Assignment No: - 2

Name :- Isha Ghorpade

Enrollnment :- 22420020

Roll :- 381066

Problem Statement

Perform bag-of-words approach (count occurrence, normalized count occurrence), TF-IDF on data.

Create embeddings using Word2Vec

Objectives

- To understand text representation techniques used in Natural Language Processing (NLP).
- To study the Bag-of-Words approach using count occurrence and normalized count.
- To learn the TF-IDF technique for measuring word importance.
- To understand word embeddings using the Word2Vec model.
- To analyze how numerical text representations help machine learning models.

Software Packages Used

- Operating System: Windows / Linux / macOS
- Programming Language: Python 3.x
- Development Tools: Jupyter Notebook / Google Colab / VS Code
- Libraries Used:
  - NLTK
  - Scikit-learn
  - Gensim

Hardware Apparatus Used

- Computer or Laptop
- Processor: Intel / AMD or equivalent
- Minimum 4 GB RAM
- Internet connection (for downloading datasets and libraries)

Theory

Text Representation

Text representation is the process of converting text into numerical form so that machine learning algorithms can process it.
Common techniques include Bag-of-Words, TF-IDF, and Word Embeddings.

Bag-of-Words (BoW) Model

Bag-of-Words represents text by counting how many times each word appears in a document.
It ignores grammar and word order and focuses only on frequency.

---

1. Count Occurrence (Term Frequency)

- Counts the number of times a word appears in a document.
- Simple and easy to understand.

Example Diagram

Document: I love NLP and I love AI

Vocabulary: [I, love, NLP, and, AI]

Count Vector:

I → 2

love → 2

NLP → 1

and → 1

AI → 1

2. Normalized Count Occurrence

- Count of each word divided by total number of words.
- Helps compare documents of different lengths.

Example Diagram

Total words = 7

Normalized Frequency:

I → 2/7

love → 2/7

NLP → 1/7

and → 1/7

AI → 1/7

Limitations of Bag-of-Words

- Ignores word meaning and context.
- Cannot capture similarity between words.
- Vocabulary size can be very large.

TF-IDF (Term Frequency – Inverse Document Frequency)

TF-IDF improves BoW by reducing the importance of common words and increasing the importance of rare but meaningful words.

Components of TF-IDF

Term Frequency (TF): How often a word appears in a document.

Inverse Document Frequency (IDF): How rare the word is across all documents.

TF-IDF Formula (Conceptual)

TF-IDF = TF × IDF

Example Diagram

Word: "NLP"

TF → High (appears often in document)
IDF → High (appears in few documents)
TF-IDF Score → High importance

Advantages of TF-IDF

- Reduces impact of common words like "is", "the", "and".
- Highlights important keywords.
- Better than simple frequency counting.

Word Embeddings

Word embeddings represent words as **dense numerical vectors** that capture semantic meaning and relationships.

Word2Vec

Word2Vec is a popular word embedding technique that learns word meanings based on surrounding words.

Word2Vec Models

- **CBOW (Continuous Bag of Words):**
  Predicts a word using surrounding context.
- **Skip-Gram:**
  Predicts surrounding words using a target word.

Word2Vec Diagram

king → [0.21, 0.45, 0.67]

queen → [0.22, 0.46, 0.66]

Similarity: king ≈ queen

Advantages of Word2Vec

- Captures semantic relationships.
- Words with similar meanings have similar vectors.
- Efficient for large datasets.

Comparison Diagram

Text

↓

Bag-of-Words → Frequency based

↓

TF-IDF → Importance based

↓

Word2Vec → Meaning & context based

Applications

- Text classification
- Sentiment analysis
- Recommendation systems
- Search engines
- Chatbots and NLP systems

Conclusion

Bag-of-Words and TF-IDF convert text into numerical form using frequency-based methods. Word2Vec provides meaningful vector representations by capturing semantic relationships. These techniques form the foundation of many modern NLP and machine learning applications.
