CMPU 366 · Natural Language Processing

Word2vec

1 October 2025

VASSAR COLLEGE
A·D·1861

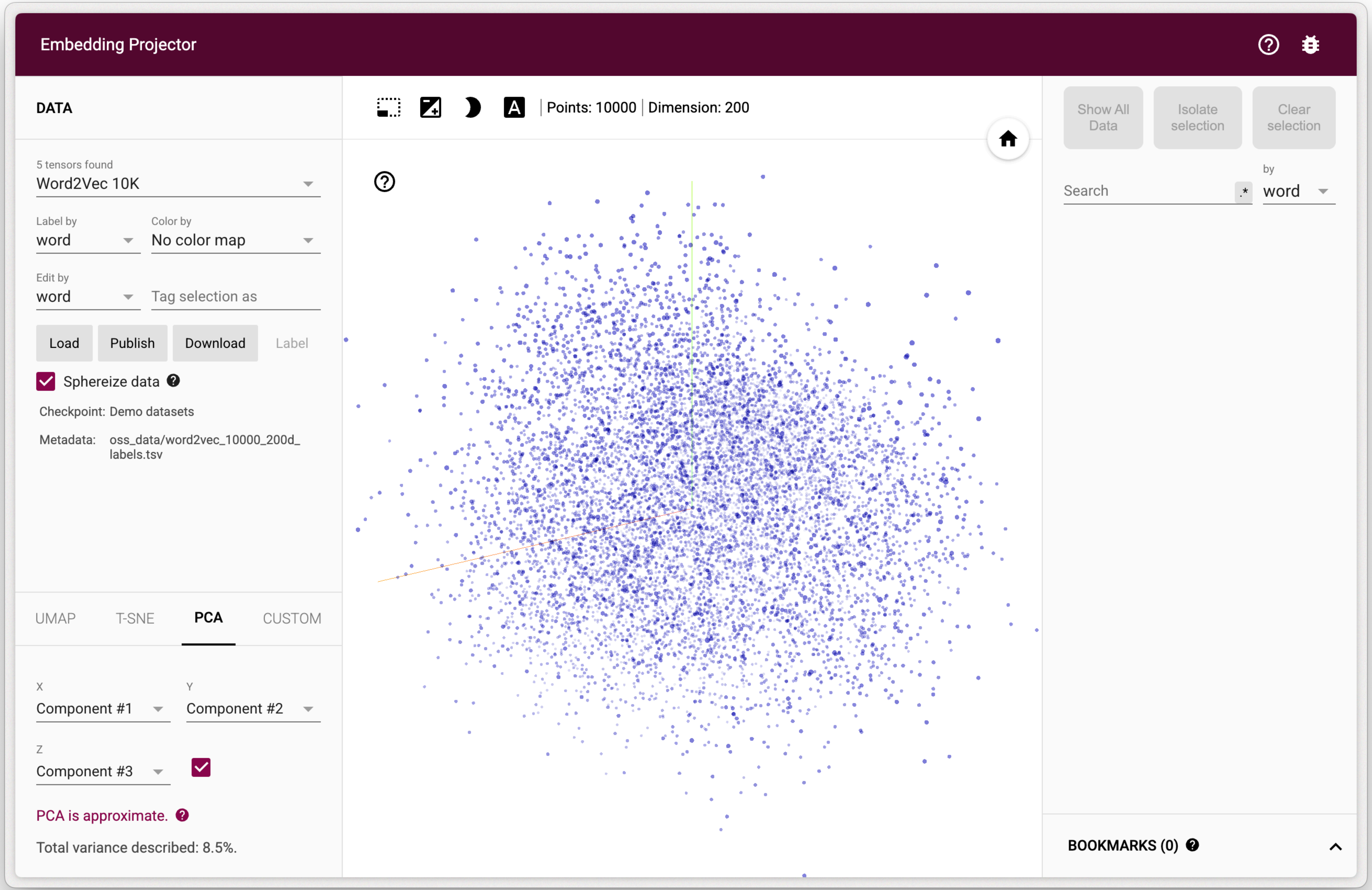
Where are we?

semantle.com

Lexical semantics is the study of how words carry meaning.

The *distributional hypothesis* is that the meaning of a word (or phrase) can be derived from the contexts it occurs in.

In *vector semantics*, we represent the meaning of a word as a vector – a point in a multi-dimensional space – that's learned from the contexts we observe the word in.



projector.tensorflow.org

Last class, we saw a way to learn a vector semantics model: Count how many times each token occurs near it (within some fixed-size window of tokens):

	eat	*fall*	*ripe*	*slice*	*peel*	*tree*	*throw*	*fruit*	*pie*	*bite*	*crab*
apple	794	244	47	221	208	160	145	156	109	104	88
orange	265	22	25	62	220	64	74	111	4	4	8

Each row is an embedding.

These simple count embeddings are:

long: there are many, many dimensions – one for every word in the vocabulary

sparse: mostly zeros because most words do not co-occur

In practice, short dense vectors perform better:

Short vectors are easier to use as features in machine learning – fewer weights to tune!

Dense vectors generalize better than explicit counts – and they may do better at capturing synonymy.

The words *car* and *automobile* are synonyms, but in the vectors we considered last class they'd be distinct dimensions.

A word with *car* as a neighbor and a word with *automobile* as a neighbor are probably similar, but the embedding wouldn't capture that.

Word2vec:
Skip-gram negative sampling (SGNS)

IDEA: Instead of counting how often each word *c* occurs near, say, *apricot*, we'll instead train a classifier on a binary prediction task:

"Is word *c* likely to show up near *apricot*?"

The weights the classifier learns are our embeddings!

