

01. BERT

Why was BERT way ahead of its time?

-
-
-

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



CS 2731

Introduction to Natural Language Processing

Session 20: BERT

Michael Miller Yoder

November 3, 2025

Course logistics: homework

- [Homework 3](#) has been released and is **due this Fri Nov 7**
 - Run Jupyter notebooks from templates on the CRCD
 - Part 1: LLM prompting
 - No need to run locally! Can run from the CRCD now. Michael will update the instructions
 - Part 2: Instruction tuning of an LLM
 - CRCD GPUs

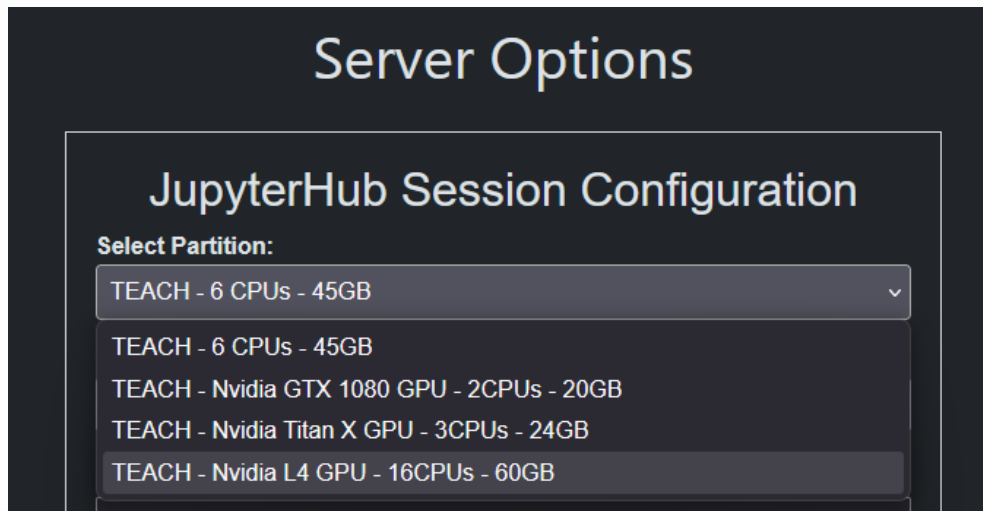
Course logistics: project

- Project progress report **due Nov 13**
 - Instructions will be released
- SCI open-source LLMs
 - There are no rate limits. Just keep in mind it is a shared resource
 - Exact models available (feel free to use any)
 - gemma3 (gemma 3:27b)
 - llama3.1 (llama3.1:70b)
 - deepseek-r1 (deepseek-r1:70b)

Prep coding activites

Notebook: 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 – Nvidia L4 GPU – 16 CPUs – 60GB' server
- Open session20_bert_politeness.ipynb
- Import packages

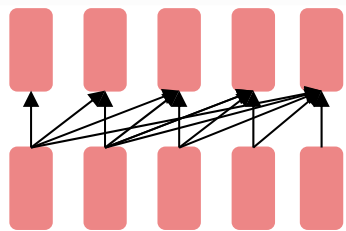


Lecture overview: BERT

- BERT and masked language modeling
- Finetuning BERT for classification and sequence labeling
- Coding activity: finetuning BERT for text classification

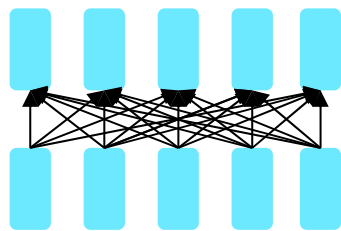
Review: Describe encoder, decoder, and encoder-decoder architectures

