Lecture 21

Distributional Semantics

Nathan Schneider

(with slides by Marine Carpuat)

ANLP | 27 November 2017

Learning Paradigms

Rule-based ("symbolic")

- e.g. finite-state morphology, WordNet similarity
  - Non-statistical: Expert specification of **exact relationship** between inputs and outputs, possibly established in a linguistic resource

Statistical

- **Supervised:** language modeling (n-gram), classification (naïve Bayes, perceptron), tagging (HMM), statistical parsing (PCFG)
  - General specification of **factors** that should influence the algorithm's decision-making; learning algorithm uses **labeled data** to determine which factors are predictive of which outputs (probabilities, feature weights)
- **Unsupervised:** word alignment, clustering (today)
  - General specification of **factors** that should influence the algorithm's decision-making; learning algorithm mines **unlabeled data** for latent structure/correlations, but sees no examples of desired outputs

Today: Semantics without Annotations

- Lexical semantics
  - Word similarity
  - Distributional hypothesis
  - Vector representations
  - Clustering
- Document “semantics”

Word Similarity

Jeff Kao Follow
Data Scientist, Software Engineer, Language Nerd, Biglaw Refugee. jeffykao.com
Nov 23 · 10 min read

More than a Million Pro-Repeal Net Neutrality Comments were Likely Faked

I used natural language processing techniques to analyze net neutrality comments submitted to the FCC from April-October 2017, and the results were disturbing.

```
"In the matter of restoring Internet freedom. I'd like to recommend the commission to undo The
Obama/Wheeler power grab to control Internet access. Americans, as opposed to Washington bureaucrats,
deserve to enjoy the services they desire. The Obama/Wheeler power grab to control Internet access is
a distortion of the open Internet. It ended a hands-off policy that worked exceptionally successfully
for many years with bipartisan support.",
"Chairman Pai:  With respect to Title 2 and net neutrality. I want to encourage the FCC to rescind
Barack Obama's scheme to take over Internet access. Individual citizens, as opposed to Washington
bureaucrats, should be able to select whichever services they desire. Barack Obama's scheme to take
over Internet access is a corruption of net neutrality. It ended a free-market approach that
performed remarkably smoothly for many years with bipartisan consensus.",
"FCC:  My comments re: net neutrality regulations. I want to suggest the commission to overturn
Obama's plan to take over the Internet. People like me, as opposed to so-called experts, should be
free to buy whatever products they choose. Obama's plan to take over the Internet is a corruption of
net neutrality. It broke a pro-consumer system that performed fabulously successfully for two decades
with Republican and Democrat support.",
"Mr Pai:  I'm very worried about restoring Internet freedom. I'd like to ask the FCC to overturn The
Obama/Wheeler policy to regulate the Internet. Citizens, rather than the FCC, deserve to use
whichever services we prefer. The Obama/Wheeler policy to regulate the Internet is a perversion of
the open Internet. It disrupted a market-based approach that functioned very, very smoothly for
decades with Republican and Democrat consensus.",
"FCC:  In reference to net neutrality. I would like to suggest Chairman Pai to reverse Obama's
scheme to control the web. Citizens, as opposed to Washington bureaucrats, should be empowered to buy
whatever products they prefer. Obama's scheme to control the web is a betrayal of the open Internet.
It undid a hands-off approach that functioned very, very successfully for decades with broad
```

Intuition of Semantic Similarity

Semantically close

- bank–money
- apple–fruit
- tree–forest
- bank–river
- pen–paper
- run–walk
- mistake–error
- car–wheel

Semantically distant

- doctor–beer
- painting–January
- money–river
- apple–penguin
- nurse–fruit
- pen–river
- clown–tramway
- car–algebra

Why are 2 words similar?

- Meaning
  - The two concepts are close in terms of their meaning
- World knowledge
  - The two concepts have similar properties, often occur together, or occur in similar contexts
- Psychology
  - We often think of the two concepts together

Why do this?

- Task: automatically compute semantic similarity between words
- Can be useful for many applications:
  - Detecting paraphrases (i.e., automatic essay grading, plagiarism detection)
  - Information retrieval
  - Machine translation
- Why? Because similarity gives us a way to generalize beyond word identities

Evaluation: Correlation with Humans

- Ask automatic method to rank word pairs in order of semantic distance
- Compare this ranking with human-created ranking
- Measure correlation

Evaluation: Word-Choice Problems

Identify that alternative which is closest in meaning to the target:

accidental	imprison
wheedle	incarcerate
ferment	writhe
inadvertent	meander
abominate	inhibit

Thesauri

- Previously we talked about dictionaries/thesauri that can help.

- But thesauri are not always available for the language of interest, or may not contain all the words in a corpus.

Distributional Similarity

"Differences of meaning correlates with differences of distribution" (Harris, 1970)

- Idea: similar linguistic objects have similar **contents** (for documents, sentences) or **contexts** (for words)

Two Kinds of Distributional Contexts

1. Documents as bags-of-words

   - Similar documents contain similar words; similar words appear in similar documents

2. Words in terms of neighboring words

   - "You shall know a word by the company it keeps!" (Firth, 1957)

   - Similar words occur near similar sets of other words (e.g., in a 5-word window)

- He handed her a glass of bardiwac.
- Beef dishes are made to complement the bardiwac.
- Nigel staggered to his feet, face flushed from too much bardiwac.
- Malbec, one of the lesser-known bardiwac grapes, responds well to Australia's sunshine.
- I dined off bread and cheese and this excellent bardiwac.
- The drinks were delicious: blood-red bardiwac as well as light, sweet Rhenish.

Word Vectors

- A word **type** can be represented as a vector of features indicating the contexts in which it occurs in a corpus

$$\vec{w} = (f_1, f_2, f_3, \cdots f_N)$$

Context Features

- Word co-occurrence within a window:

	arts	boil	data	function	large	sugar	summarized	water
apricot	0	1	0	0	1	1	0	1
pineapple	0	1	0	0	1	1	0	1
digital	0	0	1	1	1	0	1	0
information	0	0	1	1	1	0	1	0

- Grammatical relations:

	subj-of, absorb	*subj-of*, adapt	*subj-of*, behave	...	*pobj-of*, inside	*pobj-of*, into	...	*nmod-of*, abnormality	*nmod-of*, anemia	*nmod-of*, architecture	...	*obj-of*, attack	*obj-of*, call	*obj-of*, come from	*obj-of*, decorate	...	*nmod*, bacteria	*nmod*, body	*nmod*, bone marrow
cell	1	1	1		16	30		3	8	1		6	11	3	2		3	2	2

Context Features

- Feature values
  - Boolean
  - Raw counts
  - Some other weighting scheme (e.g., *idf, tf.idf*)
  - Association values (next slide)

