



Session 4 (Lab)

# Effective Word Representation by python

Azam Rabiee, PhD

September 13, 2020

# Outline

## Session 1: Introduction

- Applications
- Tasks
- Approaches

## Session 2. Basics of Linguistics

- Components
- Challenges
- Vectorization

## Session 3. Basics of ML

- word2vec
- Components
- Architectures

## Session 4 (Lab). Effective Word Representation by python

# System Requirements

**python ( $\geq$  3.6)**

## Packages

jupyter

gensim

numpy

[nltk](#)

To install the packages, you may check The following page:

<https://github.com/AzamRabiee/Lab-material-for-Intro-to-AI-ML>

Q: Who did not work with jupyter notebook before?

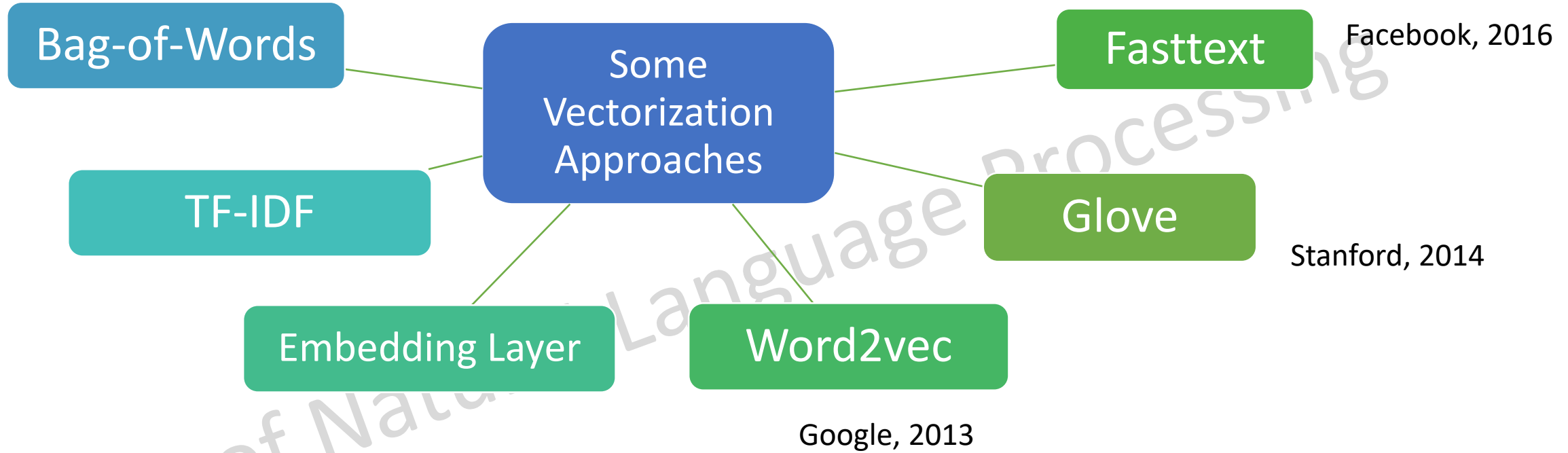
# Useful Python Library

<b>Package Name:</b>	NLTK
<b>Description:</b>	Natural Language Toolkit
<b>How to install:</b>	<code>pip install nltk</code>