Target word

apricot

Target word in corpus

… lemon , a tablespoon of $\boxed{\text{apricot}}$ jam , a pinch …

Context window of ±2 tokens

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

c_1 c_2 w c_3 c_4

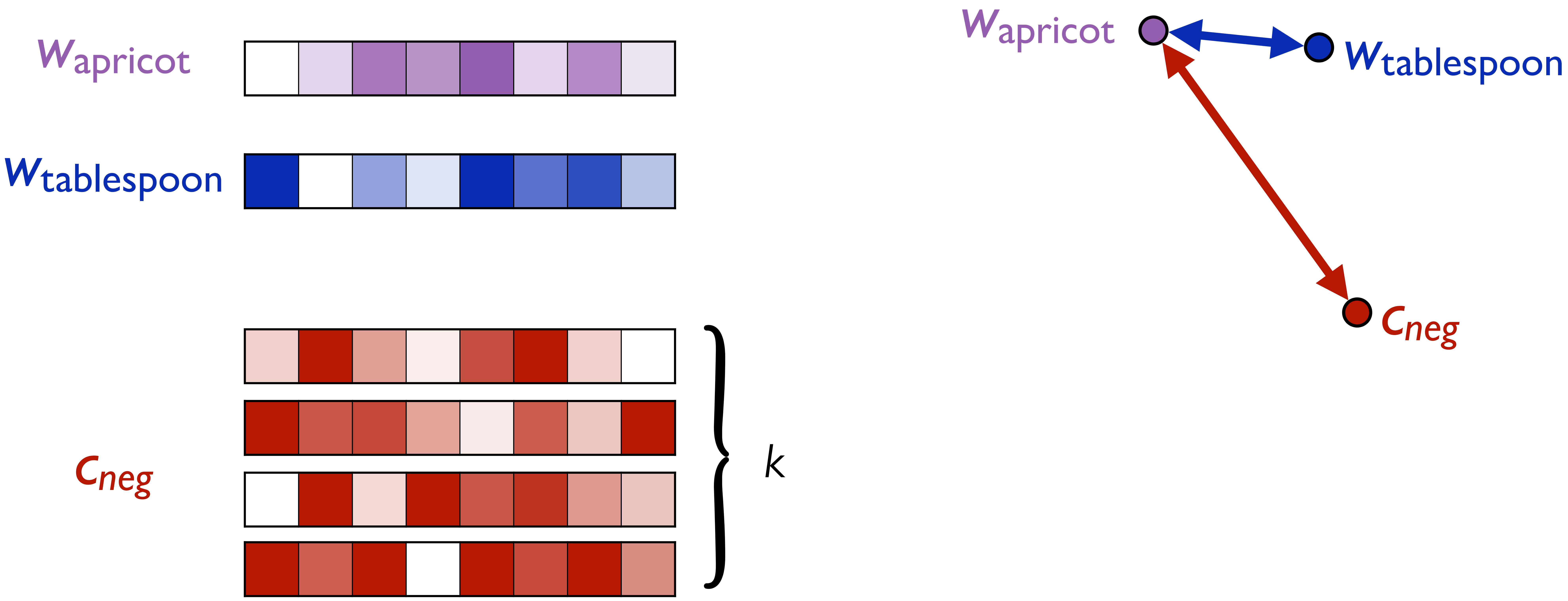
Set of context words

apricot → {*tablespoon*, *of*, *jam*, *,*}

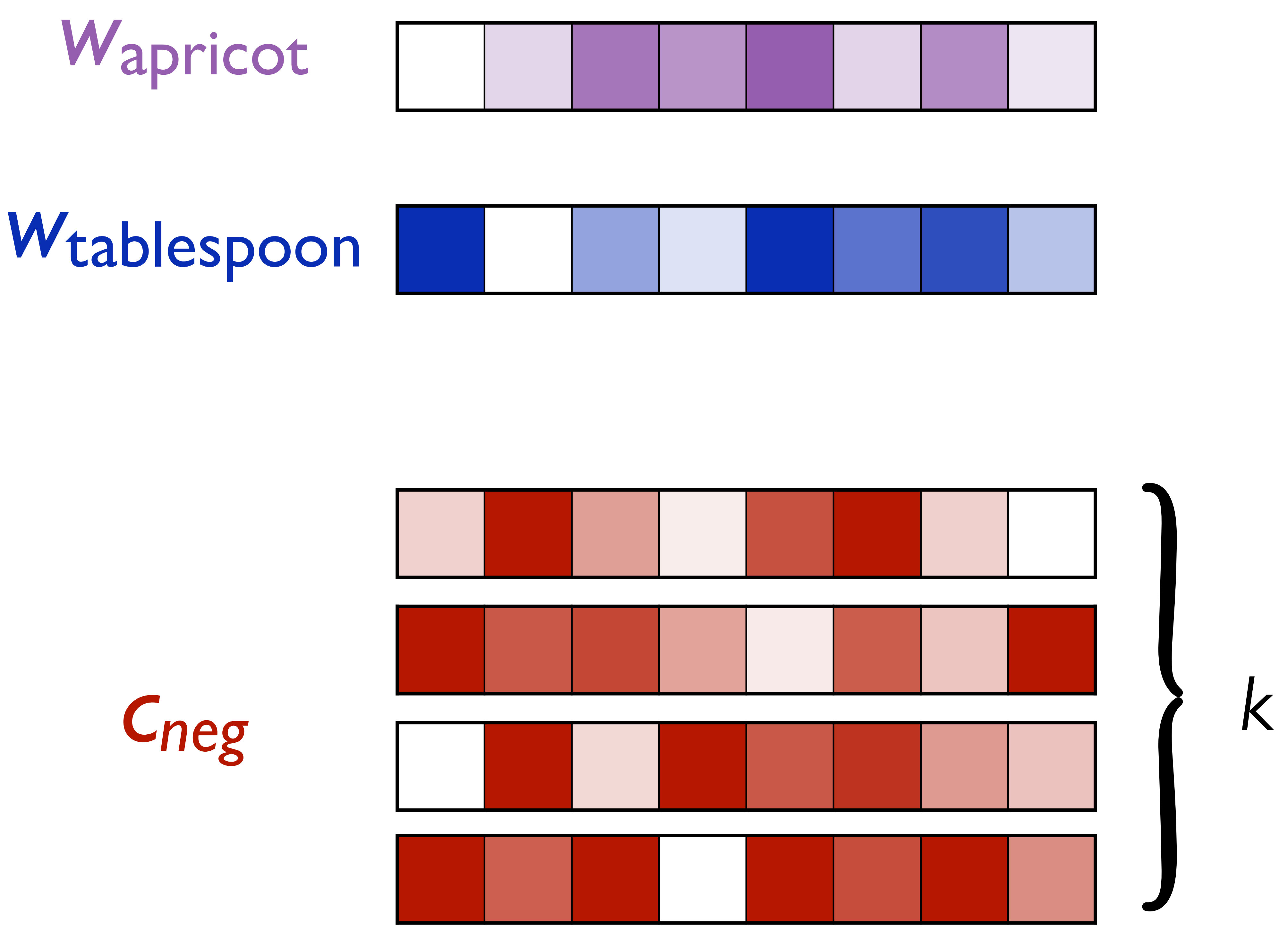
… *lemon* , *a* *tablespoon* *of* *apricot* *jam* , *a* *pinch* …

