# IN4080 – 2020 FALL
## NATURAL LANGUAGE PROCESSING

Jan Tore Lønning

# Vectors, Distributions, Embeddings

Lecture 5, Sept 14

# Today



- Lexical semantics
- Vector models of documents
- tf-idf weighting
- Word-context matrices
- Word embeddings with dense vectors

# The meaning of words



- Words (lecture 2)
  - Type – token
  - Word – lexeme – lemma
  - Meaning?

# Look into the dictionary



lemma

sense

definition

pepper, *n.*

Pronunciation: Brit. /ˈpɛpə/ , U.S. /ˈpɛpər/

Forms: OE peopor (*rare*), OE pipcer (transmission error), OE pipor , OE pipur (*rare*

Frequency (in current use):

Etymology: A borrowing from Latin. Etymon: Latin *piper*.

< classical Latin *piper* , a loanword < Indo-Aryan (as is ancient Greek *πίπερι* ); compare Sa

I. The spice or the plant.

1.

a. A hot pungent spice derived from the prepared fruits (peppercorns) of the pepper plant, *Piper nigrum* (see sense 2a), used from early times to season food, either whole or ground to powder (often in association with salt). Also (locally, chiefly with distinguishing word): a similar spice derived from the fruits of certain other species of the genus *Piper*; the fruits themselves.

The ground spice from *Piper nigrum* comes in two forms, the more pungent *black pepper* , produced from black peppercorns, and the milder *white pepper* , produced from white peppercorns: see BLACK *adj.* and *n.* Special uses 5a, PEPPERCORN *n.* 1a, and WHITE *adj.* and *n.*[1] Special uses 7b(a).

2.

a. The plant *Piper nigrum* (family Piperaceae), a climbing shrub indigenous to South Asia and also cultivated elsewhere in the tropics, which has alternate stalked entire leaves, with pendulous spikes of small green flowers opposite the leaves, succeeded by small berries turning red when ripe. Also more widely: any plant of the genus *Piper* or the family Piperaceae.

b. Usu. with distinguishing word: any of numerous plants of other families having hot pungent fruits or leaves which resemble pepper ( 1a) in taste and in some cases are used as a substitute for it.

c. *U.S.* The California pepper tree, *Schinus molle*. Cf. PEPPER TREE *n.* 3.

3. Any of various forms of capsicum, esp. *Capsicum annuum* var. *annuum*. Originally (chiefly with distinguishing word): any variety of the *C. annuum* Longum group, with elongated fruits having a hot, pungent taste, the source of cayenne, chilli powder, paprika, etc., or of the

- A word with several senses is called polysemous
- If two different words look and sound the same, they are called homonyms

- How to tell: one word or several?
  - Common origin
  - But not waterproof/easy to see

# Relations between senses



| Term | Definition | Examples | | |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Relations between senses



| Term | Definition | Examples | | |
|---|---|---|---|---|
| Synonymy | Have the same meaning in all(?)/some(?) contexts | *sofa-couch, bus-coach*<br>*big-large* | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Relations between senses



| Term | Definition | Examples | | |
|---|---|---|---|---|
| Synonymy | Have the same meaning in all(?)/some(?) contexts | *sofa-couch, bus-coach big-large* | | |
| Antonymy | Opposites with respect to a feature of meaning | *true-false, strong-weak, up-down* | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Relations between senses



| Term | Definition | Examples | | |
|---|---|---|---|---|
| Synonymy | Have the same meaning in all(?)/some(?) contexts | *sofa-couch, bus-coach big-large* | | |
| Antonymy | Opposites with respect to a feature of meaning | *true-false, strong-weak, up-down* | | |
| Hyponym-hyperonym | The <hyponym> is a type-of the <hyperonym> | *rose →flower, cow →animal, car →vehicle* | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Relations between senses



| Term | Definition | Examples | | |
|---|---|---|---|---|
| Synonymy | Have the same meaning in all(?)/some(?) contexts | *sofa-couch, bus-coach*<br>*big-large* | | |
| Antonymy | Opposites with respect to a feature of meaning | *true-false, strong-weak, up-down* | | |
| Hyponym-hyperonym | The <hyponym> is a type-of the <hyperonym> | *rose →flower, cow →animal,*<br>*car →vehicle* | | |
| | | | | |
| Similarity | | *cow-horse*<br>*boy-girl* | | |
| | | | | |
| | | | | |

