Text Feature Extraction: BoW & TF-IDF

📊 Bag of Words (BoW)

Represents text as word frequency counts

- Ignores word order and grammar
- Only considers word occurrence
- Simple frequency-based representation

⚖️ TF-IDF

Term Frequency-Inverse Document Frequency weighting

- Balances local and global importance
- Reduces weight of common words
- Increases weight of rare words

TF-IDF Components

TF (Term Frequency)
How often word appears in document (local importance)

×

IDF (Inverse Document Frequency)
How rare word is across documents (global importance)

✓ **Advantages:** Simple and interpretable

✗ **Limitations:** Loses semantic and syntactic information

📊 Bag of Words (BoW) - Practical Example

📄 Example Documents:

Doc 1: "I love machine learning"

Doc 2: "Machine learning is amazing"

Doc 3: "I love deep learning"

Step 1: Build Vocabulary

Unique words across all documents:

I | love | machine | learning | is | amazing | deep

Step 2: Count Word Frequencies

Document	I	love	machine	learning	is	amazing	deep
Doc 1	1	1	1	1	0	0	0
Doc 2	0	0	1	1	1	1	0
Doc 3	1	1	0	1	0	0	1

💡 **Result Interpretation:** Each document is represented as a 7-dimensional vector. Example: Doc 1 = [1, 1, 1, 1, 0, 0, 0] → Word order and grammar are ignored; only word occurrence counts are considered.

⚖️ TF-IDF - Practical Example

📄 Same Example Documents:

Doc 1: "I love machine learning"

Doc 2: "Machine learning is amazing"

Doc 3: "I love deep learning"

📐 TF-IDF Formula:

```
TF-IDF(word, doc) = TF(word, doc) × IDF(word)
```

```
TF(word, doc) = (word count in doc) / (total words in doc)
```

```
IDF(word) = log(total documents / documents containing word)
```

Example: Calculate TF-IDF for "learning" in Doc 1

Step 1: Calculate TF

TF("learning", Doc 1) = 1 / 4 = **0.25**

(appears 1 time / total 4 words)

Step 2: Calculate IDF

IDF("learning") = log(3 / 3) = log(1) = **0**

(3 documents / 3 documents containing "learning")

Step 3: Calculate TF-IDF

TF-IDF("learning", Doc 1) = 0.25 × 0 = **0**

→ "learning" appears in all documents, so it has low importance!

Example: Calculate TF-IDF for "deep" in Doc 3

Step 1: Calculate TF

TF("deep", Doc 3) = 1 / 4 = **0.25**

Step 2: Calculate IDF

IDF("deep") = log(3 / 1) = log(3) ≈ **1.099**

("deep" appears in only 1 document → rare word)

Step 3: Calculate TF-IDF

TF-IDF("deep", Doc 3) = 0.25 × 1.099 ≈ **0.275**

→ "deep" is a rare word, so it has high importance!

💡 **Key Point:** TF-IDF reduces the weight of common words (e.g., "learning") and increases the weight of rare words (e.g., "deep") to better represent document characteristics.