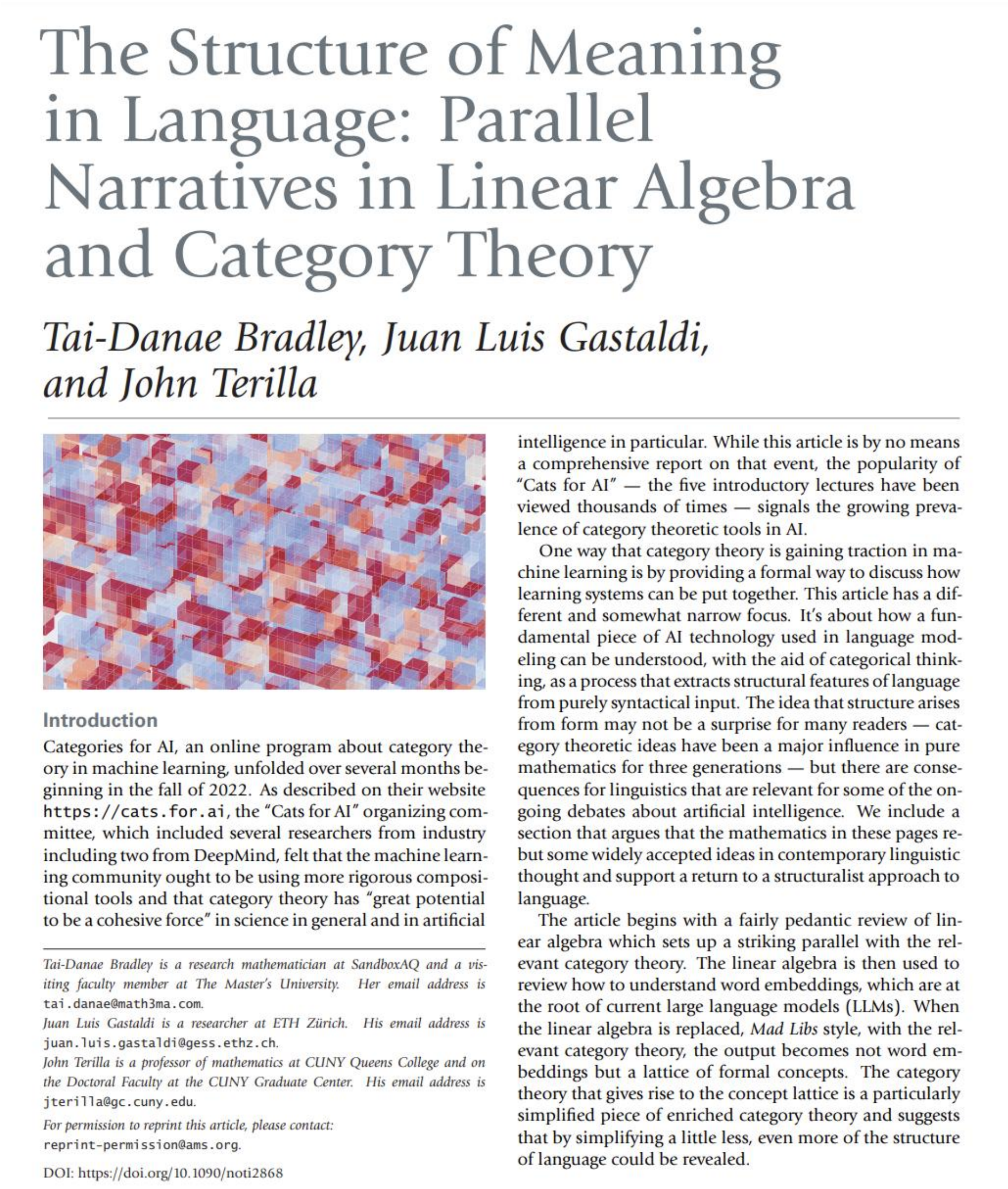
The Structure of Meaning in Language

Why can computers understand natural language



The Structure of Meaning in Language: Parallel Narratives in Linear Algebra and Category Theory

Tai-Danae Bradley, Juan Luis Gastaldi, and John Terilla

Introduction

Categories for AI, an online program about category theory in machine learning, unfolded over several months beginning in the fall of 2022. As described on their website `https://cats.for.ai`, the "Cats for AI" organizing committee, which included several researchers from industry including two from DeepMind, felt that the machine learning community ought to be using more rigorous compositional tools and that category theory has "great potential to be a cohesive force" in science in general and in artificial intelligence in particular. While this article is by no means a comprehensive report on that event, the popularity of "Cats for AI" — the five introductory lectures have been viewed thousands of times — signals the growing prevalence of category theoretic tools in AI.

One way that category theory is gaining traction in machine learning is by providing a formal way to discuss how learning systems can be put together. This article has a different and somewhat narrow focus. It's about how a fundamental piece of AI technology used in language modeling can be understood, with the aid of categorical thinking, as a process that extracts structural features of language from purely syntactical input. The idea that structure arises from form may not be a surprise for many readers — category theoretic ideas have been a major influence in pure mathematics for three generations — but there are consequences for linguistics that are relevant for some of the ongoing debates about artificial intelligence. We include a section that argues that the mathematics in these pages rebut some widely accepted ideas in contemporary linguistic thought and support a return to a structuralist approach to language.

The article begins with a fairly pedantic review of linear algebra which sets up a striking parallel with the relevant category theory. The linear algebra is then used to review how to understand word embeddings, which are at the root of current large language models (LLMs). When the linear algebra is replaced, *Mad Libs* style, with the relevant category theory, the output becomes not word embeddings but a lattice of formal concepts. The category theory that gives rise to the concept lattice is a particularly simplified piece of enriched category theory and suggests that by simplifying a little less, even more of the structure of language could be revealed.

Tai-Danae Bradley is a research mathematician at SandboxAQ and a visiting faculty member at The Master's University. Her email address is `tai.danae@math3ma.com`.

Juan Luis Gastaldi is a researcher at ETH Zürich. His email address is `juan.luis.gastaldi@gess.ethz.ch`.

John Terilla is a professor of mathematics at CUNY Queens College and on the Doctoral Faculty at the CUNY Graduate Center. His email address is `jterilla@gc.cuny.edu`.

For permission to reprint this article, please contact: `reprint-permission@ams.org`.

DOI: https://doi.org/10.1090/noti2868

Why can computers understand natural language?

The structuralist image of language behind word embeddings

Juan Luis Gastaldi

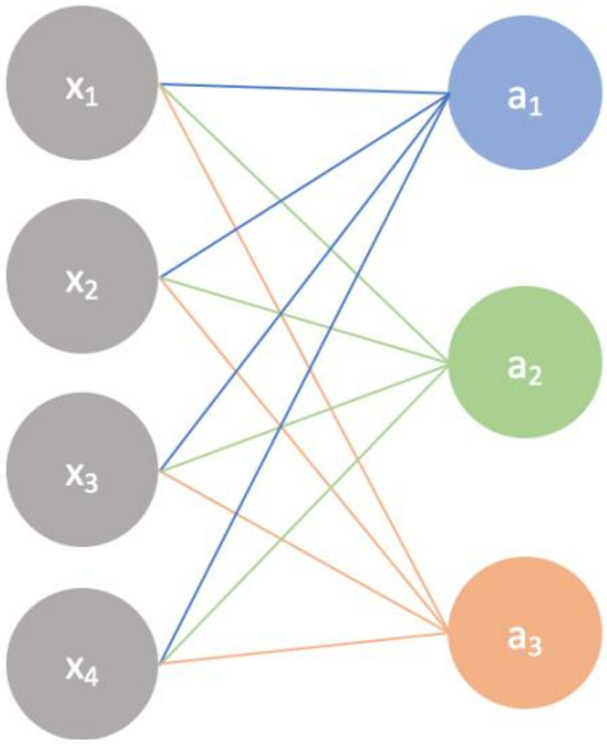
Abstract

The present paper intends to draw the conception of language implied in the technique of word embeddings that supported the recent development of deep neural network models in computational linguistics. After a preliminary presentation of the basic functioning of elementary artificial neural networks, we introduce the motivations and capabilities of word embeddings through one of its pioneering models, word2vec. To assess the remarkable results of the latter, we inspect the nature of its underlying mechanisms, which have been characterized as the implicit factorization of a word-context matrix. We then discuss the ordinary association of the "distributional hypothesis" with a "use theory of meaning", often justifying the theoretical basis of word embeddings, and contrast them to the theory of meaning stemming from those mechanisms through the lens of matrix models (such as VSMs and DSMs). Finally, we trace back the principles of their possible consistency through Harris's original distributionalism up to the structuralist conception of language of Saussure and Hjelmslev. Other than giving access to the technical literature and state of the art in the field of Natural Language Processing to non-specialist readers, the paper seeks to reveal the conceptual and philosophical stakes involved in the recent application of new neural network techniques to the computational treatment of language.

Keywords Word Embeddings · Natural Language Processing · word2vec · Neural Networks · Philosophy of Language · Matrix Models · Distributional Hypothesis · Structuralism

Originally published in *Philosophy & Technology* 34.1 (2021), pp. 149–214. ISSN: 2210-5441. DOI: 10.1007/s13347-020-00393-9. Please cite the original version.
This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 839730.
File version: 1.0.2023.02.16.

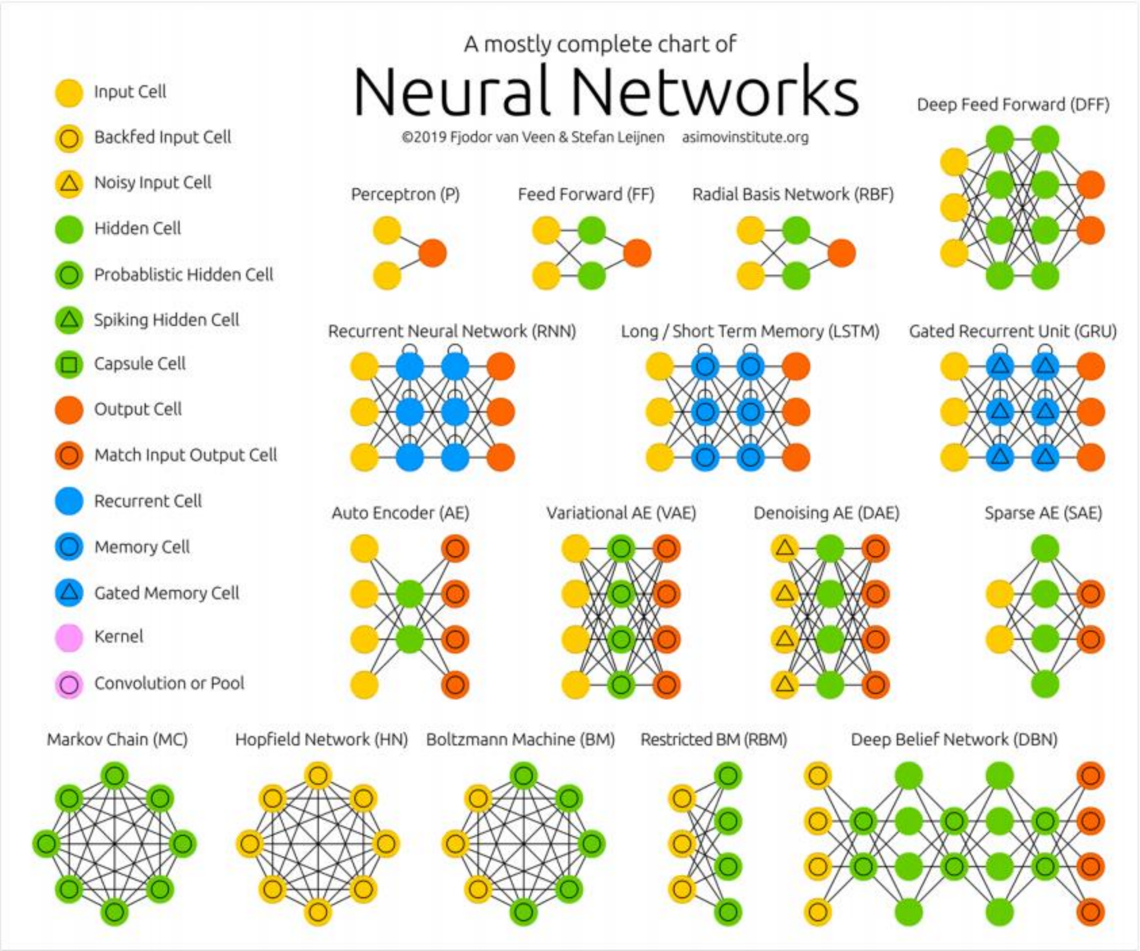
Neural Networks

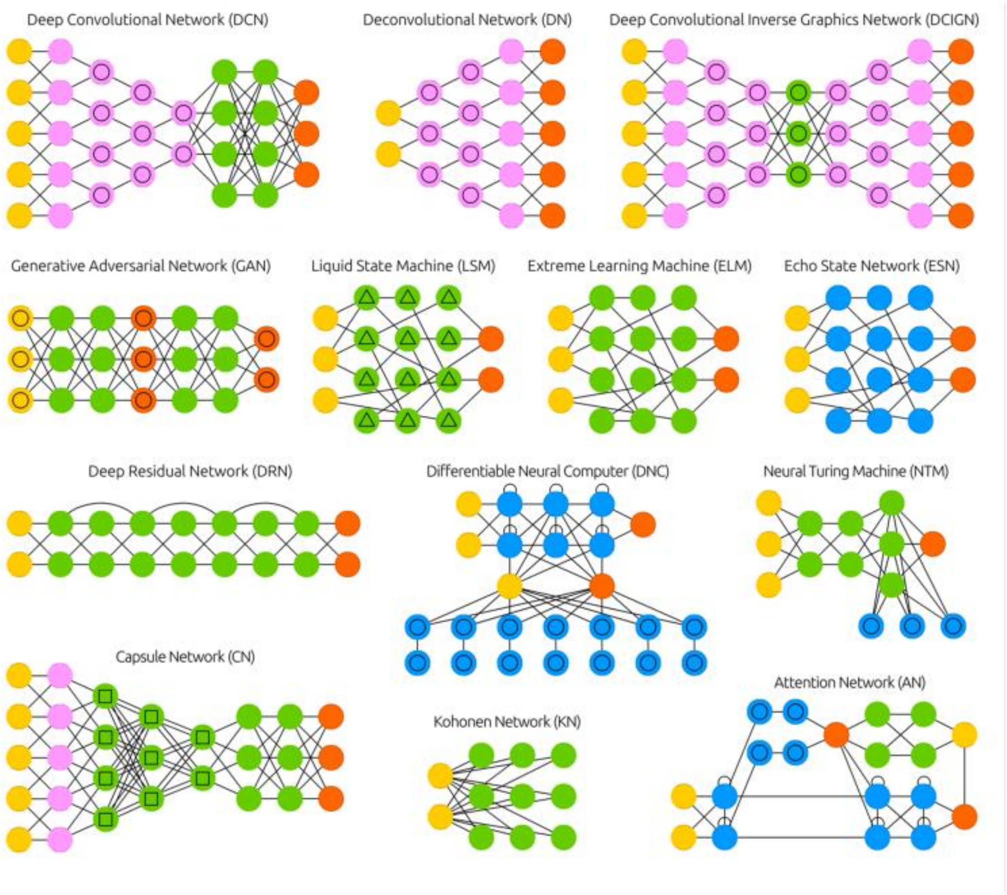


$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \end{bmatrix} \xrightarrow[\text{activation}]{} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Credit: Jeremy Jordan
https://www.jeremyjordan.me/intro-to-neural-networks/