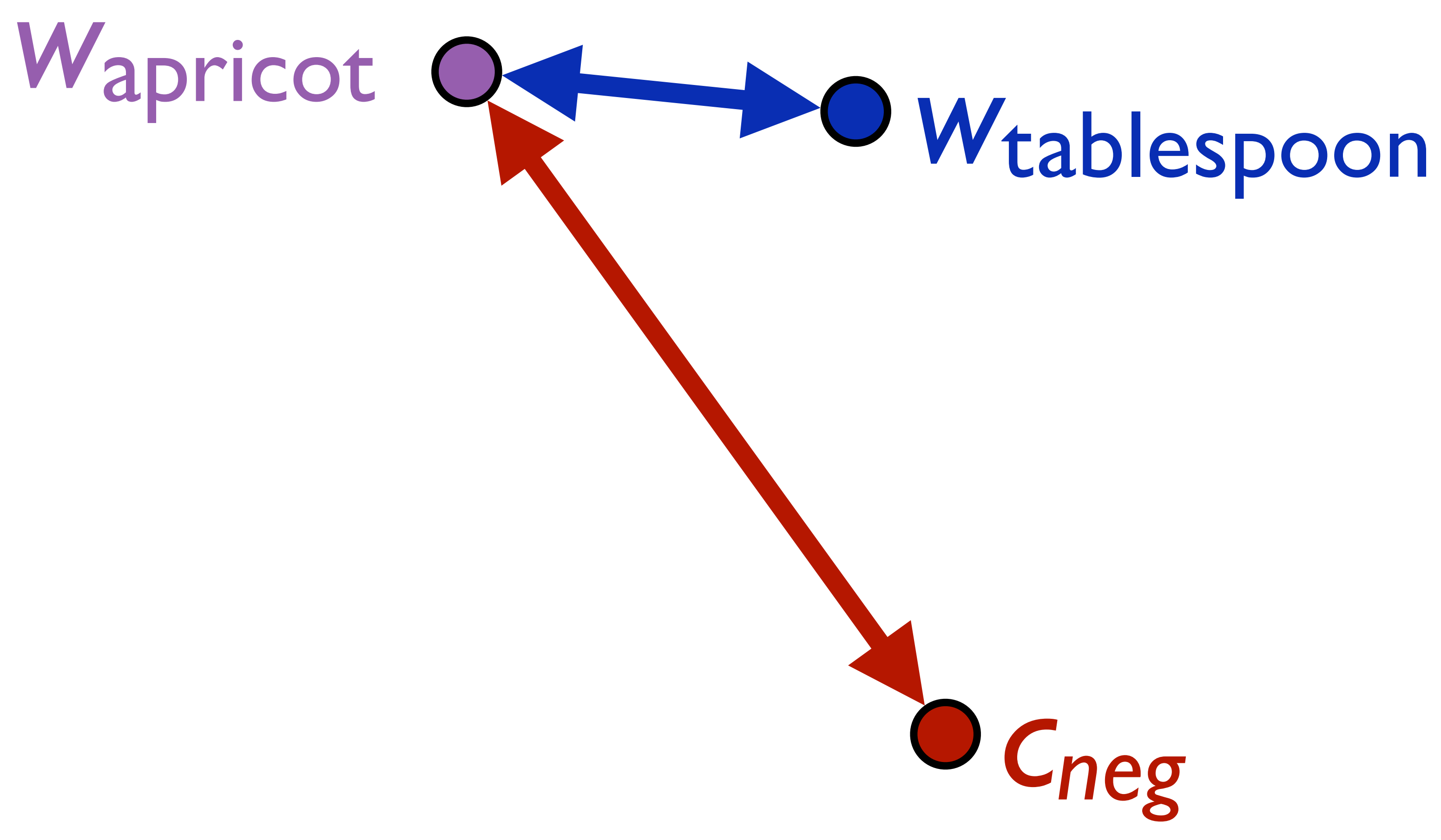
c_1 c_2 w c_3 c_4

apricot → {*tablespoon*, *of*, *jam*, *,*}



apricot → {*tablespoon*, *of*, *jam*, ,}





Algorithm goal: Context embeddings closer to target embeddings than embeddings of randomly sampled words

