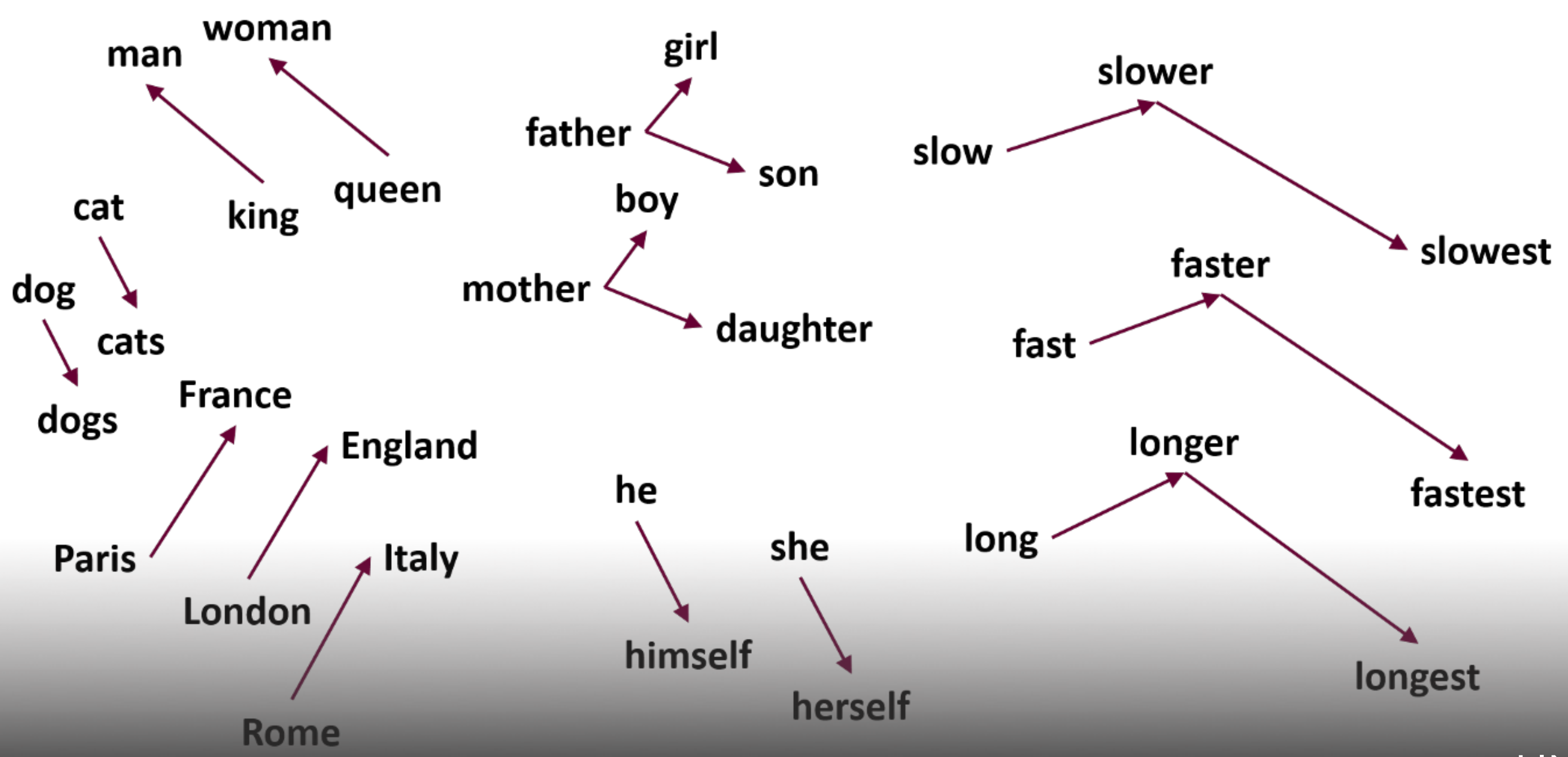
Thursday · September 9, 2021

The word2vec Models of Distributional Semantics



Word Embeddings

Semantics via Feature Vectors

Word Embeddings

What features should we use to describe word meanings?

- Pretend word meanings exist in some vector space, and "embed" them isometrically into $\mathbb{R}^n$.
- Cosine similarity should reflect "word similarity."
- The features will be latent.

The Distributional Hypothesis

John Rupert Firth

- British linguist
- Professor at University of the Punjab, UCL, and SOAS
- Studied the influence of context on language

The Distributional Hypothesis

“As Wittgenstein says, ‘**the meaning of words lies in their use**.’ The day-to-day practice of playing language games recognizes customs and rules. It follows that a text in such established usage may contain sentences such as ‘Don’t be such an ass!’, ‘You silly ass!’, ‘What an ass he is!’

The Distributional Hypothesis

"In these examples, the word *ass* is **in familiar and habitual company**, commonly collocated with *you silly* —, *he is a silly* —, *don't be such an* —. **You shall know a word by the company it keeps!** One of the meanings of *ass* is its **habitual collocation** with such other words as those above quoted."

J. R. Firth

A Synopsis of Linguistic Theory, 1930–1955 (1957)

According to Firth, is a burrito a sandwich?

Why or why not?

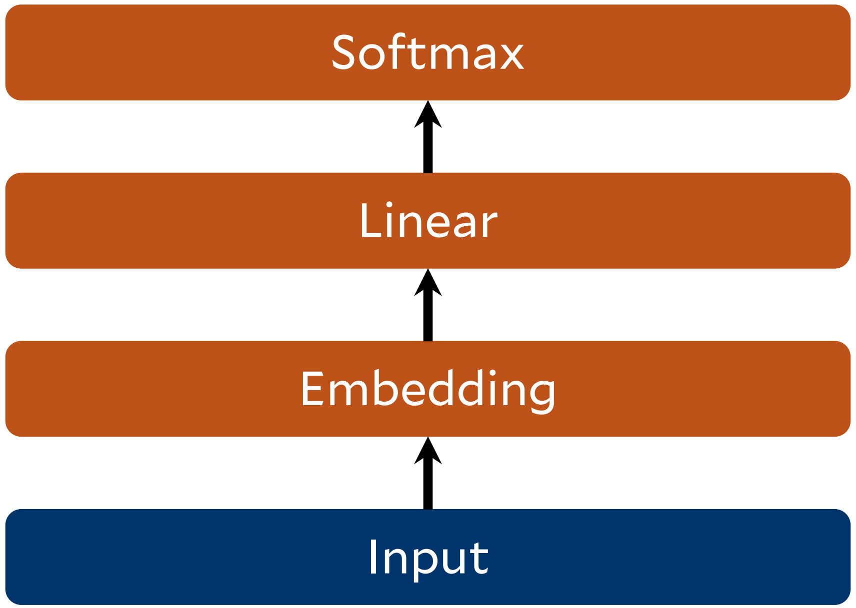
The word2vec Models

Word2vec is a family of distributional algorithms for creating word embeddings.

