# CSCE 638 Natural Language Processing Foundation and Techniques

## Lecture 9: Contextualized Representations and Pre-Training

Kuan-Hao Huang

Spring 2026

# Assignment 2

## Assignment 2

RELEASE DATE: 02/16/2026

DUE DATE: 03/03/2026 11:59pm on Gradescope

LaTeX Template: https://www.overleaf.com/read/gkcjzcvswxqt#8856b9

Name: First-Name Last-Name UIN: 000000000

---

This assignment consists of two parts: a writing section and a programming section. For the writing section, please use the provided LaTeX template to prepare your solutions and remember to fill in your name and UIN. For the programming section, please follow the instructions carefully.

Discussions with others on course materials and assignment solutions are encouraged, and the use of AI tools as assistance is permitted. However, you must ensure that **the final solutions are written in your own words**. It is your responsibility to avoid excessive similarity to others' work. Additionally, please clearly **indicate any parts where AI tools were used** as assistance.

If you have any question, please send an email to csce638-ta-26s@list.tamu.edu

---

# Lecture Plan

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

# A General Framework for Text Classification

Text $x$ → Feature (Representation) → Classifier (Model) → Label $y$
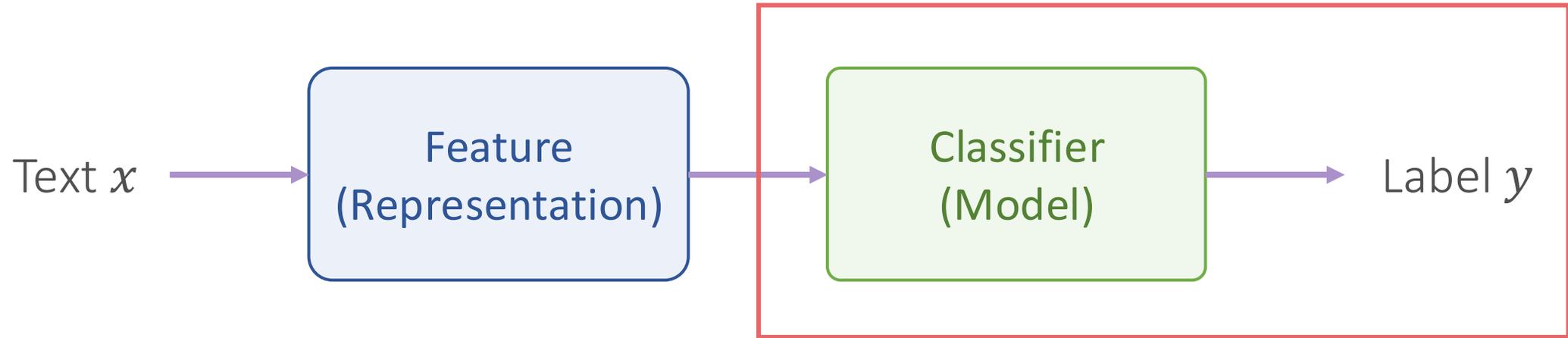
- Teach the model how to make prediction $y$
- Logistic regression, neural networks, CNN, RNN, LSTM, Transformers

# A General Framework for Text Classification
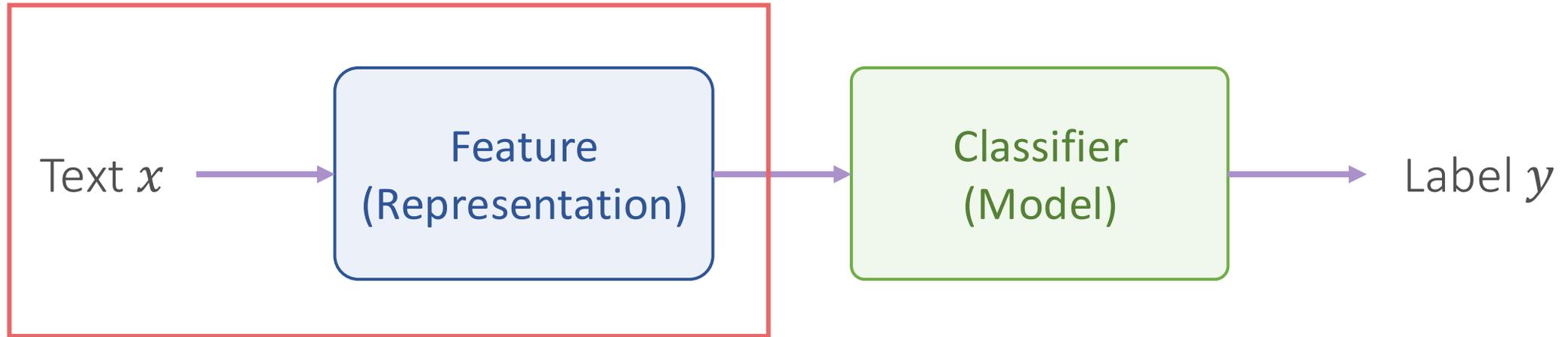


- Teach the model how to understand example $x$
- Rule-based representations
  - Bag-of-words, n-grams
- Learnable representations
  - Word2Vec (Skip-Gram and CBOW), GloVe, FastText

# Word Embeddings as Features
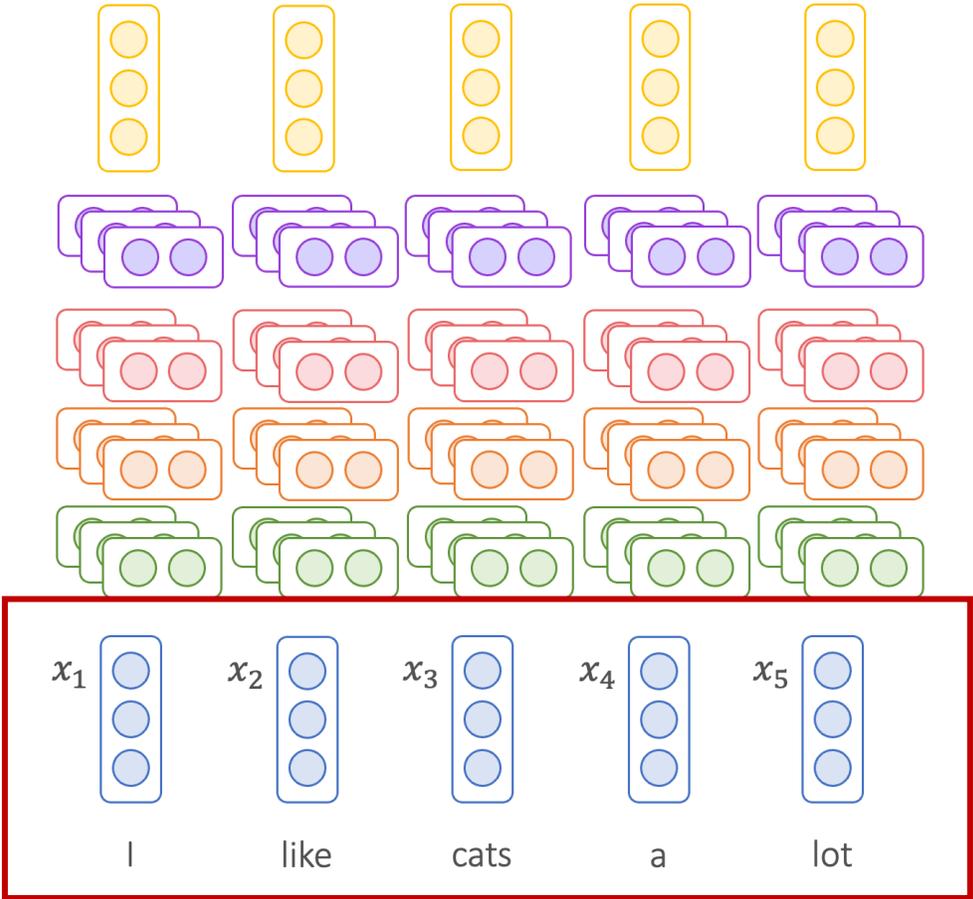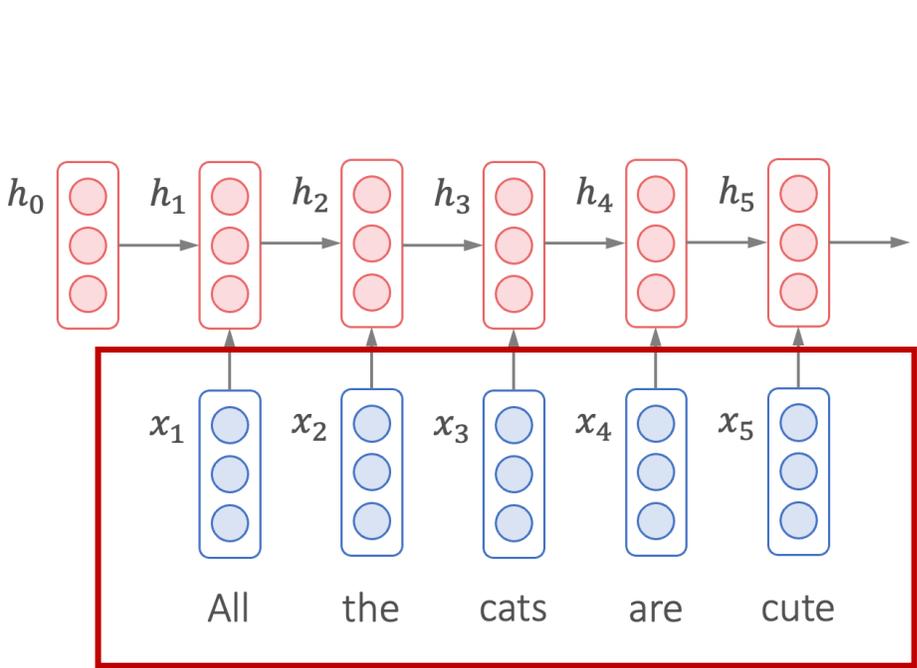
# Supervised Learning

## Training Stage

- Training data $\mathcal{D}_{train} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$
  - Example $x_i \in \mathcal{X}$, label $y_i \in C$
- Train a classifier(model) $f: \mathcal{X} \rightarrow C$ ← How to train?

## Testing Stage

- Testing data $\mathcal{D}_{test} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
- Make predictions $\tilde{y}_i = f(x_i)$
- Evaluate performance $\frac{1}{n} \sum_i S(y_i, \tilde{y}_i)$ ← Accuracy, F1 Score, etc.

# 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**[†*]
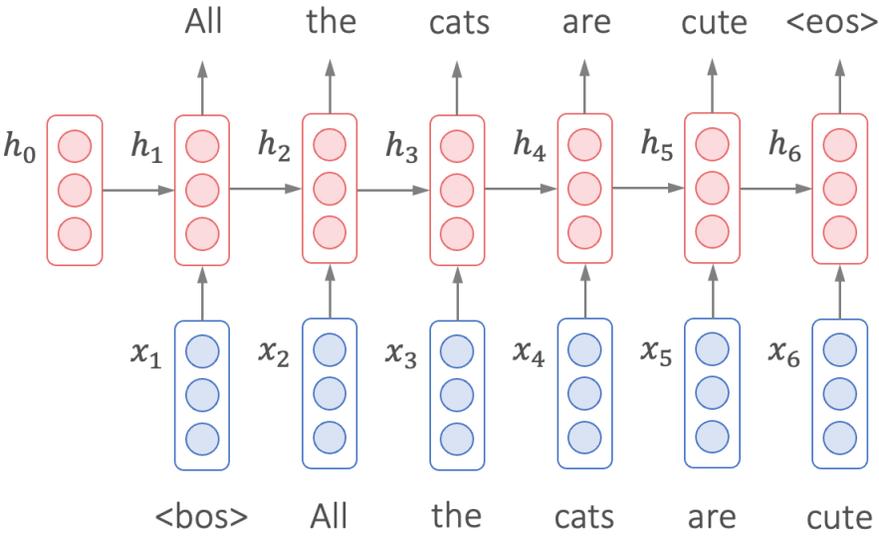{csquared,kentonl,lsz}@cs.washington.edu

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

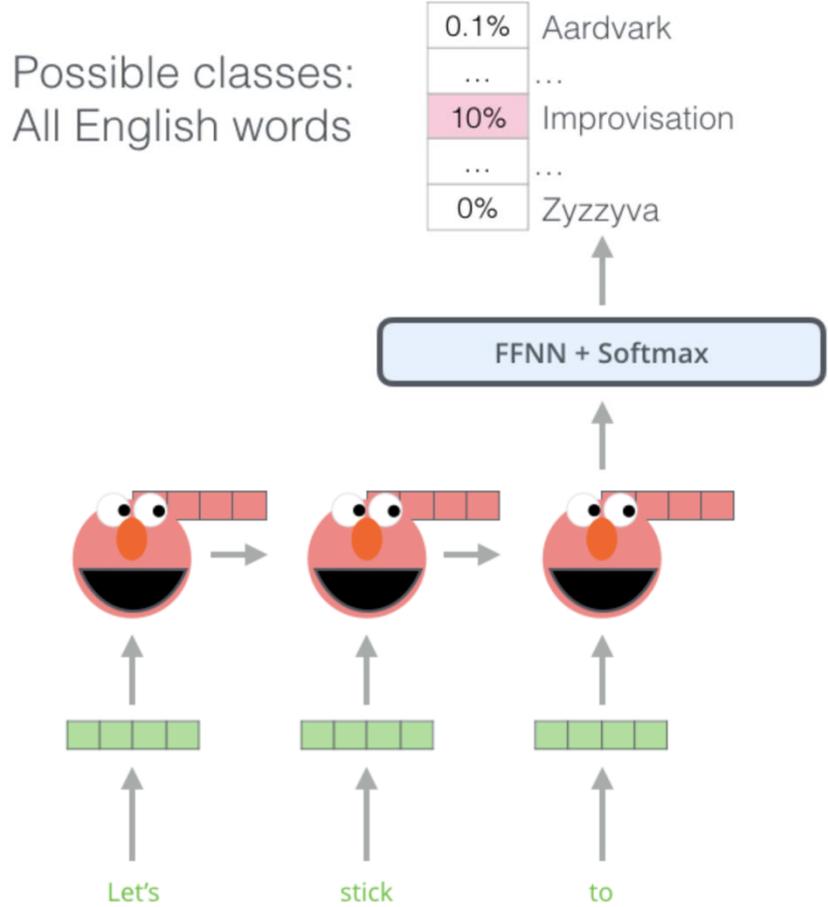# Recap: Continuous Bag of Words (CBOW) and Skip-Grams
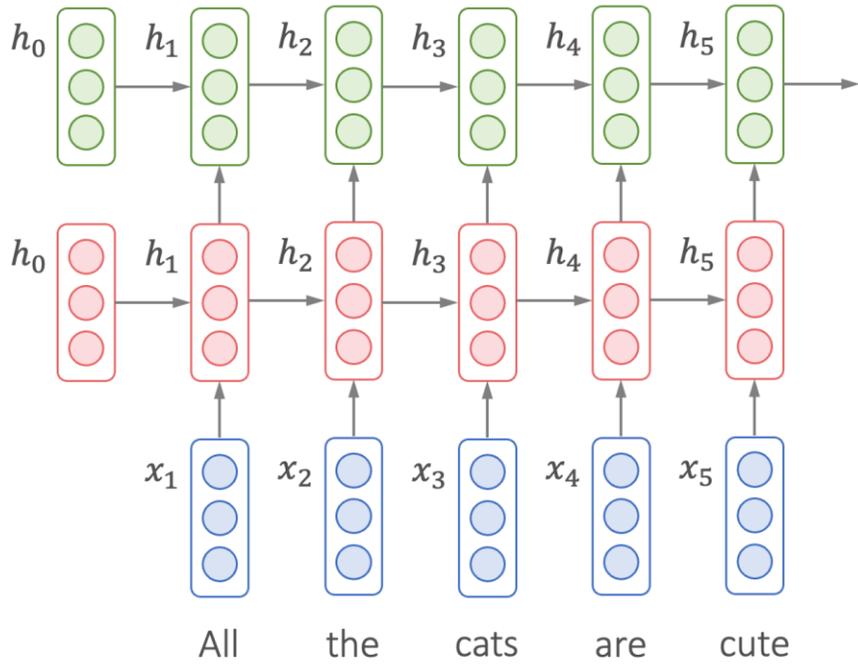
# ELMo: Language Modeling
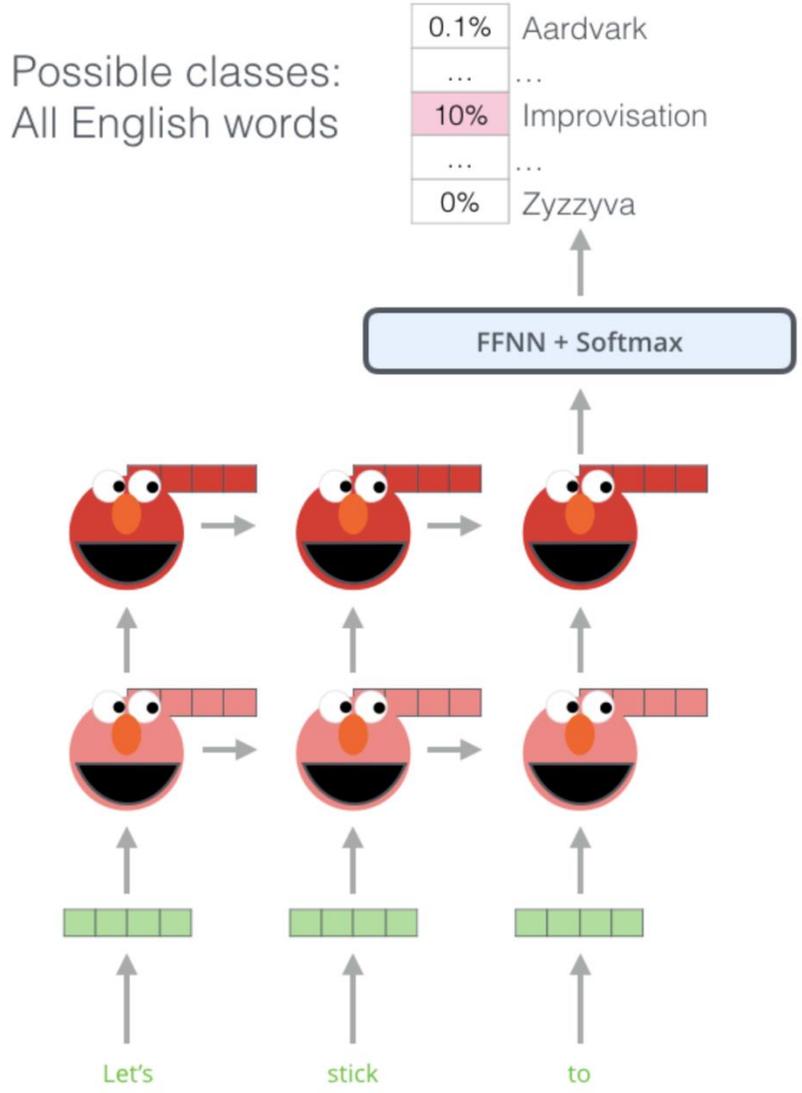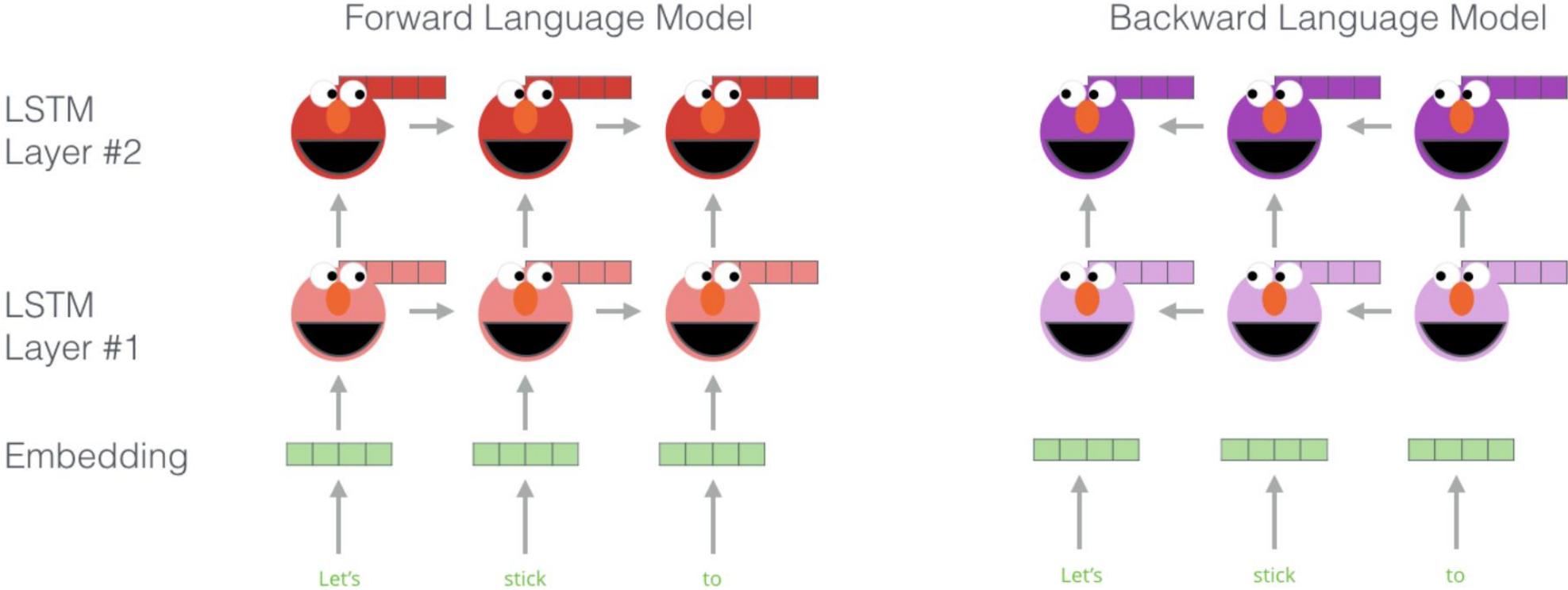
12
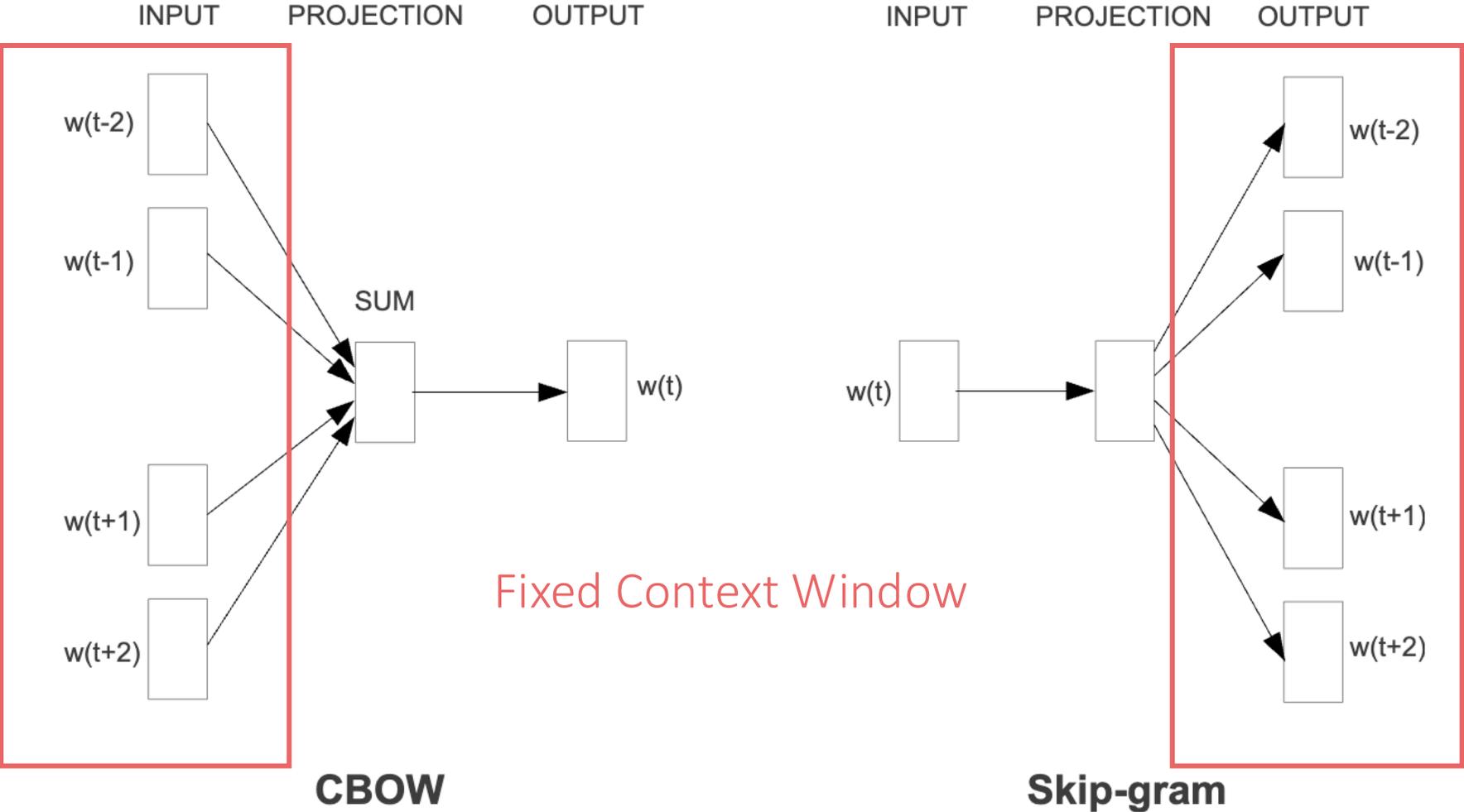
# ELMo: Language Modeling with Stacked LSTM
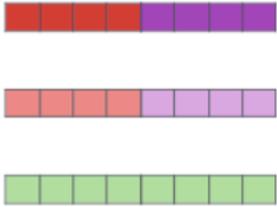
13

