Autoencoders

Denote **z** as encoded with encoder E input **x**

$$\mathbf{z} = E(\mathbf{x}, \boldsymbol{\theta}_E)$$

Decoder D recovers **x** from latent representation

$$\hat{\mathbf{x}} = D(\mathbf{z}, \boldsymbol{\theta}_D)$$

Optimal parameters learned w.r.t. loss function L

$$[\boldsymbol{\theta}_E, \boldsymbol{\theta}_D] = \arg\min L(\hat{\mathbf{x}}, \mathbf{x})$$

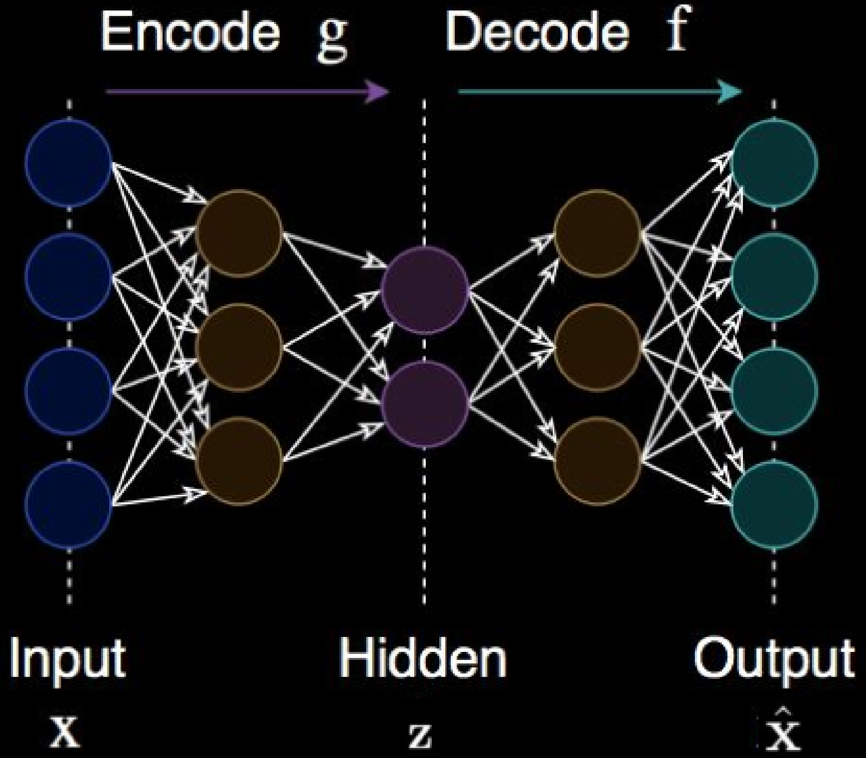


Image source: Habr post on autoencoders and GANs

Autoencoders

Denote **z** as encoded with encoder E input **x**

$$\mathbf{z} = E(\mathbf{x}, \boldsymbol{\theta}_E)$$

Decoder D recovers **x** from latent representation

$$\hat{\mathbf{x}} = D(\mathbf{z}, \boldsymbol{\theta}_D)$$

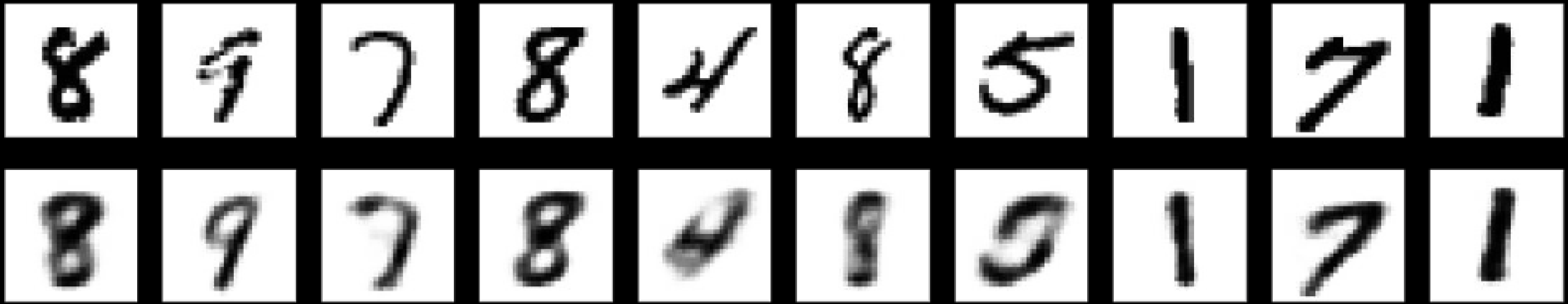
Simple example: PCA

Optimal parameters learned w.r.t. loss function L

$$[\boldsymbol{\theta}_E, \boldsymbol{\theta}_D] = \arg\min L(\hat{\mathbf{x}}, \mathbf{x})$$