Association Metric

- Commonly-used metric: Pointwise Mutual Information

$$\text{association}_{\text{PMI}}(w, f) = \log_2 \frac{P(w, f)}{P(w)P(f)}$$

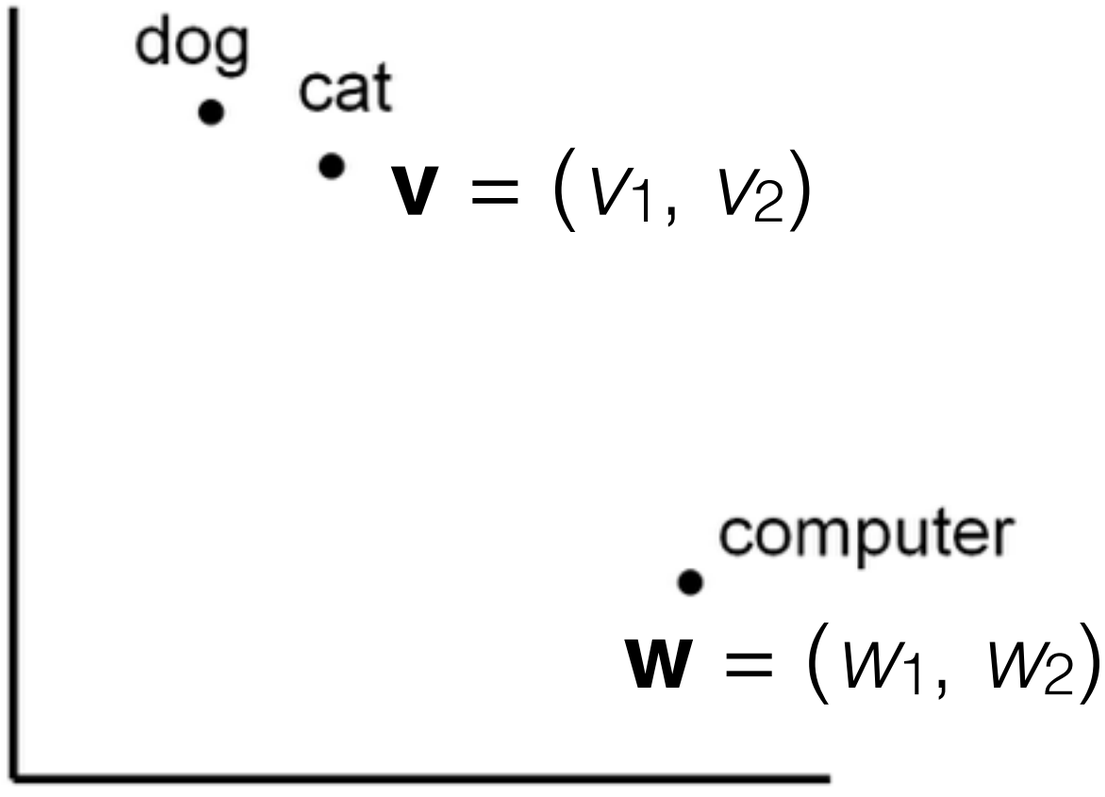
- Can be used as a feature value or by itself

Computing Similarity

- Semantic similarity boils down to computing some measure on context vectors

Words in a Vector Space

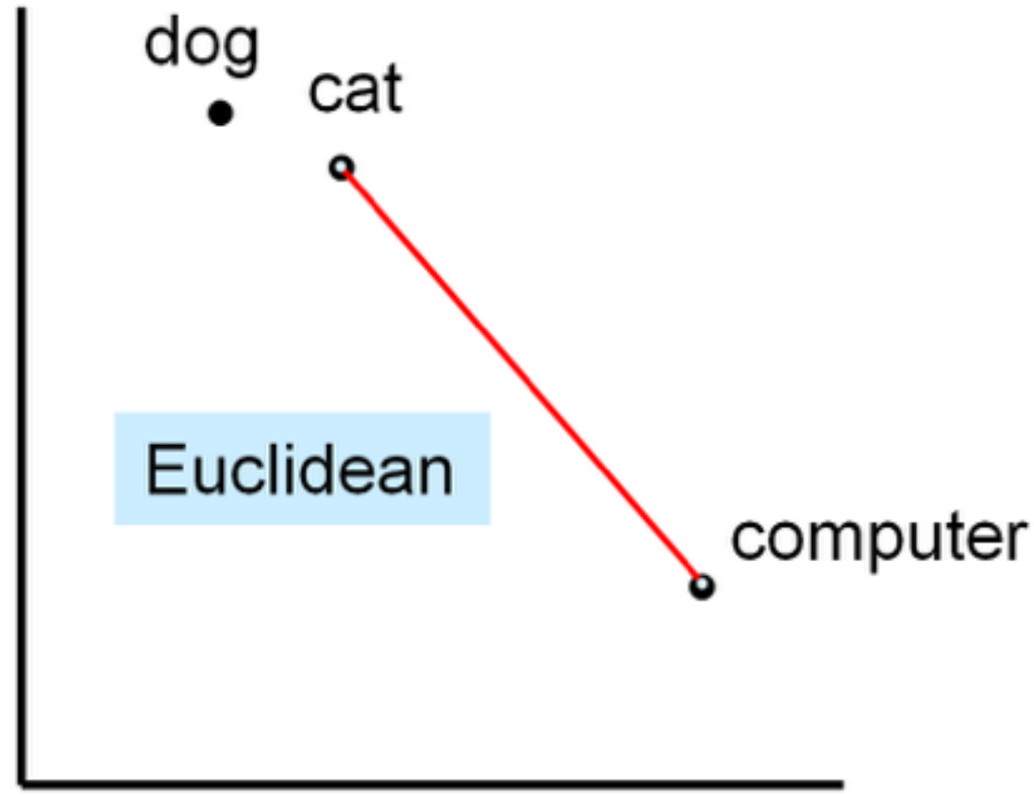
- In 2 dimensions:
  v = “cat”
  w = “computer”



Euclidean Distance

- $\sqrt{\sum_i (v_i - w_i)^2}$

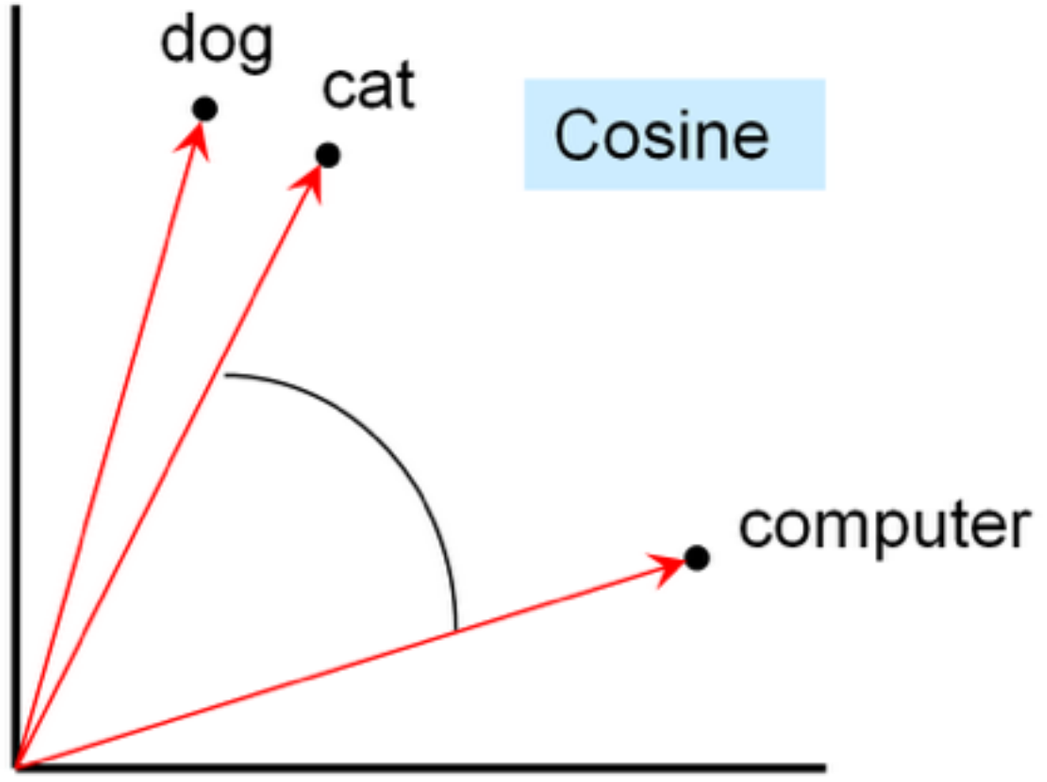
- Can be oversensitive to extreme values