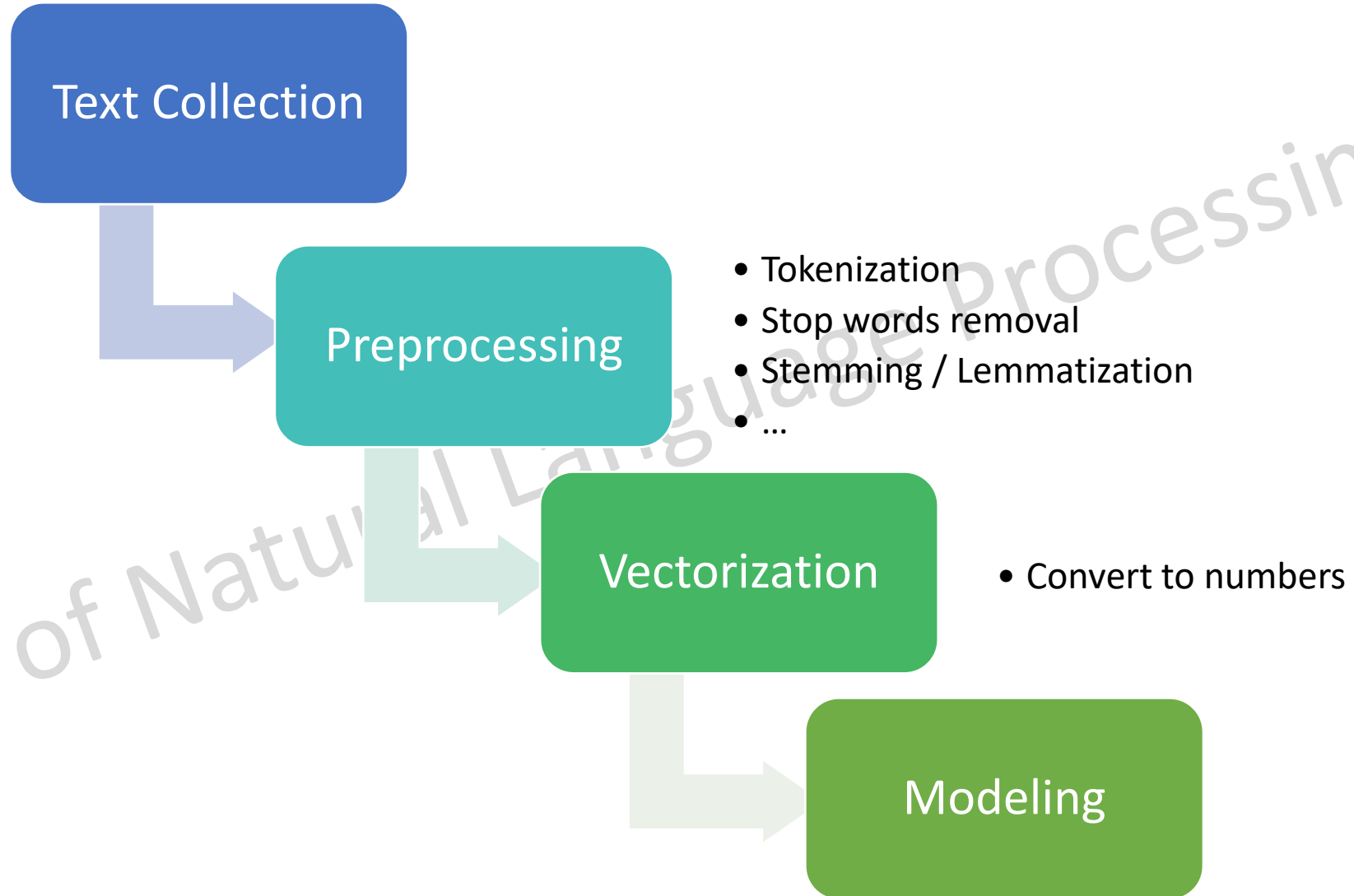
```
import nltk
```

Let's see how `nltk` works for stemming and lemmatization?

# Vectorization



# Steps of ML/DL Projects with Text Data



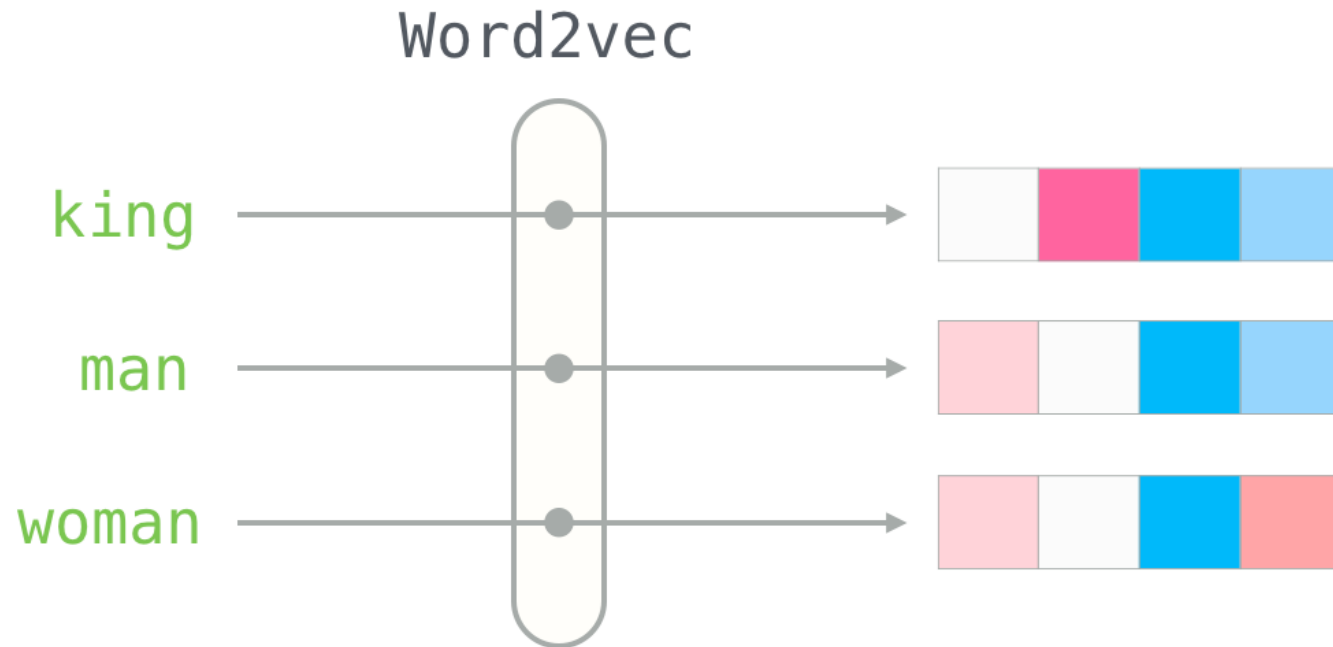
**Word2vec** is a model provided by Google in 2013  
for the **effective word embedding**.

