but…

- The SRN doesn't scale up
  - 29 words, not 60,768 (as in LSA)
  - RNNs *do* scale up (more about these below)
- Lessons to learn from it
  - prediction
  - hidden layer is a 'semantic space'
    - words are a pattern/vector within that space that reflects the context in which the word can occur
- Is there a computationally easier method?

word2vec – Tomas Mikolov, 2013

- for each word in a corpus, learn the context (surrounding words) in which it can occur

it works by scanning a corpus

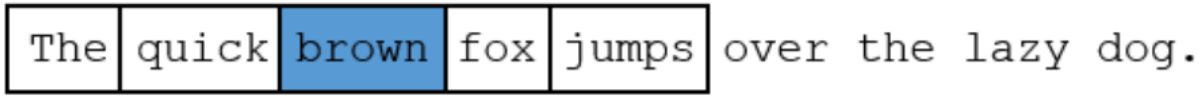
source: Chris McCormick

it works by scanning a corpus

Each time you slide the window over, you input that target word (here, "quick", to Word2Vec and set the outputs to the other words in the immediate context (here, "the", "brown", and "fox"). You then backpropagate to adjust the weights, and then slide the window one over…

source: Chris McCormick

it works by scanning a corpus



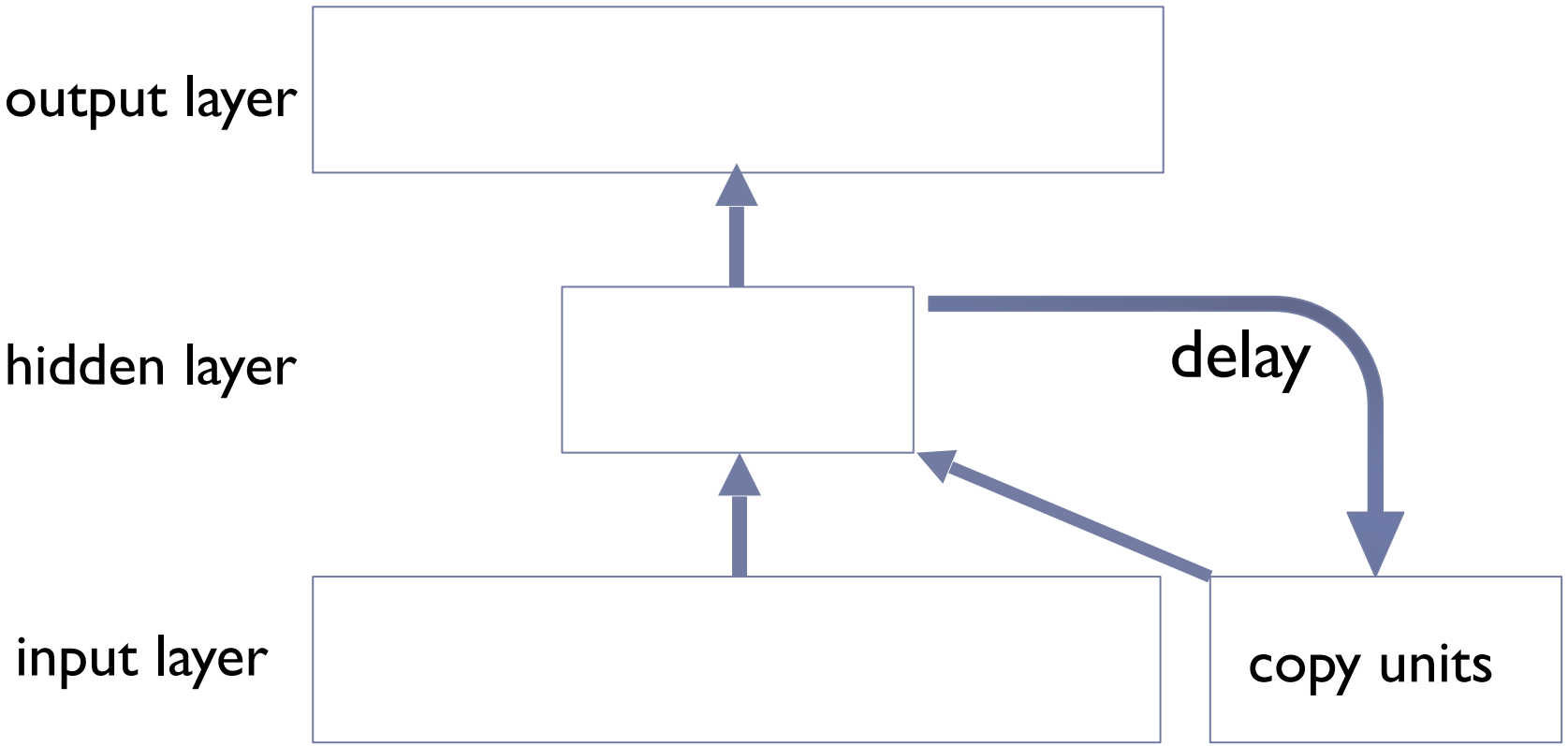
source: Chris McCormick

it works by scanning a corpus

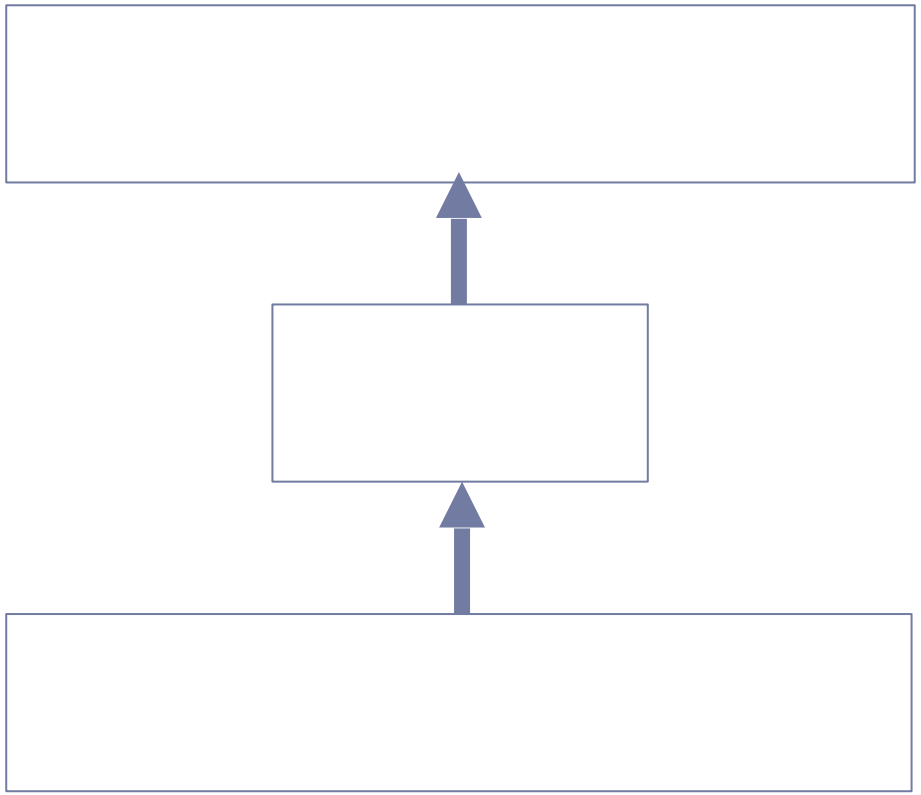
Typically, the context is 5 words before and 5 after

