

344.075 KV: Natural Language Processing Word Embedding with Matrix Factorization



Navid Rekab-Saz

navid.rekabsaz@jku.at

Agenda

- Distributional semantics & word embedding
- PMI+SVD
- GloVe

Agenda

- **Distributional semantics & word embedding**
- PMI+SVD
- GloVe

Meaning & Semantics

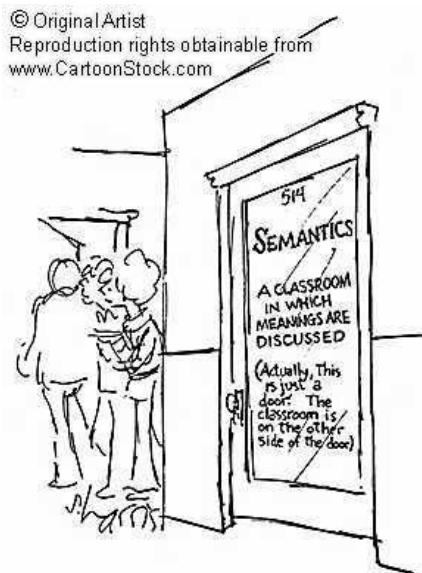
Meaning:

What is meant by a word, text, concept, or action. (Oxford)

The thing one intends to convey especially by language. (Merriam-Webster)

Semantics:

The branch of linguistics and logic concerned with meaning ... *lexical semantics* [is] concerned with the analysis of word meanings and relations between them. (Oxford)

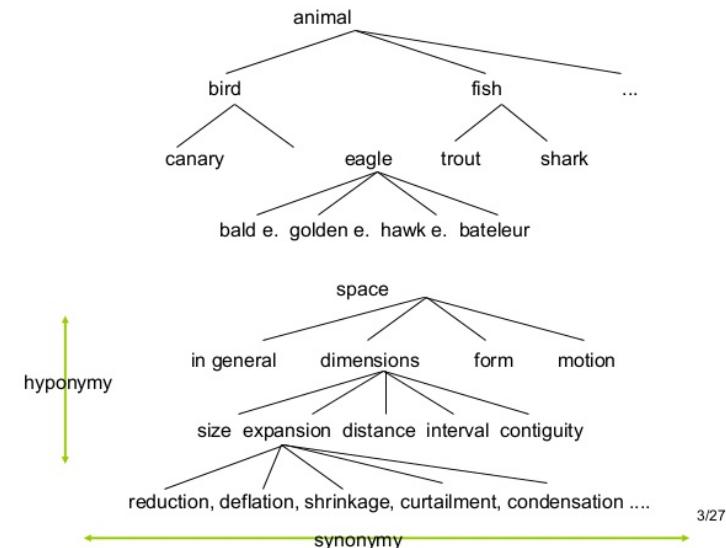


Semantics with knowledge resources



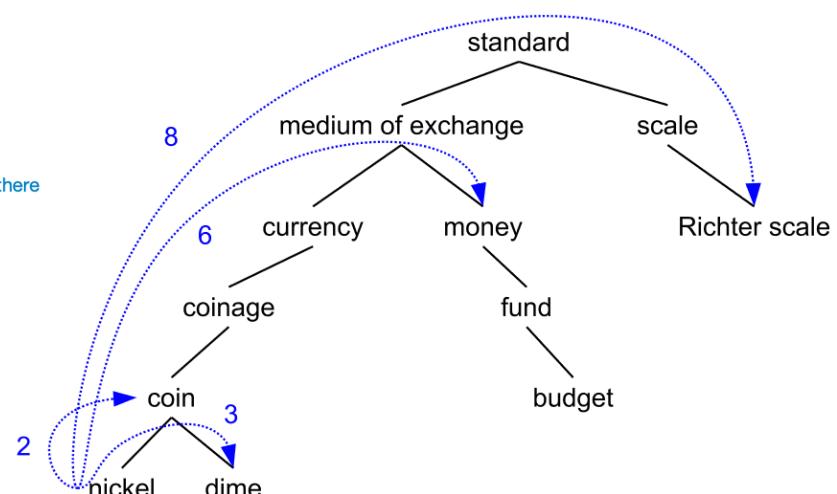
Noun language has 6 senses

1. **language, linguistic communication** - a systematic means of communicating by the use of sounds or conventional symbols; "he taught foreign languages"; "the language introduced is standard throughout the text"; "the speed with which a program can be executed depends on the language in which it is written"
 - 1 is a kind of **communication**
 - 1 has particulars:
 - dead language; words; source language; object language, target language; sign language, signing; artificial language; metalanguage; native language; natural language, tongue; lingua franca, interlanguage, koine; string of words, word string, linguistic string; barrage, outpouring, onslaught; slanguage**
2. **speech, speech communication, spoken communication, spoken language**, language, **voice communication, oral communication** - (language) communication by word of mouth; "his speech was garbled"; "he uttered harsh language"; "he recorded the spoken language of the streets"
 - 2 is a kind of **auditory communication**
 - 2 has particulars:
 - words; pronunciation, orthoepy; conversation; discussion, give-and-take, word; saying, expression, locution; non-standard speech; idiolect; monologue; spell, magic spell, charm; dictation; soliloquy, monologue**
3. **terminology, nomenclature**, language - a system of words used in a particular discipline; "legal terminology"; "the language of sociology"
 - 3 is a kind of **word**
 - 3 has particulars: **markup language; toponymy, toponomy**



Noun elephant has 2 senses

1. **elephant - five-toed pachyderm**
 - 1 is a kind of **proboscidean, proboscidian**
 - 1 is a member of **Elephantidae, family Elephantidae**
 - 1 has particulars:
 - rogue elephant; Indian elephant, Elephas maximus; African elephant, Loxodonta africana; mammoth; gomphothere**
2. **elephant - the symbol of the Republican Party; introduced in cartoons by Thomas Nast in 1874**
 - 2 is a kind of **emblem, allegory**



Read more here: <https://web.stanford.edu/~jurafsky/slp3/C.pdf>

<https://wordnet.princeton.edu>

<http://wordnet-online.freewisdom.com/definition?word=language>

<https://www.slideshare.net/AhmedAbdElwasaa/wordnet-a-database-of-lexical-relations>

Distributional Semantics



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

*J. R. Firth, A synopsis of
linguistic theory 1930–1955
(1957)*

Distributional Semantics

“A word’s **meaning** is given by the words that frequently appear close-by”

- The **context** of a word w is the set of words appear in the nearby of w (e.g. within a fixed-size window)
 - The words in context are **context-words**
- We use many contexts of w to create a representation of w
 - e.g. the meaning of *banking*

...government debt problems turning into banking crises as happened in 2009...

...saying that Europe needs unified banking regulation to replace the hodgepodge...

...India has just given its banking system a shot in the arm...

One of the most successful ideas of modern statistical NLP!



