01. BERT

Why was BERT way ahead of its time?

.

.

.

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

Source: Ramsri Goutham

CS 1671 / CS 2071 / ISSP 2071
Human Language Technologies

Session 20: BERT

Michael Miller Yoder

March 30, 2026

School of Computing and Information

Assessments: quiz

- Quiz during class **this Wed Apr 1** covering:
  - Session 17 Transformers part 1: J+M 8-8.2
  - Session 18 Transformers part 2: J+M 8.4-8.7, 8.10
  - Session 19 Post-training LLMs: J+M 9-9.3, 9.5
  - Session 20 BERT (today): J+M 10-10.2, 10.4-10.4.1, 10.6
- Final regular quiz
- There will be an extra credit quiz on a date TBD

Assessments: exam

- There will be an exam in class on **Wed Apr 8**
- Covers everything in the class up to that point
- In-class, paper exam
- Closed-book except for 1 letter page of double-sided notes
- Conceptual
- No programming
- Class next Mon Apr 6 will be a review session for the exam

Assessments: homework

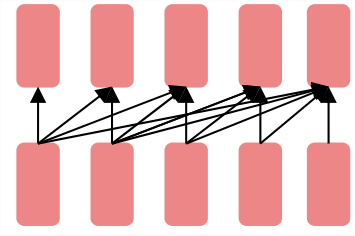
- Homework 3 has been released and is **due Apr 16**
- Run Jupyter notebooks from templates on the CRCD
- Part 1: LLM prompting
- Part 2: Instruction tuning of an LLM
  - CRCD GPUs

Lecture overview: BERT

- Coding activity from last time: finetune GPT-2 on Shakespeare
- BERT and masked language modeling
- Encoder-decoder models: T5
- Finetuning BERT for classification
- Coding activity: finetuning BERT for text classification

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

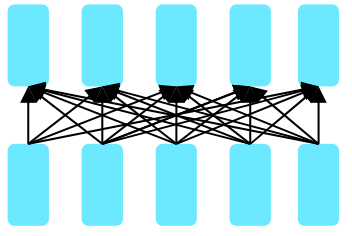
Three architectures for large language models



