Tuning: Token-Level Tasks

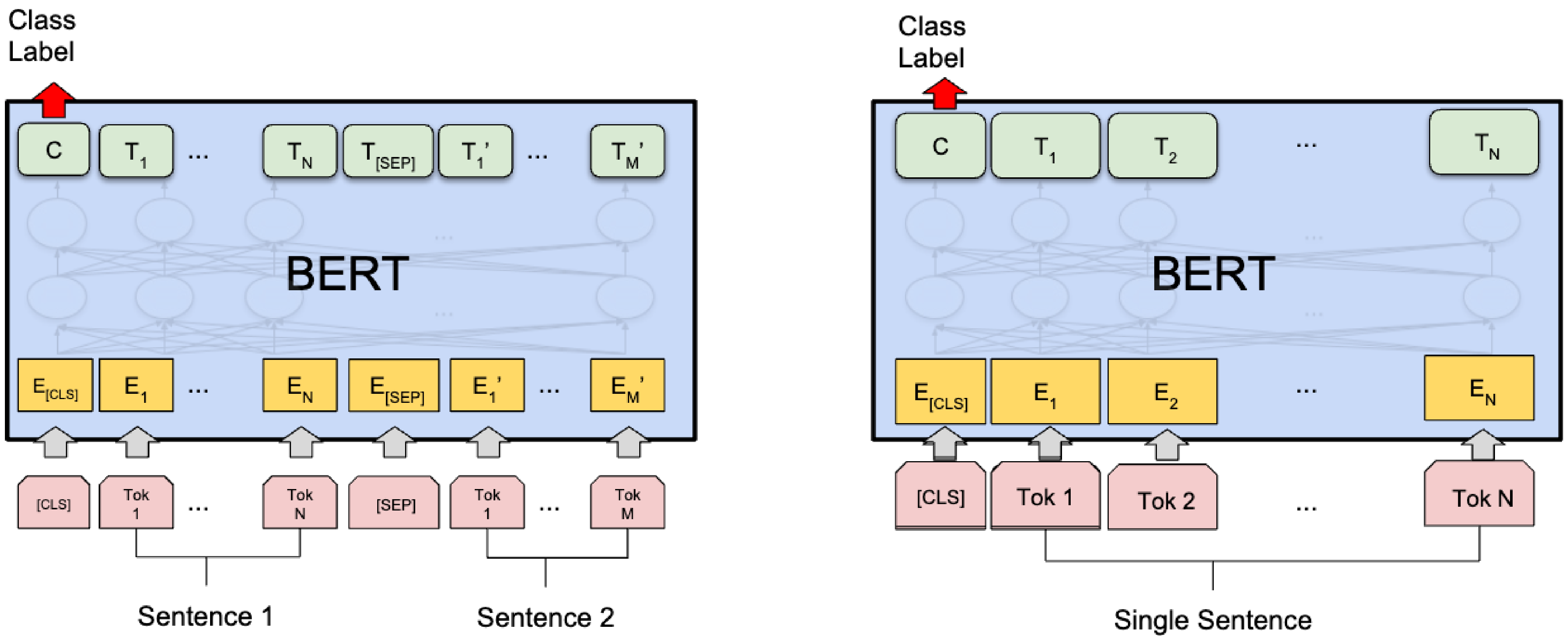
- Pre-training provides a good weight initialization



