

## 01. BERT

Why was BERT way ahead of its time?

- 
- 
- 

Because it was a masked language model even during pre-covid days!



# CS 1671/2071

## Human Language Technologies

Session 13: BERT

---

Michael Miller Yoder

March 24, 2025

# Course logistics

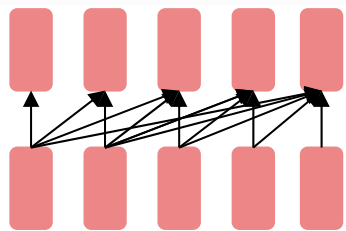
- Project progress report is **due this Thu Mar 27**. See the [project website](#) for instructions
  - **Part 1:** Data statistics and exploratory data analysis (EDA)
  - **Part 2:** A result from baseline/initial approach
  - **Part 3:** Proposal on how to use LLMs for your task
  - **Part 4:** Open questions and challenges
- I will let you know when we have a class OpenAI API account to use (\$150 total). In the meantime look into using Gemini free credits or other LLMs
- In-person exam will be **next Wed Apr 2**
  - Review session is next Mon Mar 25 during class
- Homework 3 will be released today or tomorrow. Will be on LLM prompting and will be due Apr 10

# Lecture overview: BERT

- Notebook from last time: finetuning GPT-2 on Shakespeare plays
- Subword tokenization
- BERT and masked language modeling
- Finetuning BERT for classification and sequence labeling
- Notebook for this time: finetuning BERT for text classification

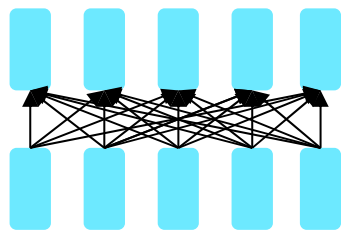
*Review:* Describe encoder, decoder, and encoder-decoder architectures

# Three architectures for large language models



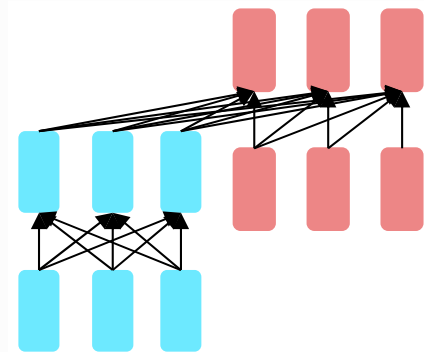
Decoders

GPT, Claude,  
Llama, Mixtral



Encoders

BERT family,  
RoBERTa



Encoder-decoders

Flan-T5, Whisper

# Notebook from last time: finetune GPT-2 on Shakespeare

- [Click on this nbgitpuller link](#) or find the link on the course website
- **Important difference from normal:** Open a 'Teach – 1 gpu, 3 hours' server

**Server Options**

Select a job profile:

Teach - 6 cores, 3 hours

Teach - 6 cores, 3 hours

Teach - 1 gpu, 3 hours

A100 - 1 gpu, 3 hours

SMP - 4 cores, 3 hours

- Open `session17_gpt2_shakespeare.ipynb`

# Subword tokenization

---



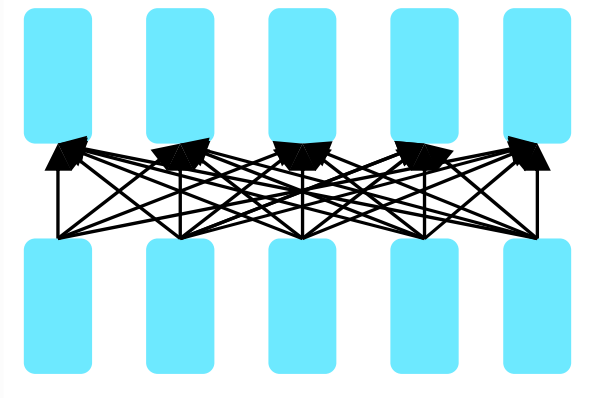
# Subword tokenization

- LLMs generally use **subword tokenization**
- E.g. byte pair encoding (BPE)
- Merges frequently seen sequences of characters together into tokens
- Repeat:
  - Choose the two symbols that are most frequently adjacent in the training corpus (say 'A', 'B')
  - Add a new merged symbol 'AB' to the vocabulary
  - Replace every adjacent 'A' 'B' in the corpus with 'AB'.
  - Until  $k$  merges have been done.
- Allows them to generalize to unseen words, handle misspellings, novel words

