

Word Embeddings (I)

CS 485, Fall 2026

Applications of Natural Language Processing

Brendan O'Connor

College of Information and Computer Sciences

University of Massachusetts Amherst

[With slides from SLP3, Mohit Iyyer, and Laure Thompson]

- Today: discuss the problem of representing word meanings, and introduce **word embeddings** as a computational representation and model to capture them
- Next time: look at word embedding results and discuss how their utility

- What do words mean in the models we've seen so far
 - Words are just strings, or vocab indexes from 1 to $|V|$
- Logic: Word meanings are first-order predicates
 - $\forall x \text{ DOG}(x) \rightarrow \text{MAMMAL}(x)$
- Seems inadequate...

logical semantics (joke)



**Barbara H.
Partee**

Distinguished University Professor Emerita of Linguistics
[Department of Linguistics](#) , [University of Massachusetts](#)
[Amherst](#), MA 01003-1100 USA

*Gregory Carlson (1977)'s story
in his UMass dissertation*

to ask one of them. What is life? What is the meaning of life?
After all, Barbara and Terry had promised to provide answers to
any question at all.

On the final day of class Barbara wore her Montague grammar
T-shirt, and she and Terry busied themselves answering our questions.
At long last, they came to my question. I anticipated a protracted
and involved answer, but their reply was crisp and succinct. First
Barbara, chalk in hand, showed me the meaning of life.

^life'

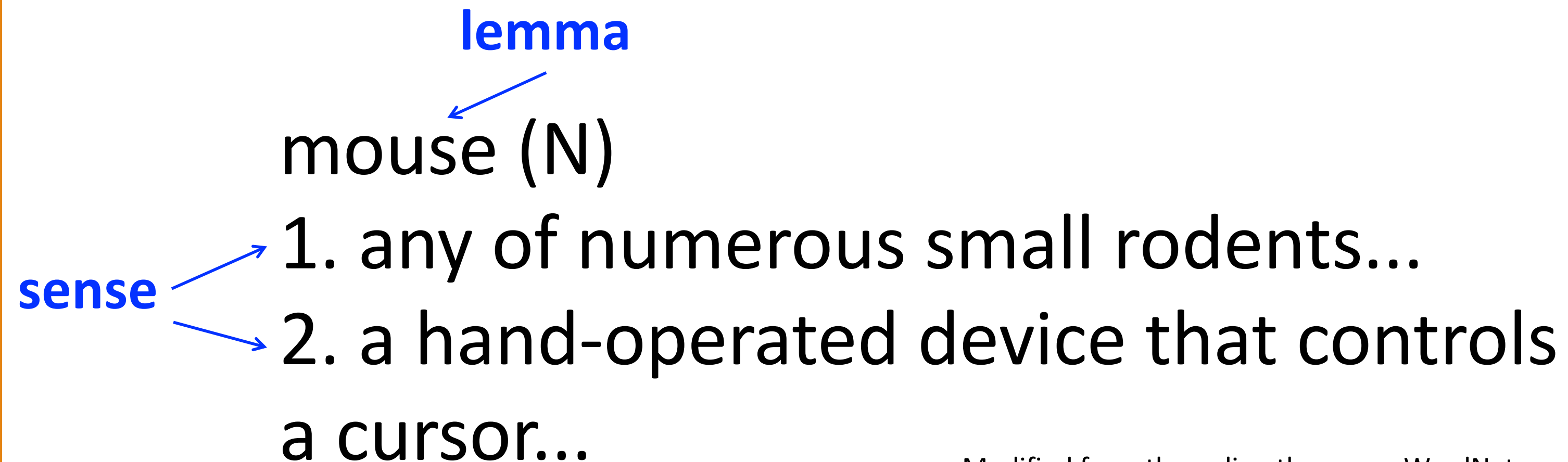
Lexical semantics

- **Lexical semantics:** Study of word meaning, especially of *content words*



- Local company doing *lexicography*: studying and documenting word meanings (and other word information)

Lemmas and senses



Modified from the online thesaurus WordNet

A **sense** or “**concept**” is the meaning component of a word
Lemmas can be **polysemous** (have multiple senses)

Relations between senses: Synonymy

Synonyms have the same meaning in some or all contexts.

- filbert / hazelnut
- couch / sofa
- big / large
- automobile / car
- vomit / throw up
- water / H₂O



Relations between senses: Synonymy

Note that there are probably no examples of perfect synonymy.

- Even if many aspects of meaning are identical
- Still may differ based on politeness, slang, register, genre, etc.



Relation: Synonymy?

water/H₂O

"H₂O" in a surfing guide?

big/large

my big sister != my large sister



Relation: Similarity

Words with similar meanings. Not synonyms, but sharing some element of meaning

car, bicycle

cow, horse

Ask humans how similar 2 words are

word1	word2	similarity
vanish	disappear	9.8
behave	obey	7.3
belief	impression	5.95
muscle	bone	3.65
modest	flexible	0.98
hole	agreement	0.3

SimLex-999 dataset (Hill et al., 2015)

Semantic field

Words that

- cover a particular semantic domain
- bear structured relations with each other.

hospitals

surgeon, scalpel, nurse, anaesthetic, hospital

restaurants

waiter, menu, plate, food, menu, chef

houses

door, roof, kitchen, family, bed

- Are related words in the same domain always similar to one another?

