Other helpful resources

- HD Computing/VSA seminar (2021): https://redwood.berkeley.edu/courses/computing-with-high-dimensional-vectors/
- Pentti Kanerva's lecture: https://www.youtube.com/watch?v=oB_mHCurNCI
- Ryan Moughan's lecture
  - Slides: https://redwood.berkeley.edu/wp-content/uploads/2021/08/Word-Embeddings-with-HD-Computing_VSA.pdf
  - Video: https://drive.google.com/file/d/1vXO4wtBI2swI6uQUew3Y3NARM6GHXV8f/view

Schedule (provisional)

Date	Week	Module	Speaker(s)
9/1	1	Introduction to computing with high-dimensional vectors	Bruno Olshausen & Pentti Kanerva
9/8	2	Overview of different HD Computing/VSA models	Denis Kleyko
9/15	3	Semantic vectors	Ryan Moughan
9/22	4	Representation and manipulation of data structures	Denis Kleyko
9/29	5	Resonator Networks	Paxon Frady
10/6	6	Analogical reasoning	Ross Gayler
10/13	7	Connections to information theory	Fritz Sommer
10/20	8	Locality-preserving encodings: representing continuous values and functions	Chris Kymn
10/27	9	Solving classification problems	Laura Galindez Olascoaga
11/3	10	Relations to neural networks	Denis Kleyko
11/10	11	Hardware implementations	Mohamed Ibrahim
11/17	12	Applications: Communication	Ping-Chen Huang

The GOFAI <-> Connectionism Debate

Connectionism and cognitive architecture: A critical analysis*

JERRY A. FODOR
CUNY Graduate Center

ZENON W. PYLYSHYN
University of Western Ontario

Connectionism, Constituency, and the Language of Thought

Paul Smolensky

Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work

JERRY FODOR*
Rutgers University and CUNY Graduate Center

BRIAN P. McLAUGHLIN
Rutgers University

THE CONSTITUENT STRUCTURE OF CONNECTIONIST MENTAL STATES: A REPLY TO FODOR AND PYLYSHYN

Paul Smolensky
University of Colorado at Boulder

Jackendoff 2002:

3.5 Four challenges for cognitive neuroscience[27]

Just as linguistic structures like Fig. 1.1 are functional characterizations that require neural instantiation, so the functional regularities that we state as rules of grammar must be neurally instantiated. To repeat a point from Chapter 2: although a great deal is known about functional localization of various aspects of language in the brain, I think it is fair to say that nothing at all is known about how neurons instantiate the details of rules of grammar. In fact, we don't even have any idea of how a single speech sound such as /p/—much less a category like NP—is instantiated in neural firings or synaptic connections. The rest of this chapter will lay out four challenges that linguistic combinatoriality and rules of language present to theories of brain function—challenges that to my knowledge have not been widely recognized in the cognitive neuroscience community.

Vector Symbolic Architectures Answer Jackendoff's Challenges for Cognitive Neuroscience

Ross W. Gayler (r.gayler@mbox.com.au)

Computing with high-dimensional vectors

Concepts, variables, attributes are represented as high-dimensional vectors (e.g., 10,000 bits)

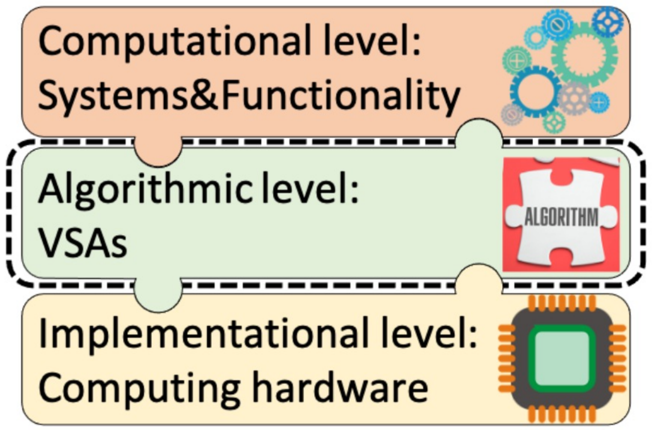
Three fundamental operations:

- multiplication (binding)
- addition (combining)
- permutation (sequencing)

