Winter semester 2022/23

344.175 VL: Natural Language Processing
Word Embedding with Matrix Factorization

Navid Rekab-saz

Email: navid.rekabsaz@jku.at
Office hours: https://navid-officehours.youcanbook.me

JOHANNES KEPLER UNIVERSITY LINZ

Institute of Computational Perception

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)

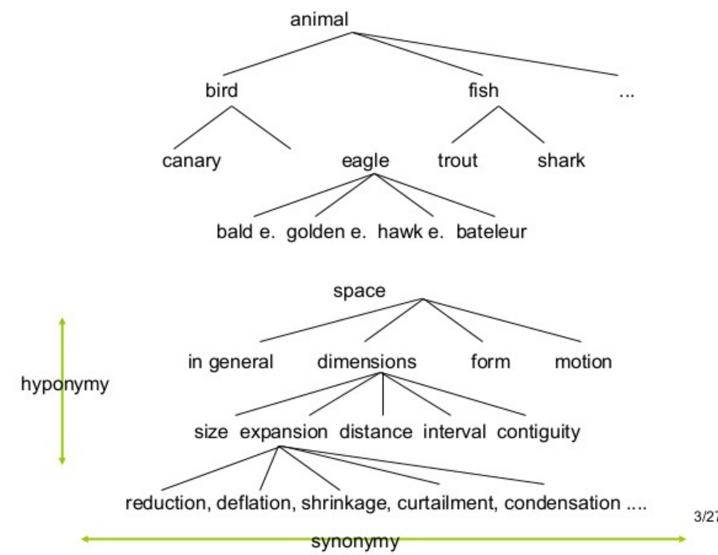
Semantics:

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

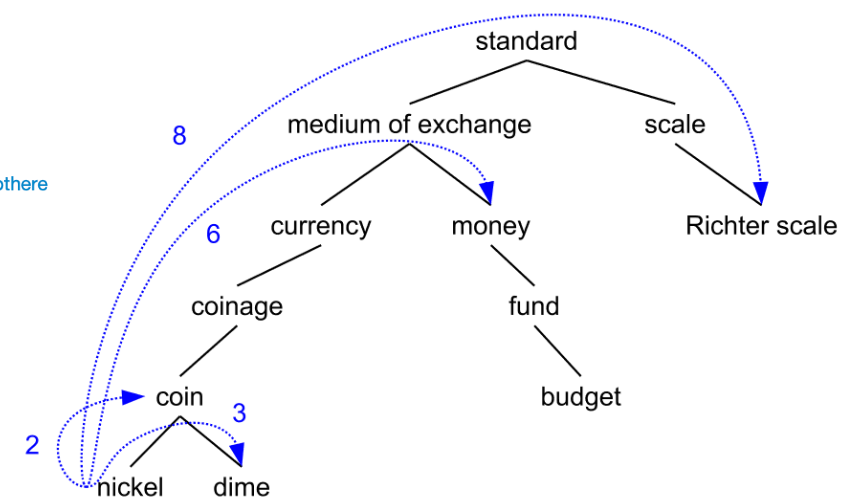
Noun language has 6 senses

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.freedicts.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!

http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture01-wordvecs1.pdf drink

sacred

beer

Tesgüino

ritual

corn

fermented

Mexico

Tarahumara people

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]

drink

sacred

beer

Tesgüino

ritual

corn

brew

Mexico

Tarahumara people

dark brown

bottle

malt

brew

Märzen

lager

Bavaria

bar

drink

beer

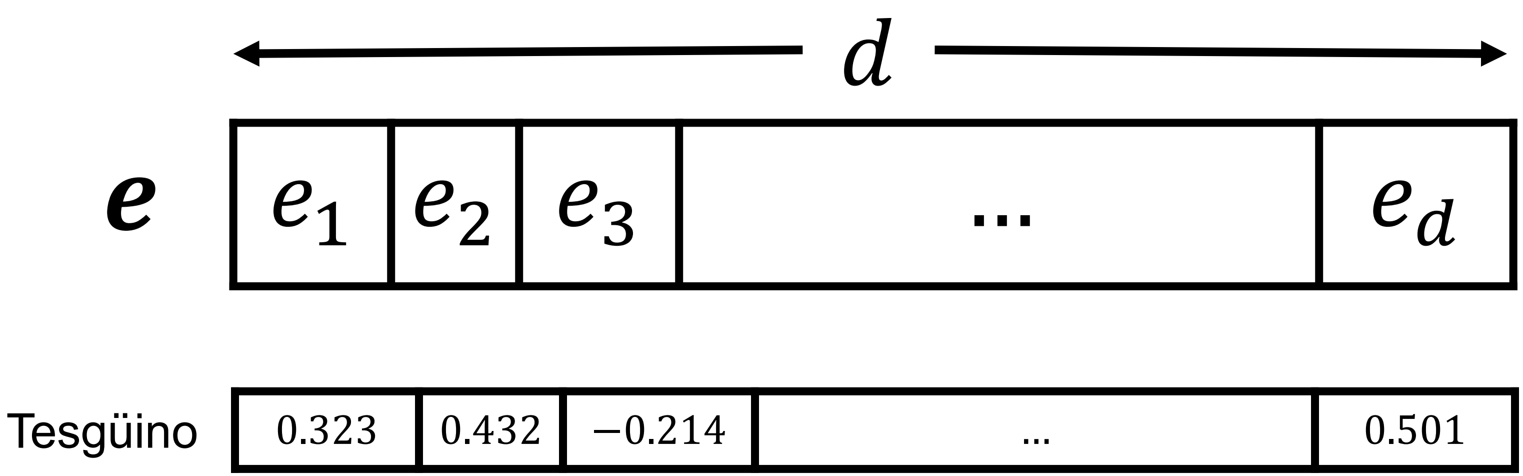
Tesgüino ⟷ Märzen

An algorithmic intuition:

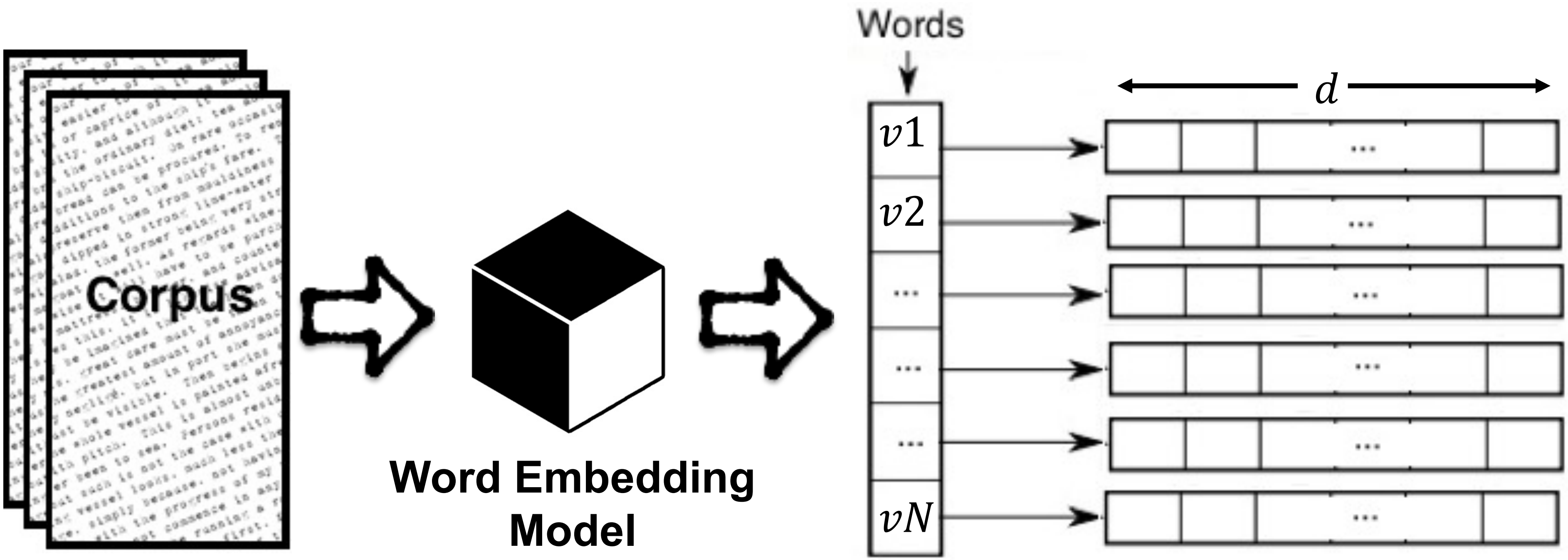
Two words are related (semantically similar) when they have many common context-words

Distributional Representation

- A word (token) is represented with a vector of d dimensions
- Each dimension can be seen as a "feature" of the entity (word/token)
- Dimensions and their values are not mutually exclusive
  - Two units can be "active" at the same time
- Each dimension forms a distribution over possible values (domain $\mathbb{R}$)
  - Realizing a word vector can be seen as, for each dimension, selecting a value according to the its underlying distribution

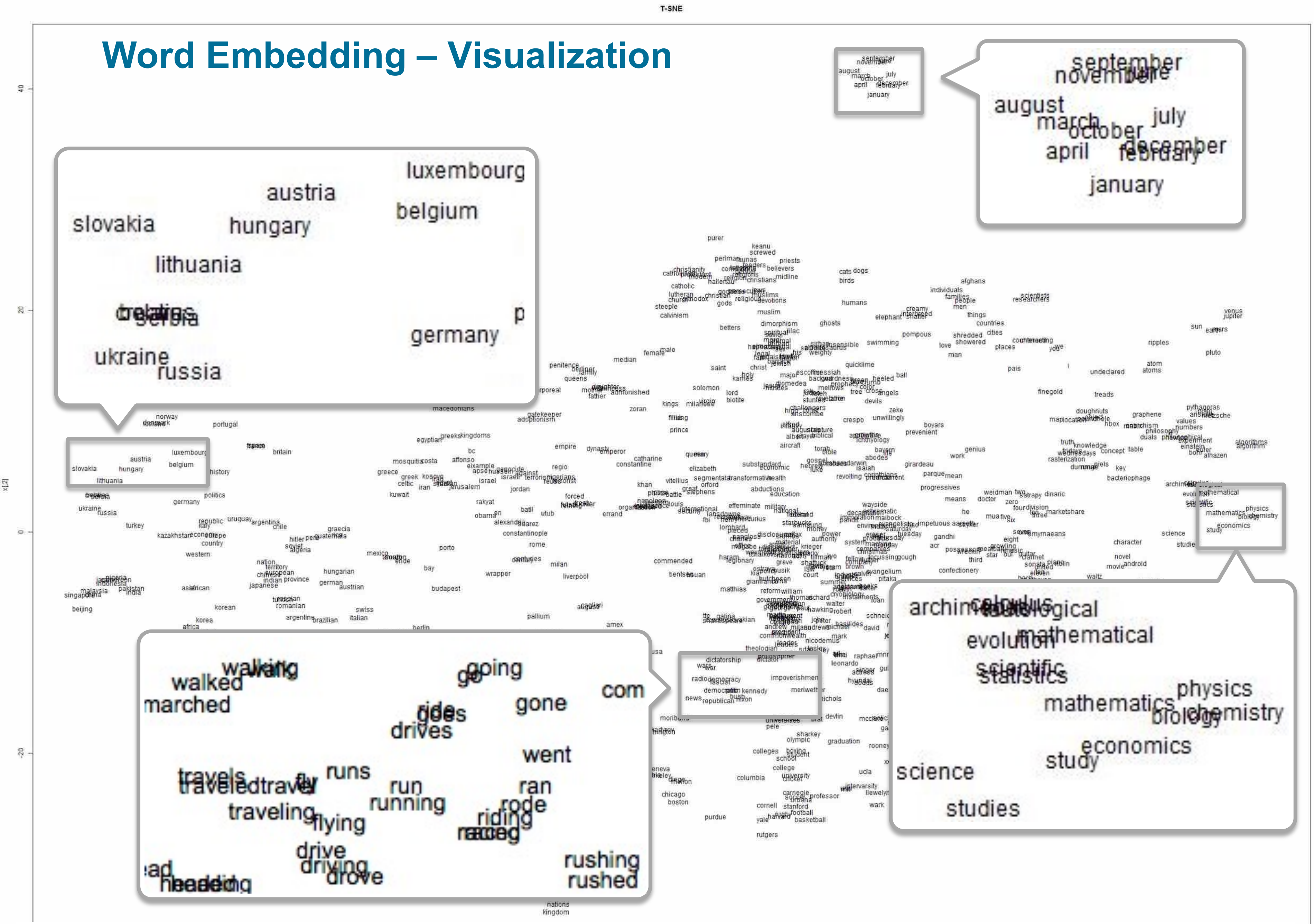


Word Embedding



- 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



Word embeddings projected to a two-dimensional space

Word Embedding – Nearest neighbors

frog	book	asthma
frogs	books	bronchitis
toad	foreword	allergy
litoria	author	allergies
leptodactylidae	published	arthritis
rana	preface	diabetes