# ELMo: Bi-Directional Language Modeling

# ELMo: Contextualized Word Embeddings



1- Concatenate hidden layers

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

$\times \; s_2$

$\times \; s_1$

$\times \; s_0$

3- Sum the (now weighted) vectors

ELMo embedding of "stick" for this task in this context

Forward Language Model

Backward Language Model

Let's    stick    to

Let's    stick    to
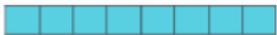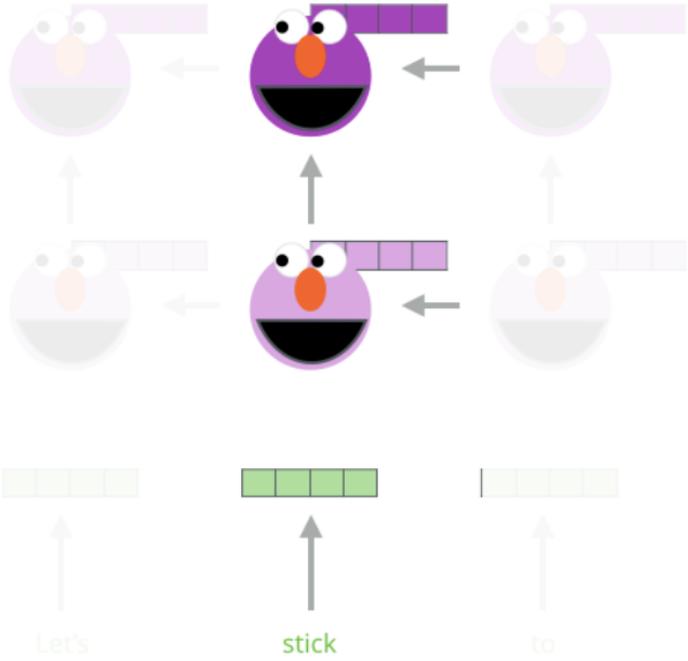
# How to Use ELMo?


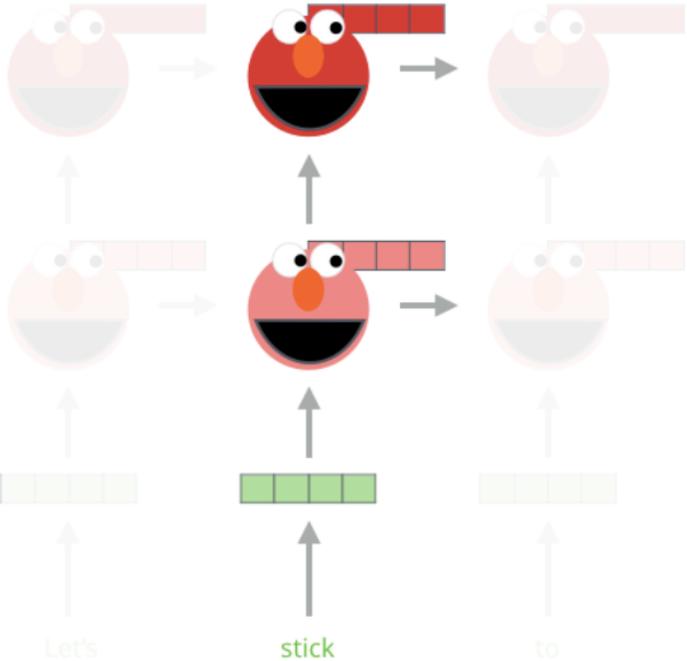
1- Concatenate hidden layers

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

$\times \quad s_2$

$\times \quad s_1$

$\times \quad s_0$

3- Sum the (now weighted) vectors

ELMo embedding of "stick" for this task in this context

Forward Language Model

Backward Language Model

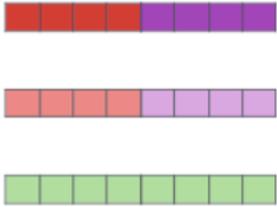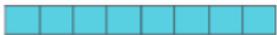Let's        stick        to

Let's        stick        to
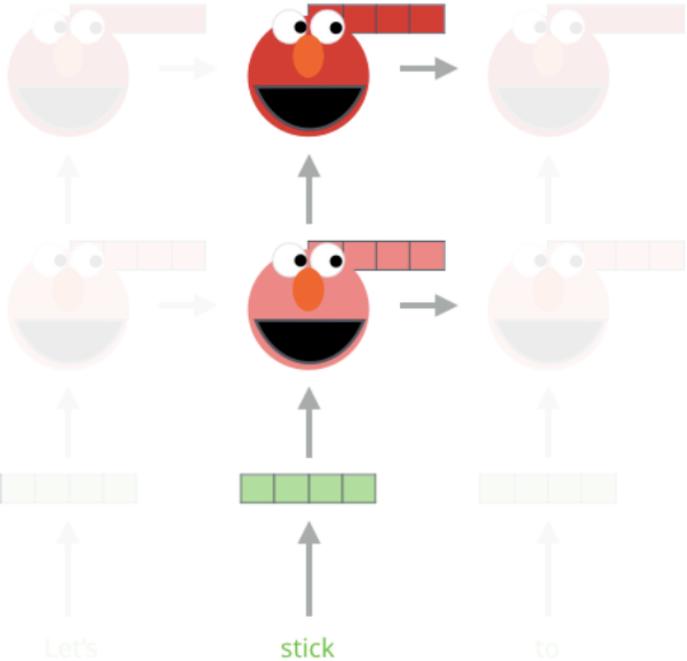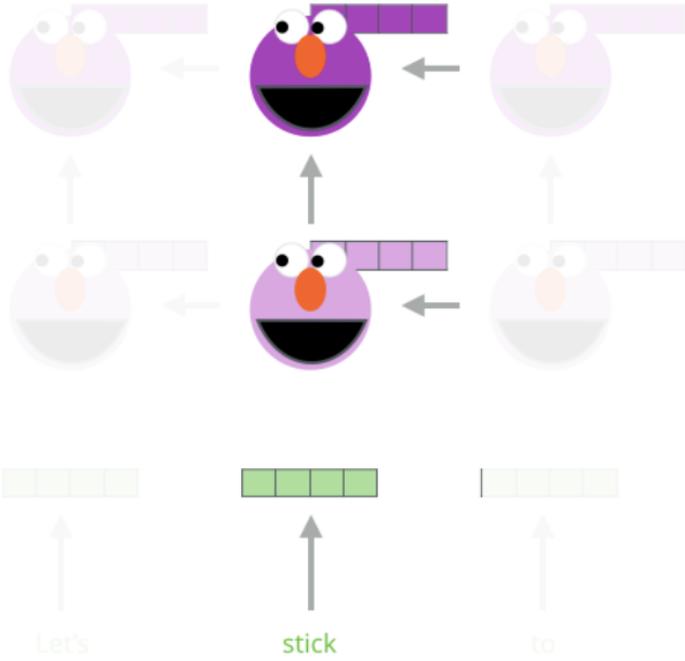
All     the     cats     are     cute

# 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 | $88.7 \pm 0.17$ |
| 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) | $91.93 \pm 0.19$ | 90.15 | $92.22 \pm 0.10$ |
| SST-5 | McCann et al. (2017) | 53.7 | 51.4 | $54.7 \pm 0.5$ |

# Lecture Plan

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

# 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
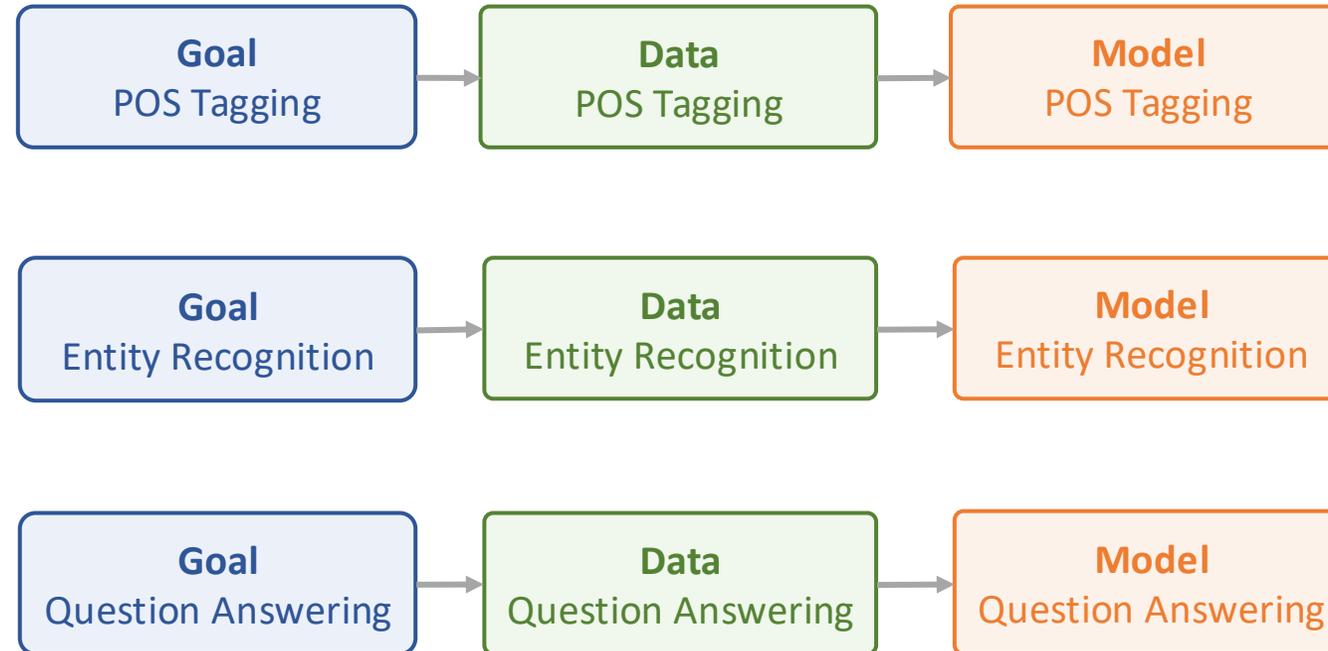
# Fine-Tuning with Pre-Training

# A General Framework for Text Classification

Text $x$ → **Feature (Representation)** → **Classifier (Model)** → Label $y$
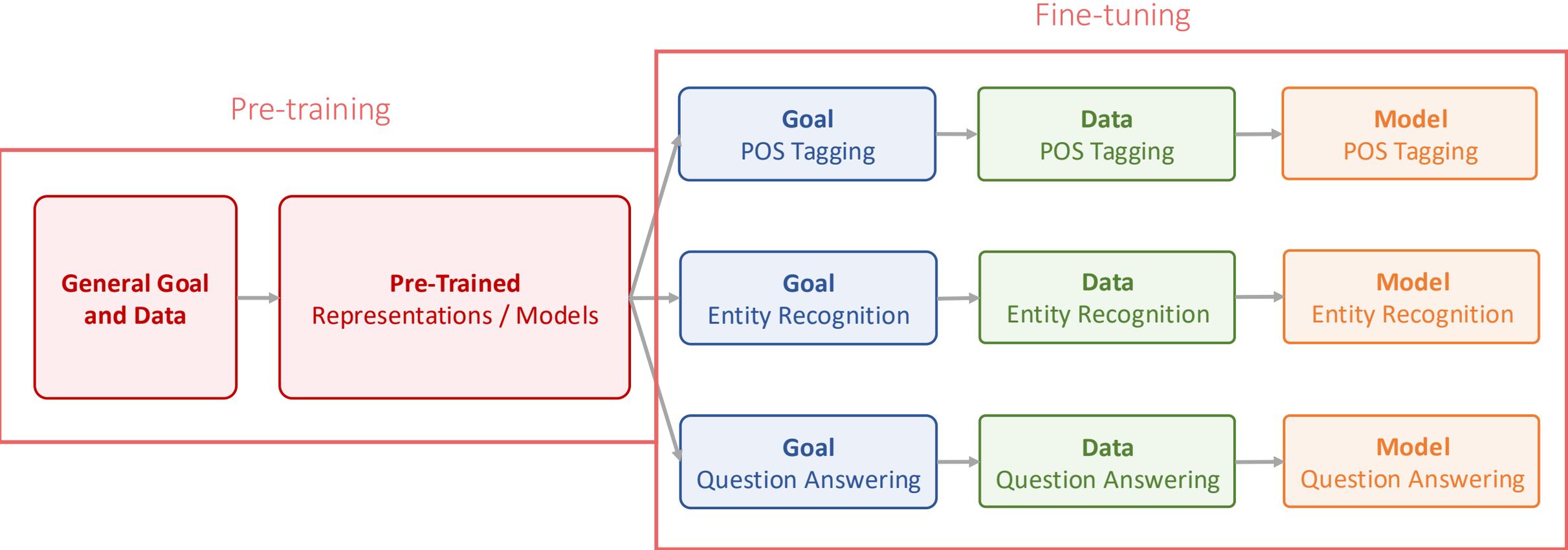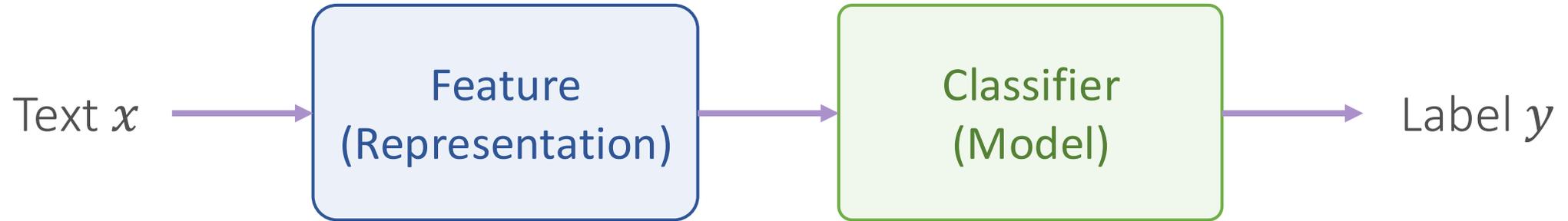
- Task-specific feature: N-gram features, TF-IDF
- Task-specific classifier: Logistic Regression, CNN, RNN, Transformers
- No pre-training

