IN2110 SPRING 2022

SPRÅKTEKNOLOGISKE METODER

Erik Velldal & Jan Tore Lønning

Plan – weeks 2 – 5

Lectures 2-5

- How to represent (language) data in a mathematical model.
- Vector space models.
- Representing
  - Documents (today)
  - Words (week 5)
- Vector-based machine learning
  - Classification (week 3)
  - Clustering (week 4)

Today

- Lexical semantics
- Word-context matrices
- Word embeddings with dense vectors
- As time permits: Recap: *k*-means clustering

Disclaimer

- I am only a substitute teacher for Erik Velldal
- The slides will be a mixture
  - Erik's slides from last year
  - My slides from IN3050 and IN4080
  - Some new slides (like this one)

Limitations of BoW models

BoW representations of text

- So far we've been assuming BoW features for representing documents.
- Often also used for representing other units of texts, like sentences.
- Many sentence-classification tasks in NLP.

BoW representations of text

- So far we've been assuming BoW features for representing documents.
- Often also used for representing other units of texts, like sentences.
- Many sentence-classification tasks in NLP.
- Example: polarity classification (part of sentiment analysis).

I was impressed, this was not bad!

$$\Rightarrow$$

{was, was, !, not, I, impressed, bad, this}

- What is missing with a BoW representation?

Dealing with compositionality

I was impressed, this was not bad!

$\neq$

I was not impressed, this was bad!

- Will have the same BoW representation! :(

Dealing with compositionality

I was impressed, this was not bad!
$\neq$
I was not impressed, this was bad!

- Will have the same BoW representation! :(
- A simplistic but much-used approximation to capture ordering constraints: n-grams (typically bigrams and trigrams).
- Ordered sub-sequences of n words.

{was, was, !, not, I, impressed, bad, this}

vs.

{'I was', 'was impressed' ... 'was not', 'not bad', 'bad, !'}

No information sharing

- No information sharing between features.
- All features are equally distinct.
- The pizza was great
- The margeritha was awesome
- The dog was sick
- Would be nice if our BoW representations knew that *pizza* and *margeritha* are similar to each other (but not to *dog*).
- We discussed one possible approach in the previous lecture. . . What?
- Will return to this issue later today. . .

Lexical semantics

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 *πέπερι*); 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* *big-large*		

Relations between senses

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

Relations between senses

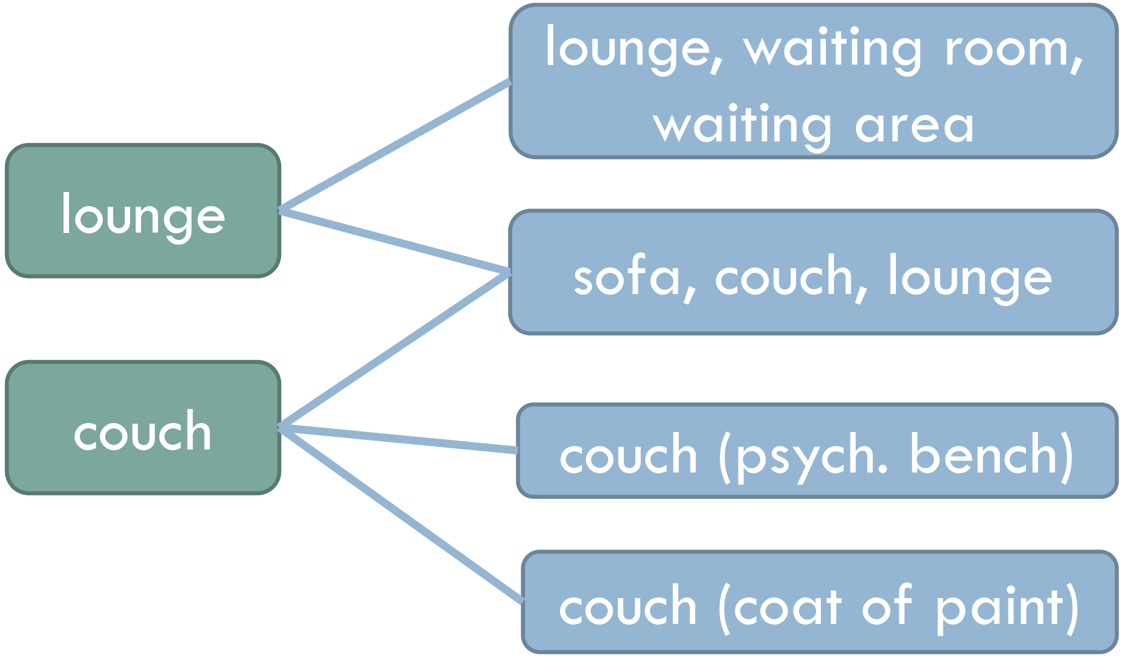
Term	Definition	Examples		
Synonymy	Have the same meaning in all(?)/some(?) contexts	*sofa-couch, bus-coach* *big-large*		
Antonym	Opposites with respect to a feature of meaning	*true-false, strong-weak, up-down*		
Hyponym-hypernym	The <hyponym> is a type-of the <hypernym>	*rose →flower, cow →animal,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

Sentiment

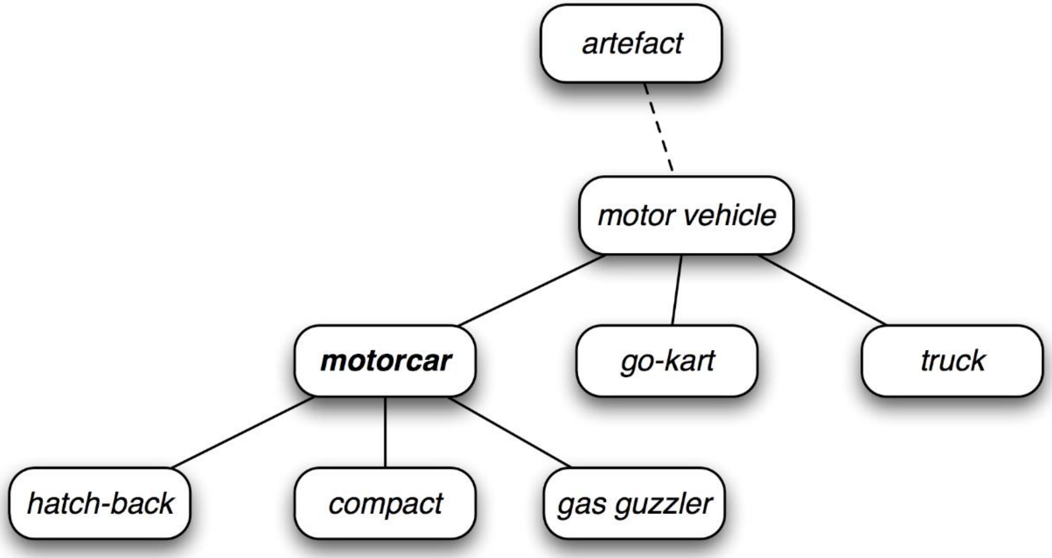
- Sometimes interested in more narrow kinds of word similarity.
- E.g. affective meanings or connotations:
- Aspects of a word's meaning related to a writer/reader's emotions, sentiment, opinions, or evaluations.
- Positive/negative connotation: *happy / sad*
- Positive/negative evaluation: *great / terrible*
- Important for NLP tasks like sentiment analysis, stance detection, argumentation mining, hate-speech detection, etc.

