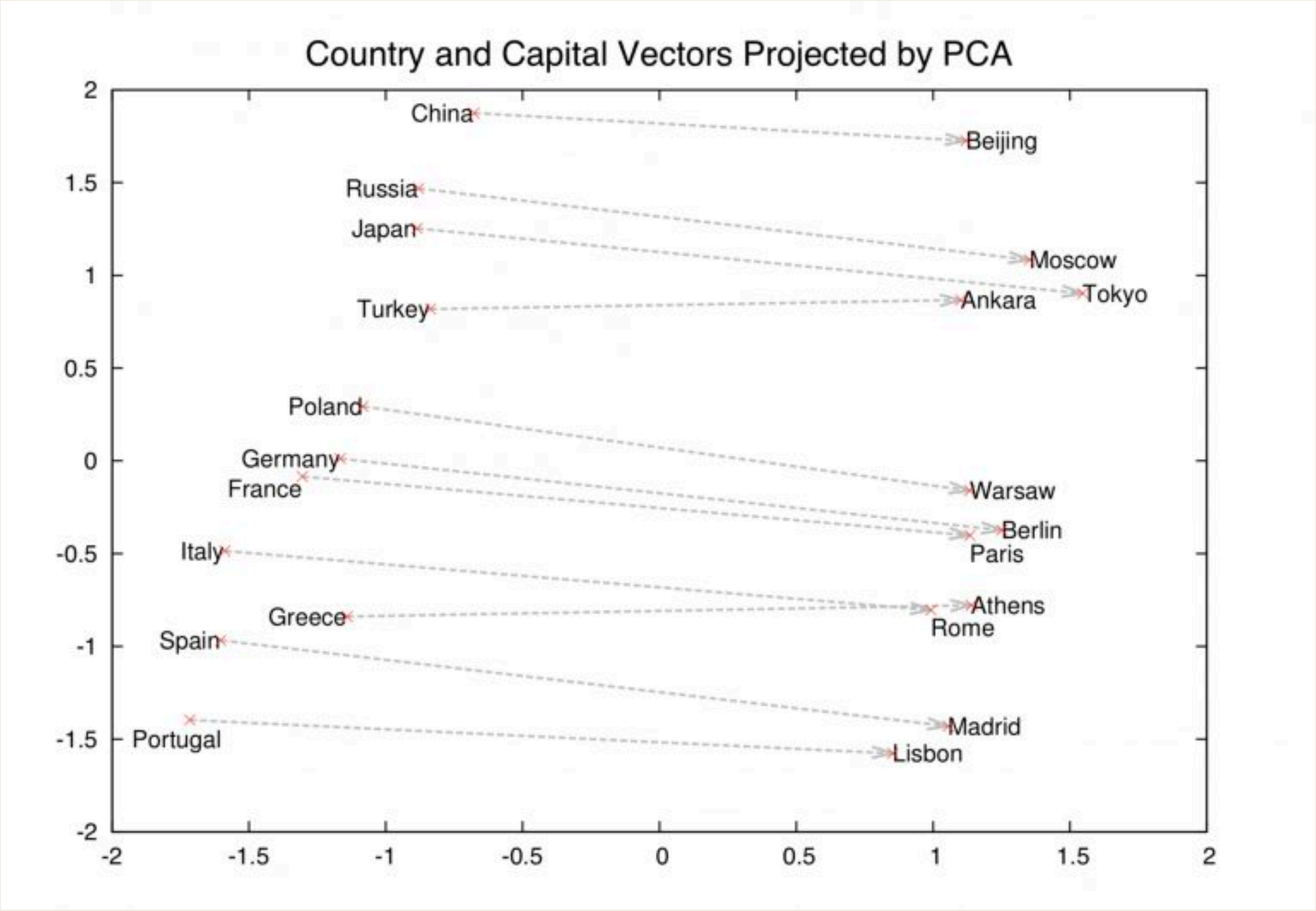
Word2Vec

A quick overview of Word2Vec's CBoW model

Word Embedding

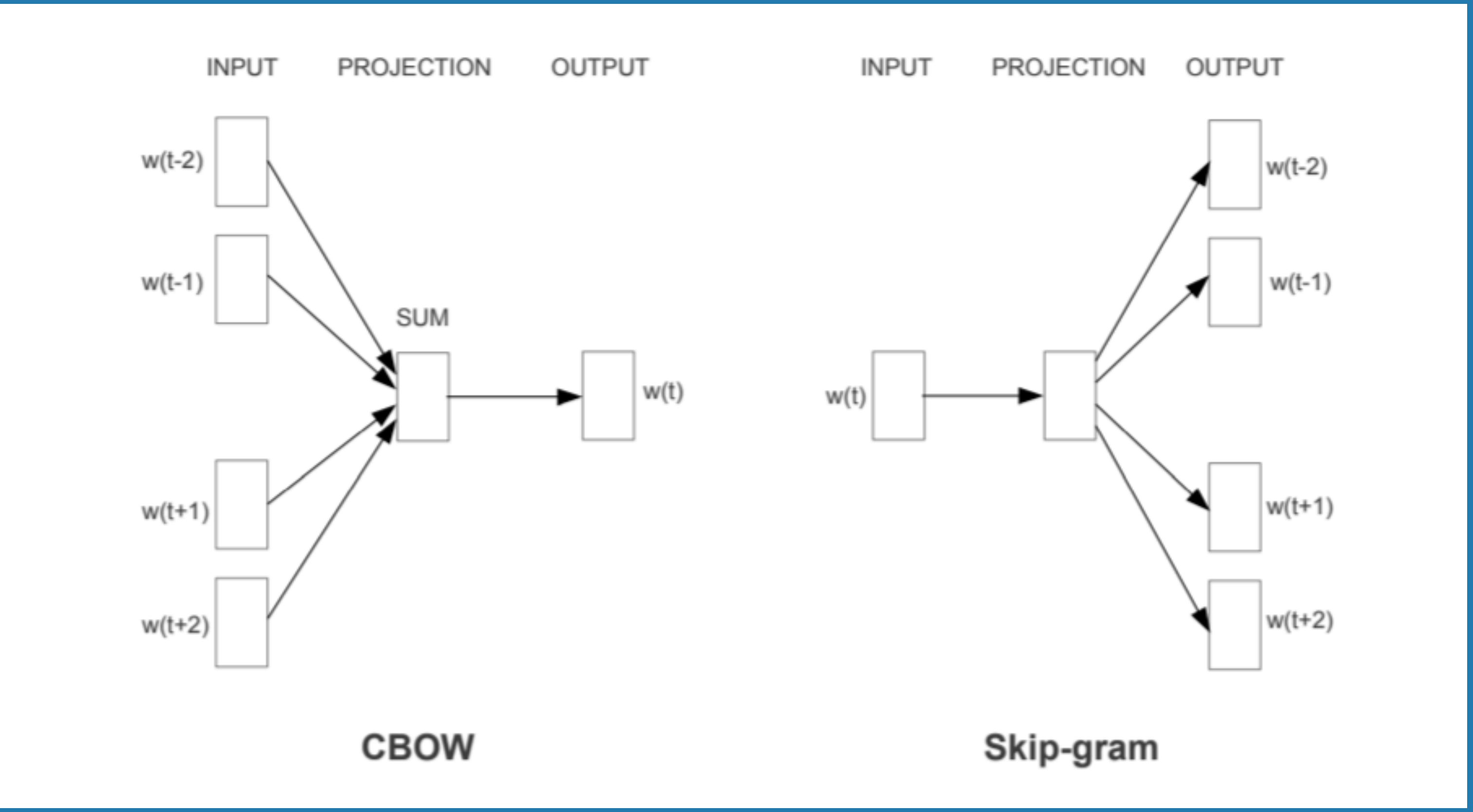


N-dimensional vectors that try to capture word-meaning and context in their values (Numerical representation of words)

Word2Vec

Training a neural network with a single hidden layer to predict a **target word** based on its **context** (**neighboring words**).

Assumption : The meaning of a word can be inferred by the company it keeps.