# A General Framework for Text Classification

Text $x$ → **Feature (Representation)** → **Classifier (Model)** → Label $y$
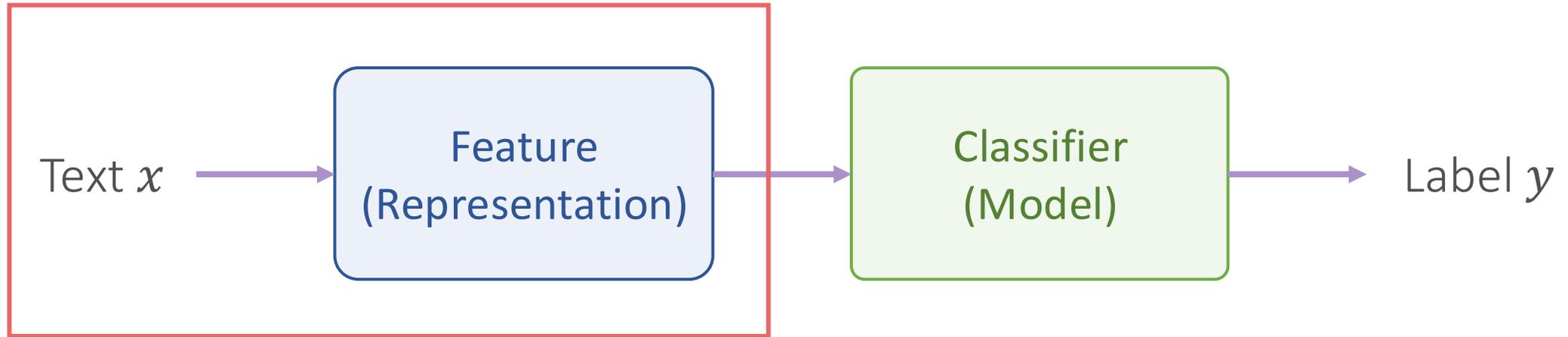
- Pre-trained feature: Word2Vec, Glove, ELMo
- Task-specific classifier: Logistic Regression, CNN, RNN, Transformers
- Pre-training on features/representations only

# A General Framework for Text Classification
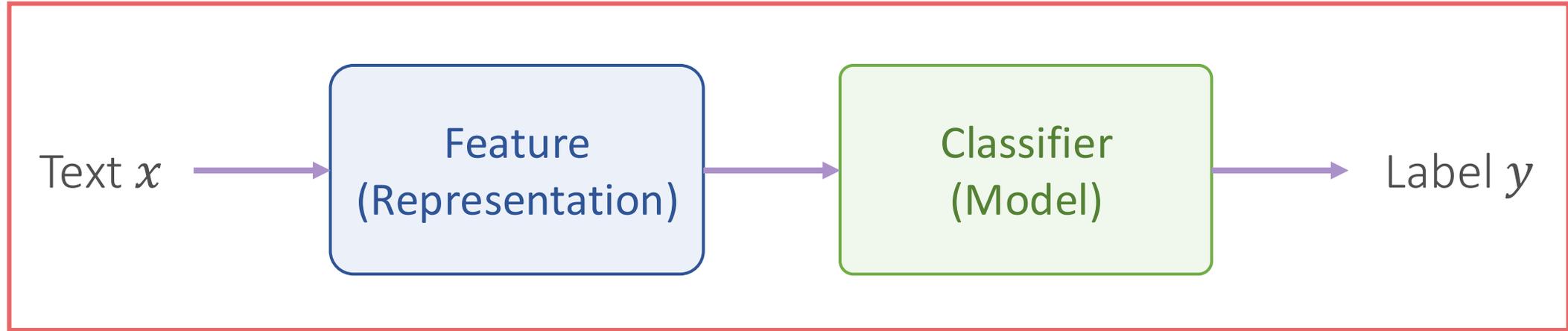


- Pre-training the whole pipeline
  - Pre-trained representations + pre-trained model weights
  - We only cover Transformer-based pre-training

# Types of Pre-Training



$$\sum_{x_t \in M(\mathbf{x})} P(x_t | \mathbf{x}_{\backslash M(\mathbf{x})})$$

Encoder only

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

Encoder-decoder

$$\sum_{t=1}^{T} P(x_t | \mathbf{x}_{<t})$$
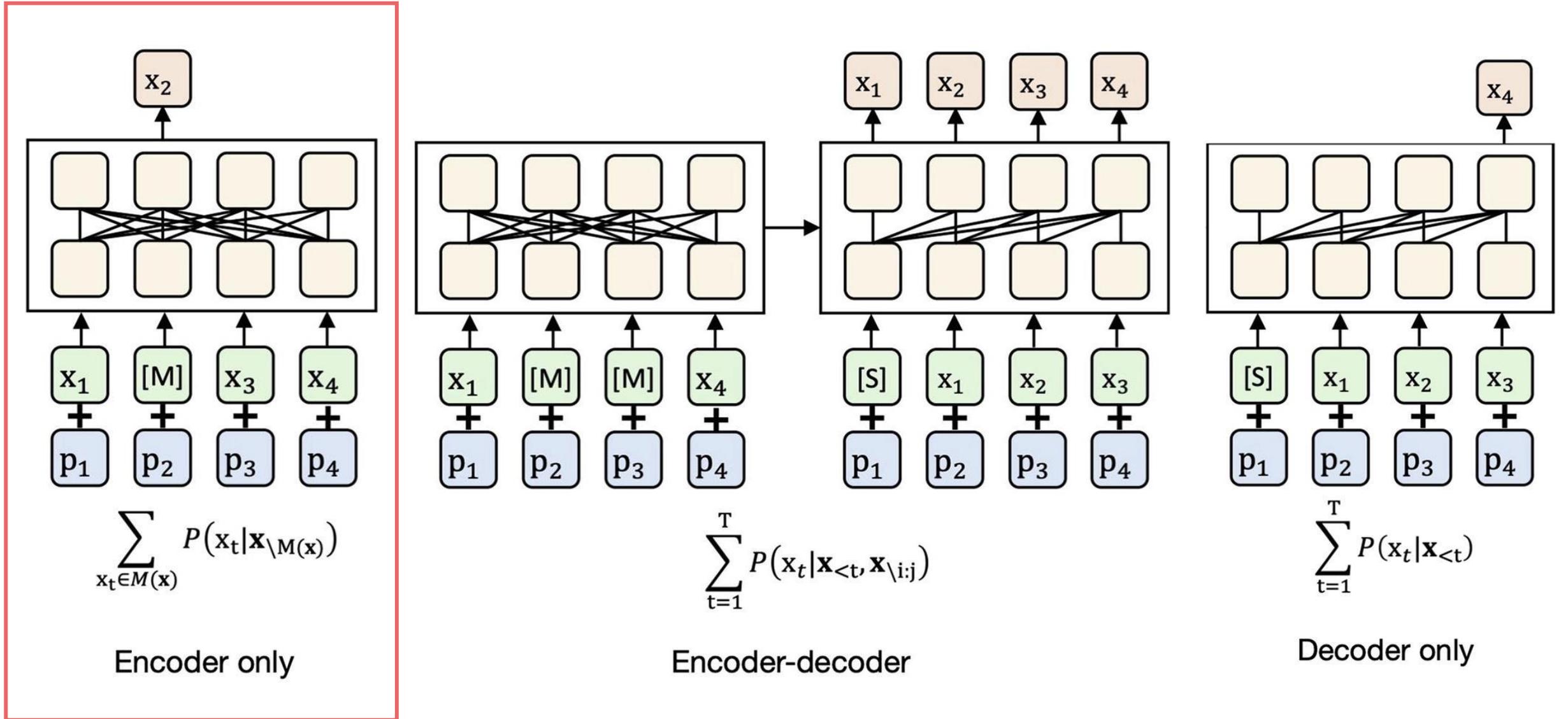
Decoder only

# 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

# 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



Use the output of the masked word's position to predict the masked word

Possible classes: All English words

| 0.1% | Aardvark |
| ... | ... |
| 10% | Improvisation |
| ... | ... |
| 0% | Zyzzyva |

Token Reconstruction

FFNN + Softmax

Stacked Transformer Encoder

BERT

Randomly mask 15% of tokens

Masking Tokens

[CLS]  Let's  stick  to  [MASK]  in  this  skit

Input

[CLS]  Let's  stick  to improvisation in  this  skit

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
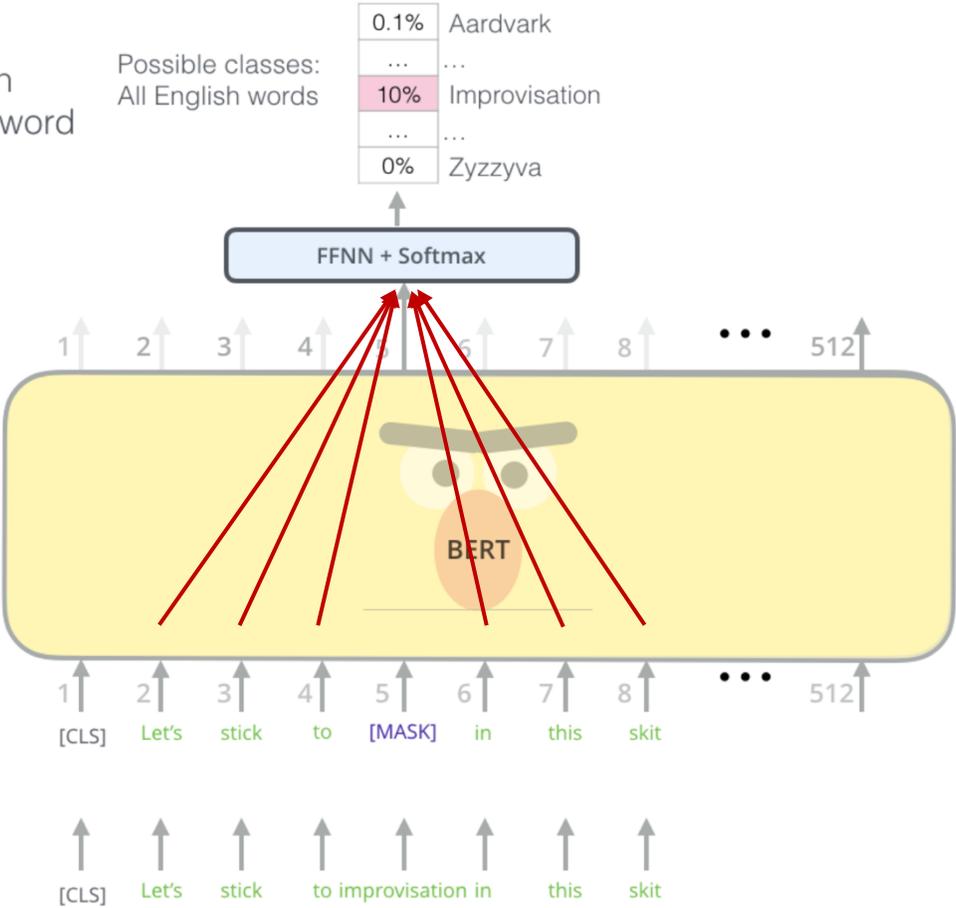
J.R.Firth 1957
- "You shall know a word by the company it keeps"
- One of the most successful ideas of modern statistical NLP!

...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...

Use the output of the masked word's position to predict the masked word

Possible classes: All English words

| 0.1% | Aardvark |
| ... | ... |
| 10% | Improvisation |
| ... | ... |
| 0% | Zyzzyva |

FFNN + Softmax

1  2  3  4  5  6  7  8  ...  512

BERT

1  2  3  4  5  6  7  8  ...  512
[CLS]  Let's  stick  to  [MASK]  in  this  skit

Randomly mask 15% of tokens

Input

[CLS]  Let's  stick  to improvisation in  this  skit

# Pre-Training Task: Next Sentence Prediction

Predict likelihood that sentence B belongs after sentence A

1%   IsNext

99%   NotNext

FFNN + Softmax

1   2   3   4   5   6   7   8   •••   512

BERT
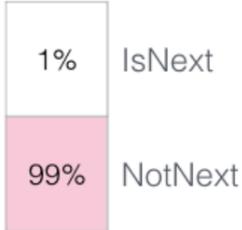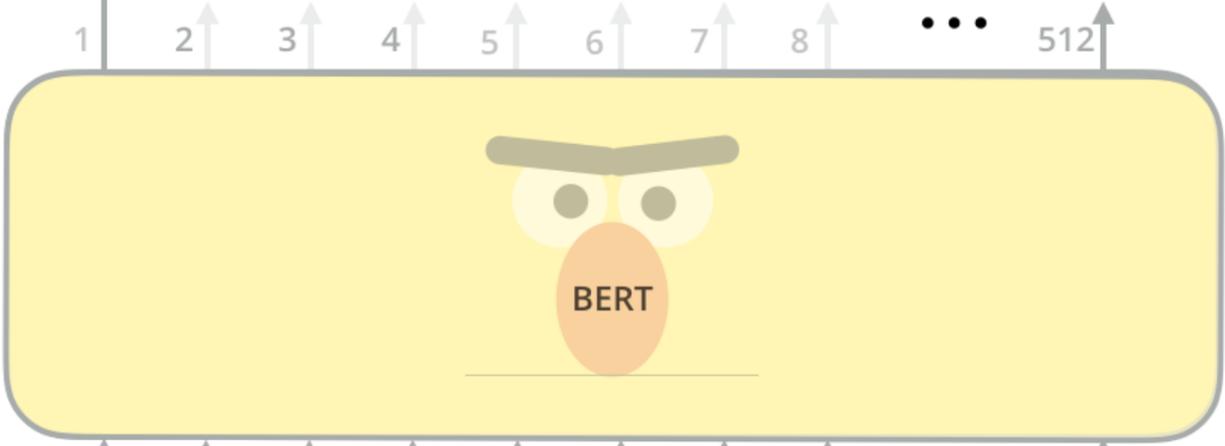
**Positive example:** real next sentence
**Negative example:** random sentence

Tokenized Input

1   2     •••   512

[CLS]   the   man   [MASK]   to   the   store   [SEP]

Input

[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]

Sentence A        Sentence B
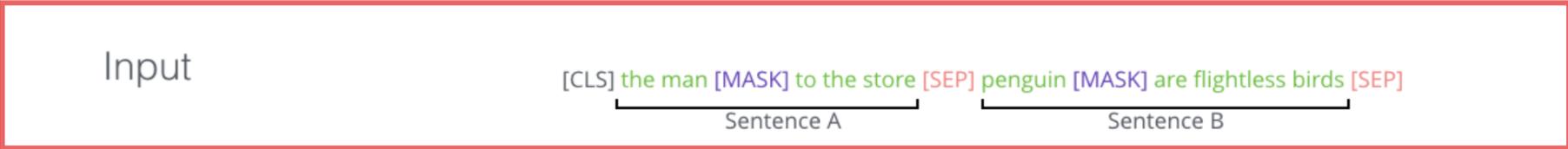
# Pre-Training Task: Next Sentence Prediction

- Why do we need this?