# Relations between senses

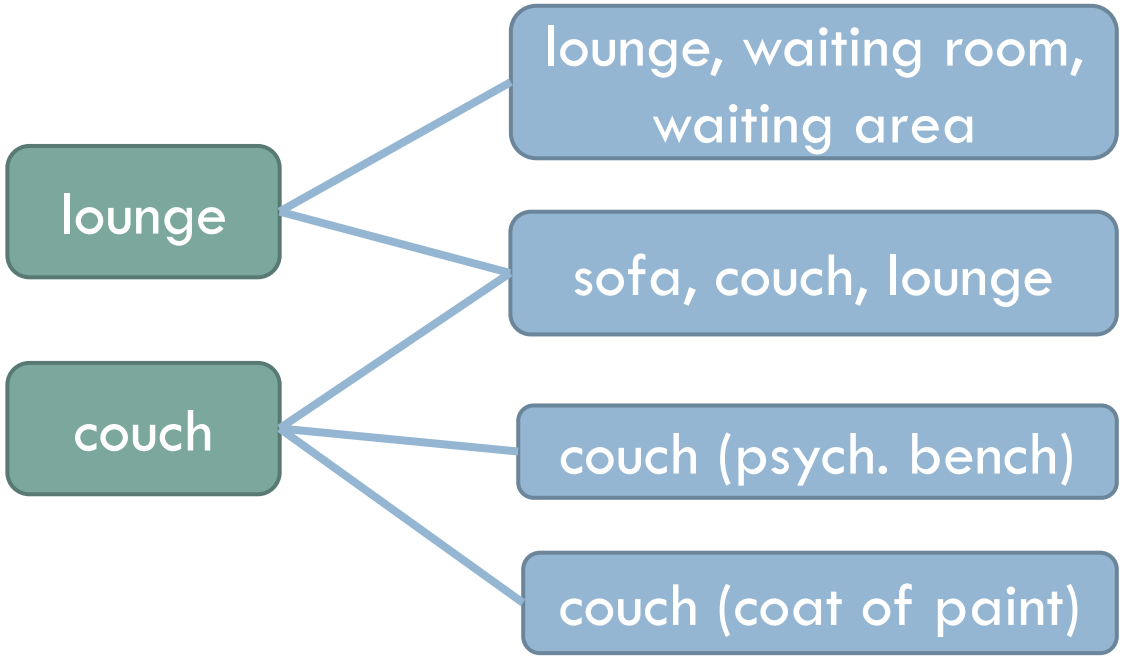


| Term | Definition | Examples | | |
|---|---|---|---|---|
| Synonymy | Have the same meaning in all(?)/some(?) contexts | *sofa-couch, bus-coach*<br>*big-large* | | |
| Antonymy | Opposites with respect to a feature of meaning | *true-false, strong-weak, up-down* | | |
| Hyponym-hyperonym | The <hyponym> is a type-of the <hyperonym> | *rose →flower, cow →animal,*<br>*car →vehicle* | | |
| | | | | |
| Similarity | | *cow-horse*<br>*boy-girl* | | |
| Related | | *money-bank*<br>*fish-water* | | |
| | | | | |

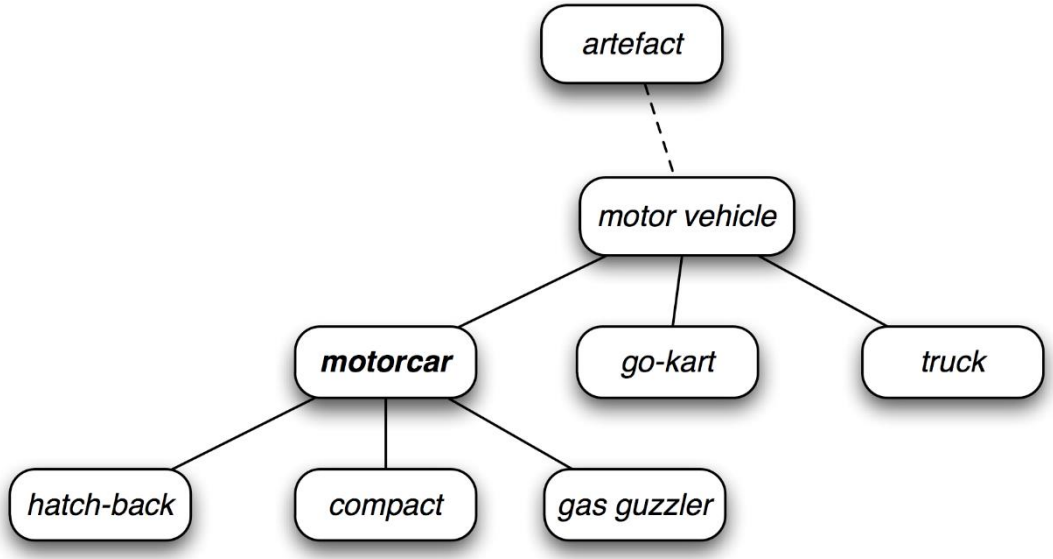
# Resources for lexical semantics: WordNet



- https://wordnet.princeton.edu
- To each word:
  - One or more synsets




- Relations between the synsets



# What does ongchoi mean?



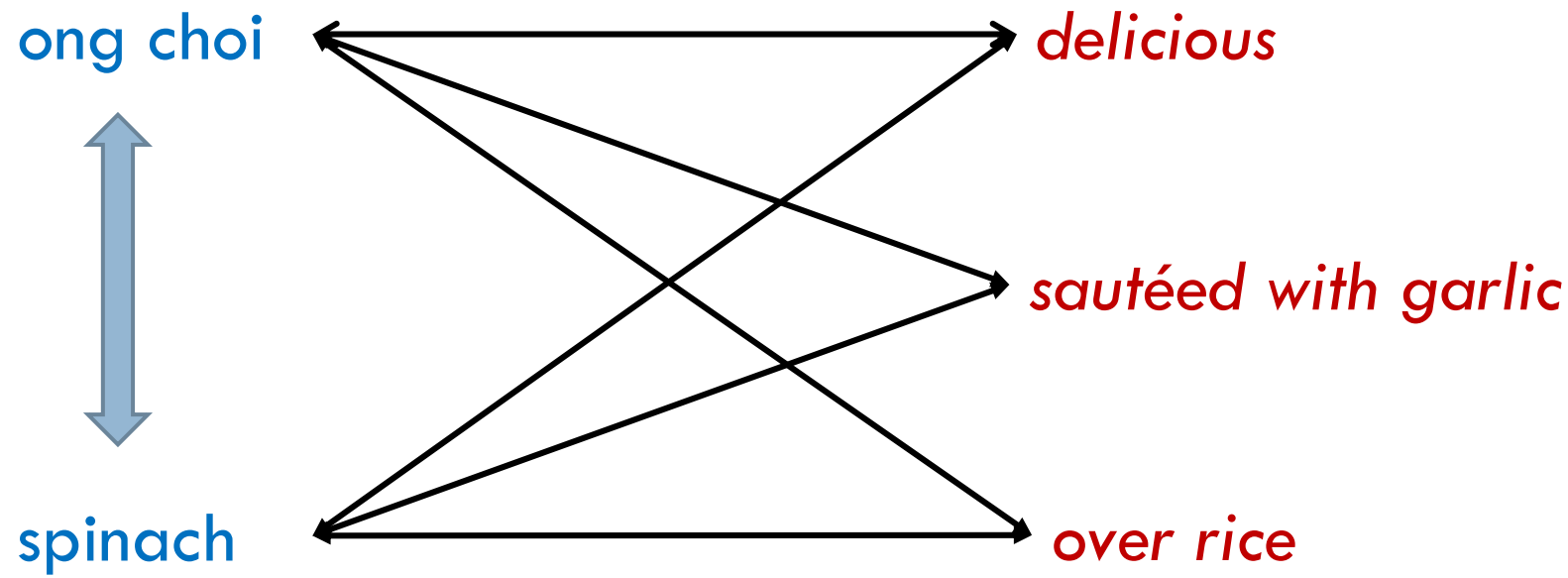
- Suppose you see these sentences:
  - *Ong choi is delicious sautéed with garlic.*
  - *Ong choi is superb over rice*
  - *Ong choi leaves with salty sauces*
- And you've also seen these:
  - *…spinach sautéed with garlic over rice*
  - *Chard stems and leaves are delicious*
  - *Collard greens and other salty leafy greens*
- Conclusion: Ongchoi is a leafy green like spinach, chard, or collard greens