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

c_1 c_2 w c_3 c_4

GOAL: Train a classifier that is given a pair of tokens (w, c), e.g., (*apricot*, *jam*) or (*apricot*, *aardvark*) and assigns the probability $P(+ \mid w, c)$ that c is actually in the context window of w.

$$P(+ \mid w, c) \approx \quad \mathbf{c} \cdot \mathbf{w}$$

Intuition: Similar words occur together. The vectors for w and c are similar if they have a high dot product.

$$P(+\mid w, c) = \sigma(\mathbf{c} \cdot \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{c} \cdot \mathbf{w})}$$

The sigmoid squishes that dot product into a probability.

$$P(+ \mid w, c) = \sigma(\mathbf{c} \cdot \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{c} \cdot \mathbf{w})}$$

Simplifying (incorrect) assumption: All the context words are independent, so we can just multiply their probabilities:

$$P(+ \mid w, c_{1:L}) = \prod_{i=1}^{L} \sigma(\mathbf{c_i} \cdot \mathbf{w})$$

Probability of target word w appearing in the window $c_{1:L}$

$$P(+\mid w,c)=\sigma(\mathbf{c}\cdot\mathbf{w})=\frac{1}{1+\exp(-\mathbf{c}\cdot\mathbf{w})}$$

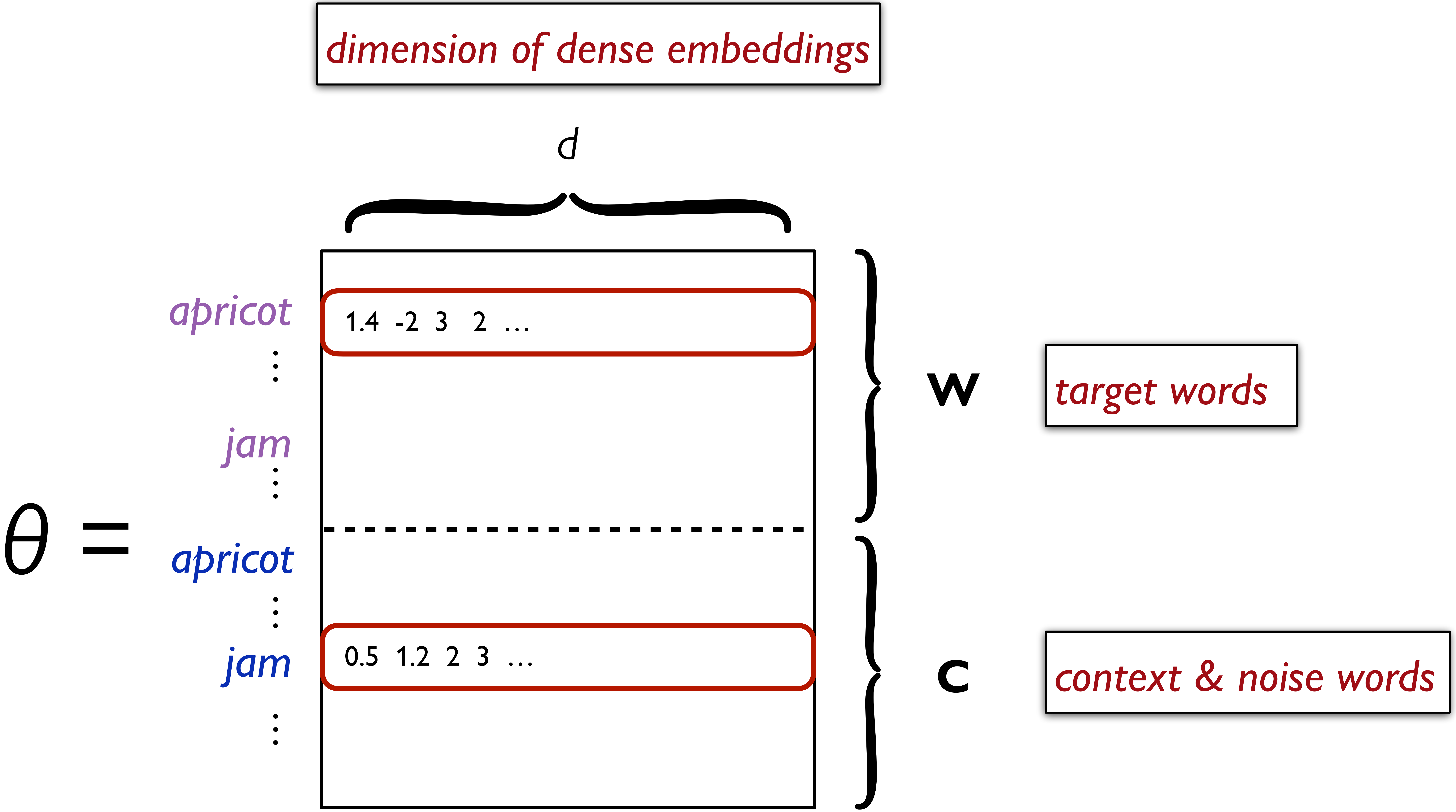
Simplifying (incorrect) assumption: All the context words are independent, so we can just multiply their probabilities:

$$P(+\mid w,c_{1:L})=\prod_{i=1}^{L}\sigma(\mathbf{c_i}\cdot\mathbf{w})$$

Probability of target word w appearing in the window $c_{1:L}$

$$\log P(+\mid w,c_{1:L})=\sum_{i=1}^{L}\log\sigma(\mathbf{c_i}\cdot\mathbf{w})$$

Embeddings as weights



Loss function

Maximize the similarity of the target with the actual context words, and minimize the similarity of the target with the *k* negative sampled non-neighbor words.

$$L = -\left[\log[\sigma(\mathbf{w} \cdot \mathbf{c}_{\text{pos}})] \ \log[\sigma(-\mathbf{w} \cdot \mathbf{c}_{\text{neg}})]\right]$$

For more than 1 negative example:

$$L = -\left[\log \sigma(\mathbf{c}_{\text{pos}} \cdot \mathbf{w}) + \sum_{i=1}^{k} \log \sigma(-\mathbf{c}_{\text{neg}_i} \cdot \mathbf{w})\right]$$

As with logistic regression, we improve the performance using gradient descent, taking a step in the direction that the loss (error) slopes down – away from the gradient of the loss function.