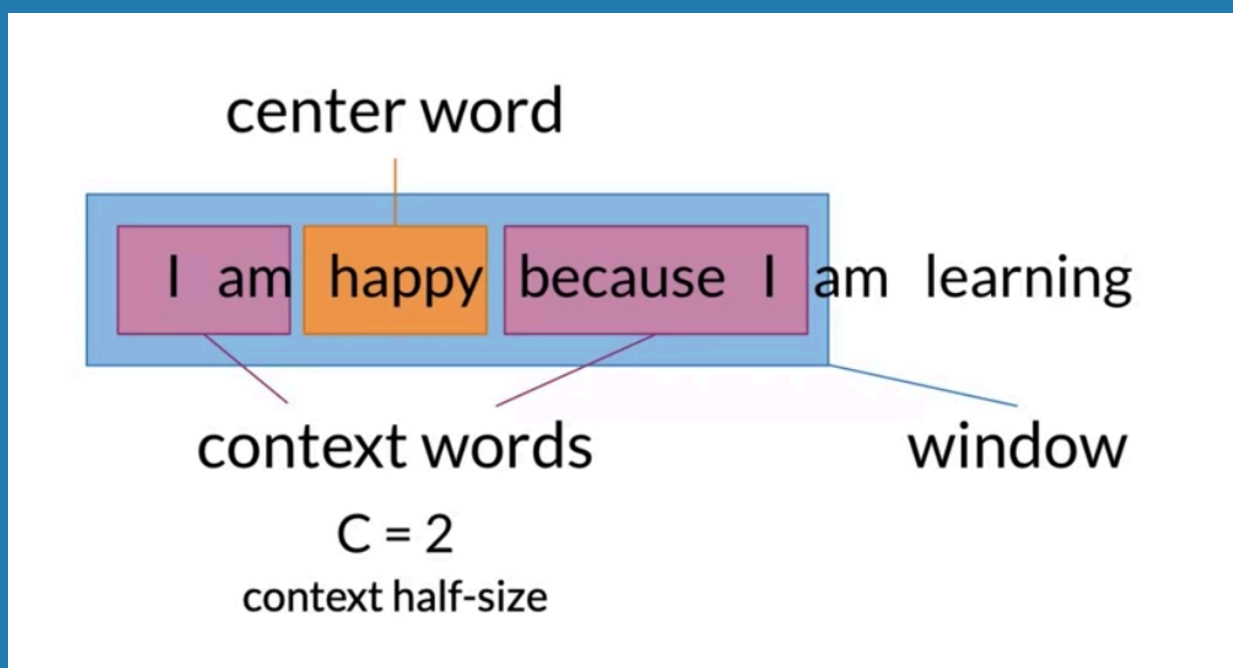


Source: Exploiting Similarities among Languages for Machine Translation paper.

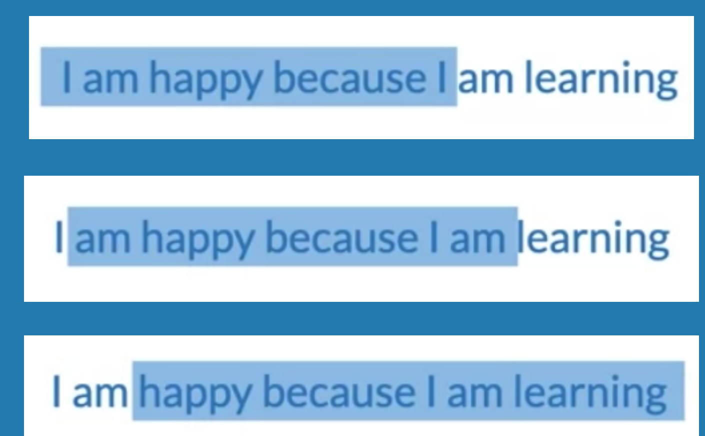
CBoW : Introduction

Continuous Bag of Words

The distributed representations of context are combined to predict the word in the middle.