# Similar



Related

(first-order association, syntagmatic)




Similar

(second-order association, paradigmatic)

# The distributional hypothesis



- Words that occur in similar contexts have similar meanings

# Today



- Lexical semantics
- Vector models of documents
- tf-idf weighting
- Word-context matrices
- Word embeddings with dense vectors

# Shakespeare (from J & M)



| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

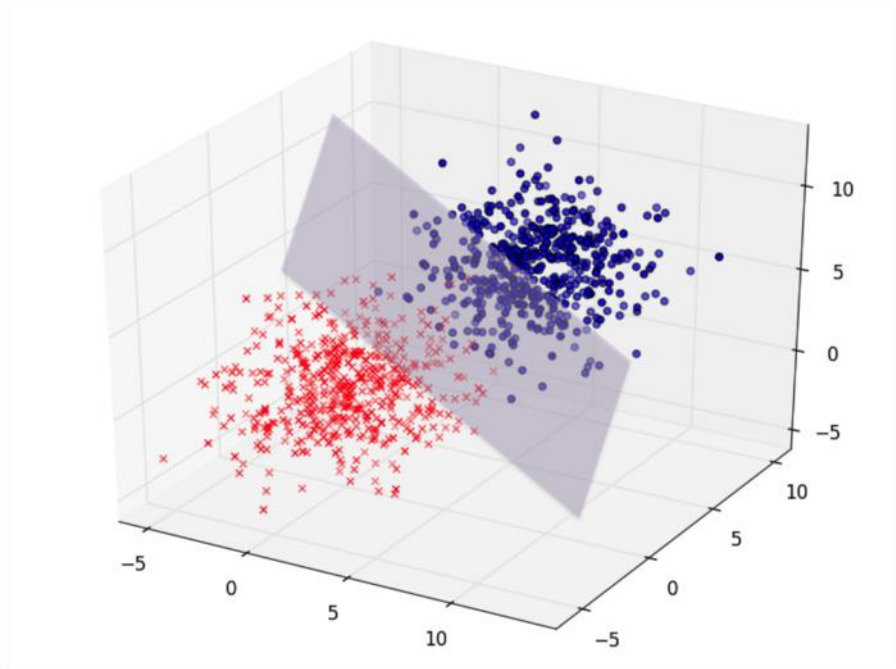
- Vectors are similar for the two comedies
- Different than the historical dramas
- Comedies have more fools and wit and fewer battles.

- Notice similarity to text classification
- Mandatory 2A, multinomial
  - The document represented by a vector with the occurrences of 35,000 terms

# Document classification



- The word vectors were used as basis for classification
- If two documents had the same vectors they were put in the same class
- Documents are similar = on the same side of the separating hyperplane

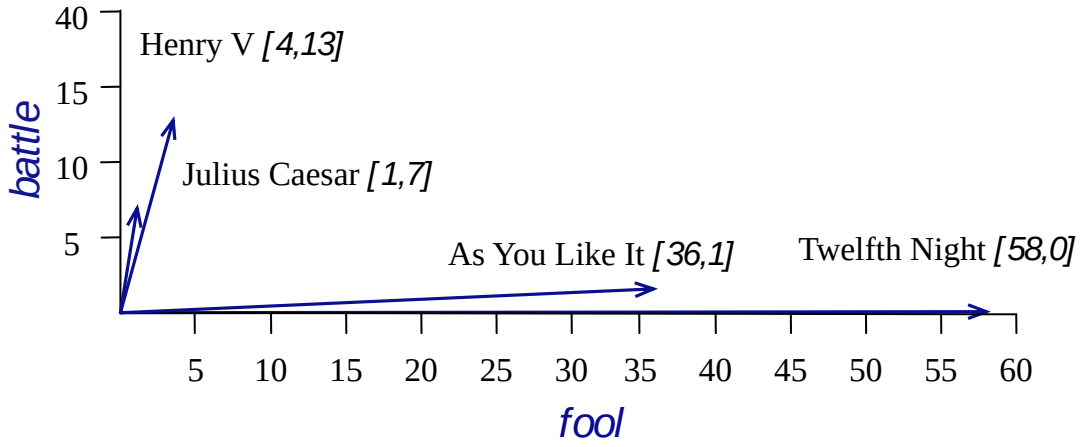



A problem to draw 35,000 dimensions

# Information retrieval (IR)



- Documents placed in the same *n*-dimensional space as in classification
- Retrieve documents similar to a given document