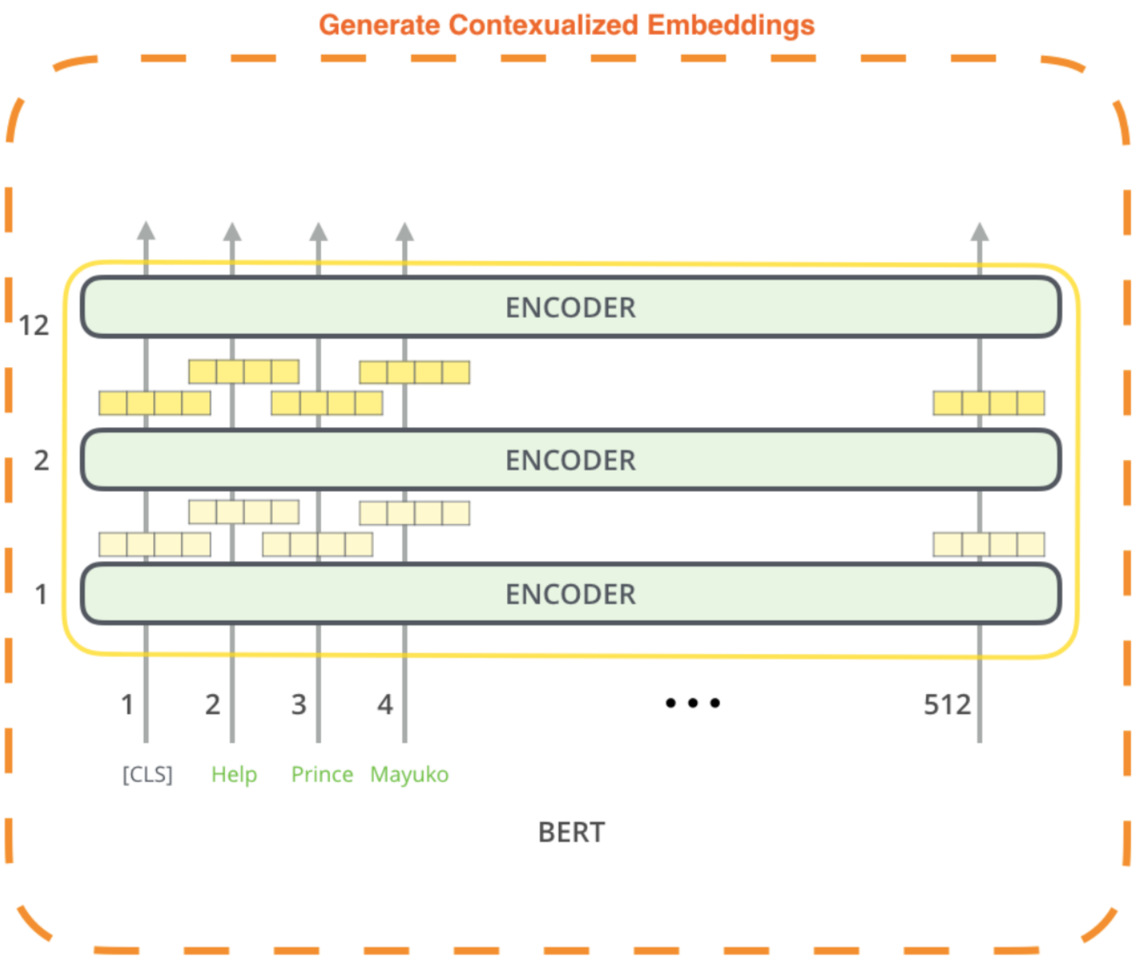


Fine-Tuning: Sentence-Level Tasks

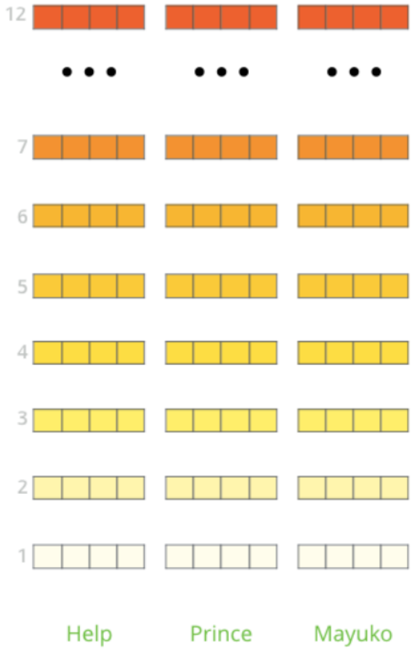
- Pre-training provides a good weight initialization



BERT as General Contextualized Representations



The output of each encoder layer along each token's path can be used as a feature representing that token.



But which one should we use?

https://medium.com/towards-data-science/breaking-bert-down-430461f60efb

Use BERT

Hugging Face

- BERT-base
  - 12 layers, hidden size = 768, 12 attention heads
  - # parameters ≈ 110M
- BERT-large
  - 24 layers, hidden size = 1024, 16 attention heads
  - # parameters ≈ 340M
- Cased vs. Uncased

Amazing Performance

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	79.8	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**

Encoder-Only: SpanBERT

SpanBERT: Improving Pre-training by Representing and Predicting Spans

Mandar Joshi[*†] **Danqi Chen**[*‡§] **Yinhan Liu**[§]
Daniel S. Weld[†ϵ] **Luke Zettlemoyer**[†§] **Omer Levy**[§]

[†] Allen School of Computer Science & Engineering, University of Washington, Seattle, WA
{mandar90,weld,lsz}@cs.washington.edu

[‡] Computer Science Department, Princeton University, Princeton, NJ
danqic@cs.princeton.edu