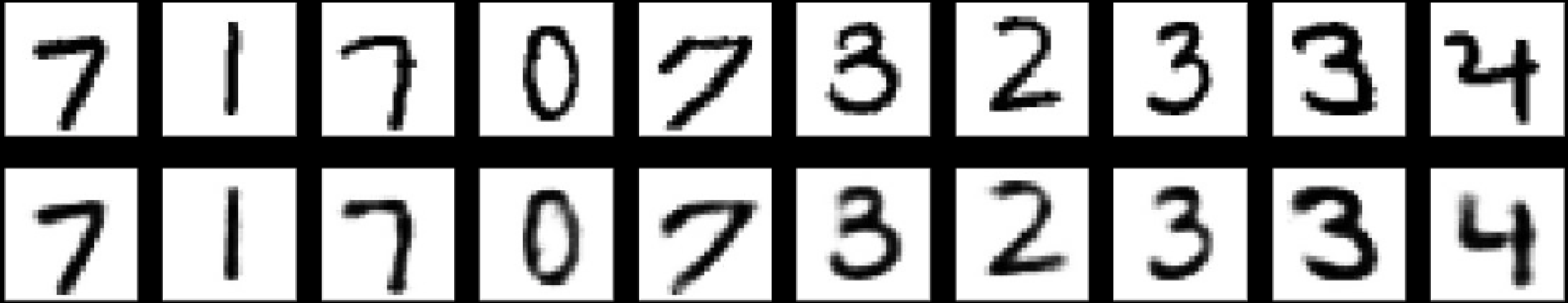
Image source: Habr post on autoencoders and GANs

PCA performance on MNIST



16 components

Convolutional performance on MNIST



7 x 7 latent space

One-hot vectors

- **One-hot vectors:**

```
           Rome    Paris                            word V
Rome   = [1, 0, 0, 0, 0, 0, ..., 0]
Paris  = [0, 1, 0, 0, 0, 0, ..., 0]
Italy  = [0, 0, 1, 0, 0, 0, ..., 0]
France = [0, 0, 0, 1, 0, 0, ..., 0]
```

Problems:

- Huge vectors
- VERY sparse
- No semantics or word similarity information included

Distributional semantics

Does vector similarity imply semantic similarity?

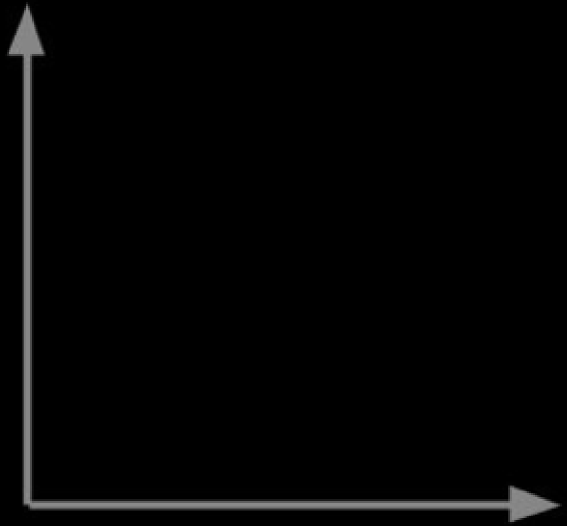
“You shall know a word by the company it keeps”

