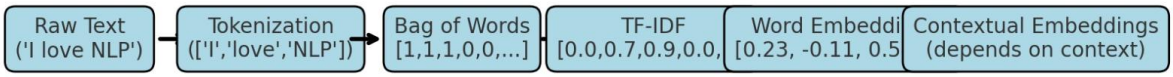
Natural language processing



1. Raw Text

Just normal sentences written by humans.

- Doc1: "Cats chase mice"
- Doc2: "Dogs chase cats"

---

2. Tokenization

Split sentences into individual words (tokens).

- Doc1 → ["Cats", "chase", "mice"]
- Doc2 → ["Dogs", "chase", "cats"]

(Case usually normalized: lowercase)

- Doc1 → ["cats", "chase", "mice"]
- Doc2 → ["dogs", "chase", "cats"]

---

3. Bag of Words (BoW)

We build a **vocabulary** of unique words across all docs:
["cats", "chase", "mice", "dogs"]

Now, count frequency in each document:

- Doc1: "cats chase mice" → [1, 1, 1, 0]

- Doc2: "dogs chase cats" → [1, 1, 0, 1]

Problem: Counts only, no importance weighting.

---

4. TF-IDF (Term Frequency – Inverse Document Frequency)

Words that appear in many documents become less important. Rare words get higher weight.

Here:

- Word "chase" appears in **both docs** → lower weight.
- Words "mice" and "dogs" appear only once → higher weight.

So the vectors might look like (approx):

- Doc1: [0.45, 0.2, 0.85, 0.0]
- Doc2: [0.45, 0.2, 0.0, 0.85]

Better than BoW since "chase" is treated as common and down-weighted.

---

5. Word Embeddings (Word2Vec, GloVe)

Instead of counts, each word is mapped to a **dense vector** learned from large corpora.

Example (fake vectors just to show idea):

- "cats" → [0.21, -0.34, 0.88]
- "dogs" → [0.19, -0.30, 0.80]
- "mice" → [0.25, -0.40, 0.72]

Notice: "cats" and "dogs" have similar vectors (they're both animals), "mice" is somewhat close too.

Captures meaning: words with similar contexts get similar vectors.

---

6. Contextual Embeddings (BERT, GPT, etc.)

The **same word changes its vector depending on context**.

Example:

- "cats" in "Cats chase mice" → vector might emphasize predator role.
- "cats" in "Dogs chase cats" → vector might emphasize prey role.

So "cats" doesn't always map to the same vector → depends on sentence meaning.