The Family of DNNs

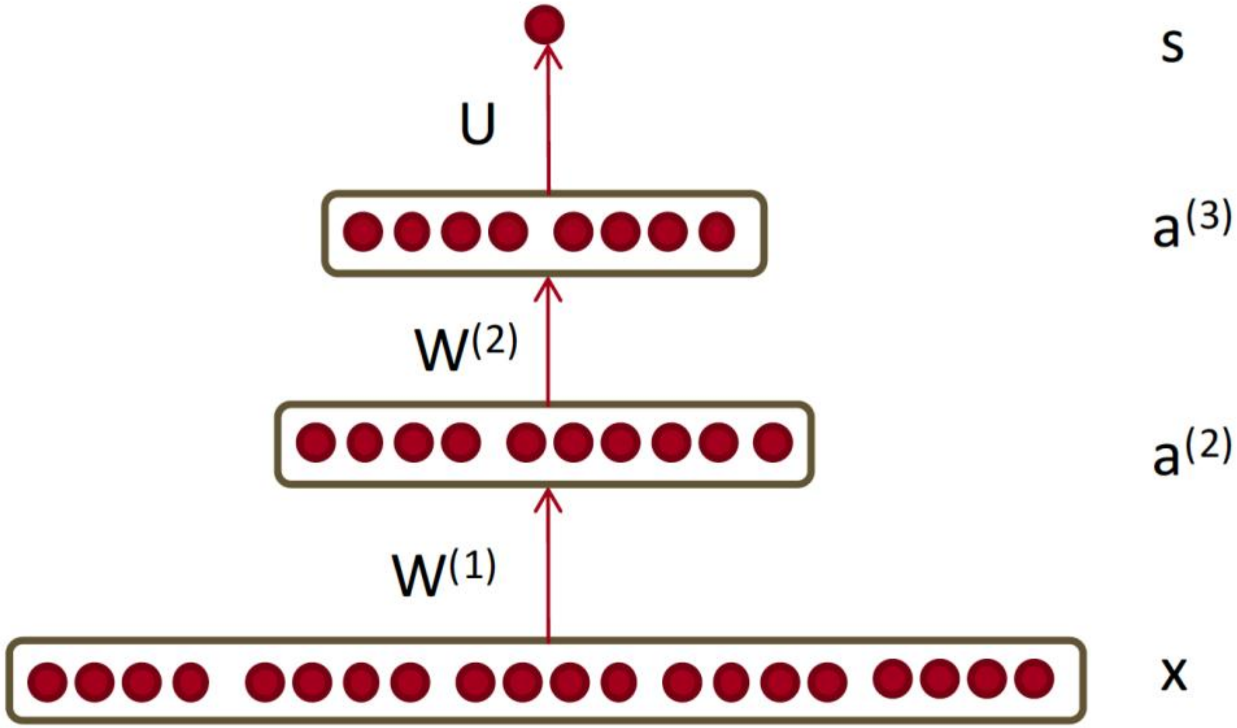



Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Differentiable Neural Computer (DNC)
Neural Turing Machine (NTM)
Capsule Network (CN)
Kohonen Network (KN)
Attention Network (AN)

Credit: https://www.asimovinstitute.org/neural-network-zoo/

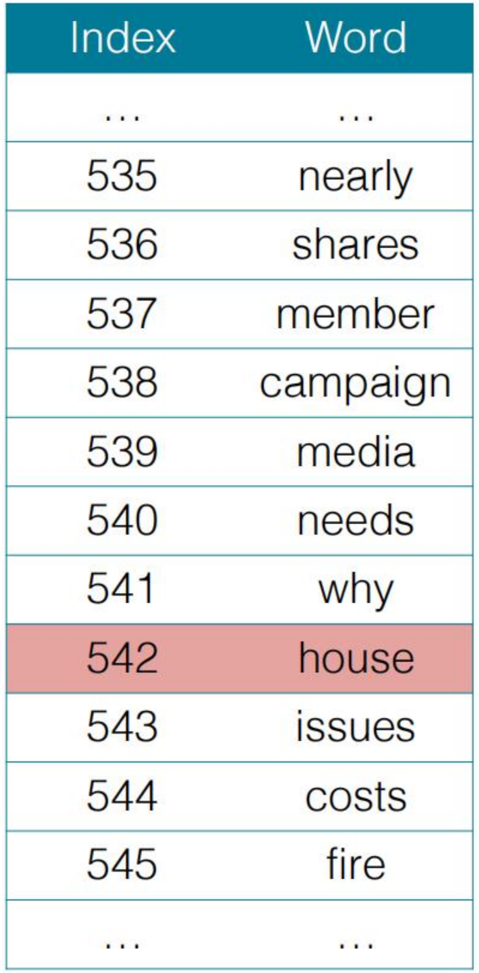
Deep Neural Nets (DNNs)

$$
\begin{aligned}
x &= z^{(1)} = a^{(1)} \\
z^{(2)} &= W^{(1)}x + b^{(1)} \\
a^{(2)} &= f\left(z^{(2)}\right) \\
z^{(3)} &= W^{(2)}a^{(2)} + b^{(2)} \\
a^{(3)} &= f\left(z^{(3)}\right) \\
s &= U^{T}a^{(3)}
\end{aligned}
$$



Credit: Manning & Socher, Stanford CS224n course, 2017

DNNs and Natural Language I



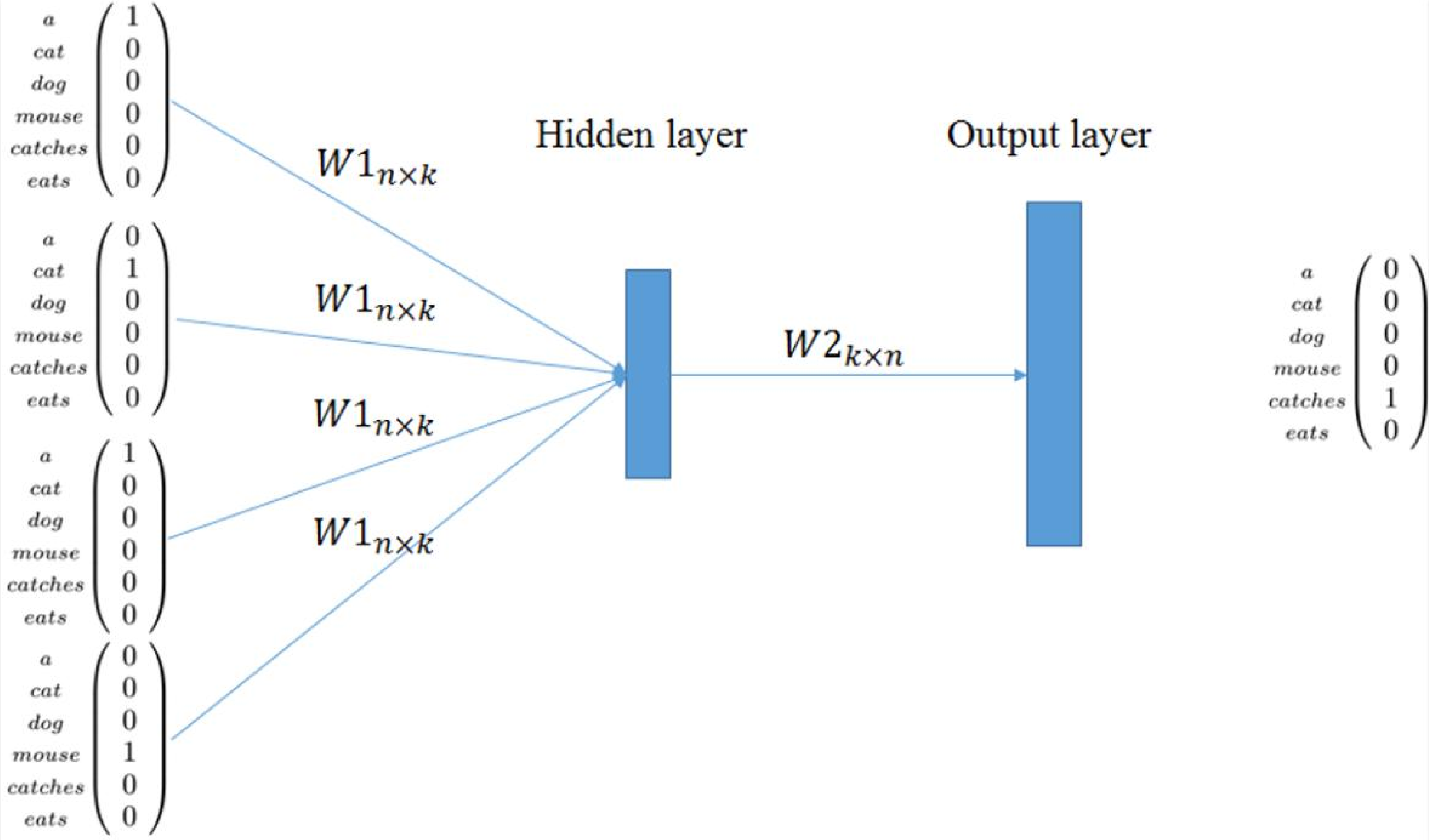
Index	Word
…	…
535	nearly
536	shares
537	member
538	campaign
539	media
540	needs
541	why
542	house
543	issues
544	costs
545	fire
…	…

$$v_{house} = (\underbrace{0, 0, 0, 0, 0, 0, 0, 0, \ldots, 0, \overbrace{1}^{542^{\text{nd}}\text{ position}}, 0, \ldots, 0, 0, 0, 0, 0, 0, 0, 0}_{\text{3 million dimensions}})$$

One-hot coding

Word Embeddings: word2vec

`a cat catches a mouse`



Dense Vector Representations

30, 000-dimensional real vector space

$$\xcancel{v_{house} = (\underbrace{0,0,0,0,0,0,0,0,\dots,0,\overbrace{1}^{542^{\text{nd}}\text{ position}},0,\dots,0,0,0,0,0,0,0,0}_{\text{3 million dimensions}})}$$

$$v_{house} = (\underbrace{0.157227, -0.0708008, 0.0539551, \dots, -0.041748, 0.00982666, -0.00494385, -0.032959}_{\text{300 dimensions}})$$

$$f : \boxed{\mathbb{R}^{n} \xrightarrow{f_1} \mathbb{R}^{n_1}} \xrightarrow{f_2} \mathbb{R}^{n_2} \xrightarrow{f_3} \cdots$$

$$\cdots \xrightarrow{f_K} \mathbb{R}^{n_K} \xrightarrow{g} \mathbb{R}^{m} \tag{1}$$

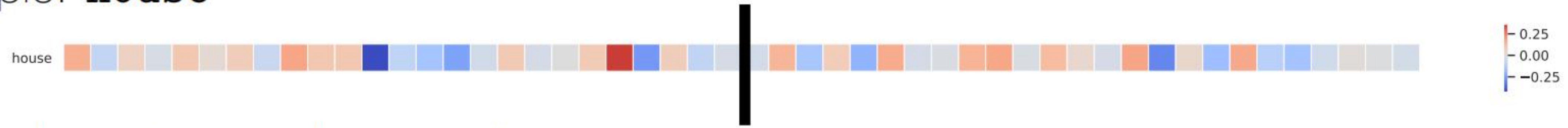
$$f_i(\mathbf{x}) = a(\mathbf{M}_i\mathbf{x} + b_i), \tag{2}$$

meaning of words

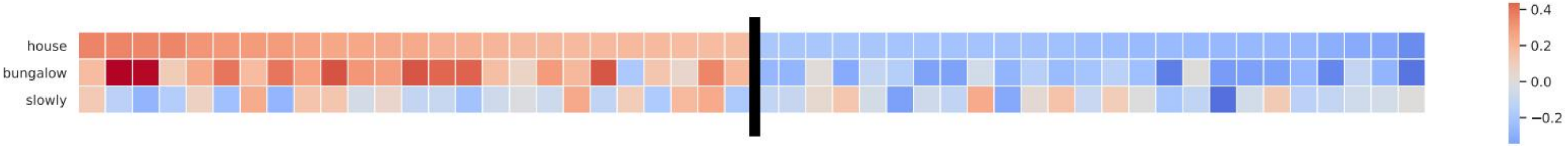
- Not only did the performance of models across different tasks increase substantially,
- but also unexpected linguistic significance was found in the vector space operations of the embedded word vectors. In particular, the inner product between two vectors shows a high correlation with semantic similarity