Resources for lexical semantics: WordNet

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



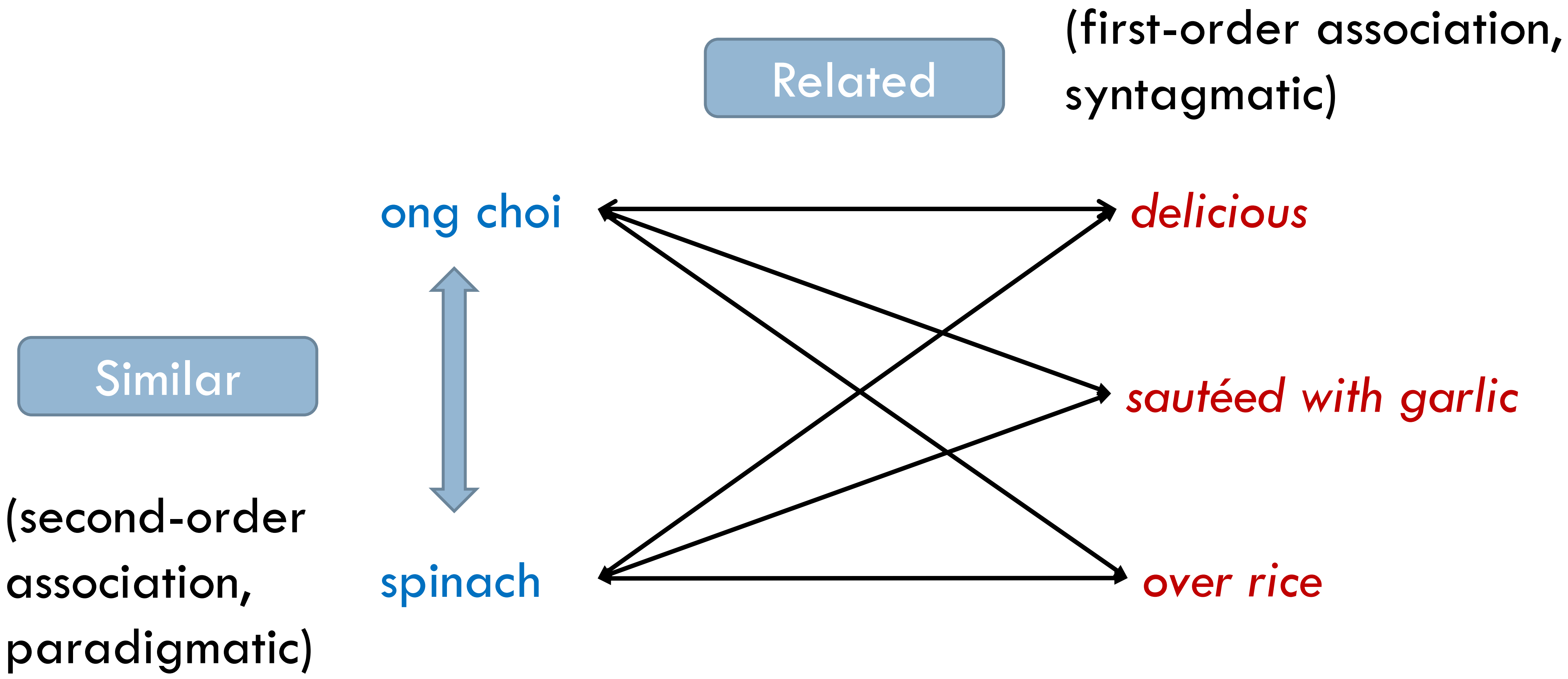
- Relations between the synsets



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)
ong choi
delicious
Similar
sautéed with garlic
(second-order association, paradigmatic)
spinach
over rice

The distributional hypothesis

- Words that occur in similar contexts have similar meanings
- Comparing meanings reduces to comparing contexts

AKA the contextual theory of meaning

- *Meaning is use.* (Wittgenstein, 1953)
- *You shall know a word by the company it keeps.* (Firth, 1957)
- *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.* (Harris, 1968)

The distributional hypothesis

- Words that occur in similar contexts have similar meanings
- Comparing meanings reduces to comparing contexts

AKA the contextual theory of meaning

- *Meaning is use.* (Wittgenstein, 1953)
- *You shall know a word by the company it keeps.* (Firth, 1957)
- *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.* (Harris, 1968)

Distributional semantics in practice

- Record contexts of words across a large corpus.
- Each word is represented by a set of features.
- Each feature records some property of the observed contexts.
- Words that are found to have similar features are expected to also have similar meaning.
- Features can be represented in a vector space model, with similarity modeled as geometrical distance.

- Some design decisions;
  - How do we define '*context*'?
  - How do we define a '*word*'?

Word-context Matrices

Remember? Term-document matrix

	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

- Example of a co-occurrence matrix
- More specifically,
  a $m \times n$ term-document matrix
  - m terms, n documents
- Count the number of occurrences of the terms in each document

- Each column represent a document
- Each row represents a term (word, feature)
- With 4 key words each document is represented as a 4-d vector
- (We could use any set of key words)

Vector repr. of words 1: A vector of documents

	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

battle is "the kind of word that occurs in Julius Caesar and Henry V"

fool is "the kind of word that occurs in comedies, especially Twelfth Night"

Vector repr. of words 2: Word-context matrix

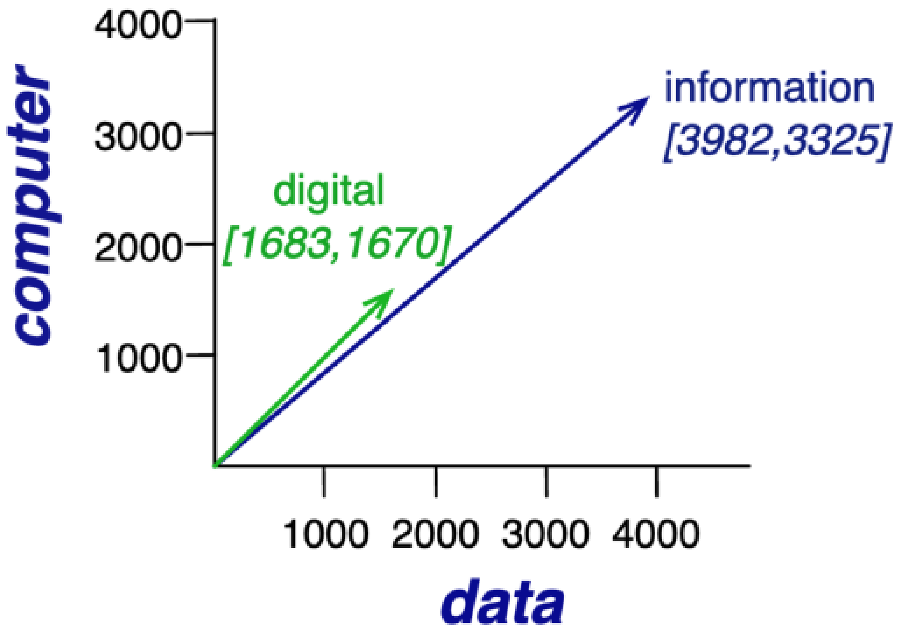
- Two **words** are similar in meaning if their context vectors are similar

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	

Term–term matrix

	aardvark	...	computer	data	result	pie	sugar
cherry	0	...	2	8	9	442	25
strawberry	0	...	0	0	1	60	19
digital	0	...	1670	1683	85	5	4
information	0	...	3325	3982	378	5	13



Word-context, or term-term, matrix

Document-term matrix

- Objects: a set of documents, D
- Features: a set of terms,
  - $T = \{t_1, t_2, \dots, t_n\}$
- Each document d is identified with a vector
  - $(v_1, v_2, \dots, v_n)$
  - 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,
  - $C = \{c_1, c_2, \dots, c_n\}$
- 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
  - $(v_1, v_2, \dots, v_n)$
  - where v_i is calculated from the frequency of c_i in all the contexts of w in T

Similarities and differences

Comments

- C=V, or C is smaller set of the most frequent terms
  - To avoid too 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

What is a context?

- For BoW: Document, sentence, or window ($\pm k$ words left/right).
- Can also use n-grams or grammatical relations.
- *Broader* context tend to capture *relatedness*.
- *Narrower* context tend to capture *similarity*.
- Like for document vectors, the raw counts are typically weighted;
- using e.g TF-IDF or pointwise mutual information, or some other association measure.

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.

