

Lecture 11: BERT and GPT

Generative Pre-trained Transformer (GPT)

- Improving Language Understanding by Generative Pre-Training (2018)

Alec Radford

Karthik Narasimhan

Tim Salimans

Ilya Sutskever

Bidirectional Encoder Representations from Transformers (BERT)

- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018)

Jacob Devlin

Ming-Wei Chang

Kenton Lee

Kristina Toutanova

BERT and GPT



UNIVERSITY OF
WATERLOO

- The GPT is built using transformer decoder blocks.
- BERT is built using transformer encoder blocks.

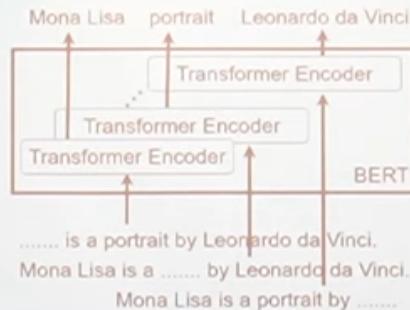
BERT: Bidirectional Encoder Representations from Transformers

Stack of encoders from the Transformer. We train it in an unsupervised manner.

We mask some words in our input sentence, pass it to the Encoder and ask it to predict the masked words. It checks the conditional probability of a token given all the words before it and ahead it. Hence, the name bi-directional.

Masked language model

- Masks words in the input and asks the model to predict the missing word.



An additional task done while training, given two sentences A and B, it checks the likeliness of sentence B following sentence A. However, studies show that this second task doesn't lead to significant performance improvement.

BERT

- BERT is designed to pretrain bidirectional representations from unlabeled text.
- Jointly conditioning on both left and right context.
- The pre-trained BERT model can be finetuned with just one additional output layer.
- It creates state-of-the-art models for a wide range of tasks, such as question answering and language inference.

[CLS] Token in BERT

Apart from a representation for every word, the entire sentence can be represented using a single CLS token. This token can be trained by adding more layers and fine-tuning the model according to our application. In fact, during the fine-tuning, CLS token is the one that is impacted the most.

[CLS] Token in BERT

- The [CLS] token is prepended to the input text and travels through the Transformer layers alongside other tokens.
- All tokens, including [CLS], gather contextual information from the entire sequence due to the self-attention mechanism.
- For sentence-level tasks, the final hidden state of the [CLS] token is used as the sentence representation.
- During fine-tuning on a specific task, the model learns to imbue the [CLS] token with a meaningful representation of the entire sentence, optimized for that task.
- Example Usage: In classification tasks, the [CLS] token representation is fed into a classifier to determine the sentence's class.

CLS is a free token, it does not represent any particular word.

Variations of BERT



BERT is basically a trained Transformer Encoder stack

1. Transformer:

1. Encoder Layers: 6
2. FFNN Hidden Layer Units: 512
3. Attention Heads: 8

2. BERT Base:

1. Encoder Layers: 12
2. FFNN Hidden Layer Units: 768
3. Attention Heads: 12
4. Total Parameters: 110 million

3. BERT Large:

1. Encoder Layers: 24
2. FFNN Hidden Layer Units: 1024
3. Attention Heads: 16
4. Total Parameters: 340 million

BERT

- **RoBERTa:**

- Optimizes BERT's training process by using more data, larger batch sizes, and longer training times, resulting in improved performance on NLP tasks.

- **TinyBERT:**

- A smaller and faster version of BERT designed for resource-constrained environments, retaining competitive performance with significantly fewer parameters.

BERT



UNIVERSITY OF
WATERLOO

1. Multilingual BERT:

1. Trained on 104 different languages, capable of "zero-shot" adaptation to a new language domain.

2. Domain Specific BERT Variants:

1. BioBERT: Retrained on a biomedical corpus.
2. SciBERT: Trained on over one million published articles.
3. BERTweet: A RoBERTa model trained on 850 million tweets.
4. FinBERT: Adapted to the financial domain.