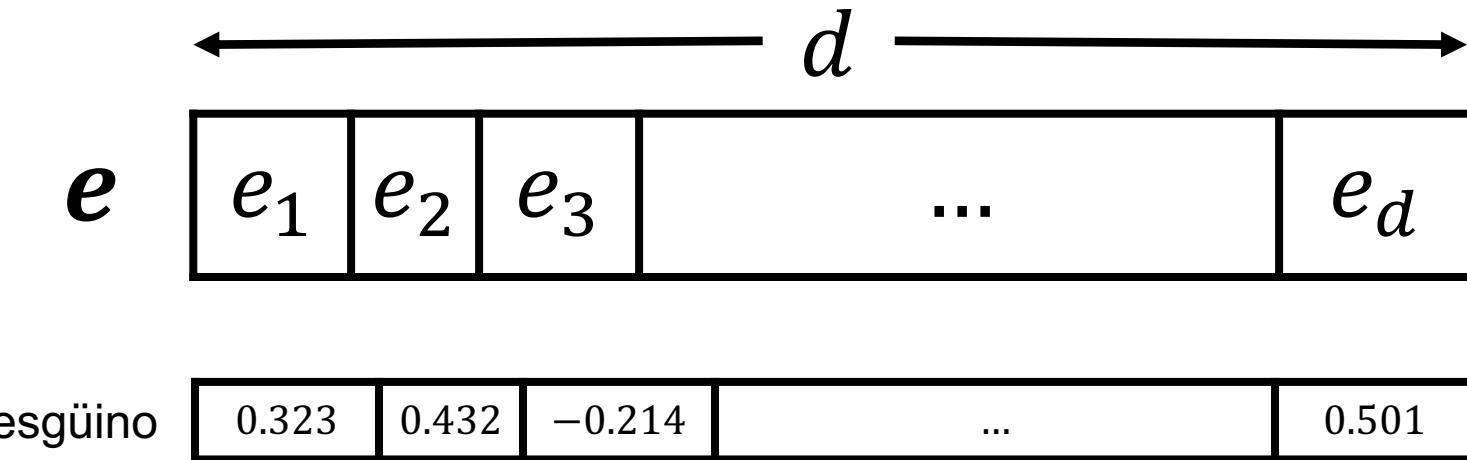
The Tarahumara people gather every year during Easter week (*semana santa*) and drink large amounts of Tesgüino together while following rituals. According to the anthropologist [Bill Merrill](#) of the Smithsonian Institute, the sacred drink chases large souls from the persons who drink it, «and so when people get drunk that's why they act like children [...] because the souls that are controlling their actions are the little souls, like little children».

<https://en.wikipedia.org/wiki/Tesgüino>

Nida[1975]

Distributional Representation

- A word (token) is represented with a vector of d dimensions
- Each dimension can be seen as a “feature” of the entity
- Units in a layer are not mutually exclusive
- Two units can be “active” at the same time



drink

sacred

beer

ritual

Tesgüino

corn

brew

Mexico

Tarahumara people

dark brown
lager

Märzen

beer

bottle
brew
drink

Bavaria

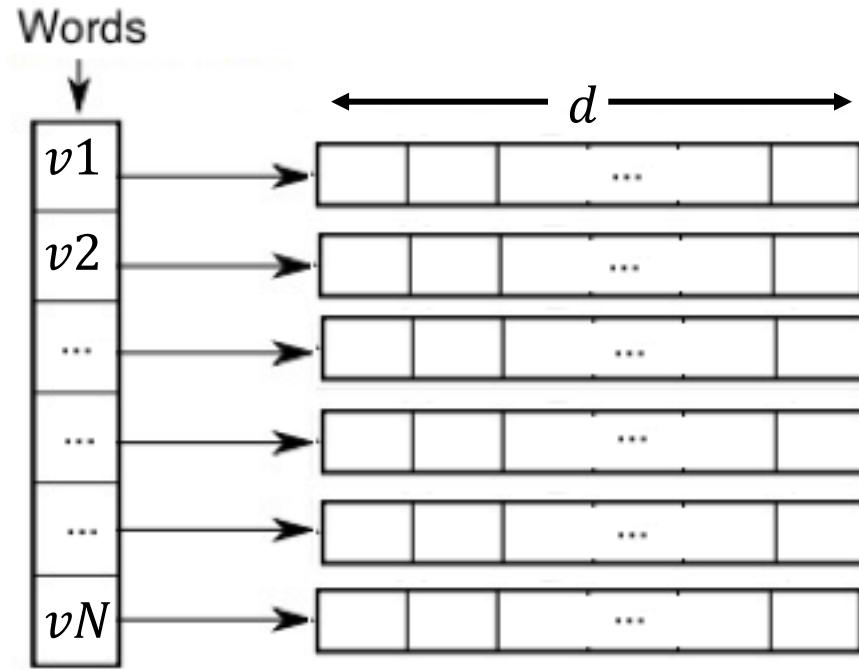
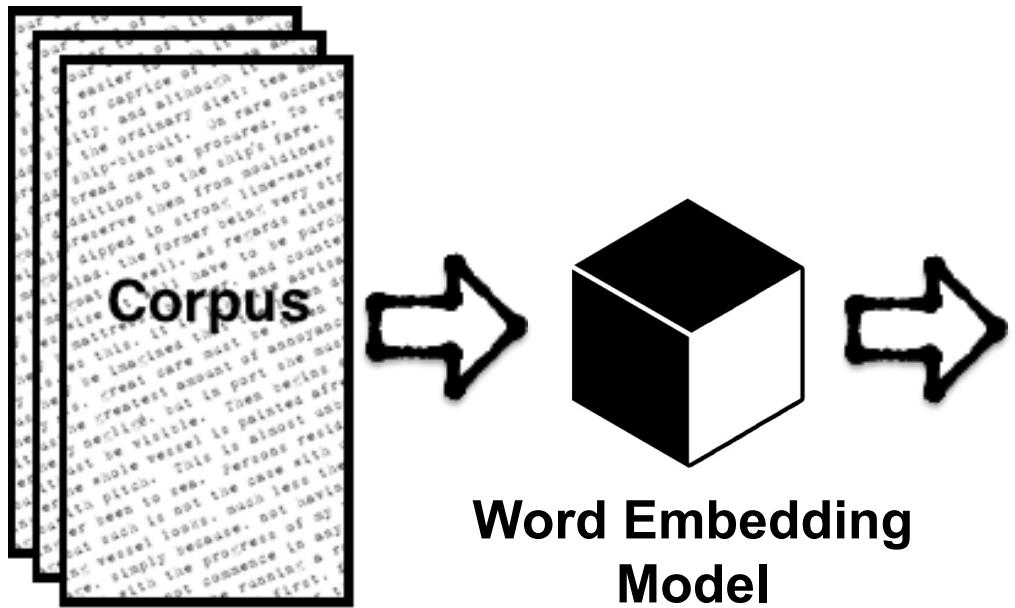


