Computational lexical semantics

Computational Linguistics

Alexander Koller

31 January 2020

The knowledge bottleneck

- Inference requires formalized *knowledge* about the world and about the meanings of words.

Which genetically caused connective tissue disorder has severe symptoms and complications regarding the aorta and skeletal features, and, very characteristically, ophthalmologic subluxation?

Marfan's is created by a defect of the gene that determines the structure of Fibrillin-11. One of the symptoms is displacement of one or both of the eyes' lenses. The most serious complications affect the cardiovascular system, especially heart valves and the aorta.

Lexical semantics

Many words are *synonymous* or at least *semantically similar.*

He's not pining! He's passed on! This parrot is no more! He has ceased to be! He's expired and gone to meet his maker! He's a stiff! Bereft of life, he rests in peace! His metabolic processes are now history! He's off the twig! He's kicked the bucket, he's shuffled off his mortal coil, run down the curtain and joined the bleedin' choir invisible!! THIS IS AN EX-PARROT!!

Information Retrieval

- In *Information Retrieval,* we want to find differently phrased documents:
  - Query: “female astronauts”
  - Document: “In the history of the Soviet space program, there were only three female cosmonauts: Valentina Tereshkova, Svetlana Savitskaya, and Elena Kondakova.”
- This will only work if system recognizes that “astronaut” and “cosmonaut” have similar meanings.

Machine Translation

- Knowledge also important to disambiguate *polysemous* words.

- Famous example by Bar-Hillel (1960):
  - "The box is in the pen."

- Correct translation depends on sense of "pen":
  - "Die Kiste ist im Stift."
  - "Die Kiste ist im Pferch."

Classical lexical semantics

- *Polysemy:* Word has two different meanings that are clearly related to each other.
    - School #1: institution at which students learn
    - School #2: building that houses school #1

- *Homonyny:* Word has two different meanings that have no obvious relation to each other.
    - Bank #1: financial institution
    - Bank #2: land alongside a body of water