Relation: Word relatedness

Also called "word association"

Words can be related in any way, perhaps via a semantic frame or field

- coffee, tea: **similar**
- coffee, cup: **related**, not similar

Relation: Antonymy

Senses that are opposites with respect to only one feature of meaning

Otherwise, they are very similar!

dark/light	short/long	fast/slow	rise/fall
hot/cold	up/down	in/out	

More formally: antonyms can

- define a binary opposition or be at opposite ends of a scale
 - long/short, fast/slow
- Be *reversives*:
 - rise/fall, up/down

Connotation (sentiment)

- Words have **affective** meanings
 - Positive connotations (*happy*)
 - Negative connotations (*sad*)
- Connotations can be subtle:
 - Positive connotation: *copy, replica, reproduction*
 - Negative connotation: *fake, knockoff, forgery*
- Evaluation (sentiment!)
 - Positive evaluation (*great, love*)
 - Negative evaluation (*terrible, hate*)

Connotation

Osgood et al. (1957)

Words seem to vary along 3 affective dimensions:

- **valence**: the pleasantness of the stimulus
- **arousal**: the intensity of emotion provoked by the stimulus
- **dominance**: the degree of control exerted by the stimulus

	Word	Score		Word	Score
Valence	love	1.000		toxic	0.008
	happy	1.000		nightmare	0.005
Arousal	elated	0.960		mellow	0.069
	frenzy	0.965		napping	0.046
Dominance	powerful	0.991		weak	0.045
	leadership	0.983		empty	0.081

Values from NRC VAD Lexicon (Mohammad 2018)



So far

Concepts or word senses

- Have a complex many-to-many association with **words** (homonymy, multiple senses)

Have relations with each other

- Synonymy
- Antonymy
- Similarity
- Relatedness
- Connotation

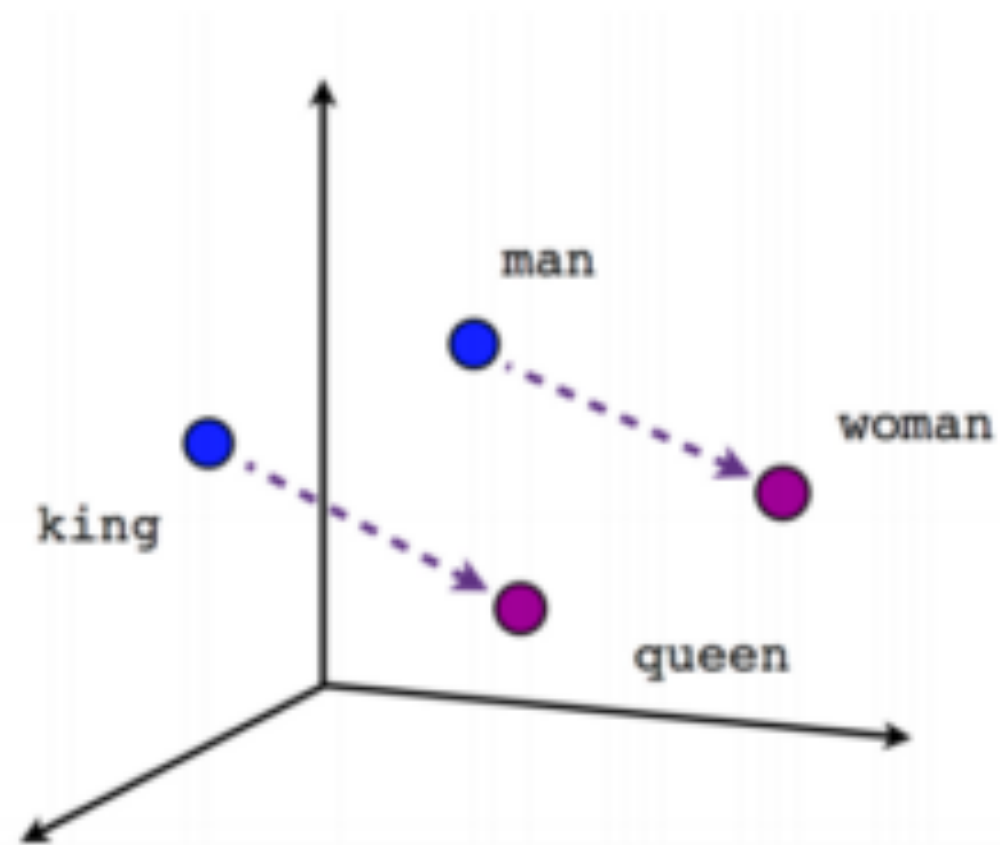
Word embeddings

in NLP, the main
way to represent
word types is with
vectors!