- Continuous Bag of Words
- Continuous Skip-Gram
- Hierarchical Softmax
- Negative Sampling

SG Neural Network Architecture

$$\boldsymbol{e} = \boldsymbol{W}^{(e)}_{x,:}$$
$$\boldsymbol{y} = \mathrm{softmax}\left(\boldsymbol{W}^{(o)}\boldsymbol{e}\right)$$



- Perceptron with embedding and linear layers
- **Input:** $x \in \mathbb{N}$
- **Output:** $\boldsymbol{y} \in \mathbb{R}^{|\mathbb{V}|}$

SG as a Prediction Task

- The SG model is usually thought of as a model that predicts words around an input word.
- **Input:** health
- **Prediction:** Brazil ’s _ minister has

Continuous Bag of Words

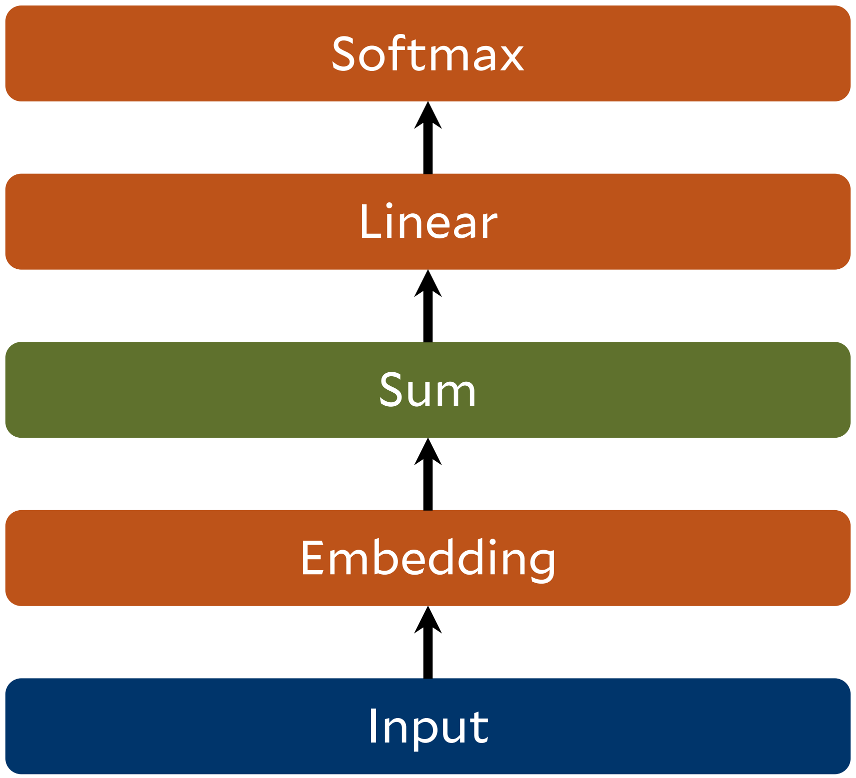
- The continuous bag of words (CBOW) model takes a skip-gram as input and predicts the middle word.
- **Input:** Brazil ’s _ minister has
- **Prediction:** health

CBOW Neural Network Architecture

$$e = \mathbf{1}^\top W^{(e)}_{x,:}$$

$$y = \mathrm{softmax}\left(W^{(o)} e\right)$$

- **Input:** $x \in \mathbb{N}^{2k}$
- **Output:** $y \in \mathbb{R}^{|\mathbb{V}|}$
- Embeddings are added together (bag of words)



Transfer Learning

- Technically, the CBOW model is a word predictor.
- (The SG model is a "context predictor.")
- Word2Vec is an example of **transfer learning**: the neural network learns something (an embedding) by being trained to do something else (word prediction).

Skip-Gram with Negative Sampling (SNGS)

- Goal: To create a word embedding $[\![w]\!] \in \mathbb{R}^d$ for each word w in a vocabulary $\mathbb{V}$.
- Words that "occur together" should have a high cosine similarity.

Logistic Regression

Binary Classification using Logistic Regression

$$y = \sigma(\boldsymbol{a}^\top \boldsymbol{x} + b)$$

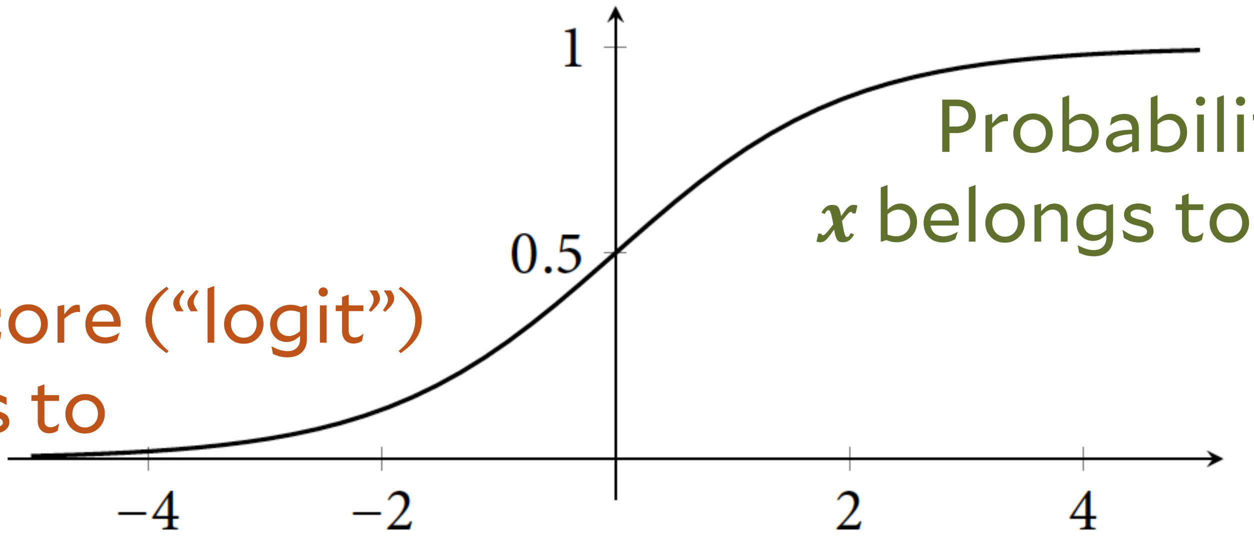
where σ is the *sigmoid function*:

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$

Logistic Regression

$$y = \sigma(\boldsymbol{a}^\top \boldsymbol{x} + b)$$

$\boldsymbol{a}^\top \boldsymbol{x} + b$
Confidence score ("logit") that $\boldsymbol{x}$ belongs to class 1



y
Probability that $\boldsymbol{x}$ belongs to class 1

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$

Logistic Regression

$$y = \sigma(\langle c \rangle^\top [\![w]\!])$$

- $\boldsymbol{x} = [\![w]\!]$, the word embedding for $w \in \mathbb{V}$
- $\boldsymbol{a} = \langle c \rangle$, the context embedding for $c \in \mathbb{V}$
- $b = 0$
- $y =$ the probability that w and c "occur together"


The New York Times
Biden's Agenda ›
Daily Political Briefing
Infrastructure Bill Passes
Increase in Child Tax Credit
$4 Trillion Economic Plan
Democrats and Lobbyists Gird for Battle Over Far-Reaching Tax Increases
⟨c⟩
⟦w⟧
Congressional committees this week begin drafting tax increases on the wealthy and corporations to pay for a $3.5 trillion social policy bill, but the targets are putting up a fight.
Skip-Gram
Window Size: 4