Word embeddings with dense vectors

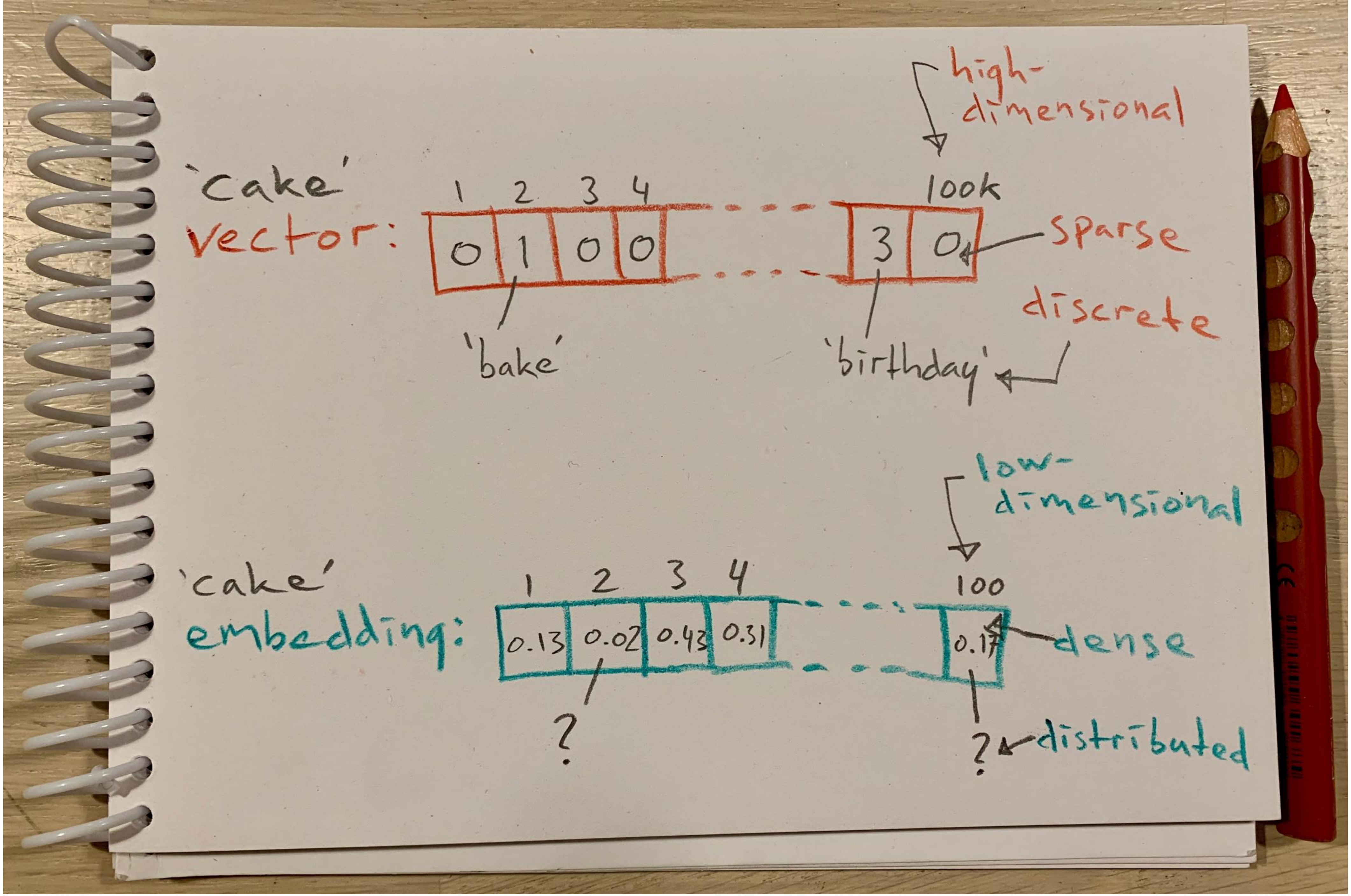
Dense vectors

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.



high-dimensional
'cake'
1 2 3 4
100k
vector:
0 1 0 0
3 0
sparse
discrete
'bake'
'birthday'
low-dimensional
'cake'
1 2 3 4
100
embedding:
0.13 0.02 0.43 0.31
0.17
dense
?
?
distributed

Word embeddings

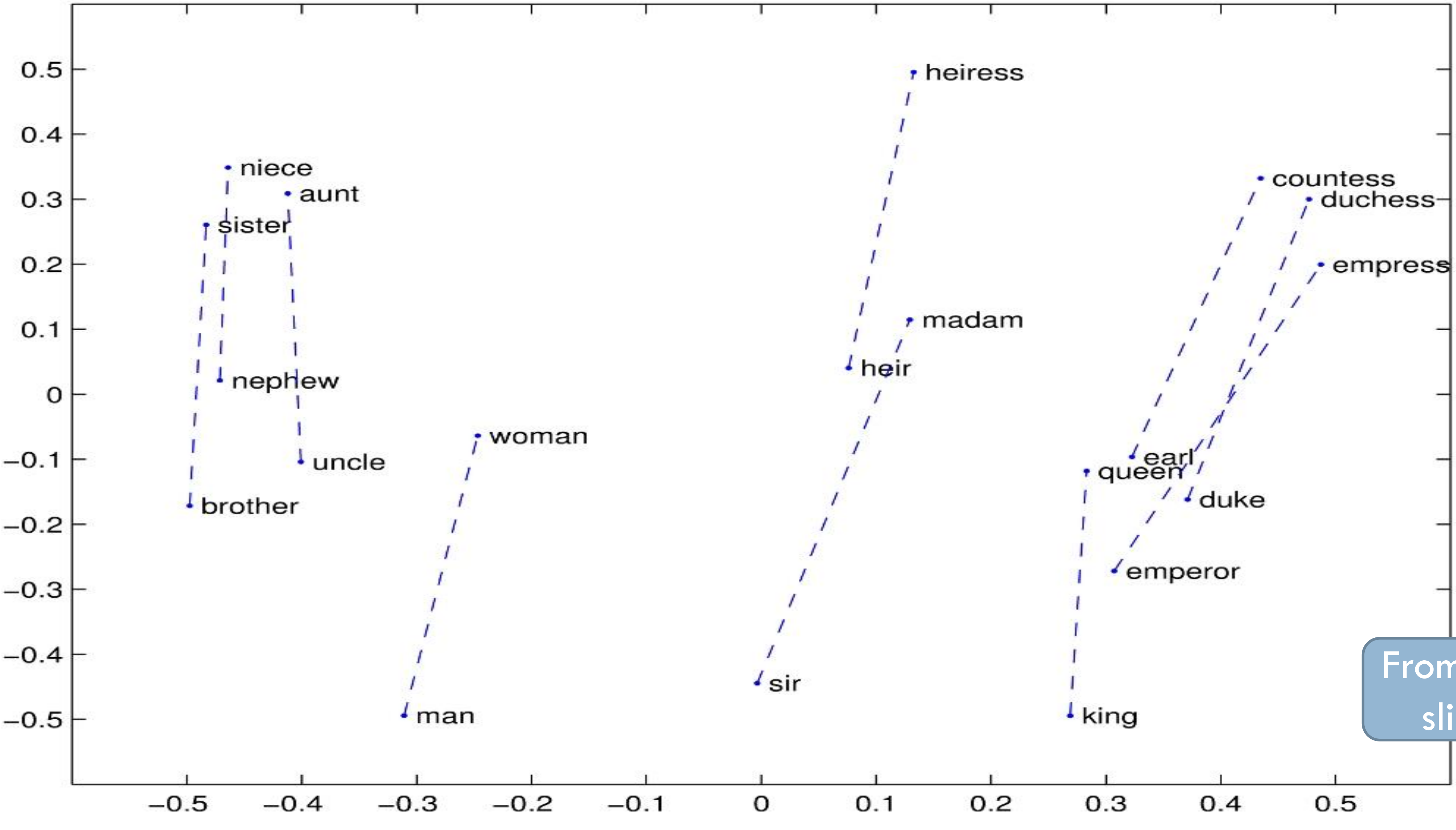
- In current LT: Each word is represented as a vector of reals
- Words are more or less similar
- A word can be similar to one word in some dimensions and other words in other dimensions