$q_i = W_j^Q x_i$

$k_i = W_j^K x_i$

$v_i = W_j^V x_i$

$x_0$ &lt;cls&gt;  $x_1$ I  $x_2$ like  $x_3$ cats  $x_4$ a  $x_5$ lot

Predict likelihood that sentence B belongs after sentence A

1% IsNext
99% NotNext

FFNN + Softmax

1  2  3  4  5  6  7  8  ···  512

BERT

1  2  ···  512

Tokenized Input

[CLS]  the  man  [MASK]  to  the  store  [SEP]

Input

[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]
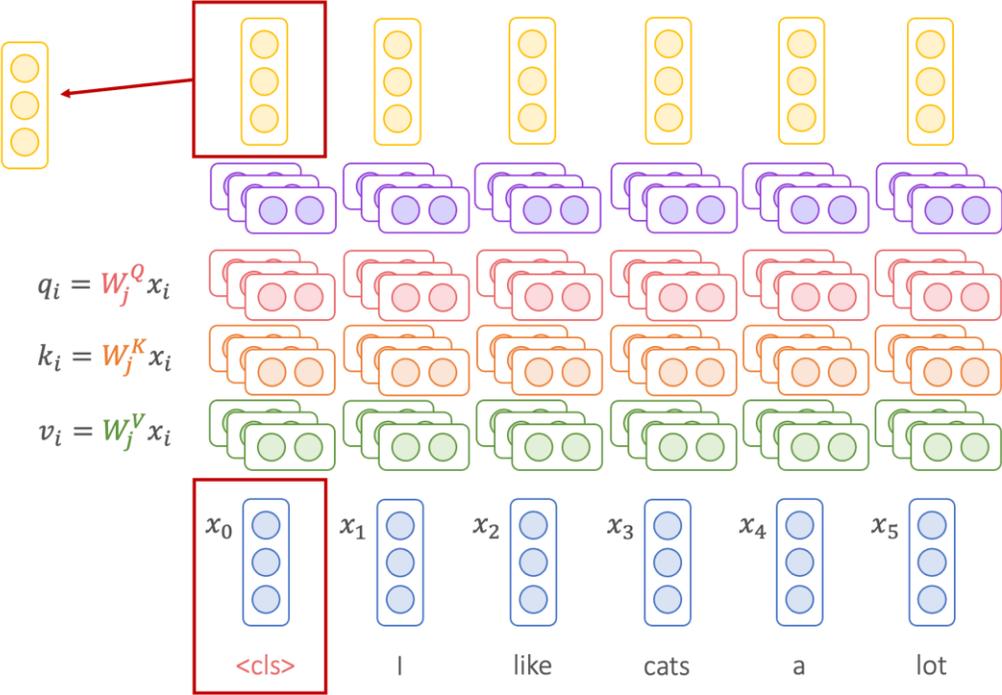
Sentence A                    Sentence B
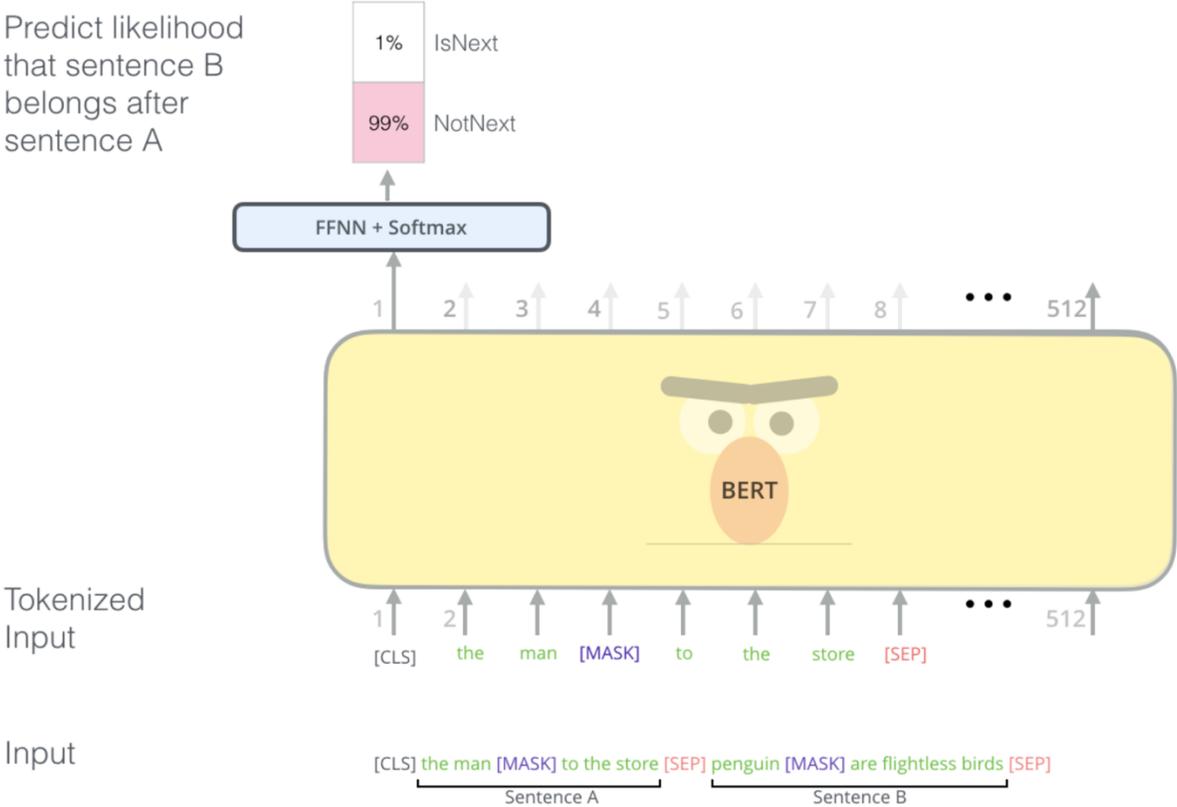
Do we really need this (?)

# Fine-Tuning: Token-Level Tasks

- Pre-training provides a good weight initialization
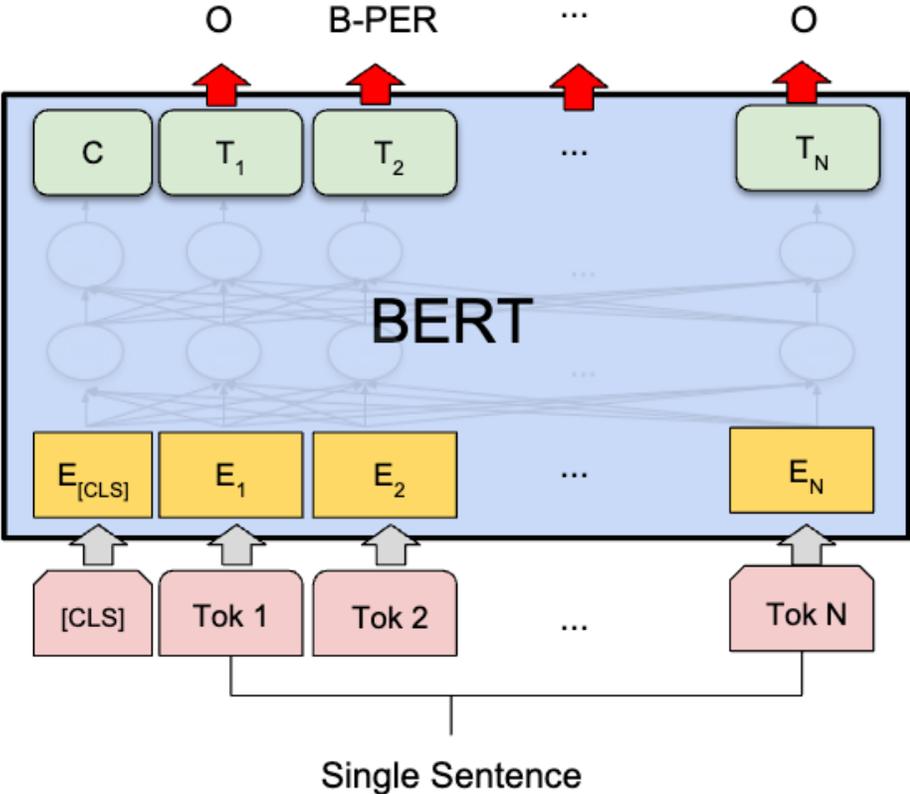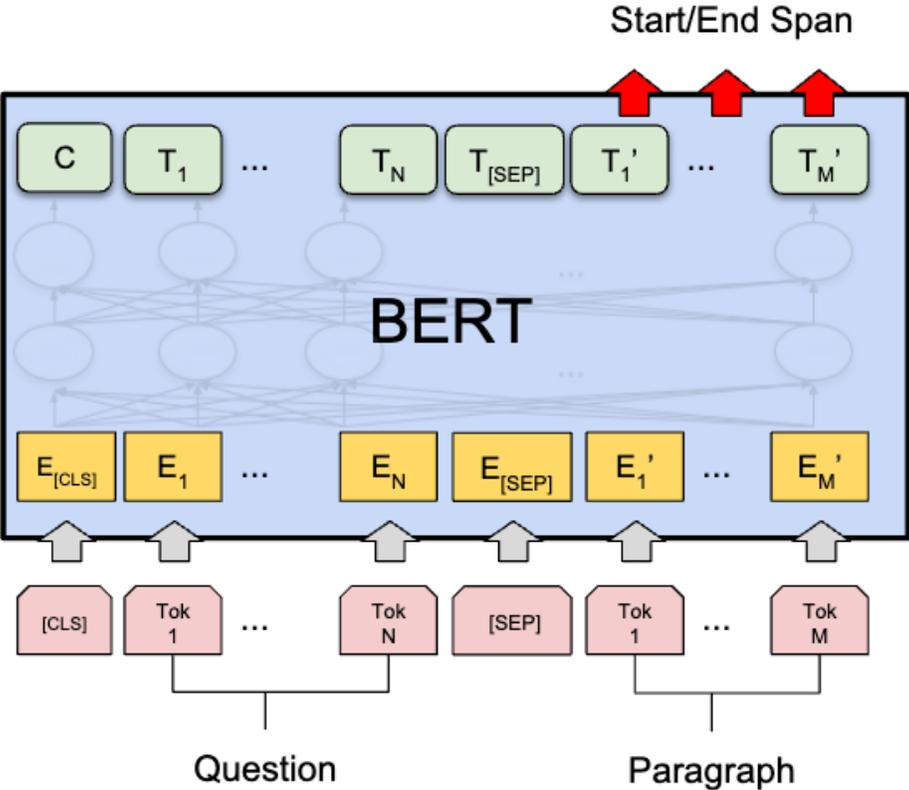
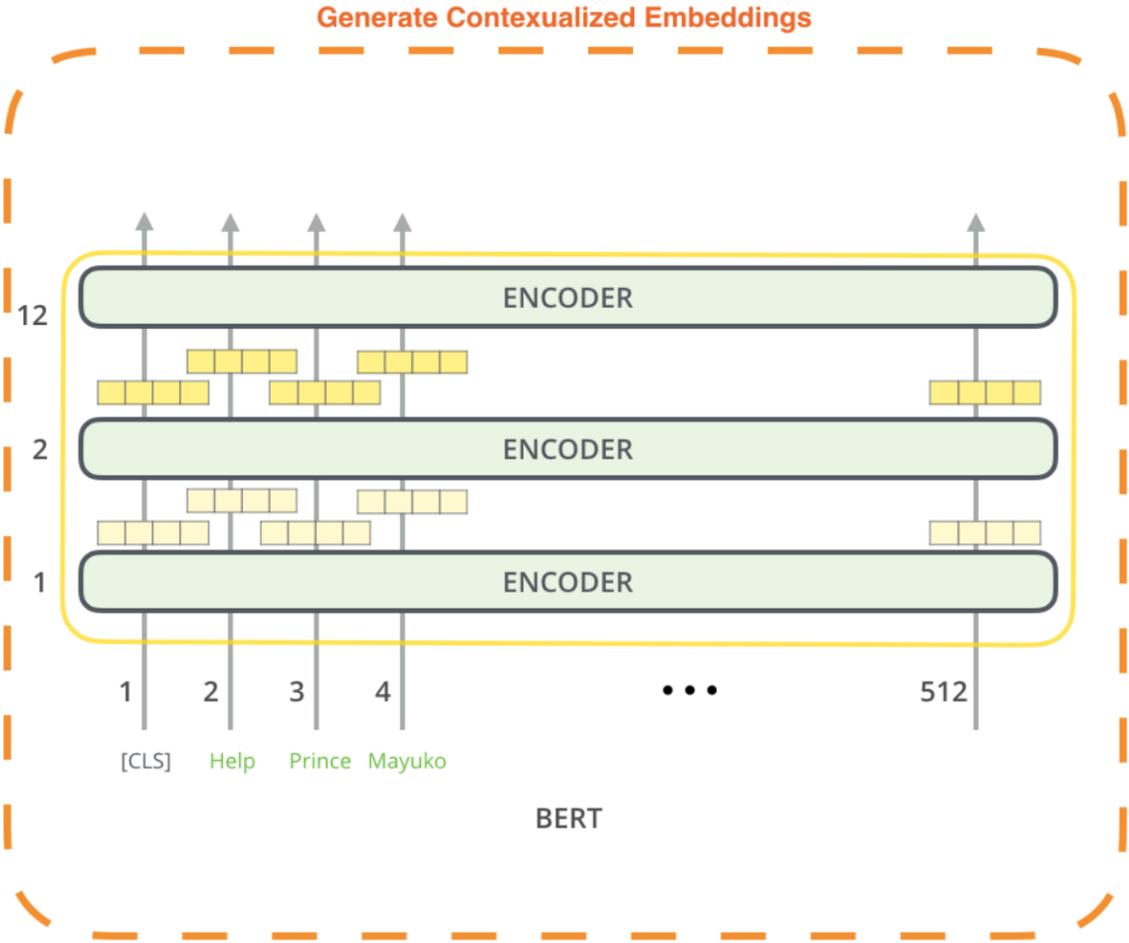BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018

# Fine-Tuning: Sentence-Level Tasks

- Pre-training provides a good weight initialization

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018
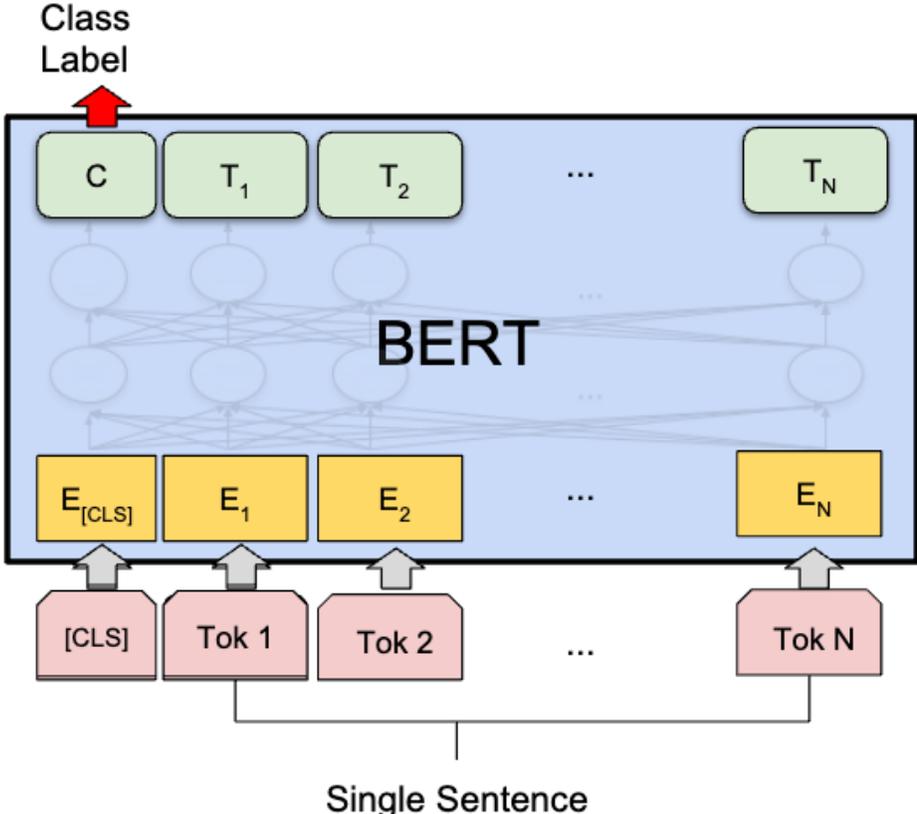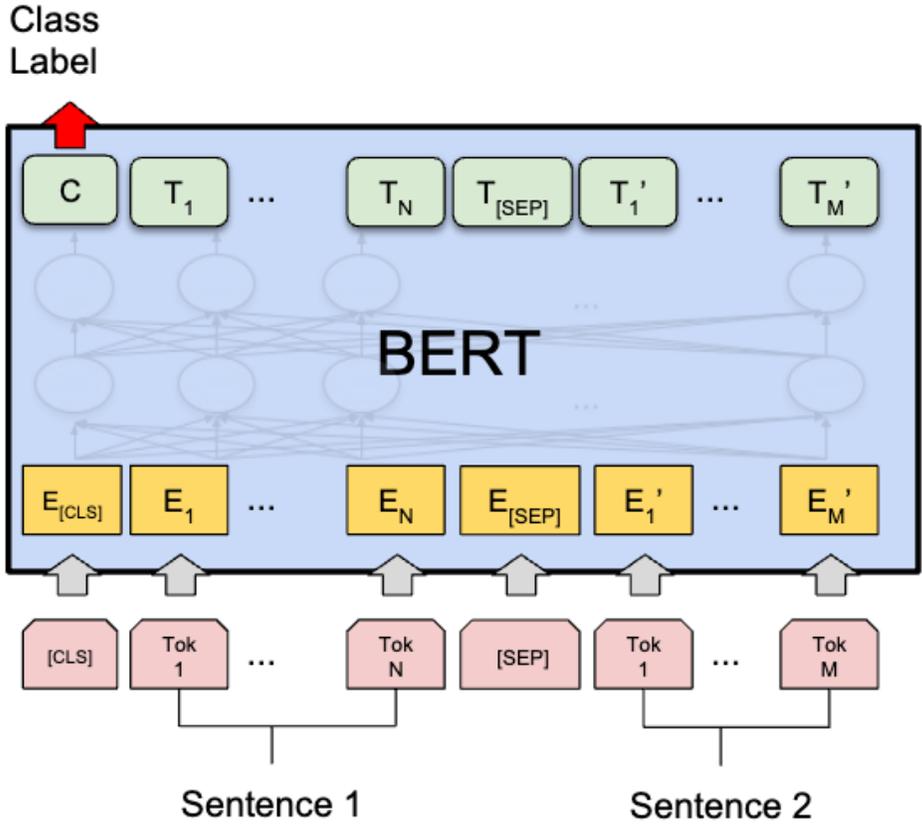
# BERT as General Contextualized Representations



**Generate Contexualized Embeddings**

BERT

The output of each encoder layer along each token's path can be used as a feature representing that token.
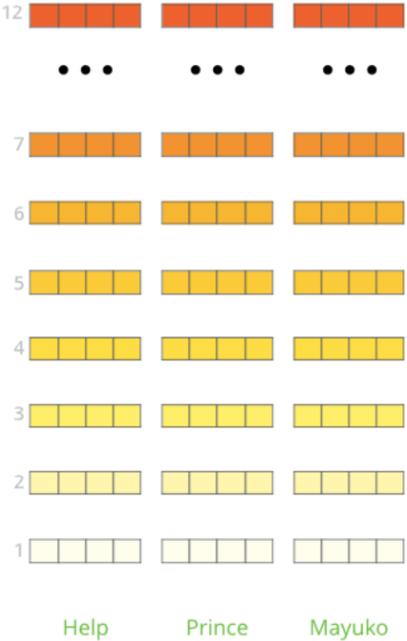
But which one should we use?