Firth, 1957

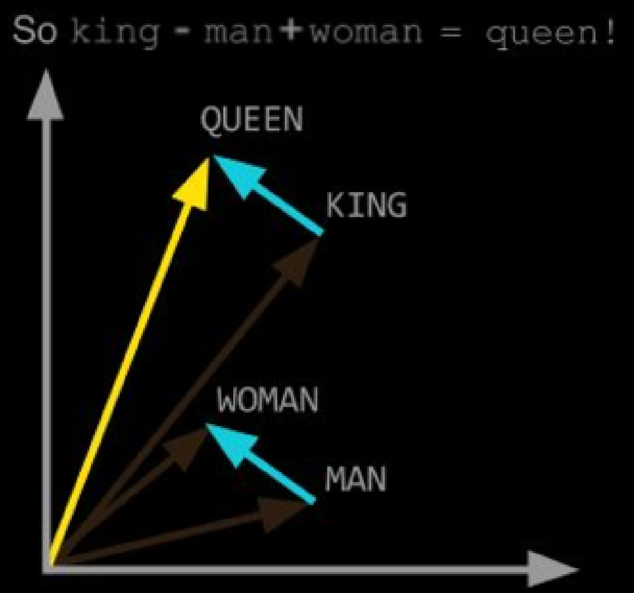
Why not to learn word vectors?

Embeddings: intuition

What is `king` - `man` + `woman`?