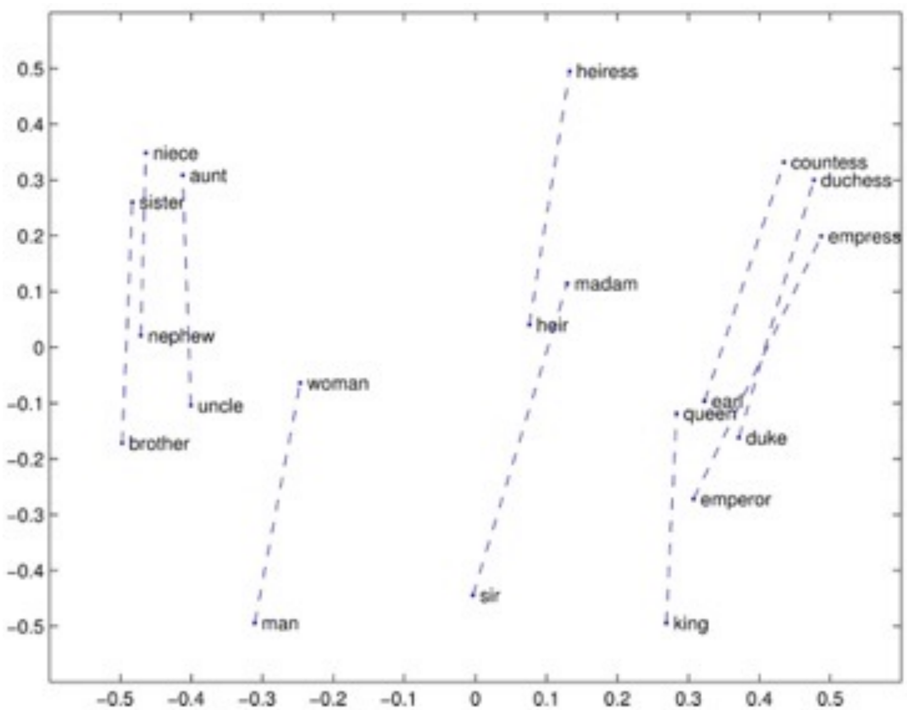


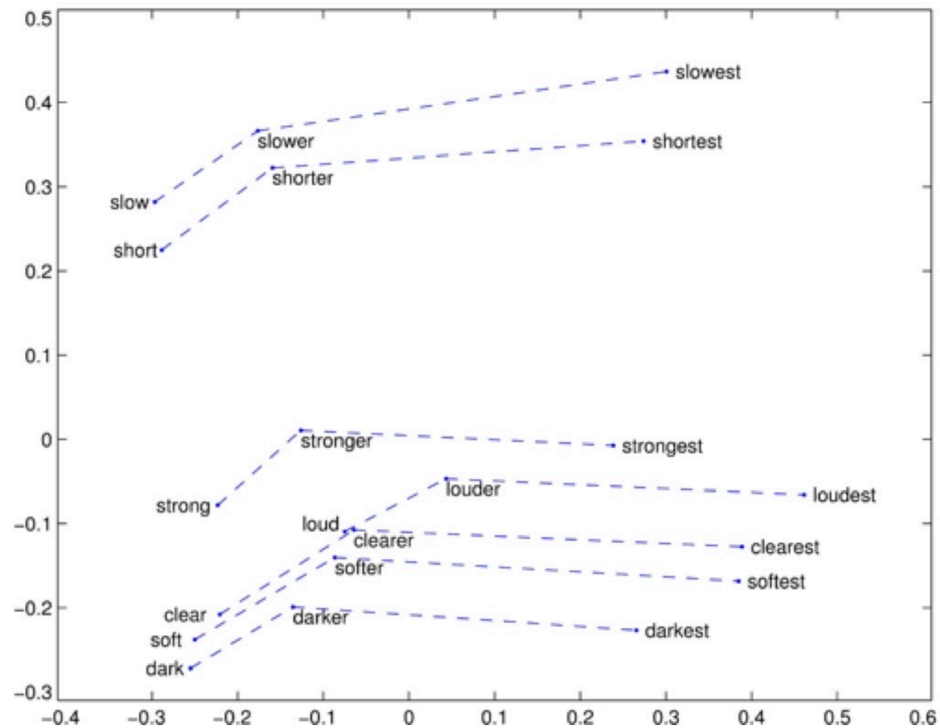
https://nlp.stanford.edu/projects/glove/

Word Embedding – Linear substructures

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

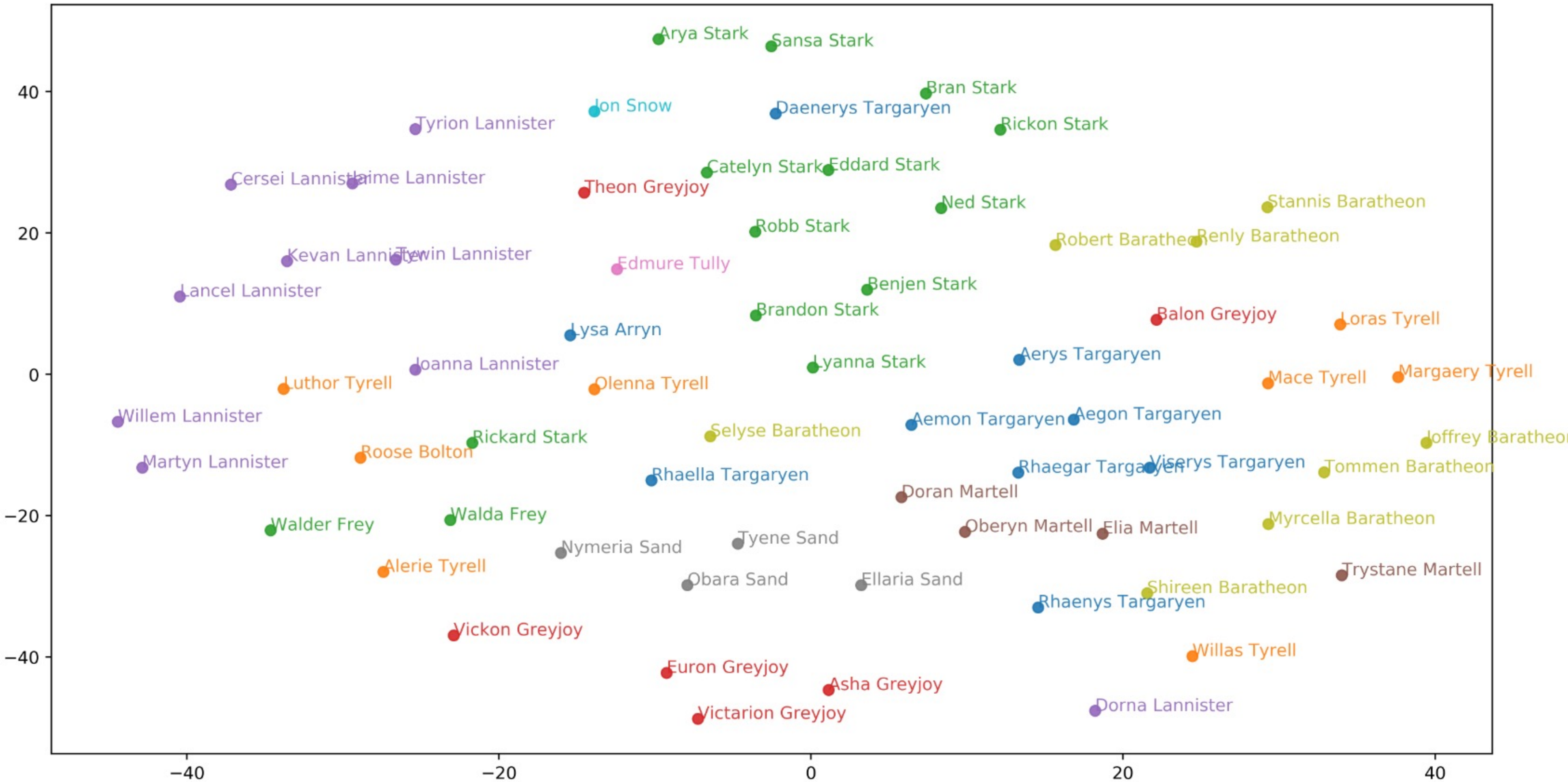
$$\boldsymbol{e}_{\text{king}} - \boldsymbol{e}_{\text{man}} + \boldsymbol{e}_{\text{woman}} = \boldsymbol{e}^*$$
$$\boldsymbol{e}^* \approx \boldsymbol{e}_{\text{que}en}$$