Dimensions

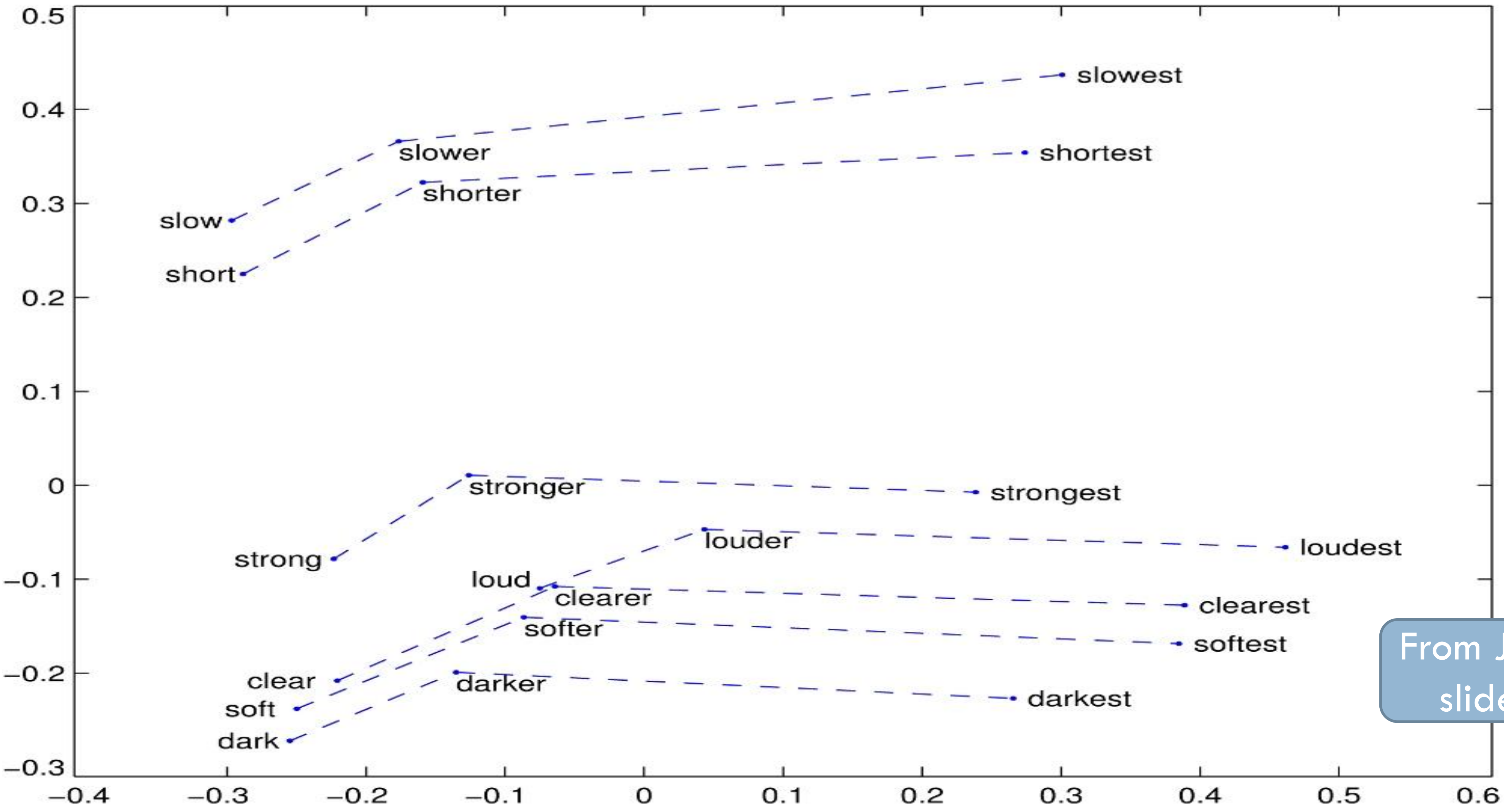
Word vectors				
dog	-0.4	0.37	0.02	-0.34
cat	-0.15	-0.02	-0.23	-0.23
lion	0.19	-0.4	0.35	-0.48
tiger	-0.08	0.31	0.56	0.07
elephant	-0.04	-0.09	0.11	-0.06
cheetah	0.27	-0.28	-0.2	-0.43
monkey	-0.02	-0.67	-0.21	-0.48
rabbit	-0.04	-0.3	-0.18	-0.47
mouse	0.09	-0.46	-0.35	-0.24
rat	0.21	-0.48	-0.56	-0.37

animal
domesticated
pet
fluffy

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



From J&M slides

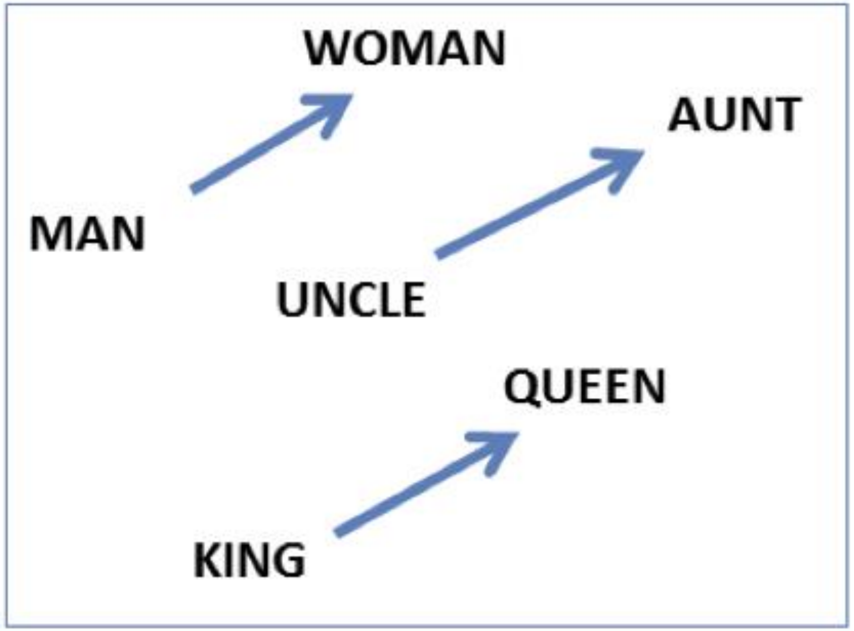


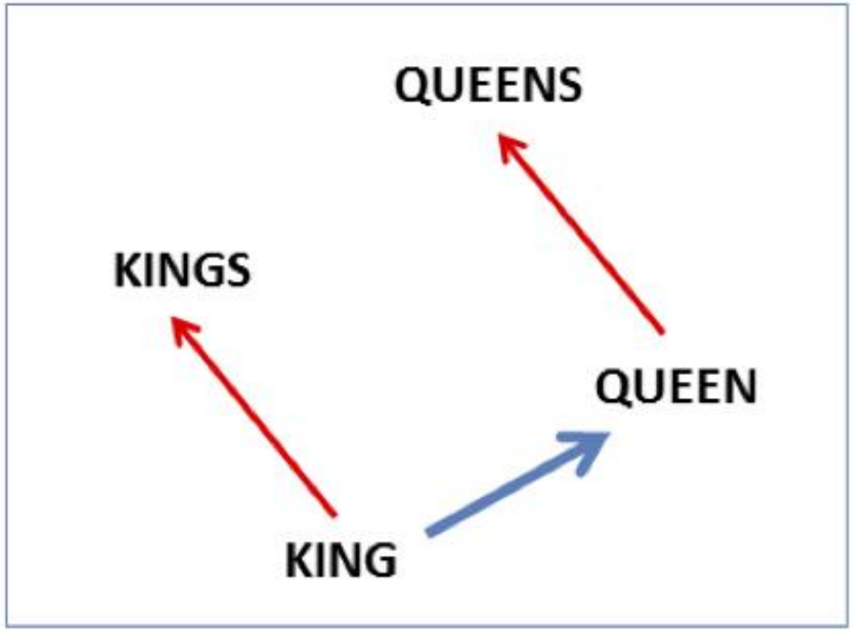
From J&M slides

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

Demo

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

Diachronic word embeddings and language change

- Another example of use-case or evaluation task:
- Embeddings can help study word history!
- Train embeddings on corpora from different time periods.
- Trace how meaning changes over time by comparing changes in vector distances and neighbor relations.

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 word embeddings come from?

- Basic idea:
  - A predication task
  - Train a machine learner to perform the task
  - Use some parts (weights) of the trained ML model as representations
- Simplest form: a bigram language model:
  - For a given word w_{i-1}, try to predict the next word w_i
  - i.e. try to estimate $P(w_i | w_{i-1})$
- No need for hand-labeled data; use running text
  - "Self-supervision"