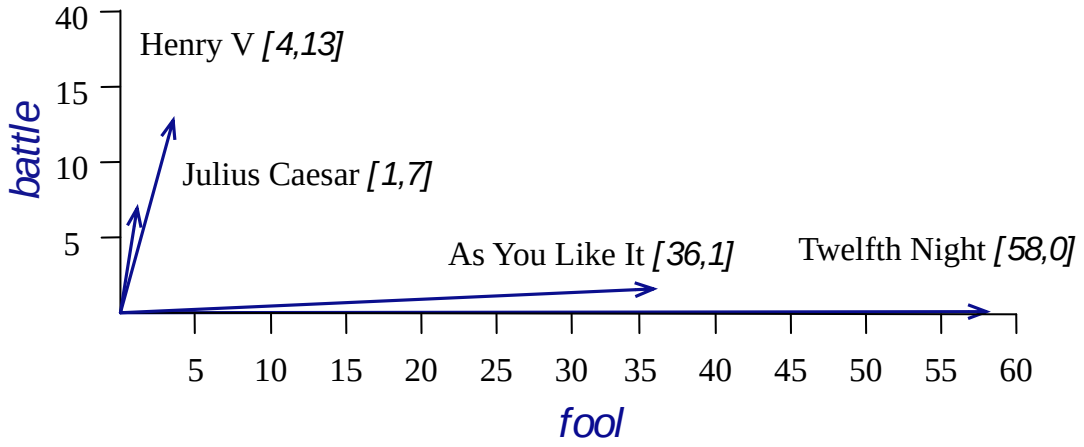


# Cosine similarity



- Several possible ways to define similarity, e.g.,
  - Euclidean
  - Manhattan
- Most common: cosine
  - Do the arrows point in the same direction?

$$\cos(\vec{V},\vec{W}) = \frac{\vec{V}\cdot\vec{W}}{|\vec{V}||\vec{W}|} = \frac{\vec{V}}{|\vec{V}|}\cdot\frac{\vec{W}}{|\vec{W}|} = \frac{\sum_{i=1}^{N} V_i W_i}{\sqrt{\sum_{i=1}^{N} V_i^2}\sqrt{\sum_{i=1}^{N} W_i^2}}$$



# Let us try: $\cos(v_1, v_2)$



## Full vectors

| | AYLI | TwNi | JuCa | HenV |
|---|---|---|---|---|
| AYLI | 1.000 | 0.950 | 0.945 | 0.949 |
| TwNi | 0.950 | 1.000 | 0.809 | 0.822 |
| JuCa | 0.945 | 0.809 | 1.000 | 0.999 |
| HenV | 0.949 | 0.822 | 0.999 | 1.000 |

## *battles & fools*

| | AYLI | TwNi | JuCa | HenV |
|---|---|---|---|---|
| AYLI | 1.000 | 1.000 | 0.169 | 0.321 |
| TwNi | 1.000 | 1.000 | 0.141 | 0.294 |
| JuCa | 0.169 | 0.141 | 1.000 | 0.988 |
| HenV | 0.321 | 0.294 | 0.988 | 1.000 |

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| **battle** | 1 | 0 | 7 | 13 |
| **good** | 114 | 80 | 62 | 89 |
| **fool** | 36 | 58 | 1 | 4 |
| **wit** | 20 | 15 | 2 | 3 |

# Today



- Lexical semantics
- Vector models of documents
- tf-idf weighting
- Word-context matrices
- Word embeddings with dense vectors

# Ways of counting: Term frequency



Alternatives

- Raw counts/absolute frequencies, TeNi = (0, 80, 58, 15)
- Binary counts (Mandatory 2A), TeNi = (0, 1, 1, 1)
- Variants of normalization.
  - Rel. frequency, $(0, \frac{80}{80+58+15}, \frac{58}{80+58+15}, \frac{15}{80+58+15})$
    - `TfidfTransformer(use_idf=False, norm = "l1")`
  - Length normalize, $(0, \frac{80}{\sqrt{80^2+58^2+15^2}}, \frac{58}{\sqrt{80^2+58^2+15^2}}, \frac{15}{\sqrt{80^2+58^2+15^2}})$
    - `TfidfTransformer(use_idf=False, norm = "l2")`
  - Sublinear TF: (1 + log(tf)), 0 when tf=0
    - `TfidfTransformer(use_idf=False, sub_linear=True)`

# Normalize or not?



- The cos-similarity measure does a form of length normalization:
  - Raw counts, relative counts, length normalized counts yield the same
- For other measures, it matters whether we normalize
  - e.g. L2-distance is relative large between documents of different lengths
- The sublinear squeezing distinguish between terms that occur often and terms that occurs very often:
  - If *term1* occurs 100 times and *term2* occurs 10 times:
  - *term1* will be considered 10 times more frequent than *term2*
  - but only 2 times as important with sublinear

# Inverse document frequency



- Intuition: A word occurring in a large proportion of documents is not a good discriminator.
- $idf_t = \log \frac{N}{df_t}$
  - $df_t$ the number of documents containing $t$.
- `TfidfTransformer(use_idf=True, smooth_idf=False)`
- Smooth: avoid dividing by zero
  - $idf_t = \log \frac{N}{df_t+1} + 1$
  - `TfidfTransformer(use_idf=True, smooth_idf=True)`

# $tf\text{-}idf$



- Tf-idf weighting
- $tf_{t,d} \times idf_t$
- `TfidfTransformer()`

- (Other ways of weighting:
  - PMI – pointwise mutual information
  - … and more)

