CS 2731 Introduction to Natural Language Processing

Session 16: BERT/LLMs lab and discussion day

Michael Miller Yoder

October 23, 2023

University of Pittsburgh | School of Computing and Information

Course logistics

- Homework 3 is released. Is due **Thu 11-02 at midnight**
- Let me know by tomorrow if anyone would **not** like their project proposal slides posted on the course website
- Wednesday's class will be a **project work day**
  - You will work with your project groups
  - Please incorporate feedback from the project proposal
  - Michael will be walking around assisting groups
  - Bring your laptop

Overview: BERT/LLMs discussion and lab day

- Sneak lecture (sorry): Contextual word embeddings
- LLMs as cultural technologies discussion post recap
- BERT for classification
- LLM activity: politeness classification with BERT **or** fine-tune GPT-2 to generate Shakespeare-like text

Contextual word embeddings

The meaning of words is contextual

Static word embeddings (word2vec, GloVe, etc):

"You shall know a word by the company it keeps" [Firth 1957]

"the complete meaning of a word is always contextual, and no study of meaning apart from a complete context can be taken seriously" [Firth 1935]

Let's use LLMs like BERT to get contextual word and sentence embeddings!

Slide adapted from John Hewitt, Yoav Goldberg

Static Word Embeddings Ignore Homography and Polysemy

Suppose you retrieve the embeddings for the following two sentences from a set of static word embeddings like GloVe or Word2Vec (trained via skip-gram or CBOW):

1. Lifting Dell laptops causes lower back pain.
2. He went back to his office to sulk.
3. She will back her truck into a fire hydrant.

The meanings of these three words are rather different but their embeddings would be the same. This is problematic.

Slide credit: David Mortensen

Contextual embeddings give words representations based on context

If you fed the same sentences...

1. You caused me lower back pain.
2. He went back to his office to sulk.

...to a contextual model like BERT or ELMo [Peters et al. 2018], the two words would have different embeddings (reflecting their differing meanings).

ELMo was a model that provided contextual embeddings based on bidirectional LSTMs, not transformers like BERT

Slide adapted from David Mortensen

How Do You Get Embeddings from BERT?

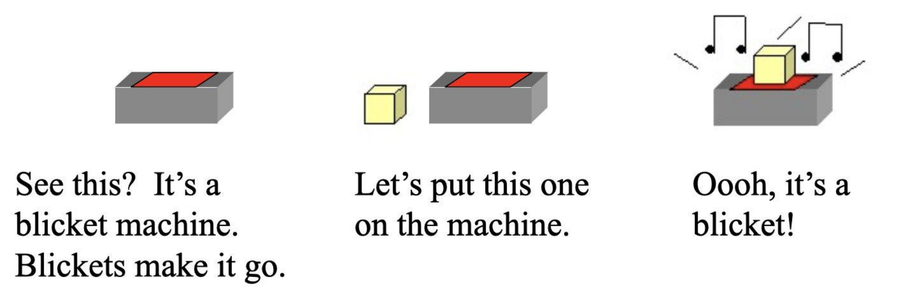
- BERT$_{BASE}$ has 12 layers
- The output of each base is an embedding
- Choose one of these, or some combination

To concatenate or sum?

LLMs as “cultural technologies”

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

- People usually 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
    - Has 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