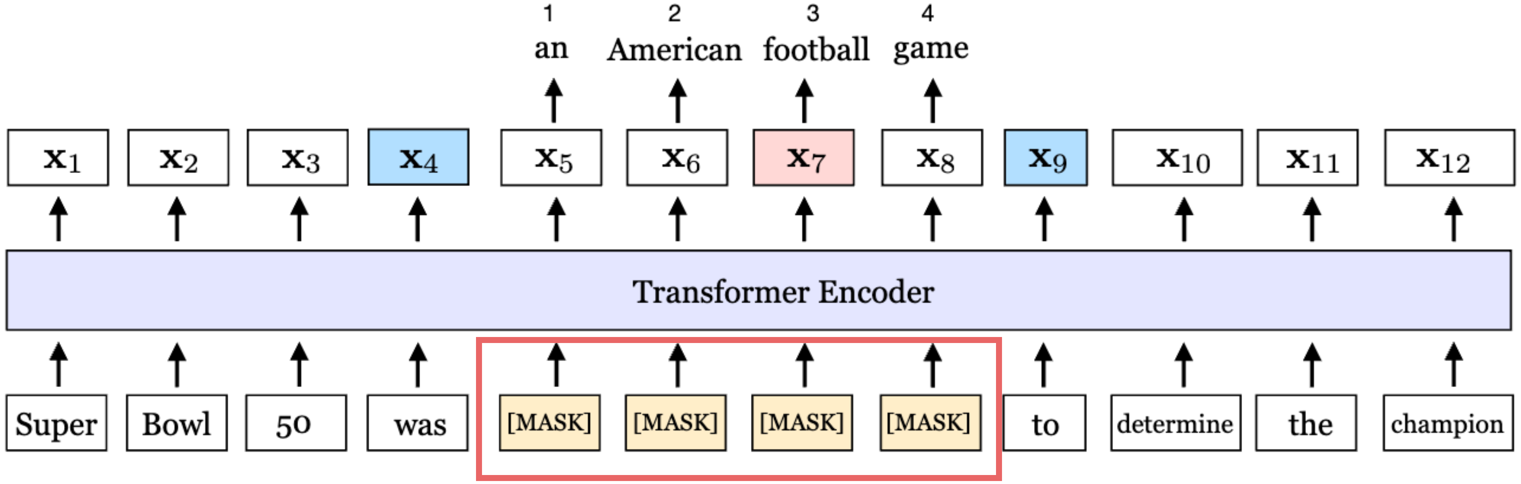
[ϵ]Allen Institute of Artificial Intelligence, Seattle
{danw}@allenai.org

[§] Facebook AI Research, Seattle
{danqi,yinhanliu,lsz,omerlevy}@fb.com

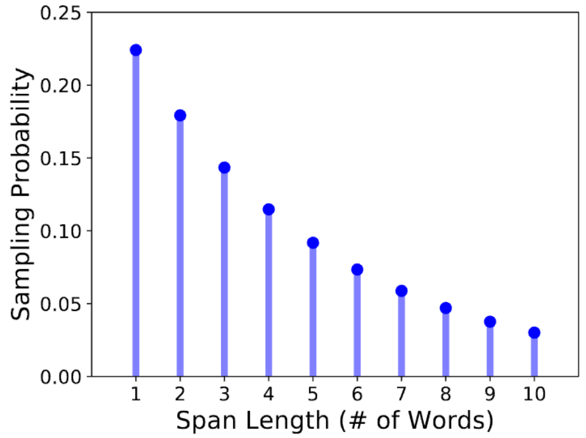
SpanBERT: Improving Pre-training by Representing and Predicting Spans, 2020

Encoder-Only: SpanBERT

$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$
$$= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)$$



- Span masking
- Single sentence training
- Span boundary objective (SBO)



Better Performance Than BERT

	NewsQA	TriviaQA	SearchQA	HotpotQA	Natural Questions	Avg.
Google BERT	68.8	77.5	81.7	78.3	79.9	77.3
Our BERT	71.0	79.0	81.8	80.5	80.5	78.6
Our BERT-1seq	71.9	80.4	84.0	80.3	81.8	79.7
SpanBERT	**73.6**	**83.6**	**84.8**	**83.0**	**82.5**	**81.5**

Use SpanBERT

Hugging Face

- SpanBERT-base
  - 12 layers, hidden size = 768, 12 attention heads
  - # parameters ≈ 110M
- SpanBERT-large
  - 24 layers, hidden size = 1024, 16 attention heads
  - # parameters ≈ 340M
- Cased vs. Uncased

Encoder-Only: RoBERTa

RoBERTa: A Robustly Optimized BERT Pretraining Approach

Yinhan Liu[*§] **Myle Ott**[*§] **Naman Goyal**[*§] **Jingfei Du**[*§] **Mandar Joshi**[†]
Danqi Chen[§] **Omer Levy**[§] **Mike Lewis**[§] **Luke Zettlemoyer**[†§] **Veselin Stoyanov**[§]

[†] Paul G. Allen School of Computer Science & Engineering,
University of Washington, Seattle, WA
{mandar90,lsz}@cs.washington.edu

[§] Facebook AI
{yinhanliu,myleott,naman,jingfeidu,
danqi,omerlevy,mikelewis,lsz,ves}@fb.com