Cosine Similarity

- Cosine distance: borrowed from information retrieval

$$\text{sim}_{\text{cosine}}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$



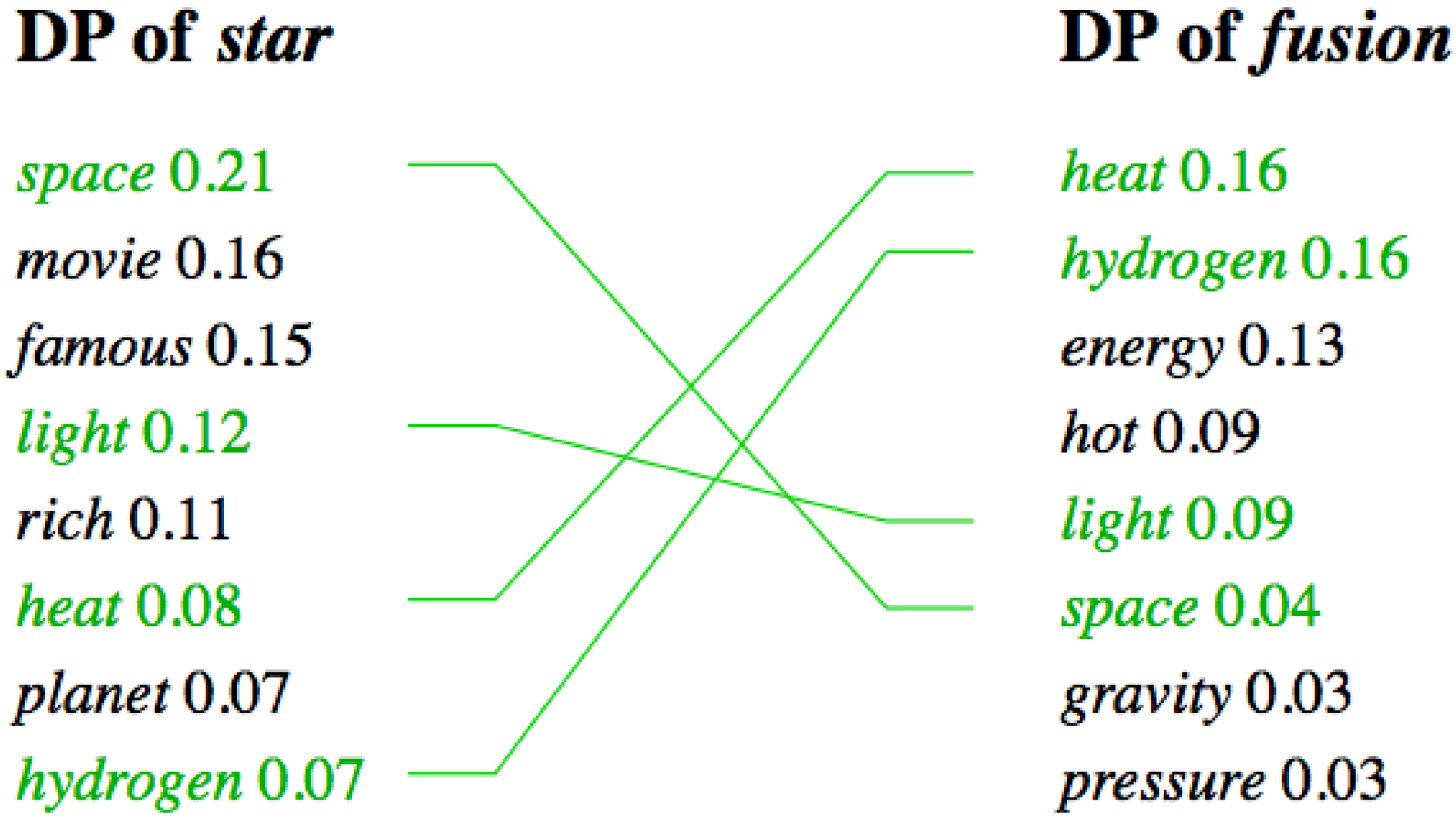
Distributional Approaches: Discussion

- No thesauri needed: data driven
- Can be applied to any pair of words
- Can be adapted to different domains

Distributional Profiles: Example

DP of *star*		**DP of *fusion***	
space	0.21	*heat*	0.16
movie	0.16	*hydrogen*	0.16
famous	0.15	*energy*	0.13
light	0.12	*hot*	0.09
rich	0.11	*light*	0.09
heat	0.08	*space*	0.04
planet	0.07	*gravity*	0.03
hydrogen	0.07	*pressure*	0.03

Distributional Profiles: Example



Problem?

DP of *star*

space 0.21
movie 0.16 ←
famous 0.15 ←
light 0.12
rich 0.11 ←
heat 0.08
planet 0.07
hydrogen 0.07

DP of *fusion*

heat 0.16
hydrogen 0.16
energy 0.13
hot 0.09
light 0.09
space 0.04
gravity 0.03
pressure 0.03

Using syntax to define a word's context

- Zellig Harris (1968)

  "The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities"

- Two words are similar if they have similar syntactic contexts

Syntactic context intuition

- Duty and responsibility have similar syntactic distribution:

Modified by adjectives	additional, administrative, assumed, collective, congressional, constitutional …
Objects of verbs	assert, assign, assume, attend to, avoid, become, breach..