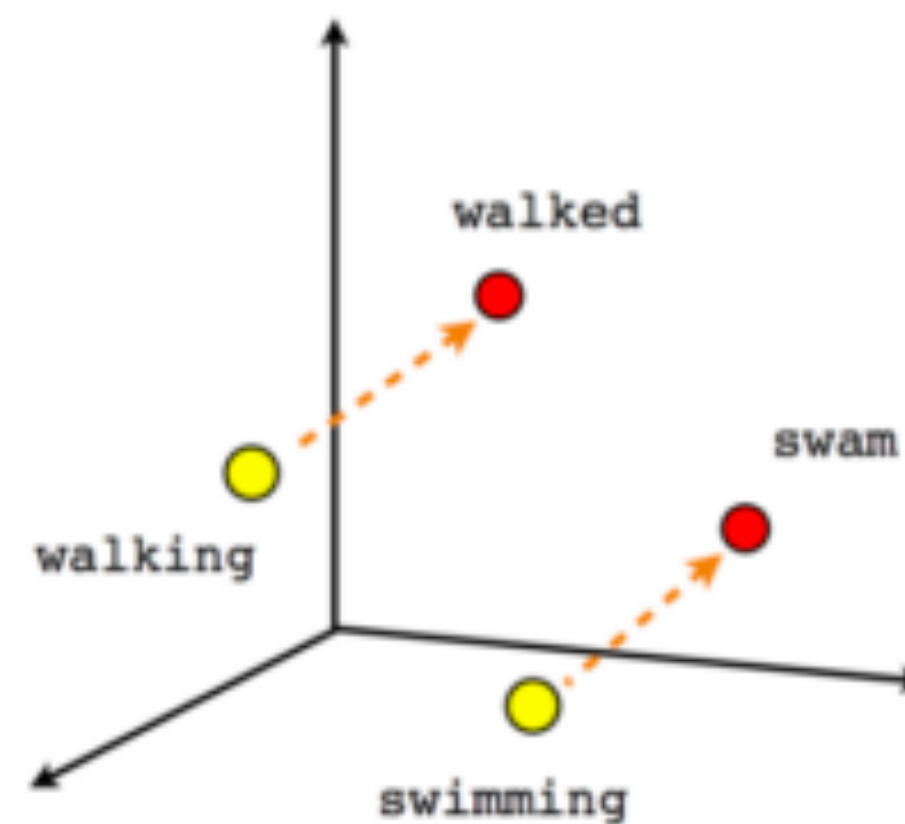
Word embeddings

- Represent words with low(ish)-dimensional vectors called **embeddings**
- Every word in vocabulary has a vector (in $\mathbf{R}^k$): these are model parameters.
- Ideally: semantically similar words get similar vectors. Or other semantic properties??

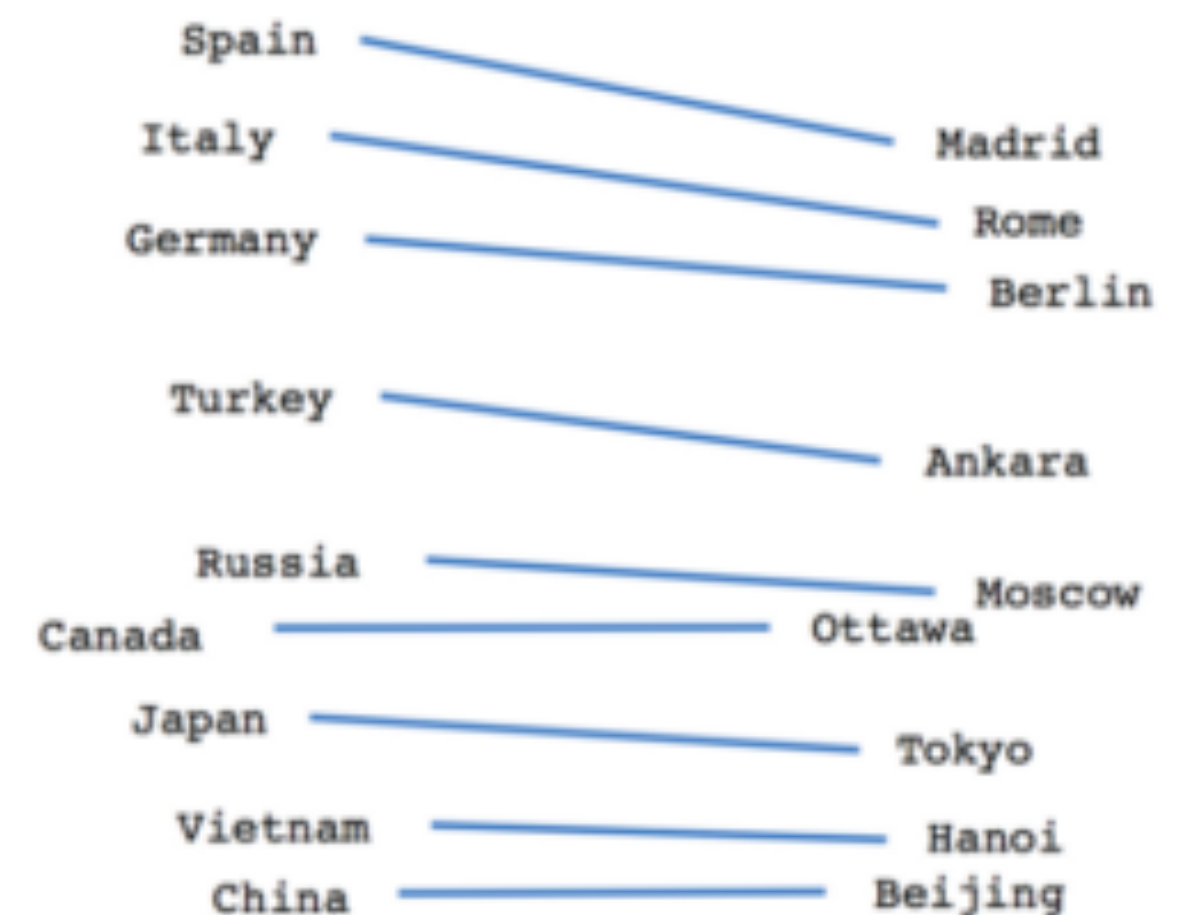
king =
[0.23, 1.3, -0.3, 0.43]