Dataset Construction

- Given: A corpus of text
- Create: A dataset $\mathbb{D} \subseteq \mathbb{V} \times \mathbb{V} \times \{0,1\}$
- $(w, c, 1) \in \mathbb{D}$ means w and c occur together
- $(w, c, 0) \in \mathbb{D}$ means w and c do not occur together
- Example:
  - $(\text{corporations}, \text{wealthy}, 1) \in \mathbb{D}$
  - $(\text{corporations}, \text{spherification}, 0) \in \mathbb{D}$

Dataset Construction

- Initialize $\mathbb{D}$ to be an empty dataset.
- For each word w in the corpus:
  - Form a skip-gram $c_1, c_2, \dots, c_i, w, c_{i+1}, c_{i+2}, \dots, c_n$ around w with window size at most m.
  - For $1 \leq j \leq n$, add $(w, c_j, 1)$ to $\mathbb{D}$.
  - Randomly sample words $c'_1, c'_2, \dots, c'_{kn}$ from the vocabulary $\mathbb{V}$.
  - For $1 \leq j \leq kn$, add $(w, c'_j, 0)$ to $\mathbb{D}$.

Maximum Likelihood Estimation

$$\max_{\langle\cdot\rangle,[\![\cdot]\!]} \left(\prod_{(w,c,1)\in\mathbb{D}} \sigma(\langle c\rangle^\top [\![w]\!]) \right) \left(\prod_{(w,c,0)\in\mathbb{D}} 1-\sigma(\langle c\rangle^\top [\![w]\!]) \right)$$

- Notice that $1-\sigma(\langle c\rangle^\top [\![w]\!]) = \sigma(-\langle c\rangle^\top [\![w]\!])$

Maximum Likelihood Estimation

$$\max_{\langle\cdot\rangle,[\![\cdot]\!]} \left(\prod_{(w,c,1)\in\mathbb{D}} \sigma(\langle c\rangle^{\top}[\![w]\!]) \right) \left(\prod_{(w,c,0)\in\mathbb{D}} \sigma(-\langle c\rangle^{\top}[\![w]\!]) \right)$$

- Notice that $1 - \sigma(\langle c\rangle^{\top}[\![w]\!]) = \sigma(-\langle c\rangle^{\top}[\![w]\!])$
- Take log for numerical stability

Maximum Likelihood Estimation

$$\max_{\langle\cdot\rangle,[\![\cdot]\!]} \left(\sum_{(w,c,1)\in\mathbb{D}} \ln(\sigma(\langle c\rangle^\top [\![w]\!])) \right) + \left(\sum_{(w,c,0)\in\mathbb{D}} \ln(\sigma(-\langle c\rangle^\top [\![w]\!])) \right)$$

- Notice that $1 - \sigma(\langle c\rangle^\top [\![w]\!]) = \sigma(-\langle c\rangle^\top [\![w]\!])$
- Take log for numerical stability
- Change to a minimization problem

Maximum Likelihood Estimation

$$\min_{\langle\cdot\rangle,[\![\cdot]\!]} -\left(\sum_{(w,c,1)\in\mathbb{D}} \ln(\sigma(\langle c\rangle^{\top}[\![w]\!]))\right) - \left(\sum_{(w,c,0)\in\mathbb{D}} \ln(\sigma(-\langle c\rangle^{\top}[\![w]\!]))\right)$$

- Notice that $1-\sigma(\langle c\rangle^{\top}[\![w]\!]) = \sigma(-\langle c\rangle^{\top}[\![w]\!])$
- Take log for numerical stability
- Change to a minimization problem
- Scale negative samples by k

Maximum Likelihood Estimation

$$\min_{\langle\cdot\rangle,[\![\cdot]\!]} -\left(\sum_{(w,c,1)\in\mathbb{D}} \ln(\sigma(\langle c\rangle^\top [\![w]\!]))\right) - \frac{1}{k}\left(\sum_{(w,c,0)\in\mathbb{D}} \ln(\sigma(-\langle c\rangle^\top [\![w]\!]))\right)$$

- Notice that $1 - \sigma(\langle c\rangle^\top [\![w]\!]) = \sigma(-\langle c\rangle^\top [\![w]\!])$
- Take log for numerical stability
- Change to a minimization problem
- Scale negative samples by k

Full Algorithm

- Build a dataset $\mathbb{D} \subseteq \mathbb{V} \times \mathbb{V} \times \{0,1\}$.
  - Examples of class 1 are taken from skip-grams in a corpus
  - Examples of class 0 are taken from negative sampling
- Solve the following minimization problem:

$$\min_{\langle\cdot\rangle, [\![\cdot]\!]} -\left(\sum_{(w,c,1)\in\mathbb{D}} \ln(\sigma(\langle c\rangle^\top [\![w]\!]))\right) - \frac{1}{k}\left(\sum_{(w,c,0)\in\mathbb{D}} \ln(\sigma(-\langle c\rangle^\top [\![w]\!]))\right)$$

- Discard the context embeddings.

Skip-Gram with Negative Sampling (SNGS)

- Goal: To create a word embedding $[\![w]\!] \in \mathbb{R}^d$ for each word w in a vocabulary $\mathbb{V}$.
- Words that "occur together" should have a high cosine similarity.


The New York Times
Biden's Agenda ›
Daily Political Briefing
Infrastructure Bill Passes
Increase in Child Tax Credit
$4 Trillion Economic Plan
Democrats and Lobbyists Gird for Battle Over Far-Reaching Tax Increases
Congressional committees this week begin drafting tax increases on the wealthy and corporations to pay for a $3.5 trillion social policy bill, but the targets are putting up a fight.
⟨c⟩
⟦w⟧
Skip-Gram
Window Size: 4

Skip-Gram with Negative Sampling (SNGS)

- What’s the effect of window size?
  - Window of 1 word
  - Window of 4 words
  - Window of 100 words