source: Chris McCormick

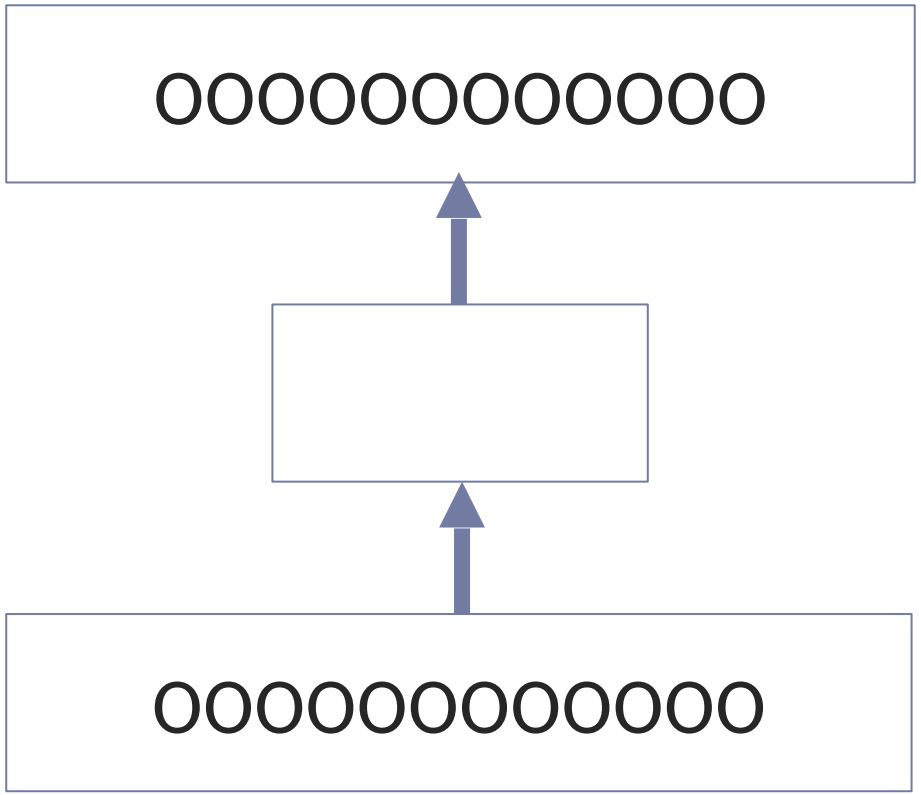
The Simple Recurrent Network (SRN)



A caricature of word2vec



“one-hot vectors” = 1 ‘on’, others ‘off’

