Distributional Models for Lexical Semantics

Lecture 2: Don't be afraid of matrices

Denis Paperno

TyLex – Voronovo 2017

Vectors and matrices

- Vectors are structures of **n** numbers

  <1,2,3,4,5,6>

- Matrices are structures of **n**x**m** numbers
  - Example: 2x3 matrix

1	2	3
4	5	6

Vectors and matrices

- One word vector:

dog	246	72	78	71	1

- Cooccurrence vectors of mutiple words form a **n**x**m** cooccurrence matrix

	bark	walk	talk	tail	bag
dog	246	72	78	71	1
cat	5	15	25	32	0
man	0	57	133	0	1
woman	2	203	407	5	18

Vectors and matrices

- An n-dimensional vector can be represented as a matrix:
  - an **n**x**1** matrix

1	2	3

  - or a **1**x**n** matrix

1
2
3

Vector multiplication

- We have used vector multiplication as part of the definition of the cosine:

$$v \cdot u = \sum_i v_i * u_i$$

equivalent to multiplication

of a 1xn and an nx1 matrices:

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} * \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Vector multiplication

- We have used vector multiplication as part of the definition of the cosine:

$$v \cdot u = \sum_i v_i * u_i$$

equivalent to multiplication

of a 1xn and an nx1 matrices:

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} * \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Matrix multiplication

matrices A and B of sizes **n**x**m** and **m**x**p**

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mp} \end{pmatrix}$$

- **AB** is a matrix of size **n**x**p**

$$\mathbf{AB} = \begin{pmatrix} (\mathbf{AB})_{11} & (\mathbf{AB})_{12} & \cdots & (\mathbf{AB})_{1p} \\ (\mathbf{AB})_{21} & (\mathbf{AB})_{22} & \cdots & (\mathbf{AB})_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{AB})_{n1} & (\mathbf{AB})_{n2} & \cdots & (\mathbf{AB})_{np} \end{pmatrix}$$

  - $(AB)_{ij}$ is the **i**th row of A *jth column of B

Matrix multiplication

- What is the result of this multiplication?

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} * \begin{bmatrix} 4 & 7 \\ 5 & 8 \\ 6 & 9 \end{bmatrix}$$

Matrix multiplication

- What is the result of this multiplication?

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} * \begin{bmatrix} 4 & 7 \\ 5 & 8 \\ 6 & 9 \end{bmatrix}$$

- 1x3 by 3x2 matrix product gives a 1x2 size

1*4+2*5+3*6	1*7+2*8+3*9

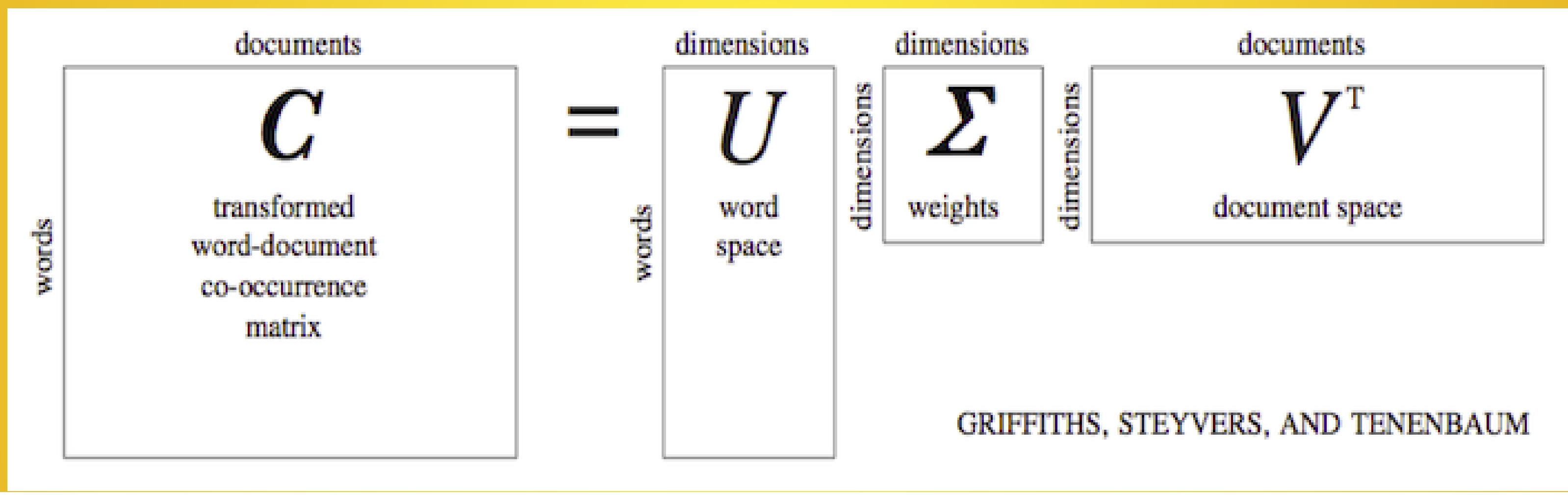
=<32,50>

Applications of matrices

- Dimensionality reduction
- Mapping distributional vector spaces:
  - From one language to another
  - From one period of time to another
- Linguistic vectors to image vectors
- Compositionality models

Dimensionality reduction in DSM

- Different matrix decomposition methods
  - SVD, NMF, LDA, neural models...