# Use BERT



- 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 |
| BERT$_{BASE}$ | 84.6/83.4 | 71.2 | 90.5 | 93.5 | 52.1 | 85.8 | 88.9 | 66.4 | 79.6 |
| BERT$_{LARGE}$ | **86.7/85.9** | **72.1** | **92.7** | **94.9** | **60.5** | **86.5** | **89.3** | **70.1** | **82.1** |

# 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

# Encoder-Only: RoBERTa

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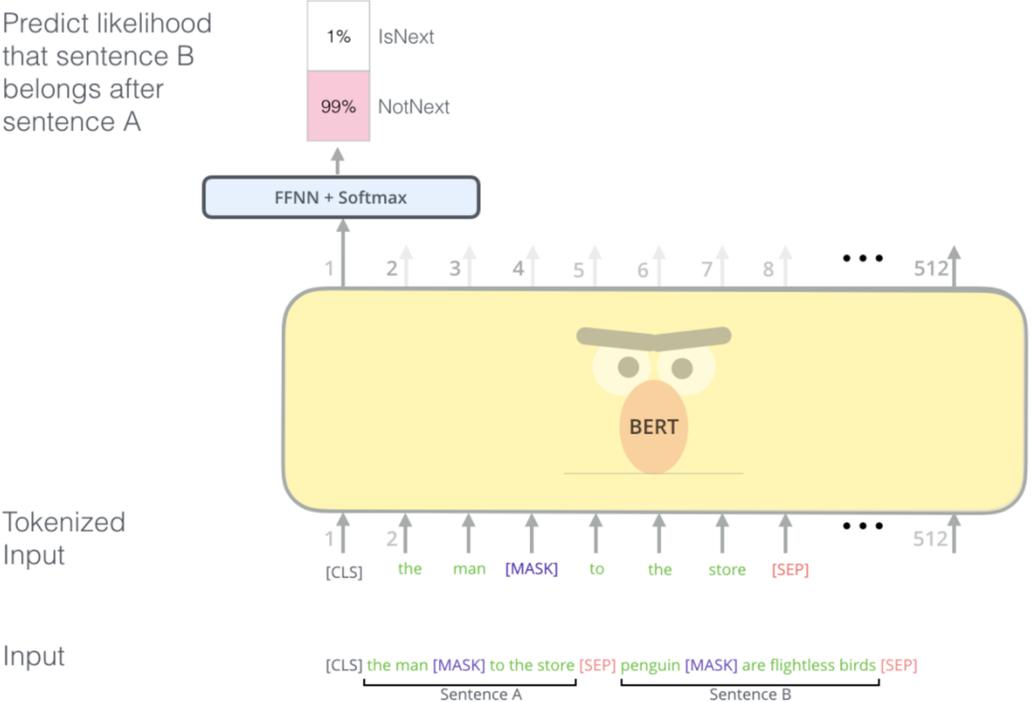- Improve the robustness of training BERT

# Static Masking vs. Dynamic Masking

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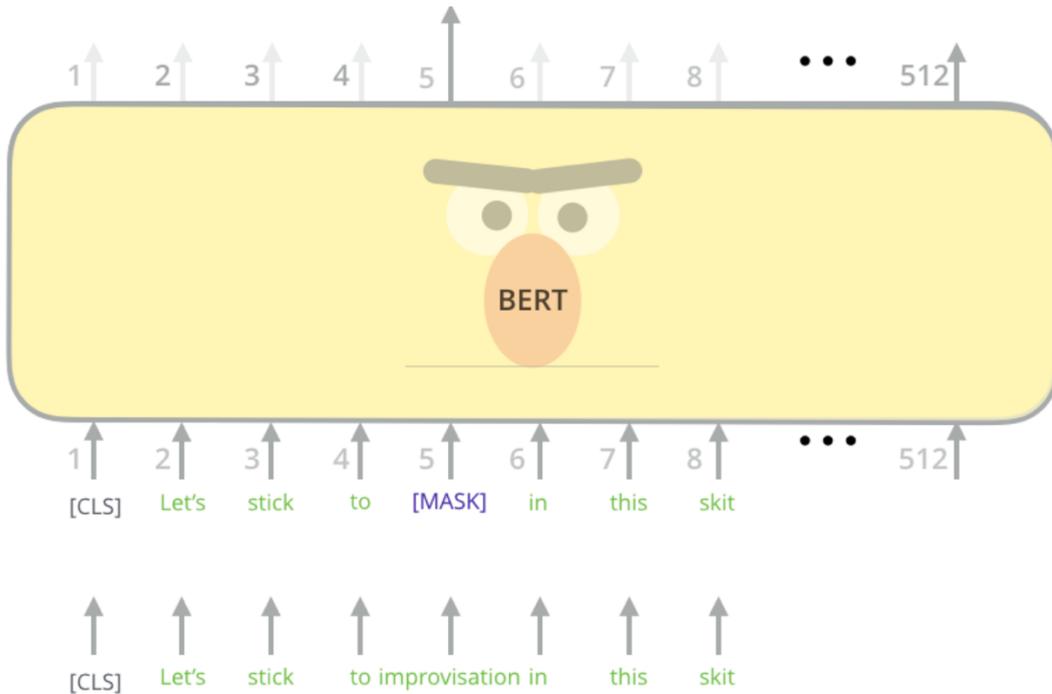


| Masking | SQuAD 2.0 | MNLI-m | SST-2 |
|---------|-----------|--------|-------|
| static  | 78.3      | 84.3   | 92.5  |
| dynamic | 78.7      | 84.0   | 92.9  |

# 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** |
| BERT$_{\text{LARGE}}$ | | | | | | |
| with BOOKS + WIKI | 13GB | 256 | 1M | 90.9/81.8 | 86.6 | 93.7 |

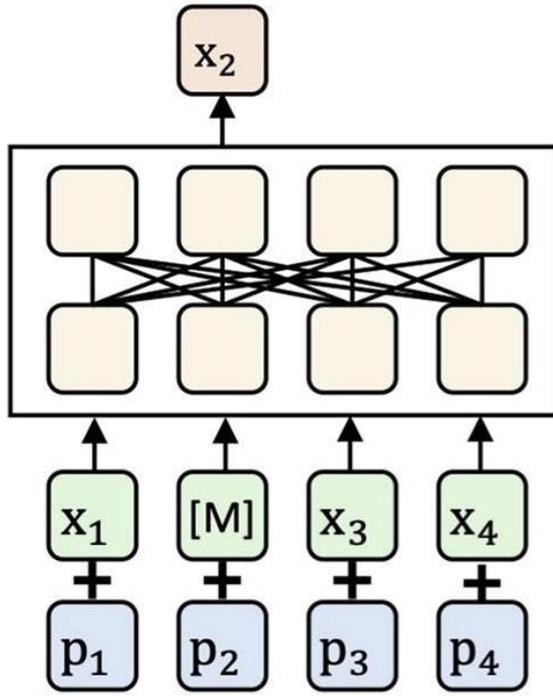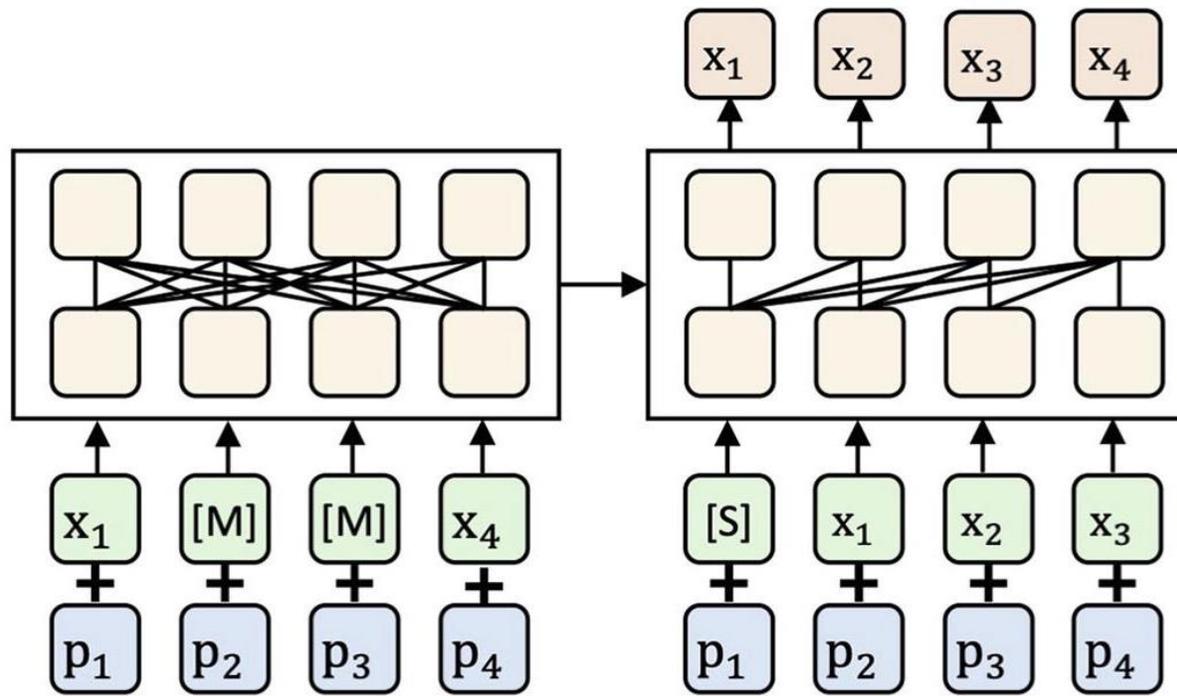# Use RoBERTa



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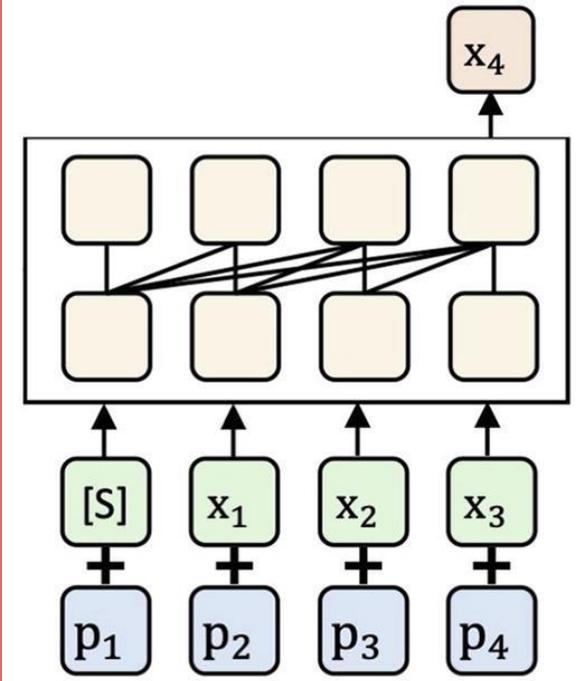  - # parameters ≈ 340M

# Types of Pre-Training



$$\sum_{x_t \in M(\mathbf{x})} P(x_t | \mathbf{x}_{\setminus M(\mathbf{x})})$$

Encoder only

$$\sum_{t=1}^{T} P(x_t | \mathbf{x}_{<t}, \mathbf{x}_{\setminus i:j})$$

Encoder-decoder

$$\sum_{t=1}^{T} P(x_t | \mathbf{x}_{<t})$$

Decoder only

# 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
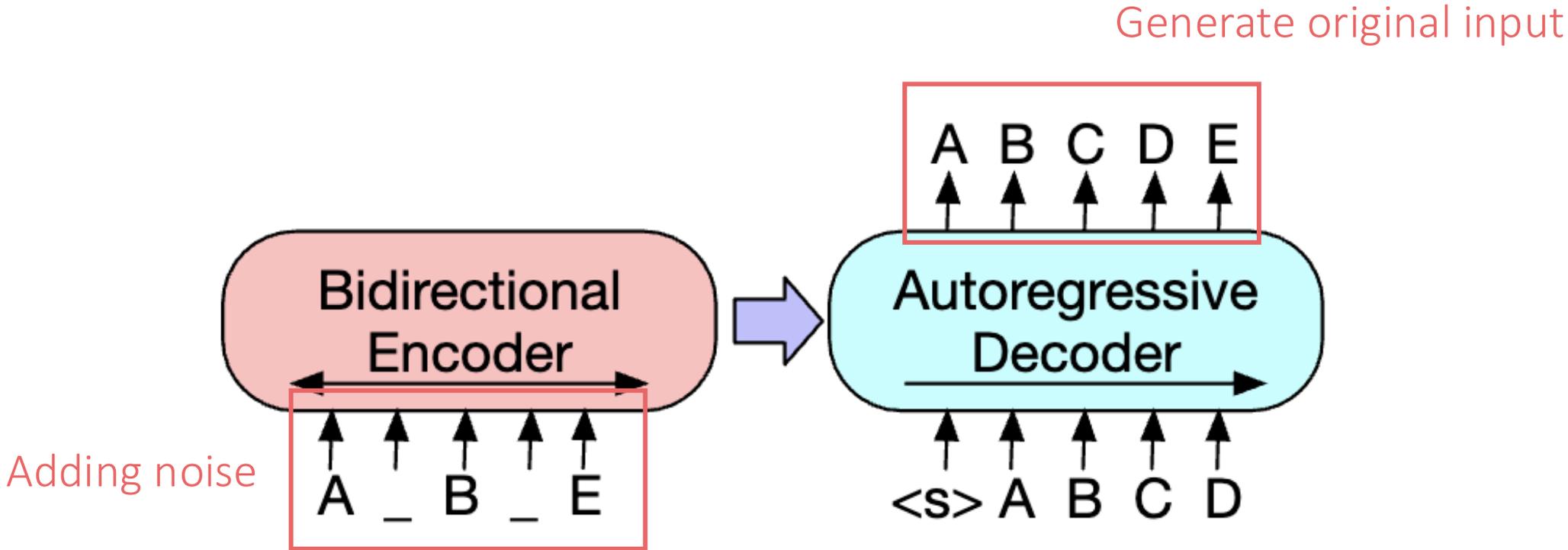{mikelewis,yinhanliu,naman}@fb.com

# Encoder-Decoder: BART

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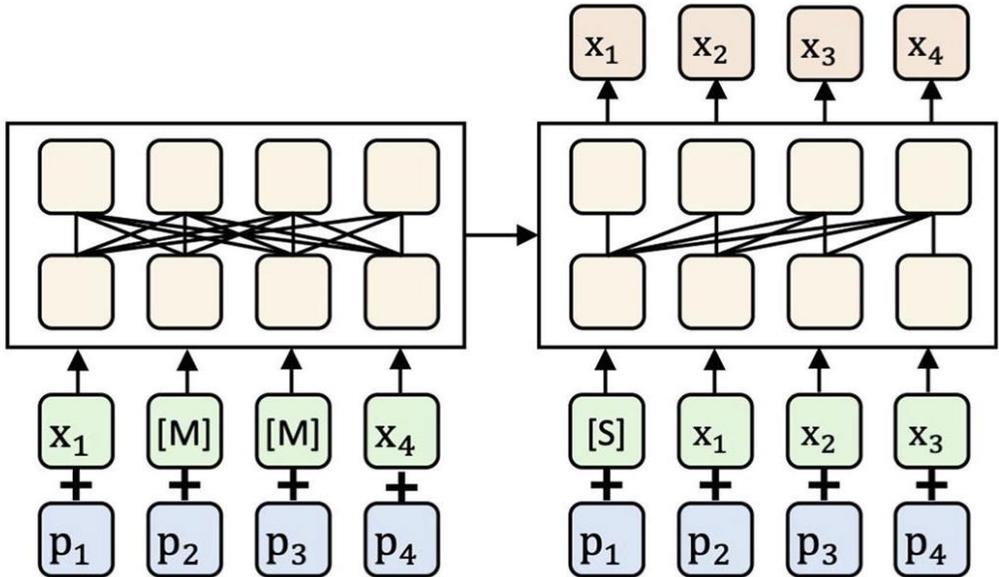


$$\sum_{t=1}^{T} P(x_t | \mathbf{x}_{<t}, \mathbf{x}_{\setminus i:j})$$
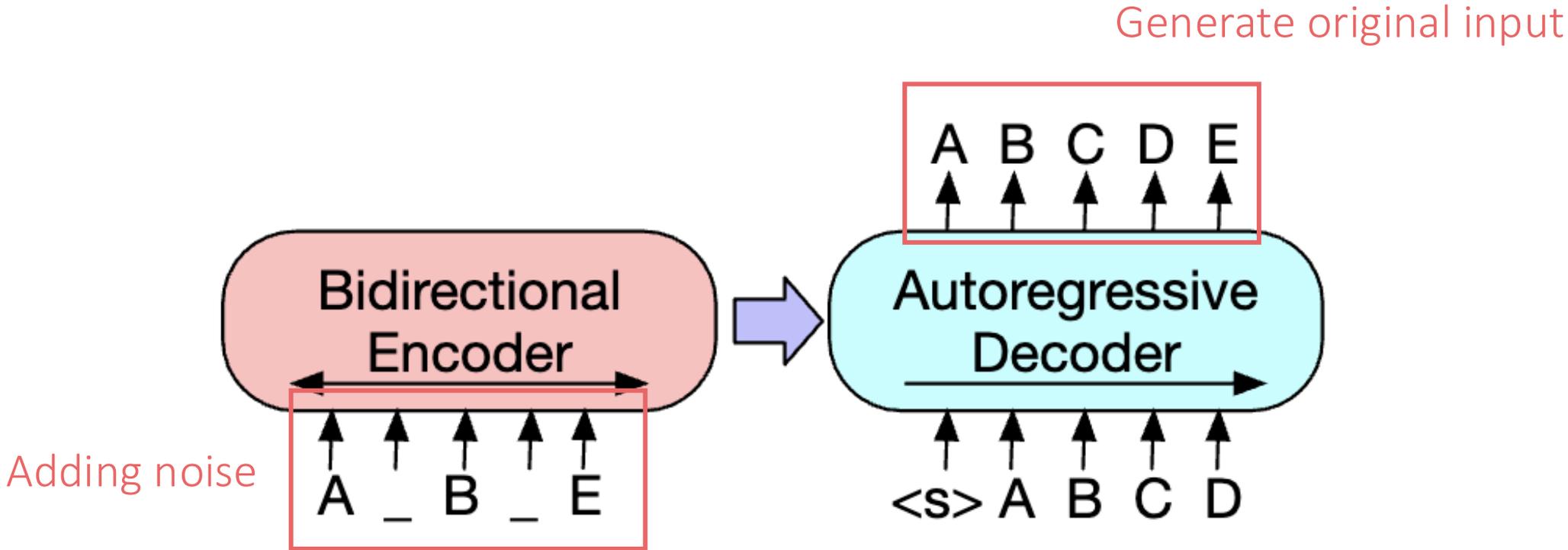
Encoder-decoder

# Denoising Autoencoder



Generate original input

Adding noise
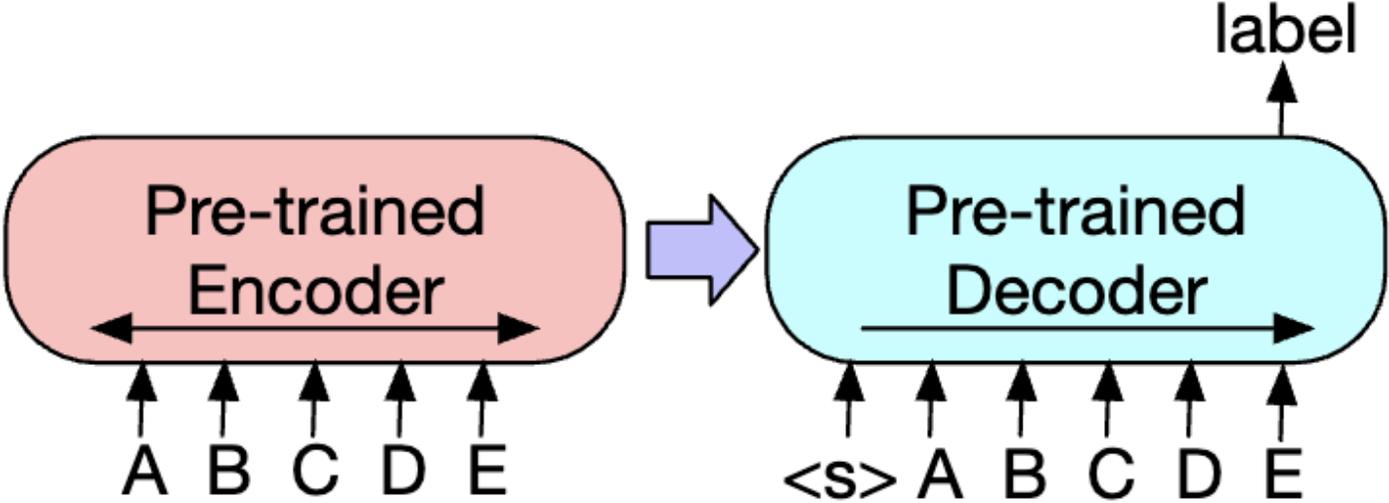
49

# 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.
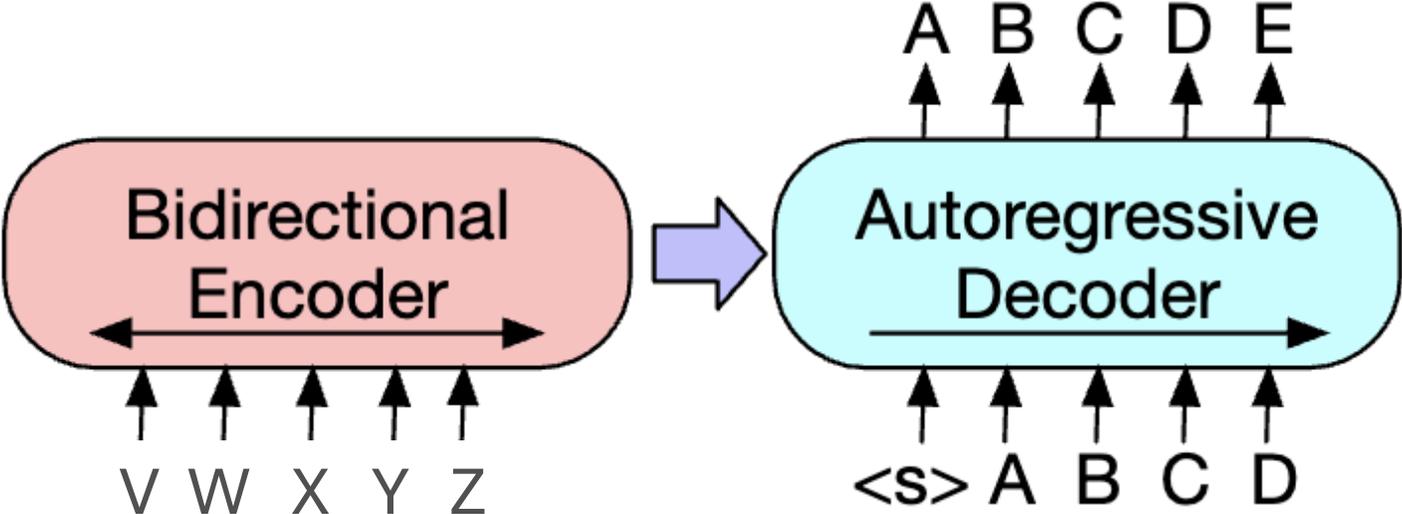- Document Rotation
  - E. FG. ABC. D ➔ ABC. DE. FG.