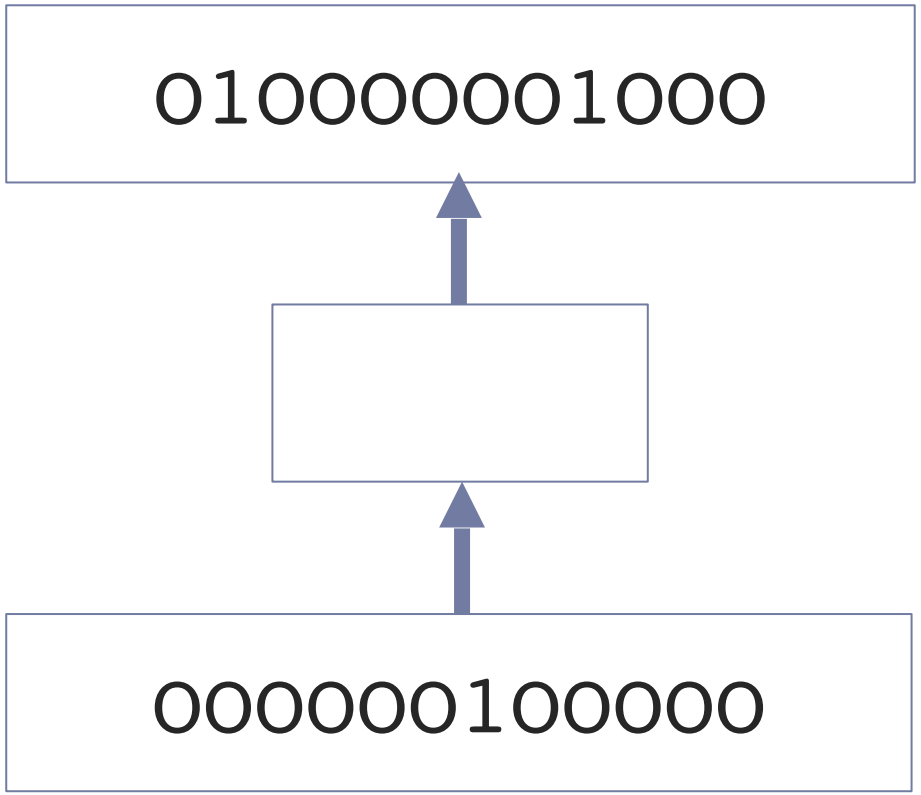


as many units as there are words (50,000?)

300 units

as many units as there are words (50,000?)