We're training a classifier, but we don't need humans to label training data for us!

We treat the words we see within the window as our *positive examples*.

We sample other words from the corpus, which don't occur in the window, as the *negative examples*.

This approach is called *self-supervision*.

Which words are close in the vector space depends on the window size

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

Sunnydale

Evernight

Blandings

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

Dumbledore

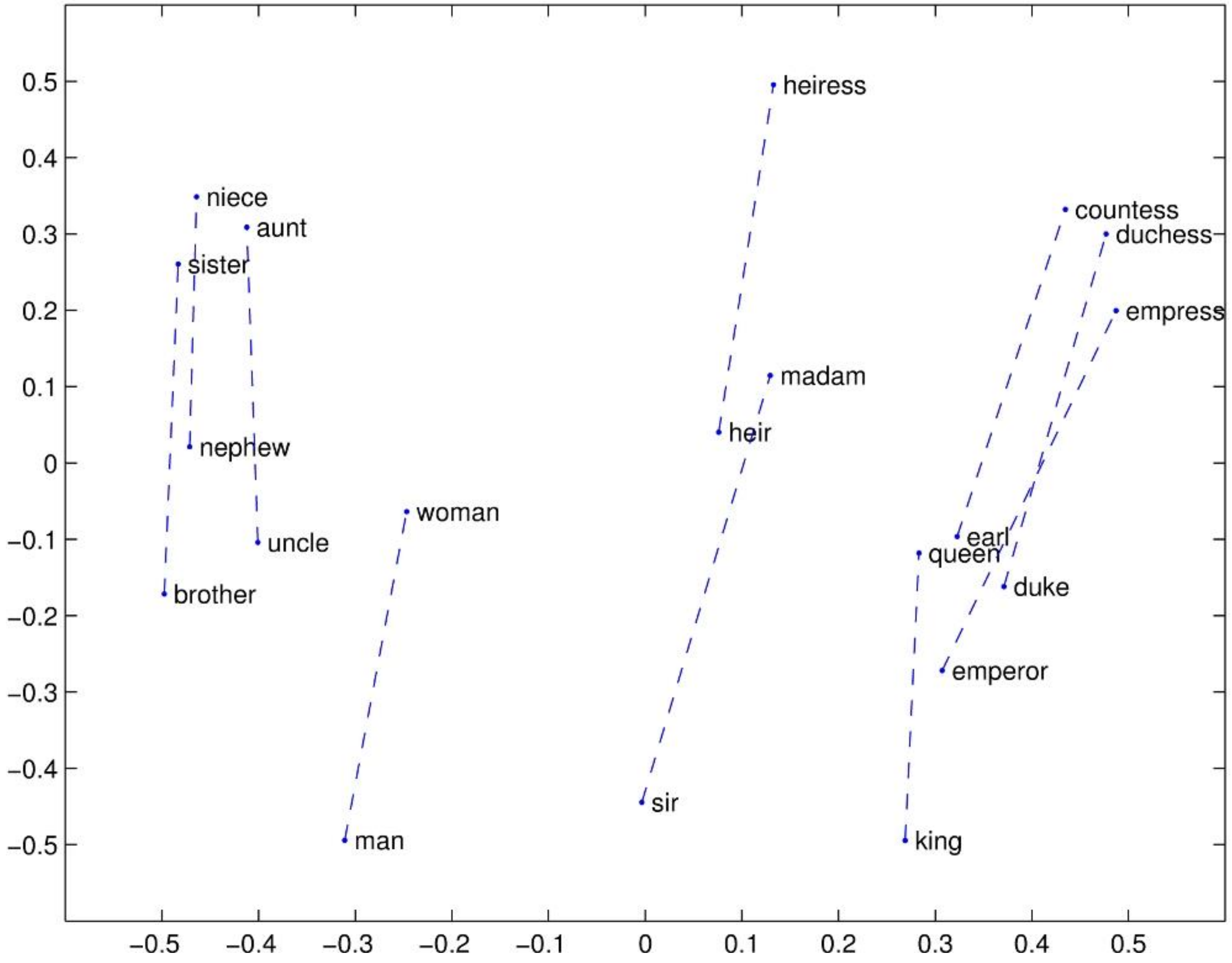
half-blood

Malfoy

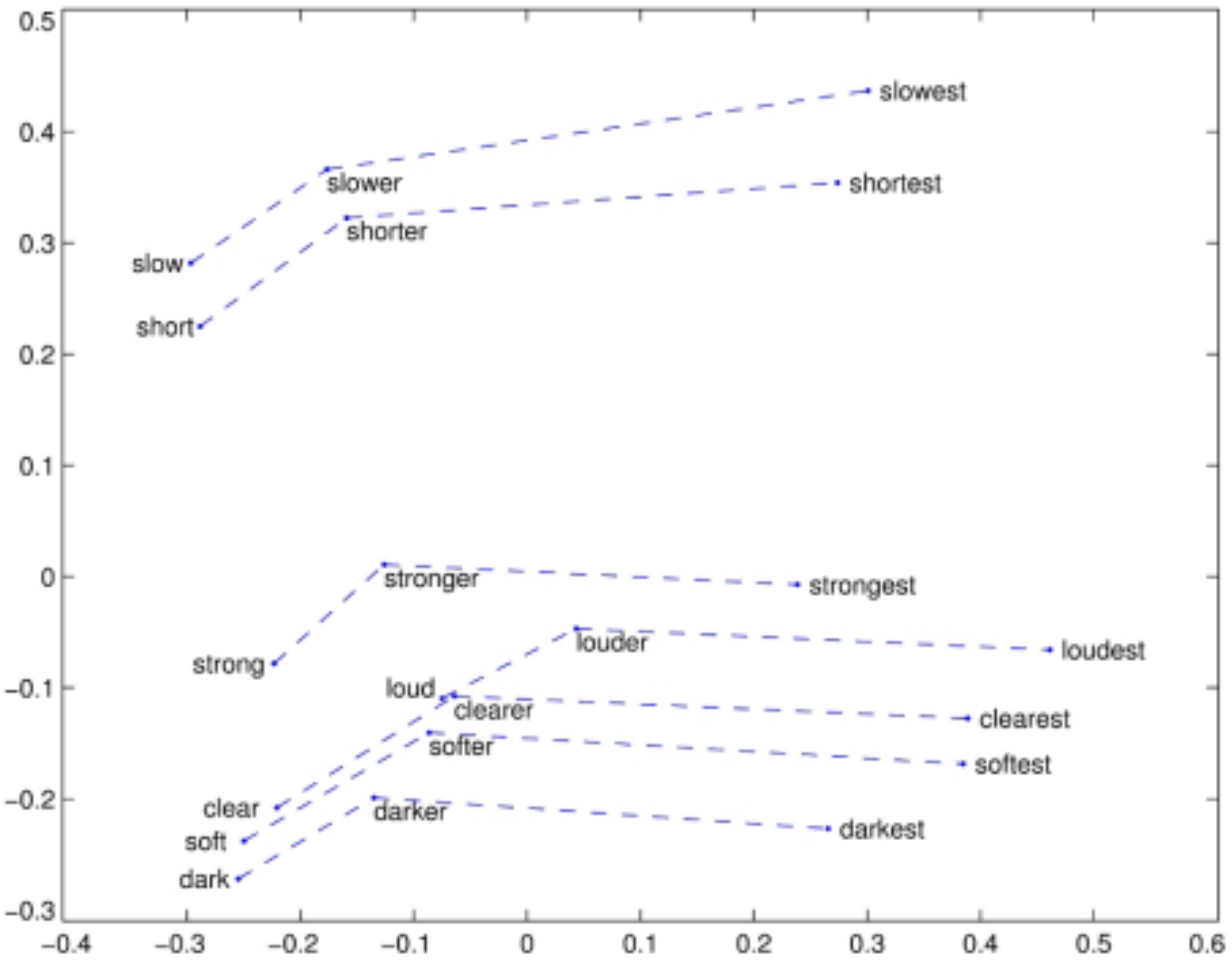
Levy and Goldberg, 2014