Decoders

GPT, Claude,

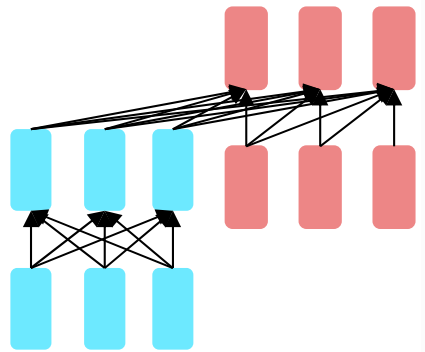
Llama, Mixtral



Encoders

BERT family,

RoBERTa



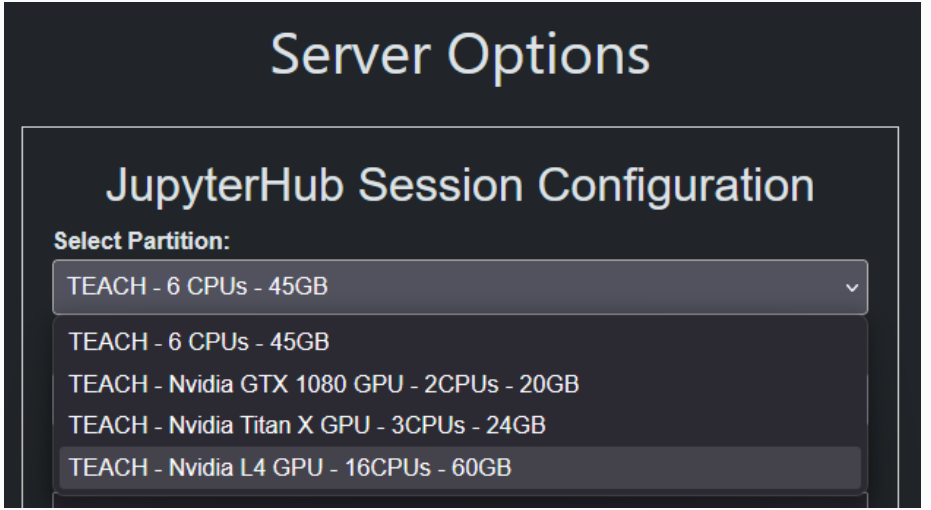
Encoder-decoders

Flan-T5, Whisper

Slide adapted from Jurafsky and Martin

Notebook: finetune GPT-2 on Shakespeare

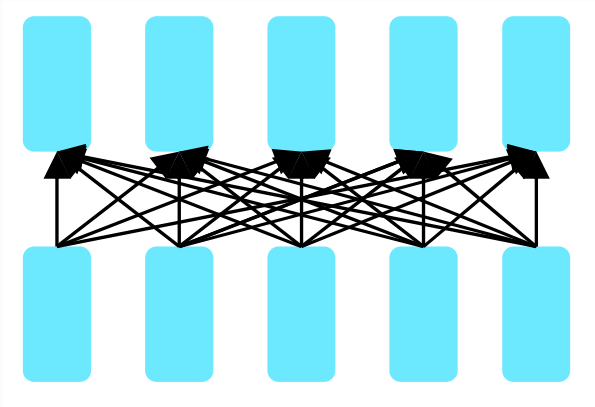
1. Click this nbgitpuller link (also available on course website)
2. **Important difference from normal:** Start a server with '**TEACH – Nvidia L4 GPU – 16 CPUs – 60GB**' server
3. Load custom environment at **/ix1/cs1671-2026s/class_env**
4. Open **session19_gpt2_shakespeare.ipynb**



Transformer encoder: BERT family

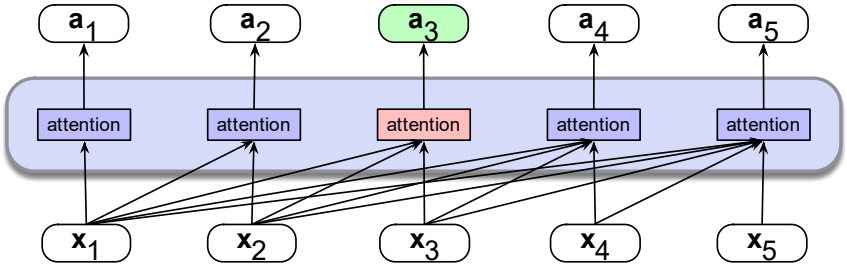
Encoders

- 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

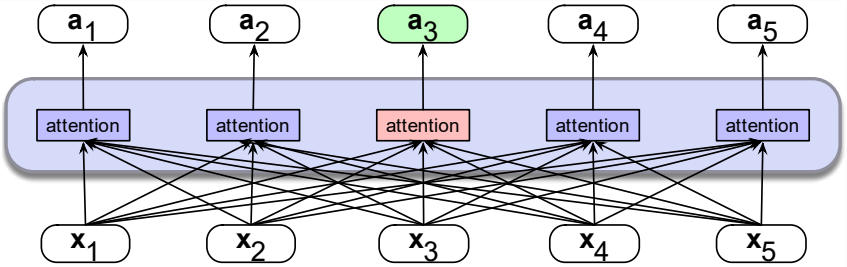


Slide adapted from John Hewitt, Jurafsky and Martin

Bidirectional Self-Attention



a) A causal self-attention layer



b) A bidirectional self-attention layer

Slide adapted from Jurafsky and Martin

Pretraining encoders: masked language modeling

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

Slide credit: David Mortensen

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)