Approximates a *field*

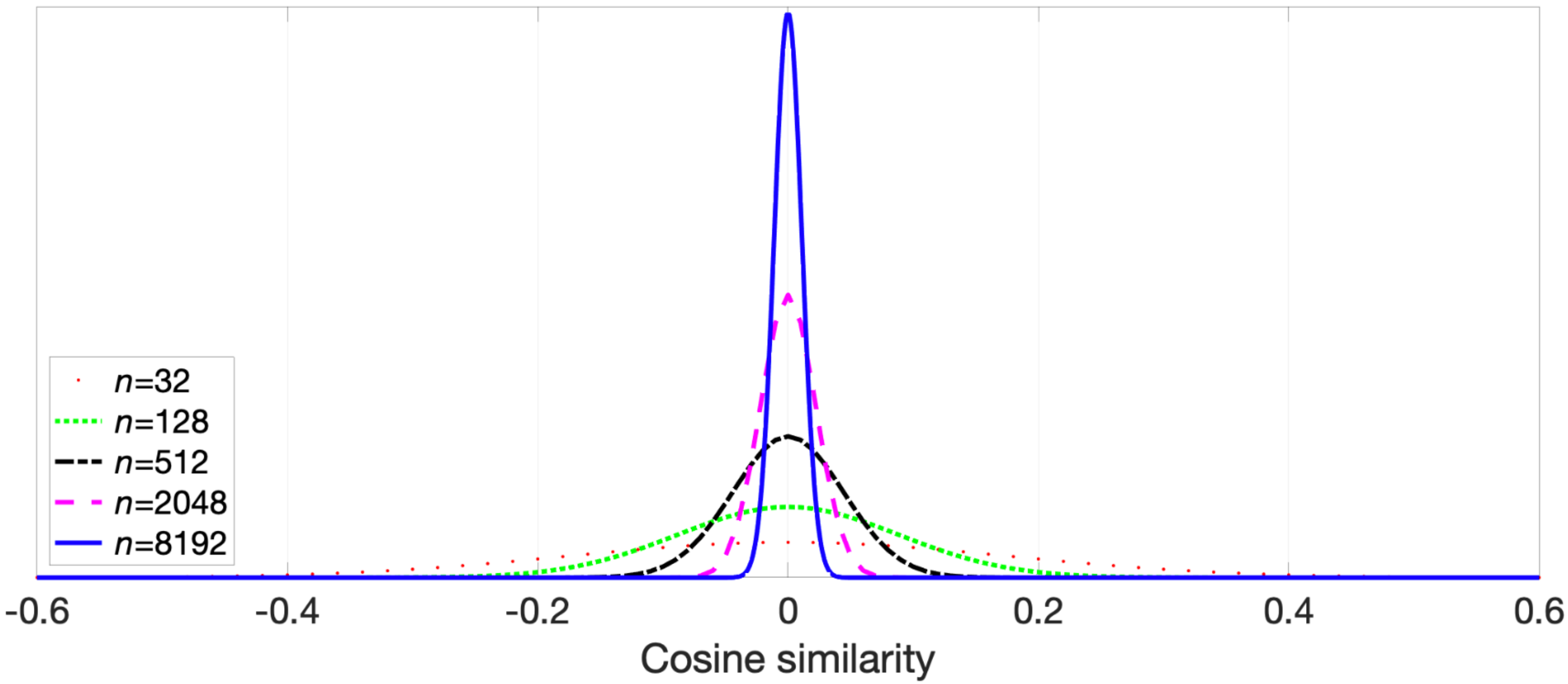
Kanerva, P (2009). Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors. *Cognitive Computing, 1*: 139-159.

