ABHYASMITRA.IN

COMPREHENSIVE UNIVERSITY STUDY NOTES

NATURAL LANGUAGE PROCESSING

Unit V — Modern Neural NLP and Transformers

Topics Covered in this Unit

Neural NLP Foundations: Motivation, Limits of Statistical Models, Dense Embeddings, RNNs & LSTMs
The Attention Mechanism: Bottleneck Problem, Self-Attention & Scaled Dot-Product Formulation
Encoder–Decoder (Seq2Seq) Architecture: Evolution of Machine Translation (RBMT -> SMT -> NMT)
Text Summarization: Extractive Sentence Ranking vs. Abstractive Generative Seq2Seq Modeling
The Transformer Architecture: Multi-Head Attention, Sinusoidal Positional Encoding & Add-and-Norm Layers
Transformer Advantages: O(1) Path Length, Parallel Training & Vanishing Gradient Elimination
Contextual Embeddings & BERT: Bidirectional Encoder Representations, Masked LM (MLM) & NSP
BERT Input Representation: Token (WordPiece), Segment (EA/EB), Position Embeddings & [CLS]/[SEP] Tokens
Downstream Fine-Tuning: Question Answering (SQuAD), Text Classification, Named Entity Recognition
Modern Generative Applications: Conversational Chatbots & Retrieval-Augmented Generation (RAG)
High-Yield University Exam Question Bank with Detailed Model Answers, Architecture & Unit Summary

Compiled in a structured, easy-to-understand, textbook style format
Specially curated for B.Tech / B.E. / BCA / MCA / B.Sc Computer Science & Data Science Students

COURSE SYLLABUS & MAPPING

[UNIT V SYLLABUS — PRESCRIBED UNIVERSITY CURRICULUM]

This section contains the official syllabus for Unit V. Verify with your university curriculum mapping.

Unit V Syllabus Breakdown:

Unit V - Modern Neural NLP and Transformers: Neural NLP Overview: Motivation, limits of statistical models, Dense vector representations. Attention Mechanism: Intuition and motivation, Self-attention, Scaled dot-product attention, Advantages over traditional models. Encoder–Decoder Models: Sequence-to-sequence architecture, applications, machine translation methods (rule-based machine translation (RBMT), Statistical Machine Translation (SMT), Neural Machine Translation (NMT)), Text summarization. Transformer Architecture: Multi-head attention, Positional embeddings, Feedforward layers, Advantages over RNN. Contextual Embeddings: BERT architecture, Token embeddings, segment embeddings. Downstream tasks: QA, text classification. Modern Applications: Chatbots, Sentiment analysis, Document classification, Retrieval-Augmented Generation (RAG).

• Course Code: _____ • Semester / Year: _____

TABLE OF CONTENTS

5.1 Neural NLP Overview & Limitations of Statistical Models

Motivation for Neural NLP

Classical statistical NLP models (N-grams, HMMs) suffered from three catastrophic structural bottlenecks:

1. Severe Data Sparsity: As n-gram history grows, probability of unseen word sequences approaches 1.0.

2. Inability to Share Statistical Strength: 'The cat purred' provides zero information about 'A kitten meowed' because words are discrete, orthogonal symbols.

3. Short Context Window: Inability to capture dependencies spanning beyond 2–3 words.

Neural NLP solves these limitations by mapping discrete words into Continuous Dense Vector Spaces R^d and passing them through deep parameterized neural networks (RNNs, LSTMs, Transformers).

Recurrent Neural Networks (RNNs) and Gated Architectures (LSTM / GRU)

Standard RNNs process sequential tokens step-by-step, maintaining a recurrent hidden state: h_t = tanh(W_hh * h_{t-1} + W_xh * x_t + b_h).