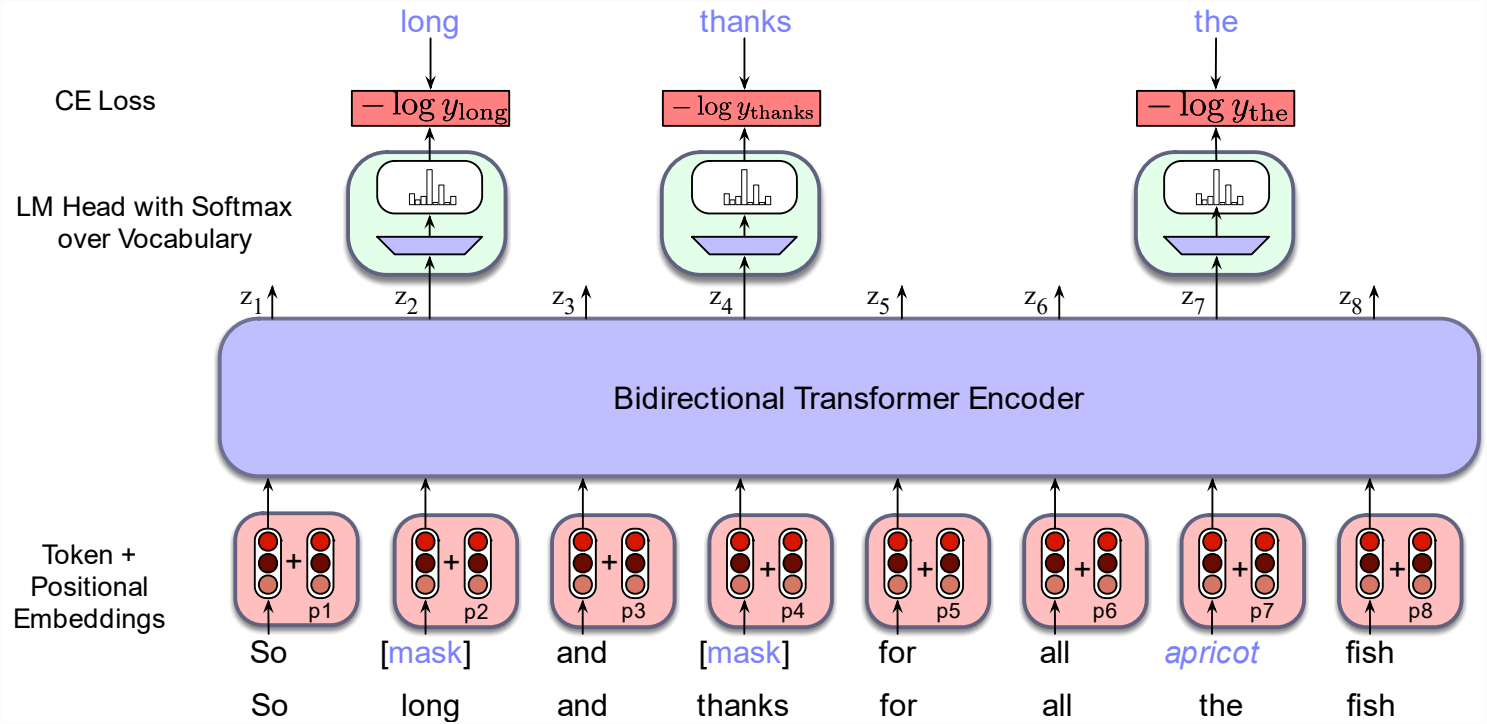
   Lunch was **delicious** -> Lunch was **gasp**

3. 10%: Token is unchanged

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

Slide adapted from Jurafsky and Martin

In detail



Slide adapted from Jurafsky and Martin

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.”

Slide adapted from John Hewitt

Extensions of BERT

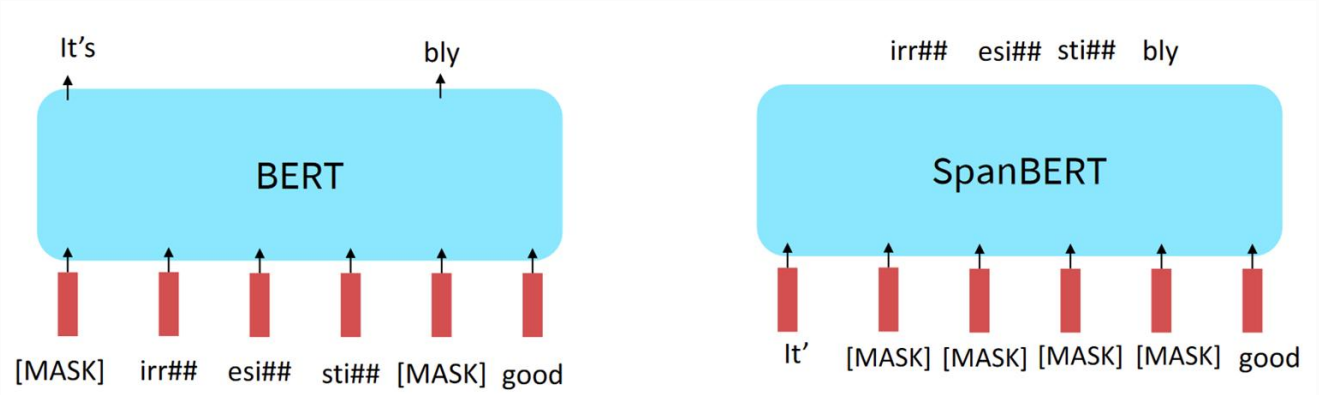
You'll see a lot of BERT variants like RoBERTa, SpanBERT, etc