Basics of Natural Language Processing

[Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013]

# word2vec

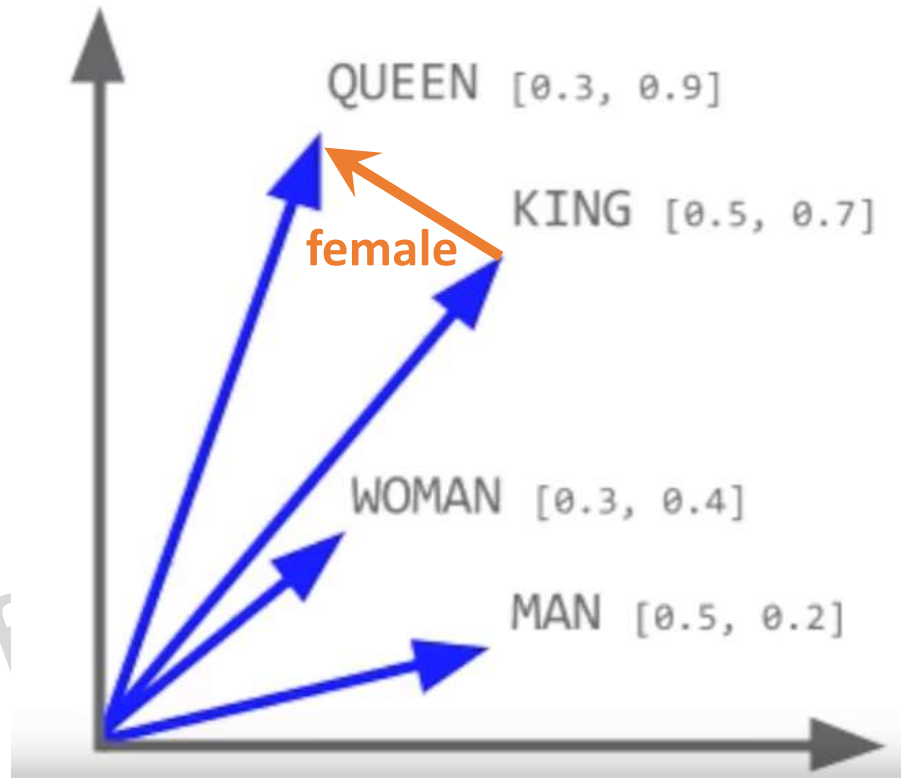
**Word2vec** is a model provided by Google in 2013  
for the **effective word embedding**.



[Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013]



