NLP

Word Embeddings

→ Word embeddings are representations of words in a continuous vector space where words with similar meanings have similar representations. They are used to capture semantic relationships between words.

→ Word Embeddings or Word vectorization is a methodology in NLP to map words or phrases from vocabulary to a corresponding vector of real numbers which used to find word predictions, word similarities/semantics.

The process of converting words into numbers are called Vectorization.

Vocabulary – unique words – 10K

Text → convert → Vector (Real Numbers)

Word Embeddings

I love NLP → ['I', 'love', 'NLP']

Sentence – Word

↓ context

↓

vector

Real Number

Semantics → Understanding meaning in context

Consider the sentences

(i) He is a bright student. → Intelligence

(ii) The sun is very bright today. → Intensity of sun light

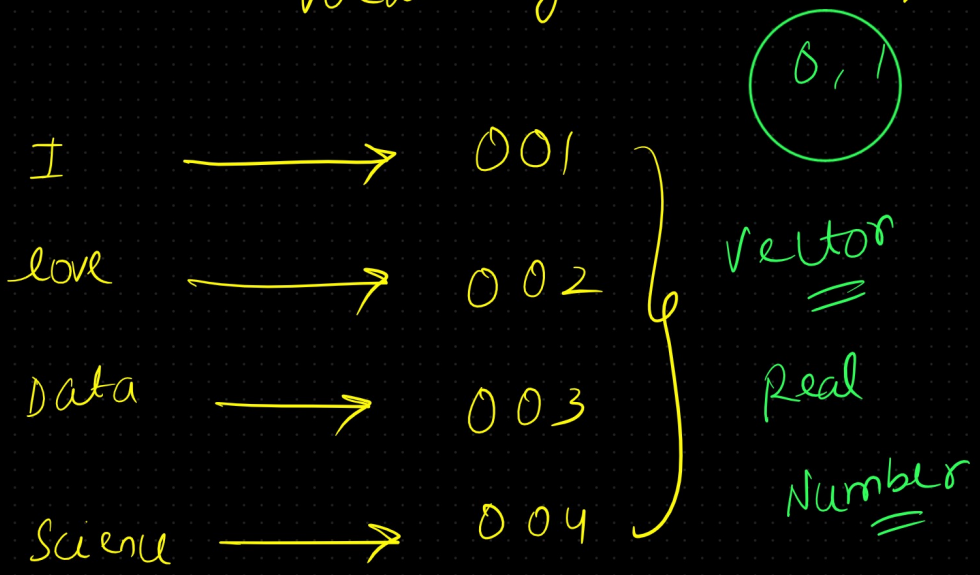
Eg → I love Data Science

→ ["I", "love", "Data", "Science"]

⇓ Vectorization

[001, 002, 0003, 0004]

Vocabulary → 10K unique words



One-Hot Encoding

love

1 0 0 0

0 1 0 0

0 0 1 0

King — Queen
Apple — Fruits
} Similar Repr^n

(i) Word 2 Vec → From Google
(ii) Fasttext → From Facebook
(iii) Glove → From Standford

Word 2 Vec

Word repr^n in vector space

King : Man : woman : ?