What knowledge do embeddings capture?

Word relations



A 2D projection of word embeddings from GloVE, a similar model to Word2vec



A 2D projection of word embeddings from GloVE, a similar model to Word2vec

Analogies

Analogy task

a : b :: aa : bb

man : king :: woman : ____?

Find bb

Analogy task

Rumelhart and Abrahamson. 1973

a : b :: aa : bb

man : king :: woman : _____?

Find bb

Vector parallelogram method

bb = b − a + aa

Find the closest word to that point

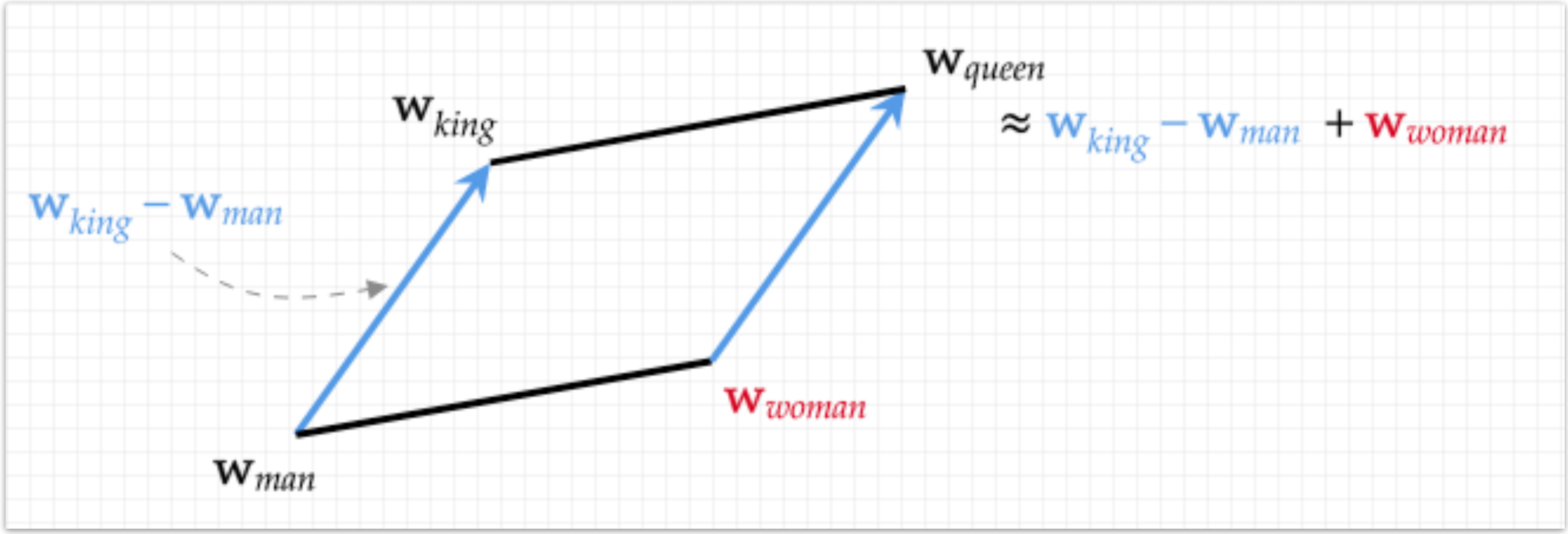


Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (Skip-gram model trained on 783M words with 300 dimensionality).*