Word Embeddings: Example

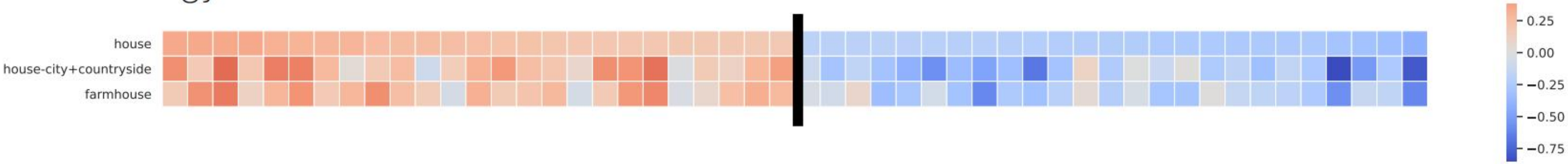
- Example: `house`



- Syntactic and semantic properties
  - Similarity

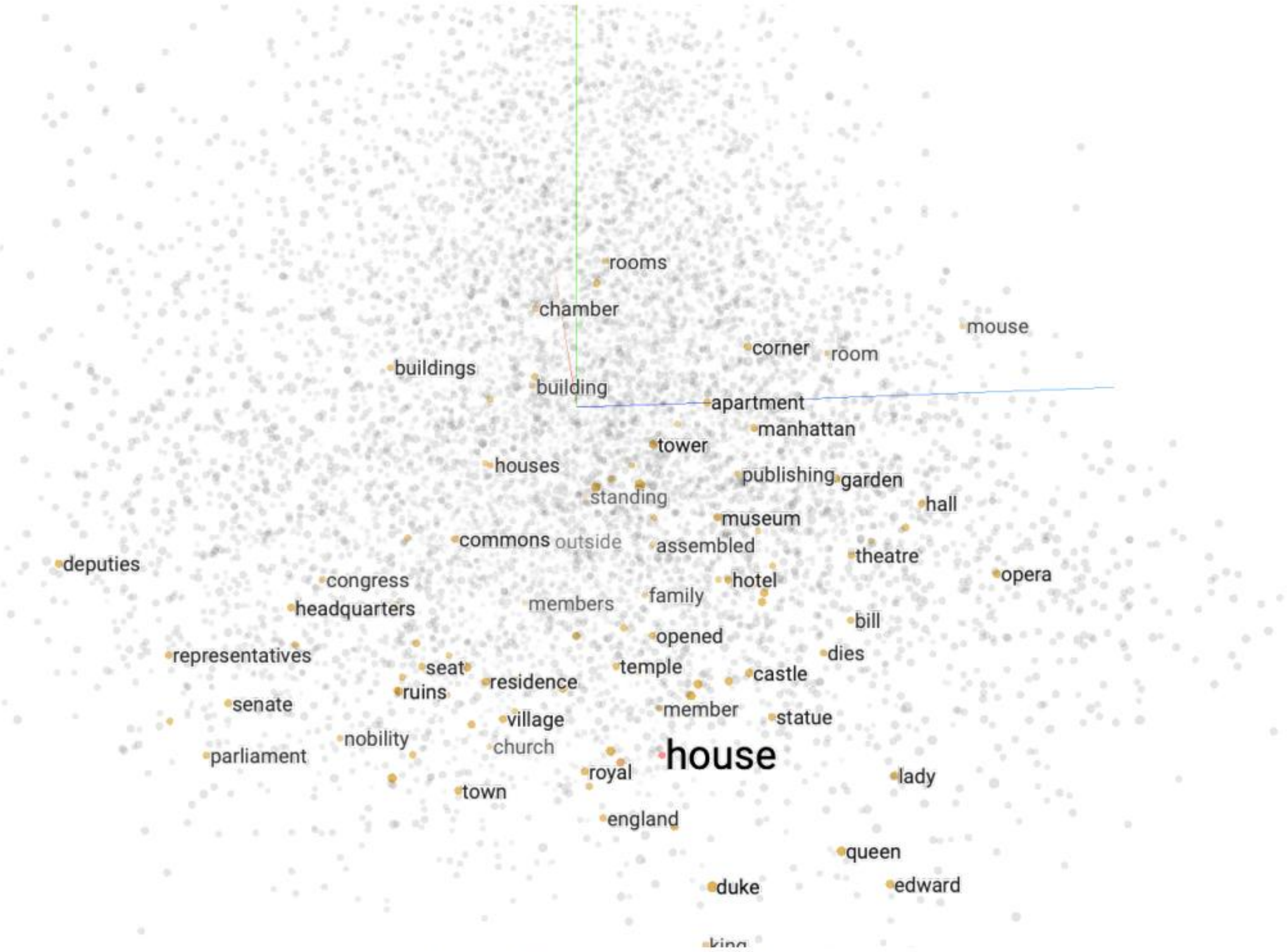


  - Analogy



Word Embeddings: Similarity

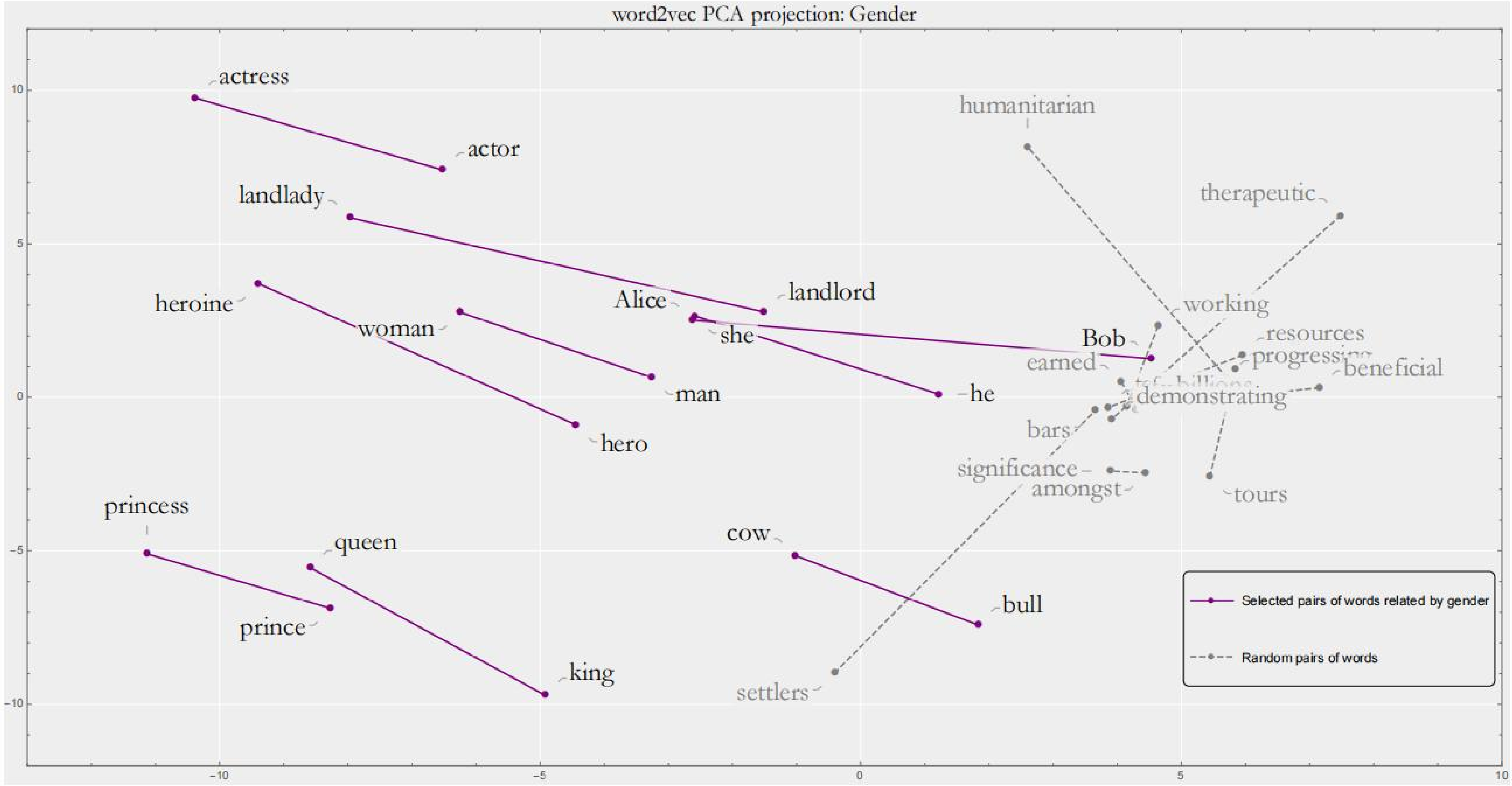
house	cosine distance
houses	0.292761
bungalow	0.312144
apartment	0.3371
bedroom	0.350306
townhouse	0.361592
residence	0.380158
mansion	0.394181
farmhouse	0.414243
duplex	0.424206
homes	0.43802



(https://projector.tensorflow.org)

Word Embeddings: Analogy

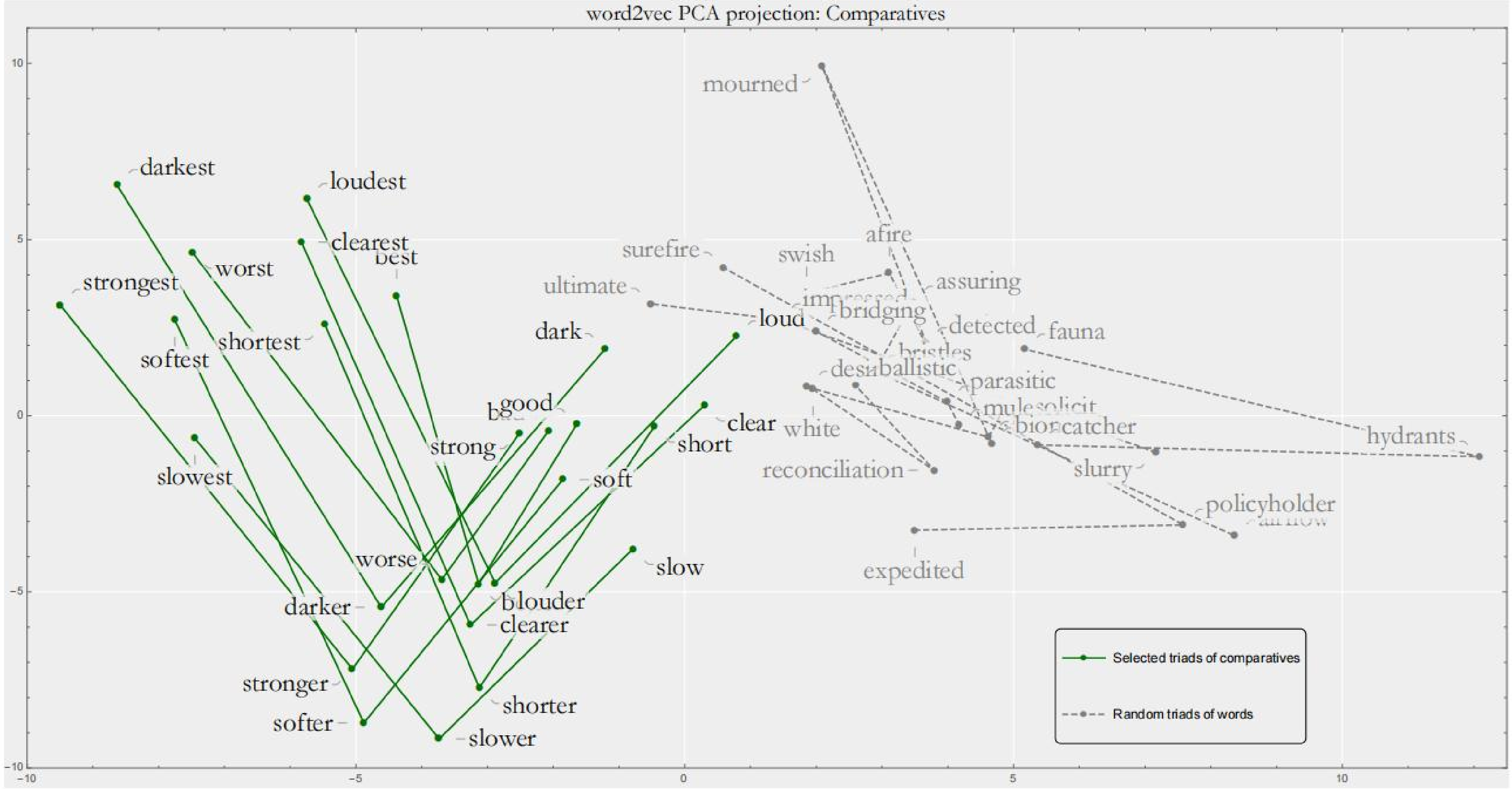
$$v_{king} - v_{queen} \approx v_{hero} - v_{heroine}$$

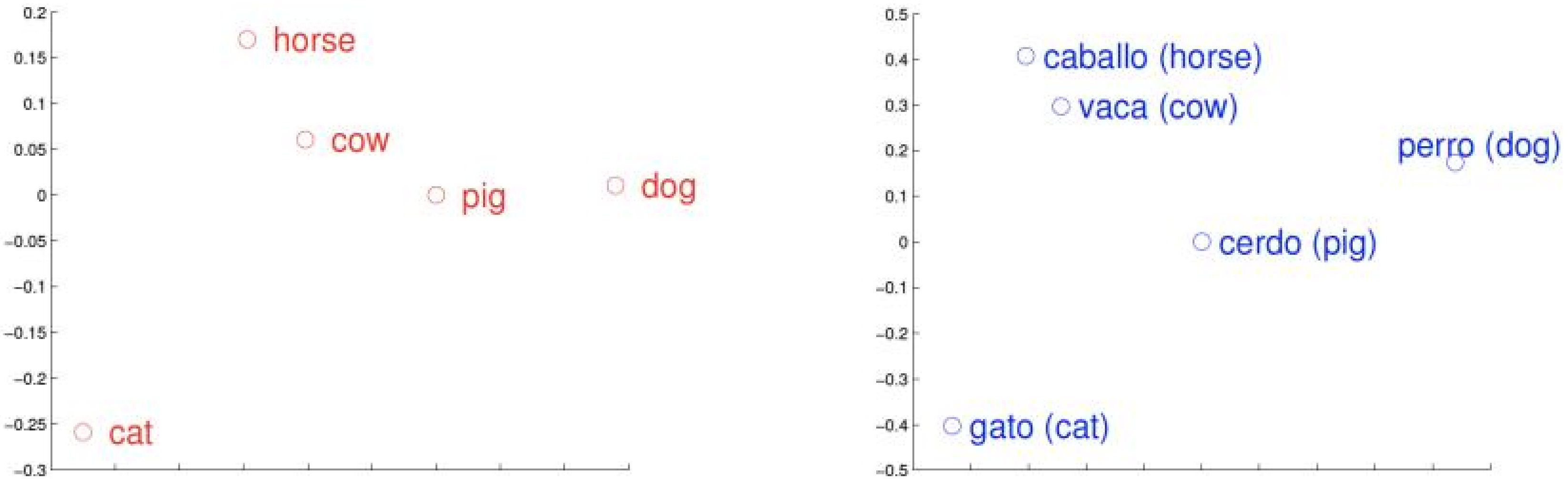


word2vec PCA projection: Gender

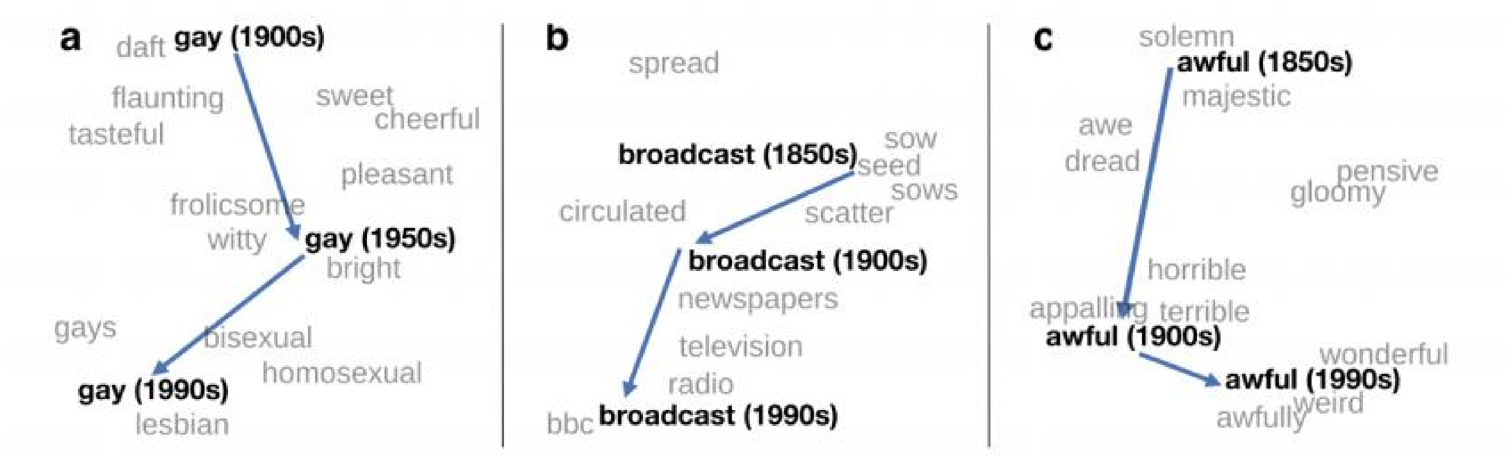
Word Embeddings: Analogy

$$v_{good} - v_{better} \approx v_{soft} - v_{softer}$$





(Mikolov, Sutskever, et al., 2013)



(Hamilton et al., 2016)

How to understand one-hot coding ?

