**Word Embedding Techniques**

There are 2 main categories of word embedding methods:

1. **Frequency-based embedding:** Embedding methods that utilize the frequency of words to generate their vector representations. Frequency-based methods use statistical measures of how often words appear in the corpus to encode semantic information.
2. **Prediction-based embeddings:** Generated by models that learn to predict words from their neighboring words in sentences. These methods aim to position words with similar contexts closely together in the embedding space. They often result in more nuanced word vectors that capture a wide array of linguistic relationships.

**Frequency-Based Embedding:**

**1. One hot encoding**

**One-hot encoding** is the simplest word representation technique where each word in the vocabulary is represented as a sparse binary vector of length equal to the vocabulary size, with a single 1 at the index corresponding to the word and 0s everywhere else. For example, in a vocabulary of 5 words, the word at position 3 would be represented as [0, 0, 1, 0, 0]. While easy to implement and useful for categorical data, one-hot encoding has major limitations: it results in very high-dimensional sparse vectors for large vocabularies, does not capture semantic meaning or relationships between words (e.g., king and queen are as distant as king and car), and cannot handle unseen words without retraining.

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

**Bag of Words (BoW)** is a word embedding technique that represents text as a vector of word counts or frequencies, ignoring grammar and word order but keeping track of word occurrence. For example, if the vocabulary is [cat, dog, eat, food] and the sentence is “dog eats food”, it might be represented as [0, 1, 0, 1] (since "dog" and "food" appear once, while "cat" and "eat" don’t appear). BoW is simple and effective for basic text classification tasks, but it produces high-dimensional sparse vectors for large vocabularies, fails to capture the semantic meaning of words, and treats different contexts or word orders as the same, limiting its ability to understand language nuances.

**3. Term Frequency-Inverse Document Frequency (TF-IDF)**

TF-IDF is a statistical measure used to evaluate the importance of a word to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

In the context of word embeddings, TF-IDF can be thought of as a very basic embedding technique, where words are represented as vectors of their TF-IDF scores across multiple documents. Despite its simplicity, TF-IDF can be effective in tasks such as information retrieval and text classification.

**4. Co-occurrence Matrix**

A co-occurrence matrix is a matrix that quantifies how often different words appear together in a corpus. In terms of word embeddings, each word is represented as a vector of its co-occurrence frequencies with other words.

This technique allows us to capture semantic relationships between words, as words that often appear together in the same context are likely to be semantically related. However, co-occurrence matrices can become very large and computationally expensive for large vocabularies.

**Prediction-Based Embeddings**

**5. Word2Vec (Skip-gram and Continuous Bag of Words)**

Word2Vec is a popular technique for learning word embeddings, based on neural networks that learn the optimal word representations by training on a large dataset. Word2Vec embeddings are efficient to compute and can capture complex linguistic patterns.

Word2Vec is a group of neural architectures that has two main variants:

* **Skip-gram:** The model is trained to predict the context given a word. This technique is particularly useful when dealing with less frequent words, as it weighs the representation of each context-word pair, allowing it to capture a broader range of semantic relationships.
* **Continuous Bag of Words (CBOW):** The model predicts the target word from its context. CBOW tends to predict common words better, since it smooths over the entire context and is therefore faster to train.

**6. FastText**

FastText is a prediction-based embedding technique that extends Word2Vec by considering subword information. This allows FastText to generate better embeddings for rare and out-of-vocabulary words.

FastText takes into account the internal structure of words while learning representations; it represents each word as a bag of character n-grams in addition to the word itself. For example, the word “apple” would be represented by the n-grams: “ap”, “pp”, “pl”, “le”, if we chose n=2, and also the whole word “apple” as a separate feature.

This technique is especially effective for morphologically rich languages, where a single word can have many different forms, and also helps in understanding suffixes and prefixes. FastText can also be used to generate word embeddings for words that didn’t appear in the training data, making it robust for handling real-world text from diverse sources.

**7. GloVe (Global Vectors for Word Representation)**

Unlike Word2Vec, which considers local context windows, GloVe learns embeddings by leveraging global word-word co-occurrence statistics from the corpus.

GloVe is based on matrix factorization techniques on the word-context matrix. It first constructs a large matrix of (words × context) co-occurrence information, essentially counting how frequently a “context” word appears with a “target” word. Then, it uses least squares regression to factorize this matrix, yielding a lower-dimensional representation.

Unlike Word2Vec, which is a predictive model, GloVe is a count-based model. This method allows for capturing intricate patterns in the data, such as linear substructures of the word vector space, which can correspond to linguistic concepts like gender, verb tense, and pluralization.