- **The Vanishing & Exploding Gradient Problem:** During Backpropagation Through Time (BPTT), repeated matrix multiplications over long sequences cause error gradients to either vanish to zero (preventing long-range learning) or explode to infinity.
- **Long Short-Term Memory (LSTM - Hochreiter & Schmidhuber, 1997):** Introduces an explicit Cell State C_t and 3 multiplicative Gating Mechanisms: • Forget Gate: f_t = sigma(W_f * [h_{t-1}, x_t] + b_f) (controls what old information to discard). • Input Gate: i_t = sigma(W_i * [h_{t-1}, x_t] + b_i) (controls what new information to store). • Output Gate: o_t = sigma(W_o * [h_{t-1}, x_t] + b_o) (controls what hidden state h_t to emit).

5.2 The Attention Mechanism: Self-Attention & Scaled Dot-Product

Intuition & Motivation of Attention (Bahdanau et al., 2014)

In traditional Seq2Seq models, the Encoder compresses an entire source sentence of arbitrary length into a single fixed-size Context Vector C (the Information Bottleneck). For long sentences (> 20 words), crucial early information is lost.

The Attention Mechanism eliminates the bottleneck by allowing the Decoder to dynamically 'look back' and attend to all Encoder hidden states, computing a dynamically weighted context vector for every generated target word.

Scaled Dot-Product Attention (Vaswani et al., 2017)

Given input vectors projected into Query (Q), Key (K), and Value (V) matrices of dimension d_k:

Scaled Dot-Product Attention Mathematical Formula

*Attention(Q, K, V) = softmax((Q * K^T) / sqrt(d_k)) * V*

- *Q * K^T: Computes pairwise similarity compatibility scores between Query tokens and Key tokens.*
- *1 / sqrt(d_k): Scaling Factor preventing large dot-product magnitudes from pushing softmax into regions with vanishingly small gradients.*
- *softmax: Normalizes compatibility scores into a probability distribution summing to 1.0.*
- *Multiplying by V: Generates a weighted linear combination of Value vectors.*

5.3 Encoder–Decoder Seq2Seq & Machine Translation

Evolution of Machine Translation Paradigms

Paradigm	Era	Core Architecture & Mechanism	Key Strengths & Weaknesses
Rule-Based Machine Translation (RBMT)	1960s–1980s	Direct bilingual morphological dictionaries and complex hand-written syntactic transfer rules.	High grammatical precision in narrow domains; completely brittle on informal text; prohibitive human rule cost.
Statistical Machine Translation (SMT)	1990s–2000s	Bayesian formulation: P(Target|Source) propto P(Source|Target) * P(Target) using Phrase-Based Alignment and N-gram Language Models (Moses).	Robust and corpus-driven; struggles with long-distance syntactic reordering; produces disjointed outputs.
Neural Machine Translation (NMT)	2015–Present	End-to-end differentiable deep neural networks (Seq2Seq with Attention / Transformers) trained on parallel sentence pairs.	State-of-the-art fluency and semantic preservation; captures long-range dependencies; requires large GPU compute.

Text Summarization: Extractive vs. Abstractive

- • **Extractive Summarization:** Identifies, scores, and extracts the most important exact sentences directly from the original document (e.g., using TextRank graph ranking or TF-IDF sentence scoring).
- • **Abstractive Summarization:** Generates brand new sentences that paraphrase and condense the core ideas using an Encoder-Decoder Seq2Seq / Transformer model (e.g., T5, BART). Mimics human summarization.