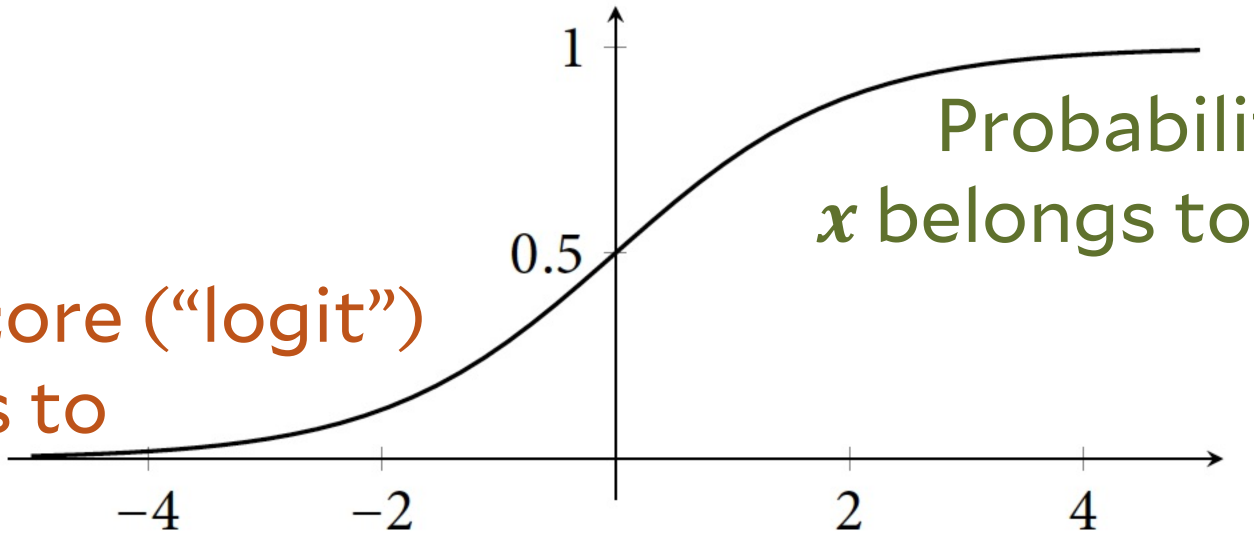
Skip-Gram Model

$$y = \sigma(\langle c \rangle^\top [\![w]\!])$$

- $[\![w]\!]$ is the word embedding for $w \in \mathbb{V}$
- $\langle c \rangle$ is the context embedding for $c \in \mathbb{V}$
- $y =$ the probability that w and c "occur together"

Logistic Regression

$$y = \sigma(\langle c \rangle^\top [\![w]\!])$$



y
Probability that
$\boldsymbol{x}$ belongs to class 1

$\langle \boldsymbol{c} \rangle^\top [\![\boldsymbol{w}]\!]$
Confidence score (“logit”)
that $\boldsymbol{x}$ belongs to
class 1

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$

Dataset Construction

- Given: A corpus of text
- Create: A dataset $\mathbb{D} \subseteq \mathbb{V} \times \mathbb{V} \times \{0,1\}$
- $(w, c, 1) \in \mathbb{D}$ means w and c occur together
- $(w, c, 0) \in \mathbb{D}$ means w and c do not occur together

Finding the embeddings: maximum likelihood estimation

- Given embeddings for words and contexts, we can compute two things:
  - $p(d = 1|w, c) = \sigma(\langle c\rangle^\top [\![w]\!])$
  - $p(d = 0|w, c) = 1 - \sigma(\langle c\rangle^\top [\![w]\!])$

Finding the embeddings: maximum likelihood estimation

$$\underbrace{\left(\prod_{(w,c,0)\in\mathbb{D}} p(d=1|w,c)\right)}_{\text{Total prob of positive w,c pairs}} \underbrace{\left(\prod_{(w,c,0)\in\mathbb{D}} p(d=0|w,c)\right)}_{\text{Total prob of negative w,c pairs}}$$

independence assumptions

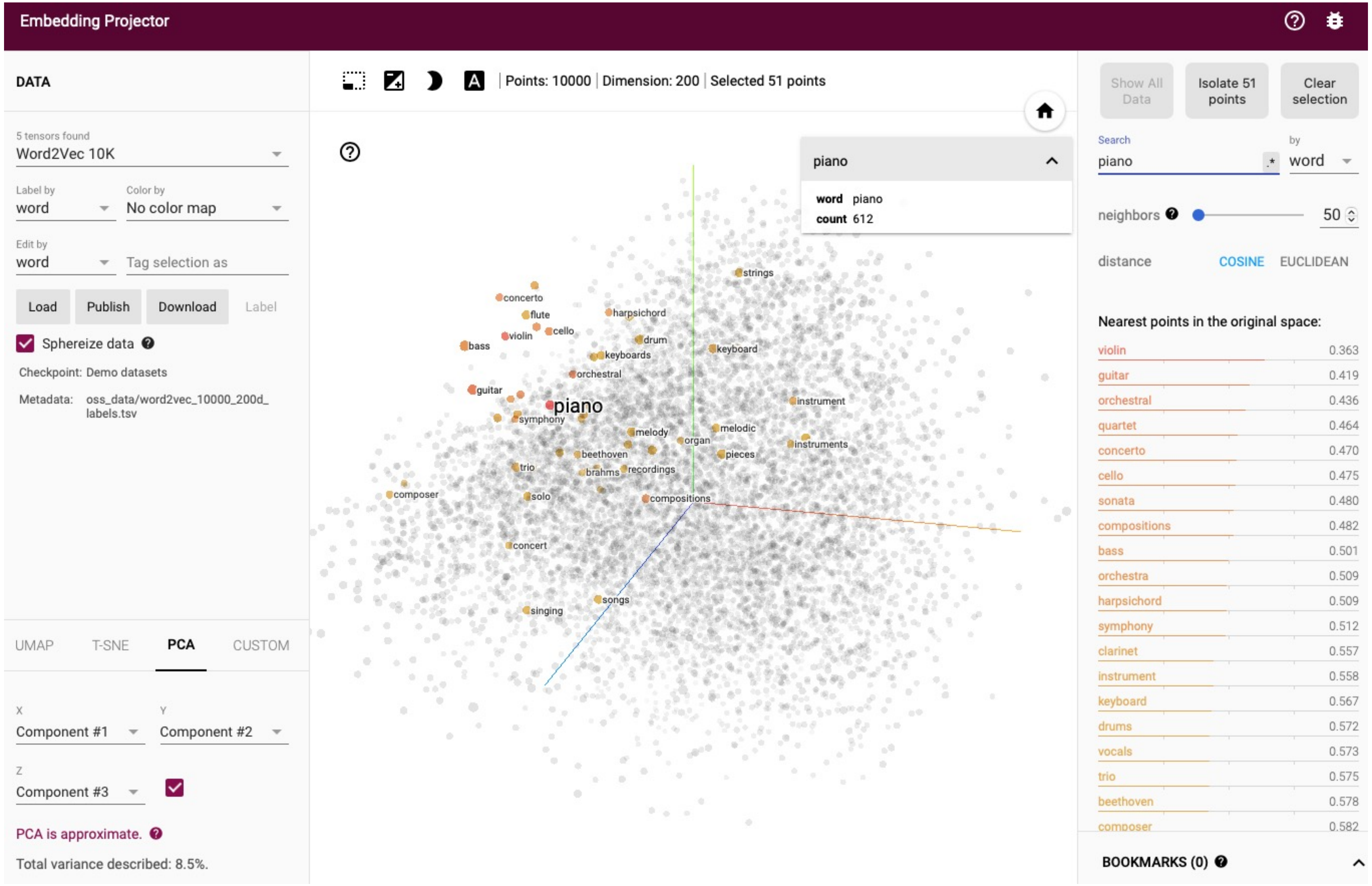
Pick the embeddings that make the observed data as likely as possible!

$$\max_{\langle\cdot\rangle,[\![\cdot]\!]} \left(\prod_{(w,c,1)\in\mathbb{D}} p(d=1|w,c)\right)\left(\prod_{(w,c,0)\in\mathbb{D}} p(d=0|w,c)\right)$$

So what do we do now?