Many instantiations of VSA vectors/operations exist



HDC/VSA	Ref.	Space of atomic HVs	Binding	Unbinding	Superposition	Similarity
TPR	[Smolensky, 1990]	unit HVs	tensor product	tensor-vector inner product	component-wise addition	sim_{dot}
HRR	[Plate, 1995a]	unit HVs	circular convolution	circular correlation	component-wise addition	sim_{dot}
FHRR	[Plate, 2003]	complex unitary HVs	component-wise multiplication	component-wise multiplication with complex conjugate	component-wise addition	sim_{cos}
SBDR	[Rachkovskij, 2001]	sparse binary HVs	context-dependent thinning	repeated context-dependent thinning	component-wise disjunction	sim_{dot}
BSC	[Kanerva, 1997]	dense binary HVs	component-wise XOR	component-wise XOR	majority rule	$dist_{Ham}$
MAP	[Gayler, 1998]	dense bipolar HVs	component-wise multiplication	component-wise multiplication	component-wise addition	sim_{cos}
MCR	[Snaider and Franklin, 2014]	dense integer HVs	component-wise modular addition	component-wise modular subtraction	component-wise discretized vector sum	modified Manhattan
MBAT	[Gallant and Okaywe, 2013]	dense bipolar HVs	vector-matrix multiplication	multiplication with inverse matrix	component-wise addition	sim_{dot}
SBC	[Laiho et al., 2015]	sparse binary HVs	block-wise circular convolution	block-wise circular convolution with approximate inverse	component-wise addition	sim_{dot}
GAHRR	[Aerts et al., 2009]	unit HVs	geometric product	geometric product with inverse	component-wise addition	unitary product

Concentration of measure: Pseudo-orthogonality in high-dimensional space



. Encoding a **set** with addition: *s* = {*a*,*b*,*c*}

S = **A** + **B** + **C**

. Encoding a **data record** with a set of bound pairs: *d* = ‘(*x* = *a*) & (*y* = *b*) & (*z* = *c*)'

D = **X*A** + **Y*B** + **Z*C**

. Extracting the value of *x* from the record:

```
X*D = X * (X*A + Y*B + Z*C)
    = X*X*A + X*Y*B + X*Z*C
    = X*X*A + (X*Y*B + X*Z*C)
    =     A +      noise
    ≈ A
```

. Encoding a **sequence** with rotation and multiplication: (a,b)

$$\mathbf{AB} = r\mathbf{A} * \mathbf{B}$$

. Extending **AB** with **C**: (a,b,c)

$$\begin{aligned}\mathbf{ABC} &= r(\mathbf{AB}) * \mathbf{C} \\ &= rr\mathbf{A} * r\mathbf{B} * \mathbf{C}\end{aligned}$$

. Extracting the first element of **ABC**:

$$\begin{aligned}ss(\mathbf{ABC} * \mathbf{BC}) &= ss(rr\mathbf{A} * r\mathbf{B} * \mathbf{C} * r\mathbf{B} * \mathbf{C}) \\ &= ss(rr\mathbf{A}) \\ &= \mathbf{A}\end{aligned}$$

where s is the inverse (counter-rotate) of r

Encoding of ‘(X=A) and (Y=B) and (Z=C)’

```
                 X   =  1 0 0 1 0 ... 0 1
                 A   =  0 0 1 1 1 ... 1 1
                ------------------------------
bind with XOR (*)  X*A =  1 0 1 0 1 ... 1 0   →   1 0 1 0 1 ... 1 0   ‘(x = a)’

                 Y   =  1 0 0 0 1 ... 1 0
                 B   =  1 1 1 1 1 ... 0 0
                ------------------------------
                 Y*B =  0 1 1 1 0 ... 1 0   →   0 1 1 1 0 ... 1 0   ‘(y = b)’

                 Z   =  0 1 1 0 1 ... 0 1
                 C   =  1 0 0 0 1 ... 0 1
                ------------------------------
                 Z*C =  1 1 1 0 0 ... 0 0   →   1 1 1 0 0 ... 0 0   ‘(z = c)’
                                               ------------------------------
                                                2 2 3 1 1 ... 2 0   sum
                                          H   = 1 1 1 0 0 ... 1 0   sum > 3/2
                                          X   = 1 0 0 1 0 ... 0 1   ‘unbind’
                                          ---------------------------
                                          X*H = 0 1 1 1 0 ... 1 1   =  A’ ≅ A
                                                       ↓
                                          Item/clean-up memory
                                          finds nearest neighbor
                                          among known vectors
                                                       ↓
                                                0 0 1 1 1 ... 1 1   =  A
```

Three Examples

- Analogical reasoning
- Language identification via trigram statistics
- Semantic modeling with word vectors

Analogical Reasoning

What is the dollar of Mexico?

Analogical Mapping with Multiplication by Hypervector

What is the Dollar of Mexico?