Model

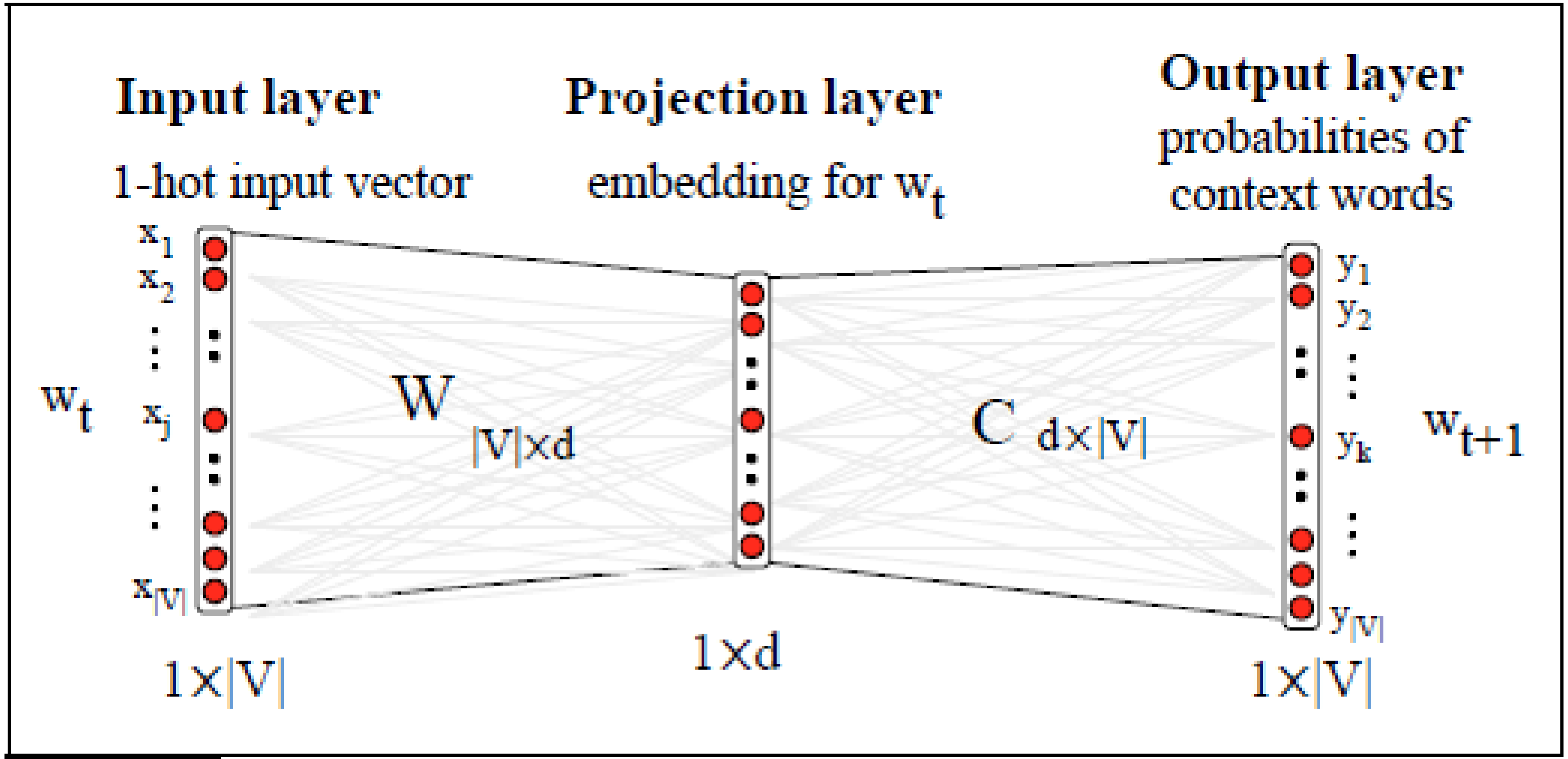


Figure 16.5 The skip-gram model viewed as a network (Mikolov et al. 2013, Mikolov et al. 2013a).

From J&M 3.ed. 2018 Ch. 16

Embeddings from a language model

- Decide the dim, *d*, of the embedding typically 50-300
- The input word, w_t, and the output word, w_{t+1}, are represented as "one-hot-vectors"
- Fully connected layers
- Train the network to predict the next word
- Use the weights $W_{|V| \times d}$

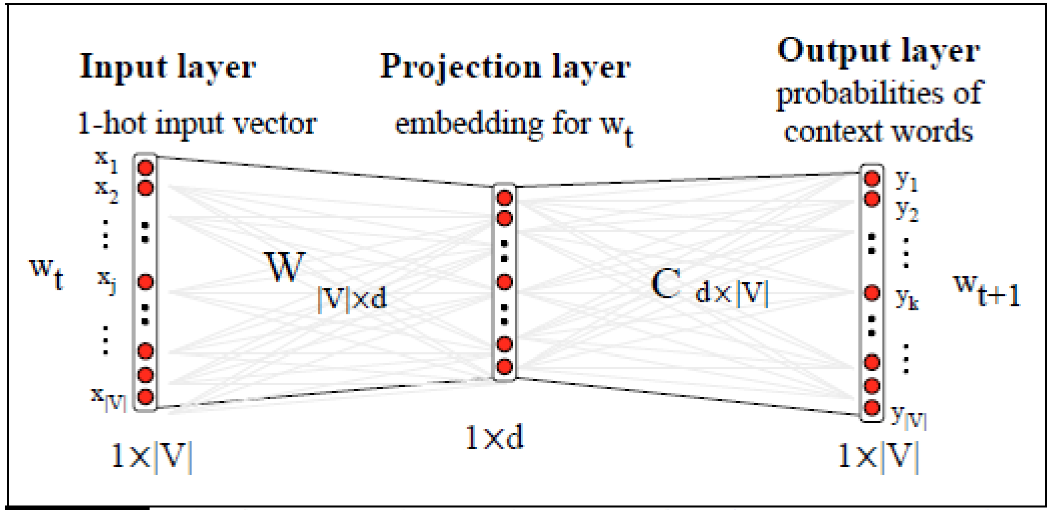


Figure 16.5 The skip-gram model viewed as a network (Mikolov et al. 2013, Mikolov et al. 2013a).

(You are not supposed to be able to explain all of this)

Embeddings from this

- Idea: Use the weight matrix $W_{|V|\times d}$ as embeddings, i.e.:
- Represent word j by $(w_{j,1}, w_{j,2}, \dots, w_{j,d})$ = the weights that sends this word to the hidden layer
- Why? since similar words will predict more or less the same words, they will get similar embeddings

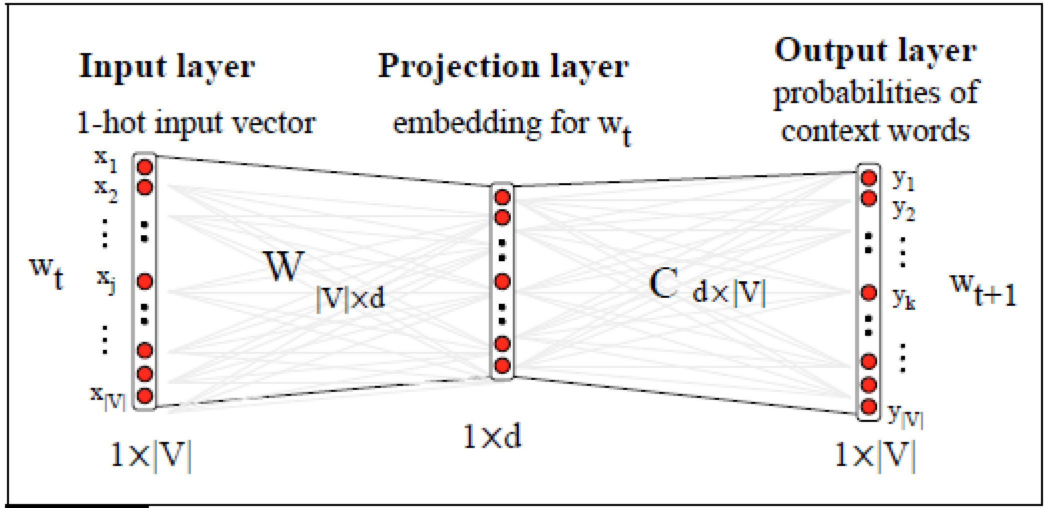
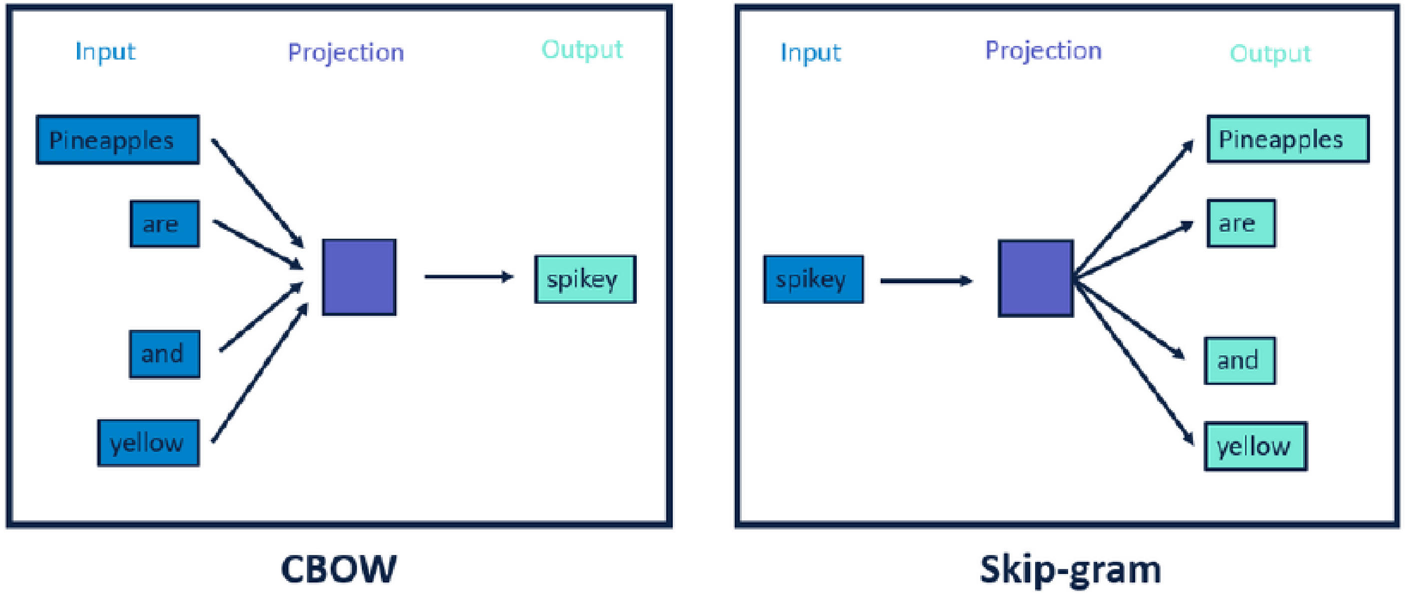


