Slide 1: Title Slide

Understanding, Training, and Evaluating Modern NLP Systems

- Introduction to transformer-based language models
- From pretraining to practical applications
- Scaling, efficiency, and responsible deployment

Slide 2: Problem Statement

• Traditional n-gram models have severe limitations
• Limited context window
• Cannot capture long-range dependencies
• Poor performance on complex language tasks
• Need for models that can:
• Learn from massive amounts of text data
• Handle diverse NLP tasks (translation, QA, summarization)
• Generate coherent, contextually appropriate text

Slide 3: Three Architectures for LLMs

• Decoders (GPT, Claude, Llama)**
• Generate text left-to-right
• Excellent for text generation
• Cannot condition on future words
• Encoders (BERT, HuBERT)**
• Bidirectional context understanding
• Strong for classification tasks
• Build rich text representations
• Encoder-Decoders (Flan-T5, Whisper)**
• Best of both worlds
• Ideal for sequence-to-sequence tasks (translation, summarization)

Slide 4: Core Training Mechanism

• The Big Idea**
• Train models to predict the next word in a sequence
• No manual labeling required—supervision comes from the text itself
• Training Process**
• Use cross-entropy loss to measure prediction error
• Teacher forcing: always provide correct context
• Optimize weights via gradient descent
• Why It Works**
• Text contains enormous amounts of knowledge and language patterns
• Learning to predict words captures linguistic structure and world knowledge

Slide 5: Practical Applications via Conditional Generation

• Sentiment Analysis**
• Prompt: "The sentiment of 'I like Jackie Chan' is:"
• Model predicts: positive/negative
• Question Answering**
• Prompt: "Q: Who wrote The Origin of Species? A:"
• Model generates: Charles Darwin
• Text Summarization**
• Prompt: [Full article] "tl;dr"
• Model generates: concise summary

Slide 6: Sampling Strategies for Generation

| Strategy | Method | Trade-off |
|

•

•

• |

| **Greedy Decoding** | Always pick most likely word | Deterministic, repetitive |

| **Top-k Sampling** | Sample from top k words | Fixed pool size |

| **Top-p (Nucleus)** | Sample from top p% probability | Adaptive, context-aware |

| **Temperature** | Reshape probability distribution | Smooth quality-diversity control |

• Key Insight**: Low temperature = more focused/factual; High temperature = more creative/diverse

Slide 7: Pretraining Data Sources

• Common Sources**
• Common Crawl: billions of web pages
• The Pile: 825 GB from 22 diverse sources
• Books, Wikipedia, academic papers, code
• Quality Filtering**
• Remove boilerplate and duplicate content
• Deduplication at URL, document, and line levels
• Toxicity and bias detection
• Ethical Concerns**
• Copyright and fair use questions
• Privacy risks (leaked personal data)
• Bias from training data sources

Slide 8: Scaling Laws

• The Power Law Relationship**
• Loss scales predictably with model size (N parameters)
• Loss scales predictably with dataset size (D tokens)
• Loss scales predictably with compute budget (C FLOPs)
• Practical Implications**
• Can predict performance improvements from early training curves
• Informs decisions about resource allocation
• Example: GPT-3 has ~175 billion parameters
• Key Formula**: $L(N) \propto (N/N_c)^{(-a_N)}$

Slide 9: Efficiency at Inference

• The Problem**
• At inference, we generate tokens one at a time
• Naively recomputing attention for all prior tokens is wasteful
• The Solution: KV Cache**
• Store key and value vectors from previous tokens in memory
• Retrieve cached values instead of recomputing
• Dramatically reduces computation per token
• Impact**: Makes real-time inference feasible for billion-parameter models

Slide 10: Parameter-Efficient Fine-Tuning

• The Challenge**
• Fine-tuning huge models requires updating billions of parameters
• Backpropagation through massive layers is computationally expensive
• LoRA Solution**
• Freeze pretrained weight matrix W
• Update low-rank decomposition: W + AB (where r << original dimension)
• Dramatically reduces trainable parameters (e.g., 0.1% of original)
• Benefit**: Adapt models to new domains with minimal computational cost

Slide 11: Evaluation Metrics

- Perplexity**
- Inverse probability of test set (normalized by length)
- Lower perplexity = better predictions
- Per-word metric for fair comparison
- Beyond Perplexity**
- Model size and memory requirements
- Energy consumption (kWh, CO2 emissions)
- Fairness: gender/racial bias, dialect representation
- Task-specific benchmarks (accuracy, BLEU, ROUGE)

Slide 12: Harms and Limitations

• Hallucination**
• Models generate plausible-sounding but false information
• No built-in fact-checking mechanism
• Copyright & Legal Issues**
• Training on copyrighted material without permission
• Fair use doctrine remains legally unclear
• Privacy Risks**
• Personal information may leak from training data
• Potential for model inversion attacks
• Bias & Toxicity**
• Amplification of biases from training data
• Generation of harmful, abusive, or discriminatory content
• Disparate performance across demographic groups

Slide 13: Summary & Key Takeaways

• What We Learned**

1. LLMs are trained via self-supervised next-word prediction

2. Transformer architecture enables efficient parallel processing

3. Scaling laws show predictable performance improvements

4. Many NLP tasks can be framed as conditional text generation

5. Efficiency techniques (KV cache, LoRA) enable practical deployment

• Open Challenges**

• Reducing hallucination and improving factuality

• Addressing copyright, privacy, and bias concerns

• Developing more efficient training methods

• Building trustworthy and interpretable systems

• The Future**: Continued research on safety, efficiency, and alignment with human values