Encoding of **USA** and **MEX**ico: **Na**me of country, **Ca**pital city, **Mon**etary unit

USA = Nam*Us + Cap*Dc + Mon*$
MEX = Nam*Mx + Cap*Mc + Mon*P

Pairing up the two--binding

Pair = USA*MEX

Analyzing the pair

Pair = Us*Mx + Dc*Mc + $*P + noise

Literal interpretation of *Dollar of Mexico* produces nonsense:

```
$*MEX = $ * (Nam*Mx + Cap*Mc + Mon*P)
      = $*Nam*Mx + $*Cap*Mc + $*Mon*P
      =  noise   +  noise   +  noise
         (nothing cancels out)
```

However, what in Mexico corresponds to Dollar in USA?

```
$*Pair = $ * (USA*MEX)
       = $ * (Us*Mx + Dc*Mc + $*P + noise)
       = $*Us*Mx + $*Dc*Mc + $*$*P + $*noise
       =  noise  +  noise  +     P +  noise
       = P + noise
       ≈ P
```

Other examples of analogical reasoning

Gayler & Levy (2007), Analogical Mapping with Vector Symbolic Architectures

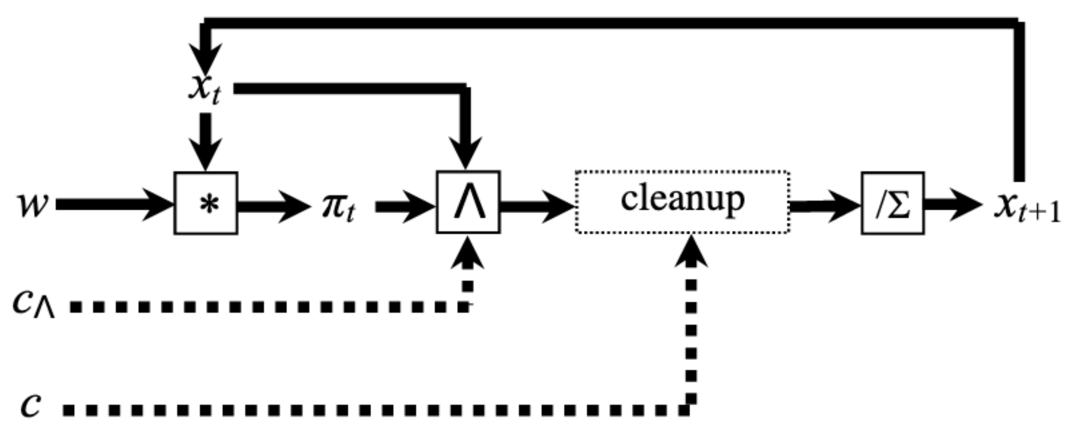


Figure 3. A neural circuit for graph isomorphism.

Plate (1994) thesis, Chapter 6: Estimating analogical similarity

Probe: Spot bit Jane, causing Jane to flee from Spot.

Episodes in long-term memory:		OA	FOR	HOR	RFB	HOS	OLI	Type
		Aspects of similarity						
E1:	Fido bit John, causing John to flee from Fido.	✓	✓	✓	✓	✓	✓	LS
E2:	Fred bit Rover, causing Rover to flee from Fred.	✓	✓	✓	×	✓	✓	AN^{cm}
E3:	Felix bit Mort, causing Mort to flee from Felix.	×	✓	✓	×	✓	✓	AN_1
E4:	Mort bit Felix, causing Felix to flee from Mort.	×	✓	✓	×	✓	✓	AN_2
E5:	Rover bit Fred, causing Rover to flee from Fred.	✓	✓	✓	$\frac{1}{2}$	✓	×	$SS^{\times I}$
E6:	John fled from Fido, causing Fido to bite John.	✓	✓	✓	✓	×	✓	$SS^{\times H}$
E7:	Mort bit Felix, causing Mort to flee from Felix.	×	✓	✓	×	✓	×	$FA^{\times I}$
E8:	Mort fled from Felix, causing Felix to bite Mort.	×	✓	✓	×	×	✓	$FA^{\times H}$
E9:	Fido bit John, John fled from Fido.	✓	✓	×	✓	×	✓	SS^{-H}
E10:	Fred stroked Rover, causing Rover to lick Fred.	✓	×	✓	×	×	×	OO_1
E11:	Fred stroked Rover, Rover licked Fred.	✓	×	×	×	×	×	OO_2

Kent & Maudgalya (2020), Vector symbolic scene analogies

A : B　　C : D

Language identification from trigram statistics
(Joshi, Halseth, Kanerva 2017)

Encode a trigram vector for each three-letter sequence A, B, C as

$$ABC = \rho(\rho(A)) * \rho(B) * C = \rho\rho A * \rho B * C$$

Add all trigram vectors of a text into a 10,000-D Profile Vector. For example, the text segment

"the quick brown fox jumped over ..."

