

Hide menu

Lecture: Word Embeddings

- ✓

Video: Week Introduction

1 min
- ✓

Video: Overview

2 min
- ✓

Reading: Overview
- ✓

Video: Basic Word Representations

3 min
- ✓

Reading: Basic Word Representations

5 min
- ▶

Video: Word Embeddings

3 min
- ✓

Reading: Word Embeddings

4 min
- ▶

Video: How to Create Word Embeddings

3 min
- ✓

Reading: How to Create Word Embeddings?

4 min
- ▶

Video: Word Embedding Methods

3 min
- ✓

Reading: Word Embedding Methods

4 min
- ▶

Video: Continuous Bag-of-Words Model

4 min
- ⌕

Reading: Continuous Bag of Words Model

3 min
- ▶

Video: Cleaning and Tokenization

4 min
- ⌕

Reading: Cleaning and Tokenization

5 min
- ▶

Video: Sliding Window of Words in Python

3 min
- ⌕

Reading: Sliding Window of words in Python

10 min
- ▶

Video: Transforming Words into Vectors

3 min
- ⌕

Reading: Transforming Words into Vectors

2 min
- ✓

Lab: Lecture Notebook - Data Preparation

30 min
- ▶

Video: Architecture of the CBOW Model

3 min
- ⌕

Reading: Architecture for the CBOW Model

4 min
- ▶

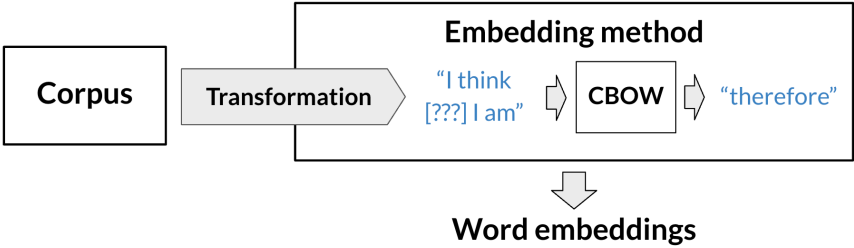
Video: Architecture of the CBOW

> Week 4 > Continuous Bag of Words Model

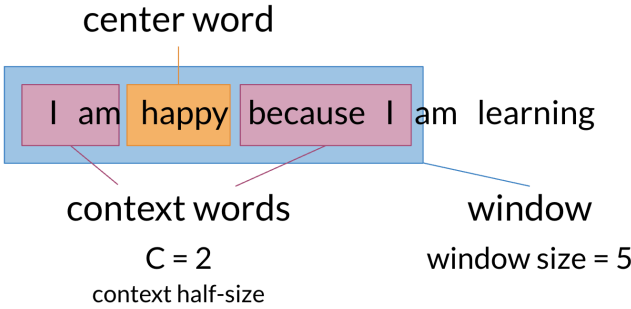
< Previous Next >

Continuous Bag of Words Model

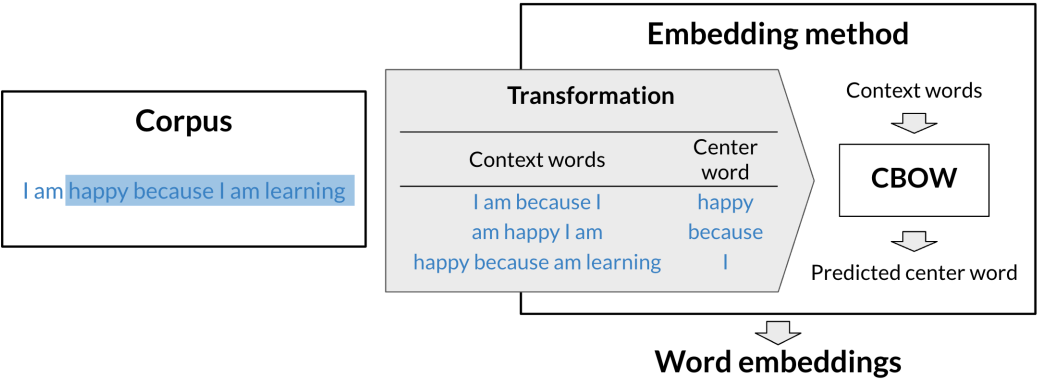
To create word embeddings, you need a corpus and a learning algorithm. The by-product of this task would be a set of word embeddings. In the case of the continuous bag-of-words model, the objective of the task is to predict a missing word based on the surrounding words.



Here is a visualization that shows you how the models works.



As you can see, the window size in the image above is 5. The context size, C, is 2. C usually tells you how many words before or after the center word the model will use to make the prediction. Here is another visualization that shows an overview of the model.



Mark as completed

Like Dislike Report an issue