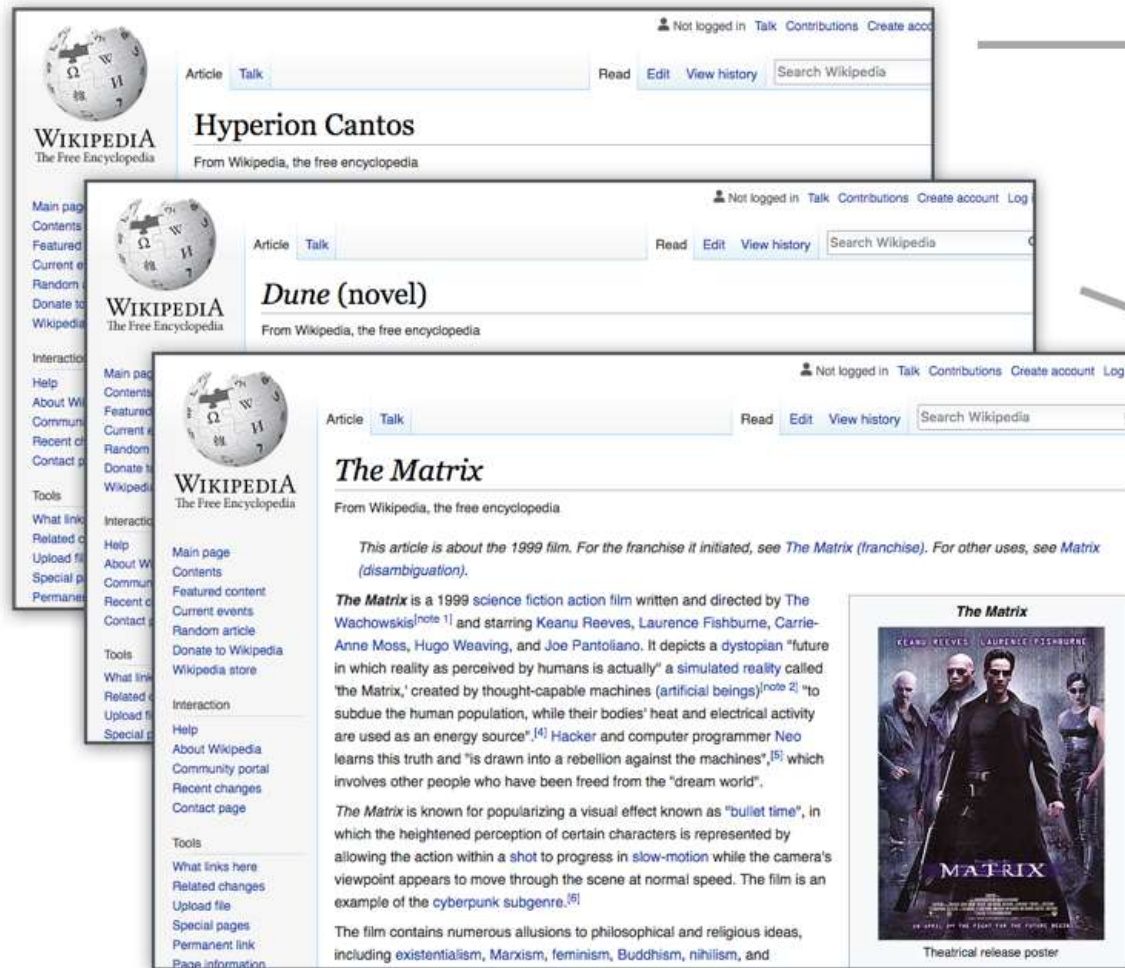
# Word Embedding



King + female = Queen  
Man + female = Woman  
Queen - royal = Woman

# Steps of word2vec





The **Hyperion Cantos** is a series of science fiction novels by Dan Simmons. The title is derived from the first novel in the series, *Hyperion* and *The Fall of Hyperion*,<sup>[13]</sup> and later came to refer to the overall storyline, including *Endymion*, *The Rise of Endymion*, and a number of short stories.<sup>[3][4]</sup> More narrowly, inside the fictional storyline, after the first volume, the Hyperion Cantos is an epic poem written by the character Martin Silenus covering in verse form the events of the first book.<sup>[5]</sup>

Of the four novels, *Hyperion* received the Hugo and Locus Awards in 1990;<sup>[6]</sup> *The Fall of Hyperion* won the Locus and British Science Fiction Association Awards in 1991;<sup>[7]</sup> and *The Rise of Endymion* received the Locus Award in 1998.<sup>[8]</sup> All four novels were also nominated for various science fiction awards.

An event series is being developed by Bradley Cooper, Graham King, and Todd Phillips for Syfy based on the first novel *Hyperion*.<sup>[9]</sup>

*Dune* is a 1965 science fiction novel by American author Frank Herbert, originally published as two separate serials in *Analog* magazine. It tied with Roger Zelazny's *This Immortal* for the Hugo Award in 1966,<sup>[3]</sup> and it won the inaugural Nebula Award for Best Novel.<sup>[4]</sup> It is the first installment of the *Dune* saga, and in 2003 was cited as the world's best-selling science fiction novel.<sup>[5][6]</sup>

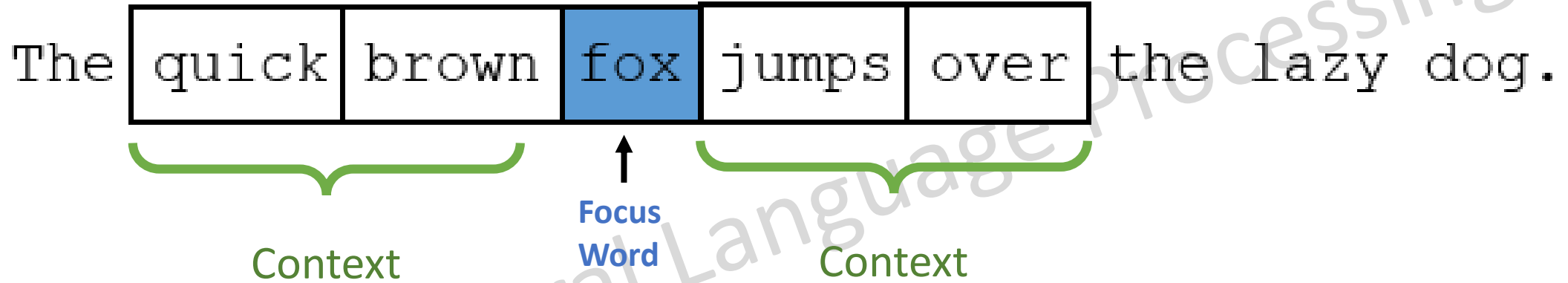
Set in the distant future amidst a feudal interstellar society in which noble houses, in control of individual planets, owe allegiance to the Padishah Emperor, *Dune* tells the story of young Paul Atreides, whose noble family accepts the stewardship of the planet Arrakis. It is an inhospitable and sparsely populated desert wasteland, but is also the only source of *melange*, also known as "spice", a drug that enhances mental abilities. As melange is the most important and valuable substance in the universe, control of Arrakis is a coveted—and dangerous—undertaking. The story explores the multi-layered interactions of politics, religion, ecology, technology, and human emotion, as the factions of the empire confront each other in a struggle for the control of Arrakis