# Denoising Autoencoder



Generate original input

Adding noise

51

# 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



- 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

# Encoder-Decoder: T5

- Text-to-Text Transfer Transformer (T5)

## Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

| | |
|---|---|
| **Colin Raffel***  | CRAFFEL@GMAIL.COM |
| **Noam Shazeer*** | NOAM@GOOGLE.COM |
| **Adam Roberts*** | ADAROB@GOOGLE.COM |
| **Katherine Lee*** | KATHERINELEE@GOOGLE.COM |
| **Sharan Narang** | SHARANNARANG@GOOGLE.COM |
| **Michael Matena** | MMATENA@GOOGLE.COM |
| **Yanqi Zhou** | YANQIZ@GOOGLE.COM |
| **Wei Li** | MWEILI@GOOGLE.COM |
| **Peter J. Liu** | PETERJLIU@GOOGLE.COM |

# Motivation: BART
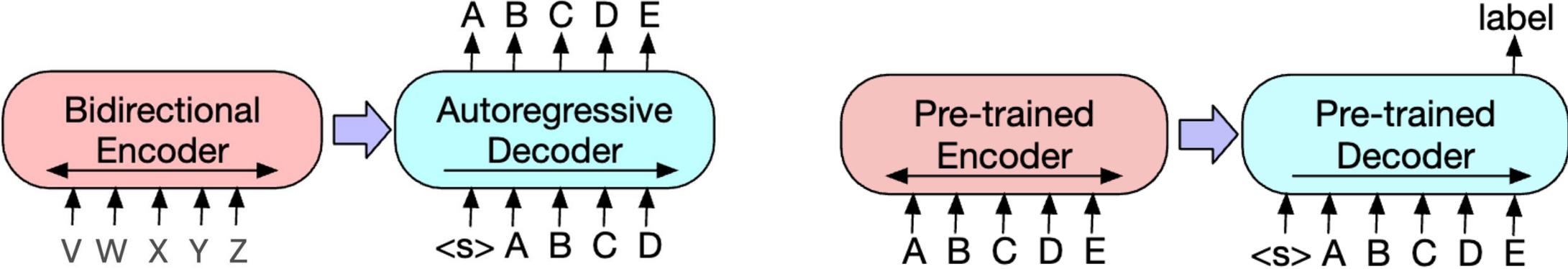


Different ways when considering classification and seq2seq generation

# Convert Everything to Text-to-Text Tasks
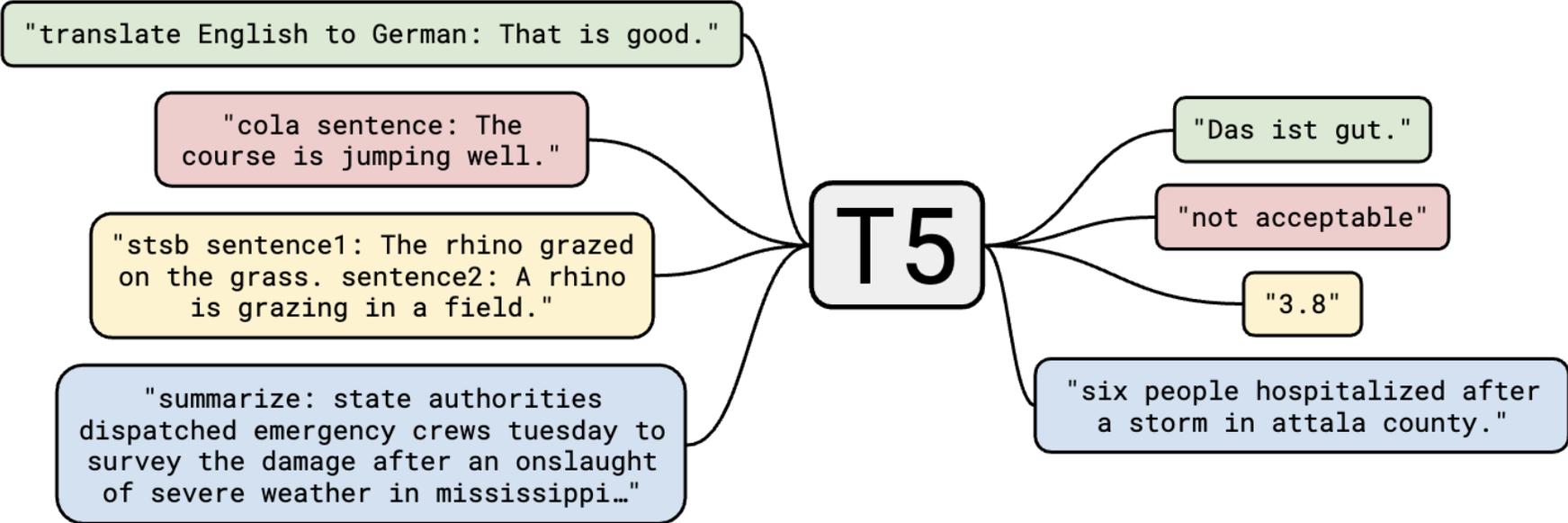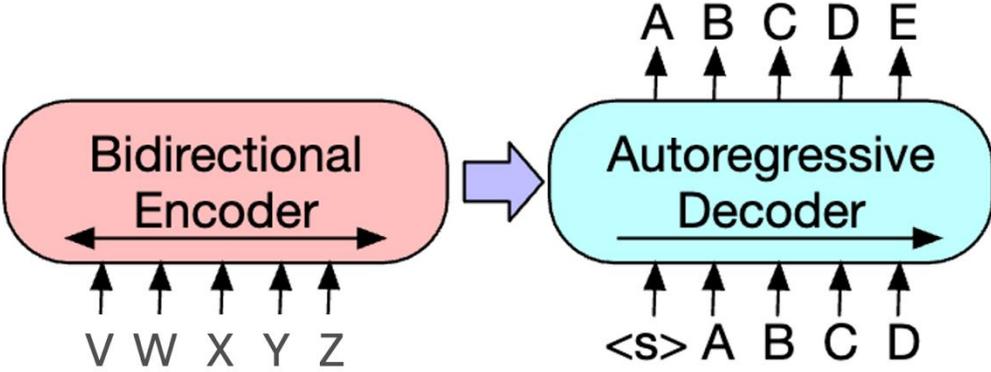
# Masked Span Reconstruction (Seq2Seq Version)

Original text

Thank you ~~for inviting~~ me to your party ~~last~~ week.

Inputs

Thank you <X> me to your party <Y> week.

Targets

<X> for inviting <Y> last <Z>

# Multi-Task Learning

- Convert everything to text-to-text tasks
- Jointly fine-tune them together

# Multi-Task Learning

## D.7 SST2

**Original input:**

> **Sentence:** it confirms fincher 's status as a film maker who artfully bends technical know-how to the service of psychological insight .

**Processed input:** sst2 sentence: it confirms fincher 's status as a film maker who artfully bends technical know-how to the service of psychological insight .

**Original target:** 1

**Processed target:** positive

# Multi-Task Learning

**D.4 MRPC**

**Original input:**

**Sentence 1:** We acted because we saw the existing evidence in a new light , through the prism of our experience on 11 September , " Rumsfeld said .

**Sentence 2:** Rather , the US acted because the administration saw " existing evidence in a new light , through the prism of our experience on September 11 " .

**Processed input:** mrpc sentence1: We acted because we saw the existing evidence in a new light , through the prism of our experience on 11 September , " Rumsfeld said . sentence2: Rather , the US acted because the administration saw " existing evidence in a new light , through the prism of our experience on September 11 " .

**Original target:** 1

**Processed target:** equivalent

# Multi-Task Learning

## D.16 WMT English to German

**Original input:** "Luigi often said to me that he never wanted the brothers to end up in court," she wrote.

**Processed input:** translate English to German: "Luigi often said to me that he never wanted the brothers to end up in court," she wrote.
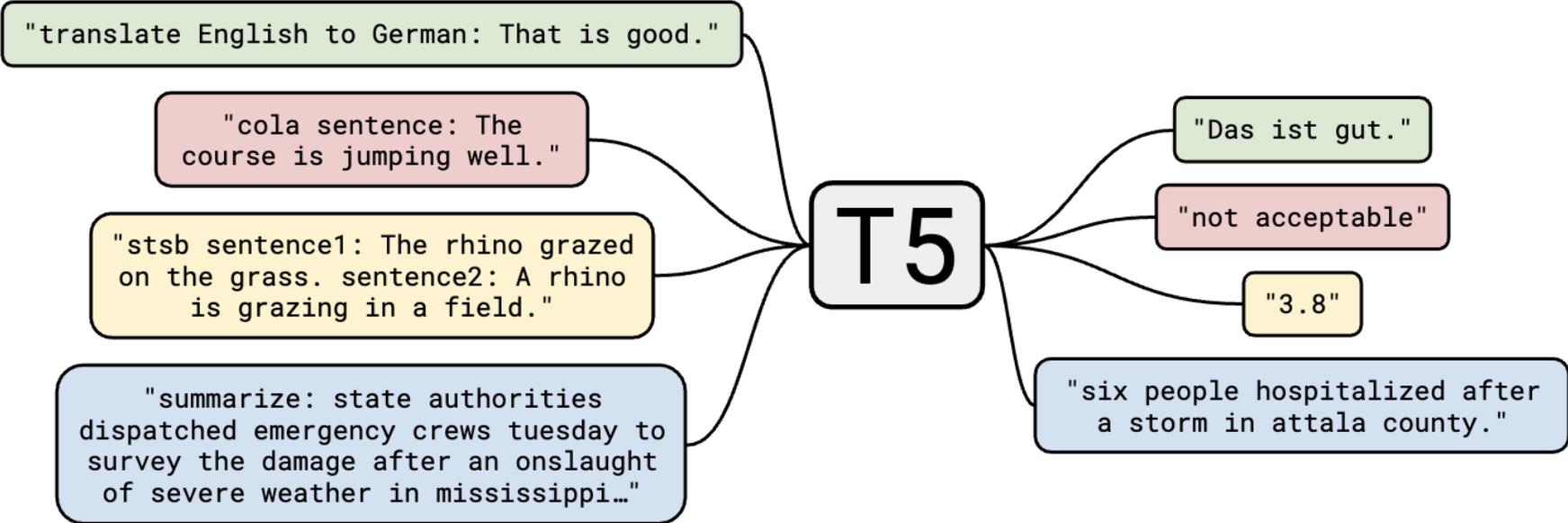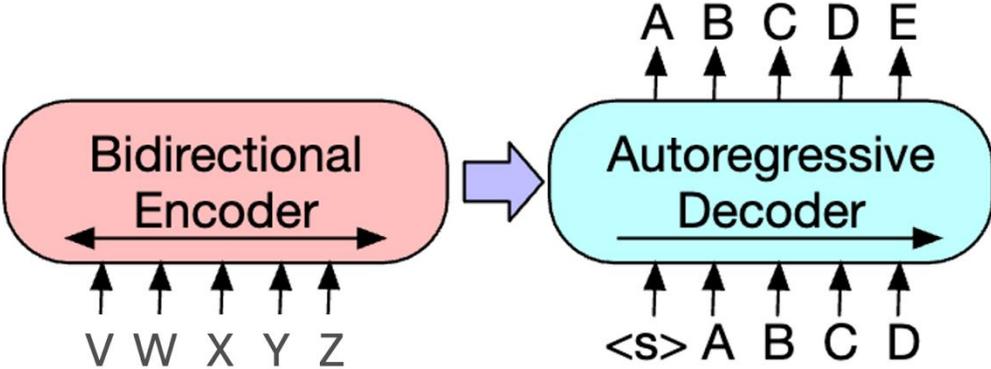
**Original target:** "Luigi sagte oft zu mir, dass er nie wollte, dass die Brüder vor Gericht landen", schrieb sie.

**Processed target:** "Luigi sagte oft zu mir, dass er nie wollte, dass die Brüder vor Gericht landen", schrieb sie.

# Relative Position
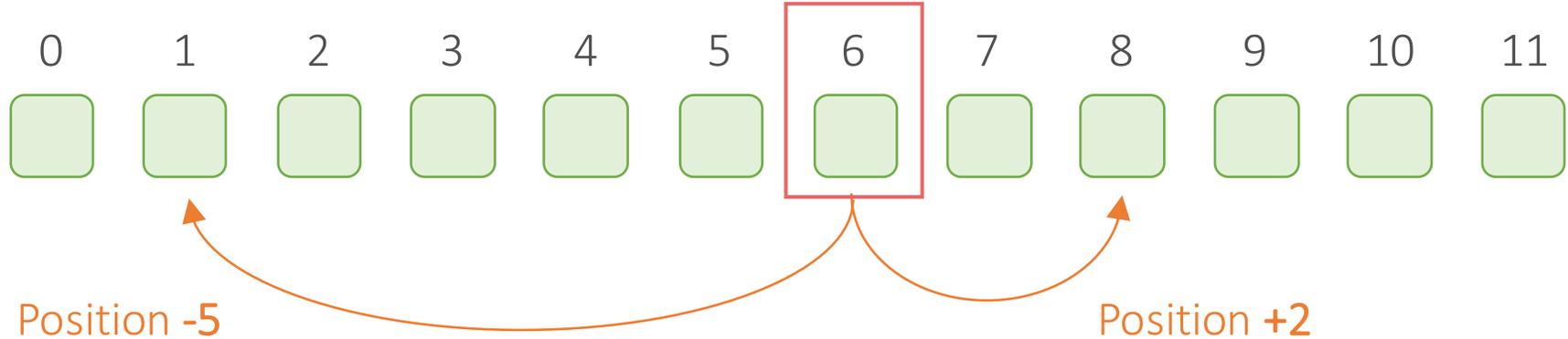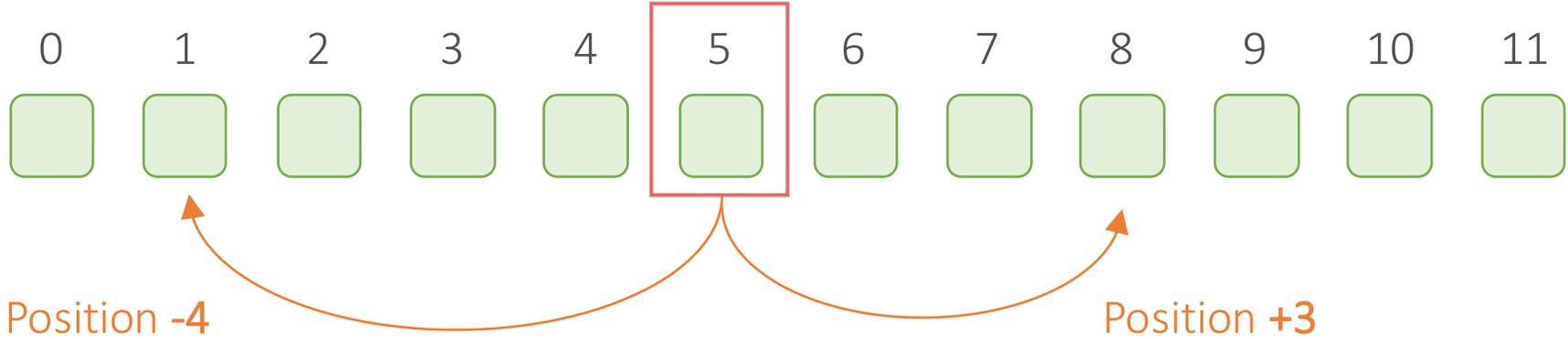
# Fine-Tuning: Text-to-Text For Everything