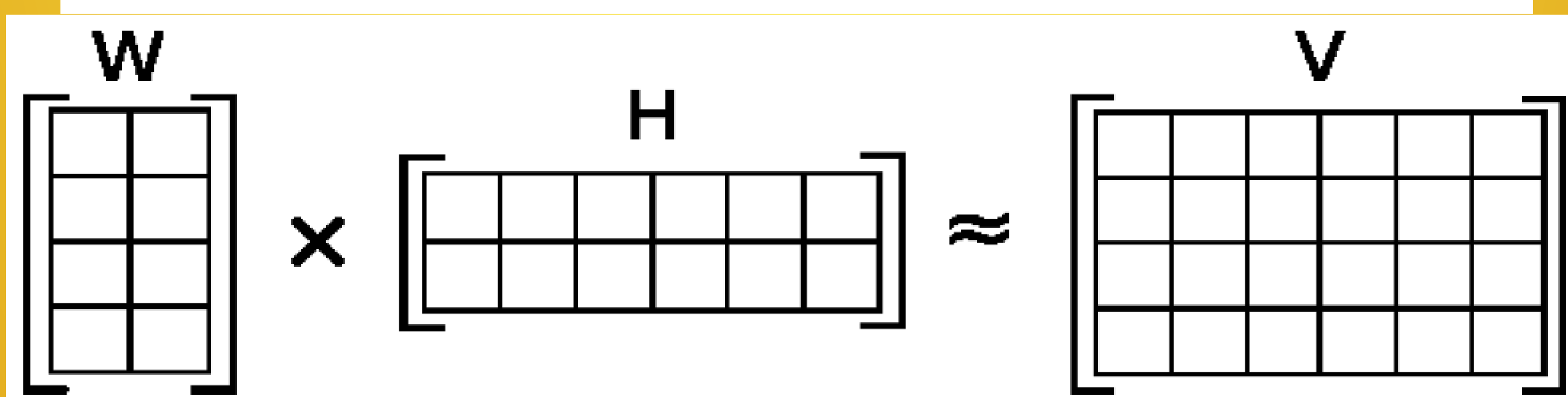
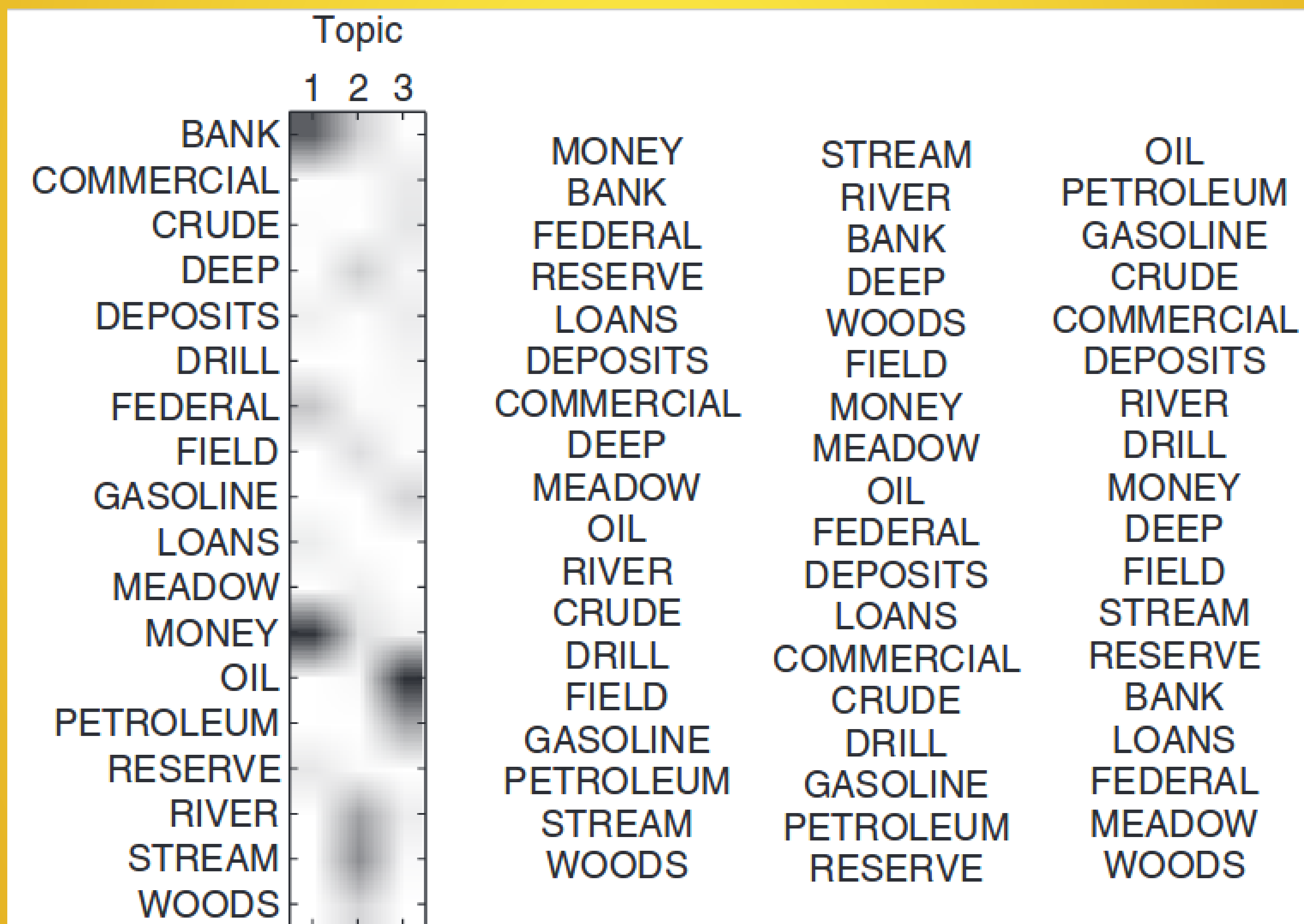


Illustration of approximate non-negative matrix factorization: the matrix V is represented

