Word Feature Representation, Vectorization, Bag of Words and Word2Vec: Conceptual and Numerical Explanation

Dr. Sambit Praharaj, Assistant Professor (II), KIIT

1 Introduction

Natural Language Processing (NLP) systems cannot directly process raw text. Text must first be converted into numerical form through **feature representation and vectorization**. This document explains:

- Word feature representation and vectorization
- Bag of Words (BOW)
- Word2Vec (CBOW and Skip-Gram)
- Step-by-step numerical training of CBOW

2 Word Feature Representation and Vectorization

Words are symbolic, whereas machine learning models require numerical input.

Definition

- **Word feature representation**: Mapping words to numerical features
- **Vectorization**: Converting text into vectors using these features

Types of Representation

- Count-based: Bag of Words, TF-IDF
- Prediction-based: Word2Vec

3 Bag of Words (BOW)

Bag of Words represents text as a vector of word frequencies.

Example

I love NLP

Vocabulary:

$$[\text{I}, \text{love}, \text{NLP}]$$

BOW Vector:

$$[1, 1, 1]$$

Limitations

- Ignores word order
- Ignores context
- Produces sparse vectors

4 Word2Vec Overview

Word2Vec is a prediction-based model that learns dense word embeddings using a shallow neural network. It has two architectures:

- Continuous Bag of Words (CBOW)
- Skip-Gram

5 CBOW Model

CBOW predicts a target word given its surrounding context words.

Corpus

I love NLP

Window Size

Window size = 1

Training Pair

$$[\text{I}, \text{NLP}] \rightarrow \text{love}$$

6 Step-by-Step Training of CBOW (Numerical Example)

Step 1: Vocabulary and Indexing

$$[\text{I}, \text{love}, \text{NLP}]$$

Index mapping:

$$\text{I} \rightarrow 0, \quad \text{love} \rightarrow 1, \quad \text{NLP} \rightarrow 2$$

Vocabulary size $V = 3$

Step 2: One-Hot Encoding

$$\text{I} = [1, 0, 0]$$

$$\text{love} = [0, 1, 0]$$

$$\text{NLP} = [0, 0, 1]$$

Step 3: Initialize Weight Matrices

Embedding dimension $d = 2$

Input-to-Hidden Weights (W_1)

$$W_1 = \begin{bmatrix} 0.2 & 0.4 \\ 0.6 & 0.1 \\ 0.5 & 0.3 \end{bmatrix}$$

Each row corresponds to a word embedding.

Hidden-to-Output Weights (W_2)

$$W_2 = \begin{bmatrix} 0.1 & 0.3 & 0.2 \\ 0.4 & 0.2 & 0.5 \end{bmatrix}$$

Step 4: Compute Context Vector

Context words: I and NLP

$$\text{Embedding(I)} = [0.2, 0.4]$$

$$\text{Embedding(NLP)} = [0.5, 0.3]$$

Average context vector:

$$h = \frac{[0.2, 0.4] + [0.5, 0.3]}{2} = [0.35, 0.35]$$

Step 5: Compute Output Scores

$$\text{scores} = h \times W_2$$

$$= [0.35, 0.35] \begin{bmatrix} 0.1 & 0.3 & 0.2 \\ 0.4 & 0.2 & 0.5 \end{bmatrix}$$

$$= [0.175,\ 0.175,\ 0.245]$$

Step 6: Apply Softmax

$$P(w_i) = \frac{e^{score_i}}{\sum_j e^{score_j}}$$

Predicted probabilities:

$$[0.33,\ 0.33,\ 0.34]$$

Step 7: Compute Error and Update Weights

Target word:

$$\text{love} = [0, 1, 0]$$

The error between prediction and target is computed using cross-entropy loss. Backpropagation updates W_1 and W_2.

This process repeats over the corpus until convergence.

7 Skip-Gram Model

Skip-Gram predicts surrounding context words given a target word.

Example

$$\text{love} \to \text{I}$$

$$\text{love} \to \text{NLP}$$

8 Dense Word Embeddings

After training, the rows of W_1 form dense word embeddings.

Properties

- Dense and low-dimensional
- Capture semantic similarity
- Learned automatically from data

Vector Arithmetic

$$\text{king} - \text{man} + \text{woman} \approx \text{queen}$$

9 Limitation of Word2Vec

Word2Vec produces **static embeddings**. The same word has the same vector regardless of context.

Example

- bank (financial meaning)
- bank (river side)

Both meanings share the same embedding.

10 Summary Comparison

Feature	BOW	Word2Vec
Vector type	Sparse	Dense
Context usage	No	Yes
Semantic meaning	No	Yes
Learning method	Counting	Prediction

11 Conclusion

Word feature representation and vectorization are fundamental to NLP. While Bag of Words provides a simple frequency-based approach, Word2Vec learns dense semantic embeddings through predictive training using CBOW and Skip-Gram models.