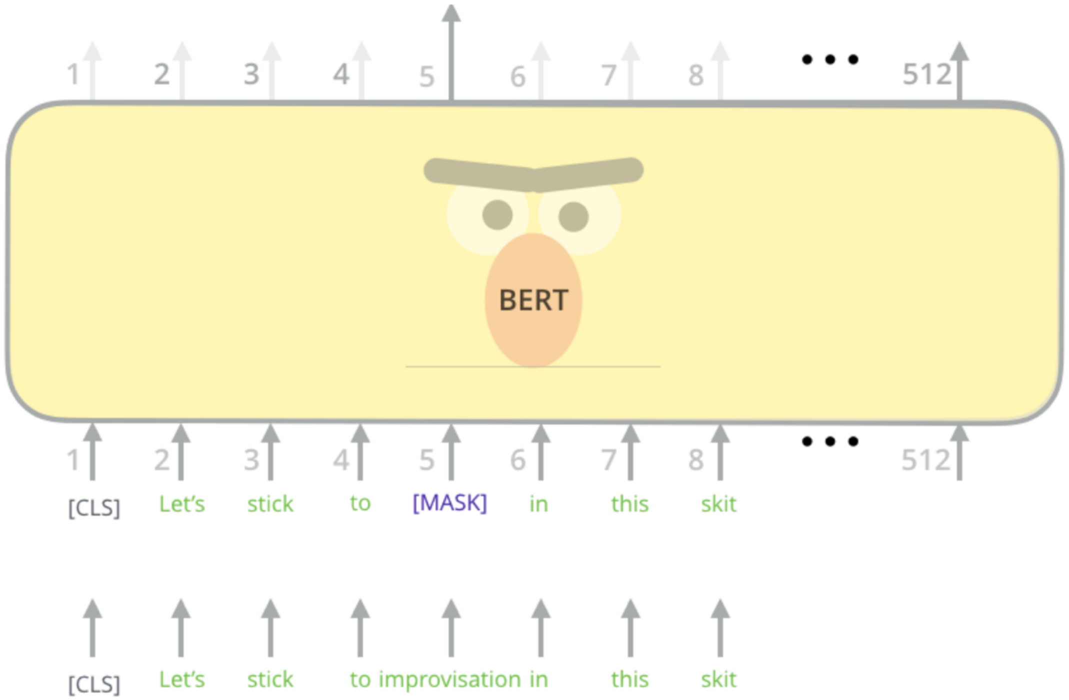
RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019

Encoder-Only: RoBERTa

- **R**obustly **o**ptimized **BERT** **a**pproach (RoBERTa)
- BERT is still under-trained
- Improve the robustness of training BERT

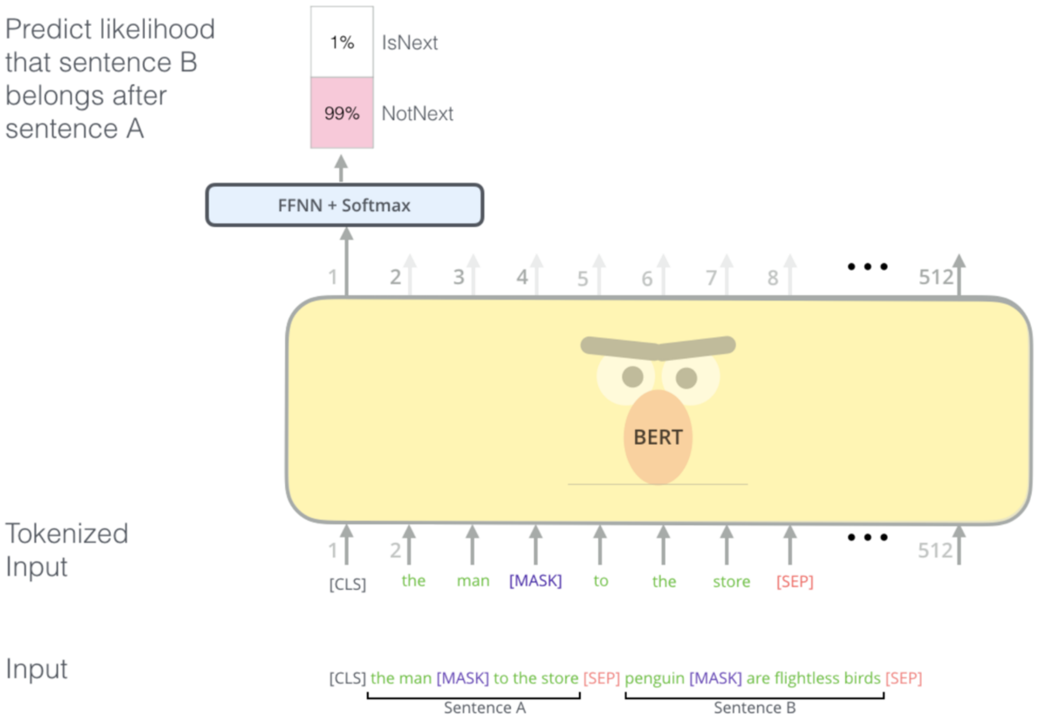
Static Masking vs. Dynamic Masking

- **Static masking:** decide masked words during data pre-processing
- **Dynamic masking**: decide masked words right before feeding into models



Masking	SQuAD 2.0	MNLI-m	SST-2
static	78.3	84.3	92.5
dynamic	78.7	84.0	92.9

Removing Next Sentence Prediction Task



Model	**SQuAD 1.1/2.0**	**MNLI-m**	**SST-2**	**RACE**
Our reimplementation (with NSP loss):				
SEGMENT-PAIR	90.4/78.7	84.0	92.9	64.2
SENTENCE-PAIR	88.7/76.2	82.9	92.1	63.0
Our reimplementation (without NSP loss):				
FULL-SENTENCES	90.4/79.1	84.7	92.5	64.8
DOC-SENTENCES	90.6/79.7	84.7	92.7	65.6