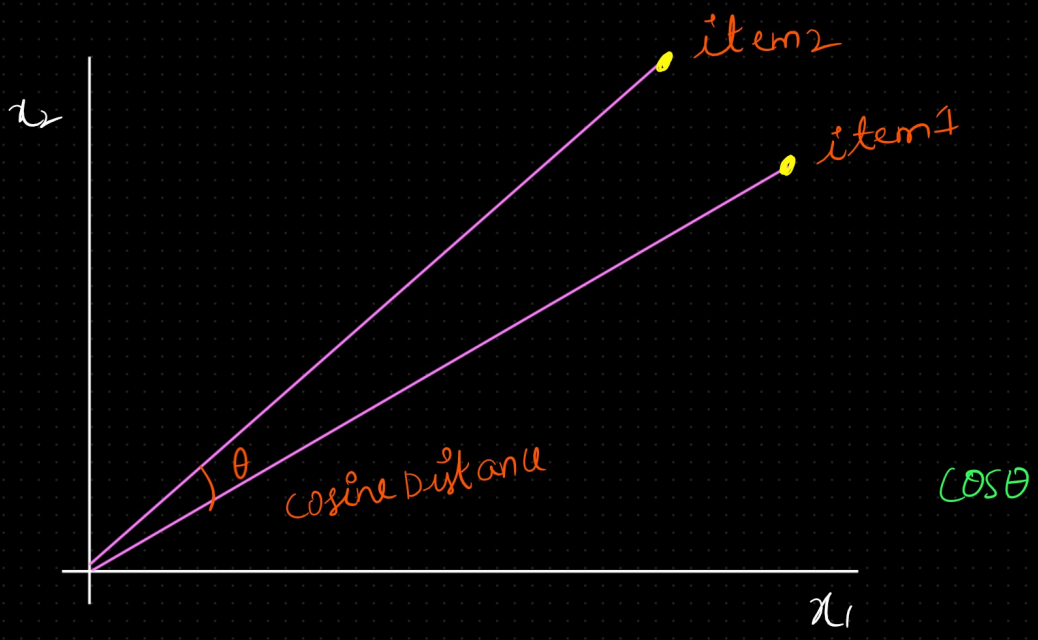
King : Man : Woman : Queen

King − man + woman = queen

Using Word2Vec, the words "king" and "queen" will have similar vectors because they share similar contexts.

[0.95] [0.96]

Cosine Distance / Similarity



Mathematically, it measures the cosine of the angle between two vectors (item1, item2) projected in an N-dimensional vector space. The advantageous of cosine similarity is, it predicts the document similarity even Euclidean is distance.

"Smaller the angle, the higher the similarity" — Cosine Similarity.

two training Algo / model

(i) Continuous Bag of words (CBOW)

(ii) Skip-gram

Continuous Bag of words (CBOW)

CBOW predicts the target word (center word) from a set of context words (surrounding words). It focuses on the context to guess the word in the middle. This method is generally faster and works well with larger datasets.

Eg → Let's learn Data Science with datasciencelearn channel.

Step 1: Window size → 5 → odd no.

Independent feature

Output / Target

Let's learn Scienu with

Data

Scienu

with

How it works:

Given a context (e.g., "The cat sat on the __"), CBOW tries to predict the missing word "mat" based on the surrounding words "The," "cat," "sat," and "on."
The model takes the average of the context word vectors and uses this to predict the center word.
Example:
For the sentence "The quick brown fox jumps over the lazy dog":

Context words: ["The", "quick", "brown", "fox", "jumps", "over", "the", "dog"]
Target word: "lazy"
CBOW will use the vectors of the context words to predict the word "lazy."

Skip gram

Skip-gram, on the other hand, predicts context words given a target word. It takes a single word as input and tries to predict the surrounding words within a certain window size. This method is better at capturing rare words and works well with smaller datasets.

How it works:

Given a word (e.g., "jumps"), Skip-gram tries to predict the context words ("The," "quick," "brown," "fox," "over," "the," "lazy," "dog") within a certain window size.
The model takes the center word and tries to predict the words that are within a defined window around it.
Example:
For the sentence "The quick brown fox jumps over the lazy dog":

Target word: "jumps"
Context words: ["The", "quick", "brown", "fox", "over", "the", "lazy", "dog"]
Skip-gram will use the vector of the target word "jumps" to predict the surrounding words. → context word

Comparison
CBOW:

Predicts the target word using context words.
Faster and more efficient on larger datasets.
Tends to smooth over less frequent words.
Skip-gram:

Predicts context words using the target word.
Better for capturing rare words.
Slower but more effective on smaller datasets.

Glove (Global Vectors For Word Representation)

GloVe captures global statistical information by factorizing the word co-occurrence matrix. The word vectors generated by GloVe will show that "apple" and "fruit" are related because they often appear together in similar contexts.

Fasttext

FastText, developed by Facebook, extends Word2Vec by considering subword information. For example, it can understand that "playing" and "played" are related because it breaks words into character n-grams and represents them in the vector space.

Named Entity Recognition (NER)
NER is a subtask of information extraction that seeks to locate and classify named entities in text into predefined categories such as person names, organizations, locations, dates, etc.

Example:
For the sentence "Barack Obama was born in Hawaii," an NER system would identify:

"Barack Obama" as a PERSON,
"Hawaii" as a LOCATION.