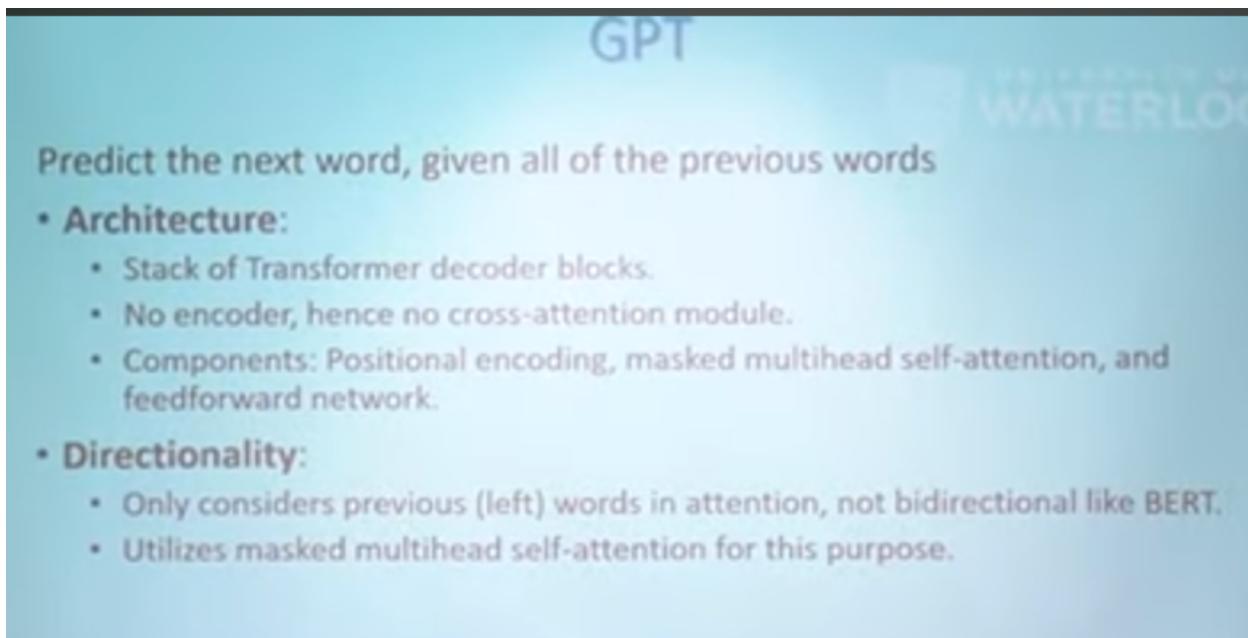
BioBERT: BERT trained on proteins

GPT: Generative Pre-Trained Transformer

Stack of Decoders

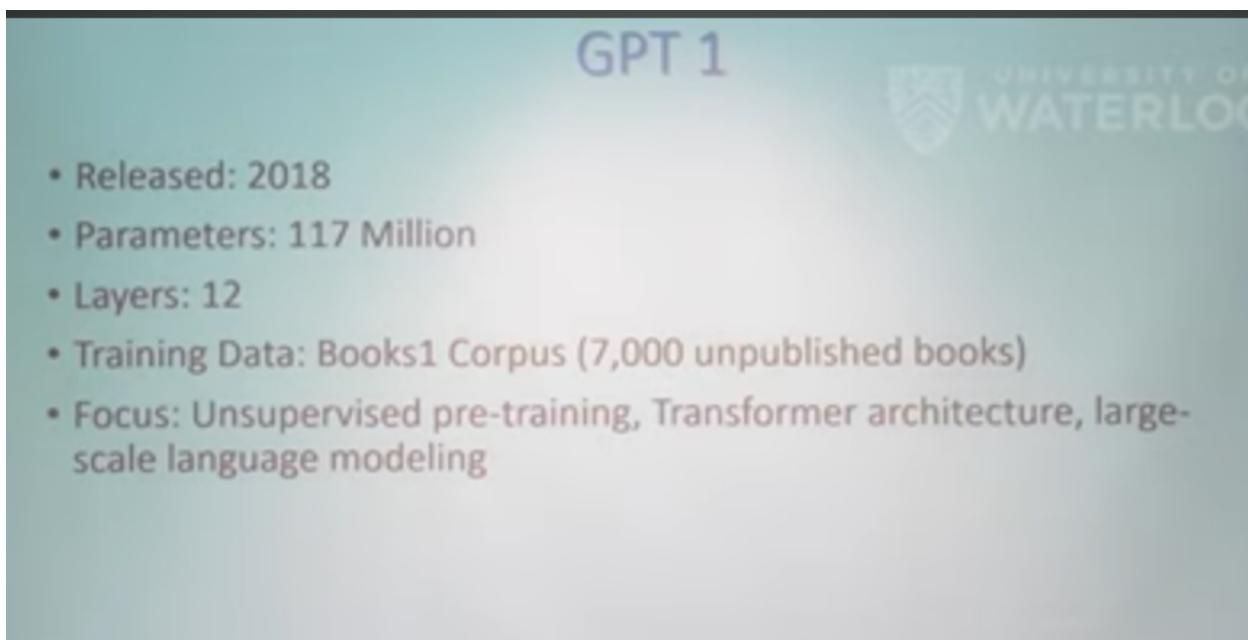
Since there is no Encoder to GPT, the Cross Attention layer has nothing to attend to(because it gets the key and value from output of encoder). So in GPT, the decoders only have the Masked Attention layer and Feed Forward NN.

In GPT, the process of training is to predict the next word in the sequence.



The slide features a light blue background with the text "GPT" at the top left and the University of Waterloo logo at the top right. Below the title, the text "Predict the next word, given all of the previous words" is displayed. The main content is organized into two sections: "Architecture" and "Directionality", each with a bulleted list of points.

- **Architecture:**
 - Stack of Transformer decoder blocks.
 - No encoder, hence no cross-attention module.
 - Components: Positional encoding, masked multihead self-attention, and feedforward network.
- **Directionality:**
 - Only considers previous (left) words in attention, not bidirectional like BERT.
 - Utilizes masked multihead self-attention for this purpose.



The slide features a light green background with the text "GPT 1" at the top left and the University of Waterloo logo at the top right. The main content is a bulleted list of key characteristics of the GPT 1 model.