Three architectures for large language models



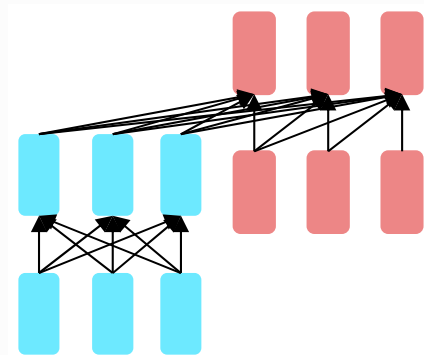
Decoders

GPT, Claude,
Llama, Mixtral



Encoders

BERT family,
RoBERTa



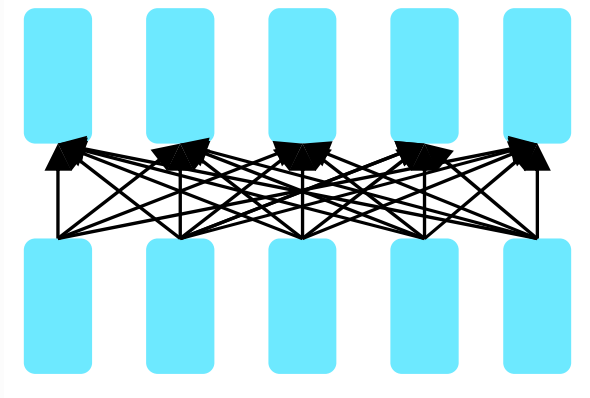
Encoder-decoders

Flan-T5, Whisper

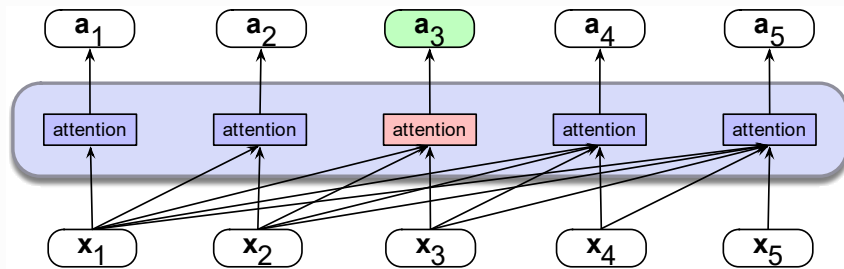
- Transformer encoder: BERT family
-

Encoders

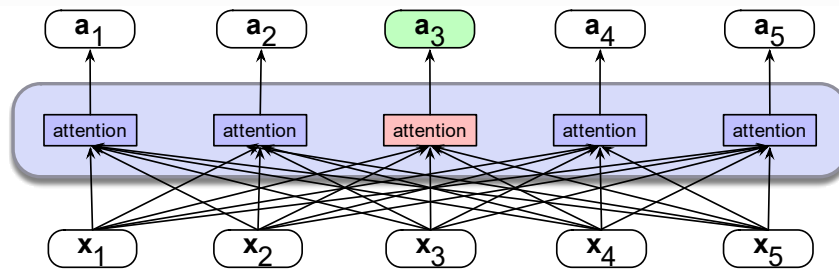
- So far, we've looked at (causal, left-to-right) language model pretraining
- But what about tasks where we want to peek at future tokens?
- Encoders can access bidirectional context
- Map sequences of input embeddings to sequences of output embeddings that have been contextualized using information 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]**