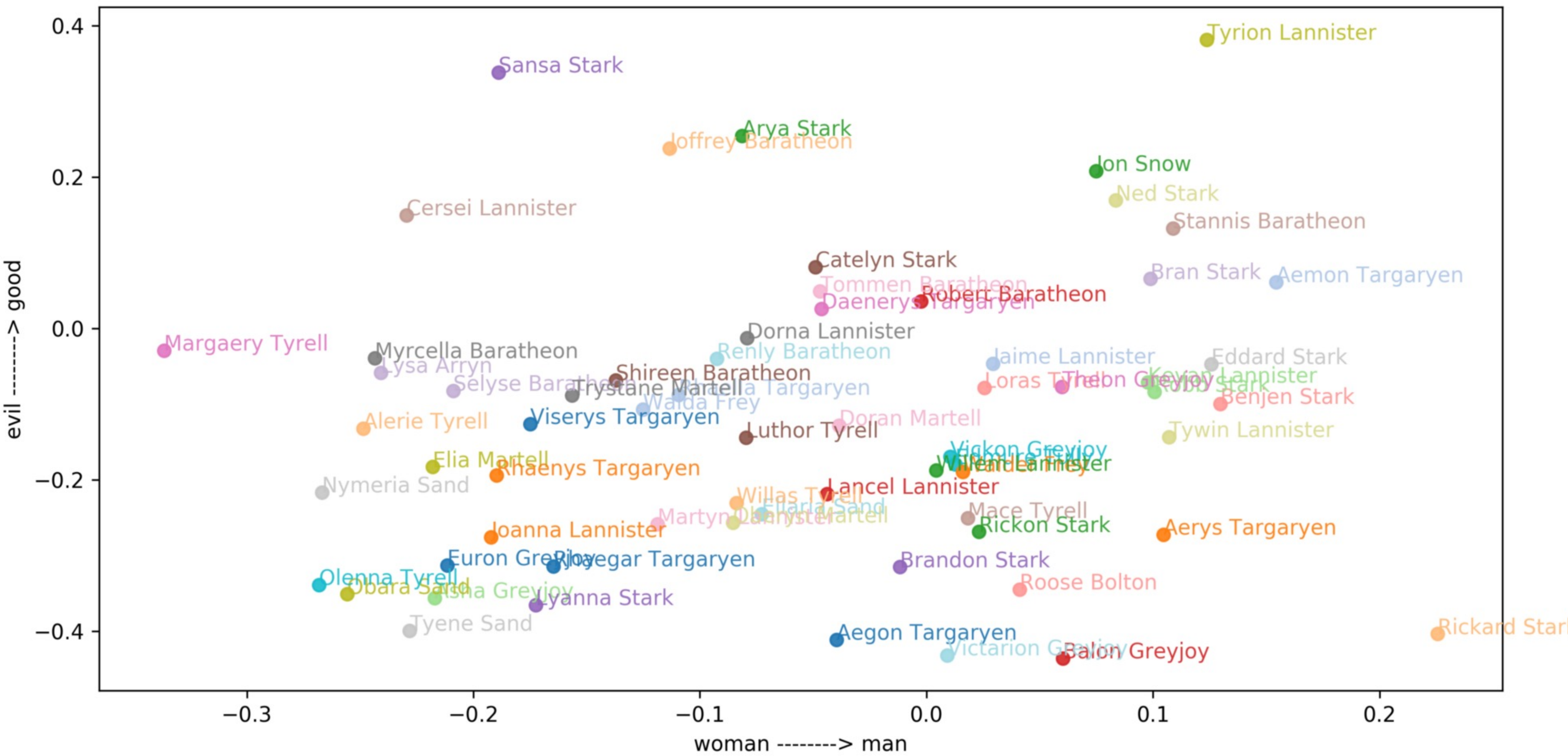
https://nlp.stanford.edu/projects/glove/

Word Embedding – Games of Thrones!



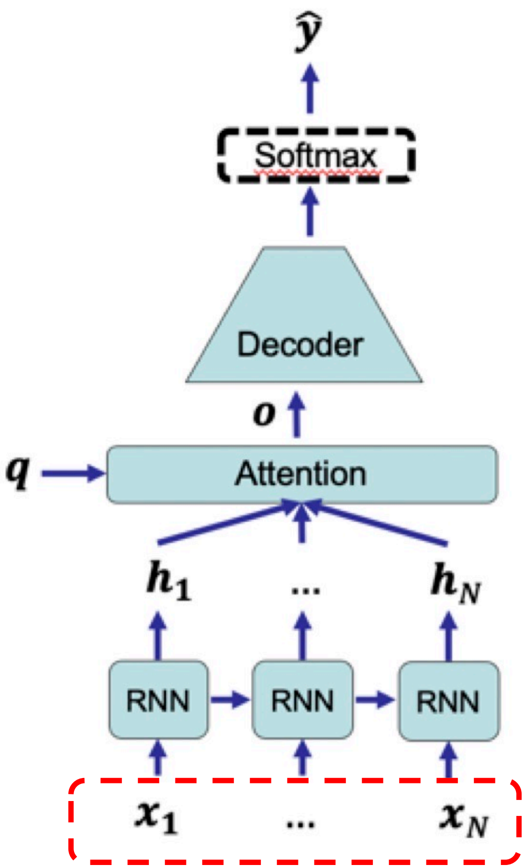
https://towardsdatascience.com/game-of-thrones-word-embeddings-does-r-l-j-part-2-30290b1c0b4b