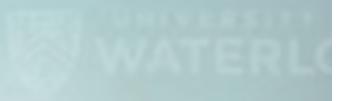
- Released: 2018
- Parameters: 117 Million
- Layers: 12
- Training Data: Books1 Corpus (7,000 unpublished books)
- Focus: Unsupervised pre-training, Transformer architecture, large-scale language modeling

GPT 2



- Released: 2019
- Parameters: ~1.5 Billion
- Layers: 48
- Training Data: 40GB (English)
- Focus: Transformer architecture, self-attention mechanism

GPT 3



- Released: 2020
- Parameters: 175 Billion
- Layers: 175
- Training Data: 570GB (Multilingual)
- Focus: Few-shot learning, prompt engineering, Python support

- Release: Not Yet
- Parameters: ~100 Trillion (speculative)
- Layers: Unknown
- Training Data: Larger, more diverse (speculative)
- Focus: GPT-4 is known to be a multimodal model, capable of processing both text and image inputs to generate text outputs. Advanced few-shot learning, improved NLU and NLG, reasoning and inference

T5: Text-to-Text Transformer

T5 is roughly BERT and GPT put together. But this is only in terms of the architecture.

One interesting thing done in T5 is that, they have cast many natural language processing problems as text-to-text problems. For example, sentiment analysis involves labeling a sentence as positive or negative. So this can be cast as a text-to-text problem by training it to give out the token positive or negative.