Male-Female



Verb tense



Country-Capital

why use vectors to encode meaning?

- computing the similarity between two words (or phrases, or documents) is *extremely* useful for many NLP tasks...
- Q: how **tall** is Mount Everest?
A: The official **height** of Mount Everest is 29029 ft
- ... and is foundational to neural networks models of language (later)

Word similarity for plagiarism detection

MAINFRAMES

Mainframes **are primarily** referred to large computers with **rapid**, advanced processing capabilities that **can execute and** perform tasks **equivalent to many** Personal Computers (PCs) machines **networked together**. It is **characterized with high quantity** Random Access Memory (RAM), very large secondary storage devices, and **high-speed** processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by **many and** most enterprises **and organizations**. **This is** one of its advantages. Mainframes are also suitable to cater for those applications **(programs)** or files that are of very **high**

MAINFRAMES

Mainframes **usually are** referred those computers with **fast**, advanced processing capabilities that **could perform by itself** tasks **that may require a lot of** Personal Computers (PC) Machines. **Usually mainframes would have lots of** RAMs, very large secondary storage devices, and **very fast** processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, **these computers** have the capability of running multiple large applications required by most enterprises, **which is** one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very **large** demand

- **Word Embeddings** (or *Word Vectors*):
 - Give each word a k-dimensional vector
 - a vector is a list of numbers
 - a vector is a point/direction in k-dimensional space
- A representation of words capable of (rough) synonyms, and maybe other aspects of meaning (affect, gender, formality...)
- We need to learn embeddings from text data.
 - Supports *transfer learning*
 - There are efforts for hand-built, dictionary-like word meaning databases (WordNet), but learning from corpora usually works better

Distributional Semantics

“You shall know a word by the company it keeps!” — Firth (1957)

Intuitions: Harris (1954)

“If A and B have almost identical environments except chiefly sentences which contain both, we say they are synonyms: *oculist* and *eye- doctor* .”

Intuition of distributional word similarity

A bottle of *tesgüino* is on the table
Everybody likes *tesgüino*
Tesgüino makes you drunk
We make *tesgüino* out of corn.

- From context words humans can guess **tesgüino** means...

