Dan Jurafsky and James Martin

Speech and Language Processing

Chapter 6:
Vector Semantics

Let's define words by their usages

In particular, words are defined by their environments (the words around them)

Zellig Harris (1954): **If A and B have almost identical environments we say that they are synonyms**.

What does ong choi mean?

Suppose you see these sentences:

- Ongchoi is delicious **sautéed with garlic**.
- Ongchoi is superb **over rice**
- Ongchoi **leaves** with salty sauces

And you've also seen these:

- …spinach **sautéed with garlic over rice**
- Chard stems and **leaves** are **delicious**
- Collard greens and other **salty** leafy greens

Conclusion:

- Ongchoi is a leafy green like spinach, chard, or collard greens

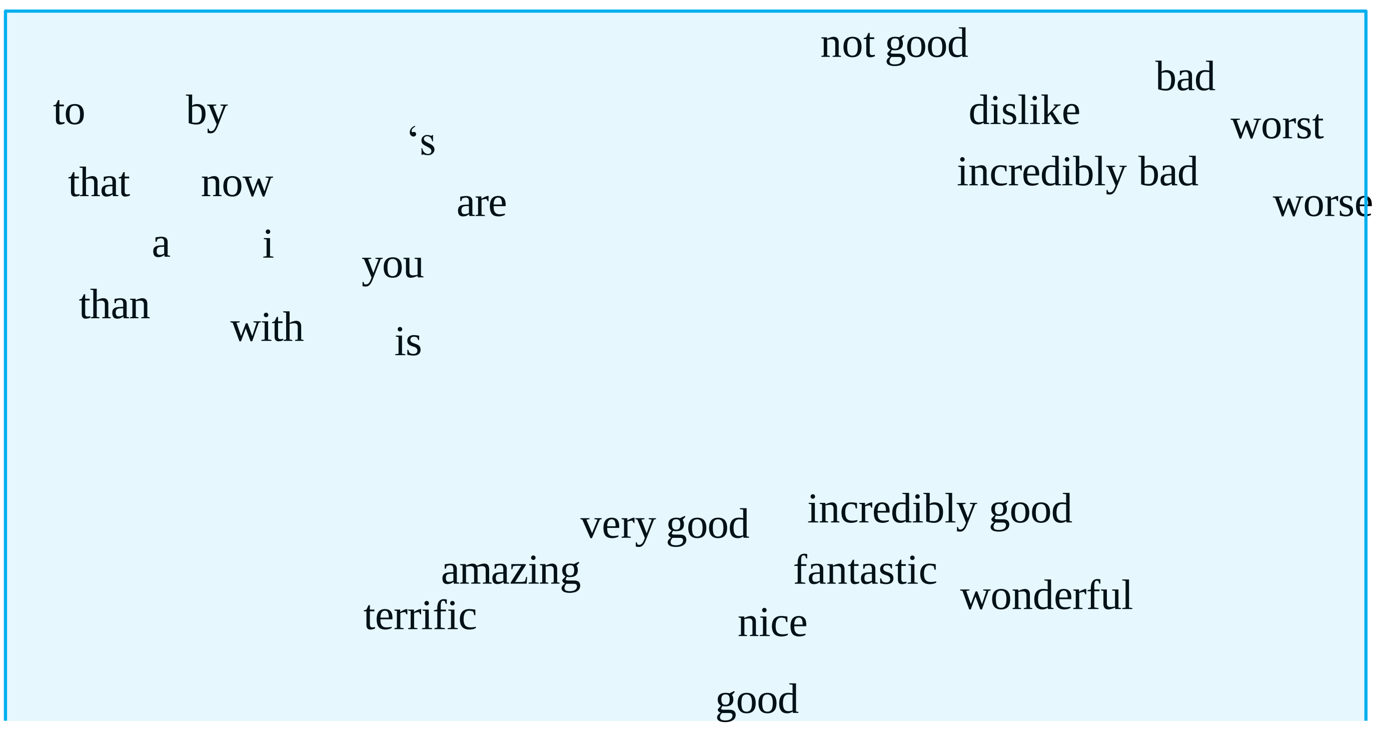
Ong choi: *Ipomoea aquatica* *"Water Spinach"*