True Byte-Pair Encoding (BPE)

- BERT: BPE with unicode characters
  - Vocabulary size: 30K
- RoBERTa: BPE with bytes
  - Vocabulary size: 50K

Training Details

- Trained longer
- 10x data
- Bigger batch sizes

Much Better Performance Than BERT

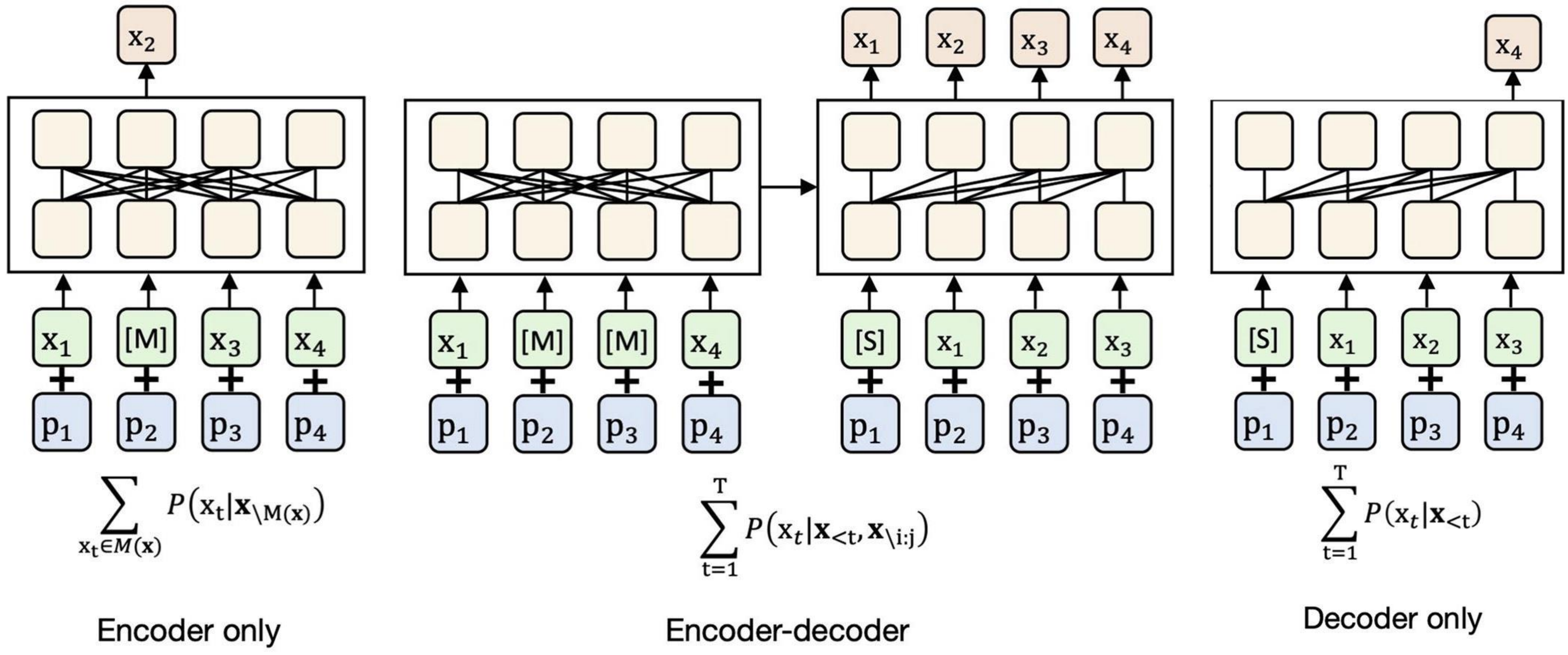
Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7

Use RoBERTa

Hugging Face

- RoBERTa-base
  - 12 layers, hidden size = 768, 12 attention heads
  - # parameters ≈ 110M
- RoBERTa-large
  - 24 layers, hidden size = 1024, 16 attention heads
  - # parameters ≈ 340M

Types of Pre-Training



https://www.sciencedirect.com/science/article/pii/S2095809922006324

Encoder-Decoder: BART

- **B**idirectional and **A**uto-**R**egressive **T**ransformers (BART)