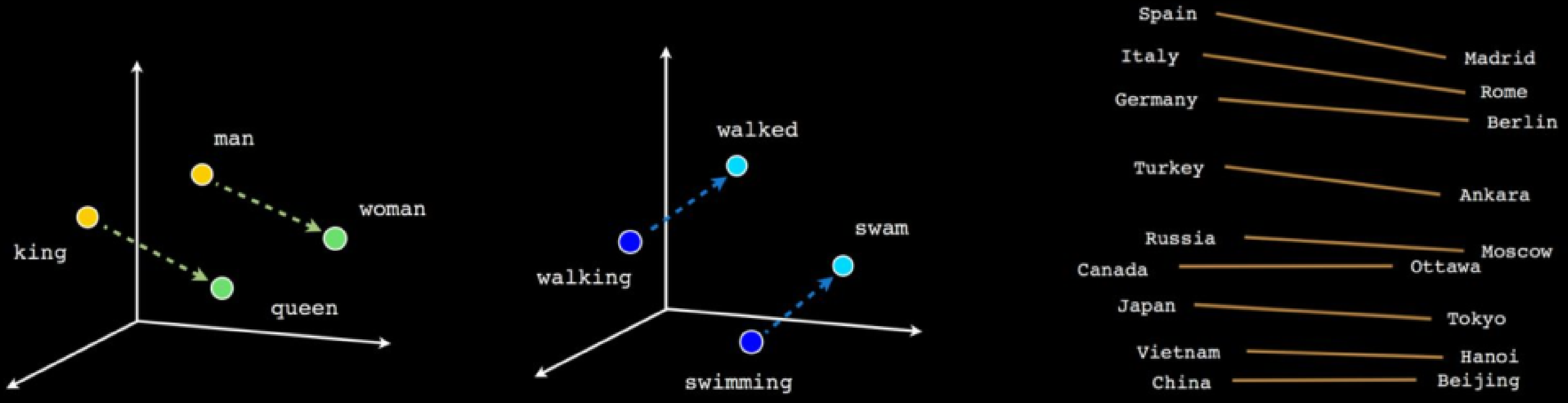


Embeddings: intuition



Word2vec

- **Word2vec** (Mikolov et al. 2013) - a framework for learning word embeddings



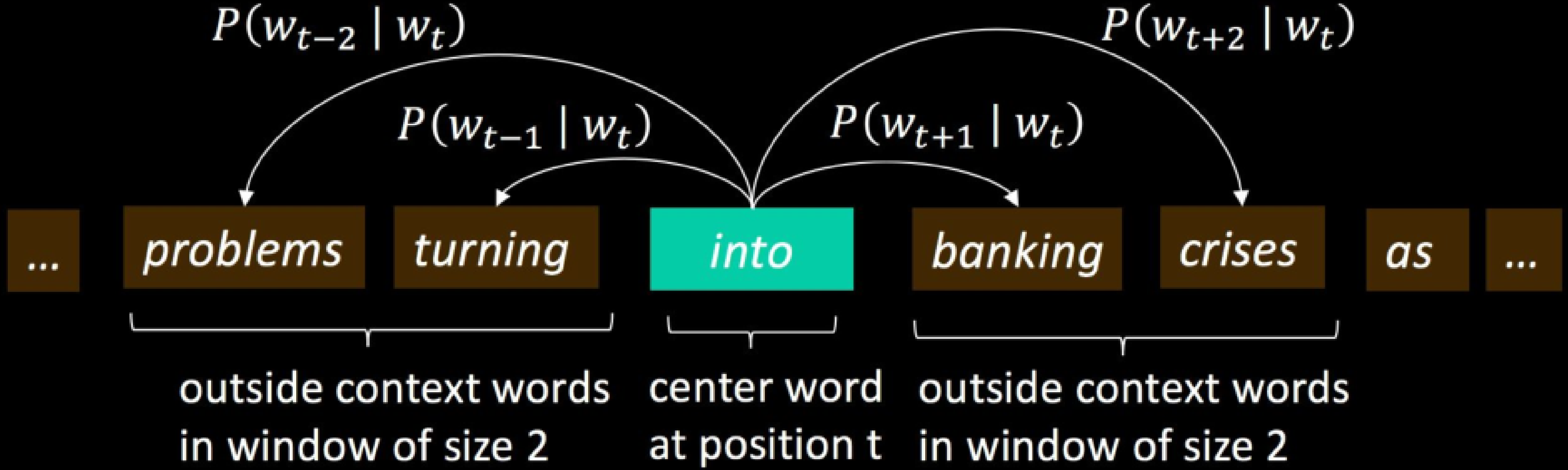
Male-Female | Verb tense | Country-Capital

Embeddings: word2vec

Source Text		Training Samples
The quick brown fox jumps over the lazy dog.	⟹	(the, quick) (the, brown)
The **quick** brown fox jumps over the lazy dog.	⟹	(quick, the) (quick, brown) (quick, fox)
The quick **brown** fox jumps over the lazy dog.	⟹	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown **fox** jumps over the lazy dog.	⟹	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

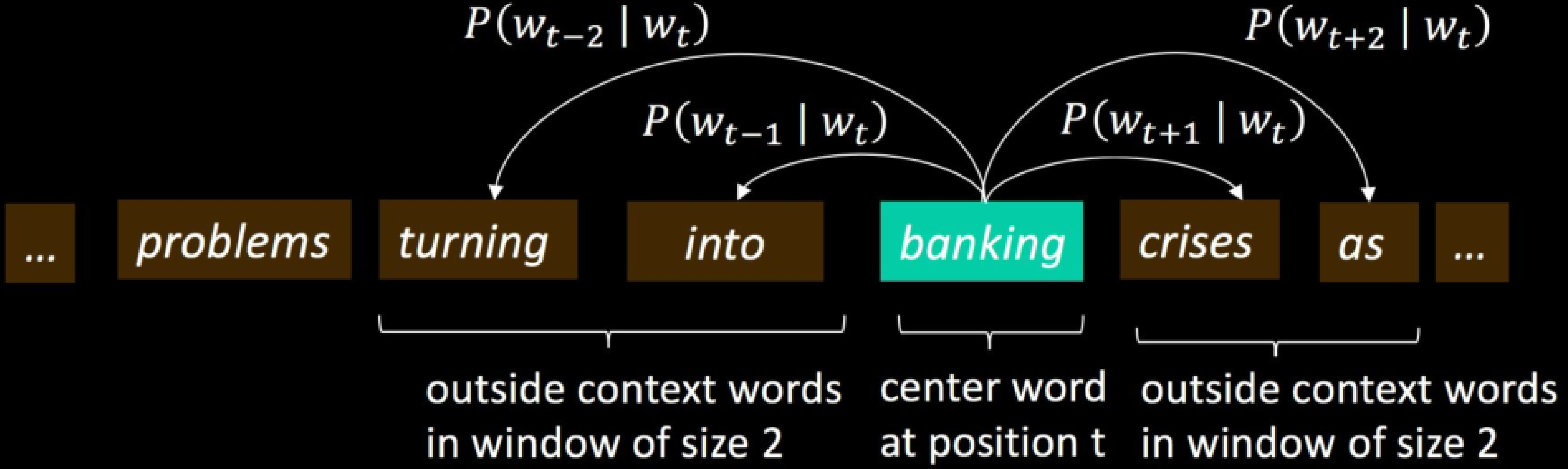
Source: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Embeddings: word2vec