Intuition of distributional word similarity

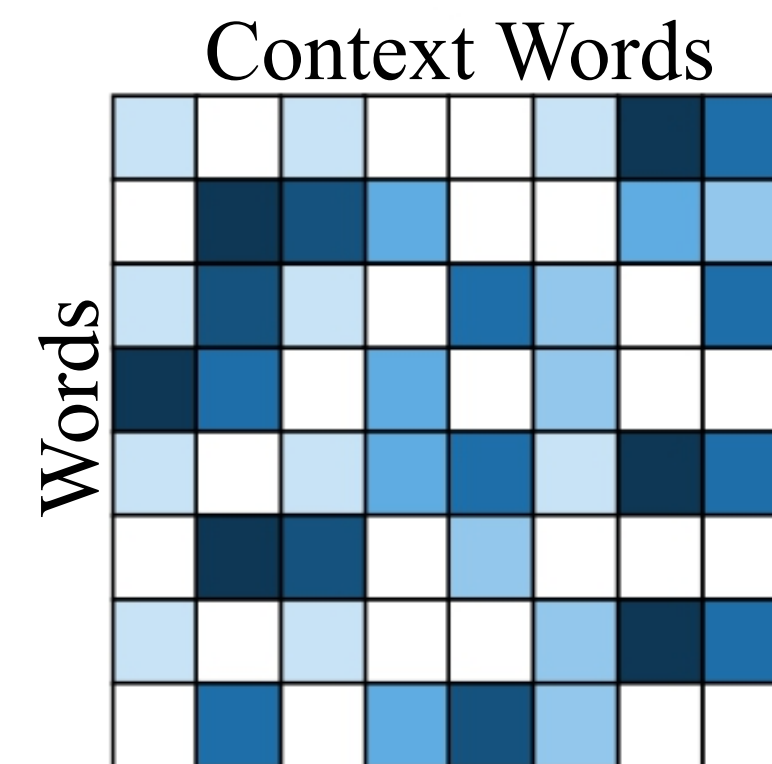
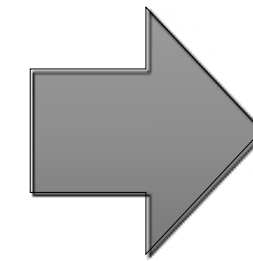
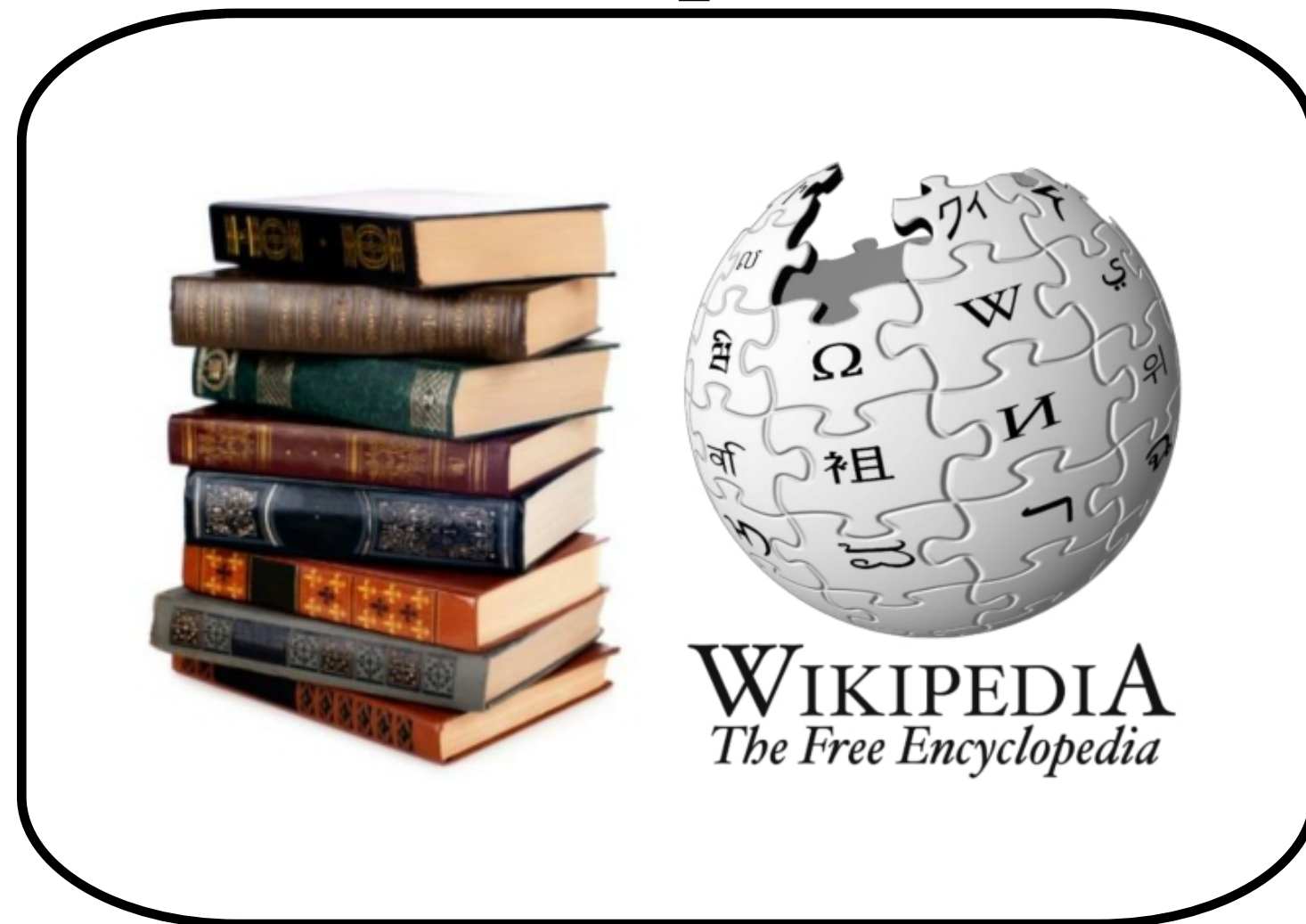
A bottle of **tesgüino** is on the table
Everybody likes **tesgüino**
Tesgüino makes you drunk
We make **tesgüino** out of corn.