5.4 The Transformer Architecture (Vaswani et al., 2017)

THE TRANSFORMER ARCHITECTURAL BLUEPRINT

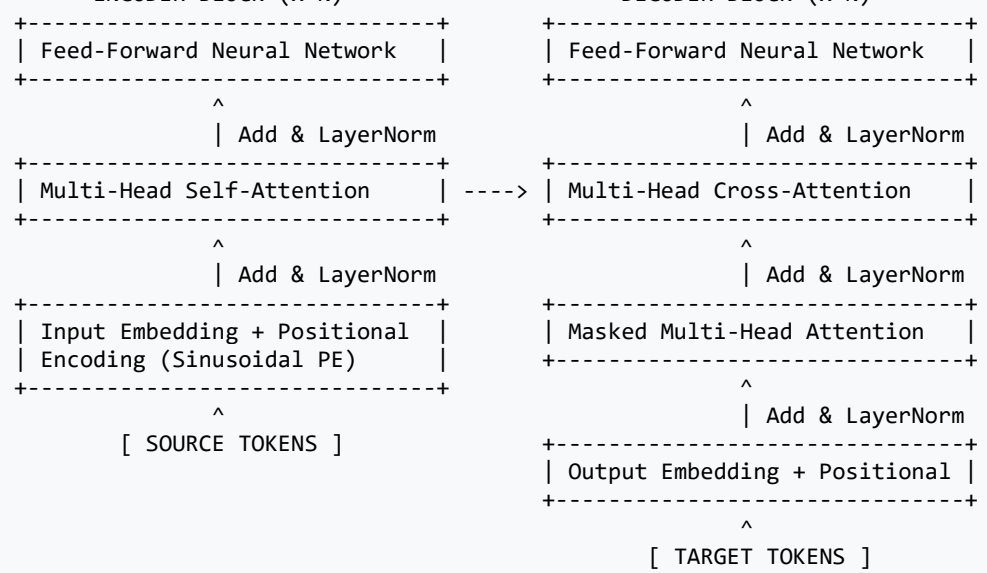

Figure: Complete Transformer Architecture with Multi-Head Self-Attention, Cross-Attention & Residuals

Core Architectural Components of the Transformer

- **1. Multi-Head Attention:** Linearly projects Q, K, V into h different representation subspaces in parallel, enabling the model to jointly attend to information at different syntactic and semantic positions: MultiHead(Q, K, V) = Concat(head_1, head_2, ..., head_h) * W^O where head_i = Attention(Q * W_i^Q, K * W_i^K, V * W_i^V).
- **2. Sinusoidal Positional Encoding:** Because Transformers lack recurrent loops, word order information is injected by adding sinusoidal position vectors directly to input embeddings: • PE_{(pos, 2i)} = sin(pos / 10000^(2i / d_model)) • PE_{(pos, 2i+1)} = cos(pos / 10000^(2i / d_model))

- **3. Residual Connections & Layer Normalization (Add & Norm):** Every sub-layer output is wrapped with a residual skip-connection followed by Layer Normalization: Output = LayerNorm(x + SubLayer(x)). Prevents vanishing gradients across deep stacks.
- **4. Position-Wise Feedforward Networks (FFN):** Two linear transformations with a ReLU or GELU activation: FFN(x) = max(0, x * W_1 + b_1) * W_2 + b_2.

Advantages of Transformers over Recurrent Neural Networks (RNNs)

Comparison Dimension	Recurrent Networks (RNN / LSTM)	The Transformer Architecture
Sequential Training Operations	O(n) sequential steps (cannot process step t until step t-1 completes).	O(1) sequential operations (100% parallel matrix operations over all tokens).
Maximum Path Length for Long Dependencies	O(n) intermediate network recurrence steps (prone to forgetting).	O(1) direct attention connections between any two arbitrary tokens.
Computational Scalability	Slow on modern multi-GPU / TPU hardware.	Highly optimized for GPU tensor parallelism; scales to hundreds of billions of parameters.
Vanishing / Exploding Gradients	Frequent in deep unrolled sequential graphs.	Completely mitigated via residual connections and layer normalization.

5.5 Contextual Embeddings & BERT Architecture

BERT: Bidirectional Encoder Representations from Transformers (Devlin et al., 2018)
BERT is a pretrained contextual language representation model based on a multi-layer Bidirectional Transformer Encoder. Unlike static embeddings (Word2Vec) where a word receives an identical static vector, BERT dynamically conditions on BOTH left and right context across all layers, generating context-aware polysemic representations.