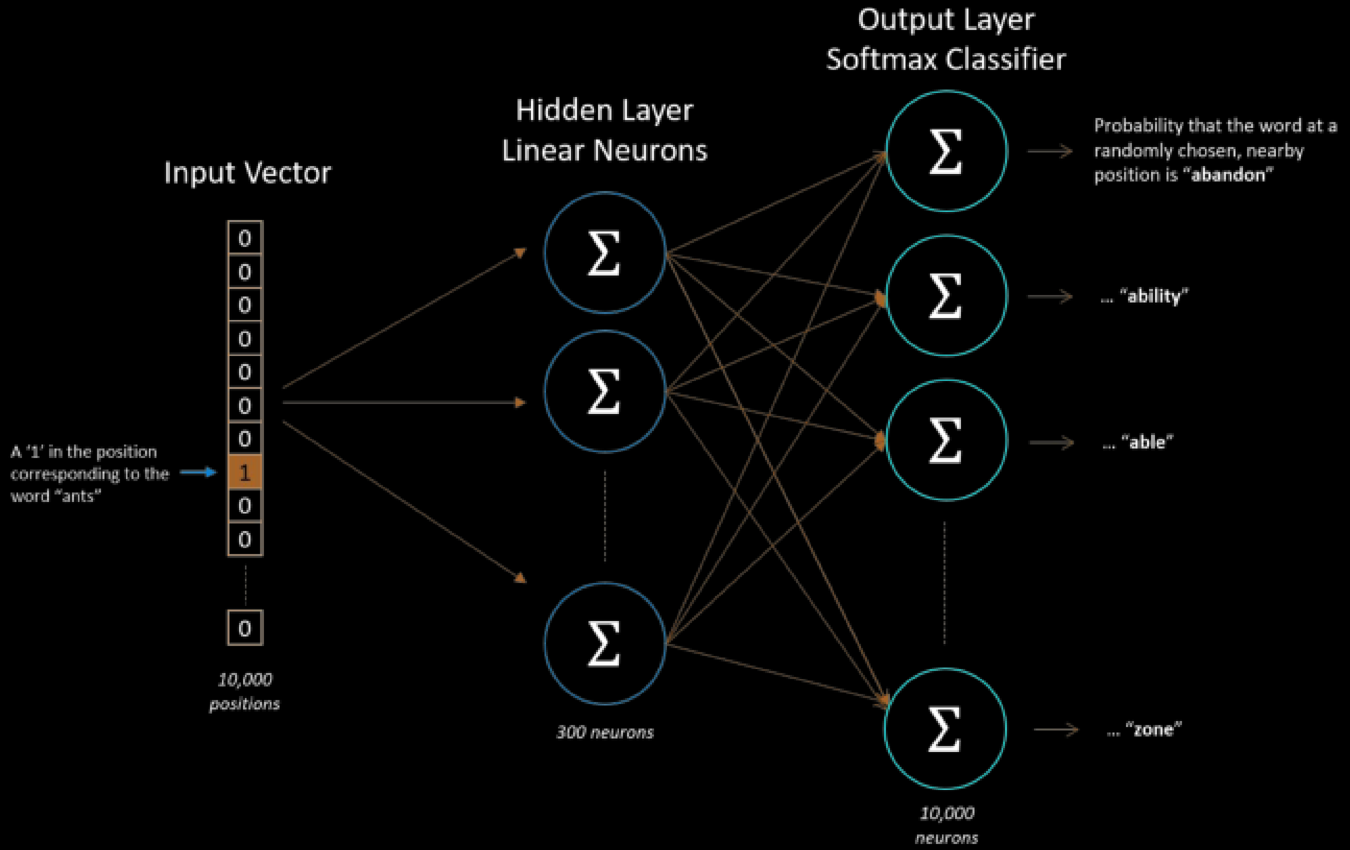
Source: CS224n: http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture01-wordvecs1.pdf

