Slide 1: Title Slide

Understanding the Foundations, Training, and Applications

- Introduction to transformer-based language models
- From pretraining to practical applications
- Key techniques for scaling and optimization
- Understanding capabilities and limitations

Slide 2: Problem Statement

- Traditional n-gram models have limited context and knowledge
- Need models that can handle complex language understanding tasks
- Must scale to billions of parameters while remaining efficient
- Balance between model performance, cost, and safety
- Key Question:** How can we train massive models to predict text and solve diverse NLP tasks?

Slide 3: Three Architectures for Language Models

- Decoders (GPT, Claude, Llama)**
- Generate text left-to-right
- Can't condition on future words
- Ideal for text generation tasks
- Encoders (BERT, HuBERT)**
- Bidirectional context
- Condition on both past and future
- Strong for representation learning
- Encoder-Decoders (Flan-T5, Whisper)**
- Combine strengths of both approaches
- Excellent for sequence-to-sequence tasks
- Used in translation and speech recognition

Slide 4: How LLMs Learn - Self-Supervised Training

• The Core Algorithm:**

1. Take massive corpus of text (Common Crawl, Wikipedia, books, web)

2. At each position, predict the next word

3. Use cross-entropy loss to minimize prediction error

4. Update weights via gradient descent

• Why It Works:**

• No manual labels needed—text provides its own supervision

• Models learn language structure, facts, and reasoning

• Enormous pretraining data teaches vast amounts of knowledge

• Teacher Forcing:** Always provide correct history when training, not model's predictions

Slide 5: Sampling Strategies for Text Generation

- Random Sampling**
- Choose words by probability distribution
- Problem: rare words in tail cause weird outputs
- Top-k Sampling**
- Keep only top k most probable words
- Renormalize and sample from remaining words
- Fixed k may not adapt to different contexts
- Top-p (Nucleus) Sampling**
- Keep top p% of probability mass instead of fixed k
- More robust across different contexts
- Most commonly used in practice
- Temperature Sampling**
- Reshape probability distribution (don't truncate)
- Low temperature ($\tau < 1$): more greedy/focused
- High temperature ($\tau > 1$): more diverse/creative

Slide 6: Scaling Laws and Efficiency

- Scaling Laws:** Performance improves with power-law relationship to:
- Model size (# parameters)
- Dataset size (# training tokens)
- Compute budget (FLOPs used)
- KV Cache:** Store key/value vectors during inference to avoid recomputation
- Parameter-Efficient Fine-Tuning (PEFT):**
- LoRA: Update low-rank decomposition instead of full weights
- Freeze most parameters, train only small adapter matrices
- Reduces memory and compute costs dramatically
- Example:** Llama 3.1 405B has 405 billion parameters—efficiency is critical!

Slide 7: Diverse Applications Through Prompting

- Sentiment Analysis**
- Prompt: "The sentiment of 'I like Jackie Chan' is: [positive/negative]"
- Compare probabilities of candidate words
- Question Answering**
- Prompt: "Q: Who wrote The Origin of Species? A:"
- Generate answer tokens iteratively
- Text Summarization**
- Prompt: "[Article text] tl;dr:"
- Model learns to generate summaries from seeing pattern in training data
- Key Insight:** Transformers' large context windows allow them to condition on entire documents, enabling sophisticated multi-step reasoning

Slide 8: Training Data and Ethical Considerations

- Common Data Sources:**
- Common Crawl (billions of web pages)
- Wikipedia, books, academic papers
- The Pile (22 diverse datasets, 825 GiB)
- Quality & Safety Filtering:**
- Remove boilerplate, duplicates, adult content
- Toxicity detection (with mixed results)
- Deduplication at URL, document, and line levels
- Remaining Challenges:**
- Copyright: Much training data is copyrighted material
- Privacy: Web data may contain personal information
- Bias: Models reflect biases in training data
- Consent: Website owners may not have agreed to data use

Slide 9: Evaluation and Harms

- Evaluation Metrics:**
- **Perplexity:** Inverse probability normalized by sequence length (lower is better)
- **Task-specific benchmarks:** Accuracy on downstream tasks
- **Fairness metrics:** Gender/racial bias, performance across dialects
- **Efficiency:** Energy usage, memory requirements
- Major Harms to Address:**
- **Hallucination:** Generating false information confidently
- **Toxicity & Abuse:** Generating harmful or offensive content
- **Misinformation:** Spreading false or misleading claims
- **Privacy Violations:** Memorizing and reproducing training data
- **Copyright Issues:** Training on copyrighted material without permission

Slide 10: Summary & Key Takeaways

- What We've Learned:**
- LLMs are trained to predict the next word using self-supervision
- Three architectures serve different purposes (decoder, encoder, encoder-decoder)
- Sampling strategies balance quality and diversity in generation
- Scaling laws guide efficient training decisions
- Many NLP tasks can be framed as conditional text generation
- Critical Considerations:**
- Massive scale requires efficiency techniques (KV cache, LoRA)
- Training data quality and ethics are paramount
- Evaluation must go beyond perplexity to fairness and safety
- Hallucination and misinformation remain open challenges
- Future Direction:** Building LLMs that are more efficient, safer, and more aligned with human values