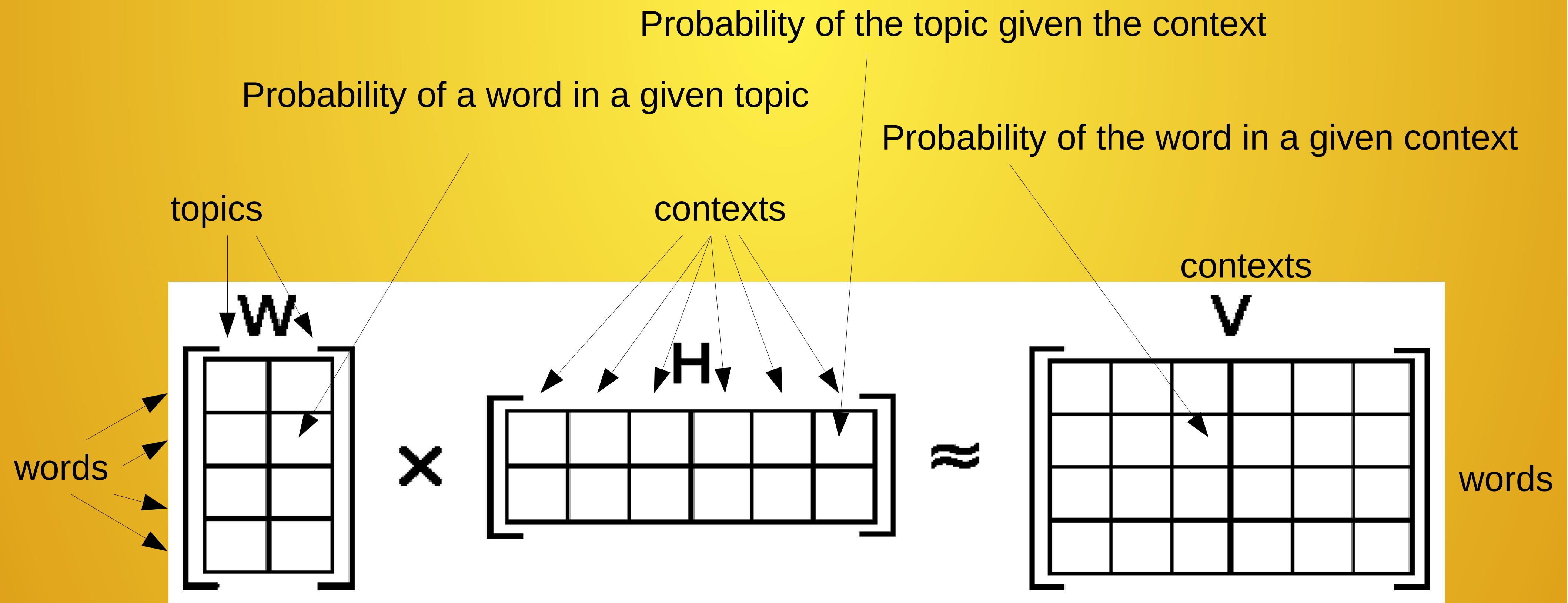
Latent dimensions as topics

- (Griffiths et al. 2007)



Probabilistic interpretation

- In Topic Models, the decomposition is meant to be interpreted as a probability distibution: $p(w|c)=\sum_t p(w|t)p(t|c)$

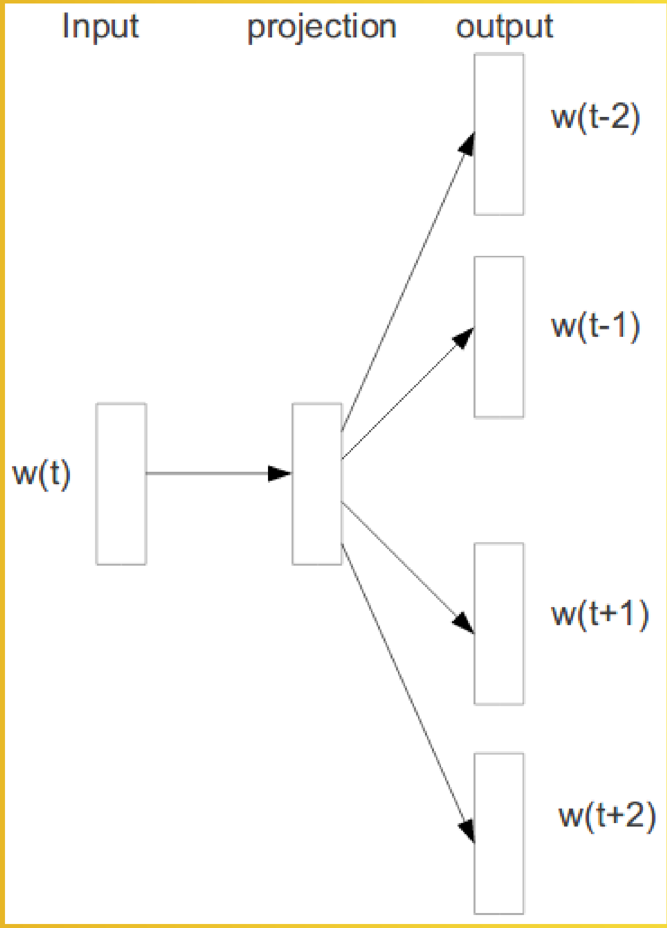


Neural word embeddings

- Recent, popular distributional semantic models based on neural networks
  - Word2vec (Mikolov et al. 2013)
  - Glove (Pennington et al. 2014)

- “embedding”=vector
  - Metaphor: words embedded in the vector space

Skip-gram model



- One of the word2vec models along with CBOW
- Formally, the model is trained at predicting context words from a given word

