Lec: 19

**Notes on Pretraining Strategies and Transformer Architectures**

**1. Pretraining Strategies**

**1.1 Masked Language Model (MLM)**

* Used in **BERT (Bidirectional Encoder Representations from Transformers)**.
* Some tokens in the input are masked, and the model predicts those masked tokens.
* A **self-supervised** approach (does not require labeled data).
* BERT, as an **encoder model**, can look at all tokens in the input.
* Unlike autoregressive models, BERT has access to the entire input sequence.

**2. Encoder-Decoder Model in Pretraining**

* The **encoder-decoder model** consists of:
  + An **encoder** that processes the input sequence.
  + A **decoder** that generates the output sequence.
* The **decoder-only model** is autoregressive, meaning at time step **t**, it only has access to tokens up to **t-1**.
* Encoder-only models (e.g., BERT) are **not ideal** for generative tasks since they lack autoregression.
* The encoder-decoder model requires more parameters and memory but can handle generative tasks better.
* The **decoder-only model** (e.g., GPT) is more efficient for generation since it follows an autoregressive setup.

**3. Transformer Architecture Recap**

* **Input processing**:
  + Input sequence is embedded.
  + Positional encoding is added.
* **Encoder:**
  + Multiple stacked blocks (e.g., 6 or 12 blocks).
  + Each block contains:
    - **Multi-head self-attention** (unmasked).
    - **Feed-forward layer**.
    - **Add & Norm layer with residual connections**.
* **Decoder:**
  + Also consists of multiple blocks.
  + Each block contains:
    - **Masked multi-head self-attention** (prevents access to future tokens).
    - **Cross-attention layer** (attends to encoder outputs).
    - **Feed-forward layer**.
  + Decoder outputs are passed through a **softmax layer** for token generation.

**4. Pretraining of Encoder-Decoder Models**

* Objective remains **language modeling** (e.g., next word prediction, sentence classification).
* Input can be an **instruction followed by a task-specific input** (e.g., summarization, sentiment analysis).
* Encoder processes the input, and the decoder predicts the target sequence in an **autoregressive** manner.

**5. Using Unlabeled Data for Pretraining**

* **Challenge**: Collecting labeled data is difficult.
* **Solution**: Self-supervised learning (no need for human-labeled data).
* **BART (Bidirectional and Auto-Regressive Transformer)** combines:
  + **Bidirectional encoding** from BERT.
  + **Autoregressive decoding** from GPT.
* **Pretraining objective**: Inject noise into the input and train the model to recover the original input.

**6. Data Corruption Strategies in BART**

1. **Token Masking** (like BERT): Mask some tokens, and the model predicts them.
2. **Token Permutation**: Shuffle token order and reconstruct the correct sequence.
3. **Sentence Rotation**: Rotate the sequence and reconstruct the original order.
4. **Token Deletion**: Remove some tokens and train the model to predict the full sequence.
5. **Span Masking**: Mask consecutive tokens instead of individual ones.

**7. Comparison: BERT vs. BART**

| **Feature** | **BERT** | **BART** |
| --- | --- | --- |
| Model Type | Encoder-only | Encoder-Decoder |
| Pretraining | Predicts masked tokens | Predicts full sequence |
| Uses Labeled Data? | No | No |
| Parameter Efficiency | More efficient | Requires more parameters |

**8. Fine-Tuning and Applications**

* **Pretrained models** are fine-tuned for downstream tasks (e.g., summarization, translation).
* Pretrained knowledge allows models to require **less labeled data**.
* **GPT-3** showed that **fine-tuning isn't always needed**, as large-scale pretraining enables zero-shot and few-shot learning.
* BART was fine-tuned on news summarization datasets and showed high effectiveness with minimal labeled data.

**9. Introduction to T5 (Text-to-Text Transfer Transformer)**

* Proposed by **Google** around the same time as BART.
* An **encoder-decoder model** designed for handling all NLP tasks as **text-to-text problems**.
* Unlike BART, T5’s training **requires labeled data**.
* The model is **task-agnostic**, meaning every NLP task is converted into a text generation problem.