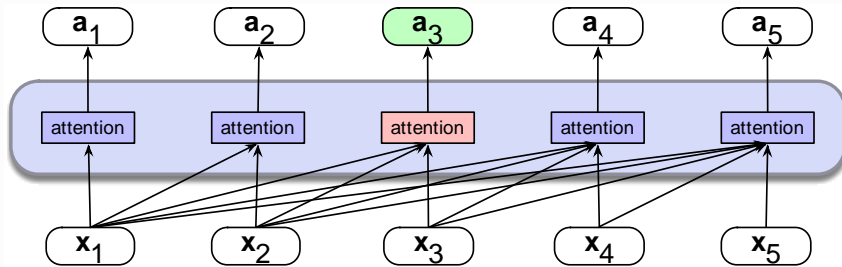
- Transformer encoder: BERT family
-

# Encoders

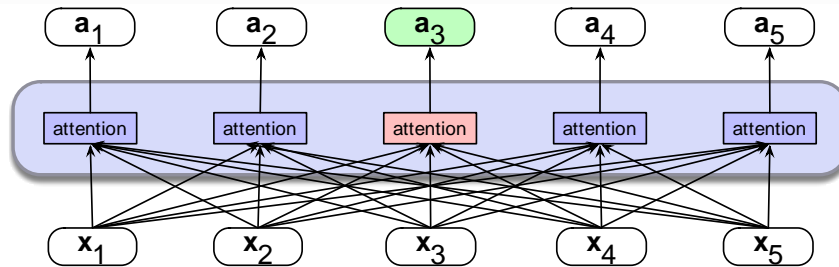
- So far, we've looked at (causal, left-to-right) language model pretraining
- But what about tasks where we want to peek at future tokens?
- Encoders can access bidirectional context
- Map sequences of input embeddings to sequences of output embeddings that have been contextualized using information from the entire sequence
- No “masking” of future words in self-attention



# Bidirectional Self-Attention



a) A causal self-attention layer



b) A bidirectional self-attention layer

# Pretraining encoders: masked language modeling

- BERT (Devlin et al. 2019) is pretrained with 2 objectives
  - Masked language modeling
  - Next sentence prediction (not as important, covered in class)

# The Cloze Task

- The **cloze** task comes from psycholinguistics (the branch of linguistics and cognitive science that uses experimental methods to study how language works in human brains).
- It is a fill-in-the-blank task:

**He drove the yellow \_\_\_\_\_ into the front of our house.**

- Subjects are presented with these frames and asked to fill in the missing words
- This allows experimenters to assess what a speaker understands about grammar, semantics, etc.
- According to the original BERT paper, this task provided the inspiration for BERT's masked language modeling (MLM) training task.
- But compare various kinds of denoising algorithms.

# MLM training in BERT

15% of the tokens are randomly chosen to be part of the masking .

Example: "Lunch was **delicious**", if delicious was randomly chosen:

Three possibilities:

1. 80%: Token is replaced with special token [MASK]