BERT Input Embeddings Architecture

For every input token, BERT constructs its final input vector as the element-wise sum of three independent embedding representations:

- **1. Token Embeddings:** Constructed using WordPiece subword tokenization (30,000 subword vocabulary). Every sequence begins with special classification token `[CLS]` and sentences are separated by `[SEP]`.

- **2. Segment Embeddings:** Distinguishes between Sentence A (E_A) and Sentence B (E_B) in paired-sentence tasks (e.g., Question Answering, Natural Language Inference).
- **3. Position Embeddings:** Learned positional vectors (up to 512 tokens) encoding token sequential position.
- **Composite Input Representation:** E_input = E_Token + E_Segment + E_Position.

BERT Pre-Training Objectives

- **1. Masked Language Model (MLM - Cloze Task):** 15% of all input tokens are randomly selected: 80% are replaced with `[MASK]`, 10% with a random word, and 10% left unchanged. BERT is trained to predict the original masked words via cross-entropy loss. Enables true deep bidirectional context learning.
- **2. Next Sentence Prediction (NSP):** Trained on sentence pairs (A, B) to predict binary label `IsNext` (50% true adjacent sentences, 50% random sentence pairings). Crucial for Question Answering and Inference.

Downstream Fine-Tuning Tasks with BERT

- • **Text Classification / Sentiment Analysis:** Feed final hidden vector of the `[CLS]` token h_{[CLS]} into a single linear classification layer: y_hat = softmax(W * h_{[CLS]}).
- • **Question Answering (SQuAD):** Predicts the Start and End token span pointers of the answer within the passage by computing dot products with learned start/end vectors.
- • **Named Entity Recognition (NER):** Feeds each individual token's output vector h_i into a linear classification layer to predict its BIO tag.

5.6 Modern Generative NLP: Conversational AI & RAG

Retrieval-Augmented Generation (RAG - Lewis et al., Meta AI 2020)

Retrieval-Augmented Generation (RAG) combines dense semantic vector retrieval with autoregressive generative language models to ground AI responses in verified, real-time external knowledge databases, drastically reducing hallucinations.

The RAG Architectural Pipeline

RETRIEVAL-AUGMENTED GENERATION (RAG) ARCHITECTURE

```
[ USER QUERY ] ------------------------------------------------+
```