Co-occurrence vectors based on syntactic dependencies

- Each dimension: a context word in one of R grammatical relations
  - Subject-of- "absorb"
- Instead of a vector of $|V|$ features, a vector of $R|V|$
- Example: counts for the word *cell*

	subj-of, absorb	subj-of, adapt	subj-of, behave	...	pobj-of, inside	pobj-of, into	...	nmod-of, abnormality	nmod-of, anemia	nmod-of, architecture	...	obj-of, attack	obj-of, call	obj-of, come from	obj-of, decorate	...	nmod, bacteria	nmod, body	nmod, bone marrow
cell	1	1	1		16	30		3	8	1		6	11	3	2		3	2	2

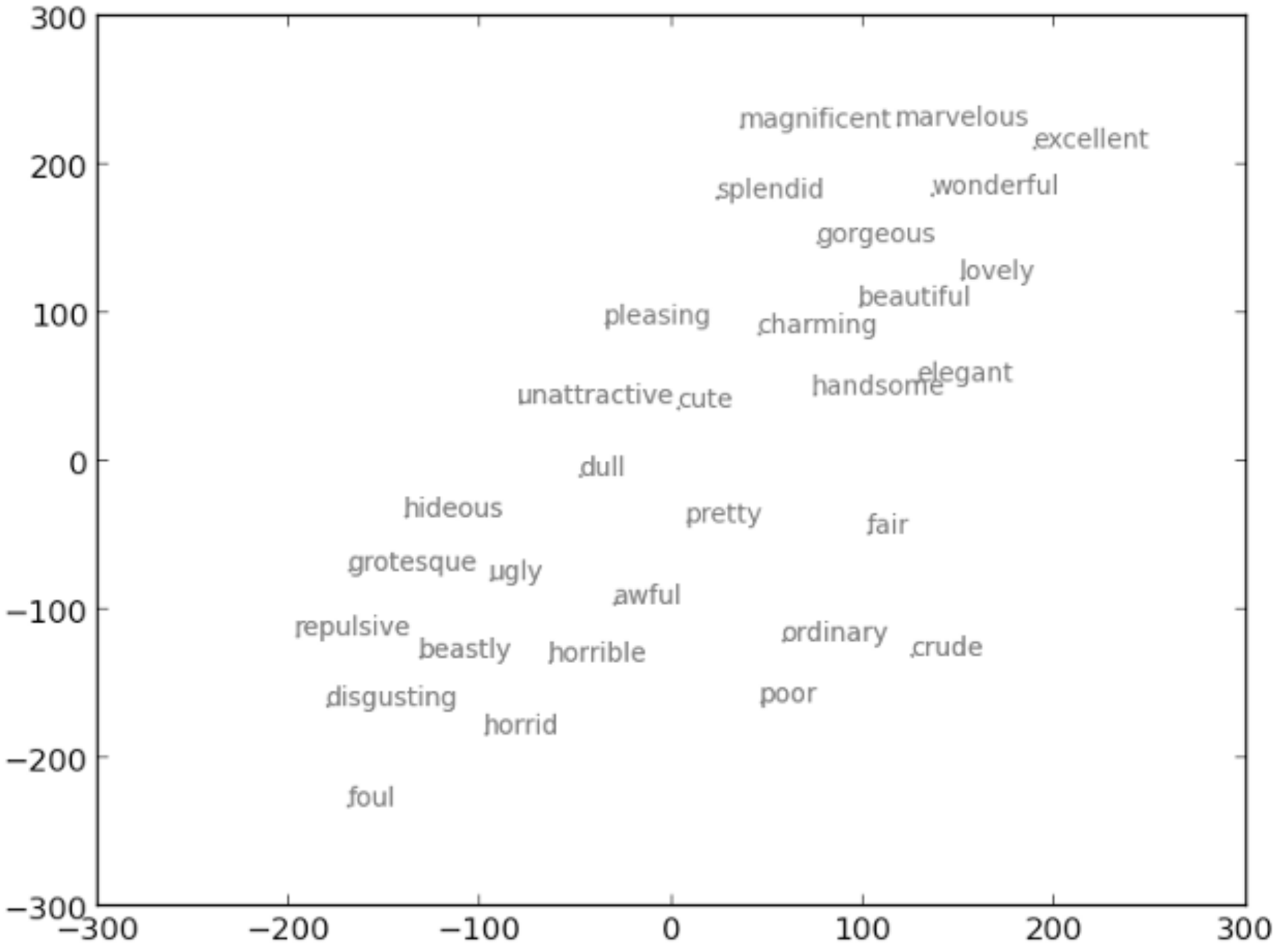
Dekang Lin, 1998 "Automatic Retrieval and Clustering of Similar Words"

What else can you do with word vectors/similarity?

Clustering

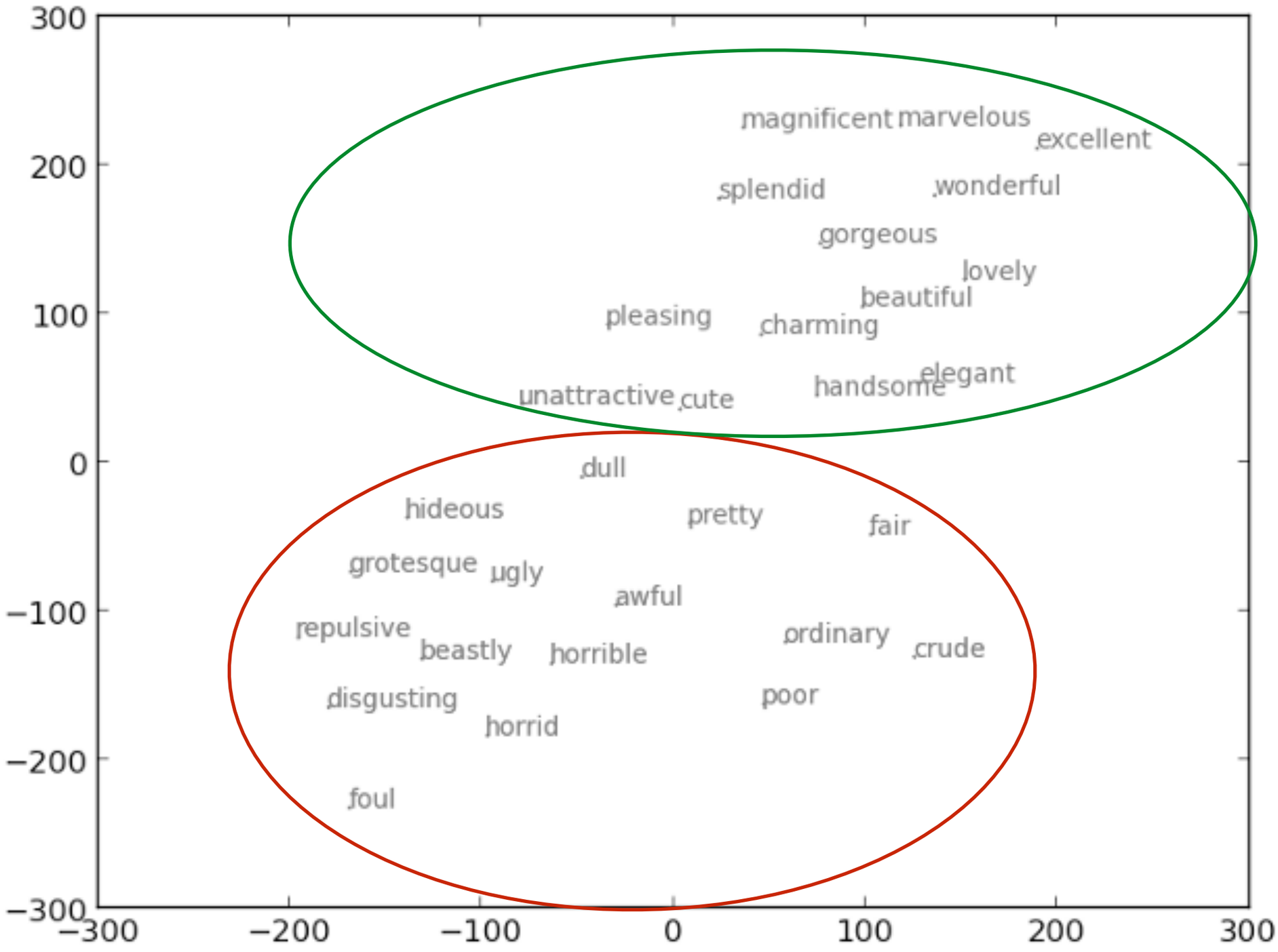
- Machine learning task of grouping similar data points together
  - *Hard* clustering: every data point goes in exactly 1 cluster

