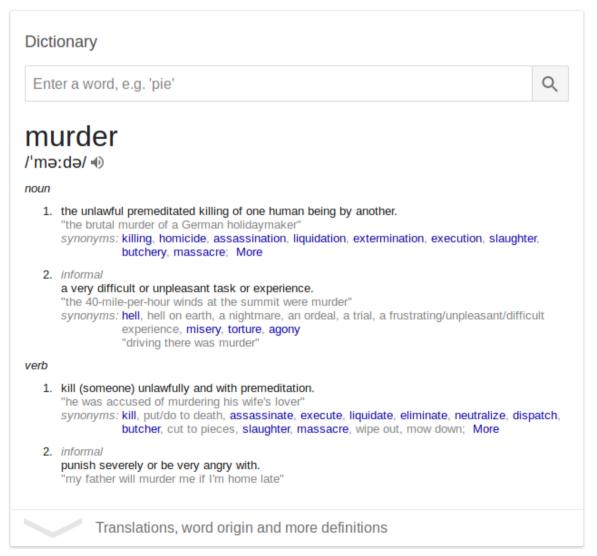
Deep Learning for NLP

Semantics

How does a computer understands meaning?

- Dictionary
- Thesuras
- WordNet



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, ninjia
- 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

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

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

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 7

Window based cooccurence matrix

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

counts	1	like	enjoy	deep	learning	NLP	flying	
1	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.

 \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