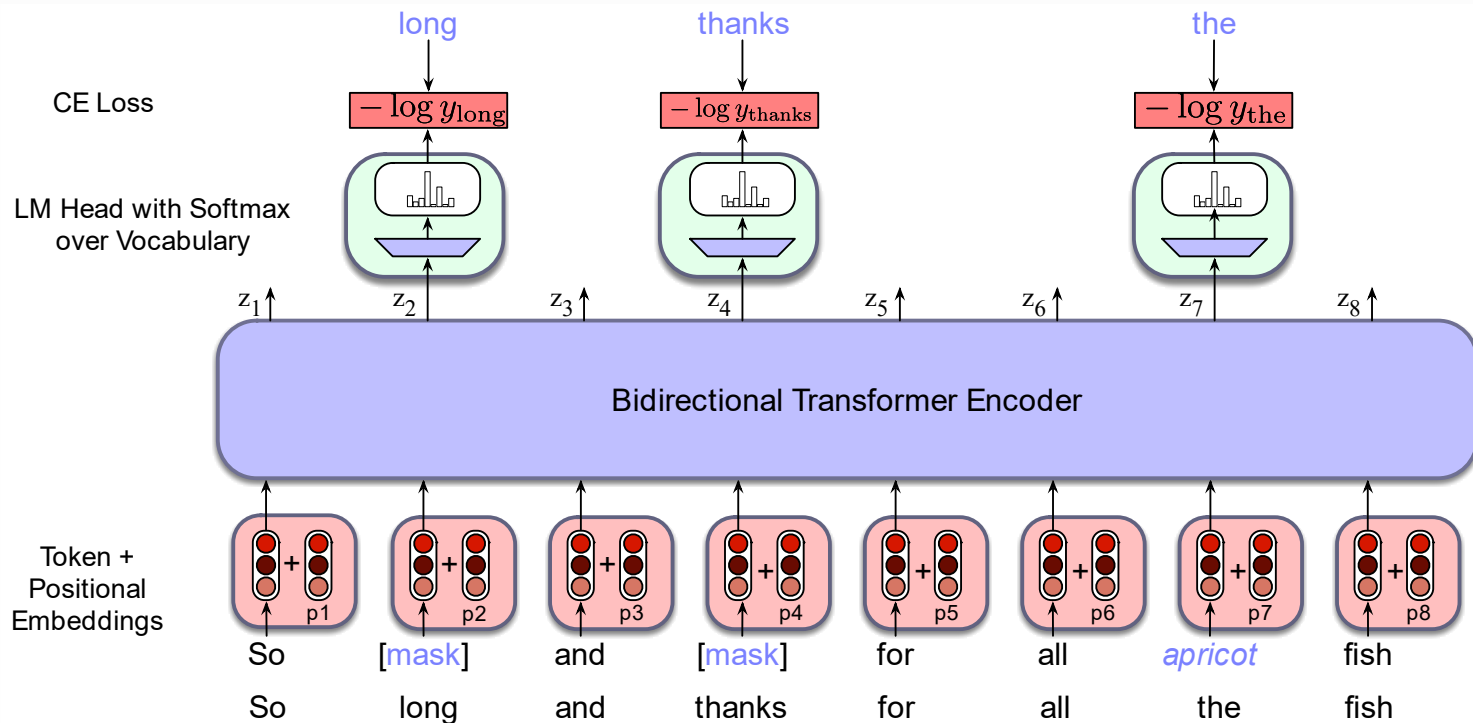
2. 10%: Token is replaced with a random token (sampled from unigram prob)

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.”

Extensions of BERT

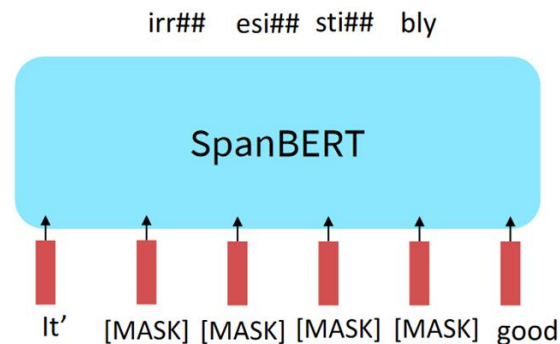
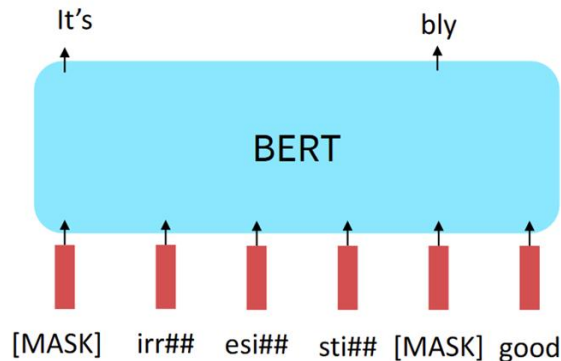
You'll see a lot of BERT variants like RoBERTa, SpanBERT, etc

Contemporary versions of BERT:

- ModernBERT
- MiniLM

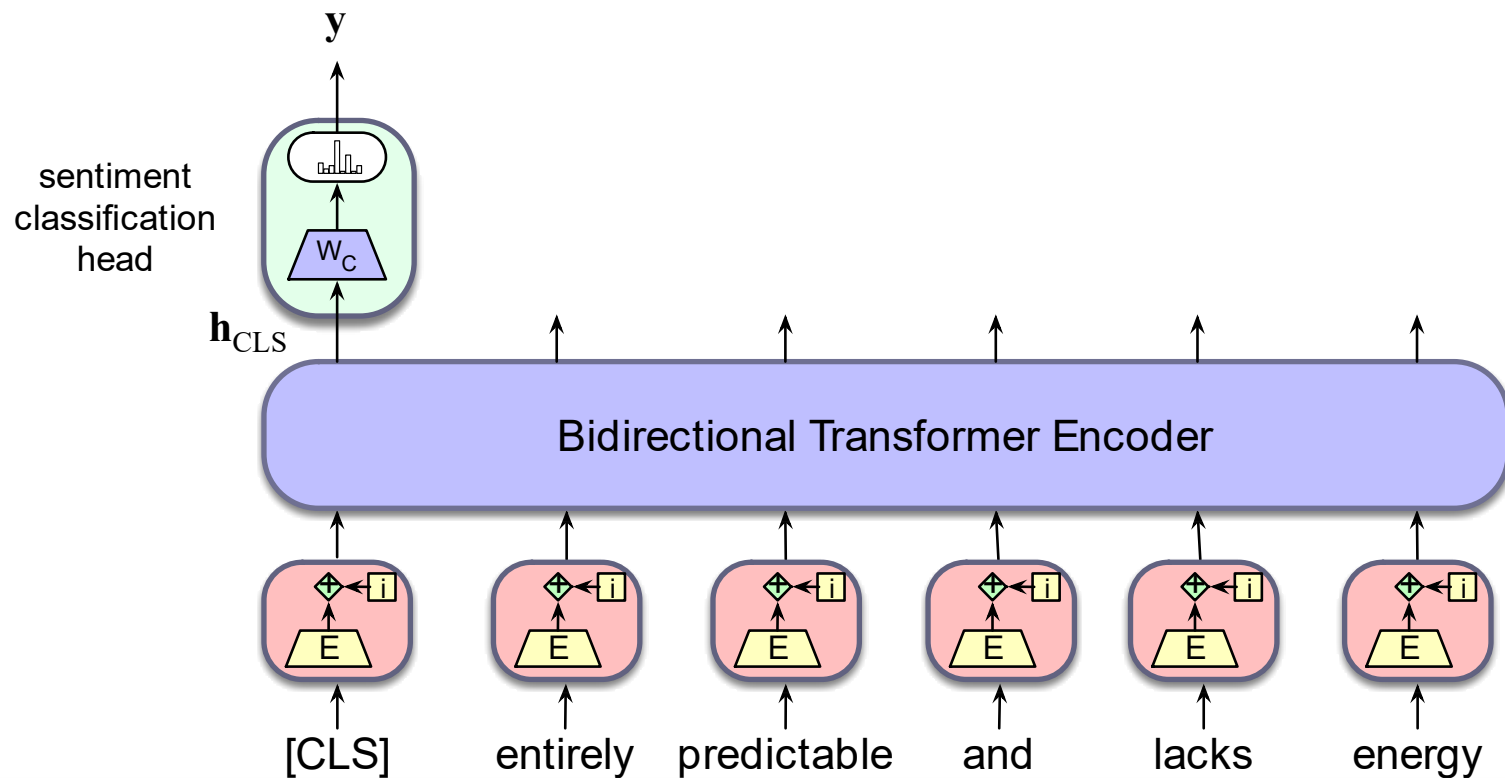
Some generally accepted improvements to the BERT pretraining formula:

- 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



Finetuning BERT for classification and sequence labeling

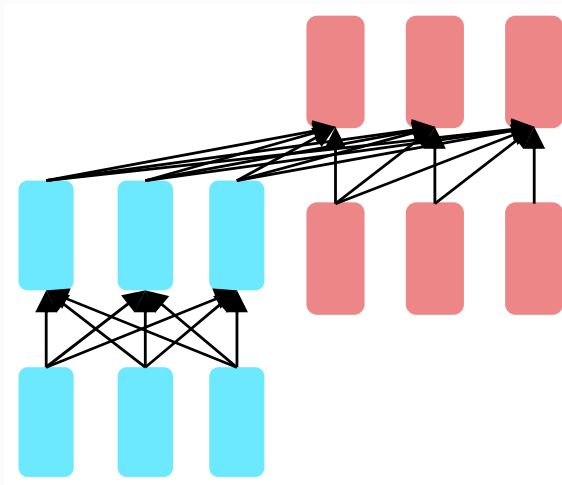
Finetuning for classification



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



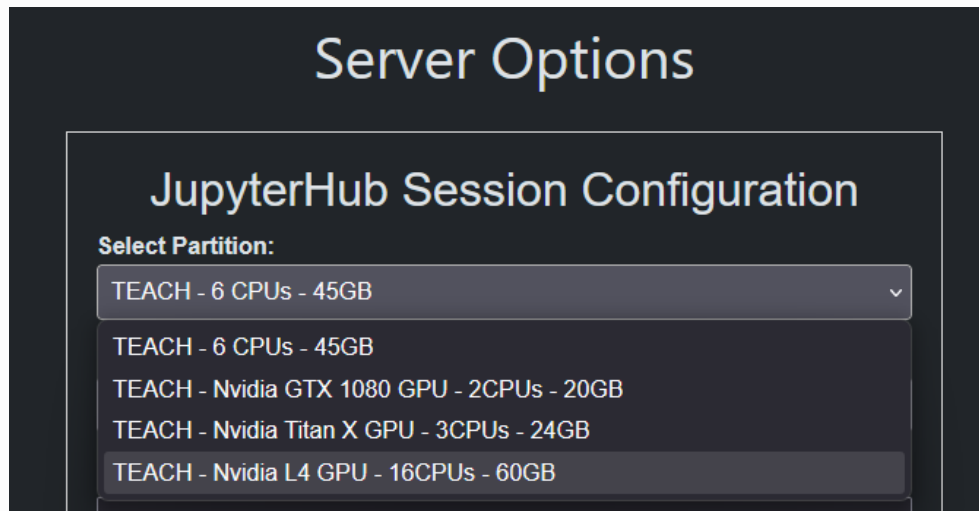
Conclusion

- 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)

Coding activities

Notebook: 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 – Nvidia L4 GPU – 16 CPUs – 60GB' server



- Open `session20_bert_politeness.ipynb`

Notebook: prompt SCI open-source LLMs

- You'll need to connect to the Pitt VPN to make API calls to SCI LLMs, even if you're on campus. Here are [instructions for doing that through the GlobalProtect app](#)
- Open `session19_prompting.ipynb`

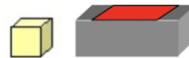
Discussion

LLMs as “cultural technologies” [Yiu et al. 2023]

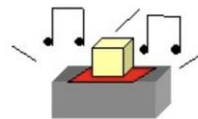
- People often debate whether LLMs are intelligent agents
- LLMs can be framed instead as “cultural technologies”: tech that enables transmission of cultural knowledge among people
 - Like earlier technologies of writing, print, libraries, internet search
 - “How you learn what grandma knows”
- Imitation vs innovation
 - Imitation: transmitting knowledge/skills from one agent to another (no notion of “truth”)
 - Innovation: “truth-seeking epistemic processes” that children do
- Experiments
 - Design new tools (use a hanger to cut a cake)
 - “Blicket detector” to detect novel causal structure



See this? It's a blivet machine.
Blivets make it go.



Let's put this one
on the machine.



Oooh, it's a
blivet!

Discussion

- Do you find the authors' analogy of pretrained LLMs as “cultural technologies” useful in conceptualizing them? How so?
- The authors argue that thinking about the “intelligence” of pretrained LLMs is the wrong question. Do you buy this framing? Can you think of any disadvantages or advantages of the “intelligence” framing vs. “cultural technologies” framing?
- What do you think about the experiments presented in the paper that test LLMs' capabilities to do tool innovation and discover novel causal relationships? Are they useful in measuring the abilities of LLMs, and what particular abilities?
- The article casts LLMs as very good imitators, but not innovators. If this is the case, does that challenge your perception of what abilities can simply be learned through imitation, as LLMs can do? Or do you think LLMs can be true innovators?