Tesgüino ←→ **Märzen**



Two words are **related** when they have many common context-words

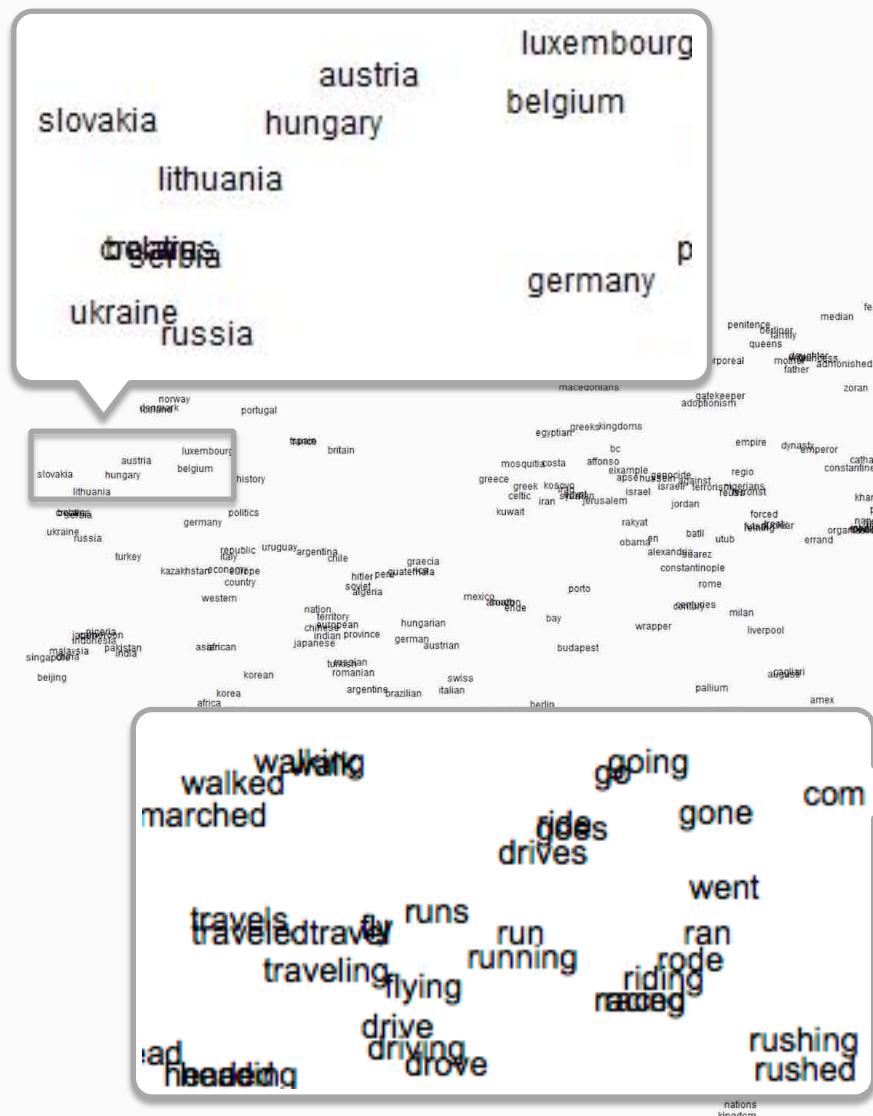
Word Embedding



- We use many contexts of words in the corpus to create vector representations of words
- When vector representations are dense, they are often called **embedding** e.g. word embedding

Word Embedding – Visualization

T-SNE



Word embeddings projected to a two-dimensional space

Word Embedding – Nearest neighbors

frog

frogs

toad

litoria

leptodactylidae

rana



book

books

foreword

author

published

preface

asthma

bronchitis

allergy

allergies

arthritis

diabetes

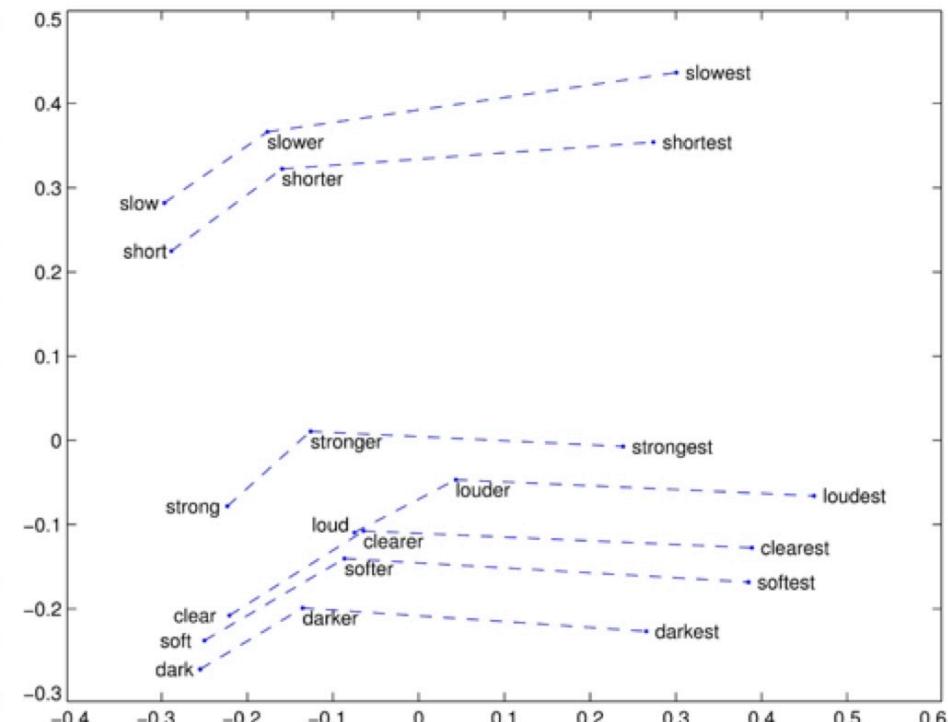
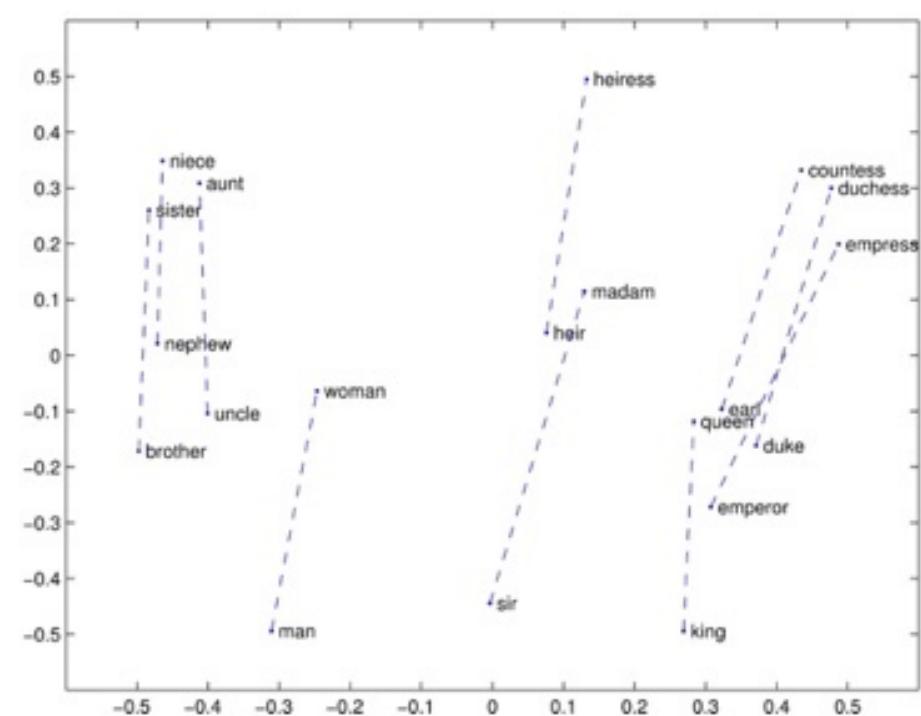
Word Embedding – Linear substructures

■ Analogy task:

