GROKKERS
AI FOR EVERYONE

Vectorization & Embedding

"Textual Alchemy: Words to numbers"

Topics to cover

- Bag of words
- Count vectorizer / Binary representation
- TF-IDF vectorizer
- Online ML with count/ TF-IDF
- Hashing Vectorizer
- BM25 & variants
- Word vectors (word2Vec)
- Glove/ FastText
- Doc2Vec
- Spacy – vectorizer
- Text Classification (demo) using IMDB

BOW/Count

"Words in Numbers:

Vectorizer – BOW/Count"

Bag of Words (BoW)

- BoW is one of the simplest vectorization techniques.
- represents a document as a vector where each element corresponds to a unique word in the entire corpus
- the value in each element represents the frequency of that word in the document.

TEXT CORPUS (10000 PDFs, 100 pages each)

"The Impact of Artificial Intelligence on Society

…

…

…

Artificial Intelligence (AI) has emerged as a transformative force in the modern world. Its rapid development and integration into various aspects of our lives have sparked both excitement and concern. In this essay, we will explore the profound impact of AI on society, touching upon its applications in healthcare, education, the job market, ethics, and more…."

Unique set of words

10000 words

index	Words (sorted)
0	AI
1	Artificial
2	Amit
3	Aron
..	..
1000	Bala
1050	Brijesh
1051	wins
..	..
5000	Storm
8500	Zebra
9000	Zoha
,..	,..

Bag of words

Input Document

"AI wins, always wins"

Vector Representation

0	1	2	3	..	1000	1051	..	9000	10000
1	0	0	0	0	0	2	0	0	0

Intuition of BOW

Why we need corpus for BOW

Vocabulary and Lexicon

A corpus is the foundation for creating a comprehensive vocabulary or lexicon.

includes a vast array of words and phrases

Statistical Analysis:

calculating word frequencies for next word predictions

collocations (words that tend to appear together)

n-grams (sequences of N words).

Example - BOW

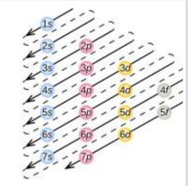
- Let's say we have two documents:
  - Document 1: "I love machine learning."
  - Document 2: "Machine learning is fascinating, and I enjoy it."

- We create a BoW representation by considering the unique words in both documents:
  - Vocabulary: ["I", "love", "machine", "learning", "is", "fascinating", "and“, "enjoy", "it"]

- We then create vectors for each document based on word frequency:
  - Document 1 BoW: [1, 1, 1, 1, 0, 0, 0, 0, 0]
  - Document 2 BoW: [1, 0, 1, 1, 1, 1, 1, 1, 1]

Key characteristics of BOW

Word Frequency: represents a document as a vector where each element corresponds to a unique word in the vocabulary and the value in each element represents the count (or sometimes a binary indication of presence/absence) of that word in the document.



Order Ignored: BoW completely ignores the order of words within a document.

Key characteristics of BOW

Vocabulary: BoW requires a pre-defined vocabulary, which is created by collecting all unique words across the entire corpus of documents.

The vocabulary size determines the dimensions of the BoW vectors.

Sparse Representation: BoW vectors are typically sparse, especially in large vocabularies, because most documents only contain a subset of the words in the vocabulary.

This sparsity can be computationally efficient but may require specialized data structures for storage.

Key characteristics of BOW

Scalability: BoW is scalable to large datasets and can be applied to a wide range of text analysis tasks, including document classification, sentiment analysis, and information retrieval.

Text Preprocessing: Before applying BoW, text preprocessing steps like tokenization (splitting text into words or tokens), stop word removal (excluding common and uninformative words), and stemming (reducing words to their base form) are often performed to improve the quality of representations.

Key characteristics of BOW

- **Document Similarity:** BoW can be used to measure the similarity between documents by calculating the cosine similarity or other distance metrics between their corresponding vectors.

- **Dimensionality**: The dimensionality of BoW vectors can be quite high, especially for large vocabularies or extensive corpora.

- This high dimensionality can impact computational resources and may require techniques like dimensionality reduction.

Key characteristics of BOW

Bag of N-grams: While BoW is typically based on individual words (unigrams), it can be extended to include multi-word sequences (n-grams) to capture more context. For example, "New York" might be treated as a single feature.

Application Flexibility: BoW is versatile and can be used in various NLP tasks, is particularly suitable for tasks where word frequency information is essential but may not perform well on tasks requiring understanding of word order and semantics.

Baseline Model: BoW often serves as a baseline or **starting point** for more complex NLP models.

Illustrations (BOW)

- **Documents:**
  - Document 1: "I love programming."
  - Document 2: "Programming is fun."
  - Document 3: "Coding is interesting."

Step by step

Step 1: Tokenization

Tokenize each document by breaking it into individual words and converting them to lowercase, removing punctuation.

Document 1: ["i", "love", "programming"]

Document 2: ["programming", "is", "fun"]

Document 3: ["coding", "is", "interesting"]

Step 2: Vocabulary Construction

Create a vocabulary by considering all unique words across the documents.

Vocabulary: ["i", "love", "programming", "is", "fun", "coding", "interesting"]

Step 3: Bag-of-Words Representation

Represent each document as a binary vector, where the values correspond to the presence (1) or absence (0) of each word in the vocabulary.

Document 1: [1, 1, 1, 0, 0, 0, 0]

Document 2: [0, 0, 1, 1, 1, 0, 0]

Document 3: [0, 0, 1, 1, 0, 1, 1]

Explanation

Document 1:

"i" is present, so the first element is 1.

"love" is present, so the second element is 1.

"programming" is present, so the third element is 1.

The rest of the elements are 0 because those words are not present.

Document 2:

"programming" is present, so the third element is 1.

"is" is present, so the fourth element is 1.

"fun" is present, so the fifth element is 1.

The rest of the elements are 0 because those words are not present.

Document 3:

"programming" is present, so the third element is 1.

"is" is present, so the fourth element is 1.

"coding" is present, so the sixth element is 1.

"interesting" is present, so the seventh element is 1.

The rest of the elements are 0 because those words are not present.

Thanks!

GROKKERS
AI FOR EVERYONE