


# Deep Learning for NLP

# Semantics


- How does a computer understand meaning?

- Dictionary
- Thesuras
- WordNet

Dictionary



## murder


*/ˈmɜːdə/* 

*noun*

1. the unlawful premeditated killing of one human being by another.  
"the brutal murder of a German holidaymaker"  
*synonyms:* [killing](#), [homicide](#), [assassination](#), [liquidation](#), [extermination](#), [execution](#), [slaughter](#), [butchery](#), [massacre](#); [More](#)
2. *informal*  
a very difficult or unpleasant task or experience.  
"the 40-mile-per-hour winds at the summit were murder"  
*synonyms:* [hell](#), hell on earth, a nightmare, an ordeal, a trial, a frustrating/unpleasant/difficult experience, [misery](#), [torture](#), [agony](#)  
"driving there was murder"

*verb*

1. kill (someone) unlawfully and with premeditation.  
"he was accused of murdering his wife's lover"  
*synonyms:* [kill](#), put/do to death, [assassinate](#), [execute](#), [liquidate](#), [eliminate](#), [neutralize](#), [dispatch](#), [butcher](#), cut to pieces, [slaughter](#), [massacre](#), wipe out, mow down; [More](#)
2. *informal*  
punish severely or be very angry with.  
"my father will murder me if I'm home late"

 Translations, word origin and more definitions

# Problems with this discrete representation

- Great as resource but missing nuances, e.g.  
**synonyms:**  
adept, expert, good, practiced, proficient, skillful?
- Missing new words (impossible to keep up to date):  
wicked, badass, nifty, crack, ace, wizard, genius, ninja
- Subjective
- Requires human labor to create and adapt
- Hard to compute accurate word similarity →

# Problems with this discrete representation

The vast majority of rule-based **and** statistical NLP work regards words as atomic symbols: *hotel*, *conference*, *walk*

In vector space terms, this is a vector with one 1 and a lot of zeroes

$[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$

Dimensionality: 20K (speech) – 50K (PTB) – 500K (big vocab) – 13M (Google 1T)

We call this a “one-hot” representation. Its problem:

*motel*  $[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$  AND  
*hotel*  $[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$  = 0

# Distributional similarity based representations

You can get a lot of value by representing a word by means of its neighbors

“You shall know a word by the company it keeps”

(J. R. Firth 1957: 11)

One of the most successful ideas of modern statistical NLP

government debt problems turning into banking crises as has happened in  
saying that Europe needs unified banking regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

# Window based cooccurrence matrix

- Example corpus:
  - I like deep learning.
  - I like NLP.
  - I enjoy flying.

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

# Problems with simple co-occurrence vectors

- Increase in size with vocabulary
- Very high dimensional: require a lot of storage
- Subsequent classification models have sparsity issues
  - Models are less robust

# Solution: Low dimensional vectors

- Idea: store “most” of the important information in a fixed, small number of dimensions: a dense vector
- Usually around 25 – 1000 dimensions
- How to reduce the dimensionality?



# Method 1: Dimensionality Reduction on X

Singular Value Decomposition of cooccurrence matrix  $X$ .

$$\begin{array}{ccccc}
 \begin{array}{c} m \\ \boxed{\phantom{X}} \\ n \\ X \end{array} & = & \begin{array}{c} r \\ \boxed{\begin{array}{c} | \phantom{U} | \phantom{U} | \phantom{U} | \phantom{\dots} | \\ U_1 U_2 U_3 \dots \\ | \phantom{U} | \phantom{U} | \phantom{U} | \phantom{\dots} | \end{array}} \\ n \\ U \end{array} & \begin{array}{c} r \\ \boxed{\begin{array}{c} S_1 S_2 S_3 \dots 0 \\ 0 \phantom{S_1} \phantom{S_2} \phantom{S_3} \dots S_r \end{array}} \\ r \\ S \end{array} & \begin{array}{c} m \\ \boxed{\begin{array}{c} \text{---} V_1 \text{---} \\ \text{---} V_2 \text{---} \\ \text{---} V_3 \text{---} \\ \vdots \\ \text{---} \end{array}} \\ r \\ V^T \end{array} \\
 \\
 \begin{array}{c} m \\ \boxed{\phantom{\hat{X}}} \\ n \\ \hat{X} \end{array} & = & \begin{array}{c} k \\ \boxed{\begin{array}{c} | \phantom{\hat{U}} | \phantom{\hat{U}} | \phantom{\hat{U}} | \phantom{\dots} | \\ U_1 U_2 U_3 \dots \\ | \phantom{\hat{U}} | \phantom{\hat{U}} | \phantom{\hat{U}} | \phantom{\dots} | \end{array}} \\ n \\ \hat{U} \end{array} & \begin{array}{c} k \\ \boxed{\begin{array}{c} S_1 S_2 S_3 \dots 0 \\ 0 \phantom{S_1} \phantom{S_2} \phantom{S_3} \dots S_k \end{array}} \\ k \\ \hat{S} \end{array} & \begin{array}{c} m \\ \boxed{\begin{array}{c} \text{---} V_1 \text{---} \\ \text{---} V_2 \text{---} \\ \text{---} V_3 \text{---} \\ \vdots \\ \text{---} \end{array}} \\ k \\ \hat{V}^T \end{array}
 \end{array}$$

$\hat{X}$  is the best rank  $k$  approximation to  $X$ , in terms of least squares.

# Simple SVD word vectors in Python

Corpus:

I like deep learning. I like NLP. I enjoy flying.

```
import numpy as np
la = np.linalg
words = ["I", "like", "enjoy",
         "deep", "learnig", "NLP", "flying", "."]
X = np.array([[0, 2, 1, 0, 0, 0, 0, 0],
              [2, 0, 0, 1, 0, 1, 0, 0],
              [1, 0, 0, 0, 0, 0, 1, 0],
              [0, 1, 0, 0, 1, 0, 0, 0],
              [0, 0, 0, 1, 0, 0, 0, 1],
              [0, 1, 0, 0, 0, 0, 0, 1],
              [0, 0, 1, 0, 0, 0, 0, 1],
              [0, 0, 0, 0, 1, 1, 1, 0]])

U, s, Vh = la.svd(X, full_matrices=False)
```

# Simple SVD word vectors in Python

Corpus: I like deep learning. I like NLP. I enjoy flying.

Printing first two columns of U corresponding to the 2 biggest singular values