There is a natural embedding

$$\begin{array}{ccc} \text{Set} & \longrightarrow & \text{functions on Set} \\ X & \longrightarrow & \mathrm{Fun}(X) \\ x & \longmapsto & \delta(x,-) \end{array}$$

If k is a field. then k^X is a Hilbert space

linear space , inner product . extra structure.

If k is $\{0,1\}$. then $\{0,1\}^X$ can regard as the set of all subset of X . $\{0,1\}^X$ is a Lattice Boolean algebra .

Category analogue

米田

$\mathcal{C}$ is a category. we have Yoneda embedding

$$y : \mathcal{C} \longrightarrow \mathrm{Set}^{\mathcal{C}^{op}} = \mathrm{Fun}(\mathcal{C}^{op}, \mathrm{Set})$$

$$X \longmapsto \hom(-, X)$$

$\mathrm{Set}^{\mathcal{C}^{op}}$ has all colimit.

And co-Yoneda embedding

$$z : \mathcal{C} \longrightarrow (\mathrm{Set}^{\mathcal{C}})^{op}$$

$$X \longmapsto \hom(X, -)$$

$(\mathrm{Set}^{\mathcal{C}})^{op}$ has all limit.

Singular Value Decomposition (SVD)

$$M = U\Sigma V^*$$

Where:

$$\begin{aligned} M &= m \times n \text{ (real or complex) matrix} \\ U &= m \times m \text{ unitary matrix} \\ \Sigma &= m \times n \text{ non-negative real rectangular diagonal matrix} \\ V* &= \text{conjugate transpose of } V\text{, a } n \times n \text{ unitary matrix} \end{aligned}$$

In particular:

- The columns of U (left singular vectors) are eigenvectors of $M \times M^*$
- The rows of V^* (right singular values) are eigenvectors of $M^* \times M$
- The non-zero elements of Σ (non-zero singular values) are the square roots of the non-zero eigenvalues of $M \times M^*$ or $M^* \times M$

How to get SVD :

$$M = U \Sigma V^*$$

$$M^* = V \Sigma U^*$$

$$M^* M = V \Sigma U^* \cdot U \Sigma V^* = V \Sigma^2 V^*$$

$$M M^* = U \Sigma V^* \cdot V \Sigma U^* = U \Sigma^2 U^*$$

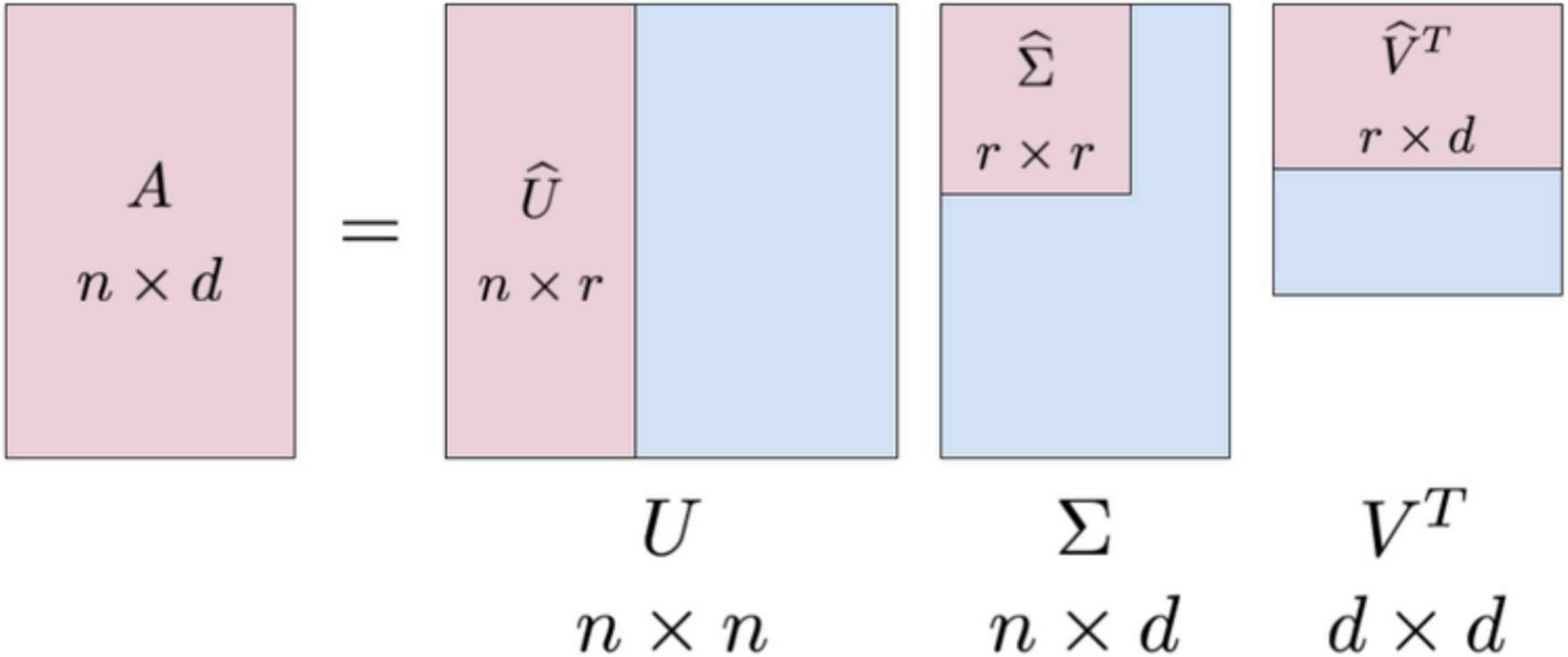
$$(M^* M) V = \Lambda V \qquad \sigma_i = \sqrt{\lambda_i}$$

$$(M M^*) U = \Lambda U$$

$$M^* u_j = \sigma_j v_j \qquad M v_j = \sigma_j u_j$$

Truncated SVD

$$M = U\Sigma V^*$$



$$X = \{-, /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, =, \mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{e}, \mathrm{f}, \mathrm{g}, \mathrm{h}, \mathrm{i}, \mathrm{j}, \mathrm{k}, \mathrm{l}, \mathrm{m}, \mathrm{n}, \mathrm{o}, \mathrm{p}, \mathrm{q}, \mathrm{r}, \mathrm{s}, \mathrm{t}, \mathrm{u}, \mathrm{v}, \mathrm{w}, \mathrm{x}, \mathrm{y}, \mathrm{z}, \mathrm{é}\}.$$

$Y = X \times X$

the : $h \longleftrightarrow t\_e$

hat : $a \longleftrightarrow h\_t$

c context $\in X$ $\in X \times X = Y$

Linguistically relevant measure

$$m : X \times Y \longrightarrow \mathbb{R}$$

$$(w, c) \longmapsto \log\left(\frac{p(w,c)}{p(w)p(c)}\right)$$

$$\begin{tikzcd} X \arrow[r] & k^Y \\ x \arrow[r, mapsto] & m(x,-) \end{tikzcd} \qquad \begin{tikzcd} Y \arrow[r] & k^X \\ y \arrow[r, mapsto] & m(-,y) \end{tikzcd}$$

$$\begin{tikzcd} & k^Y \\ X \arrow[ur] \arrow[r, hook] & k^X \arrow[u, dashed, "\exists M^*"'] \\ x \arrow[r, mapsto] & \delta(x,-) \end{tikzcd} \qquad \begin{tikzcd} \delta(-,y) & y \arrow[l, mapsto] \\ k^Y \arrow[d, dashed, "M"'] & Y \arrow[l, hook'] \arrow[dl] \\ k^X & \end{tikzcd}$$

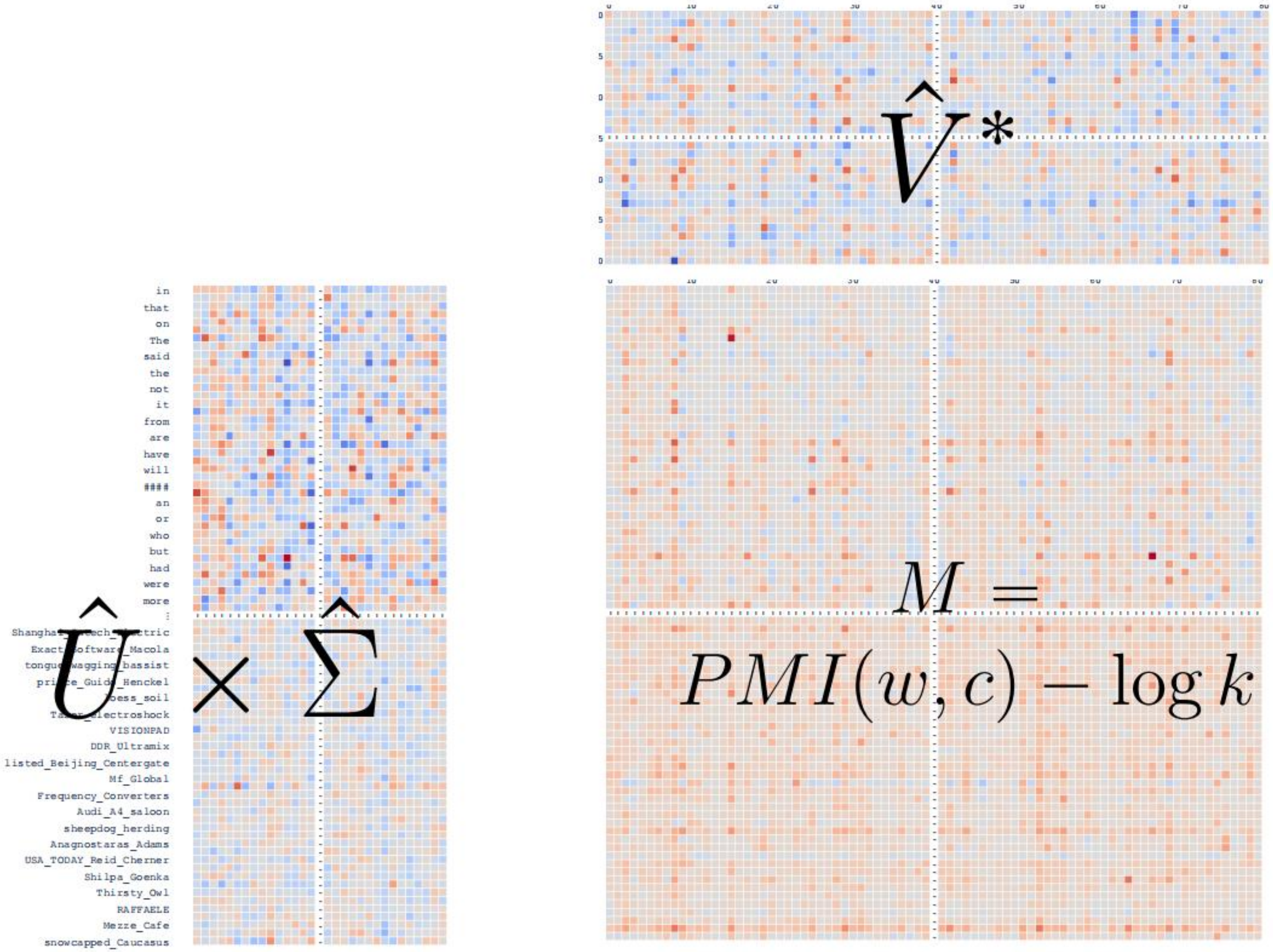
$$MM^* : k^X \longrightarrow k^X$$

$$M^*M : k^Y \longrightarrow k^Y$$

word2vec as Implicit Matrix Factorization

(Levy and Goldberg, 2014)