- Having found the “best” word and context embeddings:
  - Discard the context embeddings.
  - Use word embeddings as a lexical representation

Word embeddings and word meanings



http://projector.tensorflow.org

Word embeddings and word meanings

- Human word similarity judgments (Rubenstein and Goodenough 1965)
- Word2vec cosine similarity (Baroni, Dinu and Kruszewski 2015 give Pearson correlation of .84 for this dataset)

TABLE 1. Judged Synonymy of Theme Pairs

cord	smile	0.02	hill	woodland	1.48
rooster	voyage	0.04	car	journey	1.55
noon	string	0.04	cemetery	mound	1.69
fruit	furnace	0.05	glass	jewel	1.78
autograph	shore	0.06	magician	oracle	1.82
automobile	wizard	0.11	crane	implement	2.37
mound	stove	0.14	brother	lad	2.41
grin	implement	0.18	sage	wizard	2.46
asylum	fruit	0.19	oracle	sage	2.61
asylum	monk	0.39	bird	crane	2.63
graveyard	madhouse	0.42	bird	cock	2.63
glass	magician	0.44	food	fruit	2.69
boy	rooster	0.44	brother	monk	2.74
cushion	jewel	0.45	asylum	madhouse	3.04
monk	slave	0.57	furnace	stove	3.11
asylum	cemetery	0.79	magician	wizard	3.21
coast	forest	0.85	hill	mound	3.29
grin	lad	0.88	cord	string	3.41
shore	woodland	0.90	glass	tumbler	3.45
monk	oracle	0.91	grin	smile	3.46
boy	sage	0.96	serf	slave	3.46
automobile	cushion	0.97	journey	voyage	3.58
mound	shore	0.97	autograph	signature	3.59
lad	wizard	0.99	coast	shore	3.60
forest	graveyard	1.00	forest	woodland	3.65
food	rooster	1.09	implement	tool	3.66
cemetery	woodland	1.18	cock	rooster	3.68
shore	voyage	1.22	boy	lad	3.82
bird	woodland	1.24	cushion	pillow	3.84
coast	hill	1.26	cemetery	graveyard	3.88
furnace	implement	1.37	automobile	car	3.92
crane	rooster	1.41	midday	noon	3.94
			gem	jewel	3.94

Word embeddings and word meanings

- Analogies (Mikolov et al. 2013):
  - man is to woman as king is to *x*
- In vector form:
  - $[\![man]\!] - [\![woman]\!]$ = $[\![king]\!] - [\![x]\!]$
  - $[\![x]\!] = [\![king]\!] - [\![man]\!] + [\![woman]\!]$
  - Find word x whose embedding is closest to this result:

$$\operatorname*{argmax}_{x \in V} \cos([\![x]\!], [\![king]\!] - [\![man]\!] + [\![woman]\!])$$

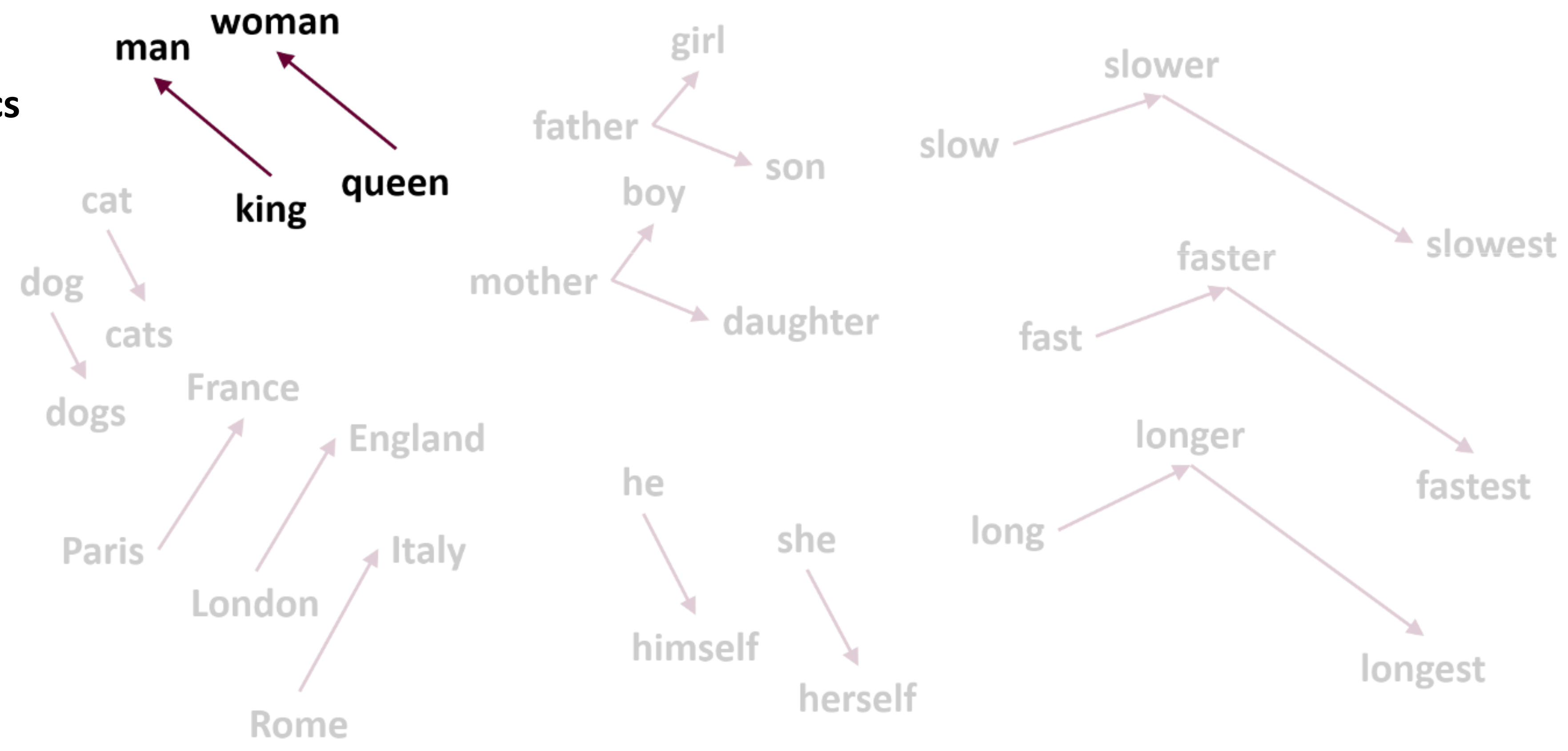
Normalize word vectors to unit length before doing the arithmetic operations!