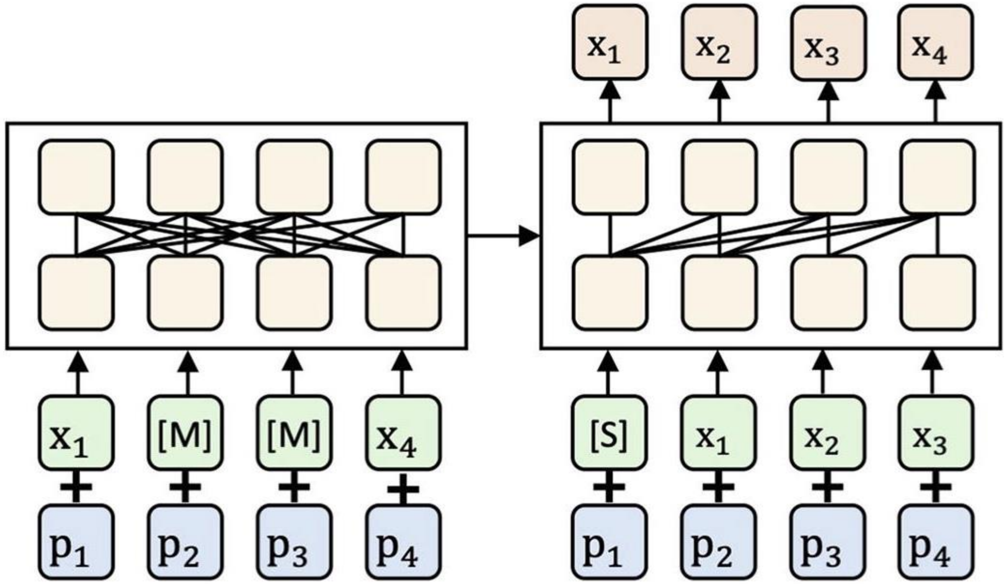
BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

Mike Lewis*, Yinhan Liu*, Naman Goyal*, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer
Facebook AI
{mikelewis,yinhanliu,naman}@fb.com

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, 2020

Encoder-Decoder: BART

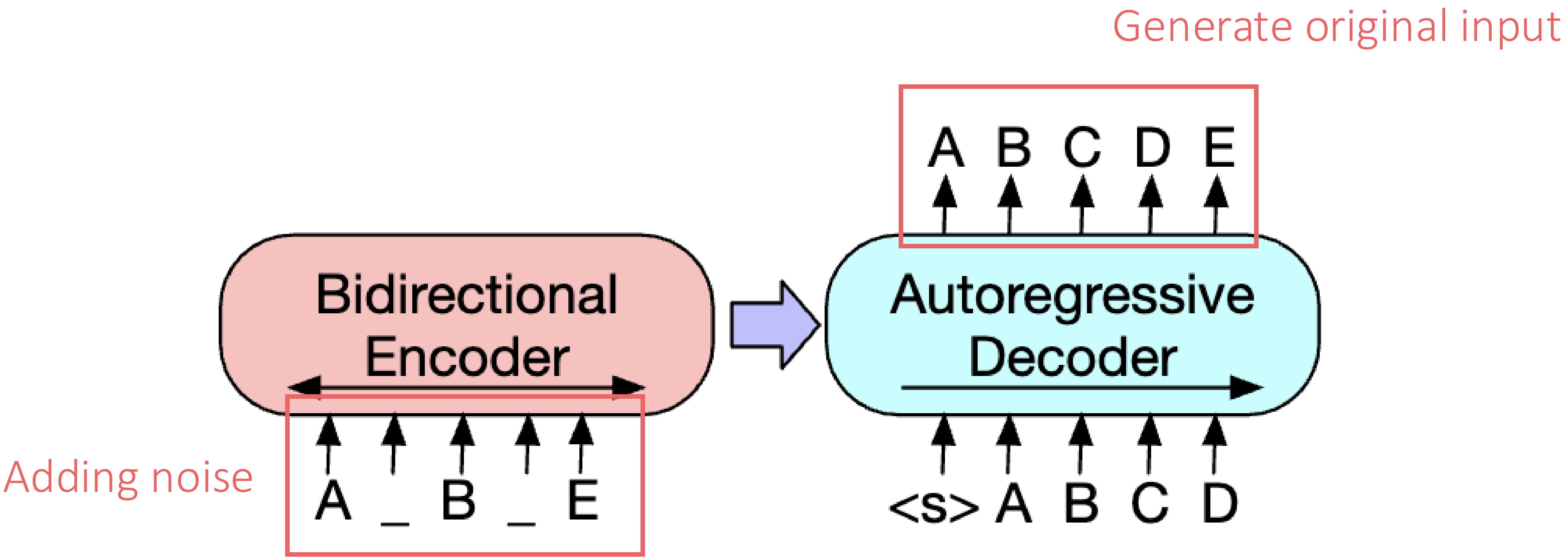
- Transformer Encoder-Decoder
- Pre-training for generation tasks but can be also used for representations



$$\sum_{t=1}^{T} P\left(\mathrm{x}_t | \mathbf{x}_{<\mathrm{t}}, \mathbf{x}_{\backslash \mathrm{i:j}}\right)$$