Embeddings as Truncated SVD



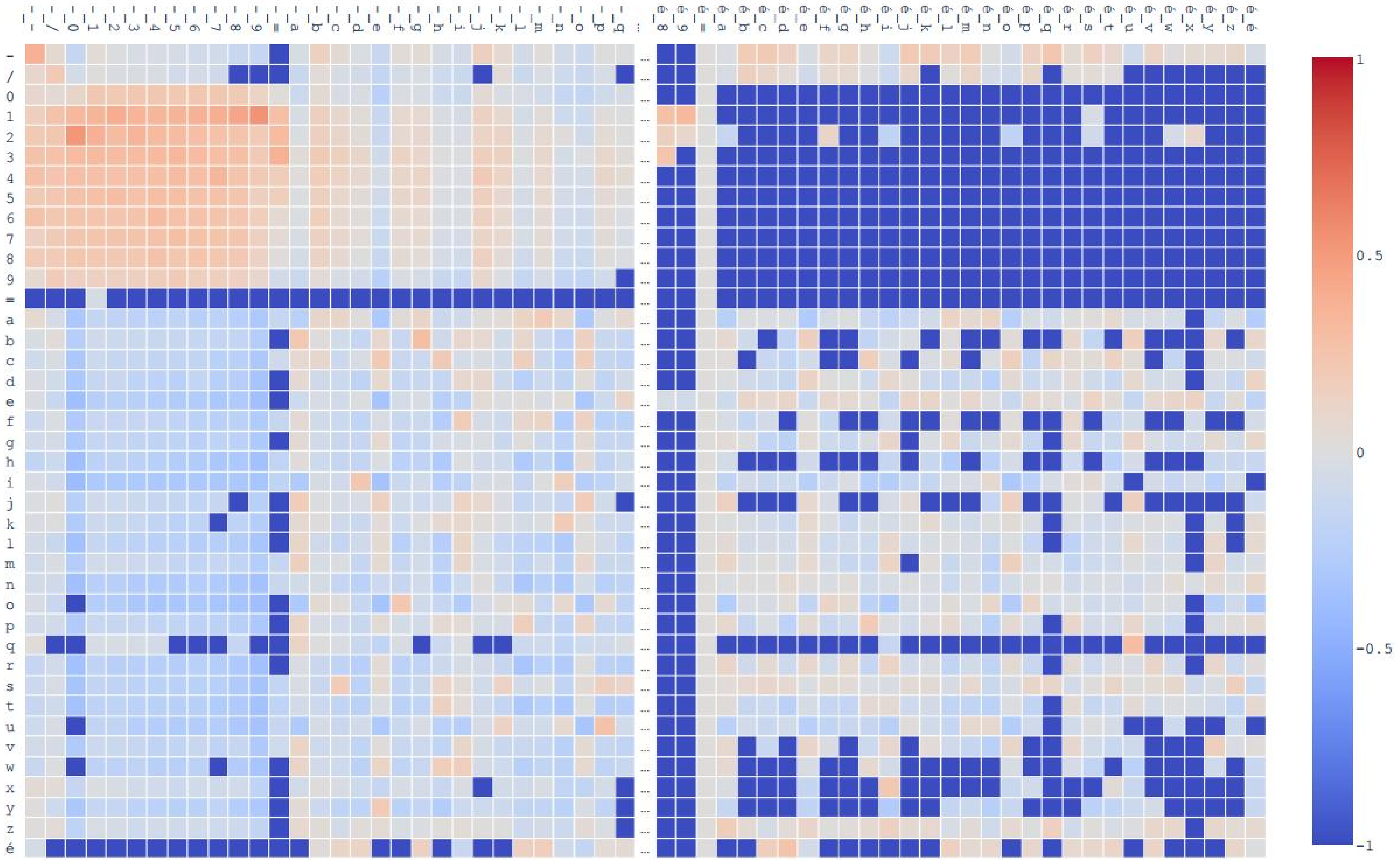
$$M \approx \hat{U} \times \hat{\Sigma} \times \hat{V}^*$$

$$PMI(w,c) = \log \frac{p(w,c)}{p(w)p(c)}$$

Example: Characters in Wikipedia

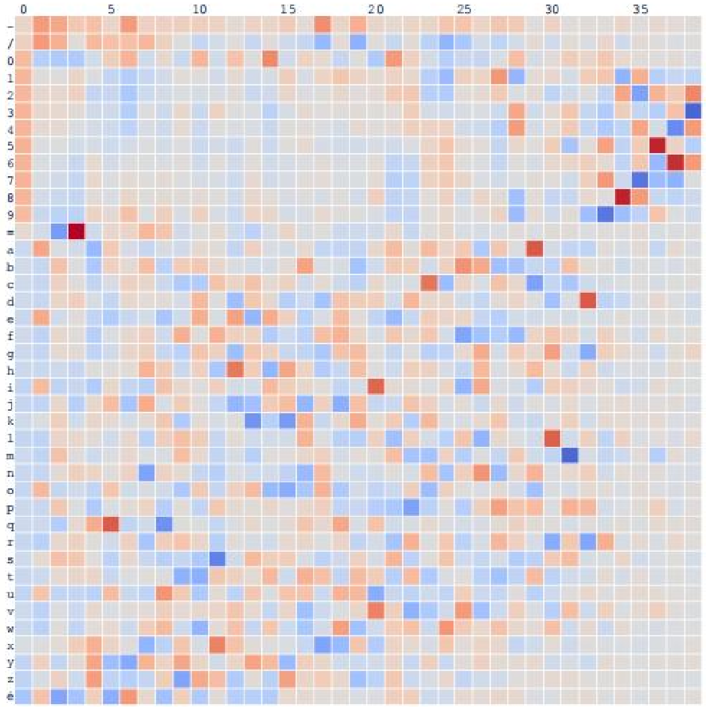
$$PMI(w, c) =$$

$$\log \frac{p(w,c)}{p(w)p(c)}$$

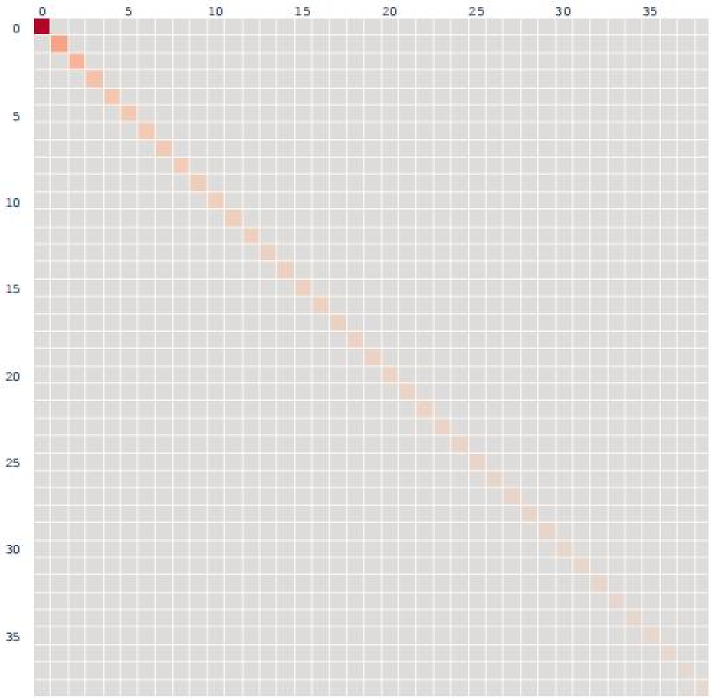


SVD of Wikipedia Character PMI Matrix

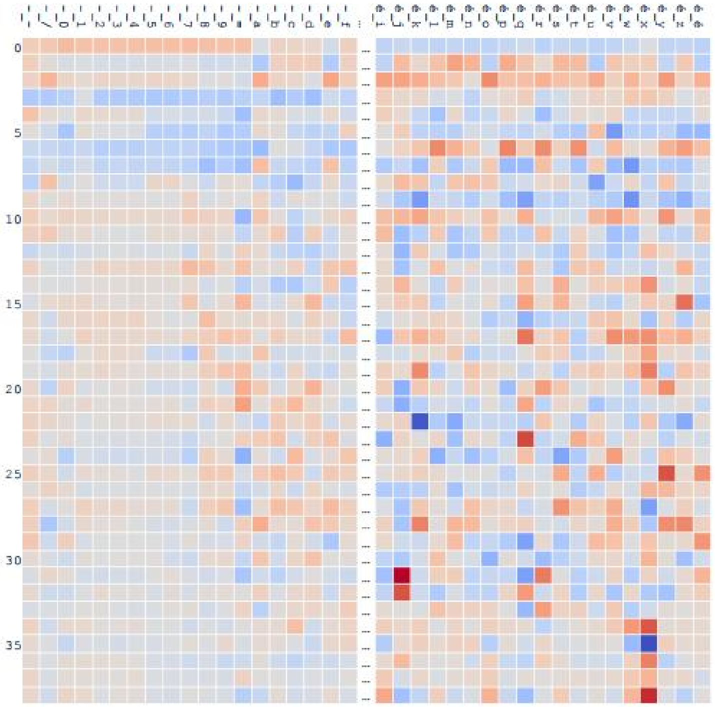
U



Σ

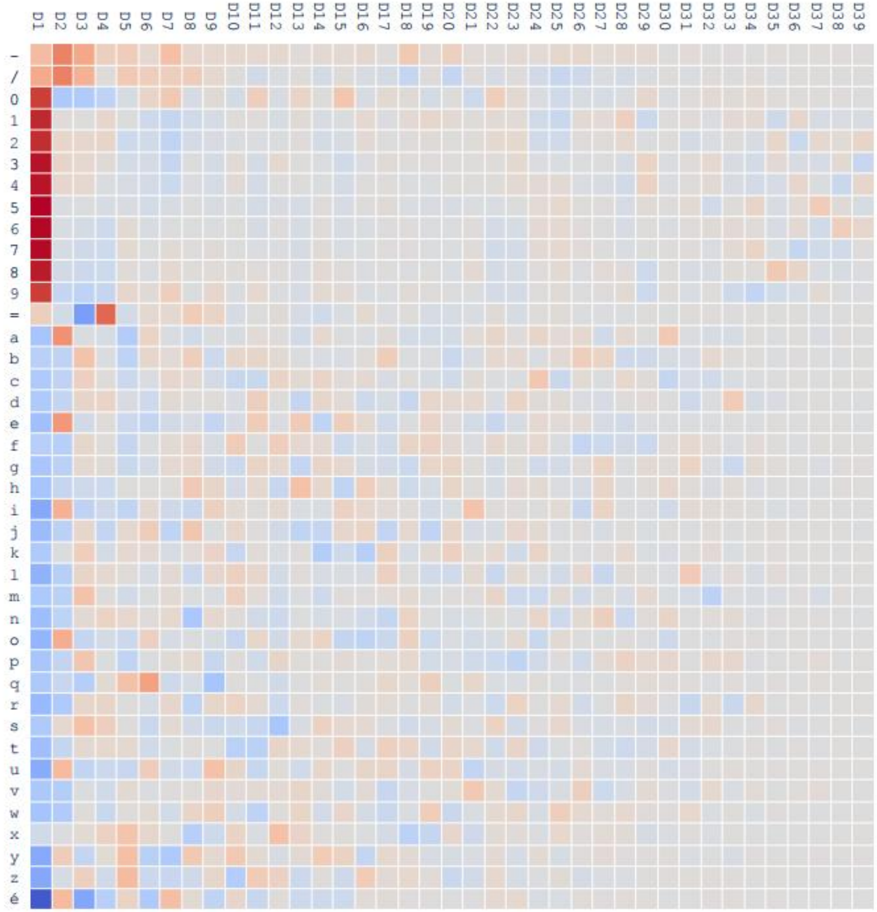


V^*



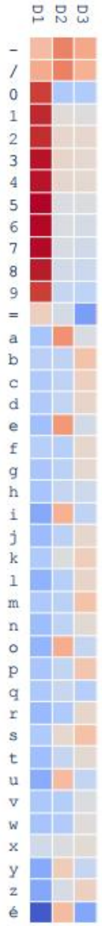
Every singular vectors represent some meaning