$$v' = \frac{v}{\|v\|}$$

Structure of the Embedding Space

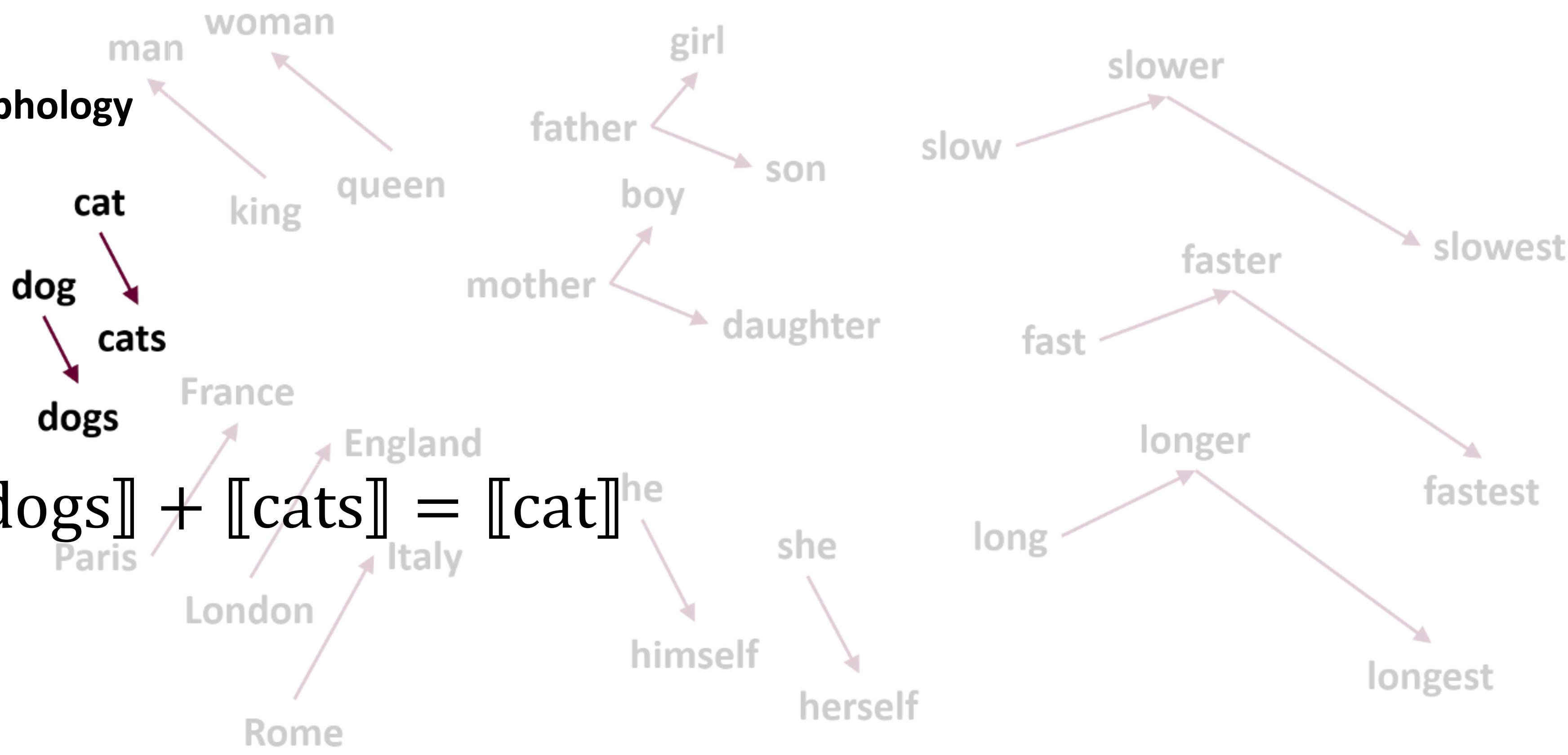
$$[\![\text{king}]\!] - [\![\text{man}]\!] + [\![\text{woman}]\!] = [\![\text{queen}]\!]$$