Yamaguchi, Wikimedia Commons, public domain

Build a new model of meaning focusing on similarity

Each word = a vector

Similar words are "nearby in space"



Define a word as a vector

Called an "embedding" because it's embedded into a space

The standard way to represent meaning in NLP

Fine-grained model of meaning for similarity

- NLP tasks like sentiment analysis
  - With words, requires **same** word to be in training and test
  - With embeddings: ok if **similar** words occurred!!!
- Question answering, conversational agents, etc

2 kinds of embeddings

Tf-idf

- A common baseline model
- Sparse vectors
- Words are represented by a simple function of the counts of nearby words

Word2vec

- Dense vectors
- Representation is created by training a classifier to distinguish nearby and far-away words

An alternative to tf-idf

Ask whether a context word is **particularly informative** about the target word.

- Positive Pointwise Mutual Information (PPMI)

It compares the observed co-occurrence frequency of two words to their expected co-occurrence if they were statistically independent.

Pointwise Mutual Information

Pointwise mutual information:

Do events x and y co-occur more than if they were independent?

$$\mathrm{PMI}(X,Y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

PMI between two words: (Church & Hanks 1989)

Do words x and y co-occur more than if they were independent?

$$\mathrm{PMI}(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}$$

Positive Pointwise Mutual Information

- PMI ranges from $-\infty$ to $+\infty$
- But the negative values are problematic
  - Things are co-occurring **less than** we expect by chance
- So we just replace negative PMI values by 0
- Positive PMI (PPMI) between word1 and word2:

$$\mathrm{PPMI}(word_1, word_2) = \max\left(\log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}, 0\right)$$

Computing PPMI on a term-context matrix

Matrix F with W rows (words) and C columns (contexts)

f_{ij} is # of times w_i occurs in context c_j

	aardvark	computer	data	pinch	result	sugar
apricot	0	0	0	1	0	1
pineapple	0	0	0	1	0	1
digital	0	2	1	0	1	0
information	0	1	6	0	4	0

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \qquad p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \qquad p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}} \qquad ppmi_{ij} = \begin{cases} pmi_{ij} & \text{if } pmi_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

Count(w,context)

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2/19
pineapple	0	0	1	0	1	
digital	2	1	0	1	0	
information	1	6	0	4	0	

p(apricot) =

a [b c d e f] g h   Total count = 19

Co-occurrence count

d — b

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

Count(w,context)

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	= 2
pineapple	0	0	1	0	1	= 2
digital	2	1	0	1	0	= 4
information	1	6	0	4	0	= 11
		7/19				19

$$p(w_i) = \frac{\sum_{j=1}^{C} f_{ij}}{N} \qquad p(c_j) = \frac{\sum_{i=1}^{W} f_{ij}}{N}$$

p(w=information,c=data) = 6/19 = .32

p(w=information) = 11/19 = .58

p(c=data) = 7/19 = .37

p(w,context)

	computer	data	pinch	result	sugar	**p(w)**	
apricot	0.00	0.00	0.05	0.00	0.05	0.11	2/19
pineapple	0.00	0.00	0.05	0.00	0.05	0.11	2/19
digital	0.11	0.05	0.00	0.05	0.00	0.21	4/19
information	0.05	0.32	0.00	0.21	0.00	0.58	11/19
p(context)	0.16	0.37	0.11	0.26	0.11		
	3/19	7/19	2/19	5/19	2/19		

4 A 00 B 10 C 01 D 11 16 $\log_2 16 = 4$

$\log_2 4 = 2$

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}}$$

p(w,context)

	computer	data	pinch	result	sugar	**p(w)**
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

pmi(information,data) = $\log_2$ (.32 / (.37*.58)) = .58

(.57 using full precision)