  $P(w,c)=\sigma(\hat{w}*\hat{c})$

  Mikolov et al. 2013

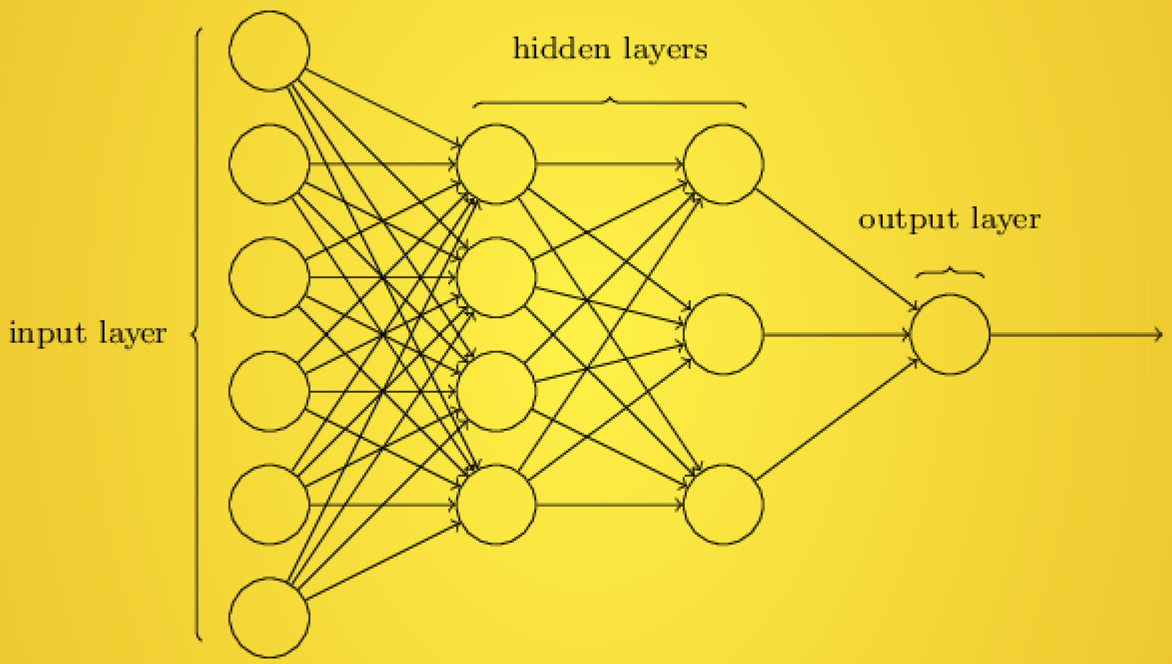
Success of neural models

- Our secret wish was to discover that it is all hype, and count vectors are far superior to their predictive counterparts. A more realistic expectation was that a complex picture would emerge, with predict and count vectors beating each other on different tasks. Instead, we found that the predict models are so good that, while the triumphalist overtones still sound excessive, there are very good reasons to switch to the new architecture.
  - Baroni et al. 2014

Results from Baroni et al. 2014

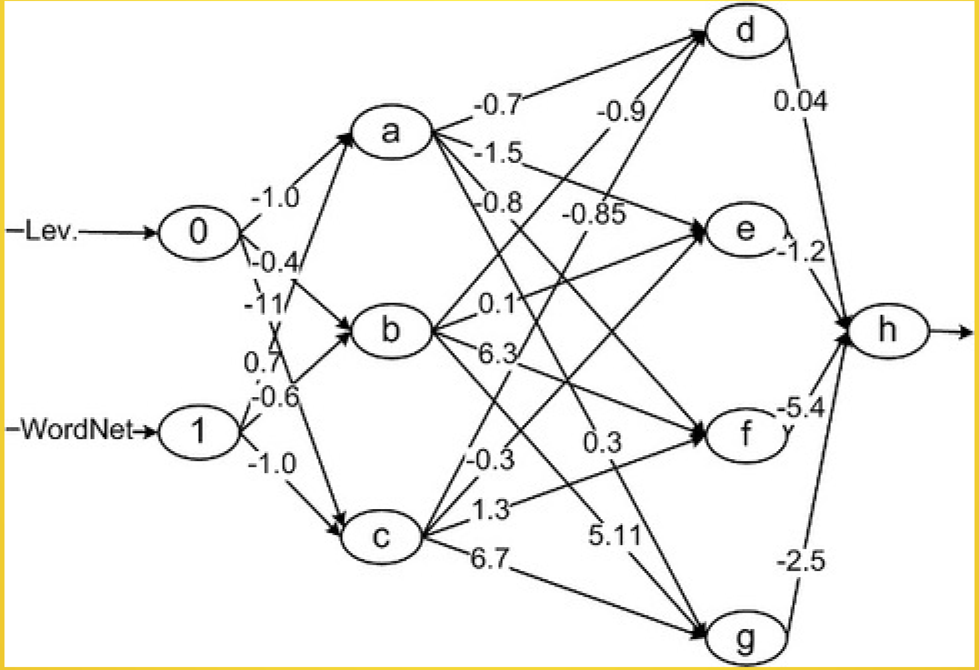
	rg	ws	wss	wsr	men	toefl	ap	esslli	battig	up	mcrae	an	ansyn	ansem
best setup on each task														
cnt	74	62	70	59	72	76	66	84	98	41	27	49	43	60
pre	84	75	**80**	**70**	**80**	91	75	86	**99**	41	28	**68**	**71**	**66**
best setup across tasks														
cnt	70	62	70	57	72	76	64	84	98	37	27	43	41	44
pre	83	73	78	68	**80**	86	71	77	98	41	26	67	69	64
worst setup across tasks														
cnt	11	16	23	4	21	49	24	43	38	-6	-10	1	0	1
pre	74	60	73	48	68	71	65	82	88	33	20	27	40	10
best setup on rg														
cnt	(74)	59	66	52	71	64	64	84	98	37	20	35	42	26
pre	(84)	71	76	64	79	85	72	84	98	39	25	66	70	61
other models														
soa	**86**	**81**	77	62	76	**100**	**79**	**91**	96	**60**	**32**	61	64	61
dm	82	35	60	13	42	77	76	84	94	51	29	NA	NA	NA
cw	48	48	61	38	57	56	58	61	70	28	15	11	12	9