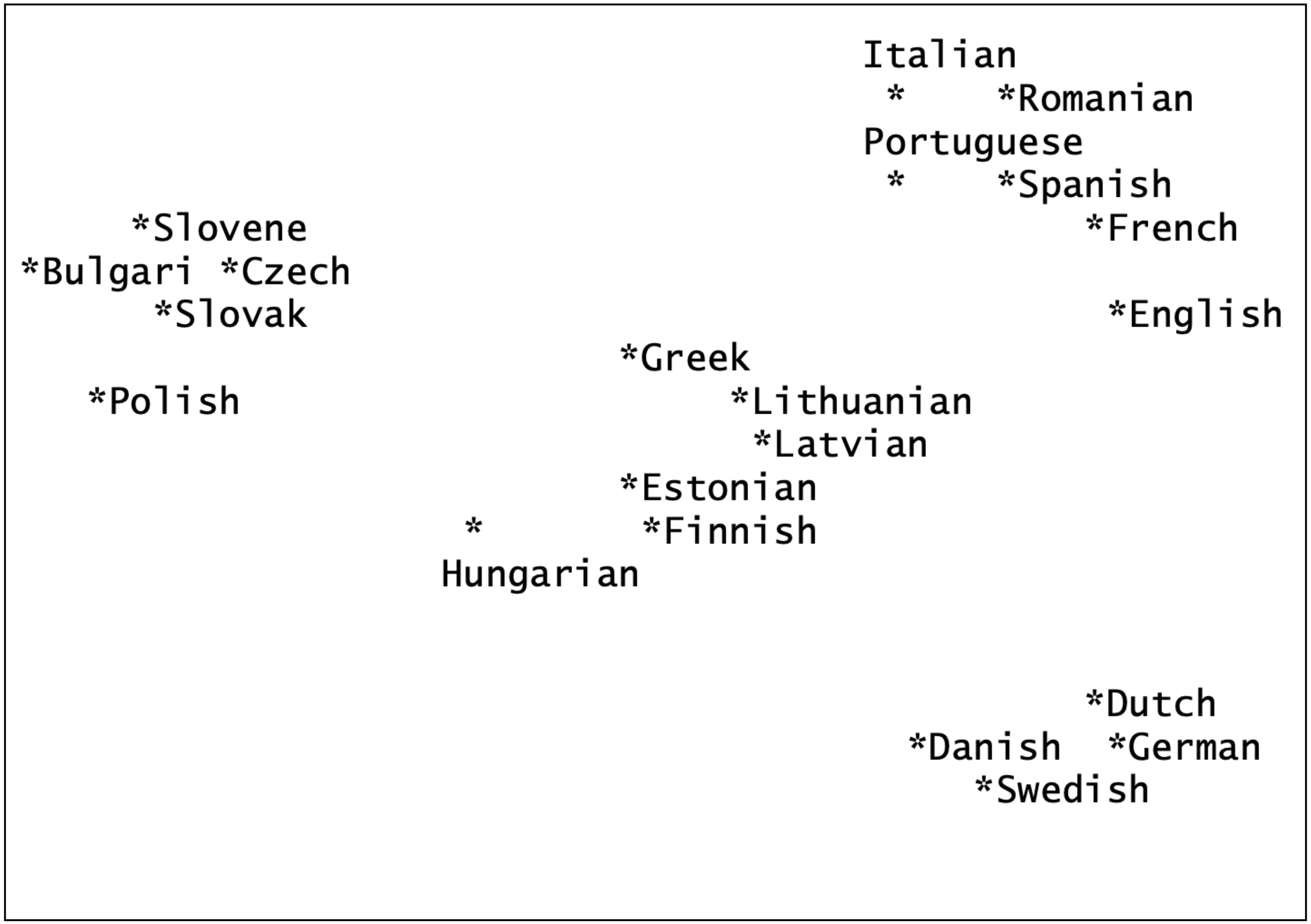
gives rise to the following trigram vectors, which are added into the profile for English