# The effect of tf-idf



| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| **battle** | 1 | 0 | 7 | 13 |
| **good** | 114 | 80 | 62 | 89 |
| **fool** | 36 | 58 | 1 | 4 |
| **wit** | 20 | 15 | 2 | 3 |

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| **battle** | 0.074 | 0 | 0.22 | 0.28 |
| **good** | 0 | 0 | 0 | 0 |
| **fool** | 0.019 | 0.021 | 0.0036 | 0.0083 |
| **wit** | 0.049 | 0.044 | 0.018 | 0.022 |

**Figure 6.8** A tf-idf weighted term-document matrix for four words in four Shakespeare plays, using the counts in Fig. 6.2. For example the 0.049 value for *wit* in *As You Like It* is the product of $\text{tf} = \log_{10}(20+1) = 1.322$ and $\text{idf} = .037$. Note that the idf weighting has eliminated the importance of the ubiquitous word *good* and vastly reduced the impact of the almost-ubiquitous word *fool*.

# Today



- Lexical semantics
- Vector models of documents
- tf-idf weighting
- Word-context matrices
- Word embeddings with dense vectors

# Word-context matrix



- Two **words** are similar in meaning if their context vectors are similar

sugar, a sliced lemon, a tablespoonful of **apricot** jam, a pinch each of,
their enjoyment. Cautiously she sampled her first **pineapple** and another fruit whose taste she likened
well suited to programming on the digital **computer**. In finding the optimal R-stage policy from
for the purpose of gathering data and **information** necessary for the study authorized in the

| | aardvark | computer | data | pinch | result | sugar | … |
|---|---|---|---|---|---|---|---|
| apricot | 0 | 0 | 0 | 1 | 0 | 1 | |
| pineapple | 0 | 0 | 0 | 1 | 0 | 1 | |
| digital | 0 | 2 | 1 | 0 | 1 | 0 | |
| information | 0 | 1 | 6 | 0 | 4 | 0 | |

# Word-context matrix



## Document-term matrix

- Objects: a set of documents, D
- Features: a set of terms,
  - $T = \{t_1, t_2, \ldots, t_n\}$
- Each document $d$ is identified with a vector
  - $(v_1, v_2, \ldots, v_n)$
  - where $v_i$is calculated from the frequency of $t_i$ in $d$.

## Word-context matrix

- Objects: a vocabulary of words, V
- Features: a set of words,
  - $C = \{c_1, c_2, \ldots, c_n\}$
- A set of texts, T
- A definition of the context of an occurrence of w in T
- Each word $w$ in V is identified with a vector
  - $(v_1, v_2, \ldots, v_n)$
  - where $v_i$is calculated from the frequency of $c_i$ in all the contexts of $w$ in T

# Word-context matrix



## Comments

- C=V, or C is smaller set of the most frequent terms
  - To avoid to large repr.
- Context, alternatives:
  - A sentence
  - A window of *k* tokens on each side
  - A document
  - Defined by grammatical relations (after parsing)

## Word-context matrix

- Objects: a vocabulary of words, V
- Features: a set of words,
  - $C = \{c_1, c_2, \ldots, c_n\}$
- A set of texts, T
- A definition of the context of an occurrence of w in T
- Each word $w$ in V is identified with a vector
  - $(v_1, v_2, \ldots, v_n)$
  - where $v_i$is calculated from the frequency of $c_i$ in all the contexts of $w$ in T

# So-far



- A word $w$ can be represented by a context vector $v_w$ where position $j$in the vector reflects the frequency of occurrences of $w_j$ with $w$.
- Can be used for
  - studying similarities between words.
  - document similarities
- But the vectors are *sparse*
  - Long: 20-50,000
  - Many entries are 0
- Even though *car* and *automobile* get similar vectors, because both co-occur with e.g., *drive*, in the vector for *drive* there is no connection between the *car* element and the *automobile* element.

# Today



- Lexical semantics
- Vector models of documents
- tf-idf weighting
- Word-context matrices
- Word embeddings with dense vectors

# Dense vectors



## How?

- Shorter vectors.
  - (length 50-1000)
  - ``low-dimensional'' space
- Dense (most elements are not 0)
- Intuitions:
  - Similar words should have similar vectors.
  - Words that occur in similar contexts should be similar.

