CS 2731 / ISSP 2230
Introduction to Natural Language Processing

Session 15: BERT/LLMs lab and discussion day

Michael Miller Yoder

February 28, 2024

Course logistics

- Project proposal presentations **Mon during class**
  - Aim for **~5 min** presentations
  - There will be Q+A after each presentation
  - Add your slides here: https://docs.google.com/presentation/d/1Xu2ebscCVlKYe_A1orOZOQEiSbZTJyjAvgVpPffUeE4/edit?usp=sharing
  - Instructions are on the project website and in the slide deck
- Project proposal feedback is coming soon (by the end of the week)

Course logistics

- Class next Wed Mar 6 will be **project work time**
  - You will work with your project groups
  - Please incorporate feedback on the project proposal
  - Michael will be walking around assisting groups
  - Bring your laptop
- Homework 3 is **due Thu Mar 7**

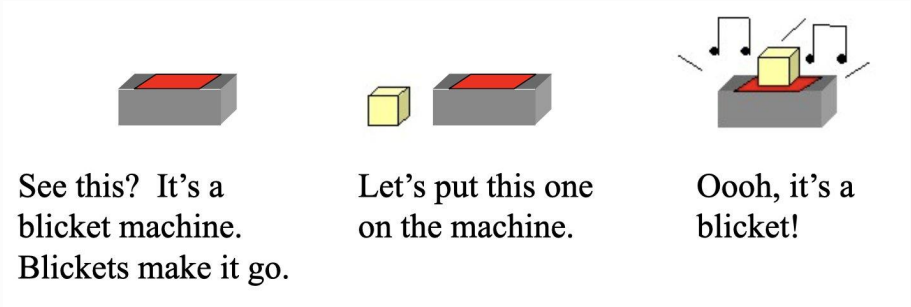
Overview: BERT/LLMs discussion and lab day

- 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

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



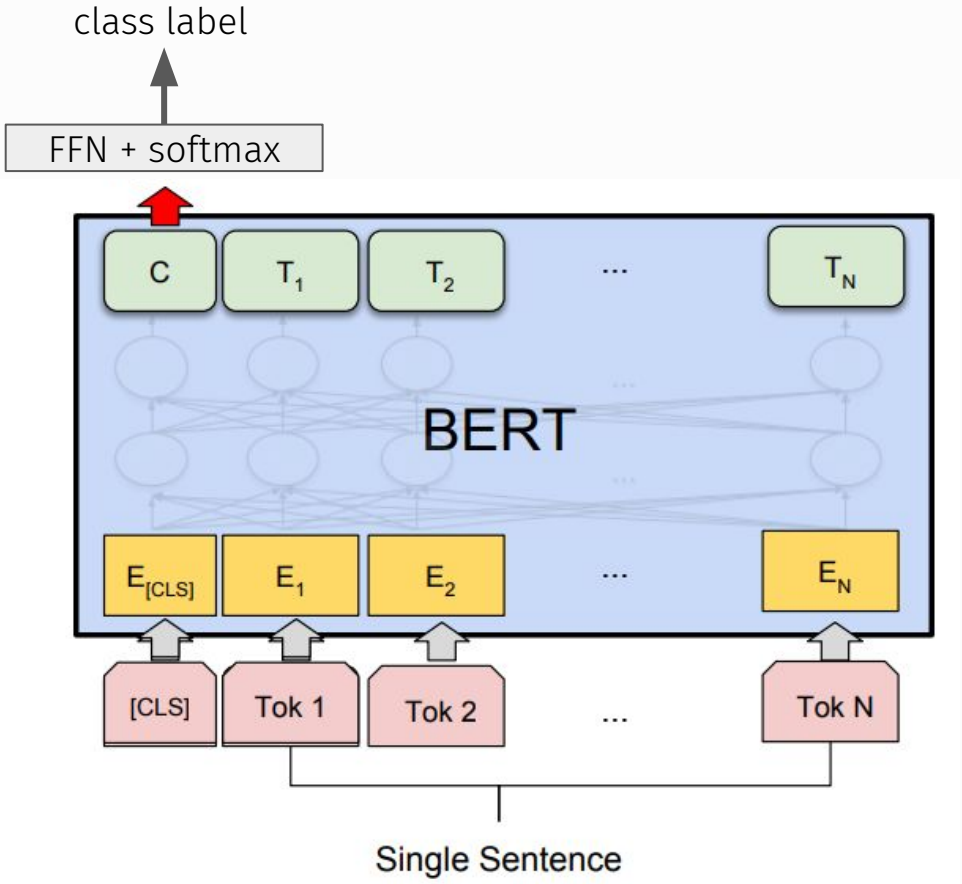
LLMs as cultural technologies [Yiu et al. 2023]

- Innovation and imitation
  - How to evaluate “innovation”, even for humans? (East)
  - LLMs are trained to imitate. Innovation would be low-likelihood events (Werner)
  - Innovation/imitation gray area (Brian)
  - Experiments don’t match academic innovation, which often combines multiple existing methods, frameworks and datasets into something new (Werner)
  - Innovation requires lots of background knowledge (Ken)
  - Does combining previously implemented ideas count as innovation? (Shayan)
  - LLMs can only say something novel when prompted with a human’s unique view (Jayden)
  - Intelligence is actually from humans who interact with them, train them, etc (Bo-Chen)
  - No “aha!” moment of novel insight is possible so not true innovators (Ken)
  - LLMs aren’t great at generating new research directions (Purva)
- Experiments rely on human, physical experience of the world (Sean)
  - And can try to train LLMs to understand the 3D physics of a simulated world (Bo-Chen)
  - Unique lived human experience is a key ingredient to how humans are different and can innovate (Jayden)
- Limitations of LLMs
  - Imitation can ignore truth, important in high-stakes settings (Ayush)
  - LLMs are getting better and may pass these tests in the future (Arushi)
  - LLMs need to be able to experiment and test causal relationships to be innovators (Xiaoyan)
  - LLMs could drive our own innovation though (Owen)

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