Clustering



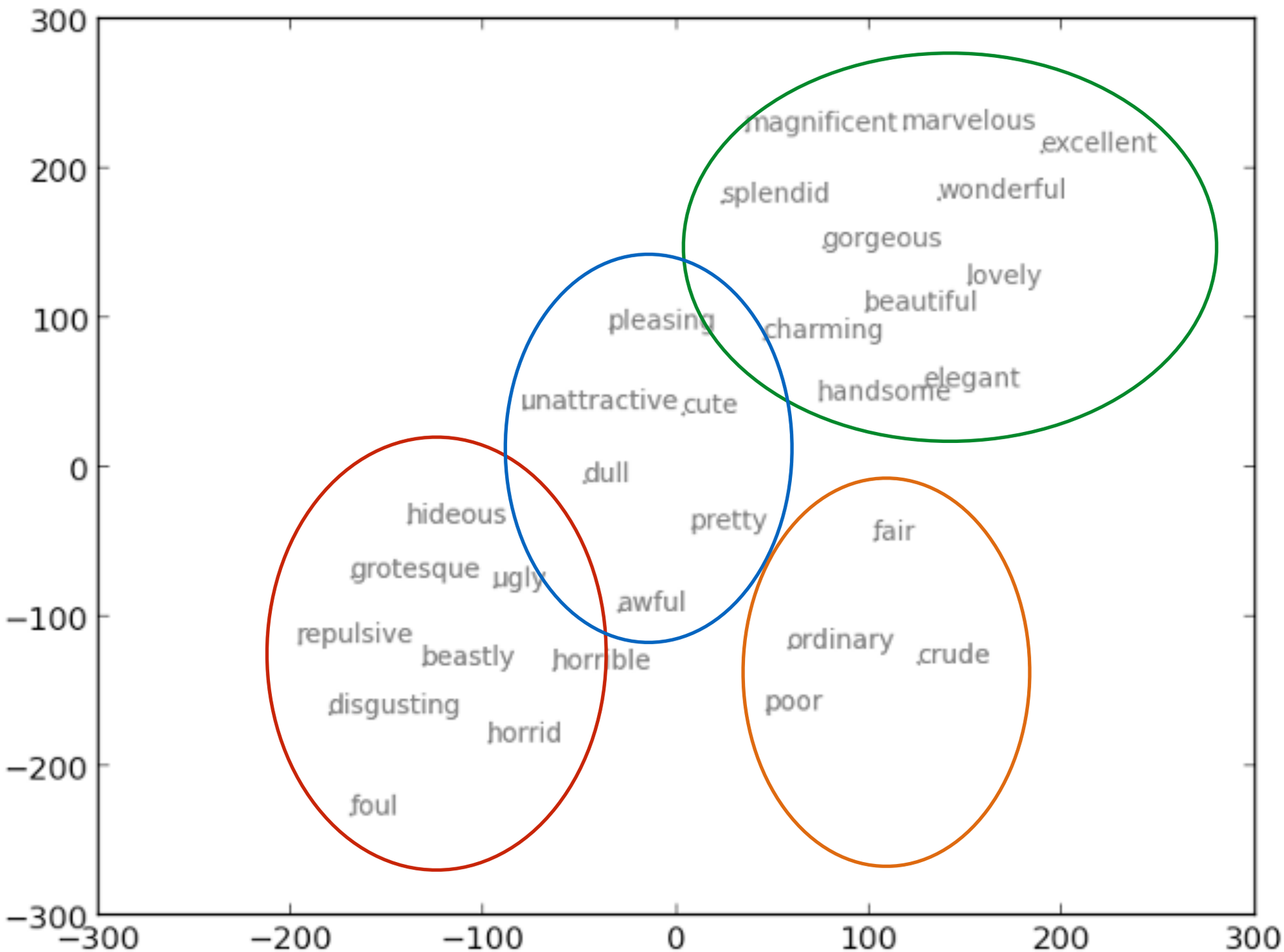
(Faruqui & Dyer 2014)

Clustering A



(Faruqui & Dyer 2014)

