POST 15 - GEN AI

UNDERSTANDING THE CONCEPT OF “WORD EMBEDDINGS” IN GENERATIVE AI & LLM

Generative AI Deep Dives,
Key concepts for Transformers - Part 2

GENERATIVE AI FOR ALL

DINESHLAL

DINESH LAL
(DIRECTOR, DATA SCIENCE)

WHAT IS COVERED IN THIS DOCUMENT?

- This document explains the topic **Word Embeddings**
- The definition of Word Embeddings, explanation with the example
- We are further discussing the key steps of creating Word Embeddings from given set of words
- The process is explained with a simple example of 5 words.

DINESHLAL

DEFINITION

Word Embeddings:

- **Word embeddings are dense, continuous vector representations of words.**
- **Think of it like a way of representing words with some set of numbers, so that machines can understand it better.** Embeddings are numerical representations of tokens.
- They act like a kind of code that captures the meaning and context of a word or subword within a specific language model.

EXAMPLE

Word	Embedding Vector (Hypothetical)
King	**[0.8, 0.2, 0.5, ...]** (lower values for "feminine" associations, higher for "power")
Queen	**[0.7, 0.3, 0.4, ...]** (similar to "king" but with slightly different values)
Man	**[0.3, 0.7, 0.1, ...]** (lower values for "royalty" associations, higher for "masculine" associations)

EXAMPLE

EXAMPLE (CONTINUED)

- These are simplified examples with only a few dimensions (usually there are hundreds or even thousands).
- The actual values wouldn't be interpretable individually, but the relative positions hold meaning.
- "King" and "queen" have more similar embeddings due to their relatedness, while "man" is positioned differently due to its contrasting meaning.
- This is a hypothetical representation; real embeddings are learned by the model during training on massive amounts of text data.

CREATION OF WORD EMBEDDINGS

Imagine we're building a transformer model and want to create word embeddings for a small vocabulary of just five words:

1. "king",
2. "queen",
3. "princess",
4. "prince", and
5. "man".

In the next few slides we will discuss the key 4 steps.

Step 1: Initializing Embedding Matrix:

1

- First, we create an embedding matrix.
- This is essentially a large table where each row represents a word in our vocabulary and each column represents a dimension in the embedding space.
- Let's choose an embedding size of 50 dimensions (meaning each embedding vector will have 50 values).
- Our initial embedding matrix might look something like this (Next Page):

CREATION EXPLAINED: STEP 1

Word	Embedding Vector (Hypothetical)
King	[-0.01, 0.03, ..., 0.22] (50 random values)
Queen	[-0.01, 0.03, ..., 0.22] (50 random values)
Princes	[-0.23, 0.87, ..., 0.15] (50 random values)
Prince	[-0.23, 0.87, ..., 0.15] (50 random values)
Man	[-0.01, 0.03, ..., 0.22] (50 random values)

Step 2: Training the Model::

2

- Now comes the magic part - training the model on a large corpus of text data.
- This data should ideally contain sentences with these words used in various contexts.
- During training, the model will iteratively adjust the values in the embedding matrix based on the observed relationships between words in the training data.

Step 2: Training the Model::

- Let's consider an example sentence: "The king and queen ruled the kingdom with justice."
- The model will process this sentence, analyzing the relationships between words. It might notice that "king" and "queen" often appear together and have similar roles. This would nudge the model to adjust the embedding vectors for "king" and "queen" in the embedding matrix to be closer together in the vector space.

Step 3: Refining Embeddings:

3

- As the model processes more sentences, it continues to refine the embedding matrix.
- Words with similar meanings (like "king" and "queen" or "prince" and "princess") will have their embeddings gradually move closer in the vector space.
- Words with contrasting meanings (like "king" and "man") will have their embeddings move further apart.

Step 4: Observing the Results:

4

- After training on a significant amount of text data, the embedding matrix will reflect the learned relationships between words.
- We won't be able to directly interpret the individual values in the embedding vectors, but the relative positions of these vectors will hold meaning.
- For instance, the embedding for "queen" might be closer to the embedding for "princess" than the embedding for "man" in the vector space. This reflects the semantic relationship between these words learned by the model through the training process.

Key Points:

- The initial embedding vectors are random and meaningless.
- Training on a large corpus refines the embeddings based on word co-occurrence and context.
- Similar words will have closer embeddings in the vector space.
- Embedding size determines the complexity of the relationships the model can capture.

Remember:

This is a simplified example with a tiny vocabulary. Real-world transformer models are trained on massive datasets with thousands or even millions of words, resulting in highly nuanced and informative word embeddings.

Thank You

SPECIAL THANKS TO CHATGPT, OPEN AI, COPILOT FOR THE SUPPORT ON CONTENT

DINESHLAL