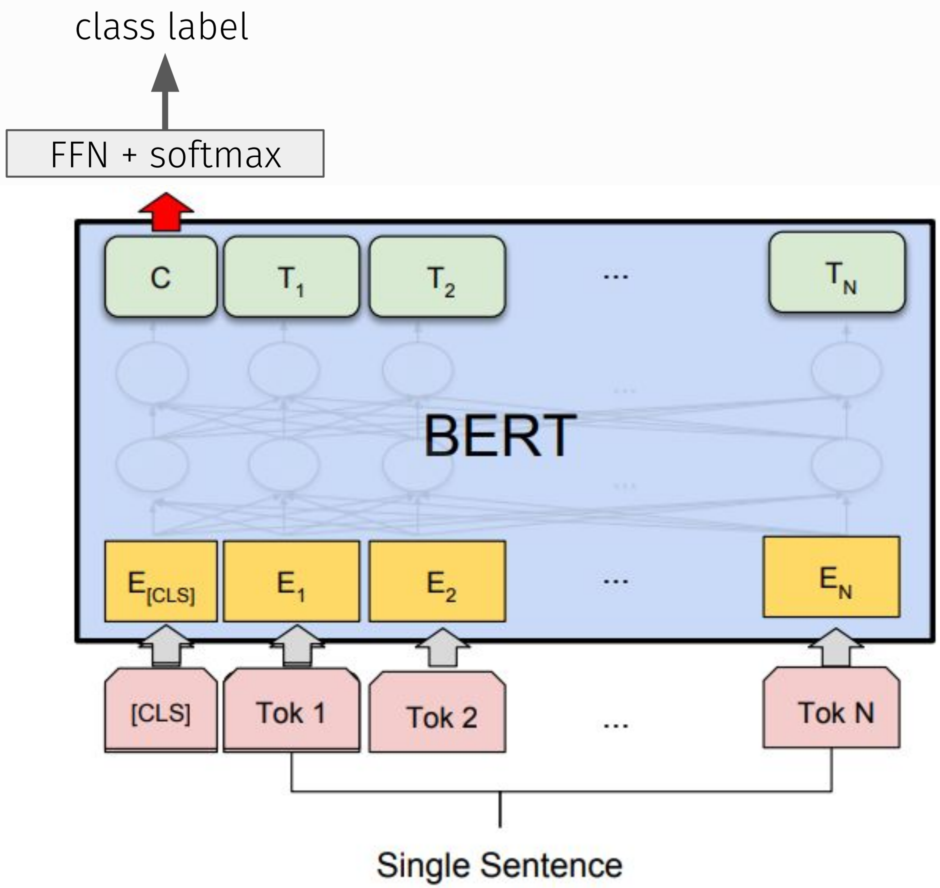
LLMs as cultural technologies [Yiu et al. 2023]

- Experiments are biased against ChatGPT since it has little concept of physical world. It can innovate and apply language to new settings (Tom)
  - Imitation can provide incredible abilities (Max, Marcelo)
  - “You need intelligence even for copying” (Bhiman)
- Why innovate if you don’t have desires and needs like humans? (Yixiao)
- Yes, they are cultural technologies to pass on information, but also are designed to mimic intelligence. “Can intelligence be communicated culturally?” (RJ)
  - Novelty comes from the world, not agents (Ben)
  - Role in affecting us: how we interact, how we share information (Norah)
- Current shortcomings of LLMs
  - They don’t automatically re-train for data drift (Bhiman)
  - They don’t check their own facts (Ben)
  - Learn rules from fewer examples, with the example of bias (Lokesh)
  - How do we improve AI models without so much data/capacity? (Gina)
- LLMs aren’t using language to accomplish a task (like making a good love letter to elicit feelings) but just to match training set (Birju)

BERT for classification

BERT for text classification

- The special [CLS] token is prepended to sentences for both training and testing BERT
- The output vector from the [CLS] token can be used as input to a FNN classifier
- This is automatically implemented in many packages (Keras, Hugging Face Trainer, PyTorch)



Lab activity

LLM activity options

1. Fine-tune BERT for text classification (politeness classification)
   a. More open-ended: you choose what package to use
2. Fine-tune GPT-2 for text generation (Shakespeare)
   a. More structured: there is a Colab notebook to start with

At the end, groups can volunteer to do code walk-throughs for the whole class

BERT for classification

- Fine-tune BERT/variant of BERT for politeness classification
- Choose a framework to use
  a. ktrain
  b. Hugging Face Trainer
  c. PyTorch (if you're familiar with it)
- Steps
  a. Load politeness data from Homework 2
  b. Split into train/dev/test with a ratio of 80/10/10
  c. Define model, set any parameters
  d. Train model
     - Can train until dev set performance goes down
  e. Evaluate accuracy on your test set
     - Tell Michael your accuracy and he will write it on the board

GPT-2 for generation

- Fine-tune GPT-2 for text generation based on Shakespeare
- **Copy** the following Colab notebook: https://tinyurl.com/3jd3f254
- Fill in the notebook and run it (with a GPU, not default CPU)
- Tell Michael some good generated examples