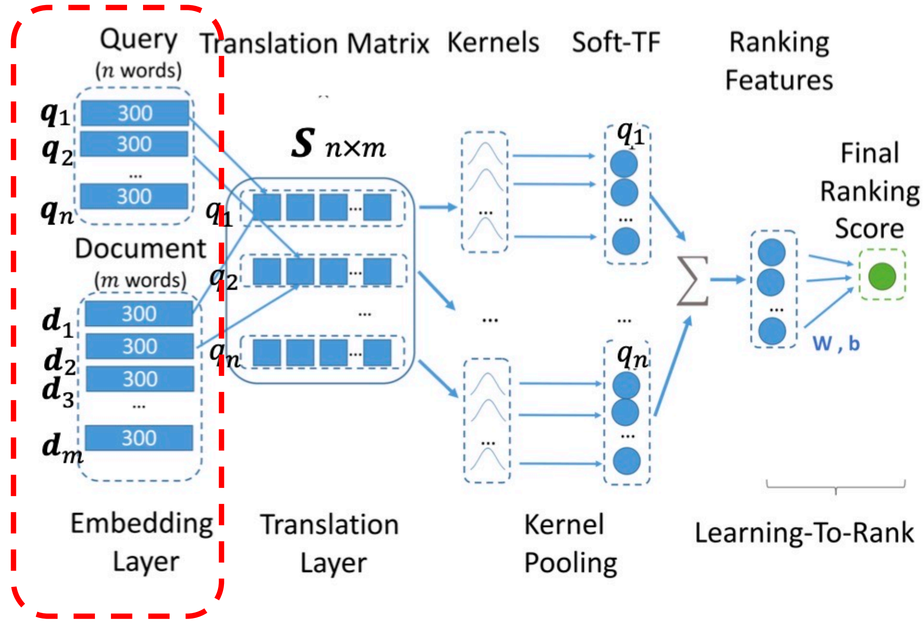
Word Embedding – Games of Thrones!

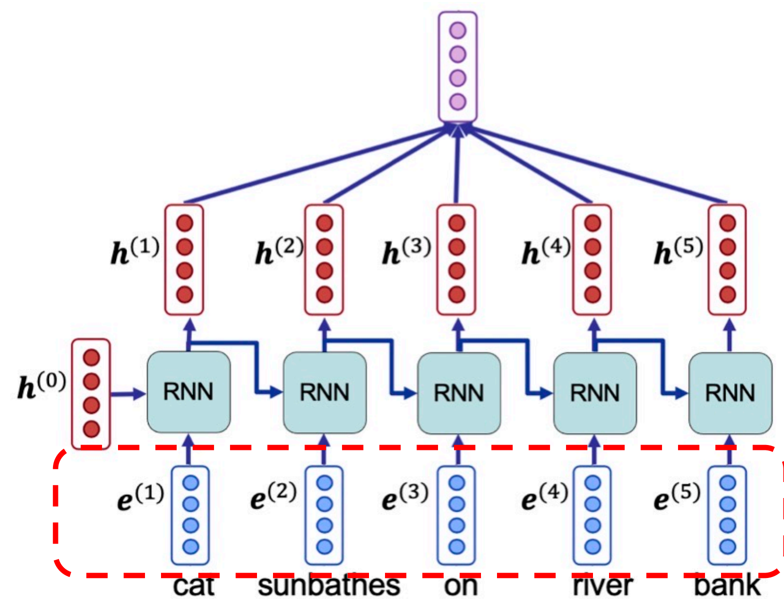


https://towardsdatascience.com/game-of-thrones-word-embeddings-does-r-l-j-part-2-30290b1c0b4b

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)

$$\mathbb{D} = [d1, d2, \dots, dM]$$

- $\mathbb{V}$ is the set of words (vocabularies) in dictionary

$$\mathbb{V} = [v1, v2, \dots, vN]$$

- Words as rows and documents as columns
- Values: number of occurrences of words in documents
- Matrix size $N \times M$

	$d1$ As You Like It	$d2$ Twelfth Night	$d3$ Julius Caesar	$d4$ 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

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

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

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

Word-Document Matrix

	$d1$ As You Like It	$d2$ Twelfth Night	$d3$ Julius Caesar	$d4$ 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:

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

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

$$\mathbb{C} = [c1, c2, \dots, cL]$$

  - 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 $c \to x_{v,c}$

sugar, a sliced lemon, a tablespoonful of	**apricot**	preserve or jam, a pinch each of,
their enjoyment. Cautiously she sampled her first	**pineapple**	and another fruit whose taste she likened
well suited to programming on the digital	**computer**.	In finding the optimal R-stage policy from
for the purpose of gathering data and	**information**	necessary for the study authorized in the

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

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

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

Point Mutual Information (PMI)

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

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

$P(v,c)$: 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 $\rightarrow |\mathcal{D}| = \sum_{i=1}^{N} \sum_{j=1}^{L} x_{i,j}$

$P(v)$: 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	1	3	0	2
$v2$	pineapple	0	0	0	2	0	1
$v3$	computer	0	2	4	0	1	0
$v4$	information	0	4	3	0	2	0

$\mathrm{PMI}(v = \text{information}, c = \text{data}) =?$

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

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

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

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

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	1	3	0	2
$v2$	pineapple	0	0	0	2	0	1
$v3$	computer	0	2	4	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	$-\infty$	$-\infty$	-0.94	1.32	$-\infty$	1.47
$v2$	pineapple	$-\infty$	$-\infty$	$-\infty$	1.73	$-\infty$	1.47
$v3$	computer	$-\infty$	0.25	0.83	$-\infty$	0.25	$-\infty$
$v4$	information	$-\infty$	0.88	0.05	$-\infty$	0.88	$-\infty$

Positive Point Mutual Information (PPMI)

- PPMI sets negative values of PMI to zero

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

Word-context matrix with PMI

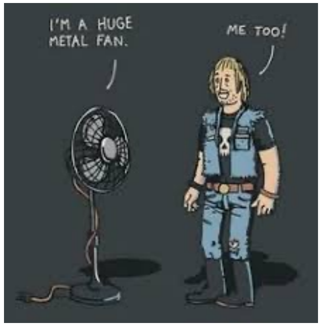
		$c1$	$c2$	$c3$	$c4$	$c5$	$c6$
		aardvark	digital	data	fruit	result	sugar
$v1$	apricot	$-\infty$	$-\infty$	-0.94	1.32	$-\infty$	1.47
$v2$	pineapple	$-\infty$	$-\infty$	$-\infty$	1.73	$-\infty$	1.47
$v3$	computer	$-\infty$	0.25	0.83	$-\infty$	0.25	$-\infty$
$v4$	information	$-\infty$	0.88	0.05	$-\infty$	0.88	$-\infty$