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

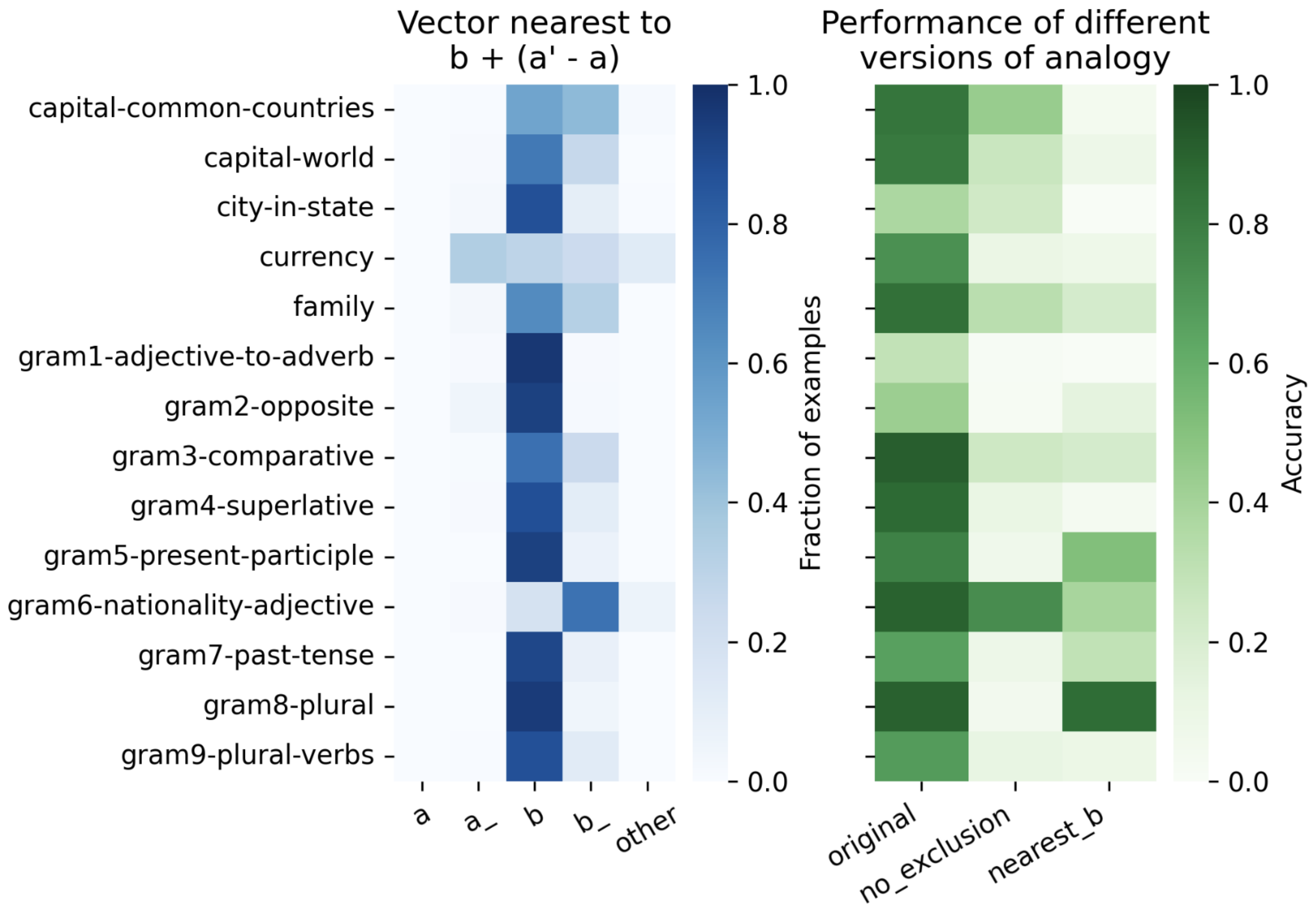
Mikolov et al. 2013

The original analysis excluded morphological variants from the possible predictions

