Embeddings

1. Introduction

In Natural Language Processing (NLP), one of the fundamental challenges is converting human language into a form that machines can understand and process. Since computers work with numbers, words and sentences must be represented numerically. Embeddings provide a powerful method for doing this by representing words, phrases, or entire texts as dense vectors of real numbers.

---

2. What Are Embeddings?

An **embedding** in NLP is a mapping of a word or text from a symbolic space (like plain text) to a continuous vector space. Each word is represented as a vector of fixed length, where similar words tend to have similar vectors.

2.1 Dense Vector Representation

Unlike one-hot encoding, which creates sparse and high-dimensional vectors, embeddings generate **dense vectors** in lower dimensions (e.g., 100 to 300 dimensions), making them more efficient and meaningful.

Example:

- One-hot for "apple" in a 10,000-word vocabulary: `[0, 0, 0, ..., 1, ..., 0]` (10,000 dimensions, mostly 0s)
- Word embedding for "apple": `[0.15, -0.27, 0.03, ..., 0.91]` (e.g., 300 dimensions, real numbers)

---

3. How Embeddings Are Learned

Embeddings are usually learned from large text corpora using neural network models. These models are trained to predict relationships between words, such as context or similarity.

3.1 Common Methods

- **Word2Vec**: Learns embeddings by predicting a word based on its surrounding context (or vice versa).
- **GloVe (Global Vectors)**: Uses word co-occurrence statistics to learn embeddings.
- **FastText**: Builds word vectors using subword information, capturing morphology.
- **BERT and Transformer-based models**: Generate **contextual embeddings**, where the vector for a word changes depending on its surrounding words.

---

4. Types of Embeddings in NLP

Type	Description
Word embeddings	Fixed vector for each word, regardless of context.
Subword embeddings	Includes character-level info, useful for rare or misspelled words.
Contextual embeddings	Word meaning depends on sentence context (e.g., BERT).
Sentence/document embeddings	Represent entire sentences or paragraphs as single vectors.

---

5. Characteristics of Good Embeddings

- Capture **semantic similarity** (e.g., “king” and “queen” are close).
- Reflect **syntactic roles** (e.g., verbs cluster together).
- Encode **linguistic structure** implicitly from training data.

Example: Word Embeddings with Word2Vec

Suppose we train a **Word2Vec** model on a large English text corpus. After training, the model generates a 300-dimensional embedding vector for each word in the vocabulary. These embeddings capture relationships between words based on their **context** (i.e., surrounding words).

Let’s consider the following vectors (simplified for explanation):

- `vec("king") = [0.52, 0.10, ..., 0.75]`
- `vec("man") = [0.45, -0.20, ..., 0.60]`
- `vec("woman") = [0.47, -0.18, ..., 0.67]`
- `vec("queen") = [0.54, 0.12, ..., 0.82]`

We can use **vector arithmetic** to explore relationships:

vec("king")−vec("man")+vec("woman")≈vec("queen")

This shows that the model has learned not just the meanings of individual words, but also the **semantic relationships** between them—like **gender analogies** in this case.

Such relationships are not hard-coded but **emerge naturally** during training by analyzing patterns of word usage across the text data.

Example: BERT Contextual Embeddings

Sentences

We’ll use the word **"bank"** in two different contexts:

1. **Sentence A**: "He went to the **bank** to deposit money."
→ Here, **"bank"** = **financial institution**
2. **Sentence B**: "She sat by the **bank** of the river and watched the water flow."
→ Here, **"bank"** = **side of a river**

How BERT Handles This

- BERT uses a **transformer architecture** that looks at all words in a sentence at once (bidirectional attention).
- It assigns a **unique vector** to "bank" in each sentence **based on surrounding words**.

Embedding Output (Simplified)

Let’s say we extract the embedding for **"bank"** in both sentences using `bert-base-uncased`.

Word in Context	BERT Embedding (first 5 dimensions shown)
"bank" in Sentence A	`[0.12, 0.53, -0.21, 0.34, 0.89, ...]`
"bank" in Sentence B	`[0.05, 0.12, -0.45, 0.77, 0.34, ...]`

Even though it's the **same word**, the embeddings are **clearly different**, because BERT understands their **contextual meaning**.

Cosine Similarity (Semantic Distance)

The similarity between the two embeddings can be measured using **cosine similarity**:

$\text{cosine\_similarity}(\text{bankA}, \text{bankB}) \approx 0.38$

> A score close to 1 means **same meaning**, close to 0 means **different meaning**. So, 0.38 indicates that BERT recognizes a **difference in meaning** between the two "bank" usages.