Creating an LLM from scratch:

So far:

Input Text → Tokenization

→ Token Embeddings + Positional Embeddings

→ [Transformer Block] × N layers

→ Final LayerNorm

→ Linear Projection (vocab\_size)

→ Softmax

→ Output Probabilities

→ Sampling/Decoding Strategy

→ Output Text

Transformer block:

LayerNorm → Multi-Head Attention → Dropout + Residual → LayerNorm → FFN → Dropout + Residual

Possible improvements:

Words can have multiple meanings (polysemy), and different words can have the same meaning (synonyms)

Can 1 word have different embeddings for each meaning? Should synonyms have the same token embeddings?

Can logits use grammar predictions to reduce search length, or does it need the whole vocabulary?

Is softmax necessary to decode outputs? Seems redundant