Lexical semantics



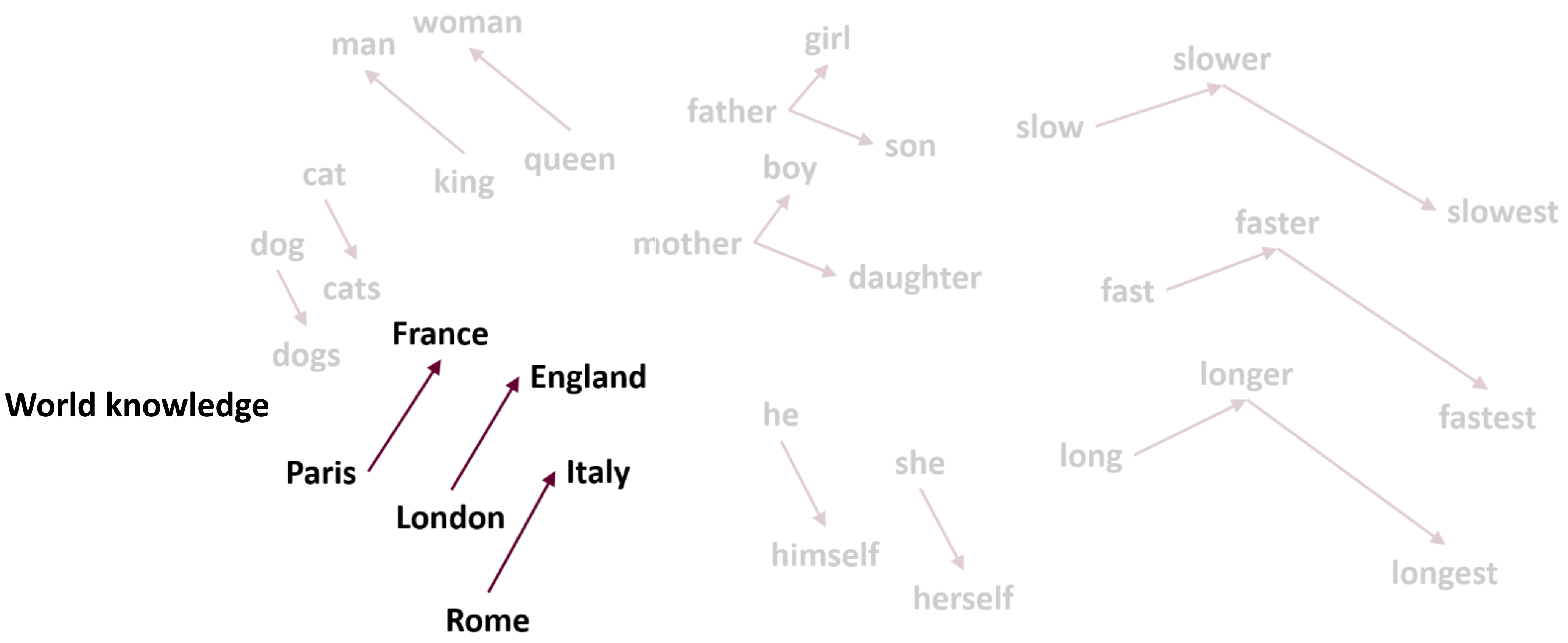
Structure of the Embedding Space

Inflectional morphology



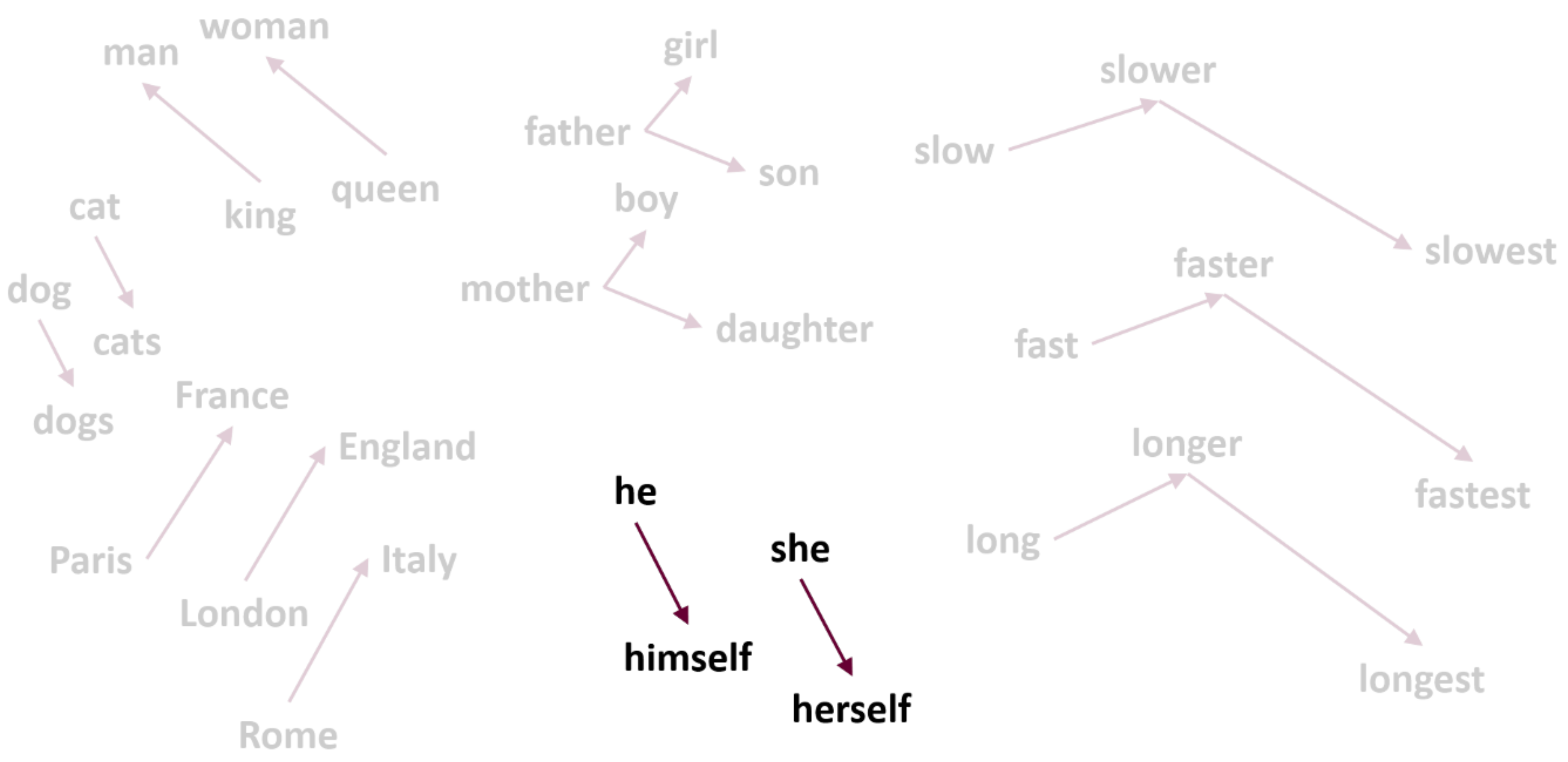
$$[\![\text{dog}]\!] - [\![\text{dogs}]\!] + [\![\text{cats}]\!] = [\![\text{cat}]\!]$$

Structure of the Embedding Space



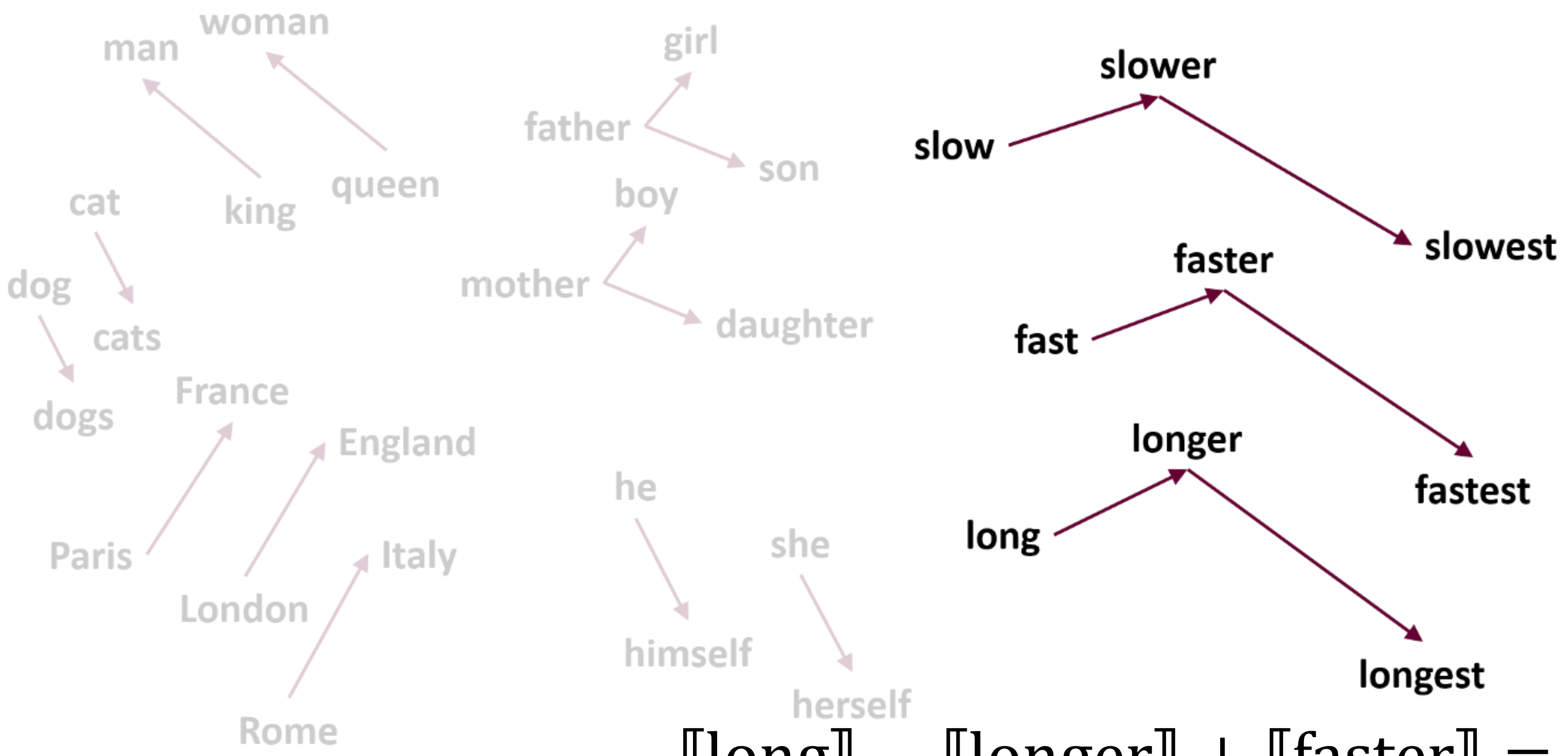
$$[\![\text{London}]\!] - [\![\text{England}]\!] + [\![\text{France}]\!] = [\![\text{Paris}]\!]$$