Figure 16.5 The skip-gram model viewed as a network (Mikolov et al. 2013, Mikolov et al. 2013a).

(You are not supposed to be able to explain all of this)

word2vec



- Example of approach that aim to estimate word embeddings.
- Implements two algorithms: CBOW and Skip-Gram
- Instead of counting how often each word w occurs near *pineapple*, train a classifier to predict: Is w likely to show up near *pineapple*?
- A word s seen near *pineapple* acts as gold positive examples.
- Randomly sample other words in the lexicon as negative examples.
- Use the learned weights as the embeddings.

Evaluation of embeddings

- 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

Evaluation of embeddings on analogy tests

- Google analogy test set examples

	Relation type	# Questions	Word pair 1		Word pair 2	
Semantic	Common capital city	506	Athens	Greece	Baghdad	Iraq
	All capital cities	4,524	Abuja	Nigeria	Accra	Ghana
	Currency	866	Algeria	dinar	Angola	kwanza
	City-in-state	2,467	Chicago	Illinois	Houston	Texas
	Man–woman	506	boy	girl	brother	sister
Syntactic	Adjective-to-adverb	992	amazing	amazingly	apparent	apparently
	Opposite	812	acceptable	unacceptable	aware	unaware
	Comparative	1,332	bad	worse	big	bigger
	Superlative	1,122	bad	worst	big	biggest
	Present participle	1,056	code	coding	dance	dancing
	Nationality adjective	1,599	Albania	Albanian	Argentina	Argentinean
	Past tense	1,560	dancing	danced	decreasing	decreased
	Plural nouns	1,332	banana	bananas	bird	birds
	Plural verbs	870	decrease	decreases	describe	describes

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

Challenges with word vectors

Conflation of word senses

- Single vector for each word.
- Problem for polysemy and homonymy.

Bias

- Human biases in the training corpus will be reflected word vectors.
- E.g. associations between gender-terms and stereotypical occupations.

Antonyms

- Senses that are opposites.
- *long/short*, *fast/slow*, *dark/light*, *up/down*, *rise/fall*, *hot/cold*, *in/out*
- Difficult to distributionally distinguish from synonyms.

Back to representing documents and sentences

- Word embeddings provide a good representation of words.
- Let us model word similarity.
- What about sentence similarity or document similarity?
- From word embeddings to text embeddings.
- Consider sentiment analysis as an example:
- We want to train and apply a sentence-level classifier to predict positive / negative polarity of sentences.

Embeddings to the rescue

- Can represent documents by concatenating or averaging word embeddings.
- For a sentence $s = w_1, \ldots w_n$ with word embeddings $x_1, \ldots x_n$, we can compute the sentence embedding as $\mathrm{vec}(s) = \frac{1}{n}\sum_{i=1}^{n} x_i$
- Can then use $\mathrm{vec}(s)$ as input to a sentence classifier.
- Benefit; similarity relations between features.
- Allows us to 'pre-train' word representations on unlabeled data, before training e.g. a sentence-classifier on labeled data.

Semantic vectors; summary

- Both traditional word vectors and more modern embeddings allow us model word similarity using raw text; no labeled data needed.
- Semantic similarity modeled as distributional similarity, which is in turn computed as vector distance.
- In modern NLP, embeddings have a very central role!
- Standard input representation to neural (deep learning) classifier.

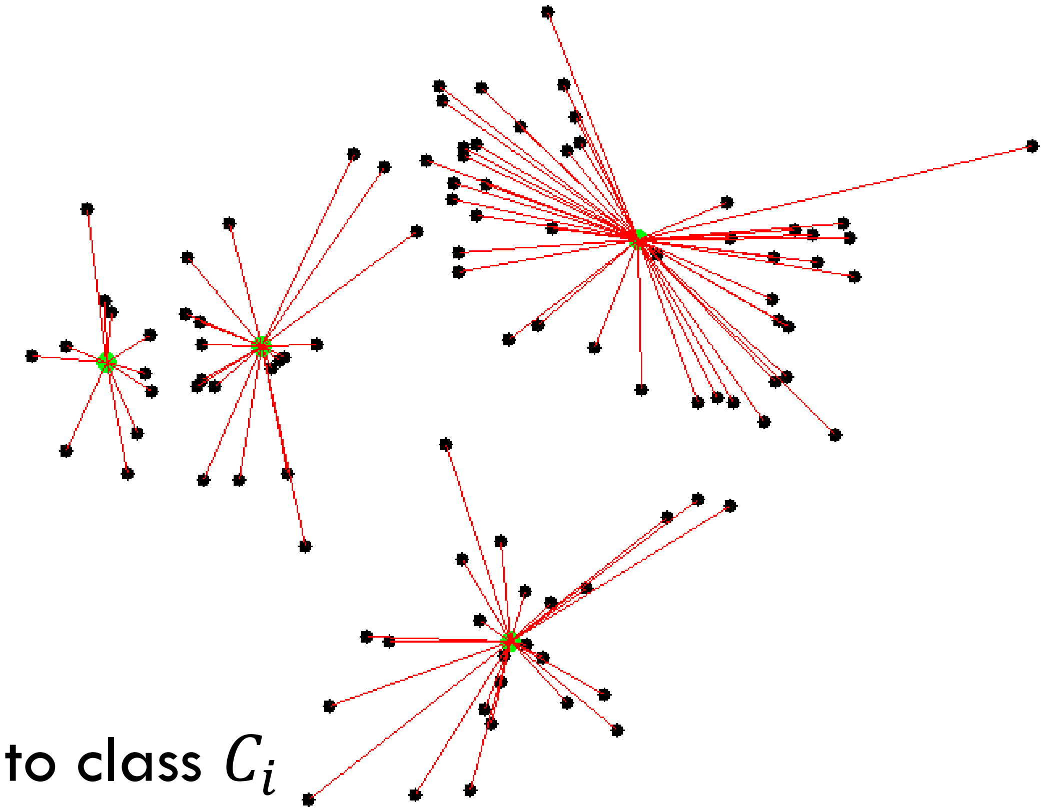
Resources

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

Recap/remaining loose ends

Slides from last week

K-means clustering



1. Decide on the number of clusters: k
2. Choose a set of arbitrary centroids:
   $\mu_1, \mu_2, \ldots, \mu_k$
3. For each item, x, in the training data,
   - find the nearest centroid μ_i, and assign x to class C_i
4. For each resulting class C_i, calculate and find the new centroid μ_i.
5. Classify each item according to the new centroids
6. Repeat from 4

Why does this work? How does this work?