input a word and give output its neighbors

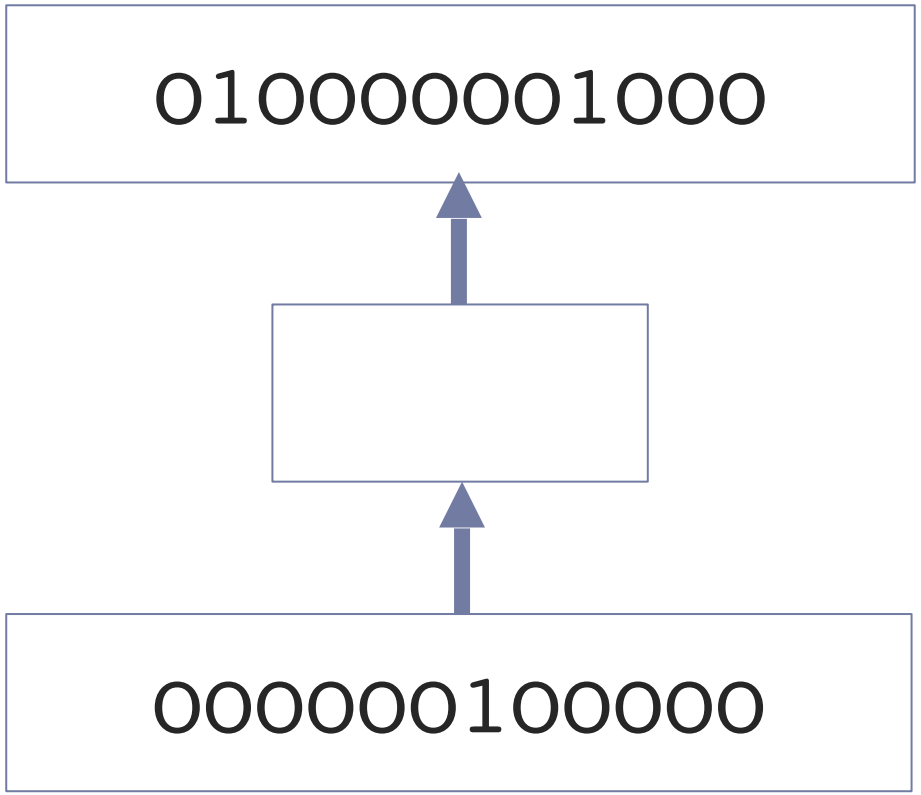


neighboring words in the corpus

300 units

a word in the corpus

there's a lot of back-propagation to do



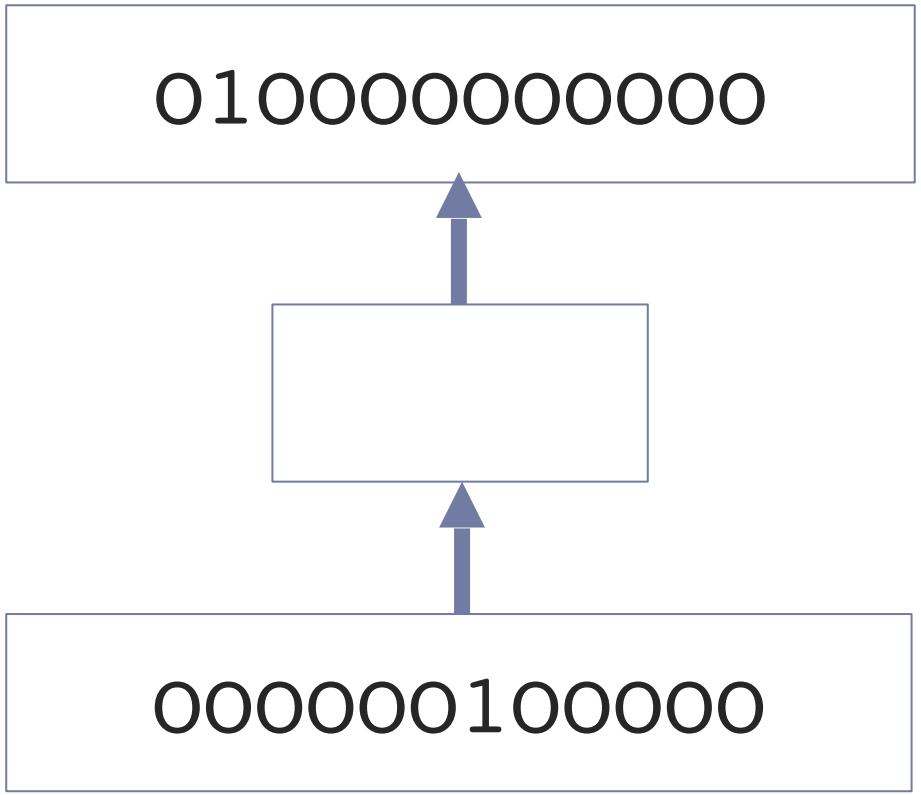
you want a 1 for the correct target words, and a 0 for the other 49,998 words

That's a lot of back-propagation!

300*50,000 = 15M connections!

but there's a simple trick to avoid it

NEGATIVE SAMPLING



you want a 1 for the correct target word, and a 0 for e.g. 5 other words

That's a lot LESS back-propagation!

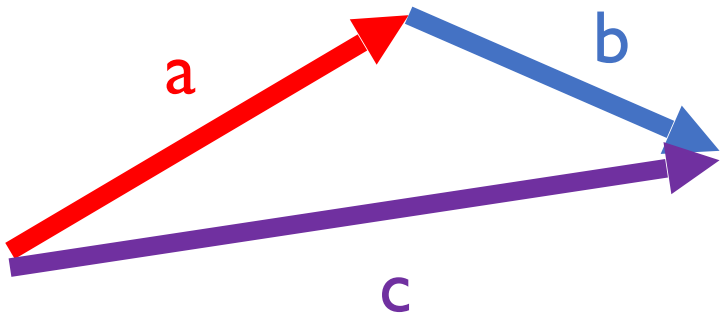
300*6 = 1,800 connections!

word2vec – Tomas Mikolov, 2013

- for each word in a corpus, learn the context (surrounding words) in which it can occur
- The hidden layer encodes the 'semantic space'
  - in fact, it encodes the same information as is encoded in the hidden layer of an SRN
  - or the matrix from Latent Semantic Analysis
- So who cares?
  - people who have HUGE corpora

