Feature extraction in text processing converts textual data into numerical data that machine learning models can work with. Let's simplify how two popular methods, Bag of Words (BoW) and tf-idf (Term Frequency-Inverse Document Frequency), do this:

1. Bag of Words (BoW)

What it does: It looks at the vocabulary of all the documents as a whole and counts how many times each word appears in each individual document. **How it works:**

- **Step 1:** Create a list (dictionary) of all unique words from all documents.
- **Step 2:** For each document, create a vector. The vector has a slot for every word in the dictionary.
- **Step 3:** Fill in the vector for each document by counting how many times each word appears in that document.

Example: Consider two documents:

- Doc 1: "Cat eats fish"
- Doc 2: "Fish eats fish"

Dictionary: [Cat, eats, fish]

- Vector for Doc 1: [1, 1, 1] (1 Cat, 1 eats, 1 fish)
- Vector for Doc 2: [0, 1, 2] (0 Cat, 1 eats, 2 fish)

2. Term Frequency-Inverse Document Frequency (tf-idf)

What it does: It measures how important a word is to a document in a collection of documents (corpus). It not only counts words but also adjusts for words that appear more frequently in the corpus.

How it works:

- **Term Frequency (tf):** Similar to BoW, count the number of times a word appears in a document. Then, normalize this by the total number of words in the document (this normalization helps to adjust for the size of the document).
- **Inverse Document Frequency (idf):** This reduces the weight of terms that appear very frequently across the document set and increases the weight of terms that appear rarely.
    - **Calculation:** It's calculated as the logarithm of the number of documents in the corpus divided by the number of documents where the term t appears.

Example: Using the same two documents:

- Assume we have a total of 100 documents, and the word "fish" appears in 5 of them.
- **tf for 'fish' in Doc 2:** tf=23\text{tf} = \frac{2}{3}tf=32 (appears 2 times out of 3 total words)
- **idf for 'fish':** idf=log⁡(1005)\text{idf} = \log(\frac{100}{5})idf=log(5100)
- **tf-idf for 'fish' in Doc 2:** tf-idf=tf×idf\text{tf-idf} = \text{tf} \times \text{idf}tf-idf=tf×idf

Tf-idf helps in understanding the relative importance of a word in a document in the context of a set of documents. The main difference from BoW is that tf-idf also considers the frequency of the word in the entire dataset, which helps to penalize too-common words and highlight important ones.

In practical terms, these feature extraction methods transform textual data into a structured, numeric format that machine learning algorithms can understand, allowing for tasks like classification, clustering, and recommendation based on textual input.