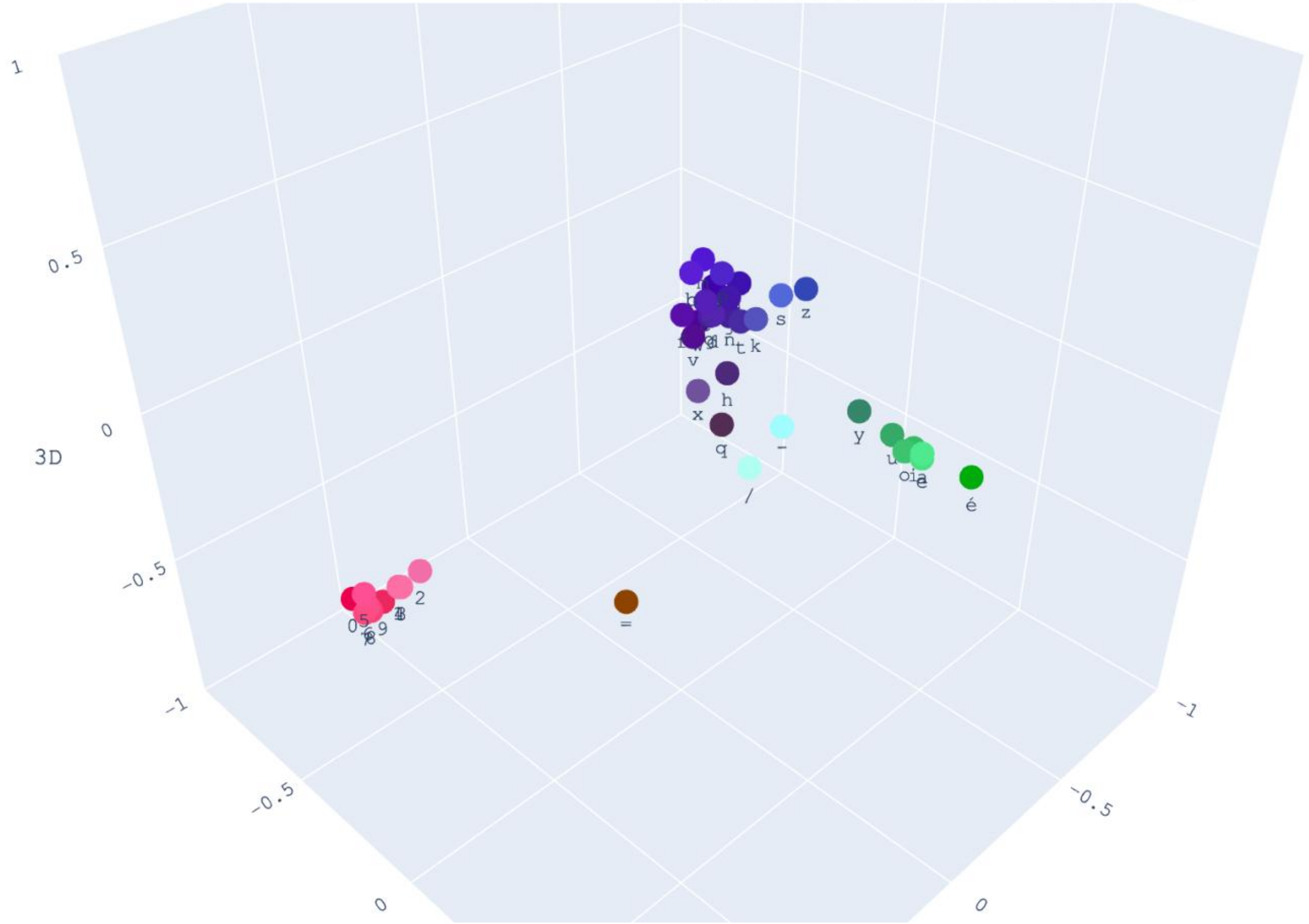
$$U \times \Sigma$$



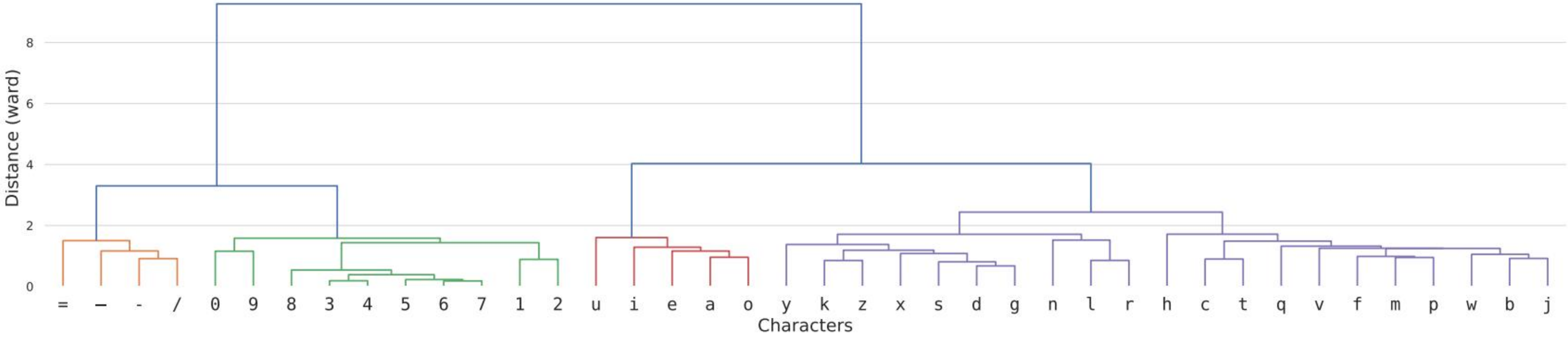
Truncate and Embed

$$\hat{U} \times \hat{\Sigma}$$





Clustering



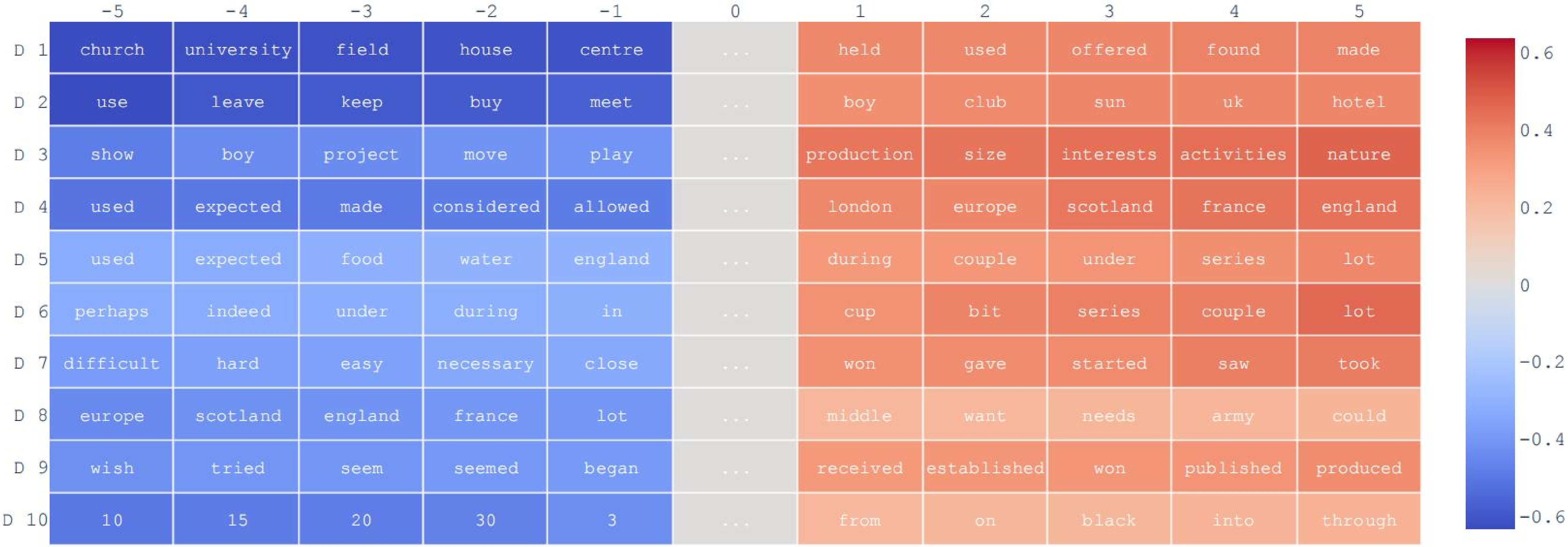
$$O := \{=, -, \text{-}, /\}$$

$$D := \{0, 9, 8, 3, 4, 5, 6, 7, 1, 2\}$$

$$V := \{\texttt{u}, \texttt{i}, \texttt{e}, \texttt{a}, \texttt{o}\}$$

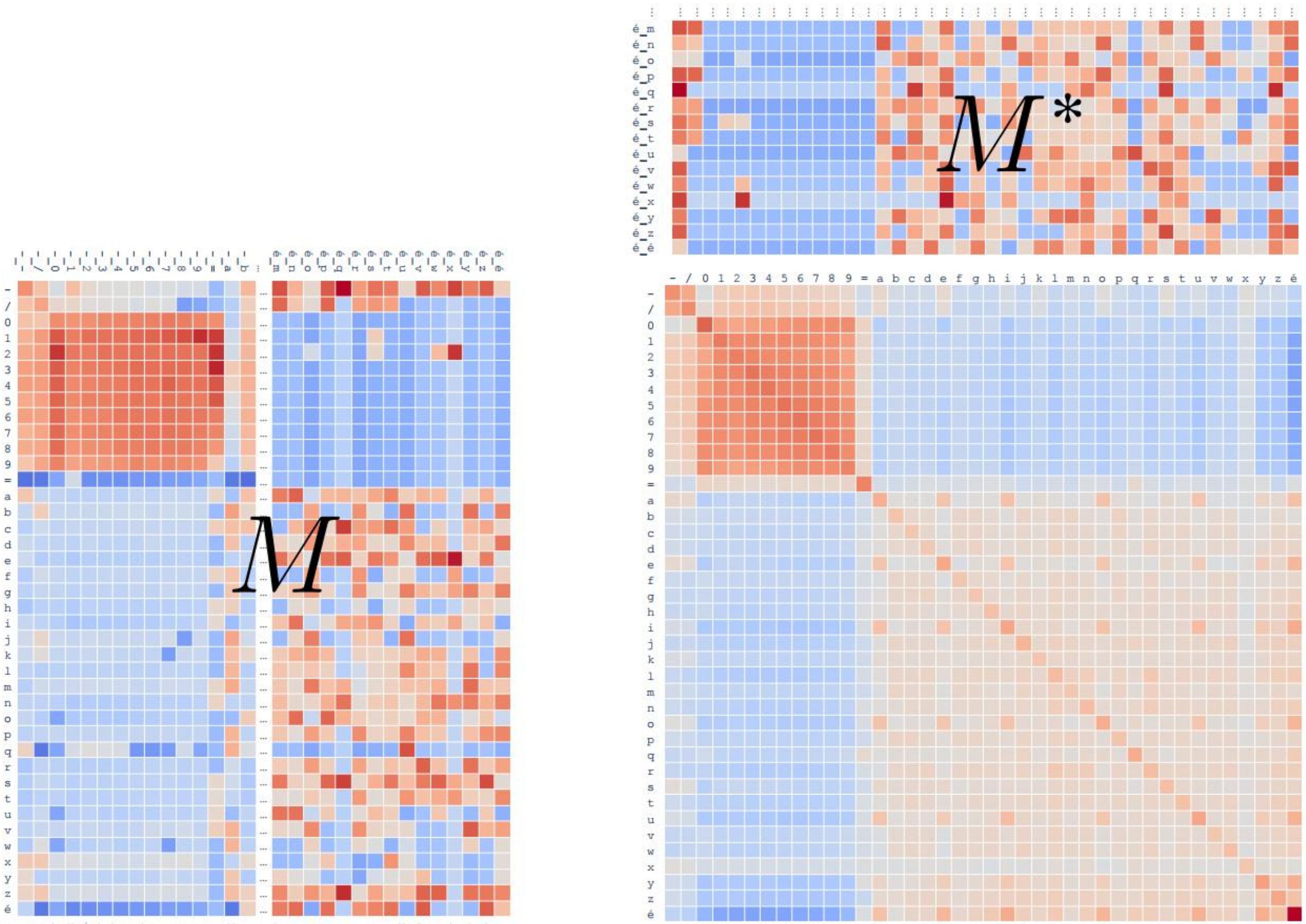
$$C := \{\texttt{y}, \texttt{k}, \texttt{z}, \texttt{x}, \texttt{s}, \texttt{d}, \texttt{g}, \texttt{n}, \texttt{l}, \texttt{r}, \texttt{h}, \texttt{c}, \texttt{t}, \texttt{q}, \texttt{v}, \texttt{f}, \texttt{m}, \texttt{p}, \texttt{w}, \texttt{b}, \texttt{j}\}$$

Words

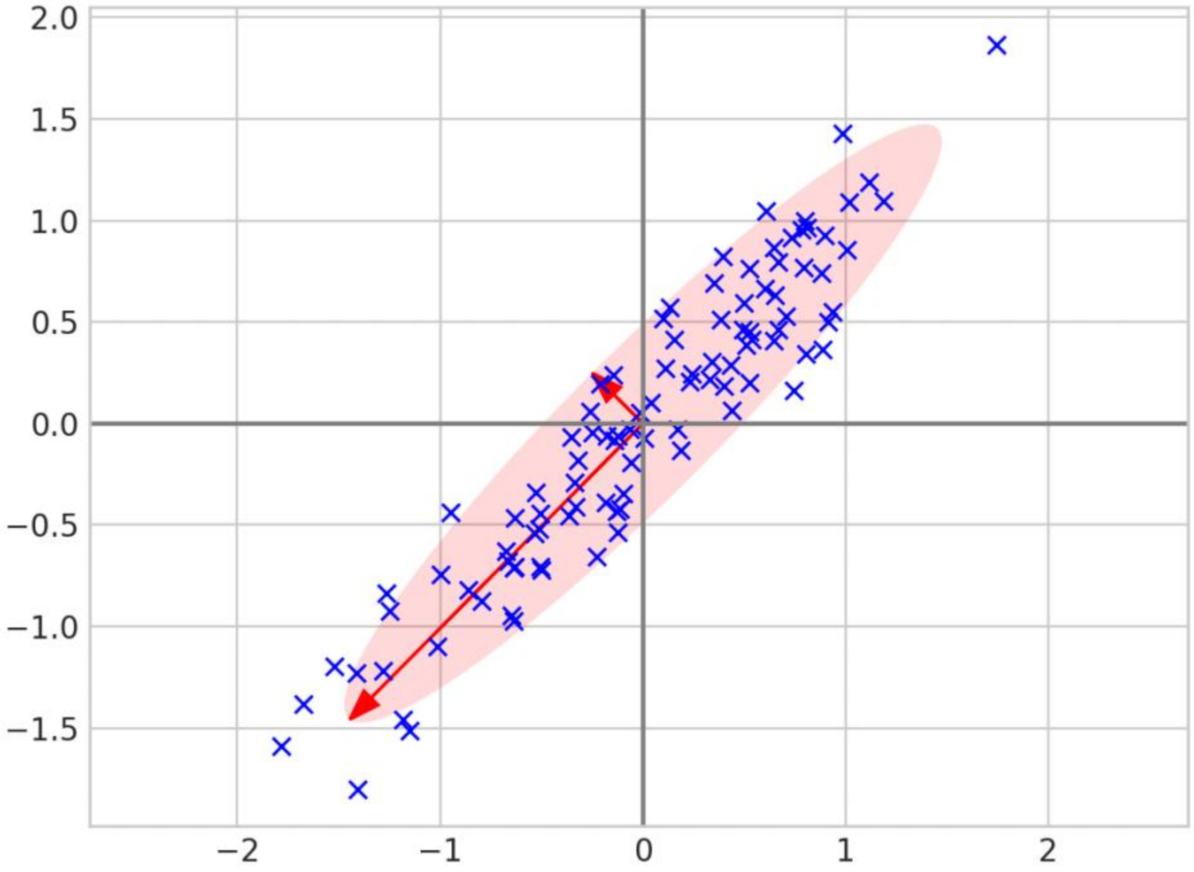


	-5	-4	-3	-2	-1	0	1	2	3	4	5
D 1	church	university	field	house	centre	...	held	used	offered	found	made
D 2	use	leave	keep	buy	meet	...	boy	club	sun	uk	hotel
D 3	show	boy	project	move	play	...	production	size	interests	activities	nature
D 4	used	expected	made	considered	allowed	...	london	europe	scotland	france	england
D 5	used	expected	food	water	england	...	during	couple	under	series	lot
D 6	perhaps	indeed	under	during	in	...	cup	bit	series	couple	lot
D 7	difficult	hard	easy	necessary	close	...	won	gave	started	saw	took
D 8	europe	scotland	england	france	lot	...	middle	want	needs	army	could
D 9	wish	tried	seem	seemed	began	...	received	established	won	published	produced
D 10	10	15	20	30	3	...	from	on	black	into	through

$M \times M^*$ as A Covariance Matrix

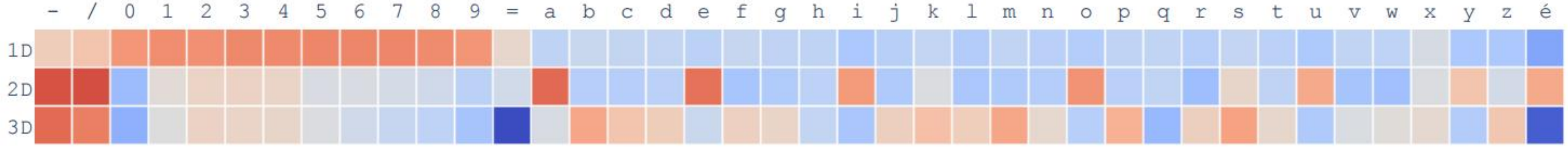


Eigenvectors and Eigenvalues

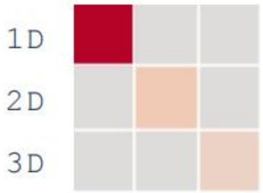


“Eigenstructure”