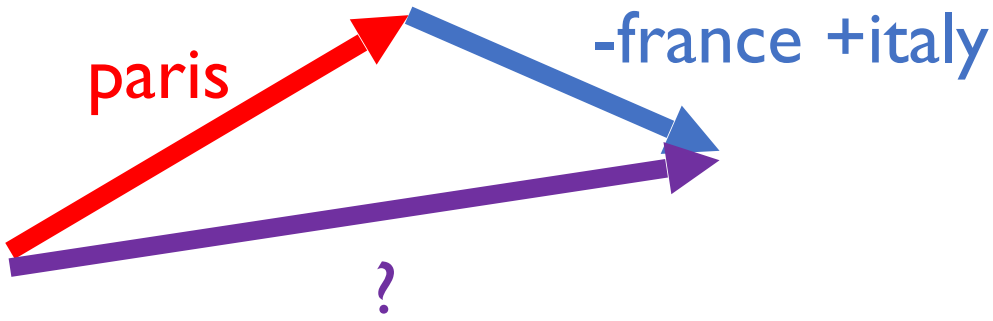
What good are word vectors?

One example: vector manipulation (adding, subtracting, etc.)



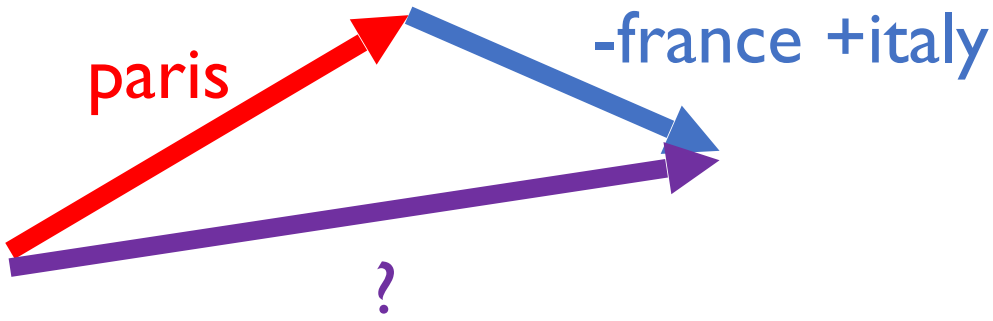
$$a + b = c$$
$$c - b = a$$

What good are word vectors?



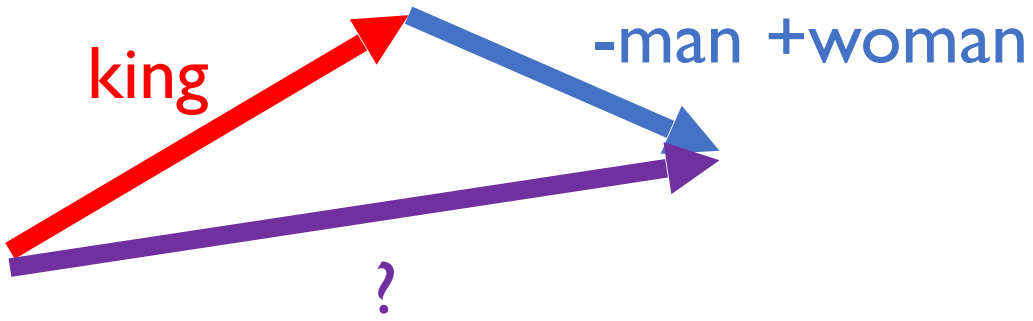
paris - france + italy
= ?

What good are word vectors?



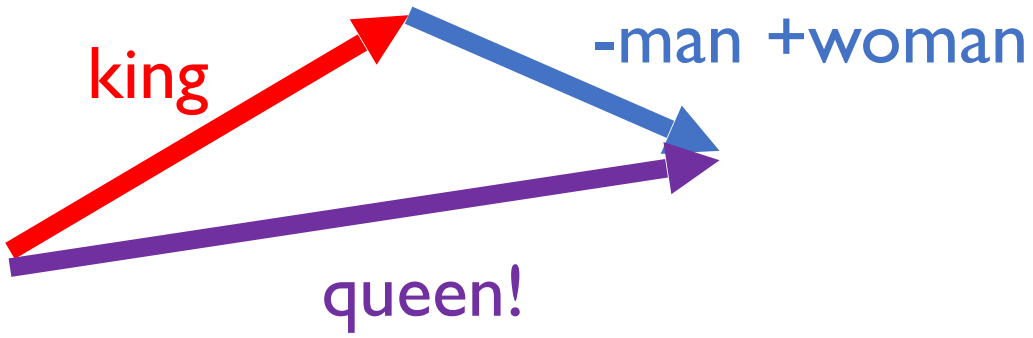
paris - france + italy ≈ rome

What good are word vectors?



king - man + woman
= ?