Embeddings: word2vec



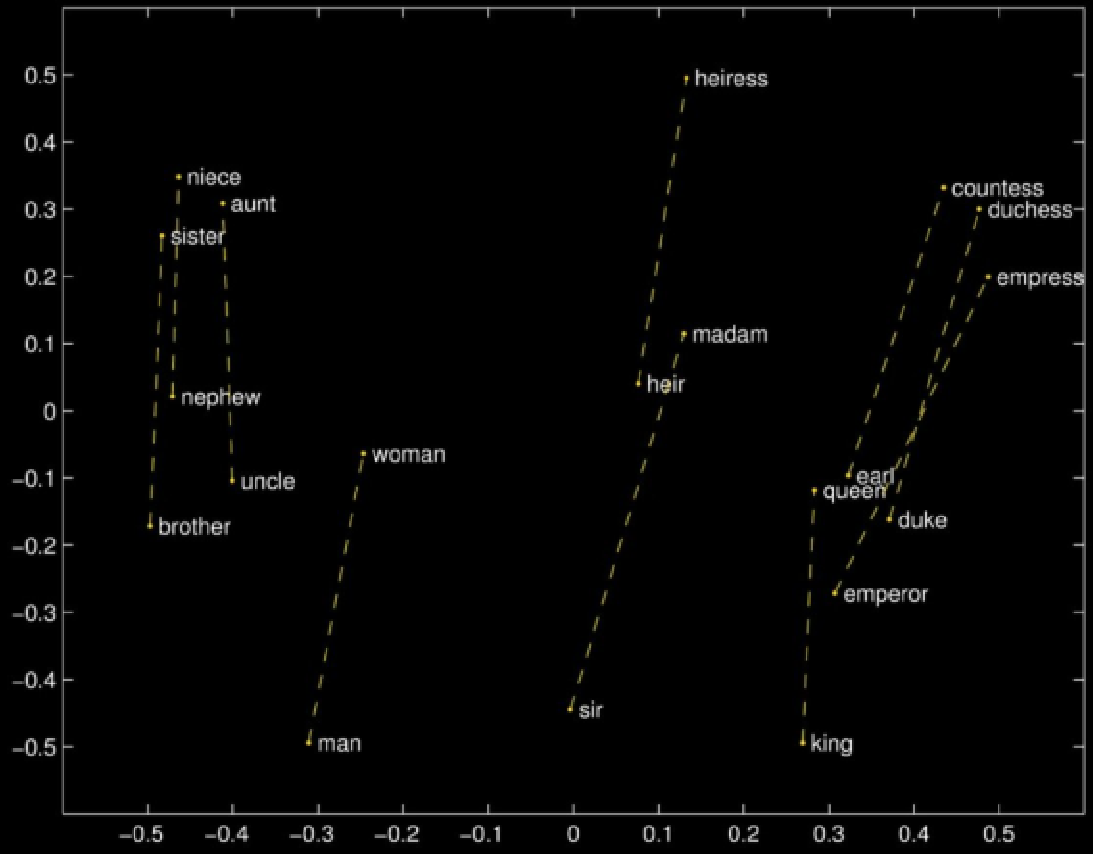
Source: CS224n: http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture01-wordvecs1.pdf