Neural networks



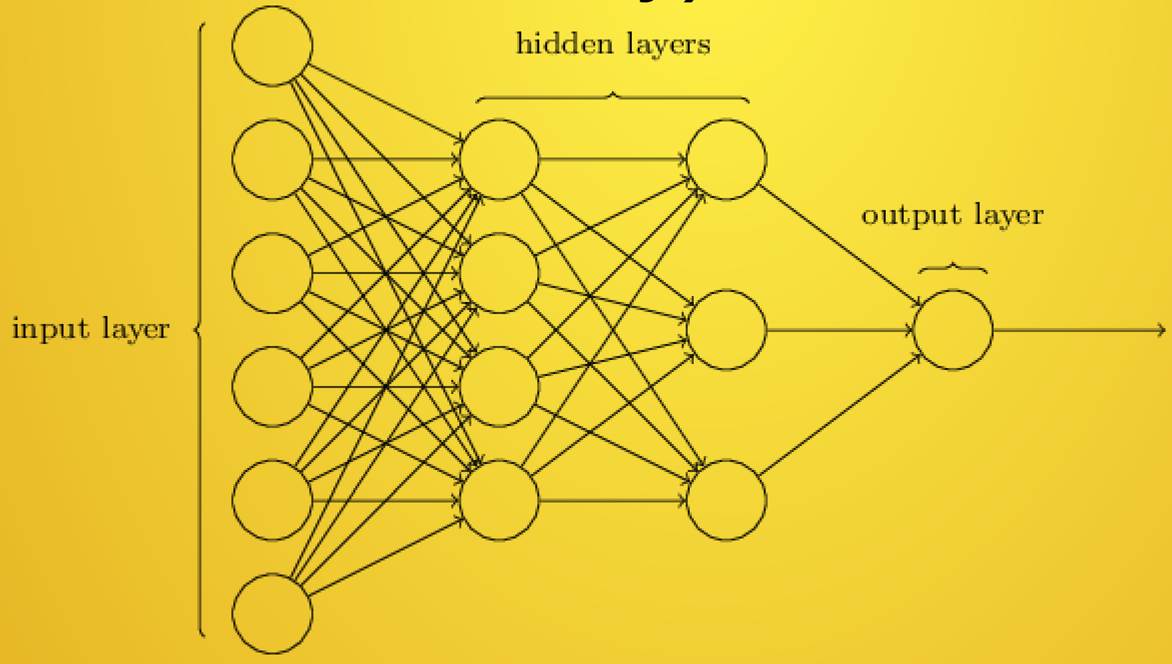
Neural network components

- Nodes (“neurons”) organized in layers
- Weighted connections between layers



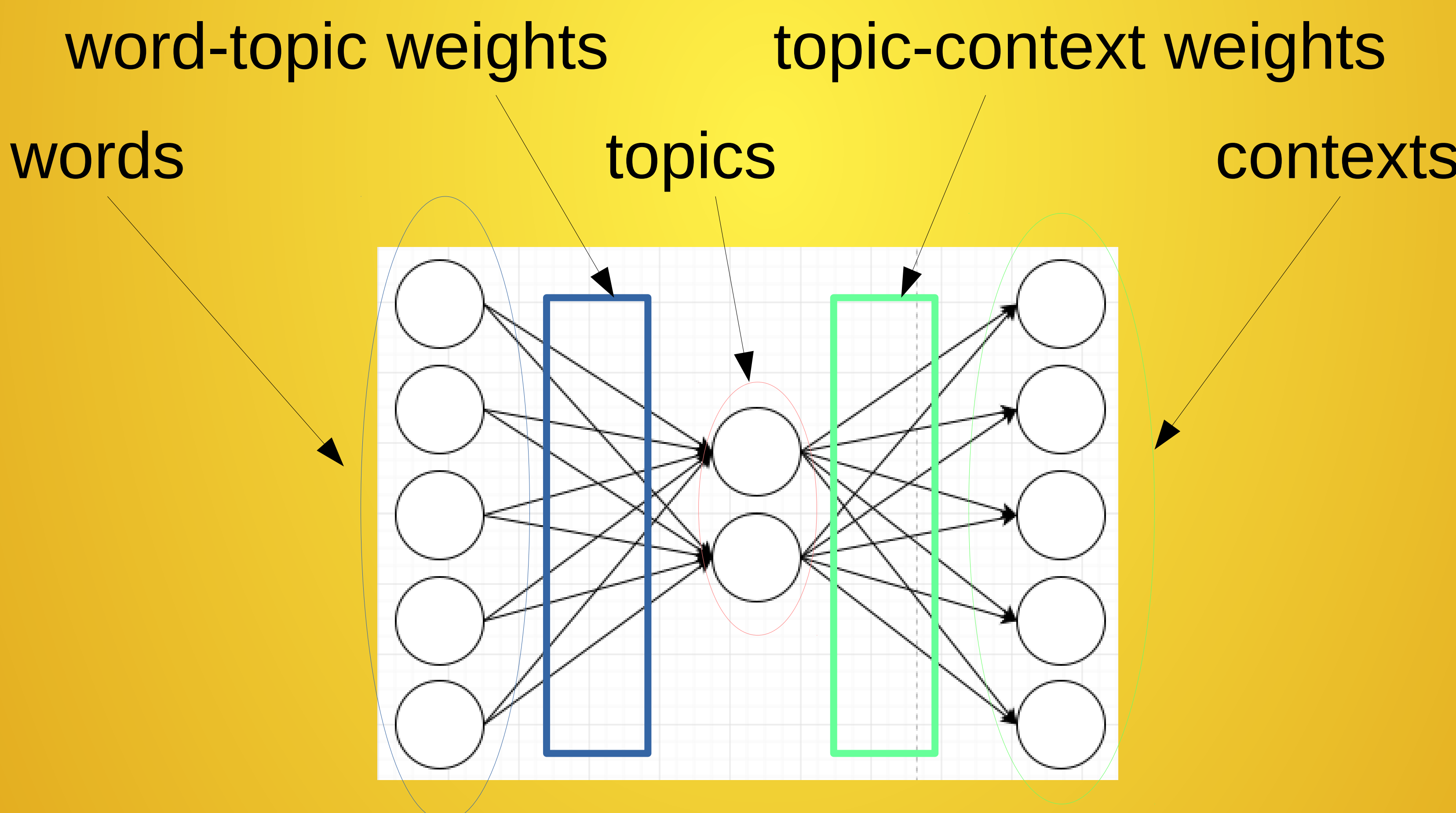
Neural networks, demystified:

- Layers = vectors
- Connections = matrices
- Signal propagation = matrix multiplication (modulo nonlinearity)



Matrix decomposition in DSM

- Matrix decomposition can be represented as a simple neural network:



GloVe

explicit matrix factorization method using neural

$$\vec{w} \cdot \vec{c} + b_w + b_c = \log\left(\#(w, c)\right) \quad \forall (w, c) \in D$$

Inside word2vec

- SG with negative sampling maximizes:

$$\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D} \left[\log \sigma(-\vec{w} \cdot \vec{c}_N)\right]$$

On average, learning will converge when

$$\vec{w} \cdot \vec{c} = \log\left(\frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)} \cdot \frac{1}{k}\right) = \log\left(\frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)}\right) - \log k$$

(Levy and Goldberg 2014)

Takehome messages about neural models

- Neural models (GloVe, word2vec) are distributional models with matrix factorization
- In particular, Skip-gram with negative sampling (SGNS) learns vectors of words and contexts to approximate (PMI(w,c)-log k)

  i.e. SGNS implicitly factorizes the (shifted) PMI matrix, like other distributional models

  Word2vec has settings other than SGNS but they perform comparably to SGNS

Using matrices as mappings

Example: Distributional onto Visual vectors

Image recognition

- Convolutional Neural Network
- Scheme from Krizhevsky et al. 2012:

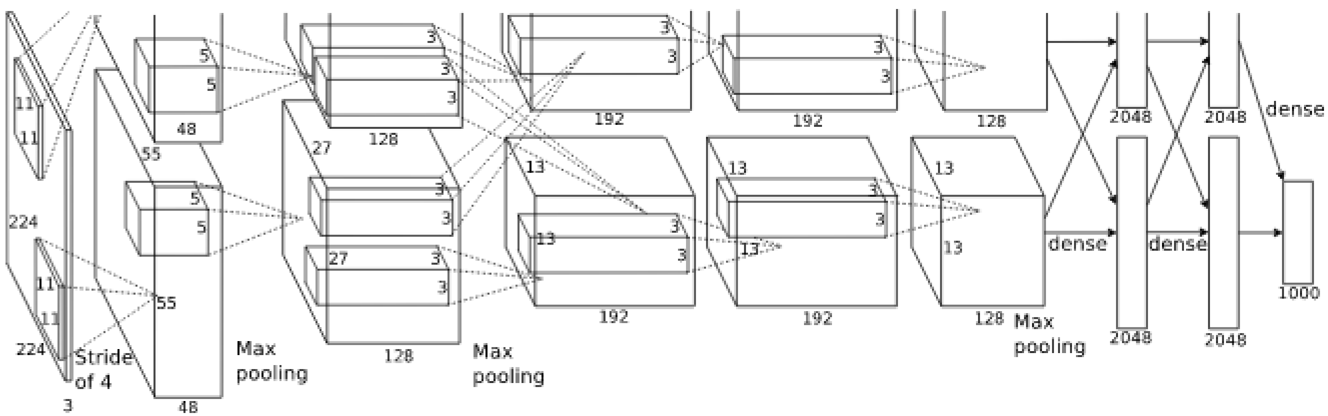


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Lazaridou et al. 2015b

Do Distributed Semantic Models Dream of Electric Sheep?

- “dreams” as averages of 20 related images
- Not including images of the word itself
- Task: relate words with “dreams”
- 20 votes per item

Lazaridou et al. 2015b

- Experiment 1: pick the right word for the dream

Figure 1: **Experiment 1:** Example dreams with correct dreamed word and confounder. Subjects showed a significant preference for the colored word (green if right, red if wrong).