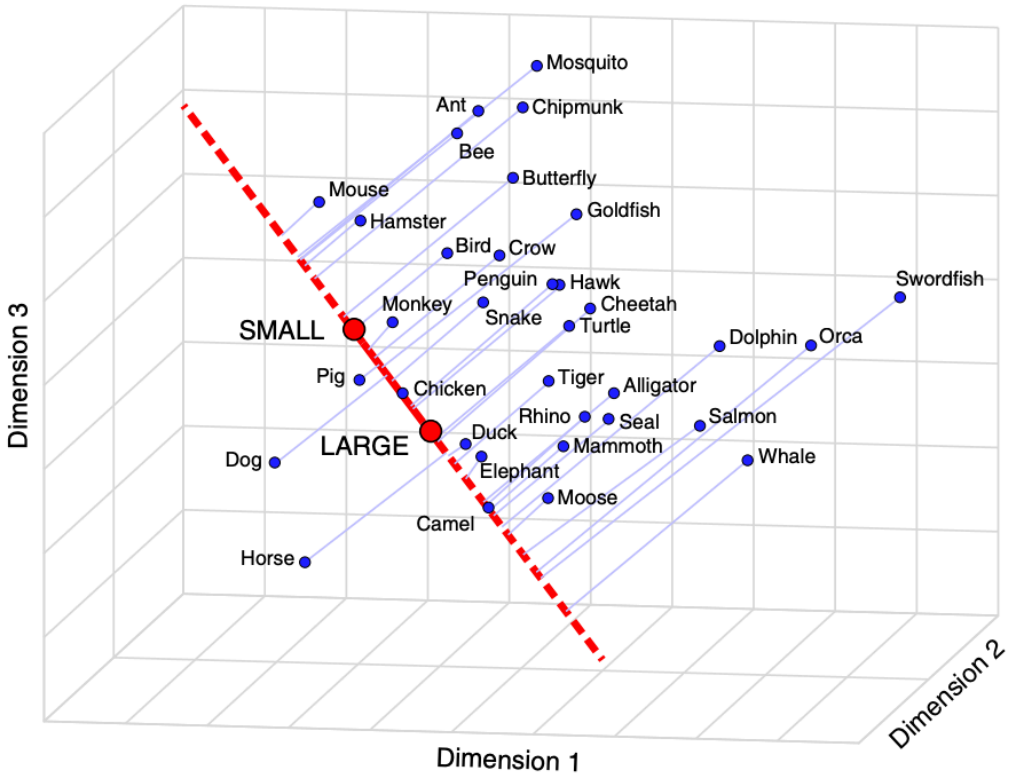
What good are word vectors?



king - man + woman ≈ queen

What good are word vectors?

The models can even contain knowledge of "physical" dimensions



Grand, G., Blank, I. A., Pereira, F., & Fedorenko, E. (2022). Semantic projection recovers rich human knowledge of multiple object features from word embeddings. *Nature human behaviour*, *6*(7), 975-987.

Alternatives to word2vec

- Recurrent Neural Networks (RNNs)
  - With LSTMs (long short term memory)
  - With GRUs (gated recurrent units)
- A problem with SRNs: the vanishing gradient problem
  - The "ripples" of past states quickly get swamped by new inputs – c.f. ripples on a pond dying down
  - LSTMs/GRUs learn to keep activation state until it's needed
- Elman's prediction task