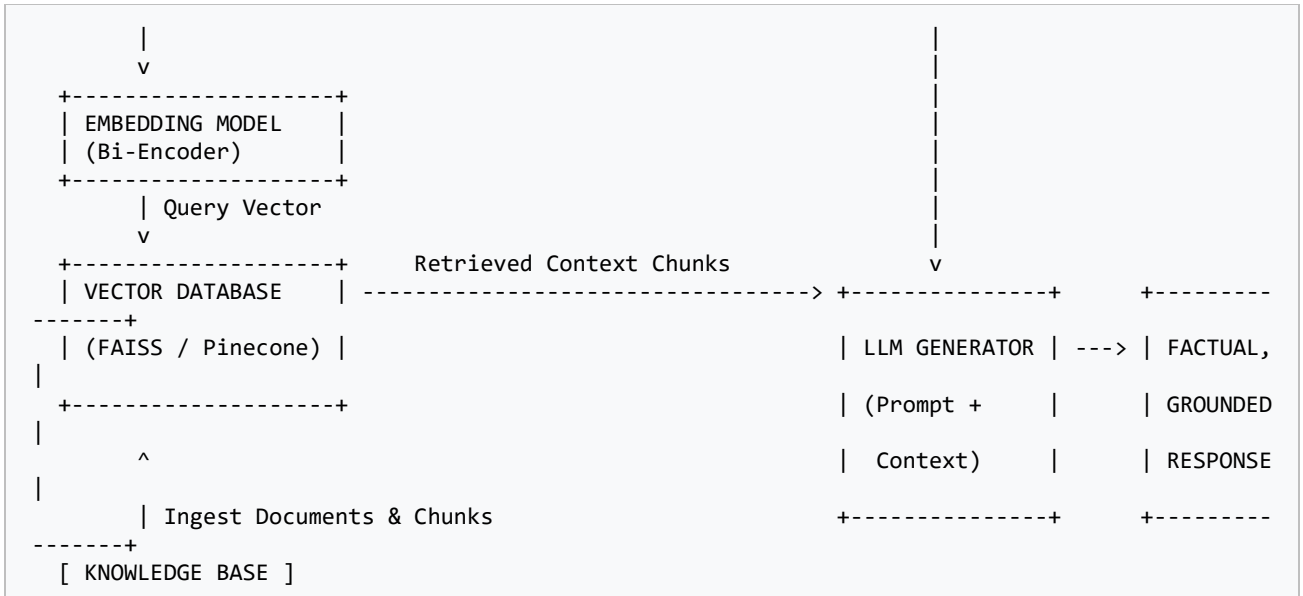

Figure: RAG retrieves top-k relevant document chunks to condition generative LLM responses

5.7 High-Yield University Exam Question Bank

Part A: Short Answer Questions (2 to 5 Marks)

Q1. State the Scaled Dot-Product Attention formula and explain why scaling is necessary. [3-4 Marks]

- **Answer:** Attention(Q, K, V) = softmax((Q * K^T) / sqrt(d_k)) * V. Scaling by 1/sqrt(d_k) prevents large dot-product magnitudes from pushing the softmax function into regions with near-zero gradients (vanishing gradients during backpropagation).

Q2. Why is Multi-Head Attention superior to Single-Head Attention? [3 Marks]

- **Answer:** Multi-Head Attention projects Queries, Keys, and Values into h distinct representation subspaces simultaneously. This allows the model to jointly attend to information at different positions and capture diverse relationships (e.g., syntactic vs semantic dependencies).

Q3. Explain the two pre-training objectives of BERT. [4 Marks]

- **Answer:** 1. Masked Language Model (MLM): Randomly masks 15% of tokens and trains the model to predict them using bidirectional context. 2. Next Sentence Prediction (NSP): Trains the model to predict whether sentence B logically follows sentence A in the corpus.

Part B: Long Answer Questions (10 to 15 Marks Outline Guides)

- **Q4. Draw and explain the complete Transformer Architecture (Vaswani et al., 2017) with Encoder, Decoder, Multi-Head Attention, and Positional Encodings. [15 Marks]**
- **Q5. Compare Recurrent Neural Networks (RNNs/LSTMs) vs. Transformers with respect to parallelism, path length, complexity, and vanishing gradients. [10 Marks]**
- **Q6. Explain BERT architecture, its three-layer input representation (Token, Segment, Position), pre-training methodology, and fine-tuning on Question Answering. [12 Marks]**
- **Q7. Explain Retrieval-Augmented Generation (RAG) architecture, dense vector search with Vector DBs, and how it solves LLM hallucination. [10 Marks]**

5.8 Unit Summary & Key Takeaways

- Neural NLP replaces discrete sparse n-grams with continuous dense vector spaces processed via deep neural networks.
- Attention eliminates the fixed-length Seq2Seq bottleneck by computing dynamic context weights: softmax(QK^T / sqrt(d_k)) * V.
- The Transformer architecture relies entirely on Multi-Head Self-Attention, Sinusoidal Positional Encoding, and Residual Layer Normalization.
- Transformers provide O(1) path length and 100% parallel training over sequential O(n) RNNs.
- BERT is a bidirectional Transformer Encoder pretrained via Masked LM (15% cloze task) and Next Sentence Prediction (NSP).
- Modern NLP leverages contextual fine-tuning (BERT), Conversational Chatbots, and Retrieval-Augmented Generation (RAG) for grounded AI.