## Properties

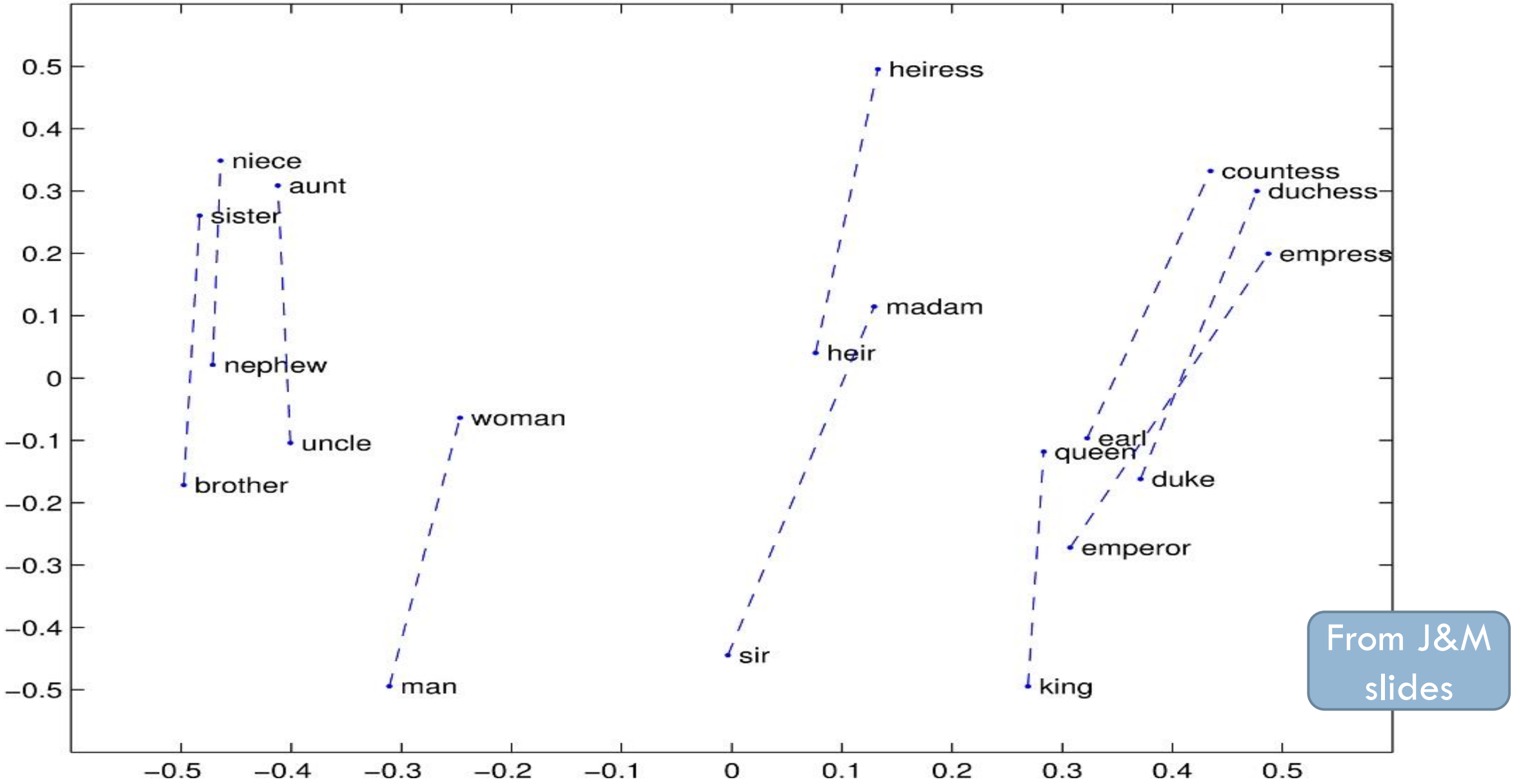
- Generalize better than sparse vectors.
- Input for deep learning
  - Fewer weights (or other weights)
- Capture semantic similarities better.
- Better for sequence modelling:
  - Language models, etc.