Clustering B

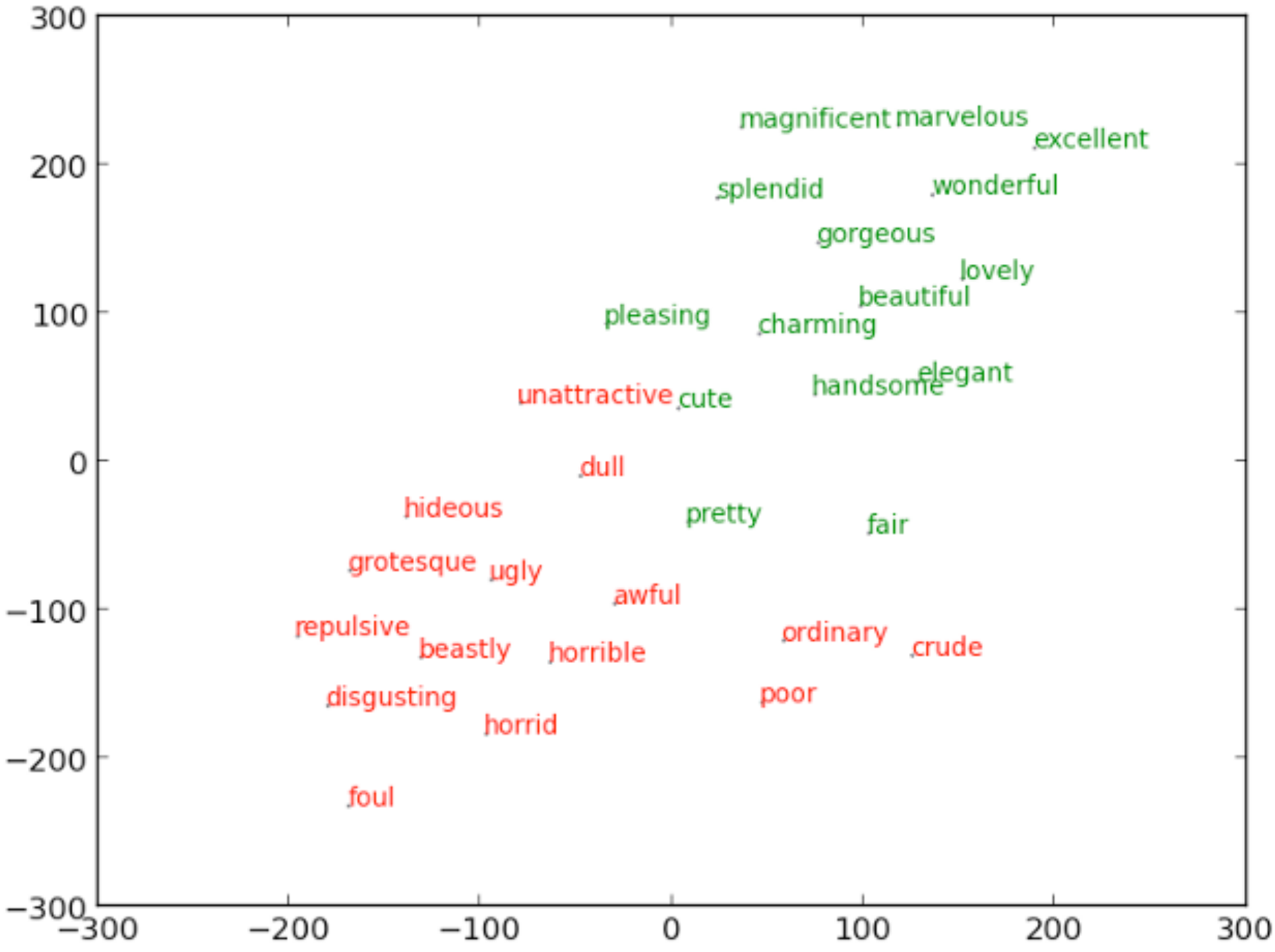


(Faruqui & Dyer 2014)

Clustering

- Machine learning task of grouping similar data points together
  - *Hard* clustering: every data point goes in exactly 1 cluster
  - How many clusters to predict? Some algorithms have *K* as a hyperparameter, others infer it.
  - Which clustering is better? May depend on the beholder/application.

Clustering for Sentiment



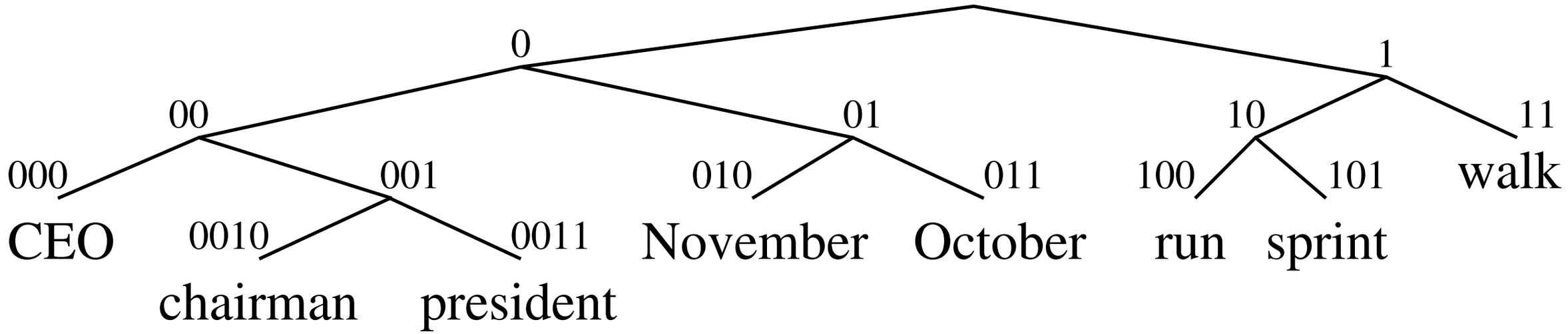
(Faruqui & Dyer 2014)

Brown Clustering

- Algorithm that produces **hierarchical** clusters based on word context vectors

- Words in similar parts of hierarchy occur in similar contexts

Chairman is 0010, “months” = 01, and verbs = 1



Brown clusters created from Twitter data:
http://www.cs.cmu.edu/~ark/TweetNLP/cluster_viewer.html

Word Embeddings

- **Dense** word vectors: e.g., 100 or 200 dimensions (rather than the size of the vocabulary)

- Can be produced by **dimensionality reduction** of the full word-context matrix

- Or with neural network algorithms such as **word2vec** (Mikolov et al. 2013)

DIMENSIONALITY REDUCTION

Slides based on presentation by
Christopher Potts

Why dimensionality reduction?

- So far, we've defined word representations as rows in **F**, a m x n matrix
  - m = vocab size
  - n = number of context dimensions / features

- Problems: n is very large, F is very sparse

- Solution: find a low rank approximation of **F**
  - Matrix of size m x d where d << n

Methods

- Latent Semantic Analysis
- Also:
  - Principal component analysis
  - Probabilistic LSA
  - Latent Dirichlet Allocation
  - Word2vec
  - …

Latent Semantic Analysis

- Based on **Singular Value Decomposition**

For any matrix of real numbers A of dimension $(m \times n)$ there exists a factorization into matrices T, S, D such that

$$A_{m\times n} = T_{m\times m} S_{m\times m} D^T_{n\times m}$$

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot & & \\ & \cdot & \\ & & \cdot \end{pmatrix} \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}^T$$

$$A_{3\times 4} \quad = \quad T_{3\times 3} \qquad S_{3\times 3} \qquad D^T_{4\times 3}$$

LSA illustrated: SVD + select top k dimensions

	d1	d2	d3	d4	d5	d6
gnarly	1	0	1	0	0	0
wicked	0	1	0	1	0	0
awesome	1	1	1	1	0	0
lame	0	0	0	0	1	1
terrible	0	0	0	0	0	1

⇓⇑

Distance from *gnarly*
1. gnarly
2. awesome
3. terrible
4. wicked
5. lame

T(erm)				
gnarly	0.41	0.00	0.71	0.00	-0.58
wicked	0.41	0.00	-0.71	0.00	-0.58
awesome	0.82	-0.00	-0.00	-0.00	0.58
lame	0.00	0.85	0.00	-0.53	0.00
terrible	0.00	0.53	0.00	0.85	0.00

×

S(ingular values)					
1	2.45	0.00	0.00	0.00	0.00
2	0.00	1.62	0.00	0.00	0.00
3	0.00	0.00	1.41	0.00	0.00
4	0.00	0.00	0.00	0.62	0.00
5	0.00	0.00	0.00	0.00	-0.00

×

D(ocument)					
d1	0.50	-0.00	0.50	0.00	-0.71
d2	0.50	0.00	-0.50	0.00	0.00
d3	0.50	-0.00	0.50	0.00	0.71
d4	0.50	-0.00	-0.50	-0.00	0.00
d5	-0.00	0.53	0.00	-0.85	0.00
d6	0.00	0.85	0.00	0.53	0.00

^{T}

gnarly	0.41	0.00
wicked	0.41	0.00
awesome	0.82	-0.00
lame	0.00	0.85
terrible	0.00	0.53

×

2.45	0.00
0.00	1.62

=

gnarly	1.00	0.00
wicked	1.00	0.00
awesome	2.00	0.00
lame	0.00	1.38
terrible	0.00	0.85