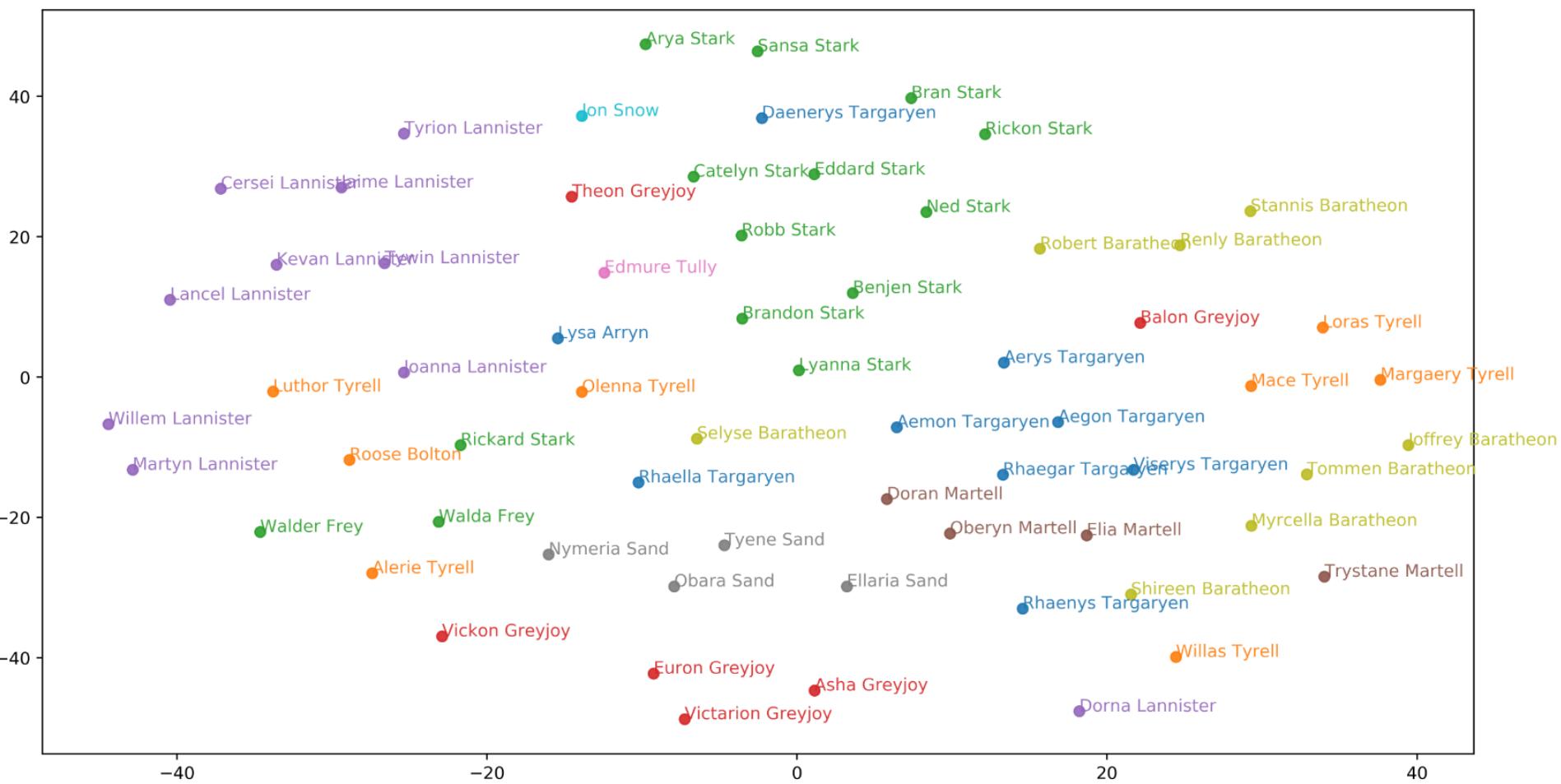
- *man* to *woman* is like *king* to ? (*queen*)