*The Matrix* is a 1999 science fiction action film written and directed by The Wachowskis<sup>[note 1]</sup> and starring Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss, Hugo Weaving, and Joe Pantoliano. It depicts a dystopian "future in which reality as perceived by humans is actually" a simulated reality called 'the Matrix,' created by thought-capable machines (artificial beings)<sup>[note 2]</sup> "to subdue the human population, while their bodies' heat and electrical activity

[<http://jalammar.github.io/illustrated-word2vec/>]

# Sliding Window

Window size=5

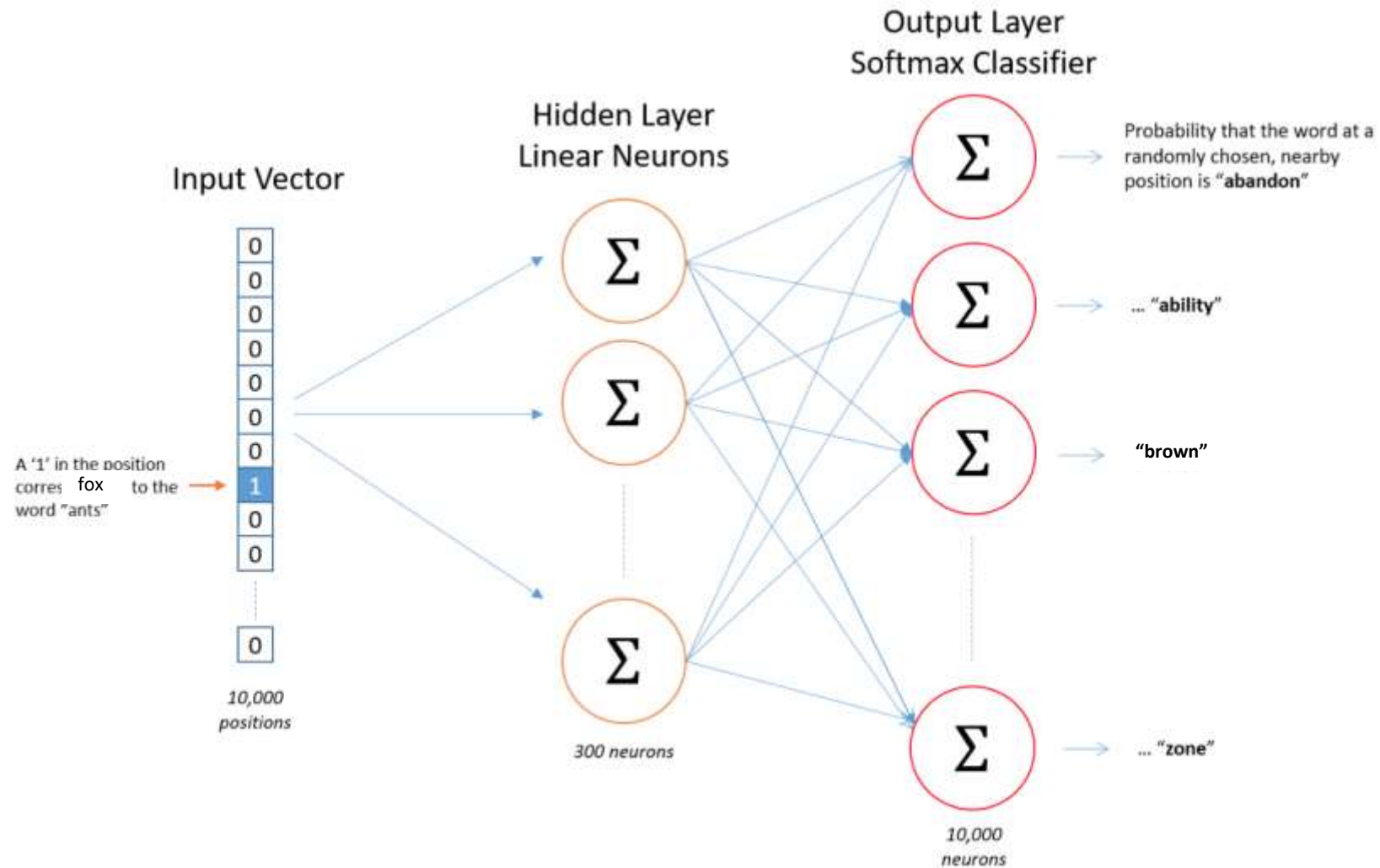


# Example Dataset

Source Text	Training Samples			
<table><tr><td>The</td><td>quick</td><td>brown</td></tr></table> fox jumps over the lazy dog. ➡	The	quick	brown	(the, quick) (the, brown)