- From context words humans can guess **tesgüino** means...
- an alcoholic beverage like **beer**
- Intuition for algorithm:
 - Two words are similar if they have similar word contexts.

i.e., contextual substitutability

Build vectors based on context

Corpus



Word-word co-occurrence matrix

sugar, a sliced lemon, a tablespoonful of
their enjoyment. Cautiously she sampled her first
well suited to programming on the digital
for the purpose of gathering data and

apricot
pineapple
computer.
information

jam, a pinch each of,
and another fruit whose taste she likened
In finding the optimal R-stage policy from
necessary for the study authorized in the

	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	

- Each row: one word vector (1 row for "target" word)
- Columns: context words
- Value: how many times the context word appears near the target word (within a context window, typically +/- 5 words or so)
- Build this matrix from a very large corpus of texts!

cosine similarity of two vectors

$$\text{cosine}(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|} = \frac{\sum_{i=1}^N v_i w_i}{\sqrt{\sum_{i=1}^N v_i^2} \sqrt{\sum_{i=1}^N w_i^2}}$$

v_i is the count for word v in context i

w_i is the count for word w in context i .

→ →

Cos(v, w) is the cosine similarity of v and w

→

→

$$\vec{a} \cdot \vec{b} = |\vec{a}| |\vec{b}| \cos \theta$$

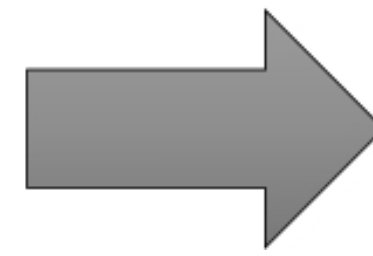
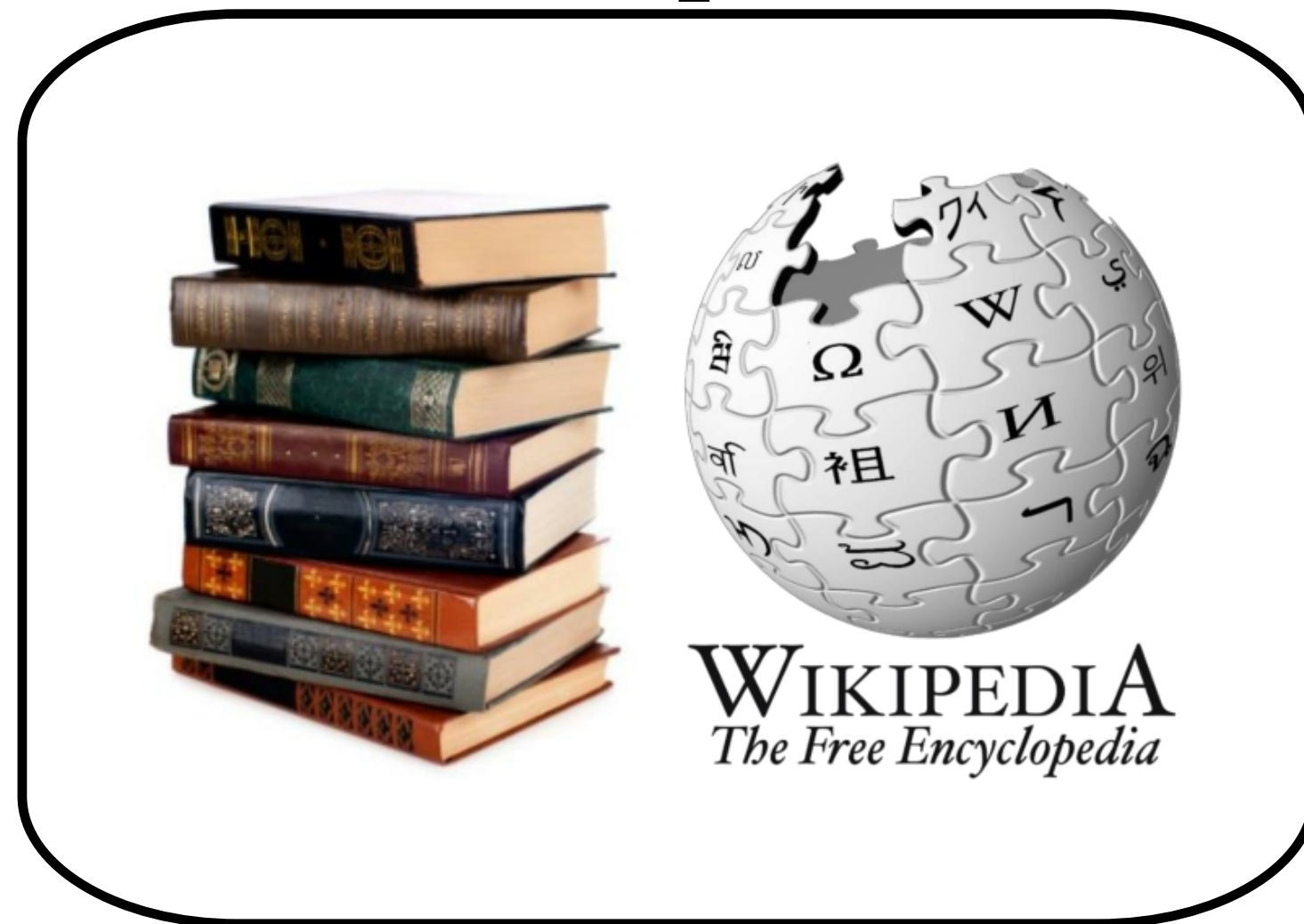
$$\frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} = \cos \theta$$