Structure of the Embedding Space



$$[\![\text{he}]\!] - [\![\text{himself}]\!] + [\![\text{herself}]\!] = [\![\text{she}]\!]$$

Structure of the Embedding Space

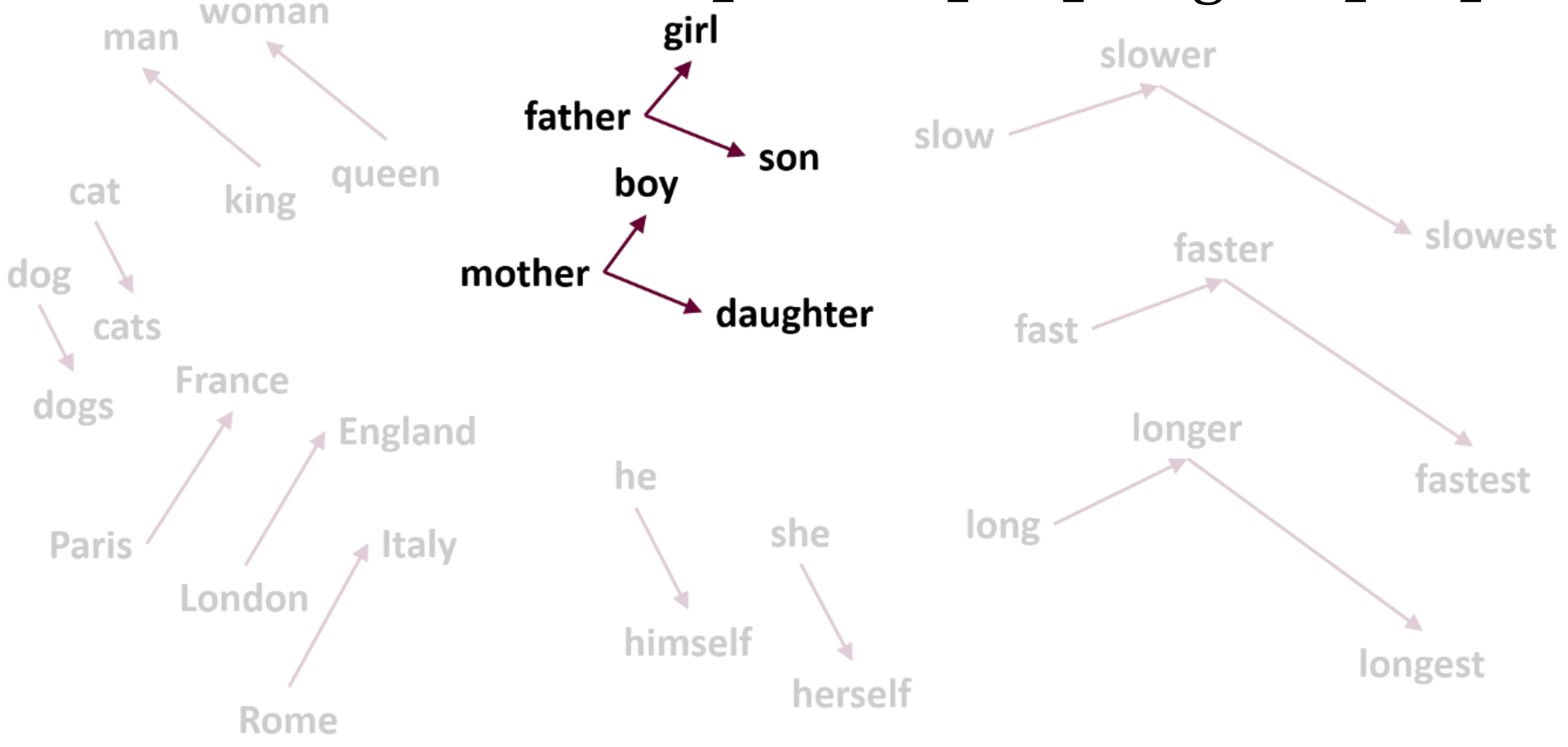


$$[\![\text{long}]\!] - [\![\text{longer}]\!] + [\![\text{faster}]\!] = [\![\text{fast}]\!]$$
$$[\![\text{faster}]\!] - [\![\text{fastest}]\!] + [\![\text{slowest}]\!] = [\![\text{slower}]\!]$$

Structure of the Embedding Space

$$\llbracket \text{mother} \rrbracket - \llbracket \text{boy} \rrbracket + \llbracket \text{girl} \rrbracket = \llbracket \text{father} \rrbracket$$

$$\llbracket \text{mother} \rrbracket - \llbracket \text{daughter} \rrbracket + \llbracket \text{son} \rrbracket = \llbracket \text{father} \rrbracket$$



Success with Analogies

- Nationality (Canada:Canadian :: France:X): 98%
- Comparatives (smart:smarter :: heavy:X): 86%
- Superlatives (smarter:smartest :: heavier:X): 56%
- Adjective to Adverb (quick:quickly :: happy:X): 24%

What doesn't work so well (at least in this simple way)

- Antonymy:

```
presence : absence :: happy : unhappy
absence : presence :: happy : proud
abundant : scarce :: happy : glad
refuse : accept :: happy : satisfied
accurate : inaccurate :: happy : disappointed
admit : deny :: happy : delighted
never : always :: happy : Said_Hirschbeck
modern : ancient :: happy : ecstatic
receded : approached :: happy : excited
departure : arrival :: happy : overjoyed
ascend : descend :: happy : anxious
asleep : awake :: happy : enthused
attractive : repulsive :: happy : disgusting
forward : backward :: happy : sorry
backward : forward :: happy : pleased
ugly : beautiful :: happy : wonderful
beginning : ending :: happy : happier
bent : straight :: happy : consecutive
worst : best :: happy : thrilled
better : worse :: happy : sad
bitter : sweet :: happy : nice
curse : bless :: happy : thankful
bless : curse :: happy : jinx
```

What doesn't work so well (at least in this simple way)

- **Hypernymy:** cat/animal, apple/fruit
- **Meronymy:** government/minister, car/tire
- But there are other methods that perform better…

Why do analogies work? (Goldberg and Levy 2014)

a:a* :: b:b*

- Need to find

$$\underset{b^*\in V}{\operatorname{argmax}} \cos(b^*, b - a + a^*)$$
$$= \underset{b^*\in V}{\operatorname{argmax}} \boxed{\cos(b^*, b)} - \boxed{\cos(b^*, a)} + \boxed{\cos(b^*, a^*)}$$

- Maximize two similarities and one difference
- Trivial solution: b* is identical to b or a*

平凡解

A problematic analogy (Goldberg and Levy 2014)

London:England :: Baghdad:X

	England	London	Baghdad	Σ
Iraq	.153	.130	.631	.654
Mosul	.130	.141	.755	.744

Similarity to Baghdad dominates choice!