# Word embeddings

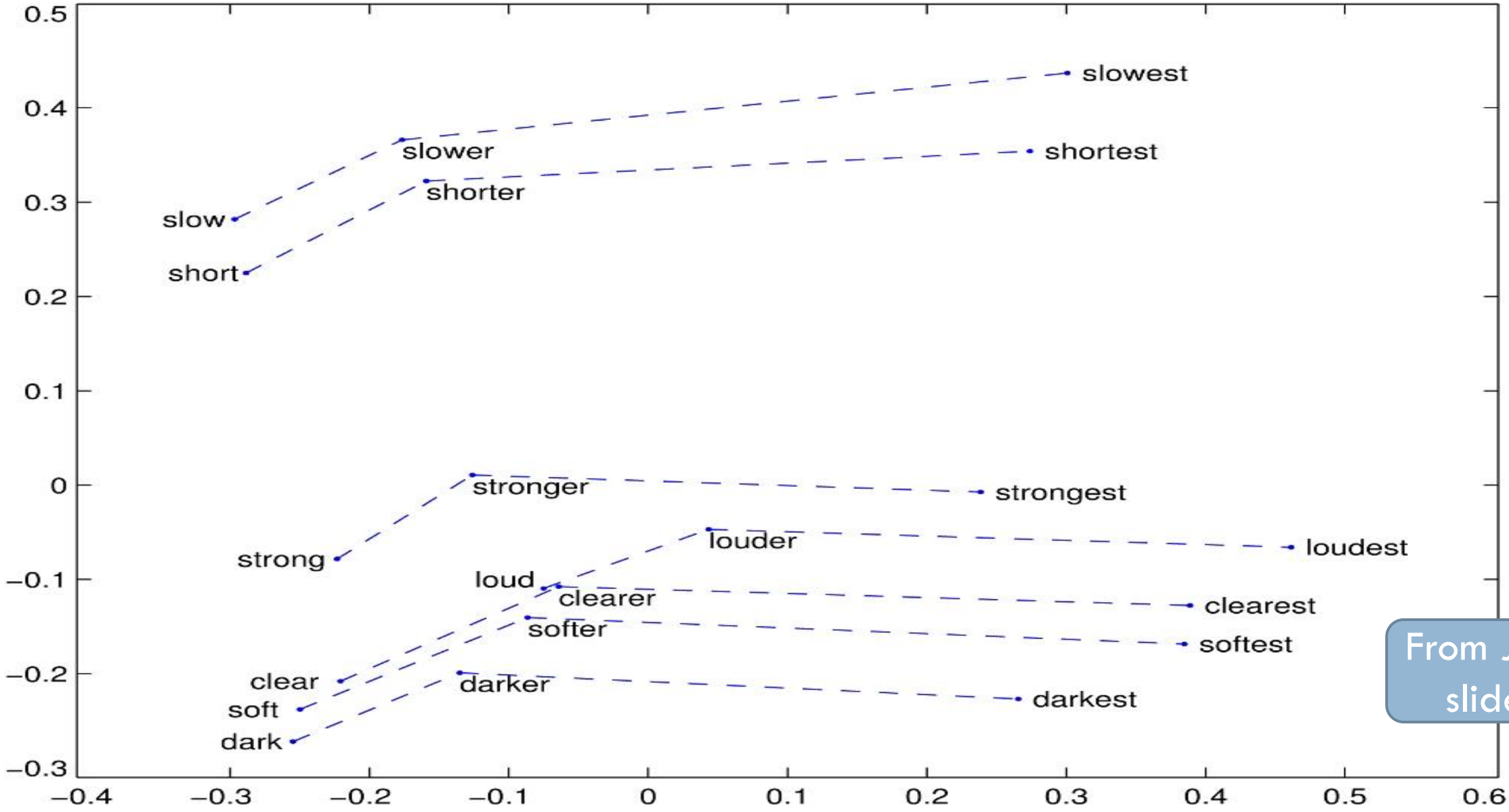


- In current LT: Each word is represented as a vector of reals
- Words are more or less similar
- A word can be similar to one word in some dimensions and other words in other dimensions


Figure from https://medium.com/@jayeshbahire




From J&M slides



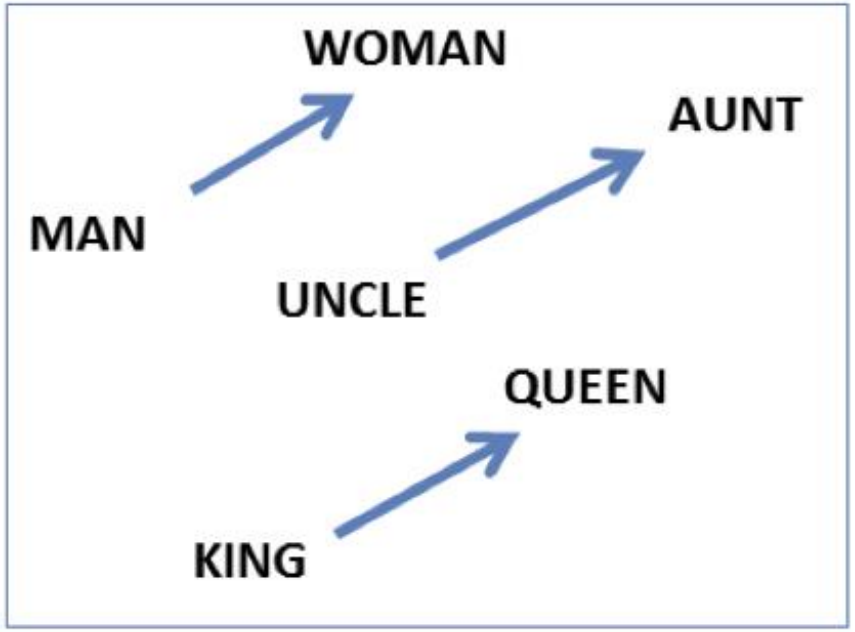

From J&M slides

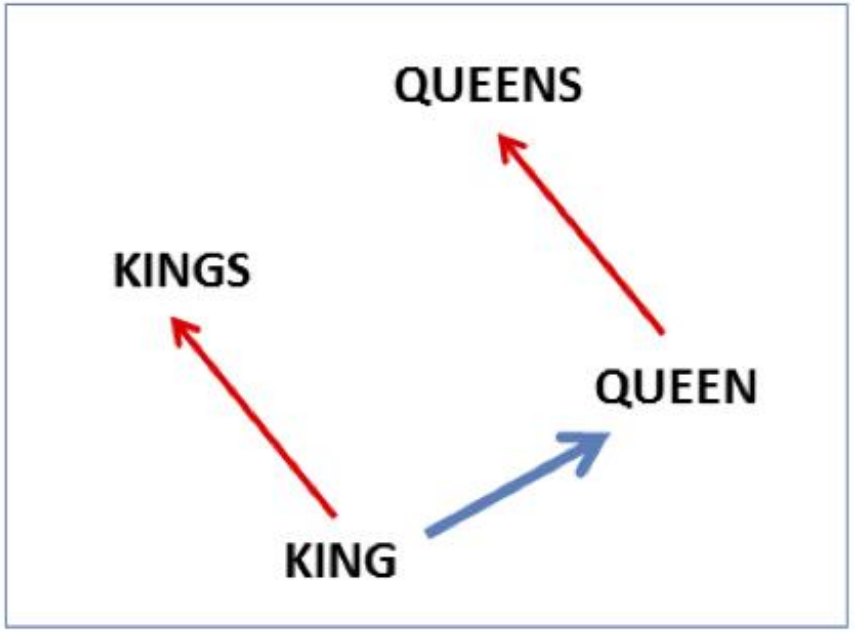
# Analogy: Embeddings capture relational meaning!



vector('*king*') - vector('*man*') + vector('*woman*') ≈ vector('queen')

vector('*Paris*') - vector('*France*') + vector('*Italy*') ≈ vector('Rome')