$$e_{\text{king}} - e_{\text{man}} + e_{\text{woman}} = e^*$$

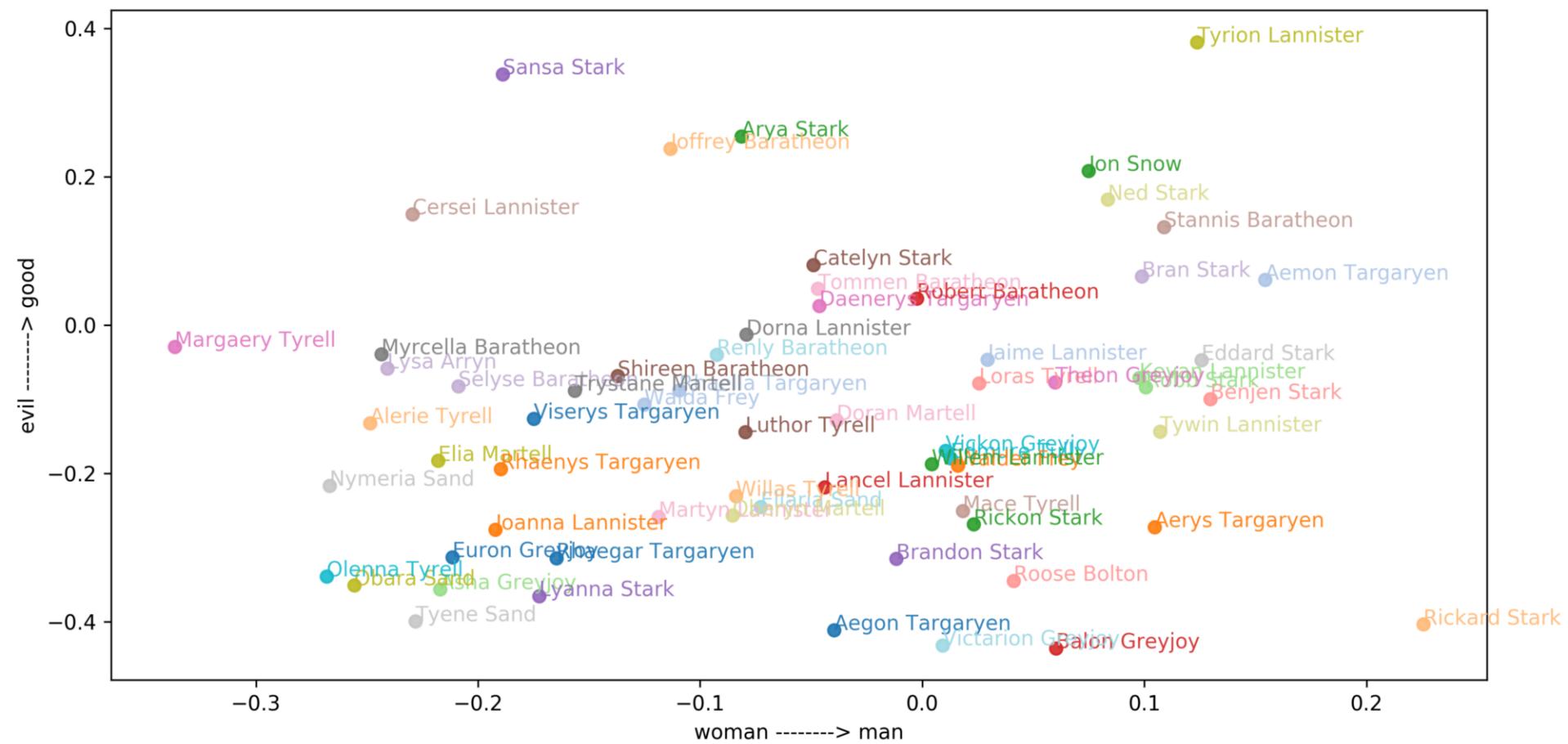
$$e^* \approx e_{\text{queen}}$$



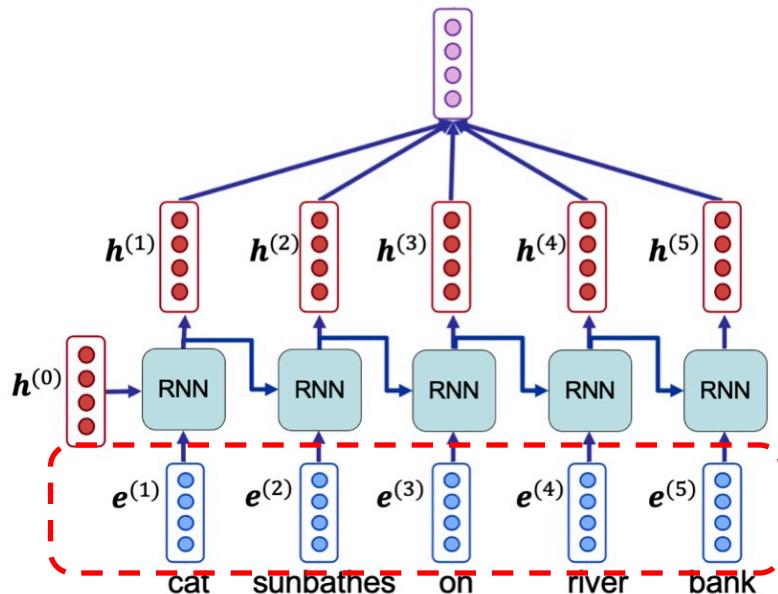
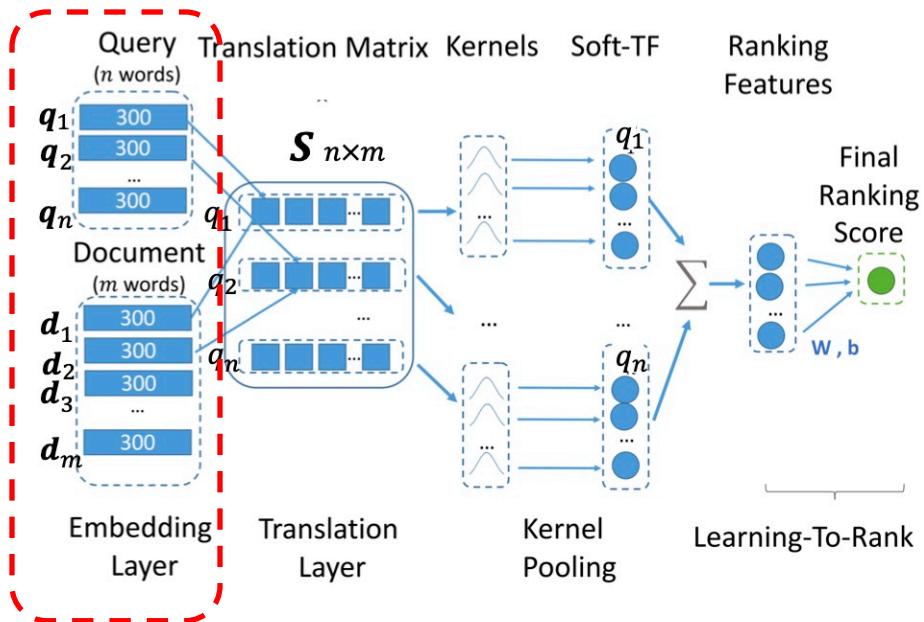
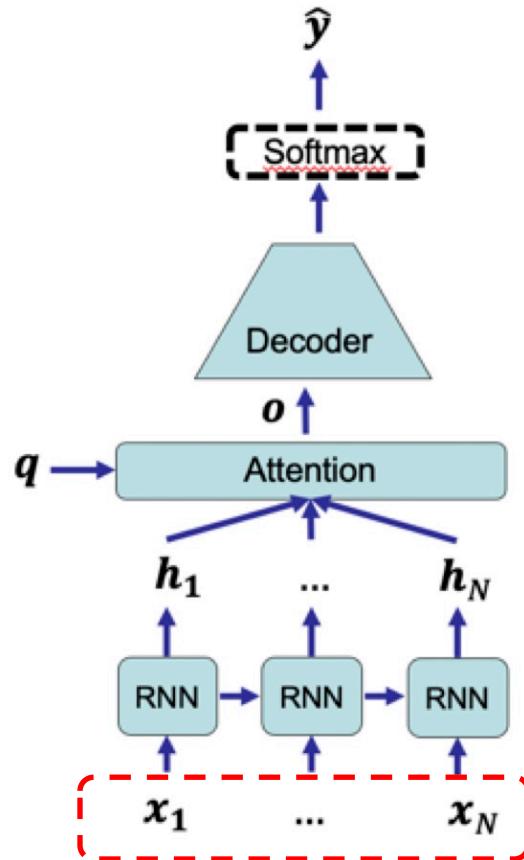
Word Embedding – Games of Thrones!



Word Embedding – Games of Thrones!



Word Embedding – Building Block of Modern NLP



Agenda

- Distributional semantics & word embedding
- **PMI+SVD**
- GloVe

Word-Document Matrix – recap

- \mathbb{D} is a set of documents (plays of Shakespeare)
$$\mathbb{D} = [d_1, d_2, \dots, d_M]$$
- \mathbb{V} is the set of words (vocabularies) in dictionary
$$\mathbb{V} = [v_1, v_2, \dots, v_N]$$
- Words as rows and documents as columns
- Values: number of occurrences of words in documents
- Matrix size $N \times M$

	d_1 As You Like It	d_2 Twelfth Night	d_3 Julius Caesar	d_4 Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0
...

Cosine

- Cosine is the normalized dot product of two vectors
 - between -1 and +1

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \cdot \frac{\mathbf{y}^T}{\|\mathbf{y}\|_2} = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}}$$

- $\mathbf{x} = [1 \quad 1 \quad 0] \quad \mathbf{y} = [4 \quad 5 \quad 6]$

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{1 * 4 + 1 * 5 + 0 * 6}{\sqrt{1^2 + 1^2 + 0^2} \sqrt{4^2 + 5^2 + 6^2}} = \frac{9}{\sim 12.4}$$

Word-Document Matrix

	$d1$	$d2$	$d3$	$d4$
	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0
...

- Similarity between two words:

$$\text{similarity}(\text{soldier}, \text{clown}) = \cos(\mathbf{e}_{\text{soldier}}, \mathbf{e}_{\text{clown}})$$

Context

- Context can be
 - Document
 - Paragraph, tweet
 - **Window of (usually 2 to 10) context words on each side of the word**
- Word-Context matrix
 - Every context-word as a dimension
$$\mathbb{C} = [c_1, c_2, \dots, c_L]$$
 - Usually $\mathbb{C} = \mathbb{V}$ and therefore $L = N$
 - Word-Context matrix is in size $N \times L$ or mostly $N \times N$

Word-Context Matrix

- Window context of 7 words
 - Values are the number of co-occurrences of v and $c \rightarrow x_{v,c}$

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

	$c1$	$c2$	$c3$	$c4$	$c5$	$c6$
	aardvark	digital	data	fruit	result	sugar
$v1$ apricot	0	0	0	4	0	2
$v2$ pineapple	0	0	0	1	0	1
$v3$ computer	0	2	5	0	1	0
$v4$ information	0	4	3	0	2	0

Word-to-Word Relations

	$c1$	$c2$	$c3$	$c4$	$c5$	$c6$
	aardvark	digital	data	fruit	result	sugar
v_1 apricot	0	0	0	4	0	2
v_2 pineapple	0	0	0	1	0	1
v_3 computer	0	2	5	0	1	0
v_4 information	0	4	3	0	2	0

- First-order co-occurrence relation
 - Words that appear **close-by** in contexts
 - Each cell $x_{v,c}$ of word-context matrix (**computer** to **data** in the example)
 - Examples of strong first-order relations: **drink** to **rosé** or **drink** to **wine**
- Second-order co-occurrence relation
 - Words that **share similar contexts**
 - Reflects the semantic similarity between words, and commonly measured by cosine (**computer** to **information** in the example)
 - Examples of strong second-order relations: **rosé** to **wine**, **Tesgüino** to **Märzen**, or **frog** to **toad**

Point Mutual Information (PMI)

- Problem with using raw co-occurrence statistics ($x_{v,c}$) in word-context matrix
 - Highly frequent words (“**and**”, “**the**”) co-occur with many words and gain high values, although they don’t convey much information
- Point Mutual Information (PMI)
 - Rooted in information theory
 - A common measure of **first-order co-occurrence** used to create word-context matrix
 - Defined as the joint probability of two events (random variables) divided by their marginal probabilities

$$\text{PMI}(A, B) = \log_2 \frac{P(A, B)}{P(A)P(B)}$$

Point Mutual Information (PMI)

- PMI of the occurrences of word v and context-word c

$$\text{PMI}(v, c) = \log_2 \frac{P(v, c)}{P(v)P(c)}$$

$P(v, c)$: probability of co-occurrence of v with c

$$P(v, c) = \frac{x_{v,c}}{|\mathcal{D}|}$$

$x_{v,c}$: number of times that v and c appear in the same context

$|\mathcal{D}|$: number of **all** co-occurrences in the corpus $\rightarrow |\mathcal{D}| = \sum_{i=1}^N \sum_{j=1}^L x_{v,c}$