**10. Examples of T5 Tasks**

1. **Machine Translation**:
   * Input: "Translate English to German: The weather is nice today."
   * Output: "Das Wetter ist heute schön."
2. **Text Classification (CoLA - Acceptability Check)**:
   * Input: "Check acceptability: The course is jumping well."
   * Output: "Not acceptable."
3. **Sentence Similarity (STS-B)**:
   * Input: "Sentence 1: The Rhino grazed on the grass. Sentence 2: A rhino is grazing in the field."
   * Output: "3.8" (Similarity Score as a String)
4. **Summarization**:
   * Input: "Summarize: The state authorities dispatched emergency crews Tuesday..."
   * Output: "Authorities deployed emergency crews."

**11. Pretraining in T5**

* Similar **corruption techniques** as BART but uses **Sentinel Tokens** instead of just [MASK].
* **Sentinel Tokens** allow different mask positions to have unique identifiers.
* During training, T5 **only predicts the masked spans**, not the entire sequence.

**12. T5 Training Data**

* Used **C4 (Colossal Clean Crawled Corpus)**, a **cleaned version of Common Crawl**.
* Preprocessing steps:
  + Removed non-English texts.
  + Removed HTML, JavaScript, and duplicates.
  + Filtered out noisy, incomplete sentences.

**13. T5 Fine-Tuning and Benchmarks**

* Fine-tuned on multiple datasets:
  + **GLUE Benchmark**: NLP classification and entailment tasks.
  + **SuperGLUE Benchmark**: Harder NLP tasks like reasoning and comprehension.
  + **CNN/DailyMail**: Summarization dataset.
  + **SQuAD**: Question-answering dataset.
  + **WMT (Machine Translation Tasks)**: English-German, English-French, English-Romanian.

**14. T5 Performance and Results**

* **Compared to BERT**, T5 showed improvements in multiple benchmarks.
* Reported **accuracies at different checkpoints** during training.
* Showed that **text-to-text framework works for diverse NLP tasks**.
* Used large-scale **TPU compute resources** for training.

**15. Summary**

* **BERT**: Good for understanding context but lacks generative capabilities.
* **BART**: Combines **BERT’s bidirectional encoding** and **GPT’s autoregressive decoding**.
* **T5**: Treats **all NLP tasks as text-to-text**, allowing for unified handling of classification, summarization, translation, and more.
* **T5 uses Sentinel Tokens**, **pretrains on C4 Corpus**, and **requires labeled data for fine-tuning**.
* \*\*T5 demonstrates that a unified text-to-text framework can handle diverse NLP problems efficiently.

**16. Advanced Pretraining Approaches**

* **Prefix Language Model (Prefix LM)**:
  + One part of the sequence is fully visible (like BERT), and the other part is autoregressive (like GPT).
* **Comparison of Architectures**:
  + **Encoder-only (BERT):** Entire input is visible.
  + **Decoder-only (GPT):** Only past tokens are visible.
  + **Prefix LM:** Partially visible + autoregressive region.

**17. GPT Evolution**

* **GPT-1 (2018)**:
  + 12-layer Transformer decoder.
  + 117M parameters.
  + Used BookCorpus dataset.
* **GPT-2 (2019)**:
  + 1.5B parameters.
  + Used Common Crawl dataset.
  + Introduced zero-shot learning.
* **GPT-3 (2020)**:
  + 175B parameters.
  + Introduced in-context learning.
  + Demonstrated human-like text generation.
* **GPT-4 (2023, Blackbox Model)**:
  + Unknown parameter count.
  + Costlier and more powerful than GPT-3.
  + Introduced **Instruction Fine-Tuning** and **Reinforcement Learning from Human Feedback (RLHF)**.

**18. Future Directions**

* **Instruction Fine-Tuning**: Training models to understand human instructions better.
* **RLHF (Reinforcement Learning from Human Feedback)**: Aligning AI models with human expectations.
* **LLaMA (Meta's Open-Source Model)**:
  + Uses **Byte Pair Encoding** and **Rotary Positional Encoding**.
  + Trained on **Common Crawl, Wikipedia, GitHub, Books**, etc.
  + Open-source alternative to GPT models.

**19. Summary**

* **BERT**: Good for understanding context but lacks generative capabilities.
* **BART**: Combines **BERT’s bidirectional encoding** and **GPT’s autoregressive decoding**.
* **T5**: Treats **all NLP tasks as text-to-text**, allowing for unified handling of classification, summarization, translation, and more.
* **GPT Evolution**: Progressed from GPT-1 to GPT-4 with increasing model sizes and capabilities.
* **Instruction Fine-Tuning & RLHF**: Key trends in improving AI alignment with human needs.