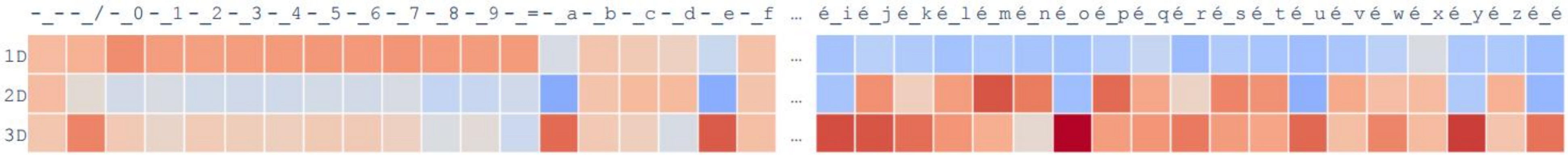
Eigenvectors of $M \times M^*$:



Eigenvalues of $M \times M^*$:

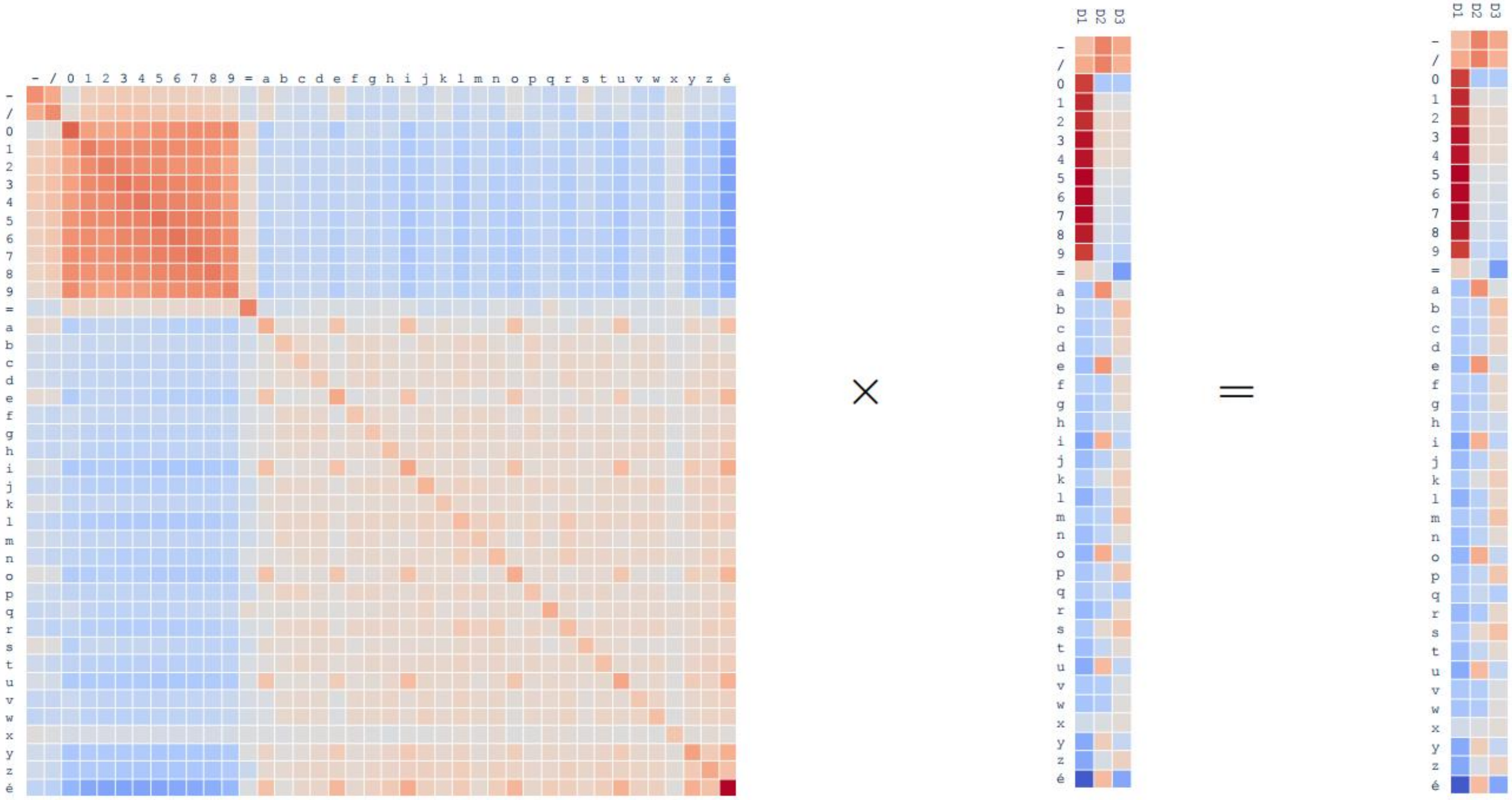


Eigenvectors of $M^* \times M$:



Eigenvectors as Fixed Points

$$(M \times M^{*})\color{red}{v}\color{black} = \lambda \color{red}{v}$$



Category analogue

$\mathcal{C}$ is a category. we have 米田 Yoneda embedding

$$y : \mathcal{C} \longrightarrow Set^{\mathcal{C}^{op}} = Fun(\mathcal{C}^{op}, Set)$$

$$X \longmapsto hom(-, X)$$

$Set^{\mathcal{C}^{op}}$ has all colimit.

And co-Yoneda embedding

$$z : \mathcal{C} \longrightarrow (Set^{\mathcal{C}})^{op}$$

$$X \longmapsto hom(X, -)$$

$(Set^{\mathcal{C}})^{op}$ has all limit.

Isbell conjugation.

$$\begin{tikzcd} \mathrm{Fun}(C^{op}, \mathrm{Set}) \arrow[r, shift left, "F^*"] & \mathrm{Fun}(C, \mathrm{Set})^{op} \arrow[l, shift left, "F_*"] \end{tikzcd}$$

$$F^* : \mathrm{Set}^{C^{op}} \rightleftarrows (\mathrm{Set}^{C})^{op} : F_*$$

$$F^*(f)(X) = \hom(f, y(X))$$

$$F_*(g)(X) = \hom(z(X), g)$$

profunctor : $m : C^{op} \times D \to Set$

$C \to (Set^D)^{op}$ $\qquad\qquad$ $D \to Set^{C^{op}}$

$c \mapsto m(c, -)$ $\qquad\qquad$ $d \mapsto m(-, d)$

$$\begin{tikzcd}
 & (Set^D)^{op} \\
C \arrow[ur] \arrow[r, "\text{Yoneda}"'] & Set^{C^{op}} \arrow[u, dashed, "F^*"']
\end{tikzcd}
\qquad
\begin{tikzcd}
(Set^D)^{op} \arrow[r, leftrightarrow, "\text{coYoneda}"] \arrow[d, dashed, "F_*"'] & D \arrow[dl] \\
Set^{C^{op}} &
\end{tikzcd}$$

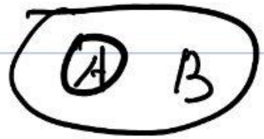
$F^*(f)(d) = hom(f, m(-,d))$ $\qquad$ $F_*(g)(c) = hom(m(c,-), g)$

Category 2 instead of Set

Profunctor (function) $r: X \times Y \longrightarrow \{0,1\}$

$$\begin{tikzcd}
 & \{0,1\}^Y \\
X \arrow[ur, "r"] \arrow[r] & \{0,1\}^X \arrow[u, dashed, "R^*"']
\end{tikzcd}$$

$0 \to 1$



$A \hookrightarrow B$

$$R^*(A) = \{ y \in Y : R(x,y) = 1 \text{ for all } x \in A \}$$

$$\begin{tikzcd}
\mathrm{set}^Y \arrow[d, "R_*"'] & Y \arrow[l, hook'] \arrow[dl, "r"] \\
\mathrm{set}^X &
\end{tikzcd}$$

$$R_*(B) = \{ x \in X : R(x,y) = 1 \text{ for all } y \in B \}$$

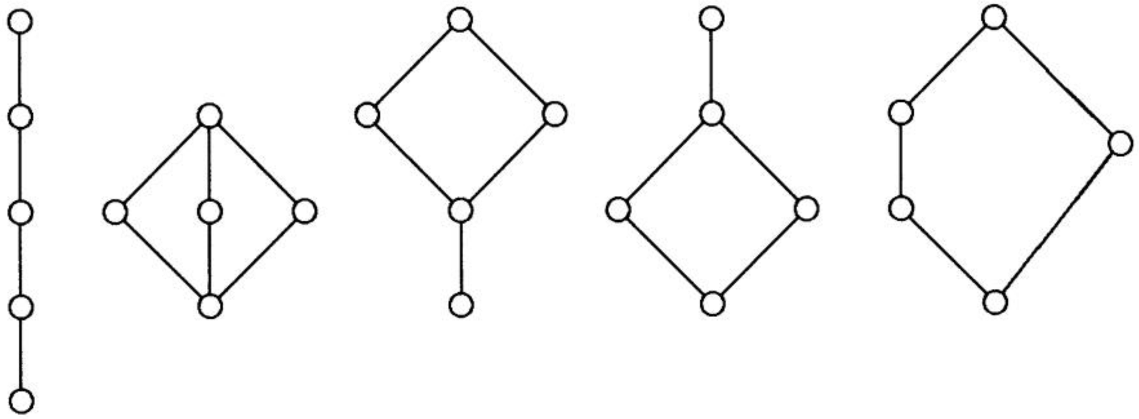
Fixed obj of R^*R_* and R_*R^* are formal concept.

Formal concept analysis

- 形式概念分析（Formal Concept Analysis，简称FCA）是一种基于格论（Lattice Theory）应用分支的数学理论，它使得概念的形式化成为可能，并以对象-属性的形式分析数据。FCA起源于20世纪80年代初，由Rudolf Wille提出，并在过去的几十年中逐渐发展成为一个知识表示和数据分析工具，广泛应用于信息检索、机器学习、数据挖掘、知识发现、文本挖掘等领域。
- FCA的一个关键优势是它提供了一种直观的方式来表示和分析复杂的数据结构，通过概念格图（Hasse Diagram）可以清晰地展示对象和属性之间的层次关系。此外，FCA还能够处理多值属性和不完全数据，使其成为一个灵活和强大的数据分析工具

Definition 9. Let $(M, \leq)$ be an ordered set and A a subset of M. A **lower bound** of A is an element s of M with $s \leq a$ for all $a \in A$. An **upper bound** of A is defined dually. If there is a largest element in the set of all lower bounds of A, it is called the **infimum** of A and is denoted by $\inf A$ or $\bigwedge A$; dually, a least upper bound is called **supremum** and denoted by $\sup A$ or $\bigvee A$. If $A = \{x, y\}$, we also write $x \wedge y$ for $\inf A$ and $x \vee y$ for $\sup A$. Infimum and supremum are frequently also called **meet** and **join**. $\diamond$

Definition 10. An ordered set $\boldsymbol{V} := (V, \leq)$ is a **lattice**, if for any two elements x and y in V the supremum $x \vee y$ and the infimum $x \wedge y$ always exist.



Definition 18. A **formal context** $\mathbb{K} := (G, M, I)$ consists of two sets G and M and a relation I between G and M. The elements of G are called the **objects** and the elements of M are called the **attributes** of the context[1]. In order to express that an object g is in a relation I with an attribute m, we write gIm or $(g, m) \in I$ and read it as "the object g **has** the attribute m".

		a	b	c	d	e	f	g	h	i
1	Leech	×	×					×		
2	Bream	×	×					×	×	
3	Frog	×	×	×				×	×	
4	Dog	×		×				×	×	×
5	Spike – weed	×	×		×		×			
6	Reed	×	×	×	×		×			
7	Bean	×		×	×	×				
8	Maize	×		×	×		×			

Figure 1.1 Context of an educational film "Living Beings and Water". The attributes are: a: needs water to live, b: lives in water, c: lives on land, d: needs chlorophyll to produce food, e: two seed leaves, f: one seed leaf, g: can move around, h: has limbs, i: suckles its offspring.

Definition 19. For a set $A \subseteq G$ of objects we define

$$A' := \{m \in M \mid gIm \text{ for all } g \in A\}$$

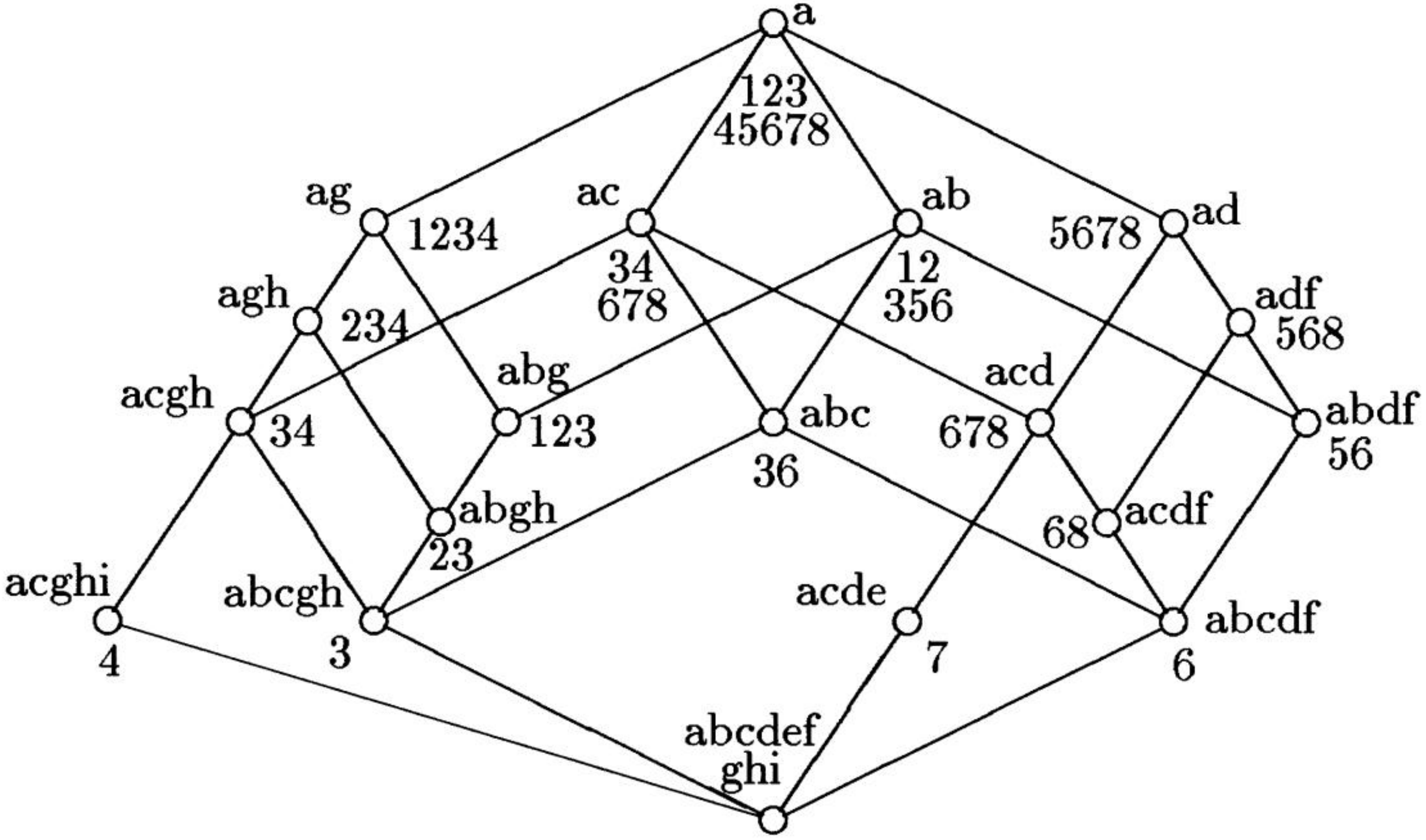
(the set of attributes common to the objects in A). Correspondingly, for a set B of attributes we define

