**1. One-Hot Encoding**

Represents each word in a vocabulary as a binary vector. Each vector has a length equal to the size of the vocabulary, with one position marked as 1 (the word's index) and the rest as 0.

Example:
Vocabulary = ["cat", "dog", "fish"]
"cat" → [1, 0, 0]
"dog" → [0, 1, 0]

---

**2. Bag of Words (BoW)**

Represents text by counting how often each word appears, ignoring grammar and word order. A vocabulary is built, and each document is converted into a vector of word counts.

Example:
Docs = ["I love NLP", "NLP is fun"]
BoW vectors might be:
Doc1 = [1, 1, 1, 0, 0]
Doc2 = [0, 0, 1, 1, 1]

---

**3. Bag of N-Grams**

An extension of BoW that uses sequences of N words instead of single words. Captures partial word order.

Example (bi-grams):
"I love NLP" → ["I love", "love NLP"]

---

**4. TF-IDF**

Stands for Term Frequency–Inverse Document Frequency. It's a statistical measure that evaluates how important a word is to a document in a collection. Common words are downweighted, while rare but important words are upweighted.

---

**5. OOV (Out-of-Vocabulary) Problem**

Occurs when a model sees a word during testing that wasn’t present during training. The model doesn't know how to handle such unknown words.

Solution: Use subword techniques or large pre-trained embeddings.

---

**6. Word Embeddings**

Dense numerical representations of words where similar words have similar vectors. Unlike one-hot, embeddings capture semantic meaning.

Examples: Word2Vec, GloVe, FastText.

---

**7. Continuous Bag of Words (CBOW)**
A Word2Vec model that predicts a word based on its surrounding context words.
Example: Given the words "I" and "NLP", predict the missing word "love".

---

**8. Skip-Gram**
The opposite of CBOW. Given a single word, the model predicts the surrounding context words.
Example: Given "love", predict words like "I" and "NLP".

---

**9. GloVe Embeddings**
Stands for Global Vectors. It creates word vectors by analyzing how often words appear together in a large text corpus. Unlike Word2Vec, it uses word co-occurrence statistics across the whole corpus.