# Deep Learning & NLP Engineer Interview Questions

## Text Processing and Representation

Q: What is the difference between stemming and lemmatization?

A: Stemming cuts off word endings to reduce to root form (e.g., 'running' -> 'run'), often inaccurately. Lemmatization uses vocabulary and morphology to return base or dictionary form (e.g., 'was' -> 'be'). Lemmatization is more accurate.

Q: Why is tokenization critical in NLP pipelines?

A: Tokenization splits text into meaningful units (tokens), which are the basic building blocks for further NLP tasks such as parsing, tagging, and classification.

## POS Tagging, Dependency Parsing, Topic & Language Modeling

Q: What are dependency parse trees? How are they useful?

A: Dependency parse trees represent grammatical structure by linking words based on their relationships. They're used in information extraction and machine translation.

Q: Explain Latent Dirichlet Allocation (LDA) for topic modeling.

A: LDA assumes documents are mixtures of topics and topics are mixtures of words. It uses a generative probabilistic model to assign topics to documents.

## Embeddings

Q: Difference between one-hot encoding, Word2Vec, GloVe, and BERT embeddings?

A: One-hot lacks context and dimensionality. Word2Vec and GloVe provide dense vectors capturing semantic meaning. BERT provides contextual embeddings that change with sentence context.

## RNN, GRU, LSTM, Seq2Seq

Q: Explain vanishing gradient problem and how LSTM mitigates it.

A: In RNNs, gradients shrink during backpropagation, affecting long-term learning. LSTM uses gates and memory cells to preserve information across time steps.

Q: What are the gates in LSTM? Provide their equations.

A: LSTM has Forget Gate, Input Gate, Output Gate.

ft = ->(Wf->[ht->1, xt] + bf)

it = ->(Wi->[ht->1, xt] + bi)

ot = ->(Wo->[ht->1, xt] + bo)

ct = ft * ct->1 + it * tanh(Wc->[ht->1, xt] + bc)

ht = ot * tanh(ct)

## Attention & Transformers

Q: What is the difference between global attention and local attention?

A: Global attention attends to all encoder hidden states; local attention restricts focus to a subset, reducing computation.

Q: What is positional encoding? Why do we need it?

A: Transformers lack recurrence, so positional encoding injects word order using sine and cosine functions to preserve sequence information.

## Model Interpretation (LIME, SHAP)

Q: What is the difference between LIME and SHAP?

A: LIME explains predictions locally using interpretable models. SHAP uses game theory to assign importance values to features globally and locally.

## Reinforcement Learning

Q: What is the Bellman Equation?

A: The Bellman Equation expresses the value of a state as the expected return from the best action, considering both immediate reward and future value.