Encoder-decoder

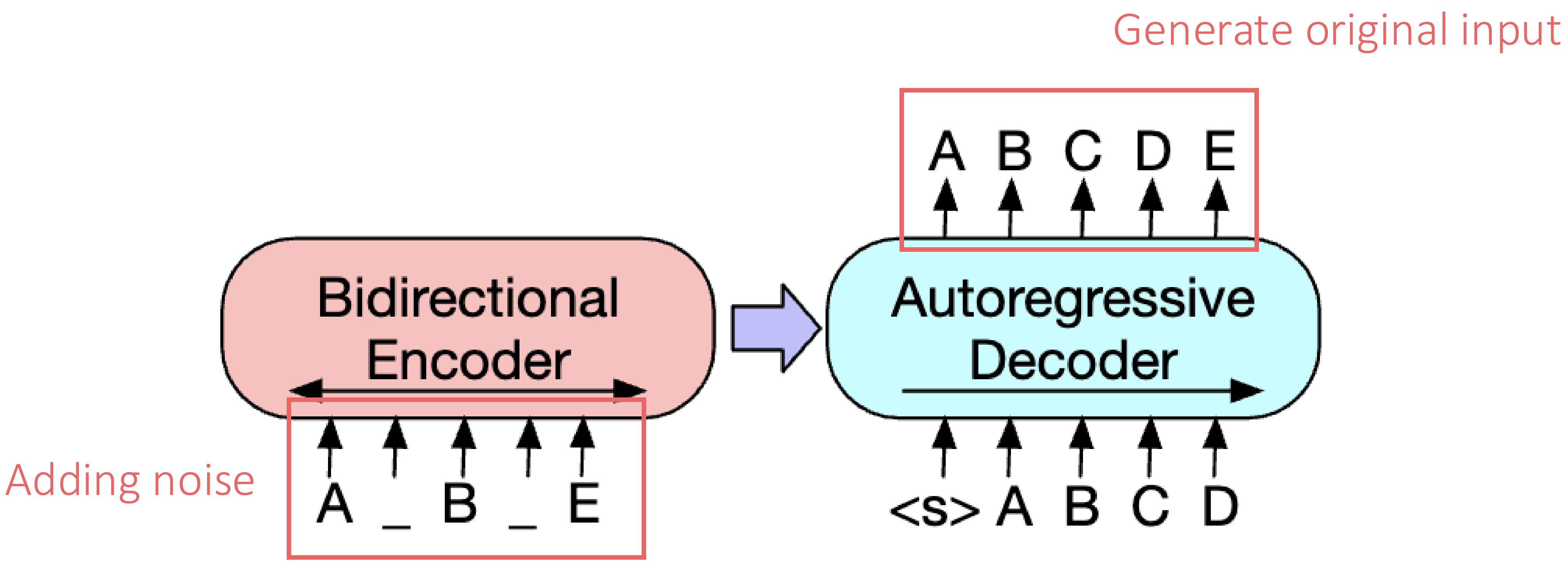
Denoising Autoencoder



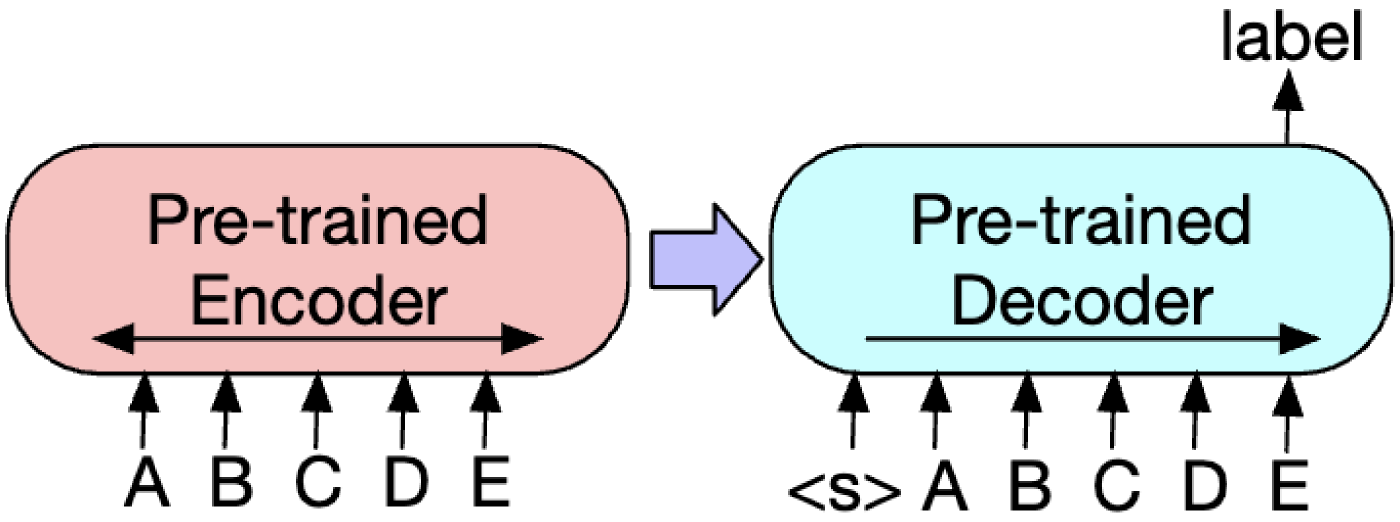
Denoising Objective

- Token Masking
  - A<mask>CD<mask>F. → ABCDEF.
- Token Deletion
  - ACDF. → ABCDEF.
- Text Infilling
  - A<mask>D<mask>F. → ABCDEF.
- Sentence Permutation
  - FG. ABC. DE. → ABC. DE. FG.
- Document Rotation
  - E. FG. ABC. D → ABC. DE. FG.