# Promising Results

| Model | QQP F1 | QQP Accuracy | MNLI-m Accuracy | MNLI-mm Accuracy | QNLI Accuracy | RTE Accuracy | WNLI Accuracy |
|---|---|---|---|---|---|---|---|
| Previous best | $74.8^c$ | $\mathbf{90.7}^b$ | $91.3^a$ | $91.0^a$ | $\mathbf{99.2}^a$ | $89.2^a$ | $91.8^a$ |
| T5-Small | 70.0 | 88.0 | 82.4 | 82.3 | 90.3 | 69.9 | 69.2 |
| T5-Base | 72.6 | 89.4 | 87.1 | 86.2 | 93.7 | 80.1 | 78.8 |
| T5-Large | 73.9 | 89.9 | 89.9 | 89.6 | 94.8 | 87.2 | 85.6 |
| T5-3B | 74.4 | 89.7 | 91.4 | 91.2 | 96.3 | 91.1 | 89.7 |
| T5-11B | **75.1** | 90.6 | **92.2** | **91.9** | 96.9 | **92.8** | **94.5** |

| Model | SQuAD EM | SQuAD F1 | SuperGLUE Average | BoolQ Accuracy | CB F1 | CB Accuracy | COPA Accuracy |
|---|---|---|---|---|---|---|---|
| Previous best | $90.1^a$ | $95.5^a$ | $84.6^d$ | $87.1^d$ | $90.5^d$ | $95.2^d$ | $90.6^d$ |
| T5-Small | 79.10 | 87.24 | 63.3 | 76.4 | 56.9 | 81.6 | 46.0 |
| T5-Base | 85.44 | 92.08 | 76.2 | 81.4 | 86.2 | 94.0 | 71.2 |
| T5-Large | 86.66 | 93.79 | 82.3 | 85.4 | 91.6 | 94.8 | 83.4 |
| T5-3B | 88.53 | 94.95 | 86.4 | 89.9 | 90.3 | 94.4 | 92.0 |
| T5-11B | **91.26** | **96.22** | **88.9** | **91.2** | **93.9** | **96.8** | **94.8** |

| Model | MultiRC F1a | MultiRC EM | ReCoRD F1 | ReCoRD Accuracy | RTE Accuracy | WiC Accuracy | WSC Accuracy |
|---|---|---|---|---|---|---|---|
| Previous best | $84.4^d$ | $52.5^d$ | $90.6^d$ | $90.0^d$ | $88.2^d$ | $69.9^d$ | $89.0^d$ |
| T5-Small | 69.3 | 26.3 | 56.3 | 55.4 | 73.3 | 66.9 | 70.5 |
| T5-Base | 79.7 | 43.1 | 75.0 | 74.2 | 81.5 | 68.3 | 80.8 |
| T5-Large | 83.3 | 50.7 | 86.8 | 85.9 | 87.8 | 69.3 | 86.3 |
| T5-3B | 86.8 | 58.3 | 91.2 | 90.4 | 90.7 | 72.1 | 90.4 |
| T5-11B | **88.1** | **63.3** | **94.1** | **93.4** | **92.5** | **76.9** | **93.8** |

# Use T5
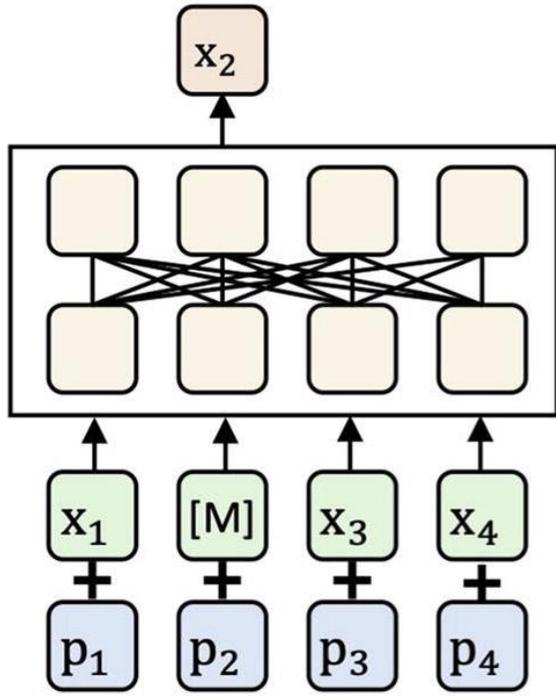


- T5-small:
  - # parameters ≈ 60M
- T5-base:
  - # parameters ≈ 220M
- T5-large:
  - # parameters ≈ 770M
- T5-3B: #
  - parameters ≈ 3B
- T5-11B:
  - # parameters ≈ 11B
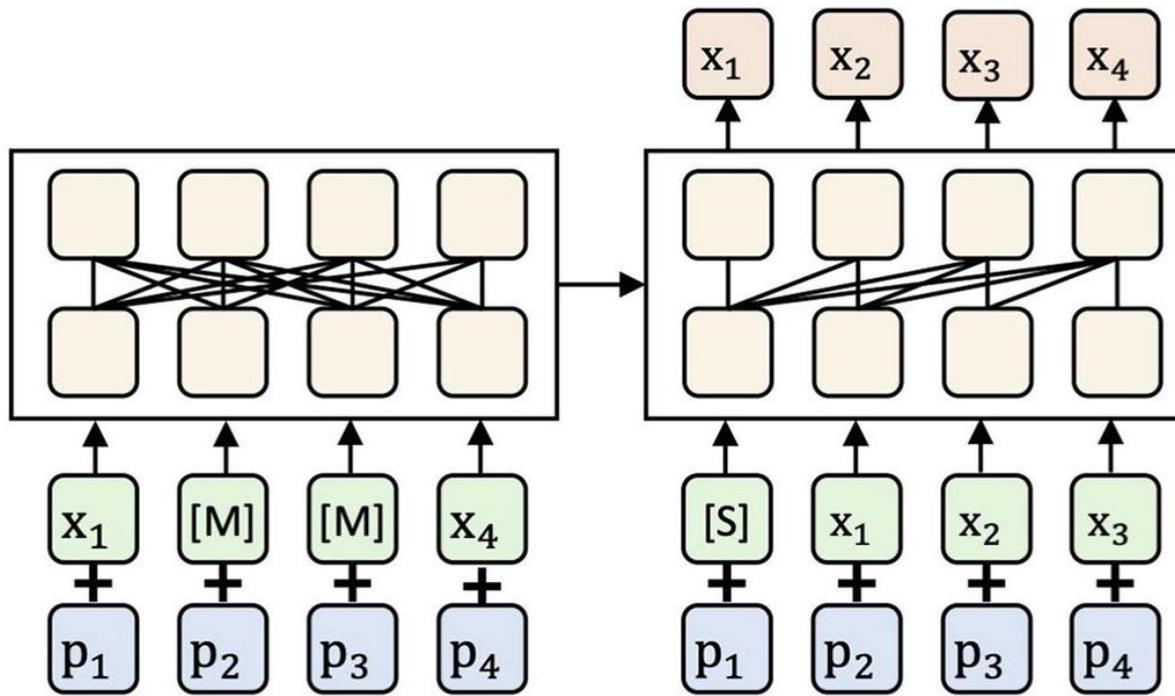
# Lecture Plan

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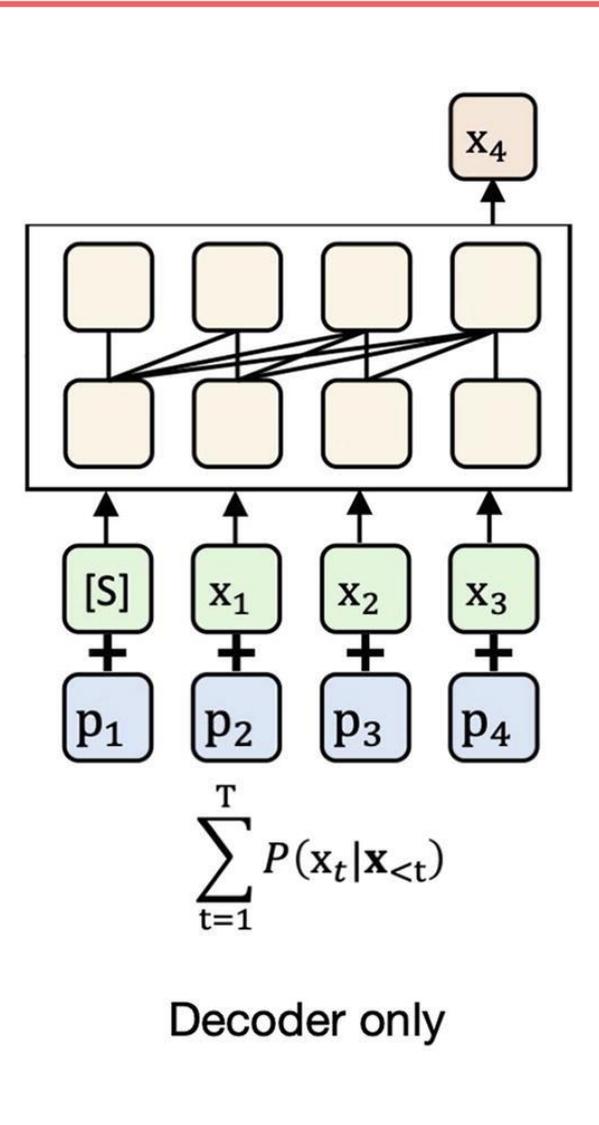  - Decoder-Only Pre-Training

# Types of Pre-Training



$$\sum_{x_t \in M(\mathbf{x})} P\big(x_t | \mathbf{x}_{\backslash M(\mathbf{x})}\big)$$

Encoder only

$$\sum_{t=1}^{T} P\big(x_t | \mathbf{x}_{<t}, \mathbf{x}_{\backslash i:j}\big)$$

Encoder-decoder

$$\sum_{t=1}^{T} P\big(x_t | \mathbf{x}_{<t}\big)$$

Decoder only

70

# Language Modeling

- Next word prediction
- Trained with large corpus
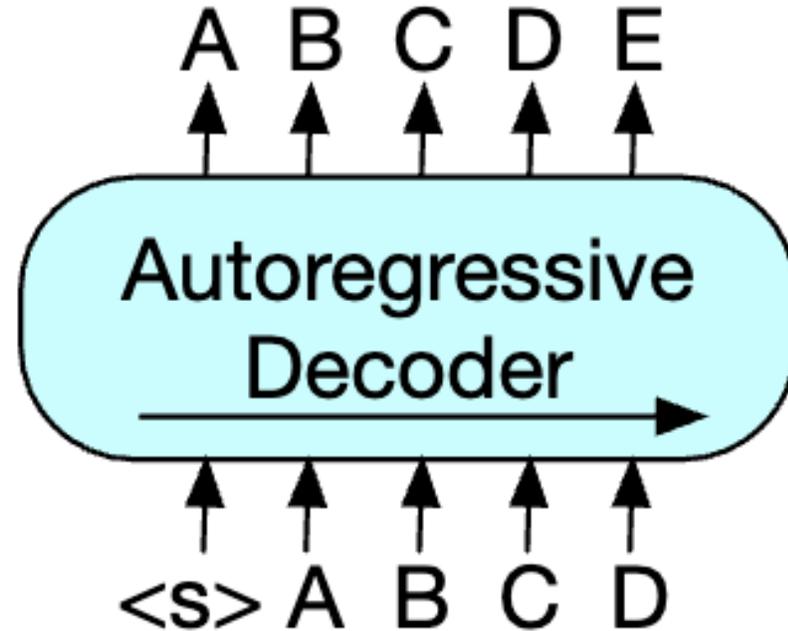
# Comparison: Masked Language Models



Use the output of the masked word's position to predict the masked word

Possible classes: All English words

| | |
|---|---|
| 0.1% | Aardvark |
| ... | ... |
| 10% | Improvisation |
| ... | ... |
| 0% | Zyzzyva |

FFNN + Softmax

1  2  3  4  5  6  7  8  •••  512

BERT

Randomly mask 15% of tokens

1  2  3  4  5  6  7  8  •••  512

[CLS]  Let's  stick  to  [MASK]  in  this  skit

Input

[CLS]  Let's  stick  to improvisation in  this  skit
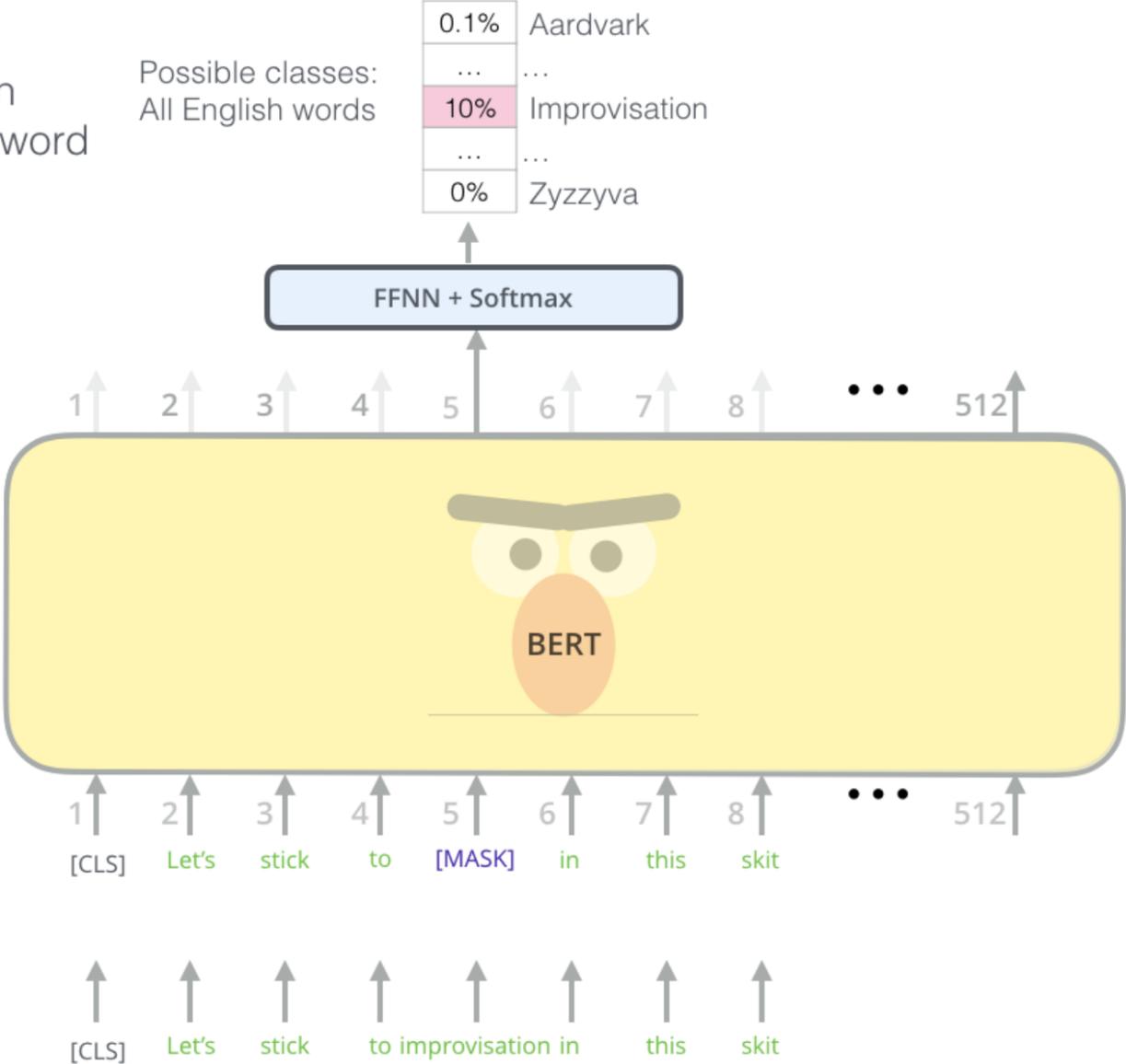
# Comparison: Causal Language Models
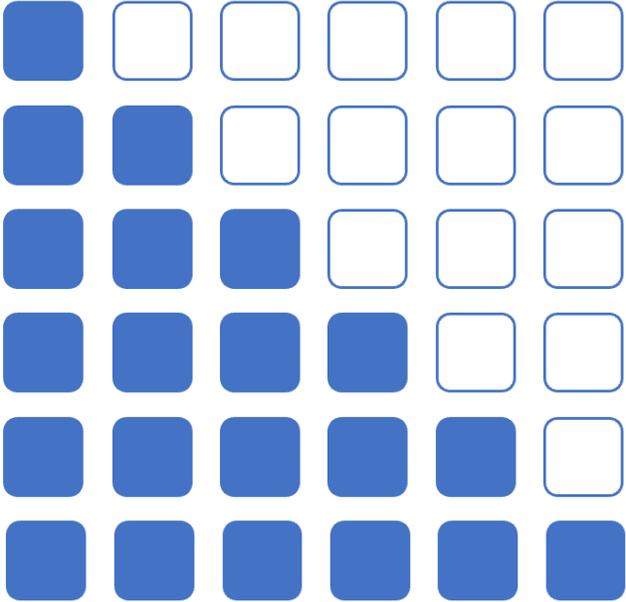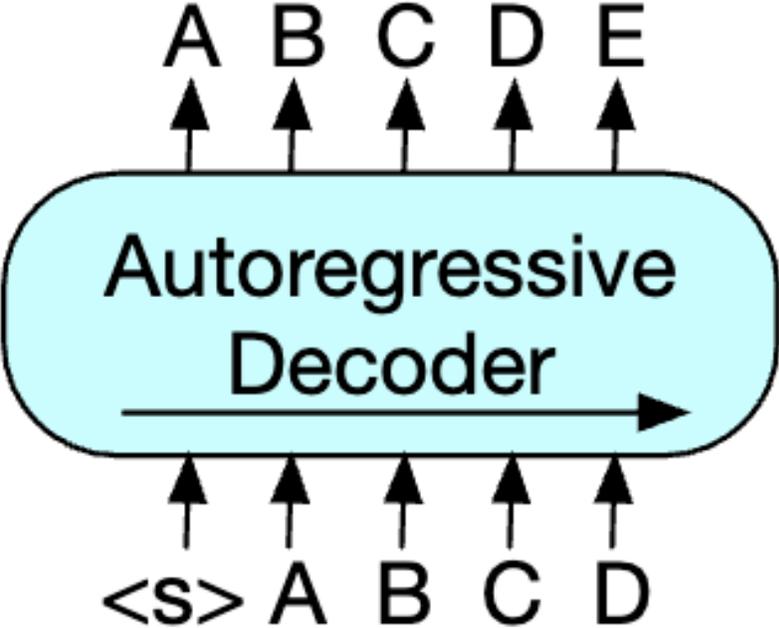