Pretraining Objectives

- Supervised training conducted on downstream tasks provided by GLUE and SuperGLUE benchmarks, reformulated into text-to-text tasks.
- Self-supervised training employs corrupted tokens: 15% of tokens are randomly removed and replaced with sentinel tokens.

Training T5 is a combination of both: supervised and unsupervised. Training a BERT or GPT is unsupervised while training for NLP tasks like sentiment analysis, translation, summarization is supervised.

Task Adaptation

- T5 adapts to various tasks by prepending task-specific prefixes to the input, e.g., for translation: "translate English to German: ...", for summarization: "summarize: ...".
- Achieves very good results on many benchmarks including summarization, question answering, and text classification.

T5 also introduced different prompts. So if we write 'summarize:' and then the text to be summarized, it provides us

with a summary of that text.

Core Challenges with GPT models

Inherently it is not good at conversations like a chatbot and is also not good at following instructions.

The slide has a light blue background with a faint watermark of a person's face in the top right corner. The title 'Core Challenge with GPT Models:' is in large blue font at the top left. Below the title is a bulleted list of challenges:

- GPT models predict the next token based on historical data, lacking an innate ability to follow instructions.
- GPT (Generative Pre-trained Transformer) models, at their core, predict the next word or token in a sequence based on the probabilities derived from pre-training on extensive text corpora.
- They don't inherently "understand" instructions or follow commands but generate what's statistically likely to come next, given their training.

We want GPT to align to our specific instructions and also some moral rules. It should not be offensive in its response.

The Goal of Alignment:

- The aim is to align GPT's responses with specific user instructions and ethical standards, beyond just generating probable text.
- The primary objective is to bridge the gap between these statistical predictions and meaningful adherence to instructions provided by users.
- This involves ensuring that the AI's responses are not just contextually appropriate or conversationally relevant but also aligned with the specific intentions, ethical expectations, and task-oriented goals of the user.

For example, we can try to jailbreak ChatGPT. If ask about a person, we get some response. If put the same prompt again, its response might change and it might end up stating something offensive. But it all comes from the data given to the model. The offensive opinion could have come from a comment by someone about the person. So we need to align the model to what we want as a user.

Key Areas of Focus in AI Alignment

1. Learning from Human Feedback:

- Tailoring AI through human interaction.

2. Training to Follow Instructions:

- Teaching AI specific task adherence.

3. Evaluating AI Harms:

- Identifying risks in AI outputs.

4. Modifying Behavior to Mitigate Harms:

- Adjusting AI to prevent negative impacts.

Three-Phased Training Approach for Alignment

Addressing GPT Challenges - Training Strategy Overview

• To address inherent challenges with GPT models, ChatGPT's training strategy mirrors the "Instruct GPT" approach, itself an amalgamation of strategies from preceding works.

• **The Three-Phased Training Approach:**

- 1. Supervised Fine-Tuning (SFT):**
 1. "Refines a pre-trained GPT-3 model's responses for specific tasks or guidelines, enhancing its understanding and output relevance."
- 2. Training a Reward Model (RM):**
 1. "Develops a system that assesses the quality of text generated by the model, guiding it towards human-preferred responses."
- 3. Reinforcement Learning Training (RL):**

SFT: Given the prompts, you teach the model to follow certain instructions.

The Three-Phased Training Approach

- 1. Supervised Fine-Tuning (SFT):**
 - Refines a pre-trained GPT-3 model's responses for specific tasks or guidelines, enhancing its understanding and output relevance.
- 2. Training a Reward Model (RM):**
 - Develops a system that assesses the quality of text generated by the model, guiding it towards human-preferred responses.
- 3. Reinforcement Learning from Human Feedback (RLHF)**
 - Refining AI behavior through direct human feedback.

RLHF: A dataset is generated from the human feedback received and it is then used to train the model in order to refine its behaviour.