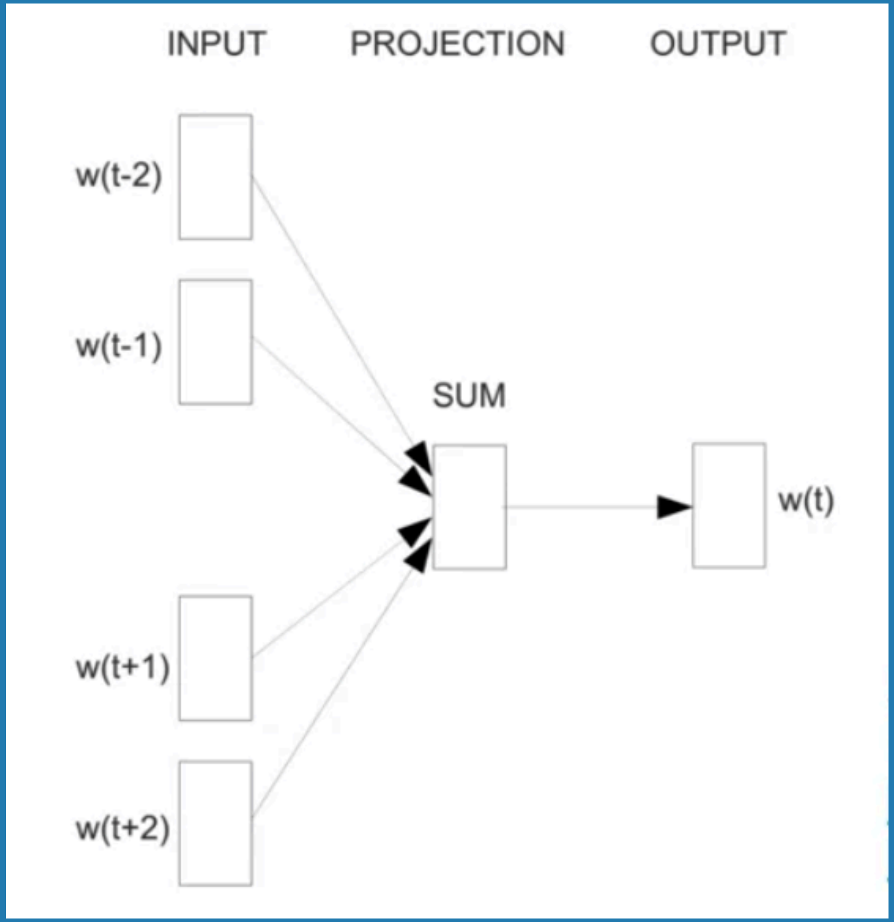
- Window size : 5
- Context half-size : 2



CBoW : Architecture

Model Input → **Context Words**

Model Output → **Center Word**



I am happy because I am learning

- Context : I, am, because, I
- Center : happy

I am happy because I am learning

- Context : am, happy, I, am
- Center : because

I am happy because I am learning

- Context : happy, because, am, learning
- Center : I

Natural Language Processing with Probabilistic Models - DeepLearning.AI (Coursera)

CBoW : Matrices

Model Input → **Context Word** (One Hot Encoded & Averaged)

