Dr. Babasaheb Ambedkar Technological University, Lonere

Presented by: Sidhant Dnyaneshwar Dange

PRN: 24030331245009

Subject: Seminar

Title: Large Language Model : Vector Data Representation Technique

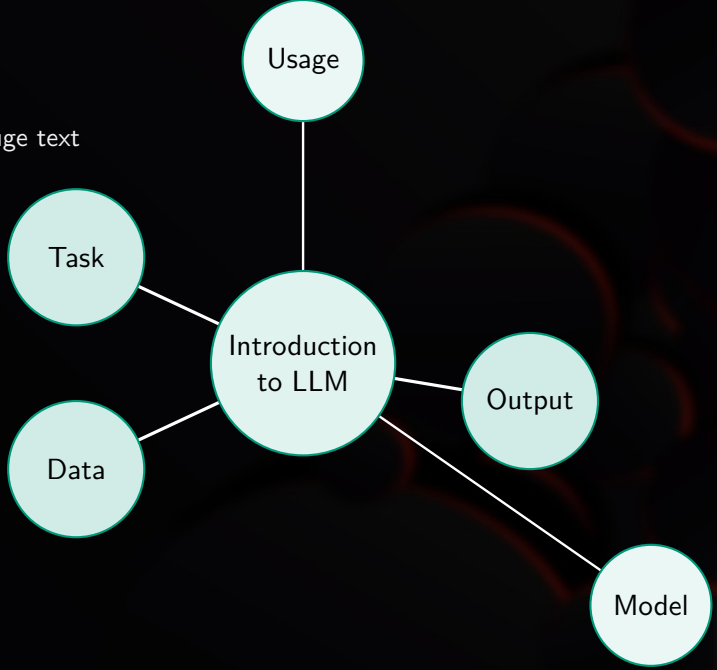
Batch: A

Branch: Computer Engineering

Guided by: Prof. Dr. Laxman.D. Netak

Introduction to LLM

1. **Training Data:** LLMs learn from huge text datasets.
2. **Human Output:** They generate responses like humans.
3. **NLP Tasks:** Used for translation, summary, Q&A.
4. **Transformer:** Works using self-attention mechanism.
5. **Applications:** Chatbots, coding tools,search engines.



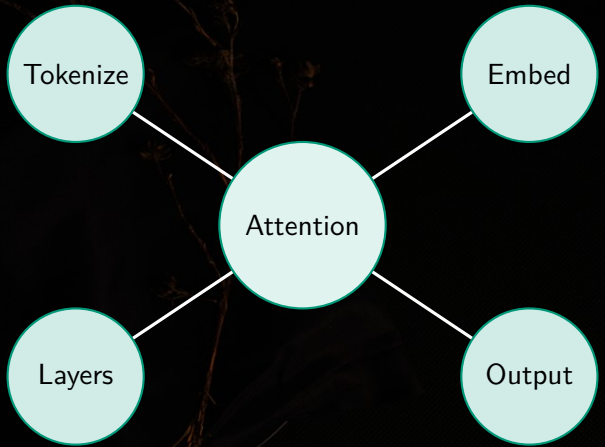
How LLMs Work

1. Tokenization: Text is broken into small units called tokens.

2. Layers: Multiple transformer layers refine context and meaning.

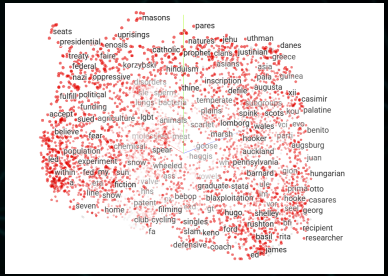
3. Embedding: Tokens are converted into vectors for processing.

4. Output: LLM predicts next tokens to generate responses.



Vector Representation in LLMs

1. **Vectors from Tokens:** LLMs convert each token into a numerical vector.
2. **Semantic Meaning:** Words with similar meaning have similar vector positions.
3. **High Dimensions:** Embeddings exist in 256–2048 dimensional mathematical space.
4. **Context Encoding:** Grammar, relationships, and patterns are stored in vector form.
5. **Input to Transformer:** These vectors enter the attention layers for processing.



Transformer Processing Steps

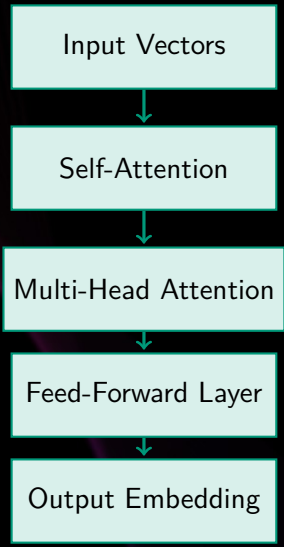
1. Self-Attention: Model checks which tokens are important to each other.