Engl += THE + HE# + E#Q + #QU + QUI + UIC + ...

	ell	eng	ita	ces	est	spa	nld	por	lav	lit	ron	pol	fra	bul	deu	dan	fin	hun	swe	slk	slv
ell	987	1	.	.	.	.	3	3	.	.	.	1	.	4	.	.	1	.	.	.	.
eng	2	982	.	4	.	.	1	.	2	.	.	.	6	.	.	1	.	2	.	.	.
ita	.	.	992	.	1	2	.	.	.	.	2	3	.	.	.	.	.	.	.	.	.
ces	1	1	.	940	1	.	.	.	1	1	1	1	.	5	1	.	.	.	.	35	12
est	1	.	.	1	983	.	.	.	3	.	.	.	3	.	1	1	5	1	1	.	.
spa	.	.	6	.	.	946	2	30	8	1	2	.	5	.	.	.	.	.	.	.	.
nld	.	1	.	.	.	.	980	1	.	.	2	1	.	.	5	9	.	.	1	.	.
por	.	1	2	.	.	1	1	991	.	.	.	.	3	1	.	.	.	.	.	.	.
lav	2	.	.	1	.	.	.	2	963	26	.	2	.	2	.	1	.	.	.	1	.
lit	2	.	1	2	1	1	.	2	18	969	.	.	1	.	.	.	.	.	.	1	2
ron	.	.	1	.	.	1	.	2	.	1	987	2	4	2	.	.	.	.	.	.	.
pol	2	1	.	3	1	.	.	.	.	.	.	984	.	4	.	.	.	.	.	4	1
fra	3	.	2	.	.	4	2	1	1	2	1	.	982	.	.	1	.	.	.	1	.
bul	1	.	.	7	.	.	4	.	.	.	.	.	.	984	.	.	.	.	.	3	1
deu	.	2	1	1	.	.	3	.	.	.	.	.	3	.	985	4	.	.	1	.	.
dan	.	2	.	.	.	.	9	.	.	.	.	.	2	.	.	974	.	.	13	.	.
fin	.	.	.	.	4	.	2	.	1	.	.	.	.	.	.	.	993	.	.	.	.
hun	.	.	.	.	.	.	6	1	1	1	.	.	.	.	.	2	.	989	.	.	.
swe	.	1	.	.	.	1	5	.	.	.	4	.	1	.	4	10	.	.	974	.	.
slk	2	.	.	72	.	.	1	.	2	1	4	18	.	6	1	.	.	.	.	881	12
slv	1	.	.	5	2	.	.	1	.	.	1	.	.	6	1	1	.	.	.	.	982