$P(v)$: probability of co-occurrence of v with **any context-word**

$$P(v) = \frac{\sum_{\check{c} \in \mathbb{C}} x_{v,\check{c}}}{|\mathcal{D}|} = \frac{X_v}{|\mathcal{D}|}$$

$P(c)$: probability of co-occurrence of c with **any word**

$$P(c) = \frac{\sum_{\check{v} \in \mathbb{V}} x_{\check{v},c}}{|\mathcal{D}|} = \frac{X_c}{|\mathcal{D}|}$$

Point Mutual Information (PMI)

Word-context matrix with raw co-occurrences ($x_{v,c}$)

	$c1$ aardvark	$c2$ digital	$c3$ data	$c4$ fruit	$c5$ result	$c6$ sugar
$v1$ apricot	0	0	0	3	0	2
$v2$ pineapple	0	0	0	2	0	1
$v3$ computer	0	2	5	0	1	0
$v4$ information	0	4	3	0	2	0

$$P(v = \text{information}, c = \text{data}) = 3/25 = 0.12$$

$$P(v = \text{information}) = 9/25 = 0.36$$

$$P(c = \text{data}) = 8/25 = 0.32$$

$$\text{PMI}(v = \text{information}, c = \text{data}) = \log_2 \frac{0.12}{0.36 \times 0.32} = 0.058$$

Point Mutual Information (PMI)

Word-context matrix with raw co-occurrences ($x_{v,c}$)

	$c1$ aardvark	$c2$ digital	$c3$ data	$c4$ fruit	$c5$ result	$c6$ sugar
$v1$ apricot	0	0	0	3	0	2
$v2$ pineapple	0	0	0	2	0	1
$v3$ computer	0	2	5	0	1	0
$v4$ information	0	4	3	0	2	0

Word-context matrix with PMI

	$c1$ aardvark	$c2$ digital	$c3$ data	$c4$ fruit	$c5$ result	$c6$ sugar
$v1$ apricot	$-\infty$	$-\infty$	$-\infty$	1.58	$-\infty$	1.73
$v2$ pineapple	$-\infty$	$-\infty$	$-\infty$	1.73	$-\infty$	1.47
$v3$ computer	$-\infty$	0.05	-2.35	$-\infty$	0.05	$-\infty$
$v4$ information	$-\infty$	0.88	0.05	$-\infty$	0.88	$-\infty$

Positive Point Mutual Information (PPMI)

- PPMI sets negative values of PMI to zero

$$\text{PPMI}(t, c) = \max(\text{PMI}, 0)$$

Word-context matrix with PMI

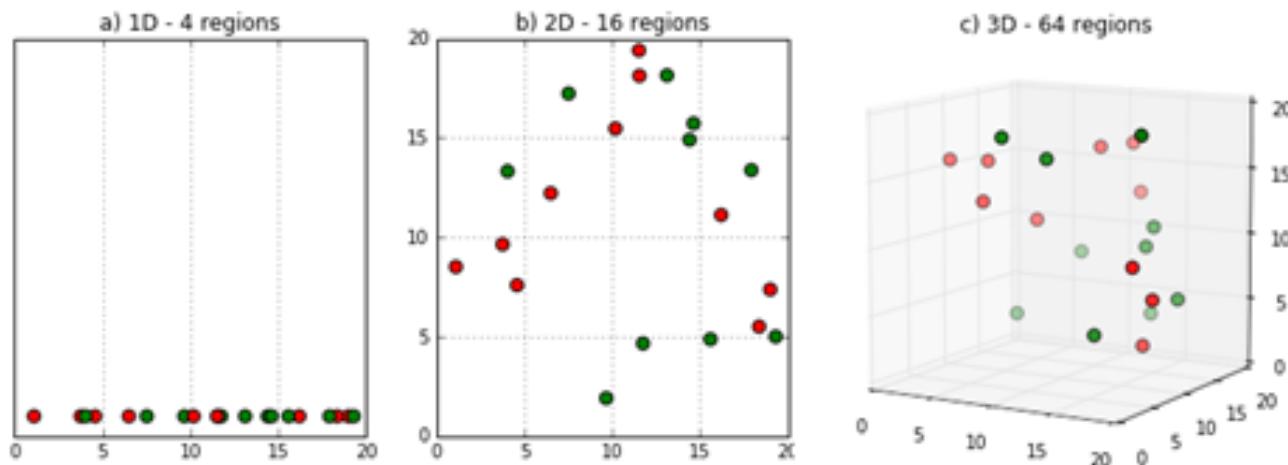
	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>
	aardvark	digital	data	fruit	result	sugar
<i>v1</i> apricot	−∞	−∞	−∞	1.58	−∞	1.73
<i>v2</i> pineapple	−∞	−∞	−∞	1.73	−∞	1.47
<i>v3</i> computer	−∞	0.05	−2.35	−∞	0.05	−∞
<i>v4</i> information	−∞	0.88	0.05	−∞	0.88	−∞

Word-context matrix with PPMI

	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>
	aardvark	digital	data	fruit	result	sugar
<i>v1</i> apricot	0	0	0	1.58	0	1.73
<i>v2</i> pineapple	0	0	0	1.73	0	1.47
<i>v3</i> computer	0	0.05	0	0	0.05	0
<i>v4</i> information	0	0.88	0.05	0	0.88	0

Curse of dimensionality – recap

- Why low-dimensional vectors?
 - Easier to store and load
 - More efficient when used as features in ML models
 - Usually better generalization due to the reduction of noise in data
 - Able to capture higher-order relations:
 - Synonyms like *car* and *automobile* can be merged into the same dimensions
 - Polysomy like *bank* (*financial institution*) and *bank* (*bank of river*) can be separated into different dimensions



Singular Value Decomposition – recap

- An $N \times M$ matrix X can be factorized to three matrices:

$$X = U\Sigma V^T$$

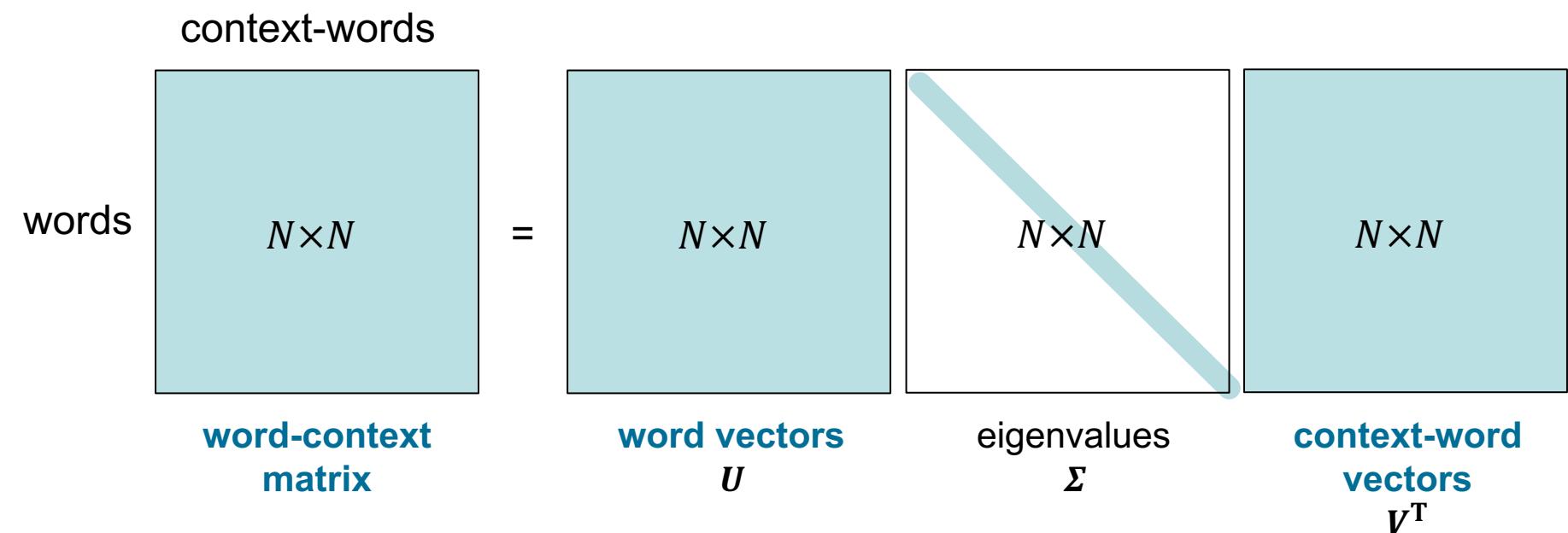
- U (**left singular vectors**) is an $N \times M$ unitary matrix
- Σ is an $M \times M$ diagonal matrix, diagonal entries
 - are **eigenvalues**,
 - show the importance of corresponding M dimensions in X
 - are all **positive** and sorted from large to small values
- V^T (**right singular vectors**) is an $M \times M$ unitary matrix