2. Multi-Head Attention: Multiple attention heads capture different relationships.

3. Feed-Forward Layer: Each token is processed through a neural network.

4. Residual Connections: Original information is preserved for stability.

5. Layer Normalization: Keeps values stable and helps better learning.



Attention Mechanism in LLMs

1. Query, Key, Value: Each token is converted into query, key, and value vectors for comparison.

2. Relevance Scoring: Attention computes how strongly one token should focus on others.

3. Softmax Weighting: Scores are normalized into probabilities for stable attention.

4. Weighted Output: Each token receives information from all others through weighted values.

5. Multi-Head Attention: Multiple heads capture different relationships simultaneously.

Types of LLM Architectures

1. Encoder-Only Models: Used for understanding and classification tasks (e.g., BERT).

2. Decoder-Only Models: Designed for text generation (e.g., GPT family).

3. Encoder–Decoder Models: Effective for translation and summarization (e.g., T5).

4. Multimodal Models: Handle text + images or audio (e.g., GPT-4V).

5. Mixture-of-Experts (MoE): Routes input to specialized experts for efficiency.

Training Process Overview

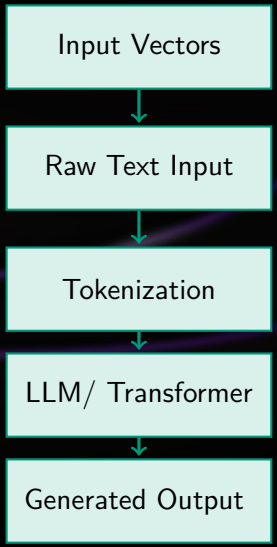
1. Raw Text Input: Model receives plain text from the user.

2. Tokenization: Text is broken into smaller units like words or subwords.

3. Vector Embeddings: Each token is converted into a numerical vector for processing.

4. Transformer Processing: Model applies attention, feed-forward networks, and multiple layers.

5. Generated Output: The final processed vector is decoded back into human-readable text.



Multi-Head Attention in LLMs

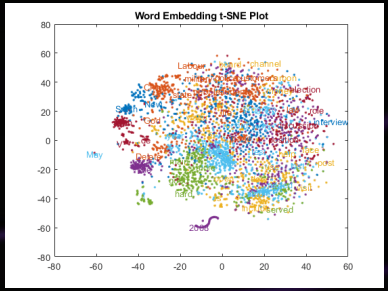
1. Parallel Attention Heads: Multiple attention heads run simultaneously to capture different types of relationships.

2. Diverse Feature Extraction: Each head learns unique patterns such as syntax, semantics, or long-range dependencies.

3. Independent Projections: Input vectors are transformed separately for each head using learned weight matrices.

4. Scaled Dot-Product Attention: Each head computes attention using Query, Key, and Value projections.

5. Concatenation & Final Output: Outputs from all heads are combined and linearly transformed to form the final representation.



Residual Connections and Layer Normalization

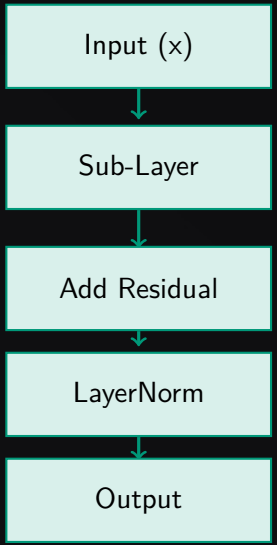
1. **Residual Path:** The original input (x) is added back to the sub-layer output.

2. **Prevents Gradient Loss:** Helps gradients flow smoothly through deep layers.

3. **Layer Normalization:** Normalizes activations to maintain stable distributions.

4. **Faster Training:** Normalization helps the model converge efficiently.

5. **Supports Deep Models:** Residual + LayerNorm enables very deep Transformers.



Self-Attention Details

1. **Token Interaction:** Each token attends to all other tokens in the sequence.

2. **Attention Scores:** Calculated using dot-products between query and key vectors.

3. **Softmax Weights:** Scores are normalized to highlight important relationships.

4. **Weighted Values:** Information is aggregated using attention weights.

5. **Parallel Processing:** Attention allows tokens to be processed simultaneously.

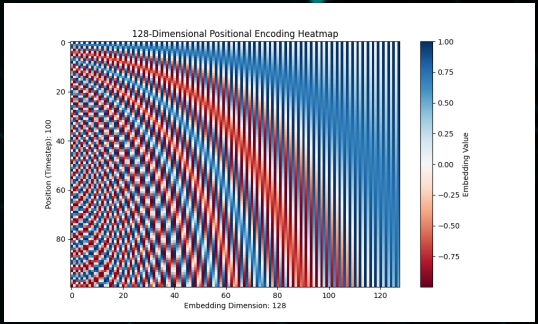
Positional Encoding in LLMs

1. Adds Word Order Information: Transformers do not understand sequence order, so positional encoding provides it.

2. Sinusoidal Functions: Positions are encoded using sine and cosine waves of different frequencies.

3. Unique Position Patterns: Each position in a sequence gets a unique vector based on its index.

4. Smooth Generalization: The sinusoidal pattern allows the model to generalize to longer sentences.



5. Combined with Embeddings: Token embeddings + positional encodings give full contextual meaning.

Applications of LLMs

1. **Text Generation:** Chatbots, story writing, email drafting, coding assistants.
2. **Translation:** High-quality translation across multiple languages.
3. **Summarization:** Convert large documents into concise summaries.
4. **Question Answering:** Accurate answers from large knowledge bases.
5. **Sentiment Analysis:** Identify emotions and opinions from text.

Advantages of LLMs

1. **Human-Like Responses:** Generate natural and coherent language outputs.
2. **Versatility:** Can perform many tasks without retraining.
3. **Context Understanding:** Maintains long-range dependencies and meaning.
4. **Scalability:** Larger models produce more accurate results.
5. **Automation Power:** Improves productivity across industries.

Limitations and Ethical Issues

1. **Hallucinations:** Model may generate incorrect or fabricated information.
2. **Bias in Data:** LLMs can inherit societal or dataset biases.
3. **Privacy Risks:** May unintentionally reveal sensitive patterns in training data.
4. **High Computation Cost:** Training and inference are resource-intensive.
5. **Misuse Potential:** Can be used for fake news, impersonation, or harmful content.

THANK YOU