From J&M slides

# Track change of meaning of words

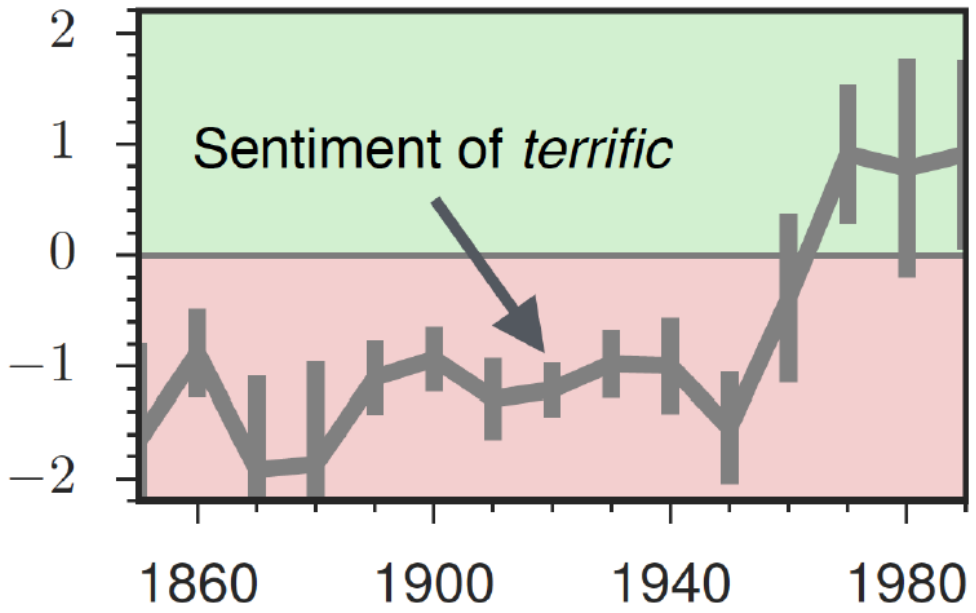



~30 million books, 1850-1990, Google Books data

From J&M slides

# Evolution of sentiment words






- Negative words change faster than positive words

From J&M slides

# Bias



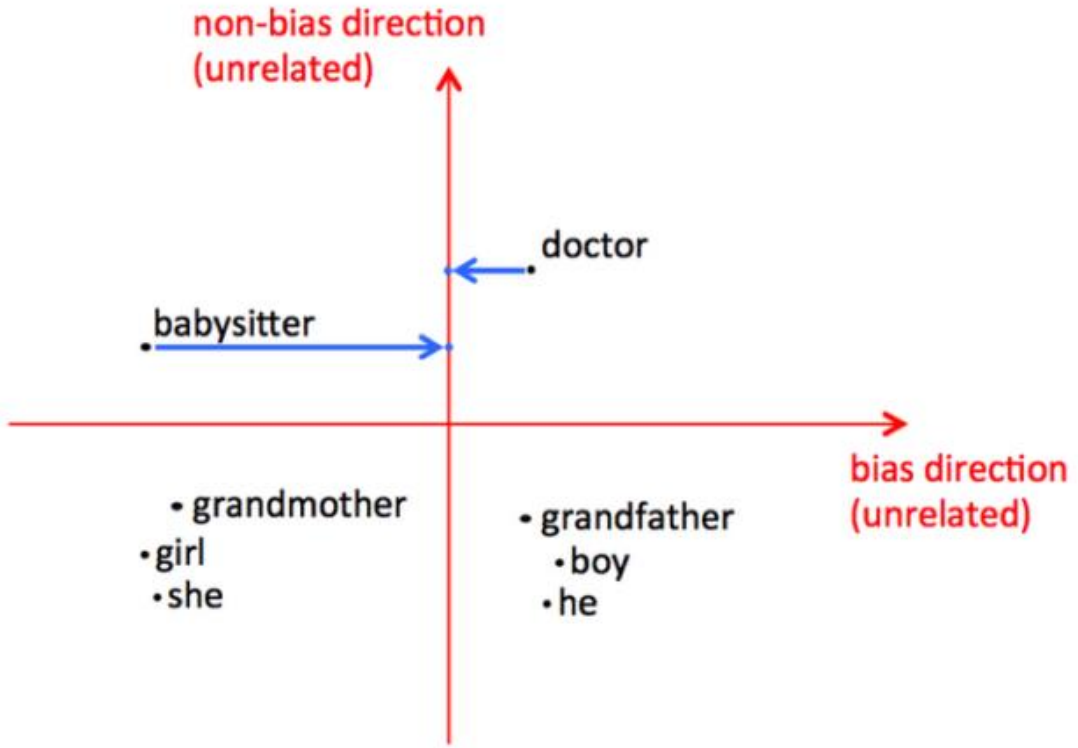
- *Man is to computer programmer as woman is to homemaker*.
- Different adjectives associated with:
  - male and female terms
  - typical black names and typical white names
- Embeddings may be used to study historical bias

# Debiasing (research topic)



- Goal: neutralize the biases
- Some positive results
- But also reports that is is not fully possible

- Is debiasing a goal?
- When should we (not) debias?




https://vagdevik.wordpress.com/2018/07/08/debiasing-word-embeddings/

# Demo



- http://vectors.nlpl.eu/explore/embeddings/en/

# Evaluation

- Extrinsic evaluation:
  - Evaluate contribution as part of an application
- Intrinsic evaluation:
  - Evaluate against a resource
- Some datasets
  - WordSim-353:
    - Broader "semantic relatedness"
  - SimLex-999:
    - Narrower: similarity
    - Manually annotated for similarity

| Word1 | Word2 | POS | Sim-score |
|---|---|---|---|
| old | new | A | 1.58 |
| smart | intelligent | A | 9.2 |
| plane | jet | N | 8.1 |
| woman | man | N | 3.33 |
| word | dictionary | N | 3.68 |
| create | build | V | 8.48 |
| get | put | V | 1.98 |
| keep | protect | V | 5.4 |

Part of SimLex-999

# Use of embeddings



- Embeddings are used as representations for words as input in all kinds of NLP tasks using deep learning:
  - Text classification
  - Language models
  - Named-entity recognition
  - Machine translation
  - etc.

# Where do the dense embeddings come from?



- Next week

# Resources



- gensim
  - Easy-to-use tool for training own models
- Word2wec
  - https://code.google.com/archive/p/word2vec/
- https://fasttext.cc/
- https://nlp.stanford.edu/projects/glove/
- http://vectors.nlpl.eu/repository/
  - Pretrained embeddings, also for Norwegian