The	quick	brown		
The <table><tr><td>quick</td><td>brown</td><td>fox</td></tr></table> jumps over the lazy dog. ➡	quick	brown	fox	(quick, the) (quick, brown) (quick, fox)
quick	brown	fox		
The quick <table><tr><td>brown</td><td>fox</td><td>jumps</td></tr></table> over the lazy dog. ➡	brown	fox	jumps	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
brown	fox	jumps		
The quick brown <table><tr><td>fox</td><td>jumps</td><td>over</td></tr></table> the lazy dog. ➡	fox	jumps	over	(fox, quick) (fox, brown) (fox, jumps) (fox, over)
fox	jumps	over		

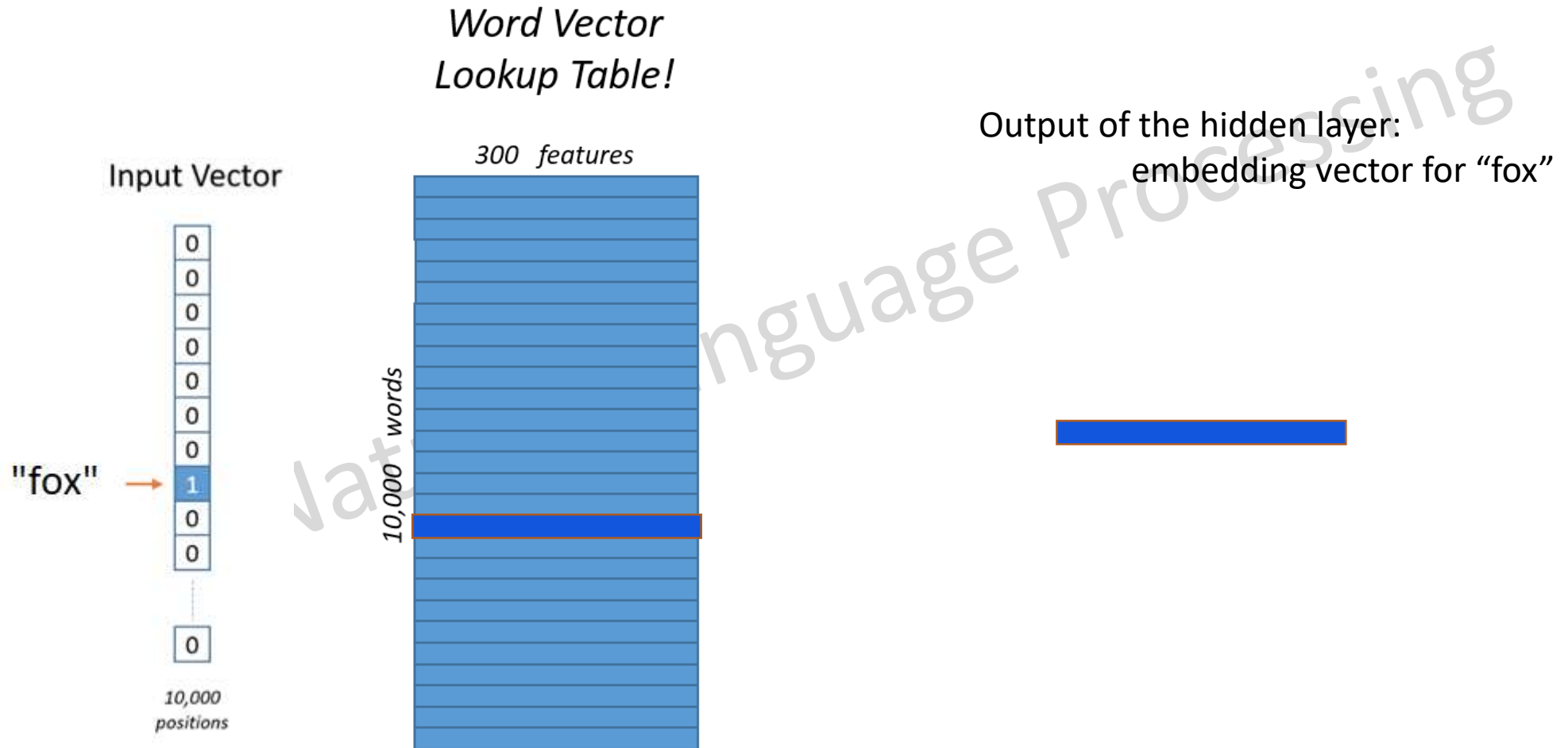
[McCormick, C.: Word2vec Tutorial - The Skip-Gram Model. <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>]