Distance from *gnarly*
1. gnarly
2. wicked
3. awesome
4. terrible
5. lame

Word embeddings based on neural language models

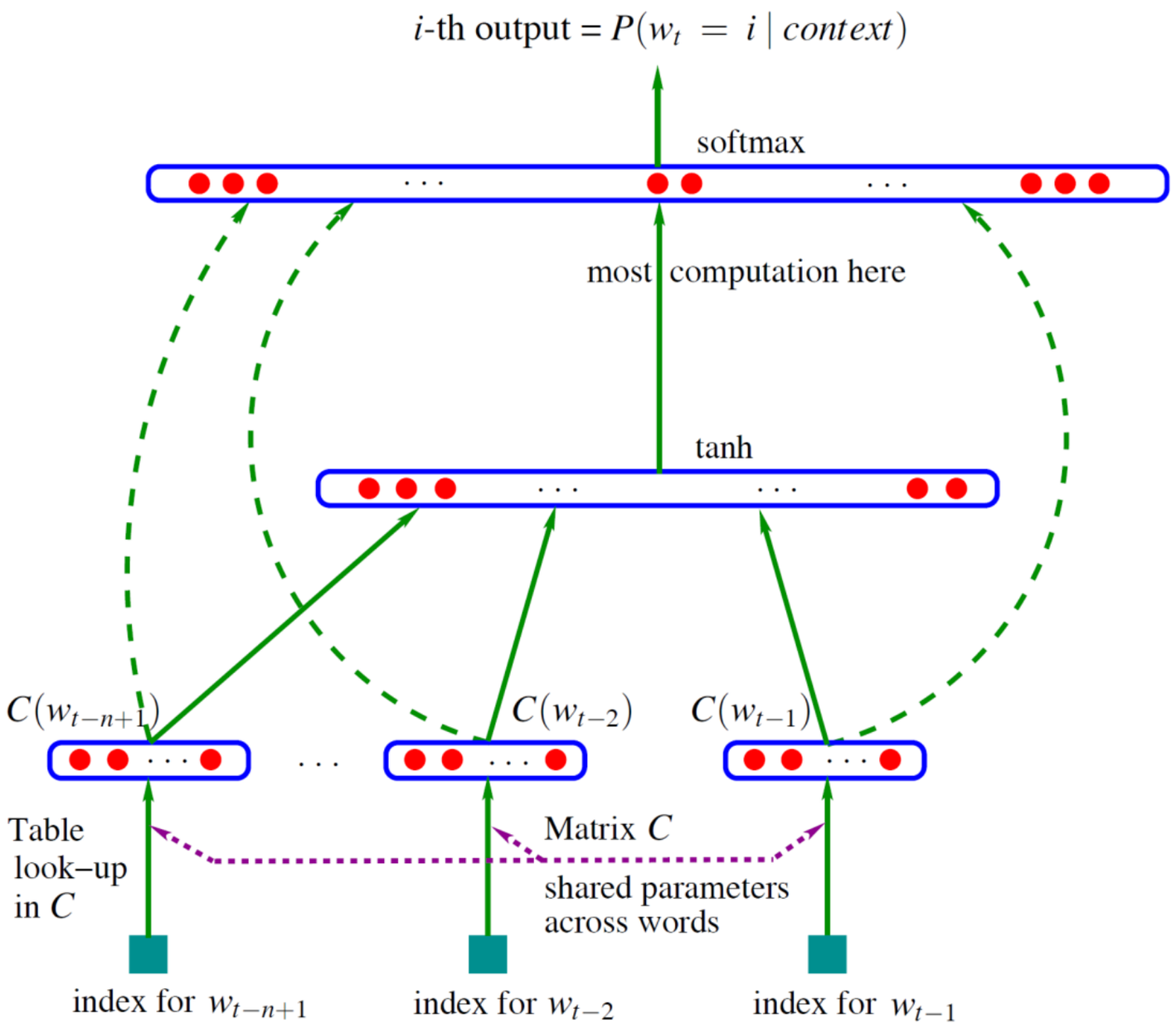
- So far: Distributional vector representations constructed based on **counts** (+ dimensionality reduction)

- Recent finding: Neural networks trained to **predict neighboring words** (i.e., language models) learn useful low-dimensional word vectors

  - Dimensionality reduction is built into the NN learning objective

  - Once the neural LM is trained on massive data, the word embeddings can be reused for other tasks

Word vectors as a byproduct of language modeling

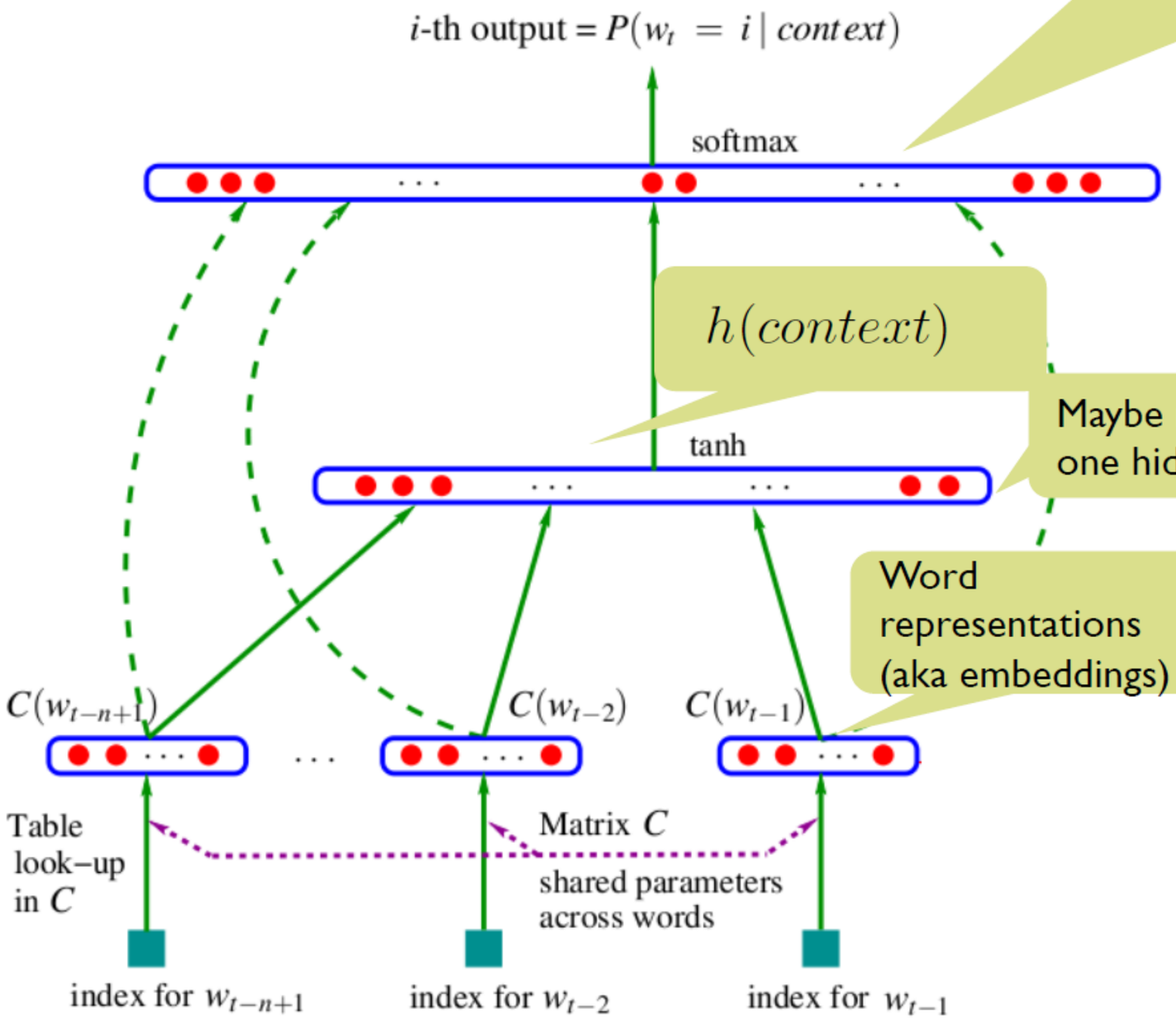


A neural probabilistic Language Model. Bengio et al. JMLR 2003

Language modeling task: context of w_t is

$w_{t-1}, w_{t-2}, \ldots, w_{t-n+1}$

$$P(w_t = i|context) = \frac{\exp(\hat{C}(i) \cdot h)}{\sum_{j=1}^{V} exp(\hat{C}(j) \cdot h)}$$



	1	2		d
cat	2.059	-1.134	...	2.004
dog	2.011	-1.005	...	0.135
...	...	...	...	...
January	-3.193	0.145	...	0.001
February	-3.016	0.196	...	0.025
...	...	...	...	...