Lazaridou et al. 2015b

- The reverse: pick the right dream

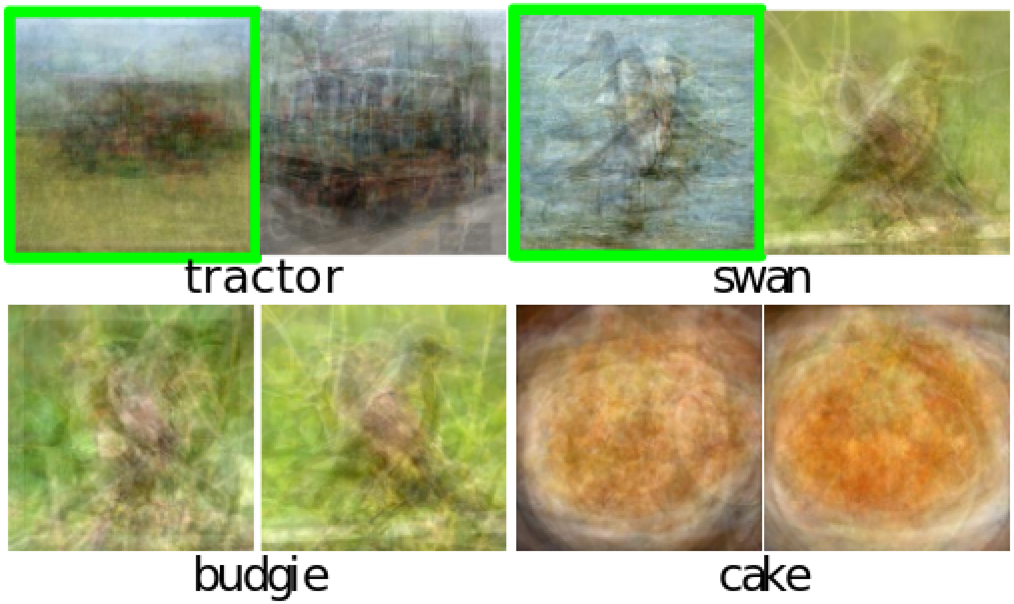


Figure 2: **Experiment 2:** Example dream pairs: the one on the left was generated from the word below the pair, the other from a confounder (clockwise from top left: *truck*, *dove*, *pie*, *parakeet*).

Results

- Experiment 1. 90% median percentage of votes for the correct image
- Experiment 2. 60% median percentage of votes for the correct image

Lazaridou et al. 2015b

- Part of why it works: visual properties

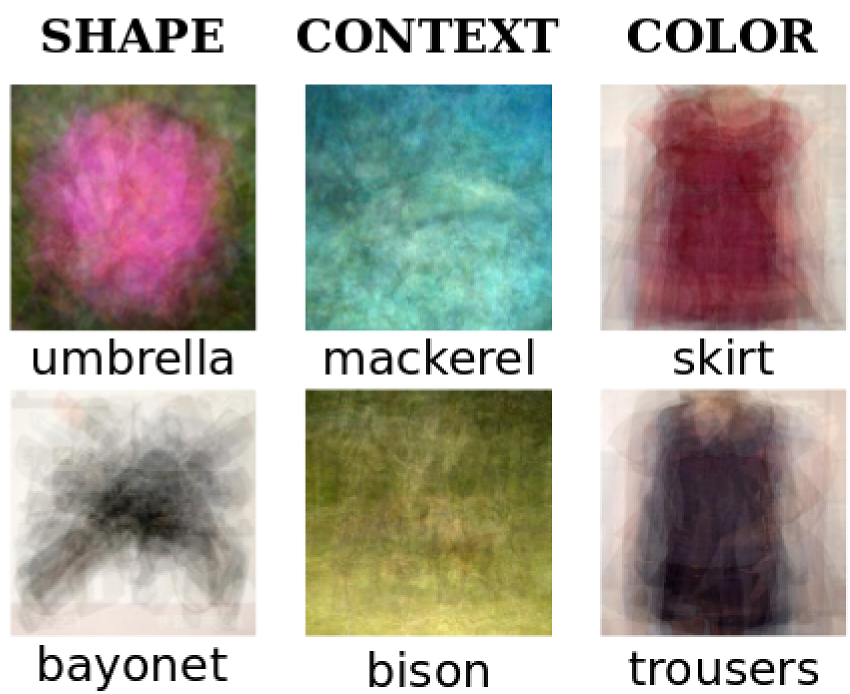
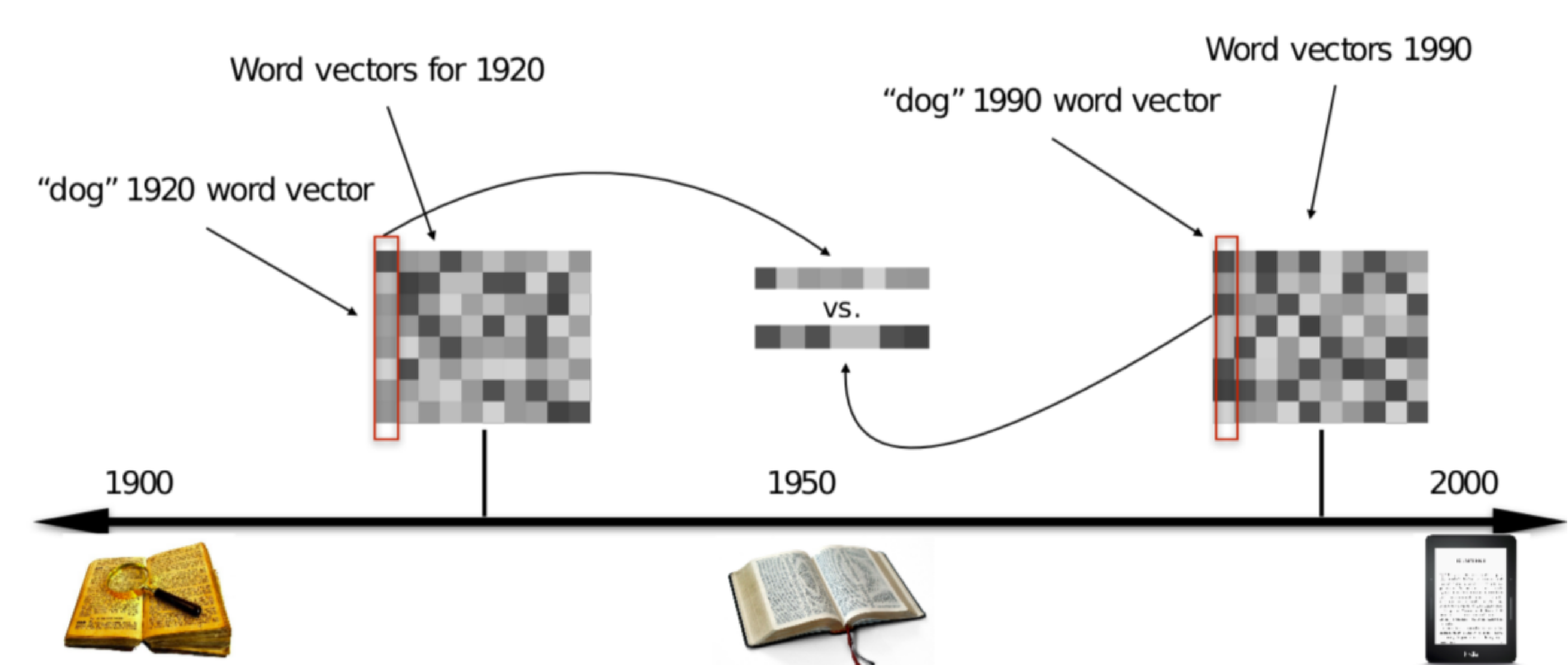