Embeddings: word2vec

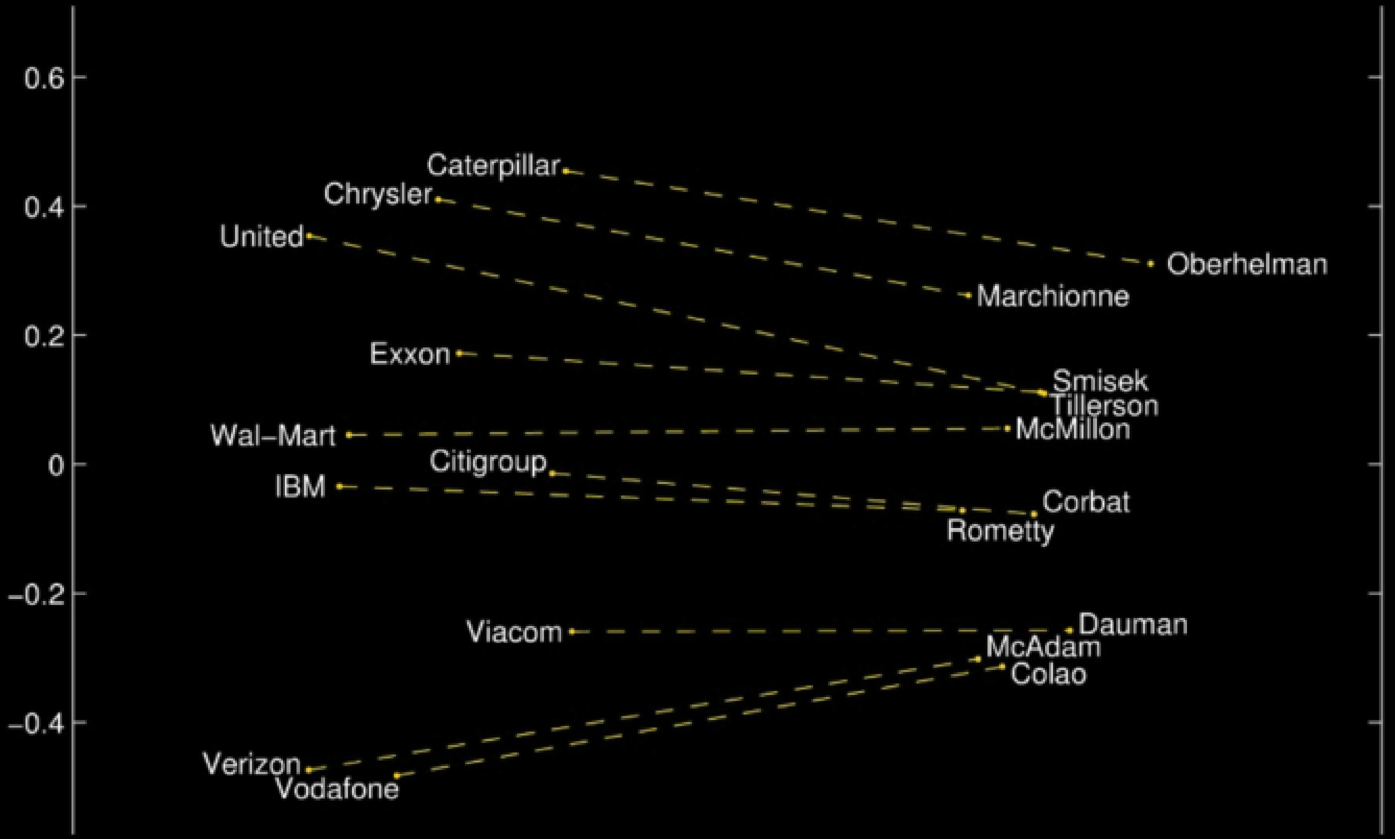


Source: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

GloVe Visualizations



GloVe Visualizations: Company - CEO

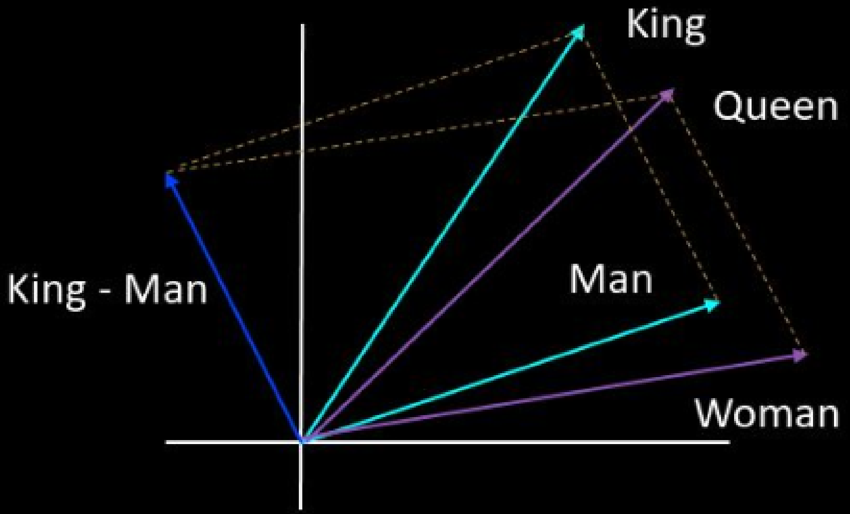


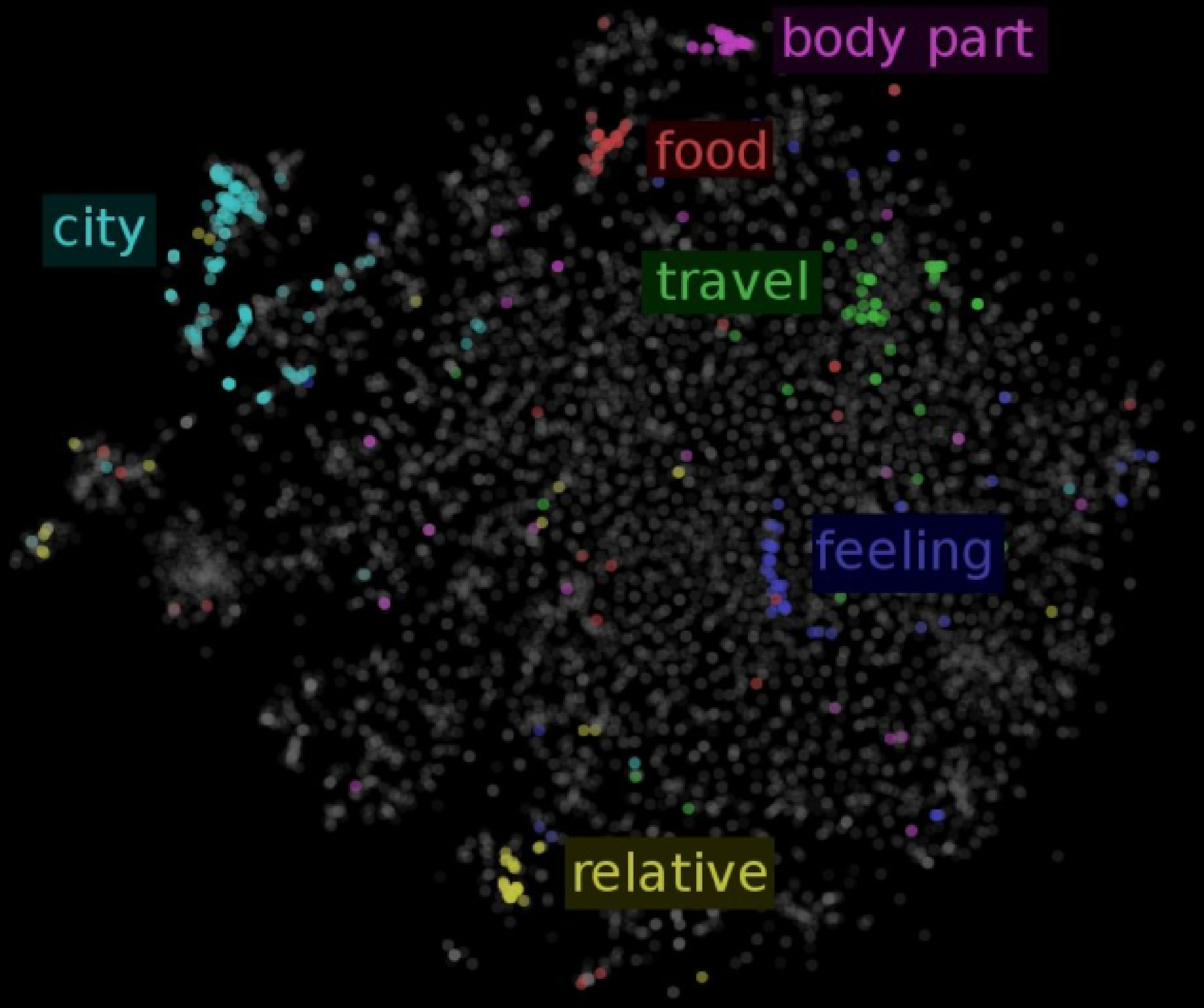
Word2vec: word analogies

King - man + woman = queen

$$\underset{x}{\text{King}} - \underset{y}{\text{man}} + \underset{y'}{\text{woman}} = \underset{target}{\text{queen}}$$

$$\cos(x - y + y', target) \rightarrow \max_{target}$$




body part
food
city
travel
feeling
relative

- Use statistics:
  - T-criterion

$$t=\frac{\overline{x}-\mu}{\sqrt{\frac{s^2}{N}}}$$

H_0 : ‘social media’ occurs with probability:

$$\mu=P(social)P(media)=\frac{C(social)(media)}{N^2}$$

H_a : ‘social media’ does not occur with such a probability

Collocations

- Use statistics:
  - Chi-squared

$$\chi^2 = \sum_{ij} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

$$E(social\ \ media) = \frac{C(social)}{N} \cdot \frac{C(media)}{N} \cdot N$$

$$O_{ij}\ \ from\ \ table$$

	w1 = social	w1 != social
w2 = media	C(social media)	C(x media) where x could be any word
w2 != media	C(social x) where x could be any word	C(any pair not starting with social or ending with media)