- The goal is a mapping
  $\gamma: O \rightarrow C = \{C_1, C_2, \dots, C_k\}$
- We need a tool, F,
  - to measure the performance of γ
- The goal is to find a γ that optimizes F, in symbols
  $$\hat{\gamma} = \underset{\gamma}{\operatorname{argmax}} F(\gamma)$$
- F is called an objective function

- Several possible objectives:
  - High similarity (=small distance) within the clusters (intra-cluster)
  - Low similarity (high distance) between the clusters (inter-clusters)

Within cluster sum of squares (intra-cluster)

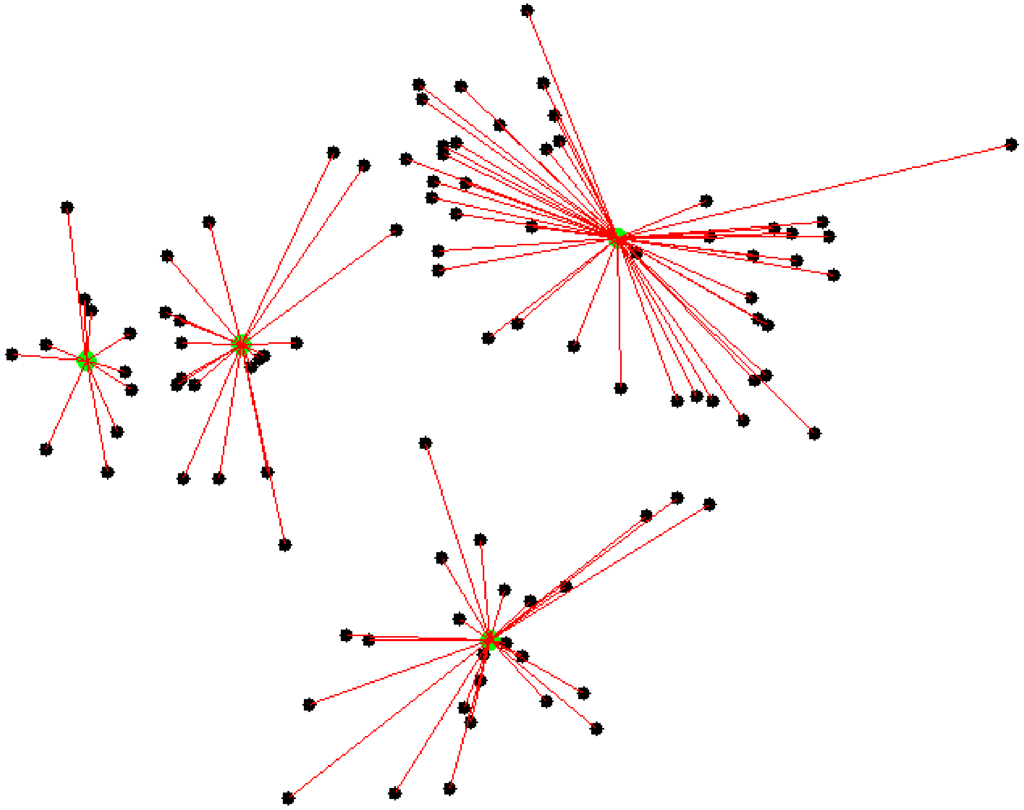
- For each cluster consider the sum of square distances:

$$SS_i = \sum_{x_j \in C_i} \left\| x_j - \mu_i \right\|^2$$

- Sum over all classes

$$WCSS = \sum_{i=1}^{k} SS_i$$

- To optimize F, is to find the γ that yields the smallest $WCSS$



Applied to *k*-means

- For each iteration:

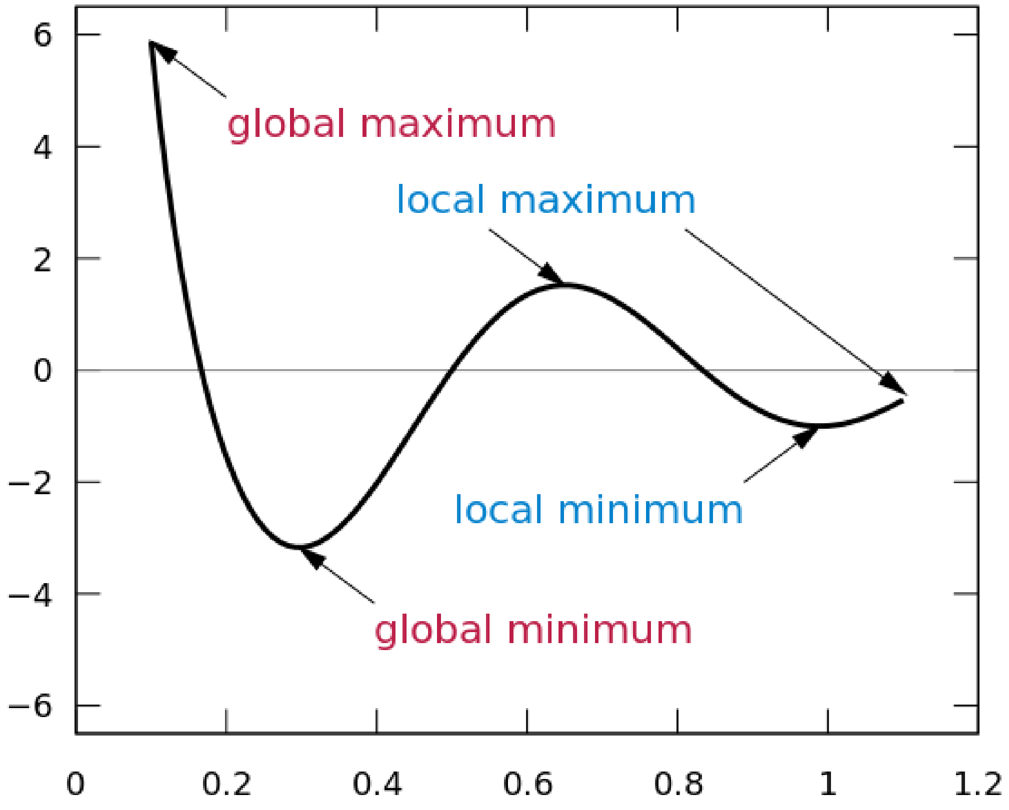
$$WCSS_{i+1} \leq WCSS_i$$

- Because:
  - Given a class, C_i, the recalculated centroid is the unique point in space that minimizes SS_i
  - If an item is moved from one class to another, its centroid-distance decreases

- Possible stopping criteria:
  - Fixed number of iterations
  - Clusters or centroids are unchanged between iterations.
  - Threshold on the decrease of the objective function (absolute or relative to previous iteration)