Singular Value Decomposition – recap

$$\begin{matrix} N \times L \\ \text{original matrix} \\ X \end{matrix} = \begin{matrix} N \times L \\ \text{left singular vectors} \\ U \end{matrix} \begin{matrix} L \times L \\ \text{eigenvalues} \\ \Sigma \end{matrix} \begin{matrix} L \times L \\ \text{right singular vectors} \\ V^T \end{matrix}$$

Applying SVD to PPMI word-context matrix

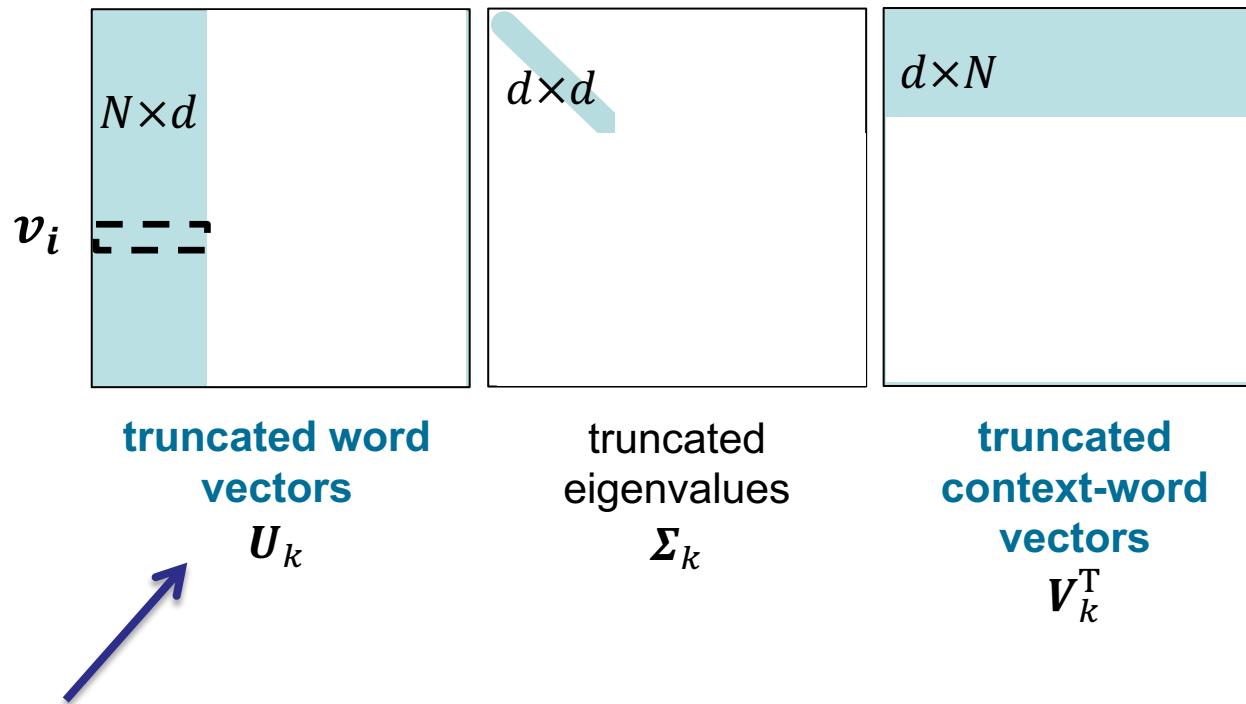
- Step 1: create a PPMI matrix of the size $N \times N$,
- Apply SVD



Applying SVD to PPMI word-context matrix

- Step 2: keep only top d eigenvalues in Σ and set the rest to zero
- Truncate the U and V^T matrices, resulting in U_k and V_k^T
- Rows in U_k are the new low-dimensional word vectors

Applying SVD to Term-Context Matrix



- U_k is the matrix of dense low-dimensional word vectors



Summary – PPMI+SVD

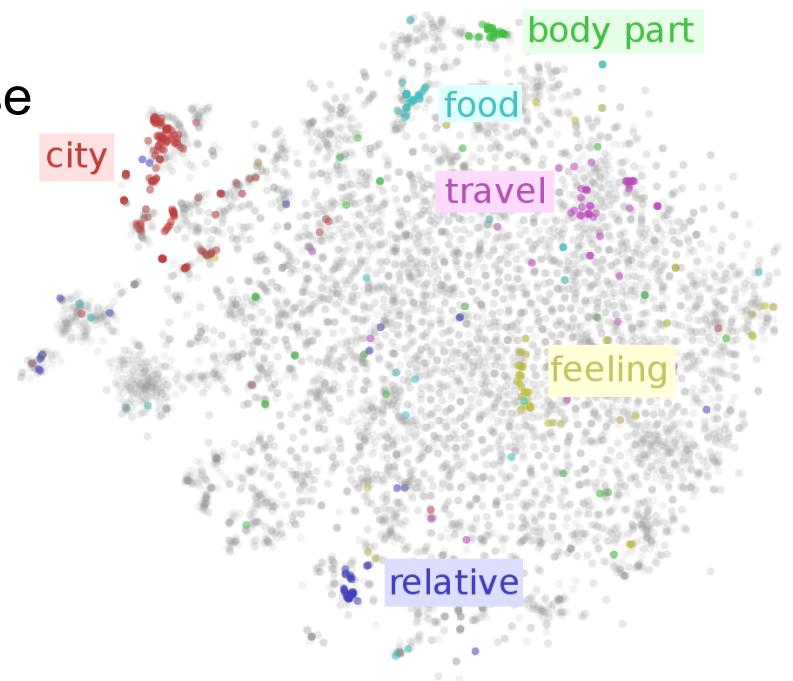
- PPMI captures first-order co-occurrences
- PPMI word-context matrix provides high-dimensional word representations. The matrix is ...
 - **highly sparse**
 - **very big**, and barely fits to memory. It usually requires sparse data structures
- As shown empirically, PMI overly favors low-frequency/rare words
- By applying SVD to PPMI word-context matrix, we achieve dense low-dimensional word embedding
 - Appropriate for calculating word-to-word semantic similarity (using cosine)
 - Computing SVD is time consuming. Usually approximations of SVD are used

Agenda

- Distributional semantics & word embedding
- PMI+SVD
- **GloVe**

GloVe: Global Vectors for Word Representations

- A well-known word embedding model
 - Pre-trained word embedding are typically used in various tasks
- Similar to PPMI+SVD, GloVe ...
 - starts from a high-dimensional sparse word-context co-occurrence matrix
 - and then applies matrix factorization to build low-dimensional word embeddings



Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. 2014.