Positive

PPMI(w,context)

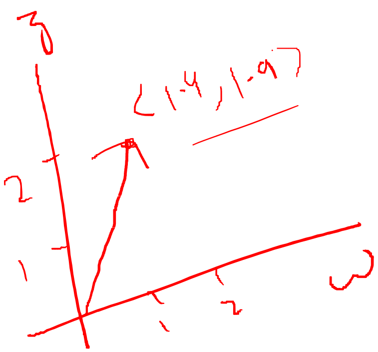
	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

	x	w	y	z	m
a	3	0	4	0	0
b	1	0	5	0	1
c	0	2	1	3	0
d	5	2	0	0	1
e	2	0	1	1	0

Represent the word "c" as vector using PPMI weights.

Total = 32

	x	w	y	z	m			
a	3	0	4	0	0	7	7/32	= 0.22
b	1	0	5	0	1	7	7/32	= 0.22
c	0	2 2/32 = 0.06	1/32 = 0.03 1	0.09 = 3	0	6	6/32	= 0.19
d	5/32 = 0.156 5	2	0	0	1	8	8/32	= 0.25
e	2	0	1	1	0	4	4/32	= 0.125
	11	4	11	4	2			
	0.34 = 11/32	4/32 (0.125)	11/32 (0.34)	4/32 (0.125)	2/32 = 0.06			



Represent the word "c" as vector using PPMI weights.

vector of "c"

x	w	y	z	m
0	$\log_2 \frac{0.06}{(0.19)(0.125)}$	$\log_2 \frac{0.03}{(0.19)(0.34)}$	$\log_2 \frac{0.09}{(0.19)(0.125)}$	0
0	$= \log_2 (2.6)$	$= \log_2 (0.5)$	$= \log_2 3.9$	0
	$= 1.4$	0	1.9	

Weighting PMI

PMI is biased toward infrequent events

- Very rare words have very high PMI values

Two solutions:

- Give rare words slightly higher probabilities

For rare events, even a few co-occurrences can produce a high PMI because the model assumes that their co-occurrence is "surprising" or informative. This is particularly problematic because:

- **Rare words have low probabilities**, so their independent probabilities $P(w)$ and $P(c)$ are small.
- The ratio $\frac{P(w,c)}{P(w)P(c)}$ becomes large, and the logarithm amplifies this effect, leading to a high PMI value.

Example of bias:

Consider a very rare word w that appears in only a few contexts. Even if it co-occurs with a particular context c just a few times, the fact that both $P(w)$ and $P(c)$ are extremely small means that the co-occurrence $P(w,c)$ appears much larger in comparison to their independent occurrences.

This causes PMI to assign a disproportionately high value to rare events, which does not necessarily reflect meaningful associations but rather the low overall probability of occurrence.

If **one of the words** (either the target word w or the context word c has a **higher probability** while the other is rare, the PMI value will still be affected.

Weighting PMI: Giving rare context words slightly higher probability

Raise the context probabilities to $\alpha = 0.75$:

$$\text{PPMI}_\alpha(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P_\alpha(c)}, 0)$$

$$P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha}$$

This helps because $P_\alpha(c) > P(c)$ for rare *c*

Consider two events, P(a) = .99 and P(b)=.01

$$P_\alpha(a) = \frac{99^{.75}}{99^{.75}+1^{.75}} = .97 \quad P_\alpha(b) = \frac{1^{.75}}{99^{.75}+1^{.75}} = .03$$

Summary for Part I

- Idea of Embeddings: Represent a word as a function of its distribution with other words
- Tf-idf
- Cosines
- PPMI

Calculate the PPMI vector of the words: *data , science*
Consider context window of size 2.

Data science is an interdisciplinary field.
Machine learning is a subset of Data science.
Data science involves statistics and programming.
Programming is essential for Data science and Artificial intelligence.
Statistics is important for Data science and Machine learning.