Contemporary versions of BERT:

- ModernBERT
- XLM-R
- MiniLM

Some generally accepted improvements to the BERT pretraining formula:

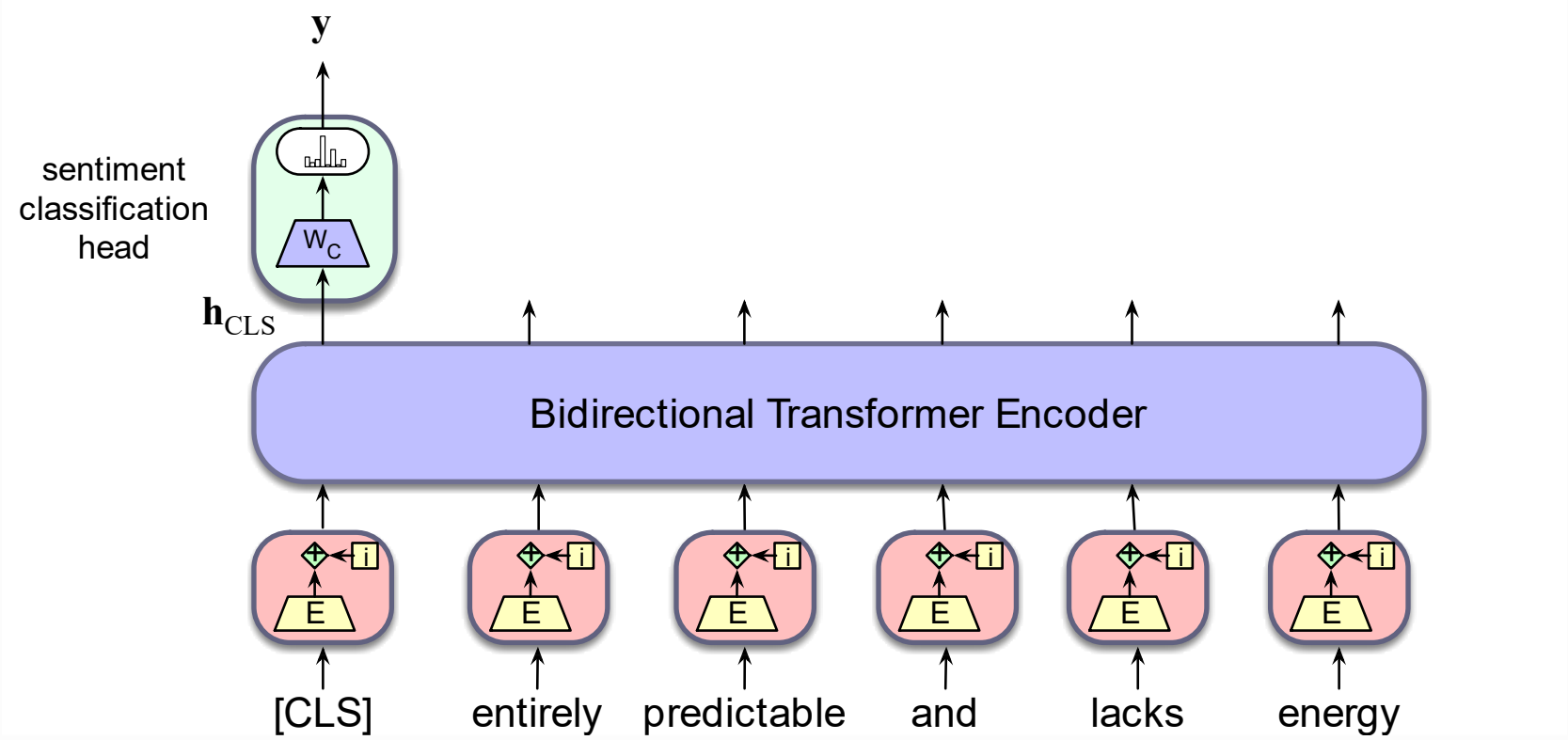
- RoBERTa [Liu et al. 2019]: mainly just train BERT for longer and remove next sentence prediction!
- SpanBERT [Joshi et al. 2020]: masking contiguous spans of words makes a harder, more useful pretraining task



