Week 1 LLM Learning Plan: Timeline and Details

Your Name

May 26–June 2, 2025

Introduction

This document summarizes Week 1 (May 26–June 2, 2025) of an LLM learning plan, detailing daily tasks, timelines, steps, knowledge highlights, reference links, quiz answers, corrections, and improvements. Designed for a full-stack web engineer with Linux expertise, it covers tokenization, embeddings, RNNs, transformers, and fine-tuning, with outputs for GitHub reference.

Daily Timeline

Day 1: Monday, May 26, 2025, 7:00–8:00 PM IST

Task: NLP Basics and Tokenization
Intro: Introduced NLP fundamentals and tokenization as the first step in text processing.
Steps:

- Tokenized "I love learning about LLMs" with NLTK (20 mins).
- Read "The Illustrated Word2Vec" by Jay Alammar (20 mins).
- Watched 3Blue1Brown's "Neural Networks, Part 1" (20 mins).

Knowledge Highlights: Tokenization splits text into words/subwords for numerical processing. Word2Vec captures semantic relationships. Neural networks underpin LLMs.
References:

- The Illustrated Word2Vec
- 3Blue1Brown Neural Networks

Output: Tokenized: `['I', 'love', 'learning', 'about', 'LLMs']`.
Test Question: How does tokenization relate to word embeddings?
Your Answer: Tokenization helps create the root form of words, used with neighbors for embedding vectors.
Correction: Tokenization splits text into tokens (not always roots; cf. stemming).

Embeddings map tokens to vectors based on context.
Improvement: Clarified tokenization's role vs. stemming.
Status: Complete.

Day 2: Tuesday, May 27, 2025, 7:00–8:00 PM IST

Task: Word Embeddings Basics
Intro: Explored how embeddings encode word meanings as vectors.
Steps:

- Read Aravind CR's "Word Embeddings in NLP" (20 mins, replaced Hugging Face doc).
- Watched Stanford CS20N Lecture 30 clips (20 mins).
- Computed cosine similarity for "king" and "queen" with `gensim` (20 mins).

Knowledge Highlights: Word2Vec (Skip-gram) uses context; GloVe uses co-occurrence. High similarity reflects shared contexts.
References:

- Aravind CR's Article
- CS20N Lecture

Output: Similarity: 0.7839.
Test Question: Why high similarity for "king" and "queen"?
Your Answer: They occur frequently together (clarified: share similar contexts).
Correction: Similarity stems from similar contexts (e.g., royalty), not just co-occurrence.
Improvement: Reinforced GloVe's context-based vectors.
Status: Complete.

Day 3: Wednesday, May 28, 2025, 7:00–8:00 PM IST

Task: Neural Networks for NLP
Intro: Studied RNNs/LSTMs and their limitations vs. transformers.
Steps:

- Read "RNNs and LSTMs" by Christopher Olah (30 mins).
- Watched 3Blue1Brown's "Gradient Descent" (20 mins).
- Explained transformer advantages (10 mins).

Knowledge Highlights: RNNs process sequentially; LSTMs mitigate vanishing gradients. Transformers use parallel attention.
References:

- Olah's Article
- AssemblyAI Blog

Output: Transformers are faster (parallel) and handle longer contexts.
Test Question: Why do transformers outperform RNNs?

Your Answer: Transformers process in parallel and hold longer context.
Correction: Added: Attention overcomes vanishing gradients.
Improvement: Supplemented with transformer-RNN contrast resource.
Status: Complete.

Day 4: Thursday, May 29, 2025, 7:00–8:00 PM IST

Task: Intro to Transformers
Intro: Explored transformer architecture and BERT tokenization.
Steps:

- Read "The Illustrated Transformer" by Jay Alammar (40 mins).
- Tokenized "Transformers power modern LLMs" with BERT (20 mins).

Knowledge Highlights: Self-attention weights token relationships. BERT uses WordPiece tokenization.
References:

- The Illustrated Transformer

Output: Token IDs: `[101, 19081, 2373, 2715, 2222, 5244, 102]`. Decoded: `[CLS] transformers power modern llms [SEP]`.
Test Question: How does BERT's tokenization support attention?
Your Answer: Tokenization divides sentences; BERT's tokens define context boundaries for attention.
Correction: Tokens convert to IDs; [CLS]/[SEP] enable attention to process relationships.
Improvement: Clarified numerical input role.
Status: Complete.

Day 5: Friday, May 30, 2025, 7:00–8:00 PM IST

Task: Hands-On with Pre-trained Models
Intro: Applied transformers for sentiment analysis.
Steps:

- Followed Hugging Face's "Getting Started with Transformers" (30 mins).
- Classified sentiment of "I love AI" with BERT in Colab (30 mins).

Knowledge Highlights: BERT's pipeline simplifies NLP tasks. High scores reflect model confidence.
References:

- Hugging Face Quicktour

Output: Sentiment: POSITIVE, Score: 0.99983.
Test Question: Share sentiment score.
Your Answer: Correct score provided.
Correction: None needed.
Improvement: Reinforced pipeline usage.
Status: Complete.

Day 6: Saturday, May 31, 2025, 7:00–9:00 PM IST

Task: Mini-Project - Text Classifier
Intro: Fine-tuned BERT for IMDB sentiment classification.
Steps:

- Followed Hugging Face's "Fine-Tuning" tutorial (1 hour).
- Fine-tuned DistilBERT on IMDB dataset (1 hour).

Knowledge Highlights: Fine-tuning adapts pre-trained models. Accuracy reflects task fit.
References:

- Hugging Face Fine-Tuning

Output: Accuracy: 92.4% (epoch 3).
Test Question: Share accuracy.
Your Answer: 92.4% with detailed metrics.
Correction: None needed.
Improvement: Noted validation loss increase for Day 7 tweaks.
Status: Complete.

Day 7: Sunday, June 1, 2025 (Extended to June 2), 7:00–9:00 PM IST

Task: Review and Reflect
Intro: Consolidated Week 1, improved classifier, summarized learnings.
Steps:

- Revisited "The Illustrated Transformer" (20 mins).
- Improved Day 6 classifier (1 hour).
- Wrote 100-word summary (20 mins).
- Answered 5-question quiz (20 mins).

Knowledge Highlights: Attention, fine-tuning, and NLP progression.
References: Same as Day 4.
Output: Accuracy: 87.7%. Summary: Covered all days. Quiz: 92% score.
Quiz Answers and Corrections:

1. **Q**: What is tokenization? **A**: Splits text into chunks for numerical vectors. **C**: Correct; tokens map to IDs. **Score**: 5/5.
2. **Q**: How do embeddings capture meaning? **A**: Skip-gram/SGNS use surrounding words' context. **C**: Correct. **Score**: 5/5.
3. **Q**: Why transformers over RNNs? **A**: Transformers use attention for long contexts. **C**: Add parallel processing. **Score**: 4.5/5.
4. **Q**: What does attention do? **A**: Self-attention attends to inputs; FFNs parallelize. **C**: Correct; weights relationships. **Score**: 5/5.

5. **Q**: How does BERT process text? **A**: Embeds, uses transformers, predicts via [CLS]. **C**: Encoder-only, no decoder. **Score**: 3.5/5.

Improvement: Adjusted learning rate, early stopping; accuracy dropped due to underfitting.
Status: Complete.