Lunch was **delicious** -> Lunch was **[MASK]**

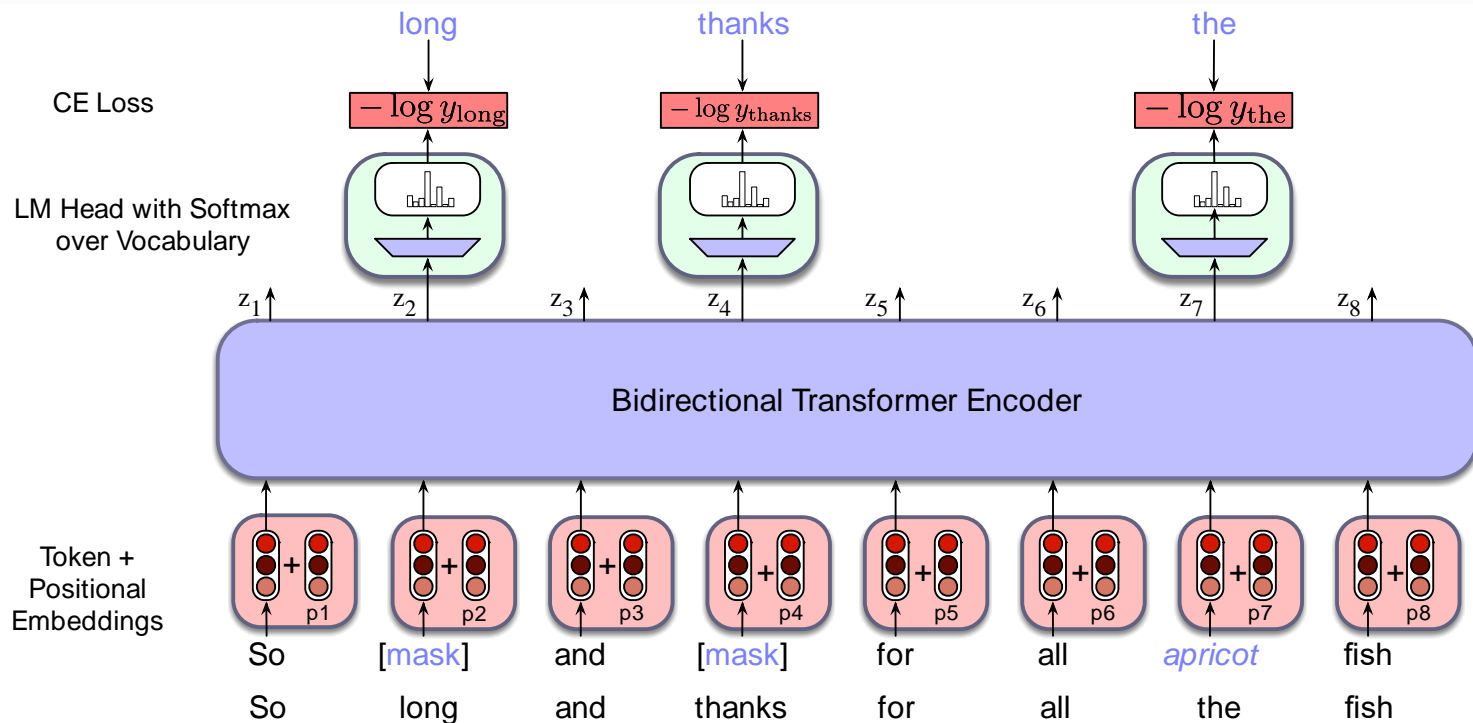
2. 10%: Token is replaced with a random token (sampled from unigram prob)