Word co-occurrence-based vectors

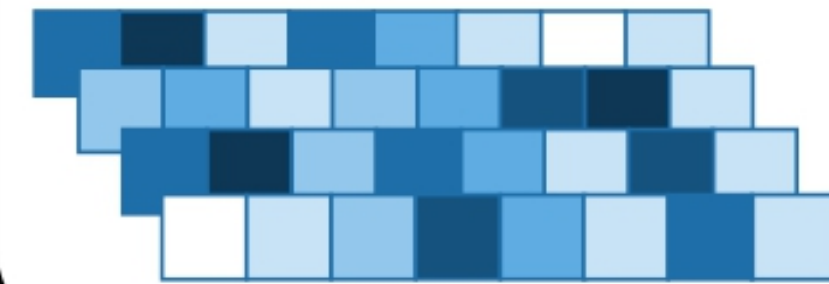
- These explicit contextual count vectors are surprisingly OK at rough similarity!
 - Adjusting for issues in highly frequent contexts can make them work better (PPMI)
- QUESTION: dimensionality of these word embeddings?
- Improvement: learn lower dimensional, *dense*, vectors for each word, to summarize what context words to expect

Neural Word Embeddings

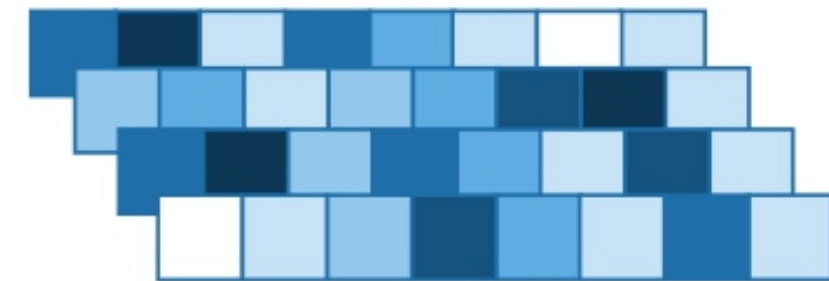
Corpus



Word



Context



Skip- Gram with Negative Sampling (SGNS)

The brown fox **jumps** over the lazy dog



SG NS: Skip- Gram Model

The brown fox jumps over the lazy dog.

Context Window Size = 2

SG NS: Skip- Gram Model

The brown fox jumps over the lazy dog.

word embedding

Simple idea: from a word, predict its context words!

Learn a vector that's good at that. Similar words should get similar vectors.

Key idea: use unlabeled text as *implicitly supervised data*

- A word s near *apricot*
 - Acts as gold ‘correct answer’ to the question
 - “Is word w likely to show up near *apricot*?”
- No need for hand-labeled supervision
- The idea comes from **neural language modeling**
 - Bengio et al. (2003)
 - Collobert et al. (2011)

Skip-gram with negative sampling (SGNS)

1. From a large source of text (e.g., Wikipedia), generate *positive examples* by pairing a target word with a word in its neighboring context
2. Create *negative examples* for that target word by randomly sampling other words in the vocabulary
3. Train a *logistic regression* model to identify whether a given pair of words is a positive or negative example
4. Use the weights of this model as the *embeddings*

Skip-Gram Training Data

Training sentence:

... lemon, a tablespoon of **apricot** jam a pinch ...