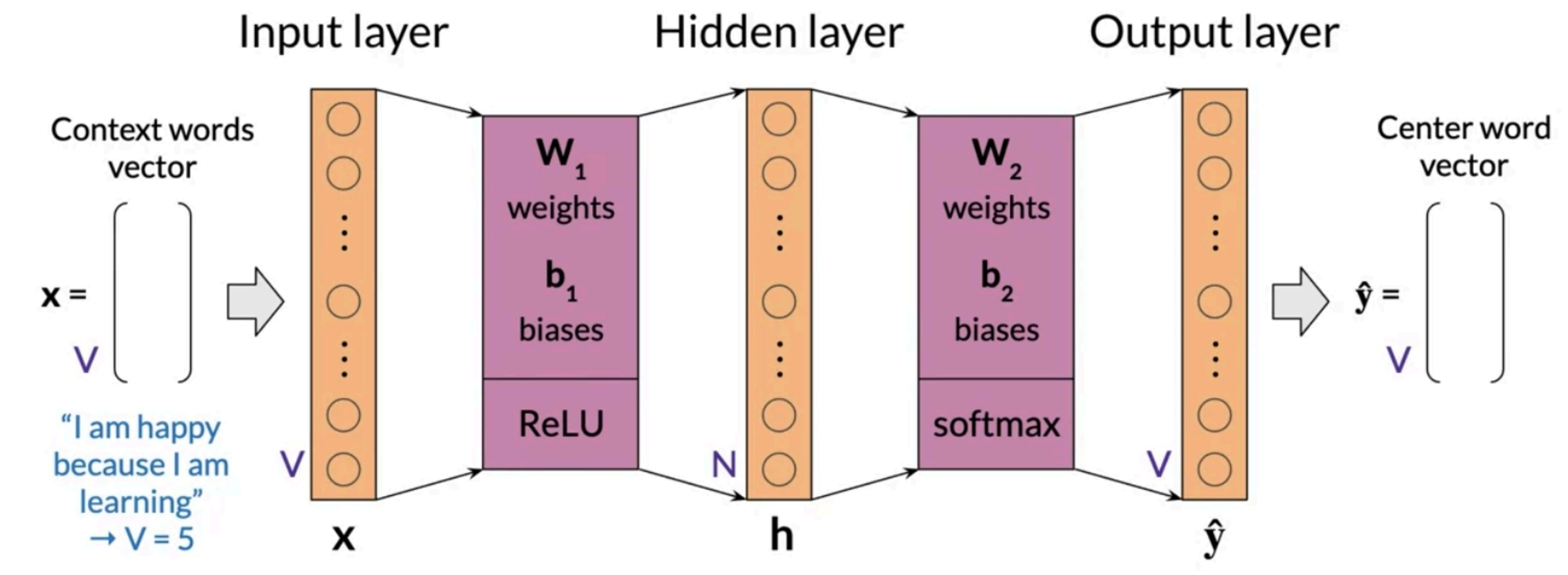
$$\left(\begin{array}{r} \\ \text{am} \\ \text{because} \\ \text{happy} \\ \text{I} \\ \text{learning} \end{array} \begin{array}{c} \text{I} \\ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \end{array} + \begin{array}{c} \text{am} \\ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array} + \begin{array}{c} \text{because} \\ \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array} + \begin{array}{c} \text{I} \\ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \end{array} \right) / 4 = \begin{array}{c} \text{I am because I} \\ \begin{pmatrix} 0.25 \\ 0.25 \\ 0 \\ 0.5 \\ 0 \end{pmatrix} \end{array}$$

Model Output → **Center Words** (One Hot Encoded)

$$\begin{array}{r} \\ \text{am} \\ \text{because} \\ \text{happy} \\ \text{I} \\ \text{learning} \end{array} \begin{array}{c} \text{am} \\ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array} \quad \begin{array}{c} \text{because} \\ \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array} \quad \begin{array}{c} \text{happy} \\ \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \end{array} \quad \begin{array}{c} \text{I} \\ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \end{array} \quad \begin{array}{c} \text{learning} \\ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{array}$$

CBoW : Model

Context words	Context words vector	*Center word*	Center word vector
I am because I	[0.25; 0.25; 0; 0.5; 0]	*happy*	[0; 0; 1; 0; 0]
am happy I am	[0.5; 0; 0.25; 0.25; 0]	*because*	[0; 1; 0; 0; 0]
happy because am learning	[0.25; 0.25; 0.25; 0; 0.25]	*I*	[0; 0; 0; 1; 0]



N : Length of the corpus, **V** : Length of the Hidden Layer

Word2Vec Summary

- Not a singular algorithm, rather a family of model architectures and optimizations that can be used to learn word embeddings from data.
- Idea is very intuitive, learning the representation of words in a classification tasks - requires little memory
- Since words and vectors have a one-to-one relationship, the problem of polysemous words or words in different tense/form cannot be solved
    - Teach, teacher, teachers are all treated as different words without any relationship
- Embedding works best for common words, however not for rare tokens or OOV (out-of-vocabulary)

References

1. **FastText vs Word2Vec** Kavita Ganesan (https://kavita-ganesan.com/fasttext-vs-word2vec/#.Yf0B0e5BxYw)
2. **NLP 101: Word2Vec — Skip-gram and CBOW** Ria Kulshrestha (Nov 25, 2019) Medium Article
3. **Natural Language Processing with Probabilistic Models** DeepLearning.AI (Coursera)
4. **Word2Vec** Tensorflow Tutorials (https://www.tensorflow.org/tutorials/text/word2vec)