Lunch was **delicious** -> Lunch was **gasp**

3. 10%: Token is unchanged

Lunch was **delicious** -> Lunch was **delicious**

# In detail





# BERT: Bidirectional Encoder Representations from Transformers

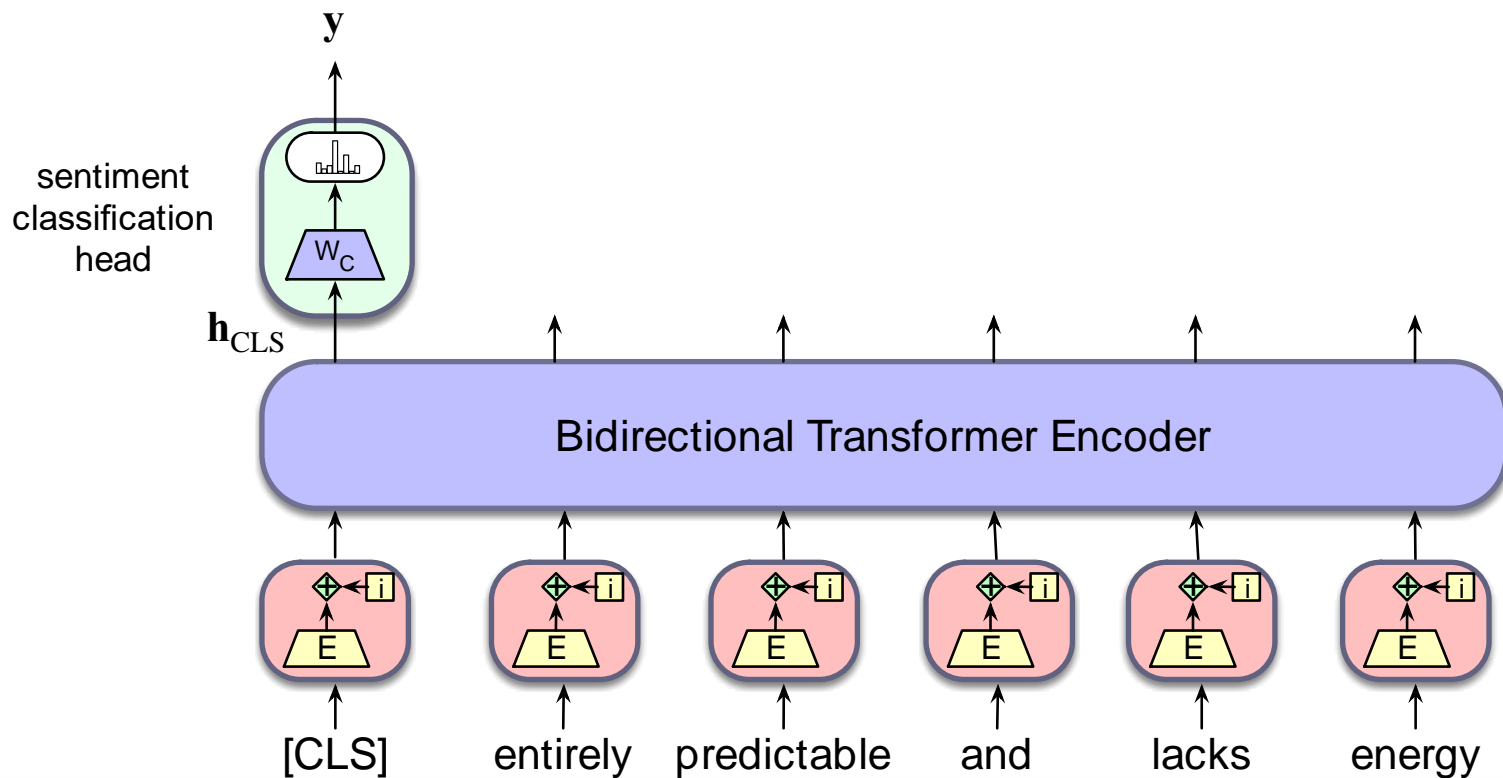
## Details about BERT

- Two models were released:
  - BERT-base: 12 layers, 768-dim hidden states, 12 attention heads, 110 million params.
  - BERT-large: 24 layers, 1024-dim hidden states, 16 attention heads, 340 million params.
- Trained on:
  - BooksCorpus (800 million words)
  - English Wikipedia (2,500 million words)
- Pretraining is expensive and impractical on a single GPU.
  - BERT was pretrained with 64 TPU chips for a total of 4 days.  
(TPUs are special tensor operation acceleration hardware)
- Finetuning is practical and common on a single GPU
  - “Pretrain once, finetune many times.”

# Finetuning BERT for classification and sequence labeling

---

# Finetuning for classification



# Fine-tuning for sequence labeling (new task!)

- Assign a label from a small fixed set of labels to each token in the sequence.
  - Named entity recognition
  - Part of speech tagging
    - Assign a part of speech (like NOUN, VERB, or ADJECTIVE) to every word in a sentence
- Labels depend not just on the word being classified, but labels of surrounding words
  - E.g. “States” is more likely to be part of a named entity if it follows the word “United”

# Named Entity Recognition

- A **named entity** is anything that can be referred to with a proper name: a person, a location, an organization
- **Named entity recognition (NER)**: find spans of text that constitute proper names and tag the type of the entity

Type	Tag	Sample Categories	Example sentences
People	PER	people, characters	<b>Turing</b> is a giant of computer science.
Organization	ORG	companies, sports teams	The <b>IPCC</b> warned about the cyclone.
Location	LOC	regions, mountains, seas	<b>Mt. Sanitas</b> is in <b>Sunshine Canyon</b> .
Geo-Political Entity	GPE	countries, states	<b>Palo Alto</b> is raising the fees for parking.

# Named Entity Recognition

Citing high fuel prices, [ORG **United Airlines**] said [TIME **Friday**] it has increased fares by [MONEY **\$6**] per round trip on flights to some cities also served by lower-cost carriers. [ORG **American Airlines**], a unit of [ORG **AMR Corp.**], immediately matched the move, spokesman [PER **Tim Wagner**] said. [ORG **United**], a unit of [ORG **UAL Corp.**], said the increase took effect [TIME **Thursday**] and applies to most routes where it competes against discount carriers, such as [LOC **Chicago**] to [LOC **Dallas**] and [LOC **Denver**] to [LOC **San Francisco**].