c1 c2 target c3 c4

Asssume context words are those in +/- 2
word window

Modeling

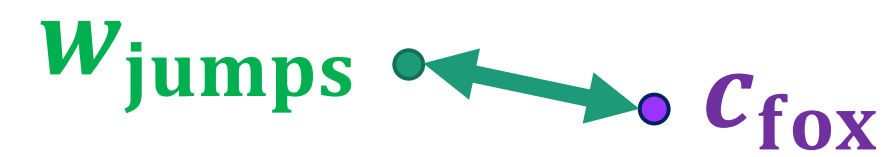
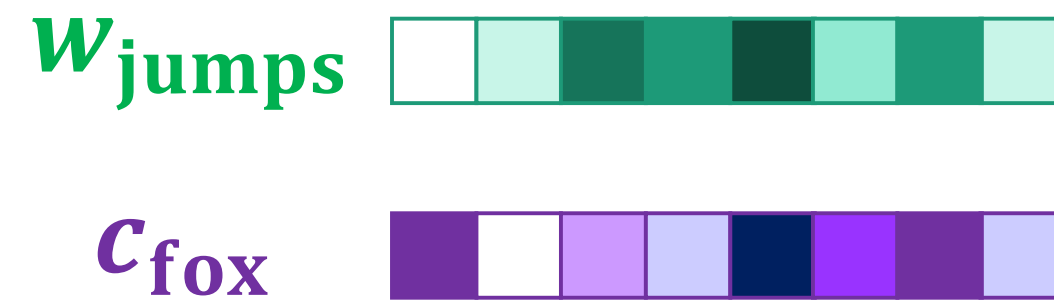
- From a large corpus, harvest many (word, context) pairs, both positive/real and negative/fake
 - $[+]$ (apricot, jam) $\leftarrow$ observed
 - $[-]$ (apricot, aardvark) $\leftarrow$ unseen
- Want binary probability
 - $P(c \mid w)$ for a real context $[+]$
 - $1 - P(c \mid w)$ for a “fake”, unseen context $[-]$
- Let u_w and v_c be their vectors.
- **$P(c \mid w) = g(u_w'v_c)$** : logistic in their *affinity/similarity*
 - $g = \text{logistic sigmoid}$
- Maximize sum of $\log P(c \mid w)$ over all (w, c) pairs (both real and negative/fake)

[after lecture - rewrote to SLP notation]

- From a large corpus, harvest many (word, context) pairs, both positive/real and negative/fake
 - $y=[+]$ $w=\text{apricot}$ $c=\text{jam}$ $\leftarrow$ observed, positive
 - $y=[-]$ $w=\text{apricot}$ $c=\text{aardvark}$ $\leftarrow$ unseen, negative sample
- Let u_w and v_c be the target word vector and context word vector
- Want binary probability
 - For a real context [+]:
 - **$P(+ | (w,c)) = g(u_w' v_c)$** : logistic in their *affinity/similarity* dot product
 - For "fake", unseen, negative context [-]:
 - **$P(- | (w,c)) = 1 - g(u_w' v_c)$**
- Maximize sum of $\log P(y | (w,c))$ over all (w, c) pairs (both real, $y=[+]$, and unseen/negative/fake, $y=[-]$)

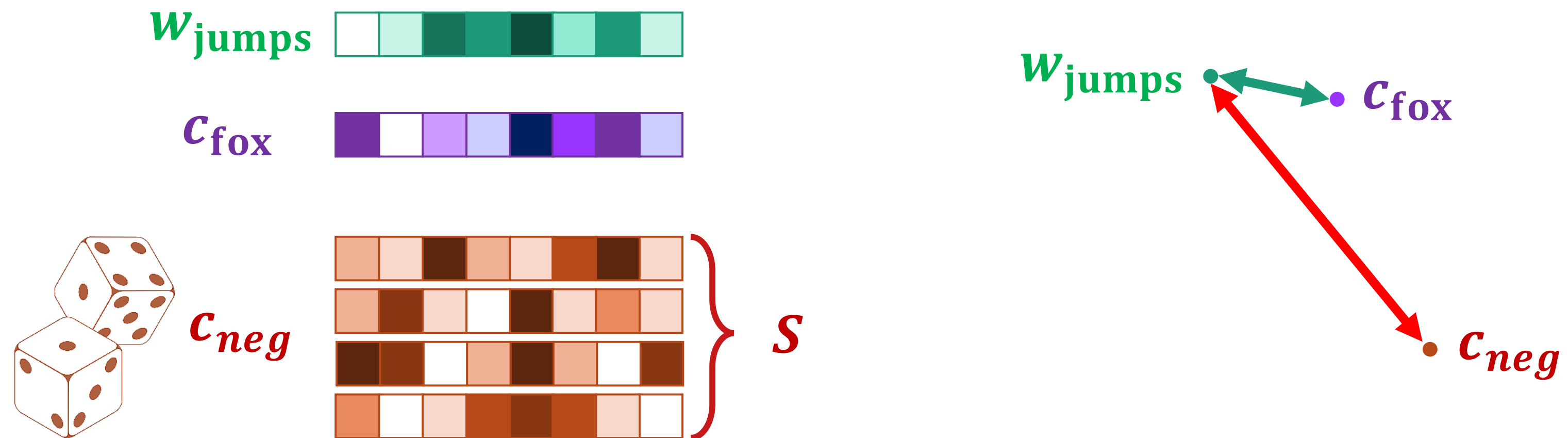
SGNS : Negative Sampling

Co-occurrence **jumps** , **fox**:



SGNS : Negative Sampling

Co-occurrence **jumps** , **fox**:



Defining contexts

Window size C affects the nature of the similarity
something like...

syntax $\longleftrightarrow$ specific meaning $\longleftrightarrow$ semantic fields or topical meaning

$C = \pm 2$ The nearest words to *Hogwarts*:

- *Sunnydale*
- *Evernight*

$C = \pm 5$ The nearest words to *Hogwarts*:

- *Dumbledore*
- *Malfoy*
- *halfblood*