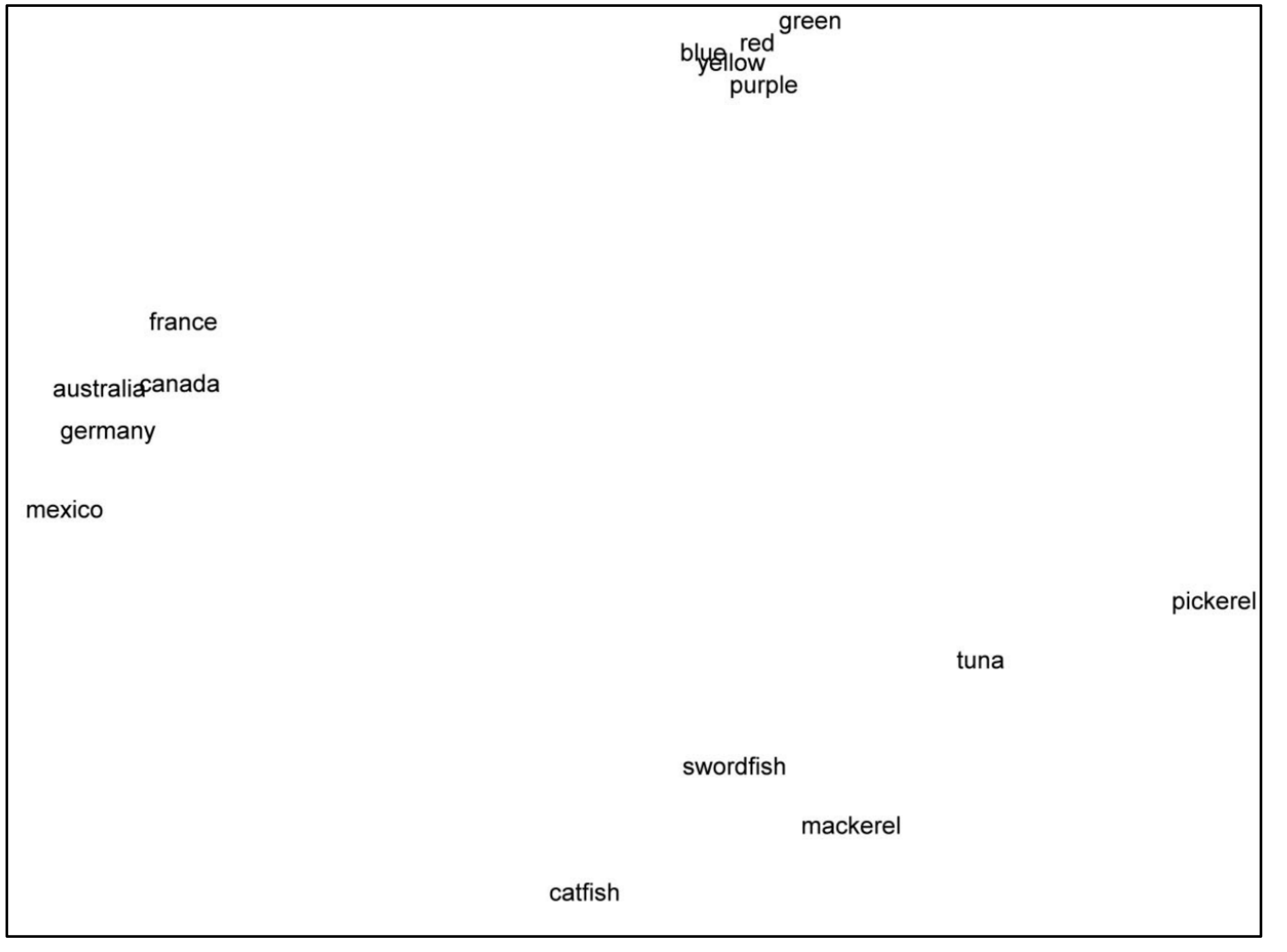
LEGEND: bul = Bulgarian, ces = Czech, dan = Danish, deu = German, ell = Greek, eng = English, est = Estonian, fin = Finnish, fra = French, hun = Hungarian, ita = Italian, lav = Latvian, lit = Lithuanian, nld = Dutch, pol = Polish, por = Portuguese, ron = Romanian, slk = Slovak, slv = Slovene, spa = Spanish, swe = Swedish.

Visualizing similarity of languages



Learning Word Embeddings

- Key idea: **words with similar meaning will occur in similar situations** → learn meaning from statistics of different situations
  - Known as *distributional hypothesis*
- Second key idea: use high-dimensional vectors (embeddings) to encode similarity



Why use HD computing/VSA for word models?

Latent Semantic Analysis

- Based on computing SVD of co-occurrence matrix

Downsides:

- SVD is computationally expensive
- Difficult to update with new entries
- Requires holding co-occurrence matrix in memory

HD computing methods

- Word embedding updates based on random context/order vectors

Upsides:

- Easy to compute
- Easy to update
- Never stores co-occurrence matrix

Prerequisite steps

- Define random “label” vectors
  - Sparse ternary: most values 0, small percentage are either +1 or -1
  - Holographic Reduced Representations: components drawn from Gaussian with mean 0 and variance 1/d (d=vector dimension)
- Mark most frequent words
- Lemmatization
  - (e.g., {'walk', 'walked', 'walks’, 'walking’} -> walk)

Context and Order Information (BEAGLE)

Example sentence: “a dog bit the mailman”

a, the, bit, mailman, Φ are fixed random label vectors (HRR*)

Context:

[dog] = a + 0 + *bit* + the + *mailman*

→ 0 + 0 + *bit* + 0 + *mailman*

(Don’t add frequent words)