Using neural word representations in NLP

- word representations from neural LMs
  - aka distributed word representations
  - aka word embeddings

- How would you use these word vectors?
- Turian et al. [2010]
  - word representations as features consistently improve performance of
    - Named-Entity Recognition
    - Text chunking tasks

Word2vec claims

Useful representations for NLP applications

Can discover relations between words using vector arithmetic

king – male + female = queen

Paper+tool received lots of attention even outside the NLP research community

try it out at “word2vec playground”:

http://deeplearner.fz-qqq.net/

Two Kinds of Distributional Contexts

1. Documents as bags-of-words

   - Similar documents contain similar words; similar words appear in similar documents

2. Words in terms of neighboring words

   - "You shall know a word by the company it keeps!" (Firth, 1957)

   - Similar words occur near similar sets of other words (e.g., in a 5-word window)

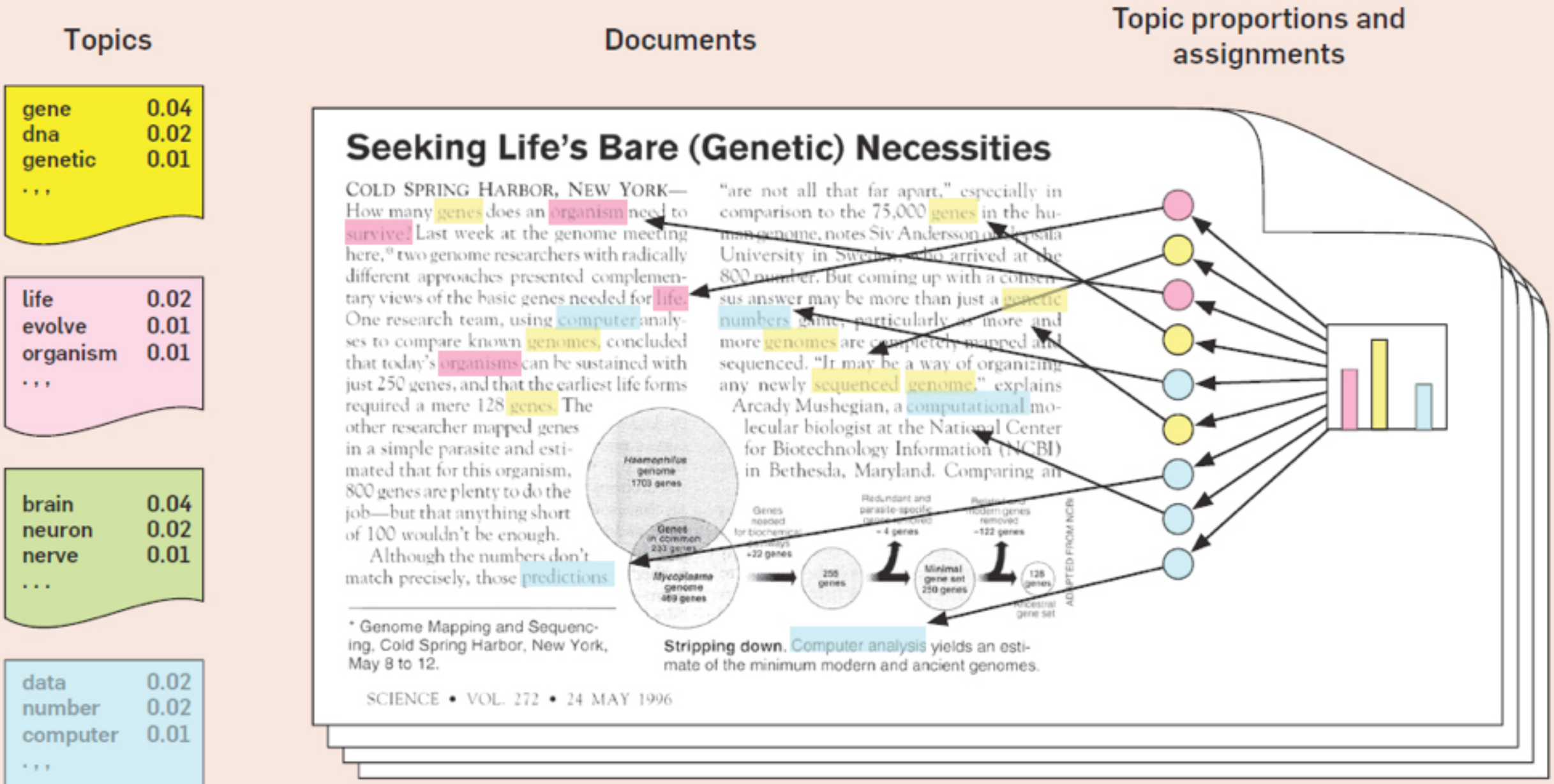
Document-Word Models

- Features in the word vector can be word context counts or PMI scores
- Also, features can be the documents in which this word occurs
  - Document occurrence features useful for **topical/ thematic** similarity

Topic Models

- Latent Dirichlet Allocation (LDA) and variants are known as **topic models**
  - Learned on a large document collection (unsupervised)
  - Latent probabilistic **clustering** of words that tend to occur in the same document. Each **topic** cluster is a distribution over words.
  - Generative model: Each document is a sparse mixture of topics. Each word in the document is chosen by sampling a topic from the document-specific topic distribution, then sampling a word from that topic.
  - Learn with EM or other techniques (e.g., Gibbs sampling)

Topic Models



http://cacm.acm.org/magazines/2012/4/147361-probabilistic-topic-models/fulltext

Visualizing Topics

https://dhs.stanford.edu/algorithmic-literacy/using-word-clouds-for-topic-modeling-results/

Summary

- Given a large corpus, the meanings of words can be approximated in terms of words occurring nearby: **distributional context**. Each word represented as a **vector** of integer or real values.
  - Different ways to choose context, e.g. context windows
  - Different ways to count cooccurrence, e.g. (positive) **pointwise mutual information**
  - Vectors can be **sparse** (1 dimension for every context) or **dense** (reduced dimensionality, e.g. with **Brown clustering** or **word2vec**)
- This facilities measuring **similarity** between words—useful for many purposes!
  - Different similarity measures, e.g. **cosine** (= normalized dot product)
  - Evaluations: human relatedness judgments; extrinsic tasks