Picture: <https://ruder.io/word-embeddings-1/>

Word-Context matrix – from PMI to GloVe

- Deriving GloVe from PMI ...
- Recall:

$$\begin{aligned}\text{PMI}(v, c) &= \log_2 \frac{P(v, c)}{P(v)P(c)} \approx \log \frac{P(v, c)}{P(v)P(c)} \\ &= \log \frac{x_{v,c}/|\mathcal{D}|}{X_v/|\mathcal{D}| \ X_c/|\mathcal{D}|} = \log \frac{x_{v,c}|\mathcal{D}|}{X_v X_c} \\ &= \log x_{v,c} - \log X_v - \log X_c + \log |\mathcal{D}|\end{aligned}$$

- $\log |\mathcal{D}|$ is a constant and can be removed:

$$\approx \log x_{v,c} - \log X_v - \log X_c$$

Word-Context matrix – from PMI to GloVe

- Deriving GloVe from PMI ...

$$\text{PMI}(v, c) \approx \log x_{v,c} - \log X_v - \log X_c$$

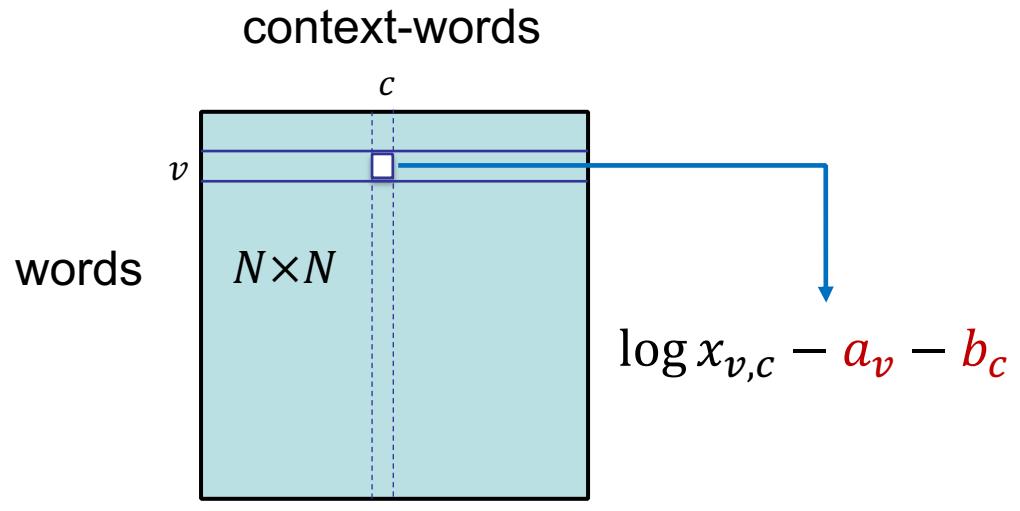
- **1st difference:** replace $\log X_v$ and $\log X_c$ with learning parameters a_v and b_c , respectively
- GloVe therefore uses:

$$\log x_{v,c} - a_v - b_c$$

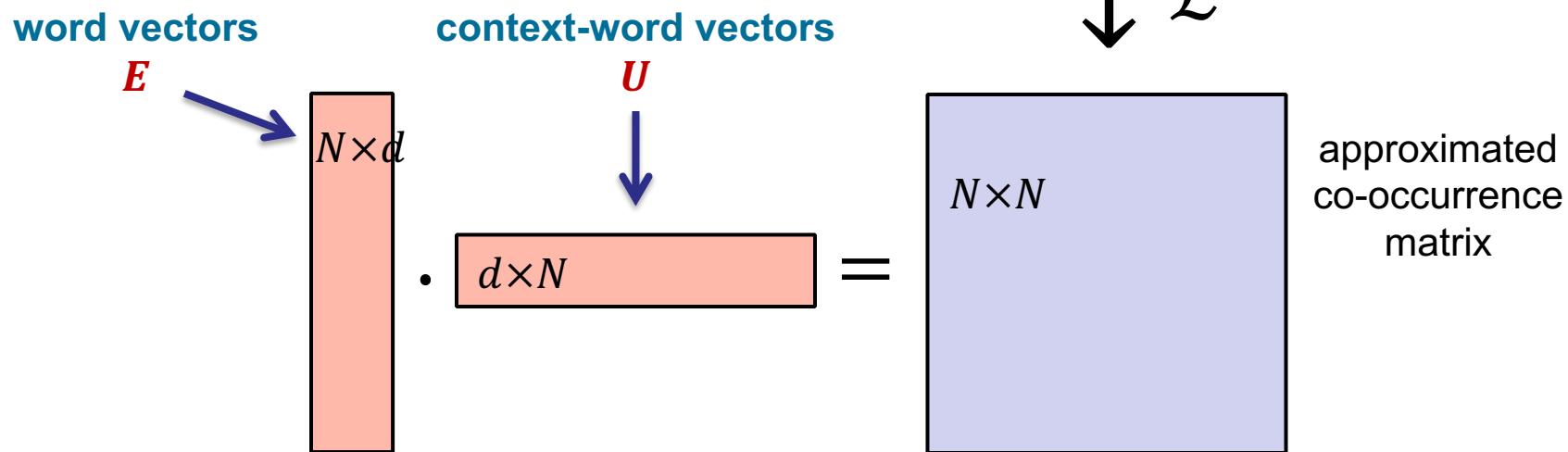
- a_v and b_c are **bias terms**
- In fact, a_v and b_c act as a “**normalization**” to log-co-occurrence, where their degrees are learned
- Vector \mathbf{a} contain the bias values of all words (like a_v)
- Vector \mathbf{b} contain the bias values of all context-words (like b_c)

Matrix factorization

GloVe word-context matrix:



2nd difference: matrix factorization



Matrix factorization

Formal definition:

We factorize GloVe co-occurrence matrix by defining two learnable parameter matrices of size $N \times d$:

- \mathbf{E} (word embeddings)
- \mathbf{U} (context-word embeddings)

such that for every (v, c) pair:

$$\mathbf{e}_v \mathbf{u}_c^T \approx \log x_{v,c} - a_v - b_c$$

\mathbf{E} , \mathbf{U} , a , and b are (learnable) model parameters

Loss function

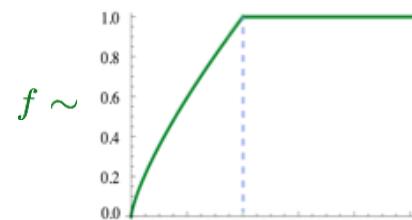
- To find optimum values for parameters, we need to define a loss function
- GloVe uses least squares loss function*:

$$\mathcal{L} = \sum_{v \in V, c \in C} (\mathbf{e}_v \mathbf{u}_c^T + a_v + b_c - \log x_{v,c})^2$$

- Using the loss function, model parameters are optimized with (Stochastic) Gradient Descent
 - SGD is discussed in the next lecture

* GloVe includes a function f to the loss, which weighs down the rare pairs and saturates on highly frequent ones. The loss function in GloVe is weighted least squares (details in paper):

$$\mathcal{L} = \sum_{v \in V, c \in C} f(x_{v,c})(\mathbf{e}_v \mathbf{u}_c^T + a_v + b_c - \log x_{v,c})^2$$



Summary – GloVe

- GloVe
 - first accumulates co-occurrence numbers
 - then defines two embedding matrices and two bias vectors, and optimize their values by approximating the logarithm of co-occurrence statistics
- Final word embeddings
 - can be the word vectors E
 - but also can be the sum of the word and context-word vectors: $E + U^T$
 - This method is used in GloVe
- Characteristics
 - Fast training (no need for SVD!) → repeat SGD till loss converges
 - Scalable to large corpora
 - Good performance even with small corpus and small vectors