Word-context matrix with PPMI

		$c1$	$c2$	$c3$	$c4$	$c5$	$c6$
		aardvark	digital	data	fruit	result	sugar
$v1$	apricot	0	0	0	1.32	0	1.47
$v2$	pineapple	0	0	0	1.73	0	1.47
$v3$	computer	0	0.25	0.83	0	0.25	0
$v4$	information	$\mathbf{0}$	0.88	0.05	0	0.88	0

Why low-dimensional vectors? – Recap

- Easier to store and load
- More efficient when used as features in ML models
- Better generalization due to the reduction of noise in data
- Able to capture higher-order relations:
  - Synonyms like *car* and *automobile* might be merged into the same dimensions



  - Polysemies like *bank (financial institution)* and *bank (bank of river)* might be separated into different dimensions

Singular Value Decomposition – Recap

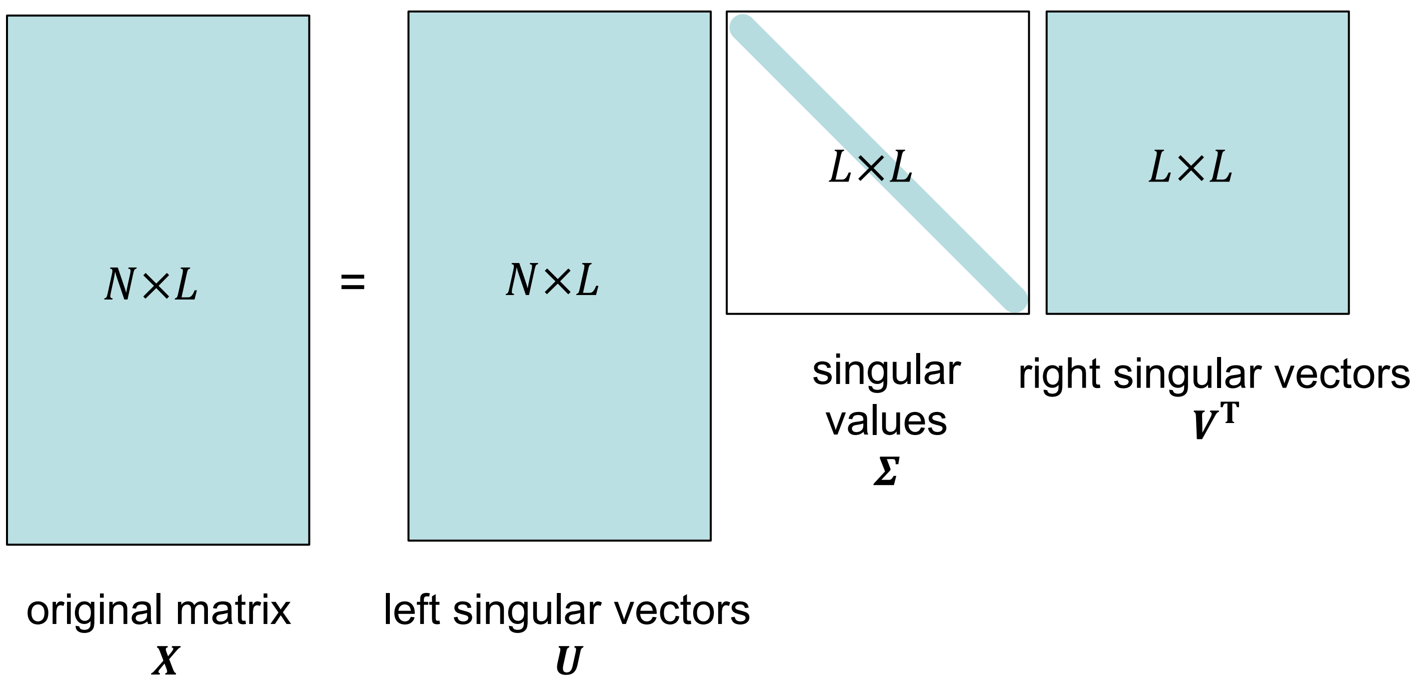
- An $N \times M$ matrix $\boldsymbol{X}$ can be factorized to three matrices:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$$

- $\boldsymbol{U}$ left singular vectors is an $N \times M$ unitary matrix
- $\boldsymbol{\Sigma}$ is an $M \times M$ diagonal matrix, diagonal entries
  - are singular values,
  - show the importance of corresponding M dimensions in $\boldsymbol{X}$
  - are all positive and sorted from large to small values
- $\boldsymbol{V}^{\mathrm{T}}$ right singular vectors is an $M \times M$ unitary matrix

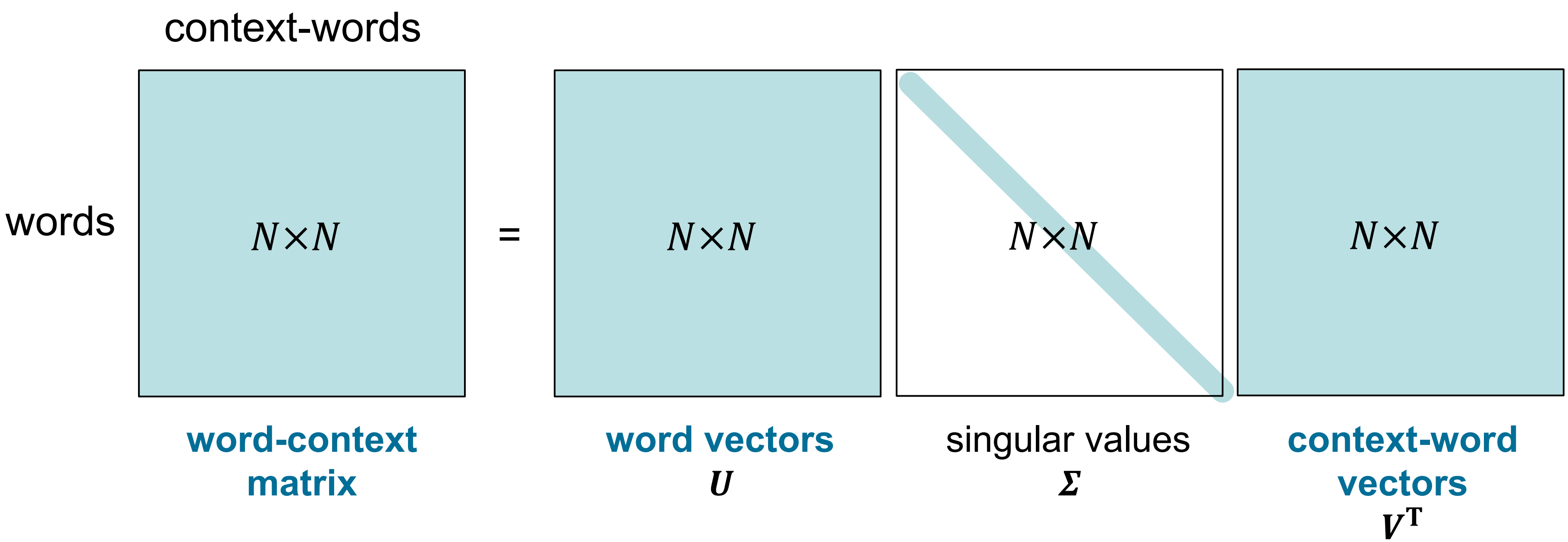
* The definition of SVD is simplified. Refer to https://en.wikipedia.org/wiki/Singular_value_decomposition for the exact definition