Causal Masking

# Decoder-Only: Generative Pre-trained Transformer (GPT)

- Improving Language Understanding by Generative Pre-Training, OpenAI 2018
  - **G**enerative **P**re-trained **T**ransformer (GPT)
- Language Models are Unsupervised Multitask Learners, OpenAI 2019
  - GPT-2
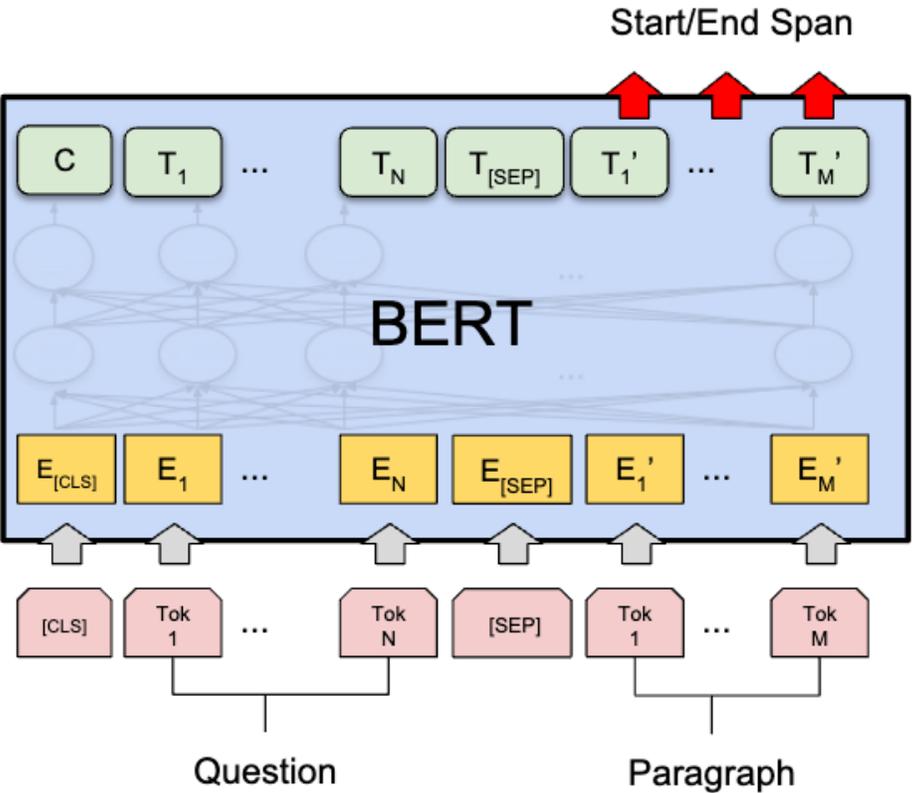- Language Models are Few-Shot Learners, OpenAI 2020
  - GPT-3
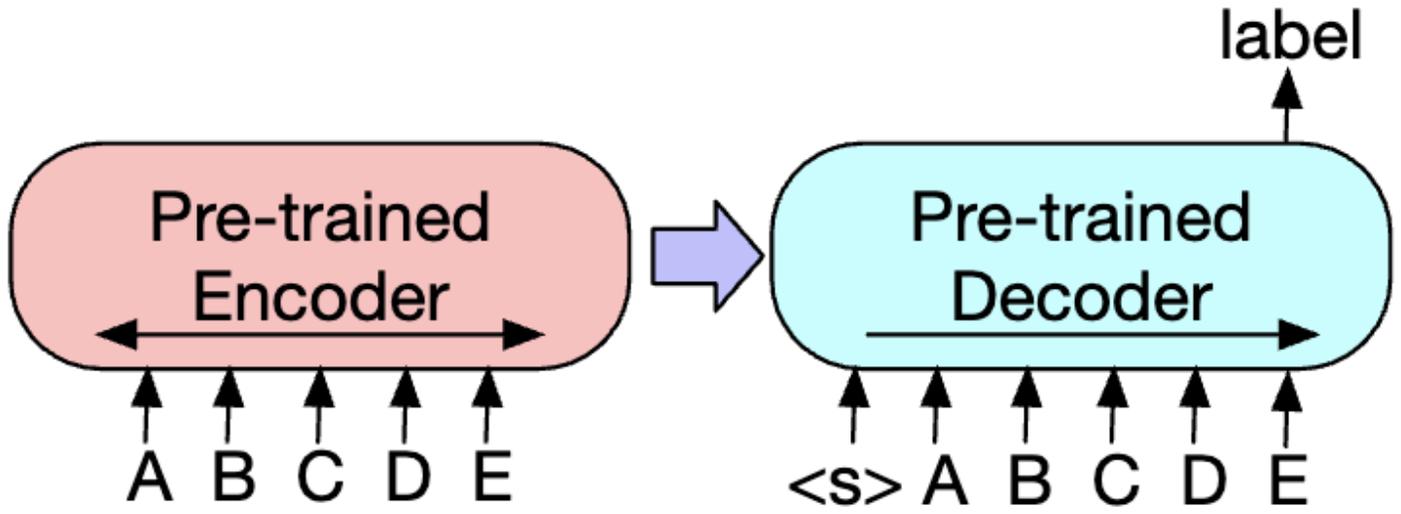
# GPT-3

- Even larger training data, even larger model size
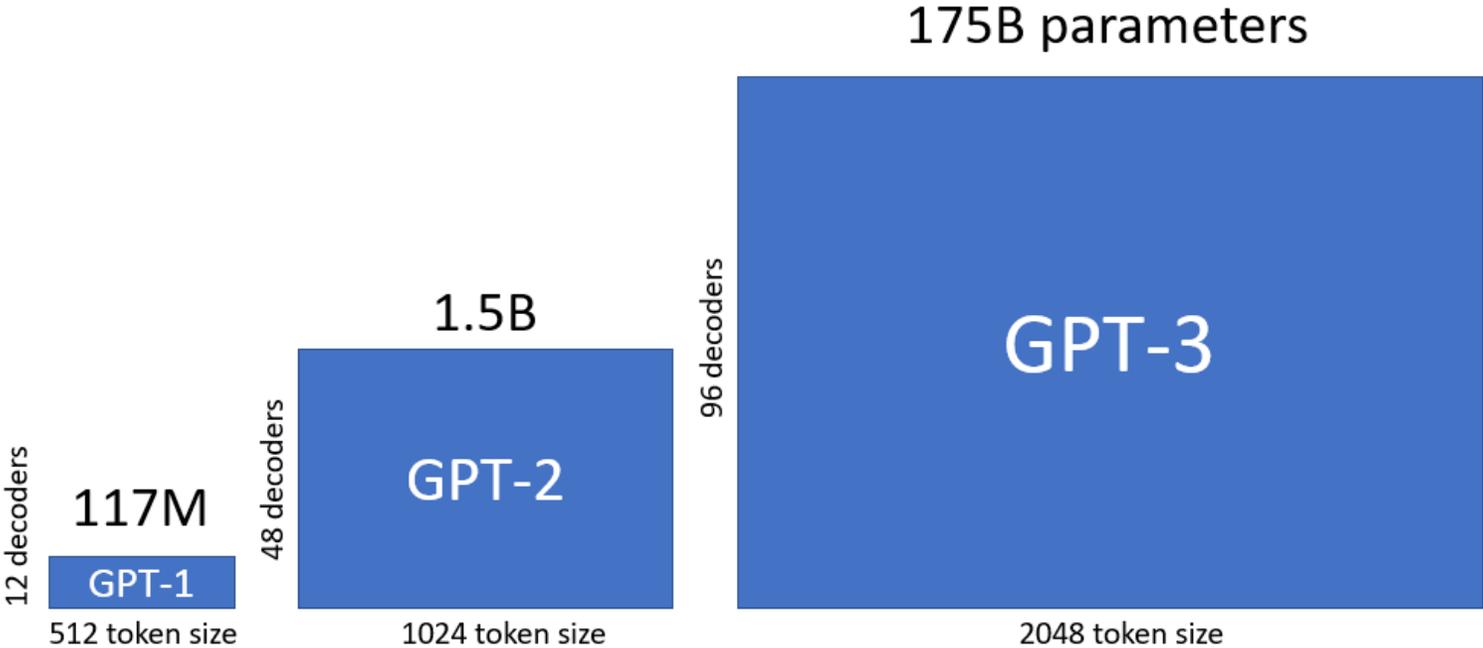
# From Text Representations to Text Generation

- BERT provides a good weight initialization (text representations) for fine-tuning classification tasks
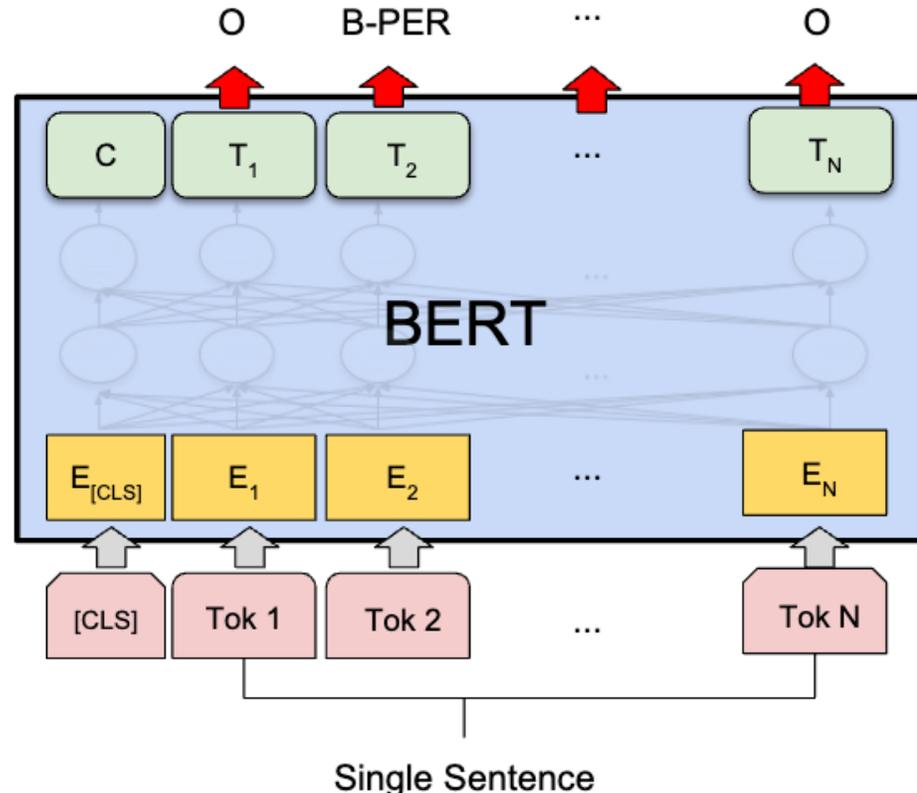
# From Text Representations to Text Generation

- BART provides a good weight initialization (text representations) for fine-tuning classification tasks

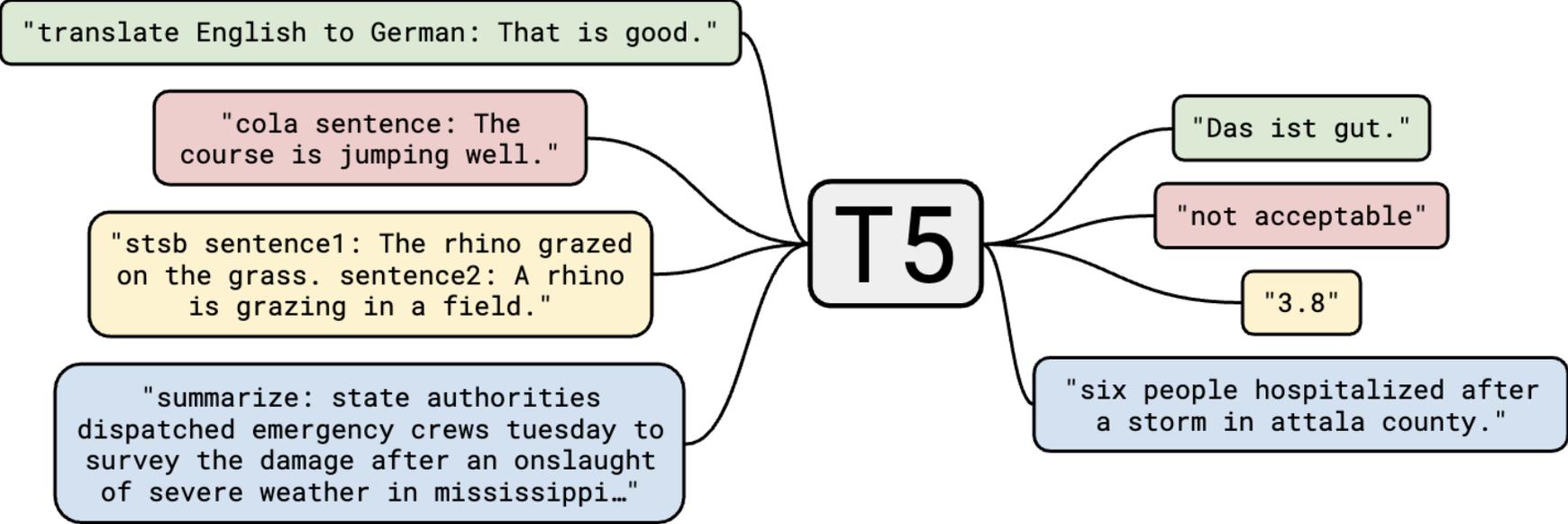# From Text Representations to Text Generation

- BART also provides a good weight initialization for fine-tuning generation tasks
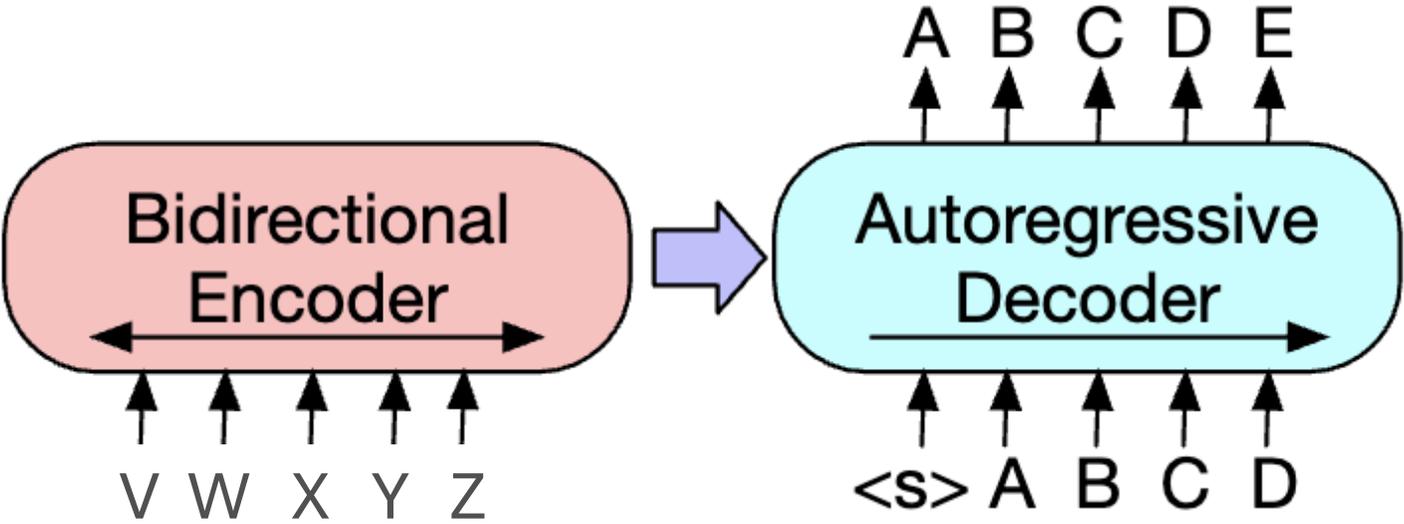
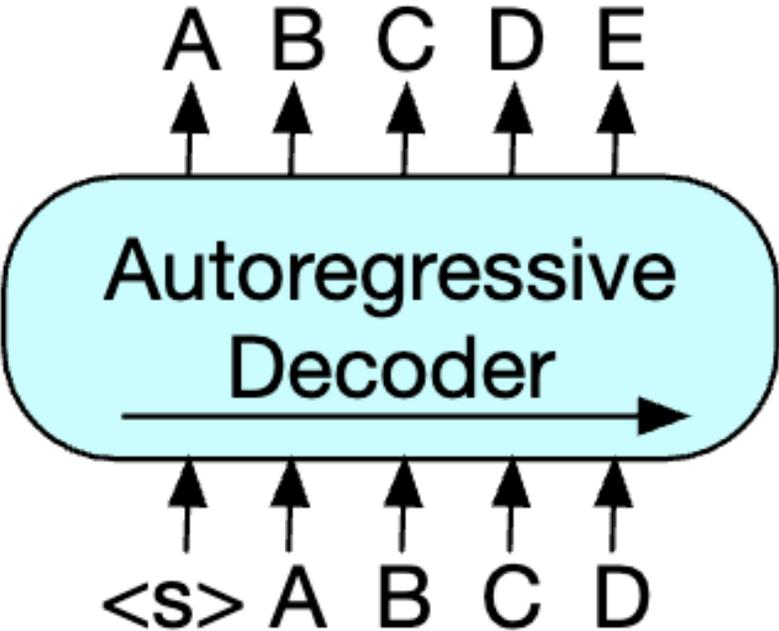# From Text Representations to Text Generation

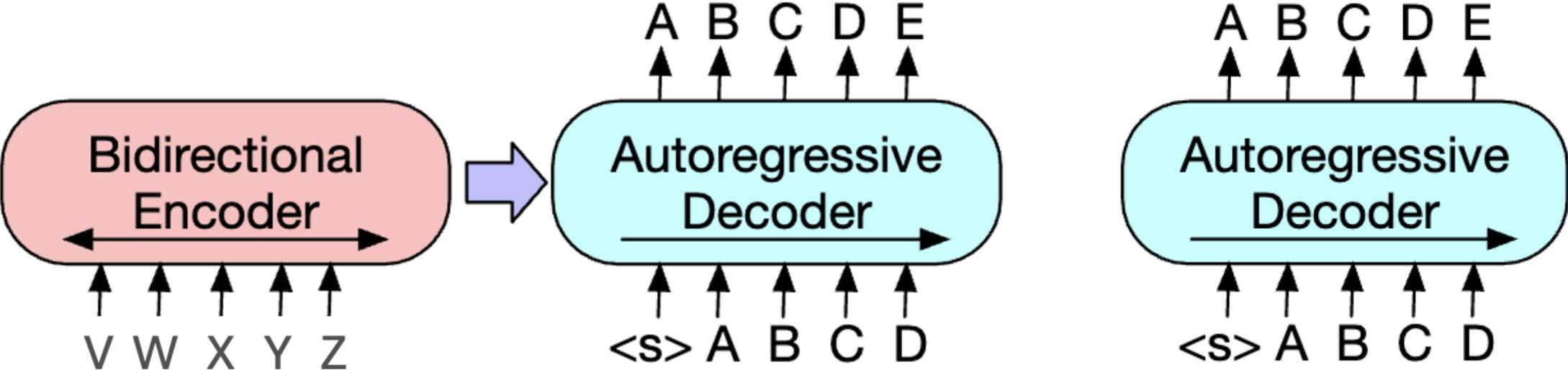- T5 provides a good weight initialization for fine-tuning both classification and generation tasks via generation framework

# From Text Representations to Text Generation

- GPT provides a good weight initialization for fine-tuning both classification and generation tasks via generation framework
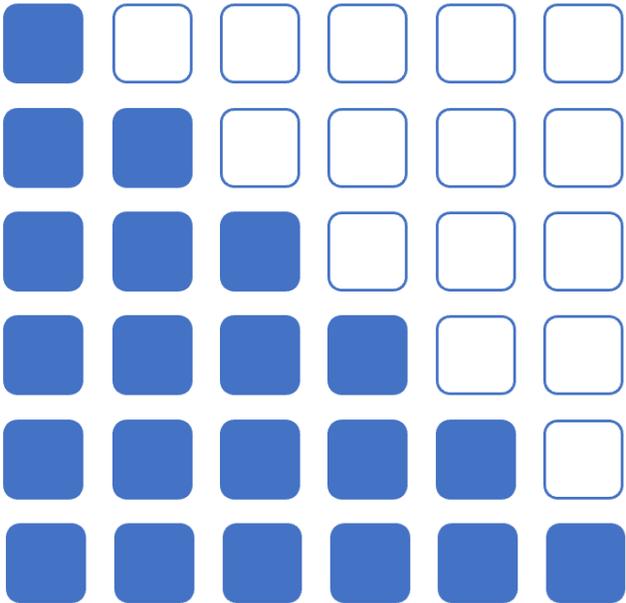


Causal Masking

# Encoder-Decoder vs. Decoder Only

# Large Language Models