$$B' := \{g \in G \mid gIm \text{ for all } m \in B\}$$

(the set of objects which have all attributes in B).[2] ◇

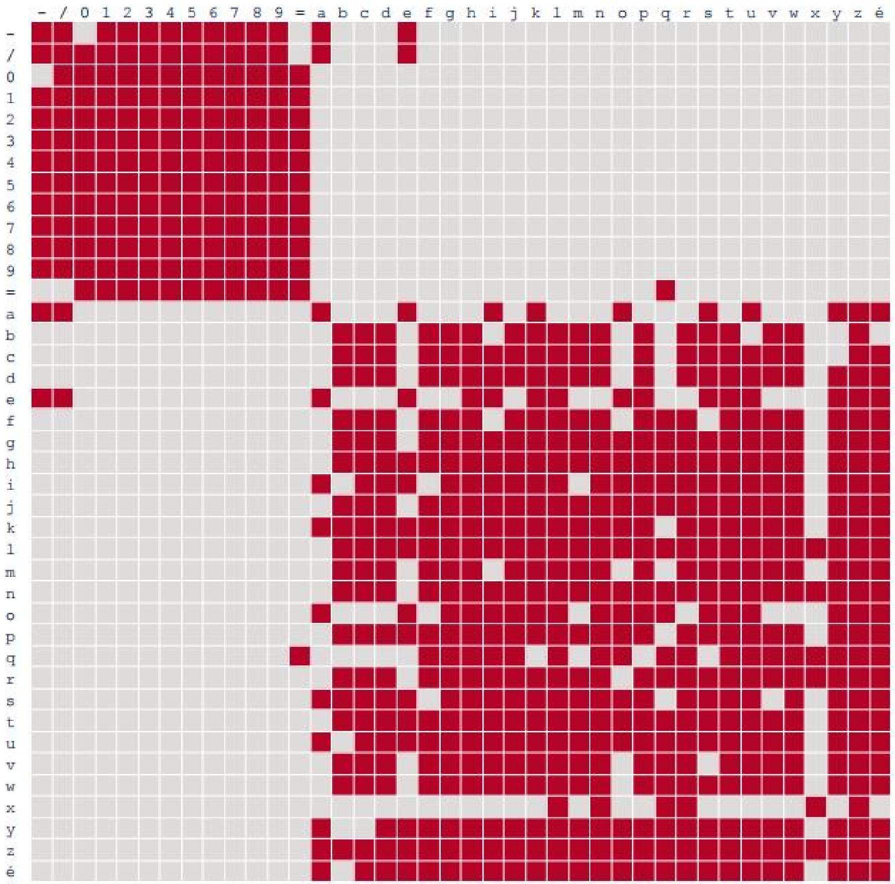
Definition 20. A **formal concept** of the context (G, M, I) is a pair (A, B) with $A \subseteq G, B \subseteq M$, $A' = B$ and $B' = A$. We call A the **extent** and B the **intent** of the concept (A, B). $\mathfrak{B}(G, M, I)$ denotes the set of all concepts of the context (G, M, I). ◇

Definition 21. If (A_1, B_1) and (A_2, B_2) are concepts of a context, (A_1, B_1) is called a **subconcept** of (A_2, B_2), provided that $A_1 \subseteq A_2$ (which is equivalent to $B_2 \subseteq B_1$). In this case, (A_2, B_2) is a **superconcept** of (A_1, B_1), and we write $(A_1, B_1) \leq (A_2, B_2)$. The relation $\leq$ is called the **hierarchical order** (or simply **order**) of the concepts. The set of all concepts of (G, M, I) ordered in this way is denoted by $\underline{\mathfrak{B}}(G, M, I)$ and is called the **concept lattice** of the context (G, M, I).

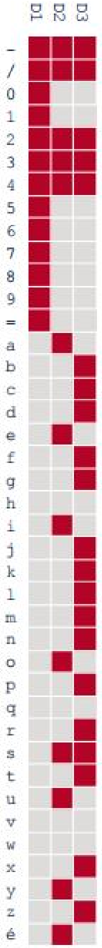


Binary: Formal Concepts

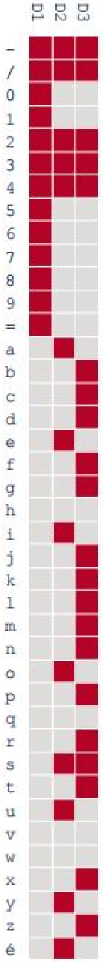
$$(M \times M^*)\color{red}{v}\color{black} = \lambda \color{red}{v}$$



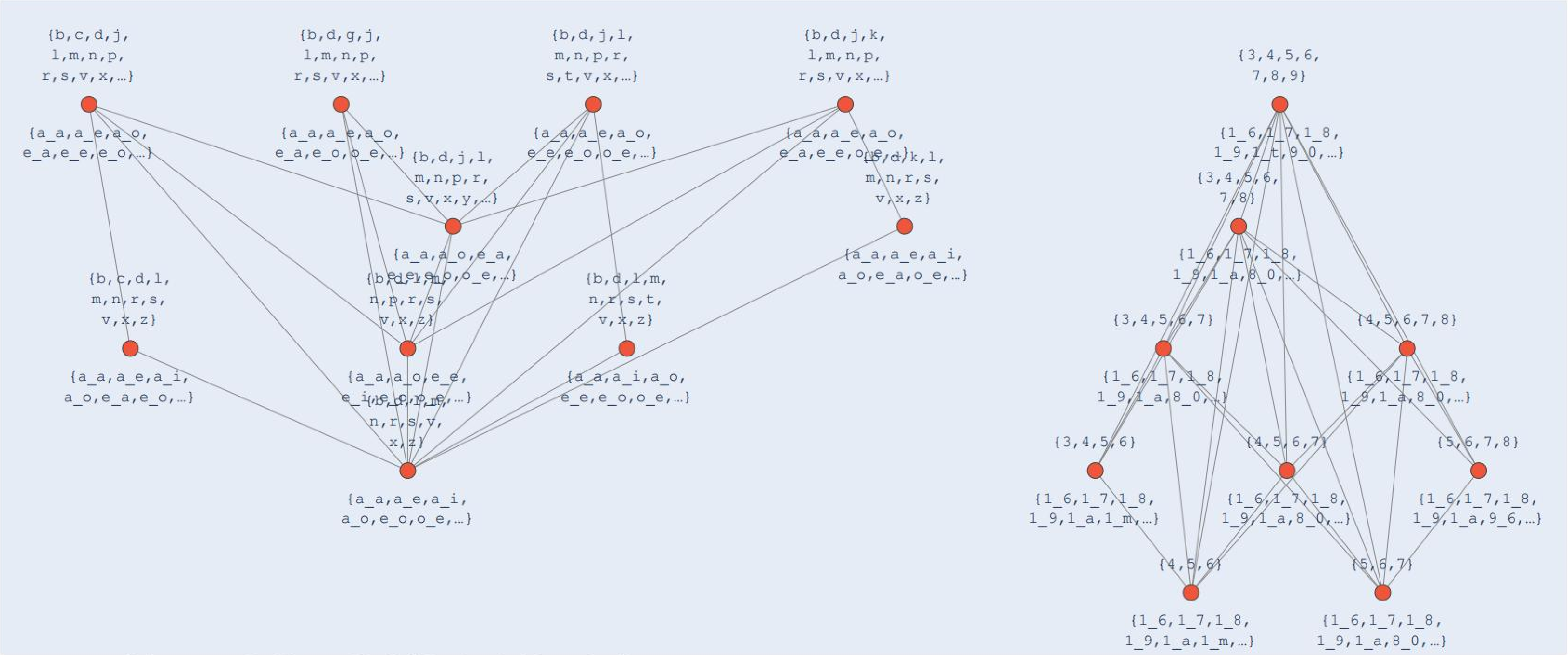
×



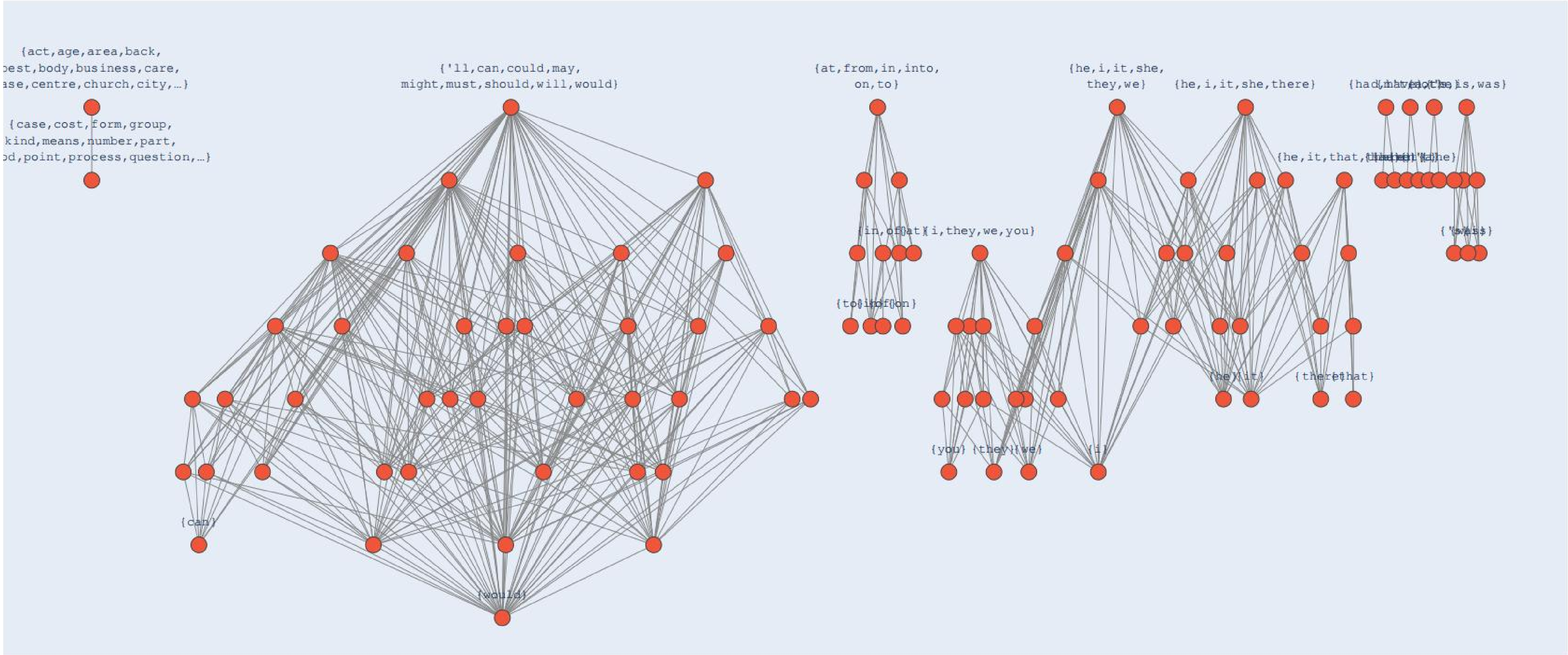
=



Formal Concepts


{b,c,d,j, l,m,n,p, r,s,v,x,…}
{a_a,a_e,a_o, e_a,e_e,e_o,…}
{b,d,g,j, l,m,n,p, r,s,v,x,…}
{a_a,a_e,a_o, e_a,e_o,o_e,…}
{b,d,j,l, m,n,p,r, s,t,v,x,…}
{a_a,a_e,a_o, e_e,e_o,o_e,…}
{b,d,j,k, l,m,n,p, r,s,v,x,…}
{a_a,a_e,a_o, e_a,e_e,o_e,…}
{b,d,j,l, m,n,p,r, s,v,x,y,…}
{a_a,a_o,e_a, e_e,e_o,o_e,…}
{b,d,k,l, m,n,r,s, v,x,z}
{a_a,a_e,a_i, a_o,e_a,o_e,…}
{b,c,d,l, m,n,r,s, v,x,z}
{a_a,a_e,a_i, a_o,e_a,e_o,…}
{b,d,l,m, n,p,r,s, v,x,z}
{a_a,a_o,e_e, e_i,e_o,o_e,…}
{b,d,l,m, n,r,s,t, v,x,z}
{a_a,a_i,a_o, e_e,e_o,o_e,…}
{b,d,l,m, n,r,s,v, x,z}
{a_a,a_e,a_i, a_o,e_o,o_e,…}
{3,4,5,6, 7,8,9}
{1_6,1_7,1_8, 1_9,1_t,9_0,…}
{3,4,5,6, 7,8}
{1_6,1_7,1_8, 1_9,1_a,8_0,…}
{3,4,5,6,7}
{1_6,1_7,1_8, 1_9,1_a,8_0,…}
{4,5,6,7,8}
{1_6,1_7,1_8, 1_9,1_a,8_0,…}
{3,4,5,6}
{1_6,1_7,1_8, 1_9,1_a,1_m,…}
{4,5,6,7}
{1_6,1_7,1_8, 1_9,1_a,8_0,…}
{5,6,7,8}
{1_6,1_7,1_8, 1_9,1_a,9_6,…}
{4,5,6}
{1_6,1_7,1_8, 1_9,1_a,1_m,…}
{5,6,7}
{1_6,1_7,1_8, 1_9,1_a,8_0,…}

Formal Concepts Words


{act,age,area,back,
best,body,business,care,
ase,centre,church,city,…}
{case,cost,form,group,
kind,means,number,part,
od,point,process,question,…}
{'ll,can,could,may,
might,must,should,will,would}
{can}
{would}
{at,from,in,into,
on,to}
{in,of}
{at}
{i,they,we,you}
{to}
{in}
{of}
{on}
{you}
{they}
{we}
{i}
{he,i,it,she,
they,we}
{he,i,it,she,there}
{he,it,that,
{he}
{it}
{there}
{that}
{had,
is,was}