# Network Architecture

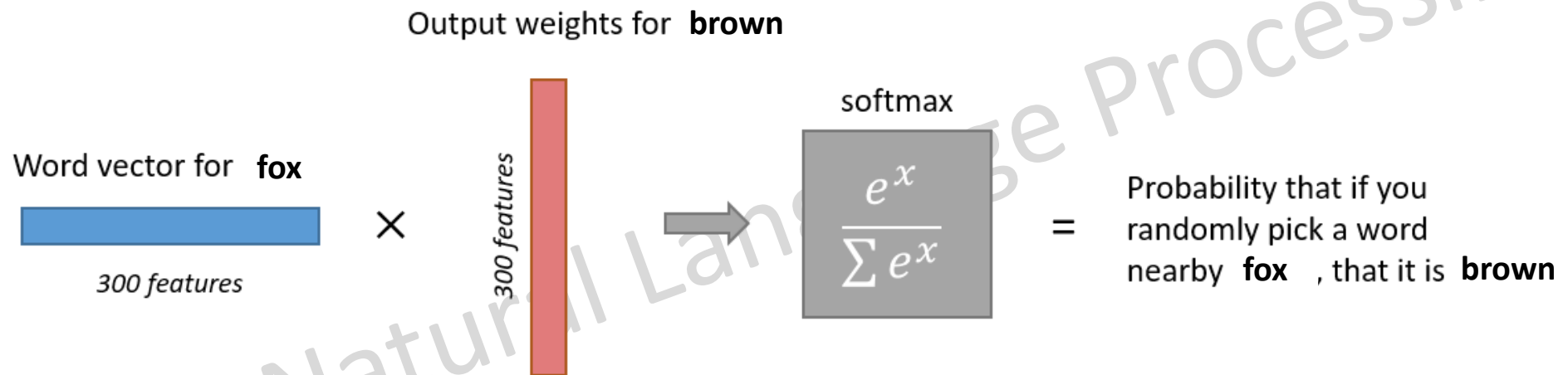


[McCormick, C.: Word2vec Tutorial - The Skip-Gram Model. <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>]

# Hidden Layer



# Output Layer





[http://www.nltk.org/nltk\\_data/](http://www.nltk.org/nltk_data/)

## NLTK Corpora

NLTK has built-in support for dozens of corpora and trained models, as listed below. To use these within NLTK we recommend that you use the NLTK corpus downloader, >>> `nltk.download()`

Please consult the README file included with each corpus for further information.

1. *perluniprops: Index of Unicode Version 7.0.0 character properties in Perl* [[download](#) | [source](#)]  
id: perluniprops; size: 100266; author: ; copyright: ; license: ;
2. *The monolingual word aligner (Sultan et al. 2015) subset of the Paraphrase Database.* [[download](#) | [source](#)]  
id: mwa\_ppdb; size: 1594711; author: ; copyright: ; license: Creative Commons Attribution 3.0 Unported (CC-BY);
3. *Punkt Tokenizer Models* [[download](#) | [source](#)]  
id: punkt; size: 13707633; author: Jan Strunk; copyright: ; license: ;
36. *Crubadan Corpus* [[download](#) | [source](#)]  
id: crubadan; size: 5288655; author: Kevin Scannell; copyright: Copyright (C) 2010 Kevin Scannell; license: GPLv3;
37. *Project Gutenberg Selections* [[download](#) | [source](#)]  
id: gutenberg; size: 4251829; author: ; copyright: public domain; license: public domain;
38. *Proposition Bank Corpus 1.0* [[download](#) | [source](#)]  
id: propbank; size: 5323498; author: ; copyright: ; license: Distributed with permission;
107. *Help on Tagsets* [[download](#) | [source](#)]  
id: tagsets; size: 34531; author: UCREL, Lancaster University; copyright: ; license: ;

[Natural Language Toolkit](#)

# Download Gutenberg Corpora

```
import nltk
from nltk.corpus import gutenberg

# download the 'gutenberg' corpora
# you can see the files at %APPDATA%/nltk_data
nltk.download('gutenberg')

# This tokenizer divides a text into words
nltk.download('punkt')

# import the corpus and convert into a list
sentences = list(gutenberg.sents('shakespeare-hamlet.txt'))
```

# Data Preprocessing

```
['[', 'The', 'Tragedie', 'of', 'Hamlet', 'by', 'William', 'Shakespeare', '1599', '']]
```

**Q: Any word preprocessing is needed?**

# Data Preprocessing

```
['[', 'The', 'Tragedie', 'of', 'Hamlet', 'by', 'William', 'Shakespeare', '1599', '']]
```



```
['the', 'tragedie', 'of', 'hamlet', 'by', 'william', 'shakespeare']
```