Slide adapted from John Hewitt

Finetuning BERT for classification

Finetuning for classification

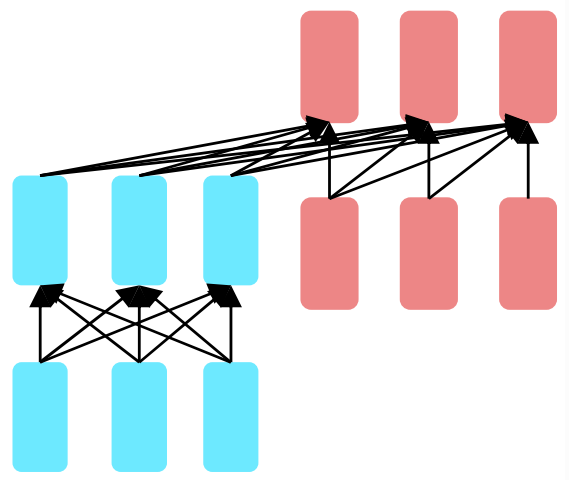


Slide adapted from Jurafsky and Martin

Encoder-decoders (T5)

Encoder-Decoders

- Trained to map from one sequence to another
- Popular for:
  - machine translation (map from one language to another)
  - speech recognition (map from acoustics to words)
- See T5 model (Raffel et al 2020)
- Decoder transformer attends to encoding



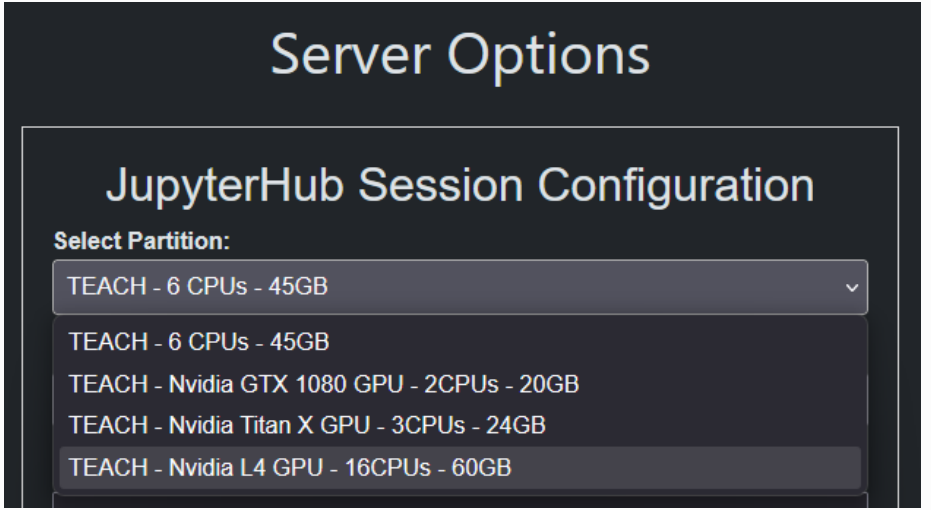
Slide adapted from Jurafsky and Martin

Conclusion

- 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 with a neural classification layer on the embedding output for the special [CLS] token (taken to represent the whole sentence)
- Encoder-decoder models like T5 map one sequence to another sequence

Notebook: finetune BERT for politeness classification

1. Click this nbgitpuller link (also available on course website)
2. **Keep using a GPU:** Start a server with '**TEACH – Nvidia L4 GPU – 16 CPUs – 60GB**' server
3. Load custom environment at **/ix1/cs1671-2026s/class_env**
4. Open **session20_bert_politeness.ipynb**



Discussion: LLMs

Now that we've gone over how different types of transformer-based LLMs are pretrained and post-trained,

1. Do you think these models understand human language? Do you think these models can reason in complicated scenarios?
2. What prior technologies have had a similar widespread social impact in information access? What lessons can we learn from how those technologies were incorporated into social life?
3. When should we be careful when using LLMs? In other words, what kind of tasks are LLMs good at, and what kind of tasks are LLMs bad at?