Example: *cherry* : *red* :: *potato* : *x*

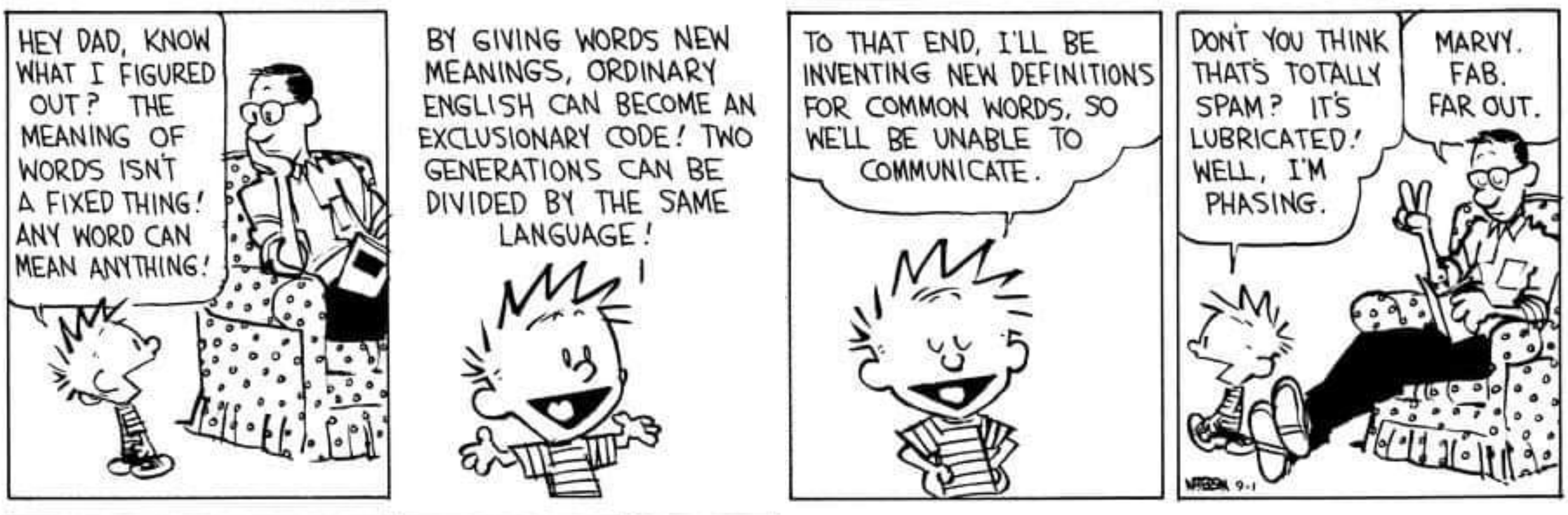
- *x* predictions are usually *potato* or *potatoes* instead of *brown*, so the former two are typically excluded

Significantly worse performance when not excluding



Linzen 2016

Using embeddings to study culture



HEY DAD, KNOW WHAT I FIGURED OUT? THE MEANING OF WORDS ISN'T A FIXED THING! ANY WORD CAN MEAN ANYTHING!
BY GIVING WORDS NEW MEANINGS, ORDINARY ENGLISH CAN BECOME AN EXCLUSIONARY CODE! TWO GENERATIONS CAN BE DIVIDED BY THE SAME LANGUAGE!
TO THAT END, I'LL BE INVENTING NEW DEFINITIONS FOR COMMON WORDS, SO WE'LL BE UNABLE TO COMMUNICATE.
DON'T YOU THINK THAT'S TOTALLY SPAM? IT'S LUBRICATED! WELL, I'M PHASING.
MARVY. FAB. FAR OUT.
WATTERSON 9-1

Train embeddings on different decades of historical text to see meanings shift:

The modern sense of each word and the grey context words computed from the most recent (modern) embedding space. Earlier points computed from embeddings trained on earlier historical data.

Hamilton et al. 2016

www.pnas.org/doi/10.1073/pnas.1720347115

PNAS

ARTICLES FRONT MATTER AUTHORS TOPICS +

SIGN IN SUBMIT

RESEARCH ARTICLE | SOCIAL SCIENCES

Word embeddings quantify 100 years of gender and ethnic stereotypes

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou Authors Info & Affiliations

Edited by Susan T. Fiske, Princeton University, Princeton, NJ, and approved March 12, 2018 (received for review November 22, 2017)

April 3, 2018 | 115 (16) E3635-E3644 | https://doi.org/10.1073/pnas.1720347115

Significance

Word embeddings are a popular machine-learning method that represents each English word by a vector, such that the geometry between these vectors captures semantic relations between the corresponding words. We demonstrate that word embeddings can be used as a powerful tool to quantify historical trends and social change. As specific applications, we develop metrics based on word embeddings to characterize how gender stereotypes and attitudes toward ethnic minorities in the United States evolved during the 20th and 21st centuries starting from 1910. Our framework opens up a fruitful intersection between machine learning and quantitative social science.

Table 2. Top adjectives associated with women in 1910, 1950, and 1990 by relative norm difference in the COHA embedding

1910	1950	1990
Charming	Delicate	Maternal
Placid	Sweet	Morbid
Delicate	Charming	Artificial
Passionate	Transparent	Physical
Sweet	Placid	Caring
Dreamy	Childish	Emotional
Indulgent	Soft	Protective
Playful	Colorless	Attractive
Mellow	Tasteless	Soft
Sentimental	Agreeable	Tidy

Garg et al. 2018

Strong biases are reflected not just in historic text, but also in contemporary corpora like the Google News data that Word2vec was trained on.

Table 1. The top 10 occupations most closely associated with each ethnic group in the Google News embedding

Hispanic	Asian	White
Housekeeper	Professor	Smith
Mason	Official	Blacksmith
Artist	Secretary	Surveyor
Janitor	Conductor	Sheriff
Dancer	Physicist	Weaver
Mechanic	Scientist	Administrator
Photographer	Chemist	Mason
Baker	Tailor	Statistician
Cashier	Accountant	Clergy
Driver	Engineer	Photographer

Garg et al. 2018

Using the analogy method on Word2vec, we find

man : *computer programmer* :: *woman* : ____________

Bolukbasi et al., 2016

Using the analogy method on Word2vec, we find

man : *computer programmer* :: *woman* : *homemaker*

Bolukbasi et al., 2016

Using the analogy method on Word2vec, we find

man : *computer programmer* :: *woman* : *homemaker*

Bolukbasi et al., 2016

There's been significant research in recent years on mitigating bias in word embeddings, but it's impossible to avoid these issues altogether when learning from naturally occurring text.

Acknowledgments

This class incorporates material from:

Jurafsky & Martin, *Speech and Language Processing*, 3rd ed. draft

Carolyn Anderson, Wellesley College

Katie Keith, Williams College