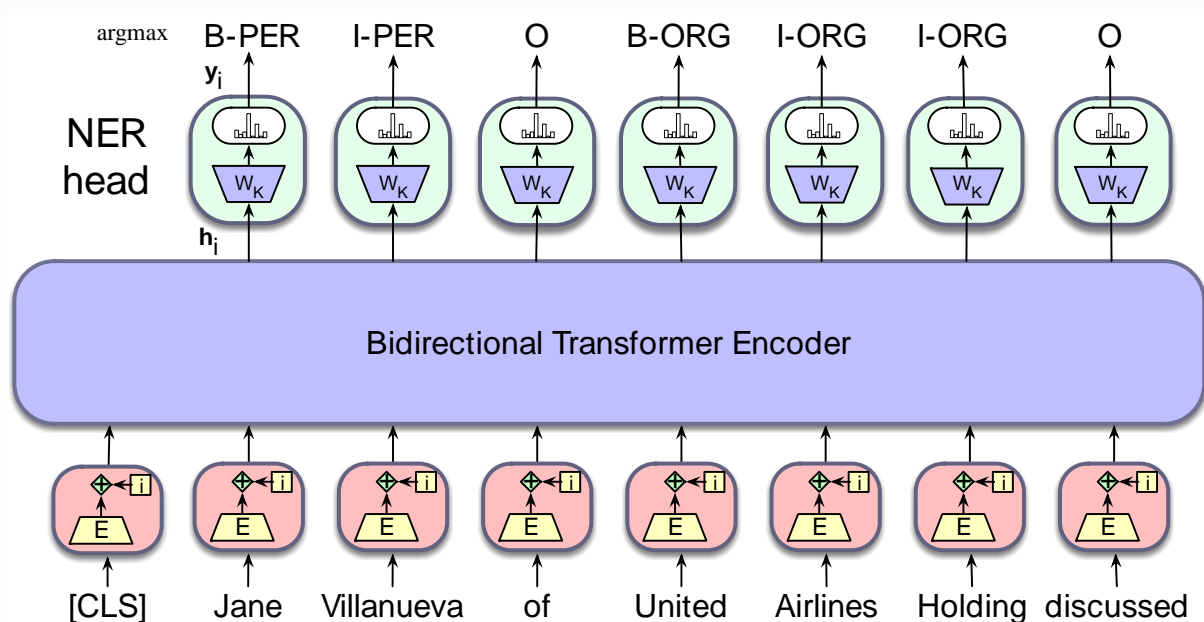
# BIO tagging [Ramshaw and Marcus 1995]

- A method that lets us turn a segmentation task (finding boundaries of entities) into a classification task

[PER Jane Villanueva] of [ORG United Airlines Holding] discussed the [LOC Chicago ] route.

Words	BIO Label
Jane	B-PER
Villanueva	I-PER
of	O
United	B-ORG
Airlines	I-ORG
Holding	I-ORG
discussed	O
the	O
Chicago	B-LOC
route	O
.	O

# Sequence labeling



$$\mathbf{y}_i = \text{softmax}(\mathbf{h}_i^L \mathbf{W}_K)$$

$$\mathbf{t}_i = \text{argmax}_k(\mathbf{y}_i)$$



# Conclusion

- LLMs use subword tokenization like BPE to learn to recognize parts of words (subword tokens). This enables them to handle words they haven't seen before
- BERT is an encoder transformer model that produces an output embedding for every input token
- BERT is pretrained on the task of masked language modeling, learning to predict masked words in the middle of sentences
- BERT is often finetuned for:
  - Classification
  - Sequence labeling, which are tasks like named entity recognition where a label is predicted for every word

# Coding activity: finetune BERT for text classification

---

# Notebook for this class: finetune BERT for politeness classification

- [Click on this nbgitpuller link](#) or find the link on the course website
- **Important difference from normal:** Open a 'Teach – 1 gpu, 3 hours' server

**Server Options**

**Select a job profile:**

Teach - 6 cores, 3 hours

Teach - 6 cores, 3 hours

Teach - 1 gpu, 3 hours

A100 - 1 gpu, 3 hours

SMP - 4 cores, 3 hours

- Open `session18_bert_politeness.ipynb`