Word2Vec

MUSTAFA HAJIJ

Purpose

Distributional hypothesis

Distributional hypothesis: words that occur in similar contexts tend to have similar meanings

Distributional hypothesis

Distributional hypothesis: words that occur in similar contexts tend to have similar meanings

Example:

1.The cat chased the mouse."

2."The dog pursued the squirrel.“

In these sentences, the words "cat" and "dog" occur in similar contexts—they are both associated with chasing or pursuing small animals. This similarity in context suggests that "cat" and "dog" have similar meanings related to animals and hunting.

Words as vectors : initial attempt

We will construct a novel model for understanding word meanings by emphasizing similarity.

In this model, each word is represented as a vector, and words that have similar meanings are positioned close to each other in the vector space.

To achieve this, we utilize a word-word co-occurrence matrix. As an initial approach, we can simply employ context vectors to represent the meanings of words.

A word-word co-occurrence matrix

A word-word co-occurrence matrix is a matrix that captures the frequency of co-occurrence of words within a given context.

It provides a representation of how often words appear together in the same context.

A word-word co-occurrence matrix

A word-word co-occurrence matrix is a matrix that captures the frequency of co-occurrence of words within a given context.

It provides a representation of how often words appear together in the same context.

To compute a word-word co-occurrence matrix, you start with a corpus of text data.
Here's a minimal example to illustrate the computation:
Consider the following corpus:

"I like apples.“
"I like bananas.“
"I like oranges."

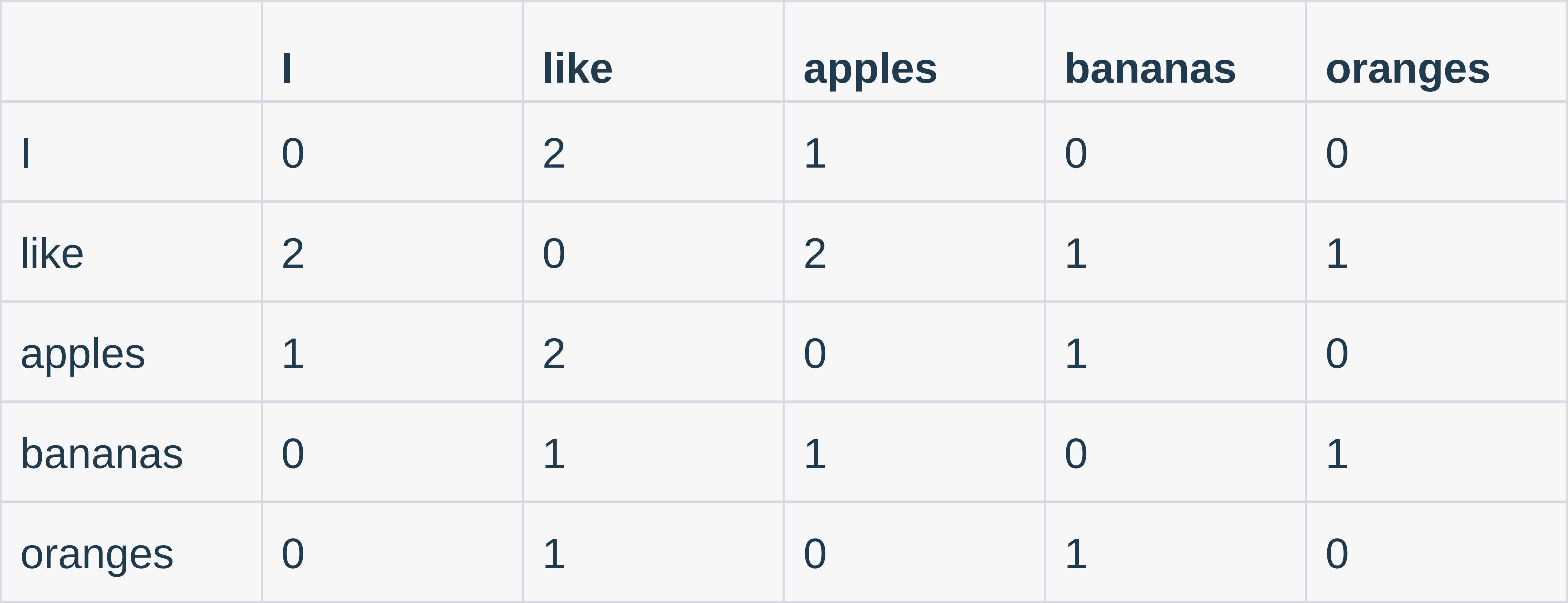
A word-word co-occurrence matrix

Create a vocabulary: First, we create a vocabulary by listing all unique words in the corpus:

Vocabulary: ["I", "like", "apples", "bananas", "oranges"]

A word-word co-occurrence matrix

Construct the co-occurrence matrix: Next, we create a matrix where each row and column represents a word from the vocabulary. The values in the matrix represent the frequency of co-occurrence of words in a given context. For simplicity, let's use a window size of 1, meaning we consider only the immediate neighboring words.