Singular Value Decomposition – Recap



Applying SVD to PPMI word-context matrix

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



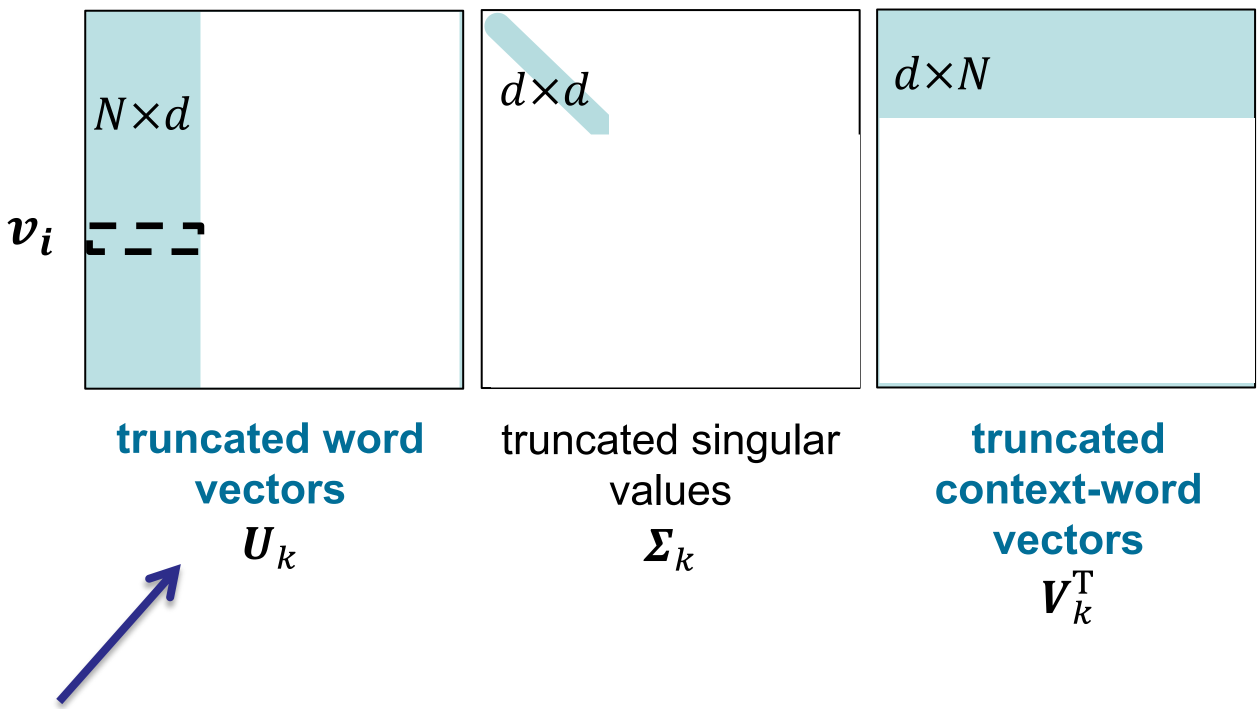
Applying SVD to PPMI word-context matrix

- Step 2: keep only top d singular values in $\boldsymbol{\Sigma}$ and set the rest to zero

- Truncate the $\boldsymbol{U}$ and $\boldsymbol{V}^{\mathrm{T}}$ matrices, resulting in $\boldsymbol{U}_k$ and $\boldsymbol{V}_k^{\mathrm{T}}$

- Rows in $\boldsymbol{U}_k$ are the new low-dimensional word vectors

Applying SVD to Term-Context Matrix



- $\boldsymbol{U}_k$ 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

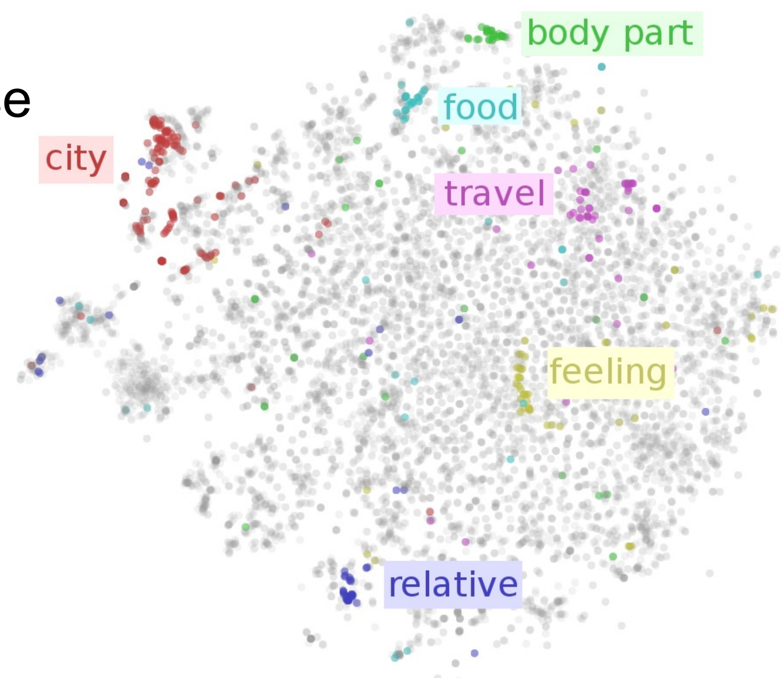
- Distributional semantics & word embedding
- PMI+SVD
- **GloVe**

Resource used in some slides: http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture02-wordvecs2.pdf