= *bit* + *mailman*

Order:

$$
\begin{aligned}
<\mathrm{dog}>_1 &= \mathrm{a} * \Phi \\
<\mathrm{dog}>_2 &= \Phi * \mathrm{bit} \\
<\mathrm{dog}>_3 &= \mathrm{a} * \Phi * \mathrm{bit} \\
<\mathrm{dog}>_4 &= \Phi * \mathrm{bit} * \mathrm{the} \\
<\mathrm{dog}>_5 &= \mathrm{a} * \Phi * \mathrm{bit} * \mathrm{the} \\
<\mathrm{dog}>_6 &= \Phi * \mathrm{bit} * \mathrm{the} * \mathrm{mailman} \\
<\mathrm{dog}>_7 &= \mathrm{a} * \Phi * \mathrm{bit} * \mathrm{the} * \mathrm{mailman}
\end{aligned}
$$

$$<\mathrm{dog}> = <\mathrm{dog}>_1 + <\mathrm{dog}>_2 + \cdots + <\mathrm{dog}>_7$$

Average over many sentences

$$\mathrm{DOG} = [\mathrm{DOG}] + <\mathrm{DOG}>$$

*Holographic Reduced Representations

Context and Order Information (RI)

Example sentence: “a dog bit the mailman”

a, the, bit, mailman, Φ are fixed random label vectors (sparse ternary)

Context:

[dog] = a + 0 + *bit* + the + *mailman*

→ 0 + 0 + *bit* + 0 + *mailman*

= *bit* + *mailman*

Order:

$$\langle \text{dog} \rangle = (\Pi^{-1}\text{a}) + 0 + (\Pi\,\text{bit}) + (\Pi^2\,\text{the}) + (\Pi^3\,\text{mailman})$$

Average over many sentences

$$\text{DOG} = [\text{DOG}] + \langle \text{DOG} \rangle$$

Example application: Sentence completion

An Example of Changing Word Activation With Both Context and Order Information

Probe	jefferson	edison	aquinas	paine	pickney	malthus
Thomas ____	**.72**	.66	.60	.35	.46	.34
Thomas ____ wrote the Declaration of Independence	**.44**	.30	.24	.29	.17	.14
Thomas ____ made the first phonograph	.33	**.45**	.29	.17	.21	.12
Thomas ____ taught that all civil authority comes from God	.30	.26	**.40**	.13	.17	.12
Thomas ____ is the author of *Common Sense*	.29	.21	.19	**.43**	.18	.13
A treaty was drawn up by the American diplomat Thomas ____	.32	.26	.27	.17	**.92**	.15
Thomas ____ wrote that the human population increased faster than the food supply	.23	.22	.21	.12	.14	**.41**

Example application: Sentence completion

Phrase	Activations
the *[brainstem]*	first (.95) best (.93) latter (.92) next (.90) same (.89) following (.94)
the *[brainstem]* is	latter (.58) following (.54) same (.54) first (.53) world (.53) best (.53)
the *[brainstem]* is much larger and more complex than the spinal cord	brainstem (.37) latter (.22) world (.20) epidermis (.19) following (.19) same (.18)
although *[ostriches]*	arguably (.26) profit (.20) single (.17) double (.17) moderate (.17) thorough (.16)
although *[ostriches]* cannot	wastefulness (.18) quasars (.17) prison (.16) prayer (.15) buckler (.14) diastase (.14)
although *[ostriches]* cannot fly they have other skills	ostriches (.24) diastase (.13) wastefulness (.13) consecutive (.12) keen (.12) buckler (.12)
electric *[eel]*	shocks (.33) motors (.33) current (.32) charges (.25) sparks (.23) generators (.18)
an electric *[eel]*	current (.36) shock (.25) generator (.23) charges (.23) swiftness (.20) eel (.20)
an electric *[eel]* can produce a discharge of several hundred volts	eel (.19) painters (.12) flexplate (.12) methanol (.12) current (.12) pcp (.12)
emperor *[penguins]*	mutsuhito (.39) trajan (.31) justinian (.29) honorius (.26) yuan (.20) claudius (.19)
[penguins] have	archaeologists (.45) scientists (.42) we (.40) biologists (.38) geologists (.36) researchers (.34)
the emperor *[penguins]* have come to their breeding grounds	penguins (.24) trajan (.14) would (.13) leninists (.13) could (.12) researchers (.12)

Aside: how does this compare to Word2Vec?

Continuous Bag of Words
(similar to Order)

Skipgram
(similar to Context)