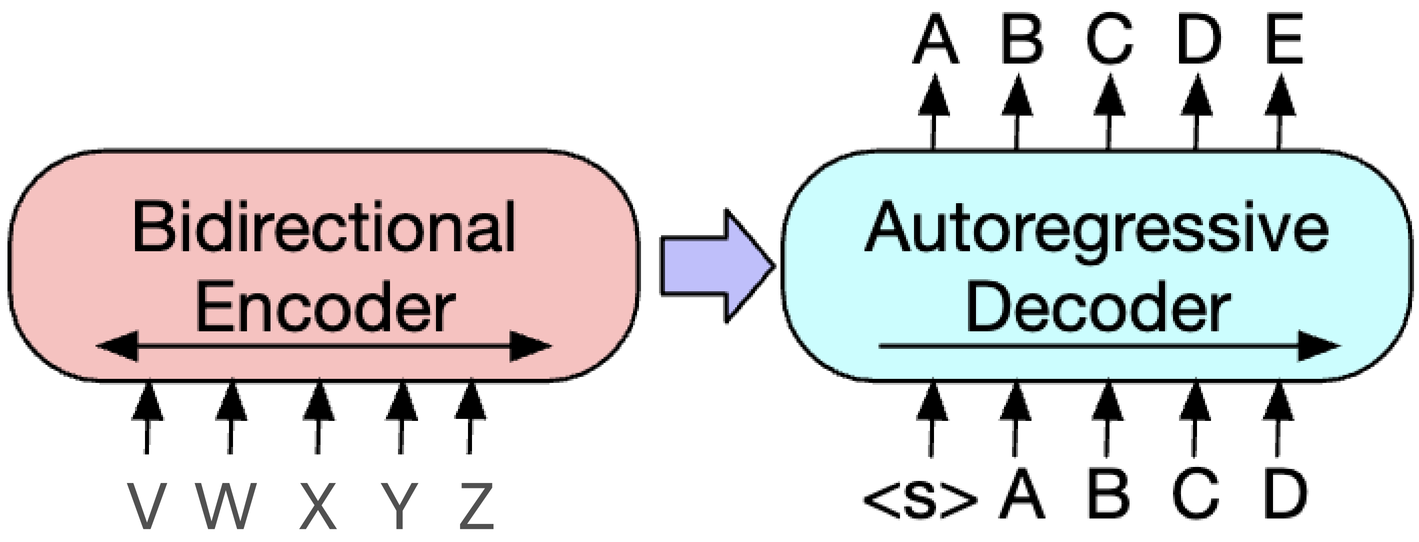
Denoising Autoencoder



Fine-Tuning: Sentence-Level Tasks



Fine-Tuning: Sequence-to-Sequence



Comparable Performance on Classification Tasks

	SQuAD 1.1 EM/F1	**SQuAD 2.0** EM/F1	**MNLI** m/mm	**SST** Acc	**QQP** Acc	**QNLI** Acc	**STS-B** Acc	**RTE** Acc	**MRPC** Acc	**CoLA** Mcc
BERT	84.1/90.9	79.0/81.8	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
RoBERTa	88.9/**94.6**	**86.5/89.4**	**90.2/90.2**	96.4	92.2	94.7	**92.4**	86.6	**90.9**	**68.0**
BART	88.8/**94.6**	86.1/89.2	89.9/90.1	**96.6**	**92.5**	**94.9**	91.2	**87.0**	90.4	62.8

Better Performance on Generation Tasks

Summarization

	CNN/DailyMail			XSum		
	R1	R2	RL	R1	R2	RL
Lead-3	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	28.10	8.02	21.72
UniLM	43.33	20.21	40.51	-	-	-
BERTSUMABS (Liu & Lapata, 2019)	41.72	19.39	38.76	38.76	16.33	31.15
BERTSUMEXTABS (Liu & Lapata, 2019)	42.13	19.60	39.18	38.81	16.50	31.27
BART	**44.16**	**21.28**	**40.90**	**45.14**	**22.27**	**37.25**

Question Answering

	ELI5		
	R1	R2	RL
Best Extractive	23.5	3.1	17.5
Language Model	27.8	4.7	23.1
Seq2Seq	28.3	5.1	22.8
Seq2Seq Multitask	28.9	5.4	23.1
BART	**30.6**	**6.2**	**24.3**

Translation

	RO-EN
Baseline	36.80
Fixed BART	36.29
Tuned BART	**37.96**

Use BART

Hugging Face

- BART-base
  - 6 layers for both encoder and decoder, hidden size = 768, 12 attention heads
  - # parameters ≈ 139M
- BART-large
  - 12 layers for both encoder and decoder, hidden size = 1024, 16 attention heads
  - # parameters ≈ 406M