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:

$$\mathrm{PMI}(v,c) = \log_2 \frac{P(v,c)}{P(v)P(c)} \approx \log \frac{P(v,c)}{P(v)P(c)}$$

$$= \log \frac{x_{v,c}/|\mathcal{D}|}{X_v/|\mathcal{D}| \; X_c/|\mathcal{D}|} = \log \frac{x_{v,c}|\mathcal{D}|}{X_v X_c}$$

$$= \log x_{v,c} - \log X_v - \log X_c + \log|\mathcal{D}|$$

- $\log|\mathcal{D}|$ 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 …

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

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

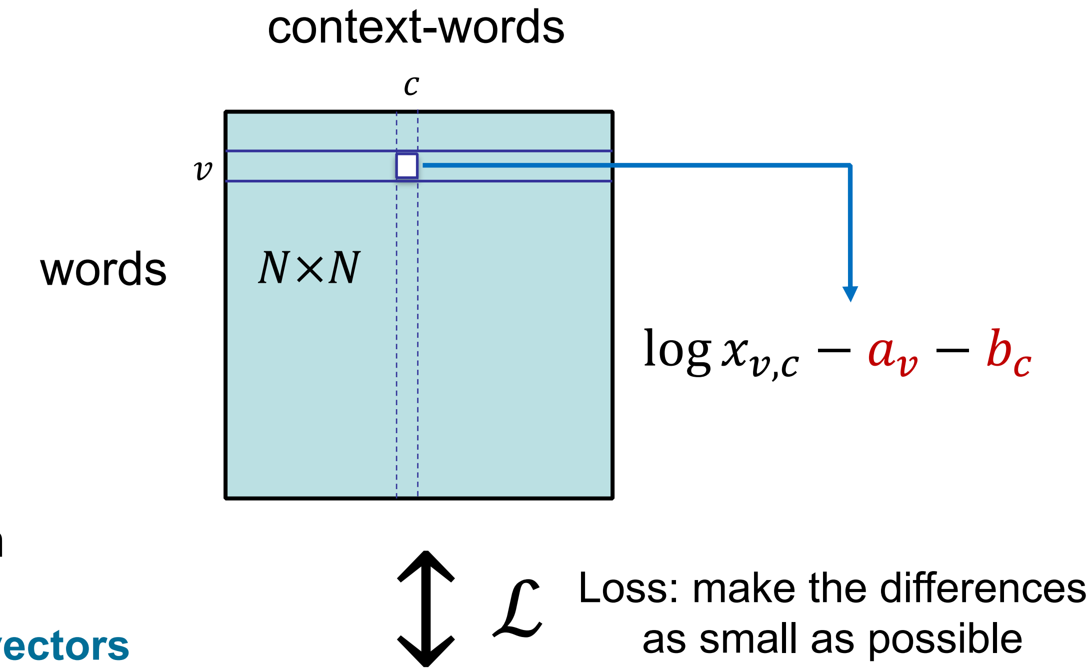
$$\log x_{v,c} - a_v - b_c$$

- 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 $\boldsymbol{a}$ contain the bias values of all words (like a_v)
- Vector $\boldsymbol{b}$ contain the bias values of all context-words (like b_c)

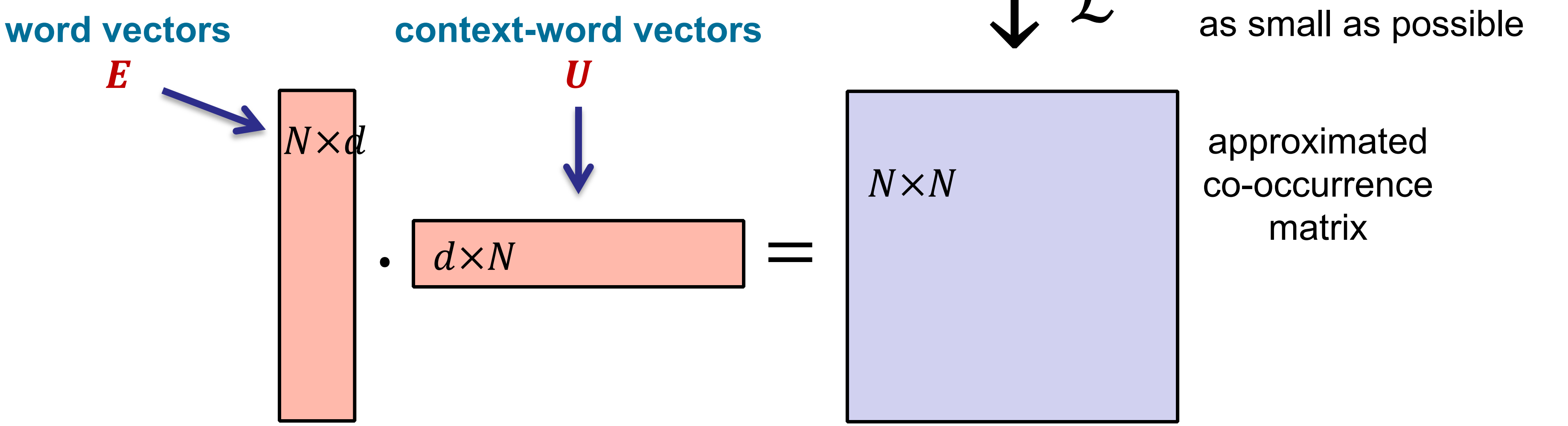
Model parameters are shown in red

Matrix factorization

GloVe word-context matrix:



2nd difference: matrix factorization



Model parameters are shown in red

Matrix factorization

Formal definition:

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

- $\boldsymbol{E}$ (word embeddings)
- $\boldsymbol{U}$ (context-word embeddings)

such that for every (v, c) pair:

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

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

Model parameters are shown in red

Loss function

- To find optimum values for parameters, GloVe uses least squares loss function:*

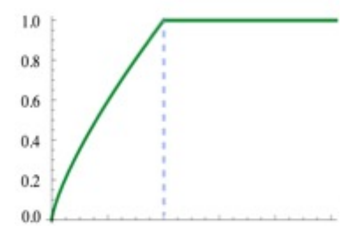
$$\mathcal{L} = \sum_{v \in \mathbb{V}, c \in \mathbb{C}} \left(\boldsymbol{e}_v \boldsymbol{u}_c^{\mathrm{T}} + a_v + b_c - \log x_{v,c} \right)^2$$

- Using the loss function, model parameters are optimized with Alternating Least Squares (ALS)
  - More about ALS is available here: http://stanford.edu/~rezab/classes/cme323/S15/notes/lec14.pdf

* 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):

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

$f \sim$



GloVe – Summary

- 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 $\boldsymbol{E}$
  - but also can be the sum of the word and context-word vectors: $\boldsymbol{E} + \boldsymbol{U}^{\mathrm{T}}$
    - This method is used in GloVe

- Characteristics
  - Fast training (no need for SVD!) $\rightarrow$ repeat SGD till loss converges
  - Scalable to large corpora
  - Good performance even with small corpus and small vectors