Figure 3: Examples illustrating properties of dream synthesis by image averaging.

Application to diachronic change

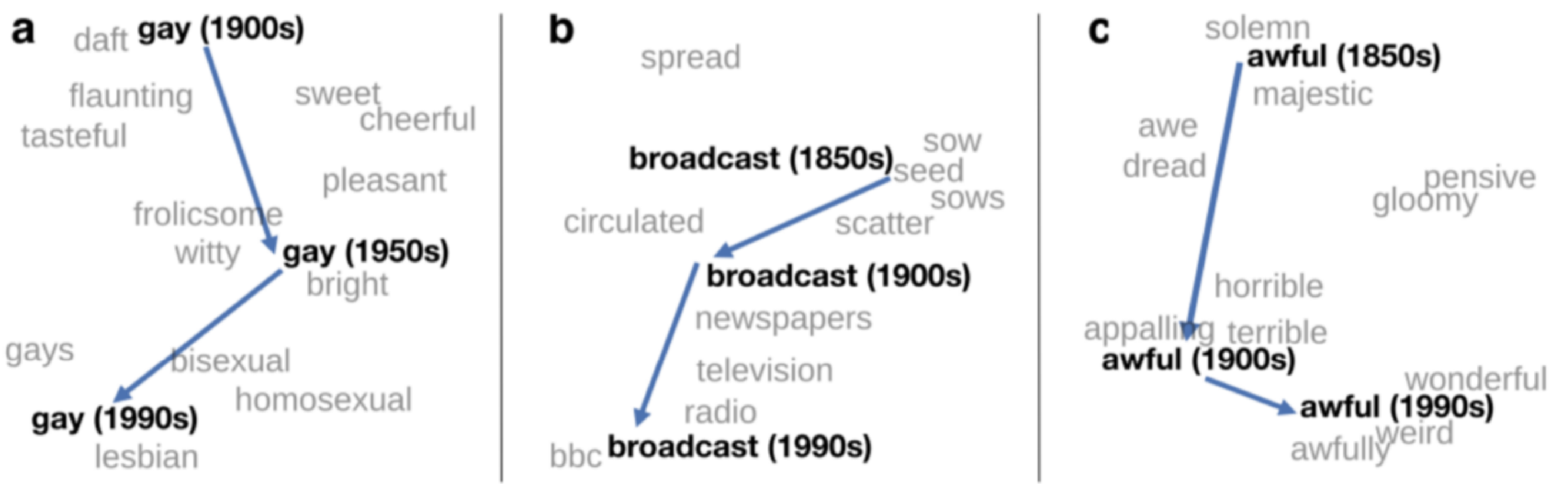
- Hamilton et al. 2016 built distributional vectors for each decade over 150 years and used a matrix to map them to each other



Models and alignment

- Three models:
  - PPMI
  - SVD
  - SGNS
- For the latter two, they had to find a matrix to align vectors of one decade to another

Reasonable performance for known lexical semantic changes



Evaluation

- 28 expert-attested pairwise shifts, e.g.

  - ***gay, homosexual***
  - ***fatal, lethal***

  moving closer together

  - ***broadcast, seed***
  - ***nice, refined***

  drifting further apart

Examples of discovered shifts

- Top-10 shifts from 1900s to 1990s: how many are sensible?
- Performance: SGNS(8) > SVD(4) > PPMI(1)

Method	Top-10 words that changed from 1900s to 1990s
PPMI	know, got, would, decided, think, stop, remember, **started**, must, wanted
SVD	harry, **headed**, **calls**, **gay**, wherever, male, **actually**, special, cover, naturally
SGNS	**wanting**, **gay**, **check**, **starting**, **major**, **actually**, touching, harry, **headed**, romance

Word	Language	Nearest-neighbors in 1900s	Nearest-neighbors in 1990s
wanting	English	lacking, deficient, lacked, lack, needed	wanted, something, wishing, anything, anybody
asile	French	refuge, asiles, hospice, vieillards, infirmerie	demandeurs, refuge, hospice, visas, admission
widerstand	German	scheiterte, volt, stromstärke, leisten, brechen	opposition, verfolgung, nationalsozialistische, nationalsozialismus, kollaboration

Predicting rate of change

$$\Delta(w_i) \propto f(w_i)^{\beta_f^{<0}} \times d(w_i)^{\beta_d^{>0}}$$

Rate of semantic change

Frequency

Polysemy score

- Frequent words change more slowly
- Polysemous words change faster
- Criticism from Dubossarsky et al. 2017: Most of the observed effects come from statistical noise; the factors are real but their effect is much smaller.

Thank you!

...and see you tomorrow for more applications and a demo!