- Change to lowercase
- Remove numbers, punctuations and any thing rather than 'a' to 'z'

# Useful Python Library

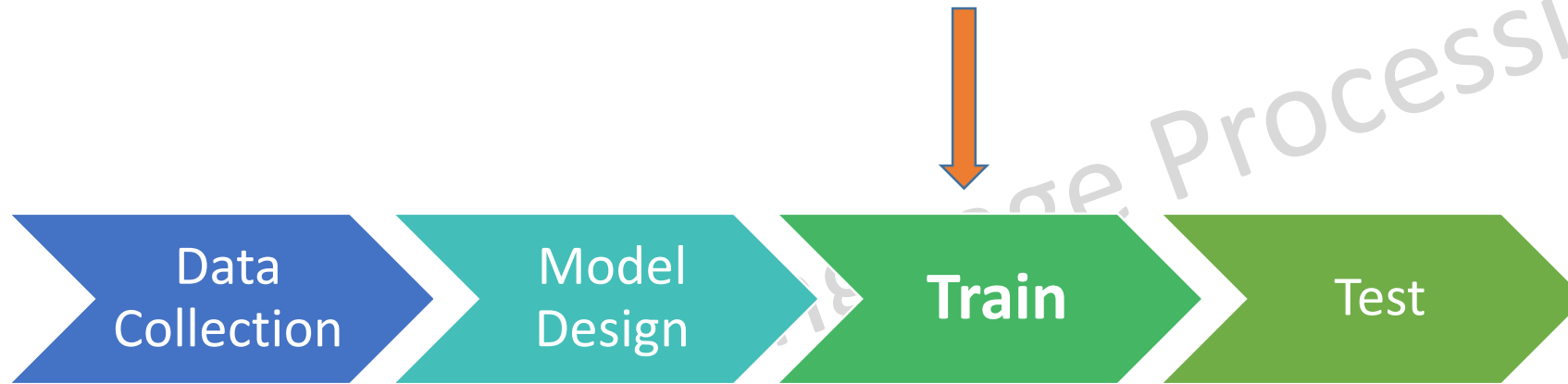
<b>Package Name:</b>	re
<b>Description:</b>	Regular expression
<b>How to install:</b>	It is already installed by python

```
import re

# The following code returns true if the
word contains only 'a-z' and 'A-Z'

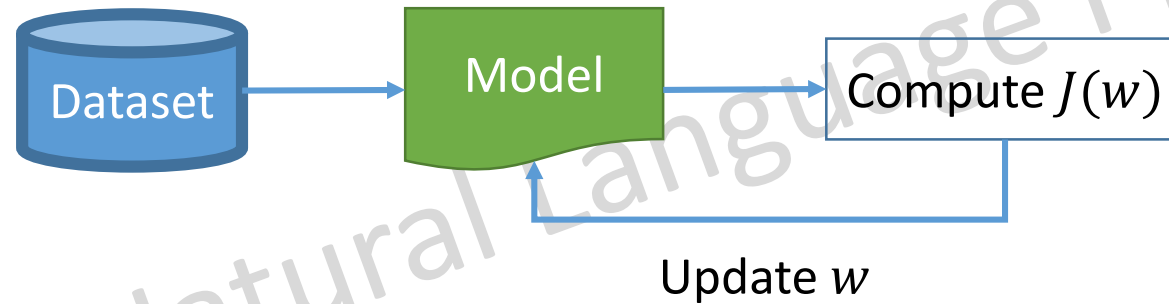
re.match('[a-zA-Z]+', word)
```

# Steps of word2vec



# Training Process

Finding optimum parameters that minimize a desired cost function  $J(w)$ .



# Useful Python Library

<b>Package Name:</b>	gensim
<b>Description:</b>	Generate Similar; useful package for text-mining and NLP
<b>How to install:</b>	pip install gensim

```
import gensim
```

**Note:** we are going to use the **Word2Vec** class of this package.



# Word2Vec Class

```
from gensim.models import word2vec
```

[\[https://radimrehurek.com/gensim/models/word2vec.html\]](https://radimrehurek.com/gensim/models/word2vec.html)

# Word2Vec Class

```
model = word2vec(sentences=sentences, size=20, window=3, iter=100)
```

The above line makes an instance of the Word2Vec class, named model, initiates it, and trains the network.

**sentences:** training data (has to be a list with tokenized sentences)

**size:** dimension of embedding space

**window:** number of words accounted for each context (if the window size is 3, 3 word in the left neighborhood and 3 word in the right neighborhood are considered)

**iter:** number of training iterations

[\[https://radimrehurek.com/gensim/models/word2vec.html\]](https://radimrehurek.com/gensim/models/word2vec.html)

Let's see every thing in Google Colab