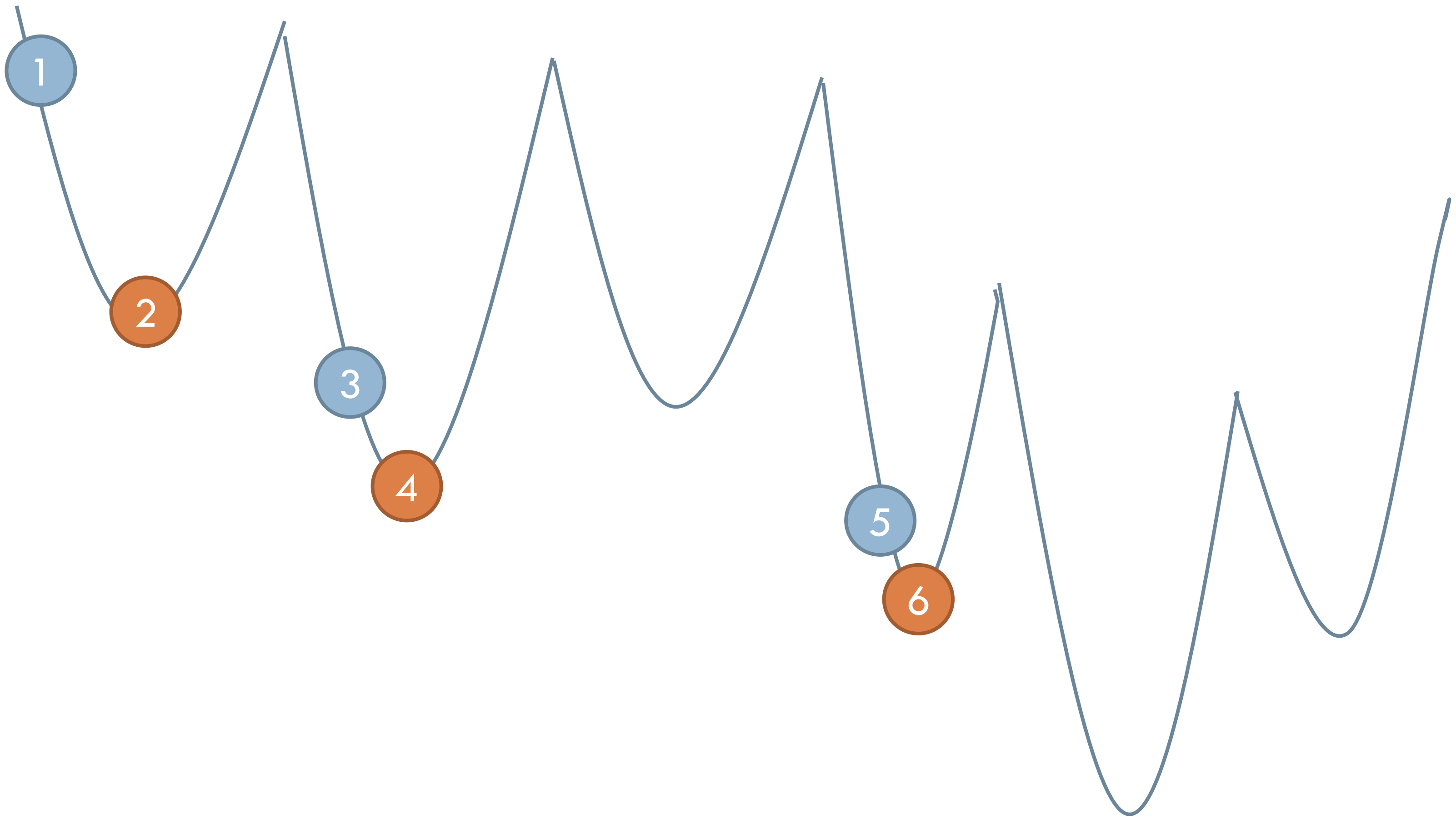
Properties of k-means

- The time complexity is linear, $O(kn)$
- Guaranteed to converge, but not to find the global optimal solution:
  - Depends on choice of initial centroids



Illustrating the WCSS

- New assignment to centroids (=reclassification)
- **Best centroids for this classification**



Comments

'Seeding'

- We initialize the algorithm by choosing random seeds that we use to compute the first set of centroids, e.g:
  - pick k random objects from the collection;
  - pick k random points in the space;
  - pick k sets of m random points and compute centroids for each set; etc.
- The seeds can have a large impact on the resulting clustering.
- Outliers are troublemakers.

- No prescribed way to choose *k*.
  - In particular, more *k*-s will always give better WCSS without being intuitively better.

Intrinsic evaluation of clustering

With labeled gold-data

- Run *k*-means on the gold set (without the labels).
- Compare the clusters to the classes:
  - Purity: a good cluster will have all members from the same class

Without using gold data

- We can use some intra-cluster or inter-cluster measure,
  - E.g., WCSS to compare which initial choice of centroids is better in *k*-means

Extrinsic evaluation

- See which clustering (or lack of clustering) yields the best results in a larger task
- For example: two versions of a recommender system, and measure some of:
  - User satisfaction
  - How many recommended articles they read, or click on
  - Improvement in sales

Flat Clustering: The good and the bad

Pros

- Conceptually simple, and easy to implement.
- Efficient. Typically linear in the number of objects.

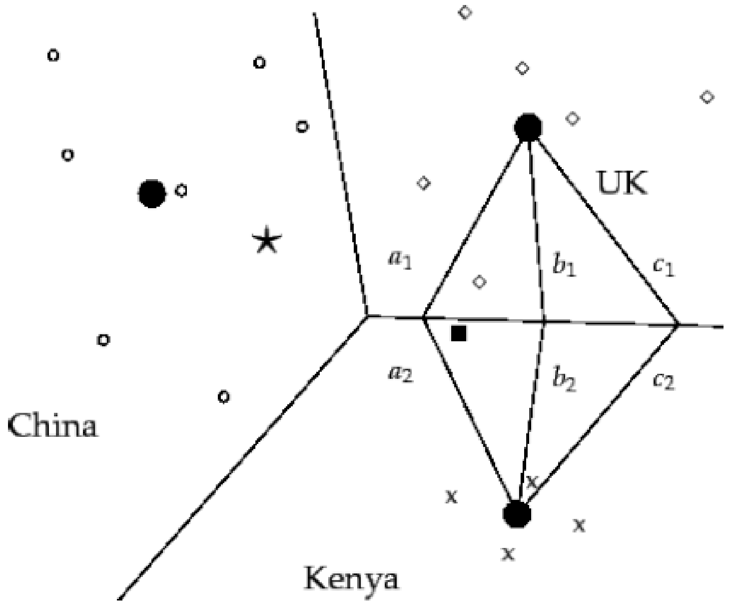
Cons

- The dependence on random seeds as in k-means makes the clustering non-deterministic.
- The number of clusters k must be pre-specified. Often no principled means of *a priori* specifying k.
- Not as informative as the more structured clusterings produced by hierarchical methods.

- In general; often difficult to evaluate clustering.

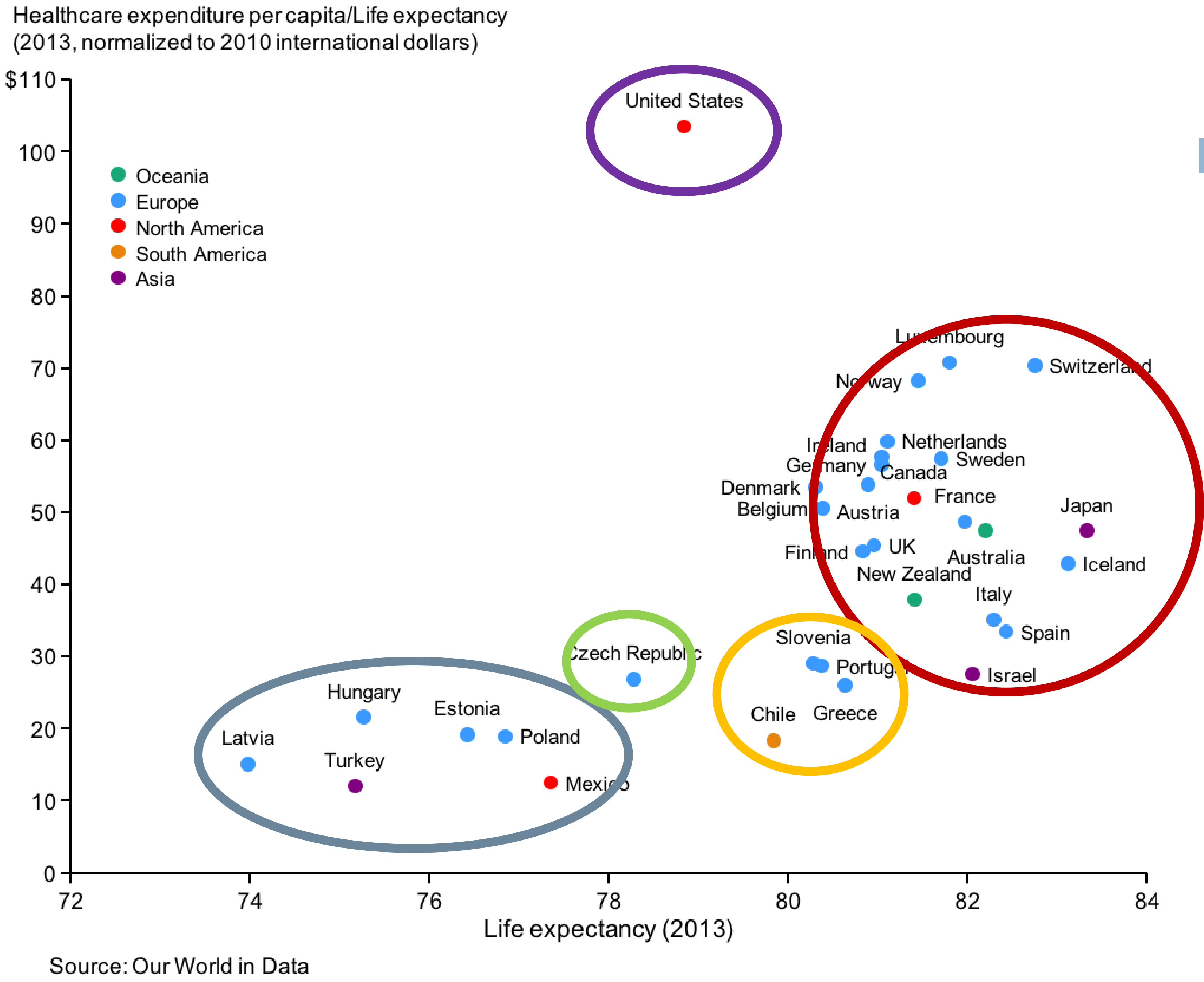
Connecting the dots

- We have seen how Rocchio classification can be thought of as a 1-Nearest-Neighbor classification with respect to the centroids.
- Note how k-means clustering can be thought of as performing Rocchio classification in each iteration.



Limitations

- Similar underling assumptions as the Rocchio classifier
- Assumes regions with the same diameter



Limitations

- Similar underling assumptions as the Rocchio classifier
- A Voronoi cell for each cluster, defined by the centroid