	I	**like**	**apples**	**bananas**	**oranges**
I	0	2	1	0	0
like	2	0	2	1	1
apples	1	2	0	1	0
bananas	0	1	1	0	1
oranges	0	1	0	1	0

Word2Vec

Input a large text corpora, V, d

- V: a pre-defined vocabulary
- d: dimension of word vectors (e.g. 300)
- Text corpora:

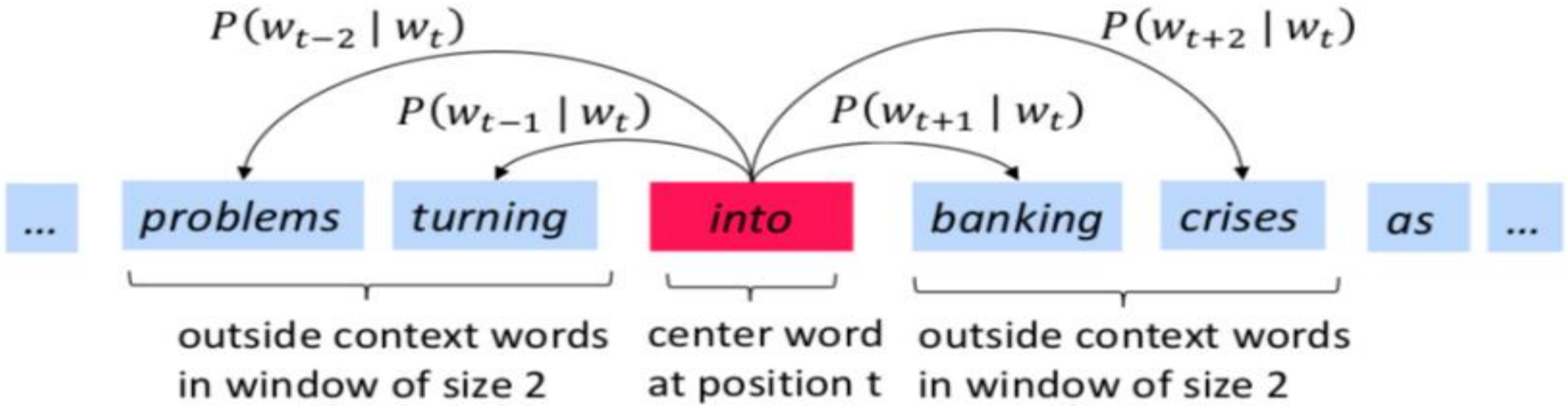
Output :

$$f : V \rightarrow \mathbb{R}^d$$

Skip-gram

The idea: we want to use words to predict their context words

- Here context: a fixed window of size 2 in every sentence



Skip-gram objective function

- For each position $t = 1,2,\ldots T$, predict context words within context size m, given center word w_j:

$$\mathcal{L}(\theta) = \prod_{t=1}^{T} \prod_{-m \leq j \leq m, j \neq 0} P(w_{t+j} \mid w_t; \theta)$$

Skip-gram objective function

How do we model $P(w_{t+j} \mid w_t; \theta)$?

Skip-gram objective function

In word2vec, we use two sets of vectors:

target word vectors (ui) and context word vectors (vi’).

These vectors help measure the likelihood of a target word appearing with a specific context word.

By taking their inner product, we can determine the strength of the relationship between the two words. Larger inner product values indicate a higher likelihood of co-occurrence. This approach allows us to capture semantic meaning and word associations effectively.

$\mathbf{u}_i \in \mathbb{R}^d$: embedding for target word *i*

$\mathbf{v}_{i'} \in \mathbb{R}^d$: embedding for context word *i’*

$$P(w_{t+j} \mid w_t) = \frac{\exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_{w_{t+j}})}{\sum_{k \in V} \exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_k)}$$