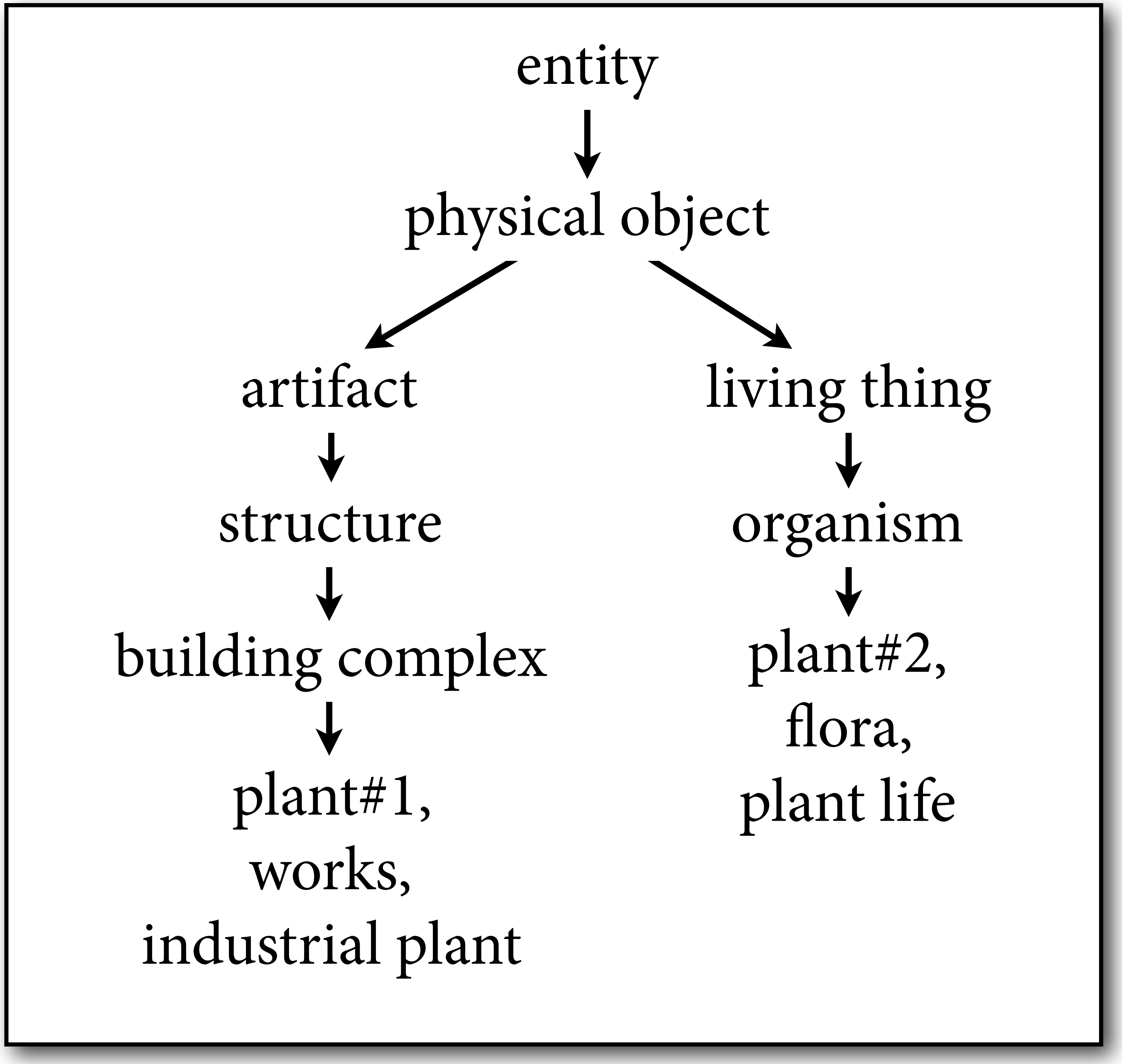
Word sense disambiguation

- *Word sense disambiguation* is the problem of tagging each word token with its word sense.

- WSD accuracy depends on sense inventory;
  state of the art is above 90% on coarse-grained senses.

- Techniques tend to combine supervised training on small amount of annotated data with unsupervised methods.

Classical lexical semantics



entity

physical object

artifact

structure

building complex

plant#1,
works,
industrial plant

living thing

organism

plant#2,
flora,
plant life

→ = hyponymy

same node = synonymy

http://wordnet.princeton.edu/

Problem

- Hand-written thesauruses much too small.
  - English Wordnet: 117.000 synsets
  - GermaNet: 85.000 synsets

- Number of word types in English Google n-gram corpus: > 1 million.

- This is not how we can solve the query expansion problem

- Learn lexical semantic knowledge automatically?

Experiment

- What is “bardiwac”? Some occurrences in corpus:
  - He handed her a glass of 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.
  - The drinks were delicious: blood-red bardiwac as well as light, sweet Rhenish.

→ Bardiwac ist a red wine.

(Stefan Evert, tutorial at NAACL 2010)

Distributional Semantics

- Basic idea (Harris 1951, Firth 1957):
  “You shall know a word by the company it keeps.”

- Assumption: Semantically *similar* words tend to occur in the context of the same words.
  - “similar” as approximation of “synonymous”

- Can observe “occur in the context of same words” on large unannotated corpora.

Cooccurrence

see who can grow the biggest flower. Can we buy some fibre, please

Abu Dhabi grow like a hot-house flower, but decided themselves to follow the

as a physical level. The Bach Flower Remedies are prepared from non-poisonous wild

a seed from which a strong tree will grow. This is the finest

↓

	factory	flower	tree	plant	water	fork
grow	15	147	330	517	106	3
garden	5	200	198	316	118	17
worker	279	0	5	84	18	0
production	102	6	9	130	28	0
wild	3	216	35	96	30	0

Co-occurrence matrix from BNC, from Koller & Pinkal 12

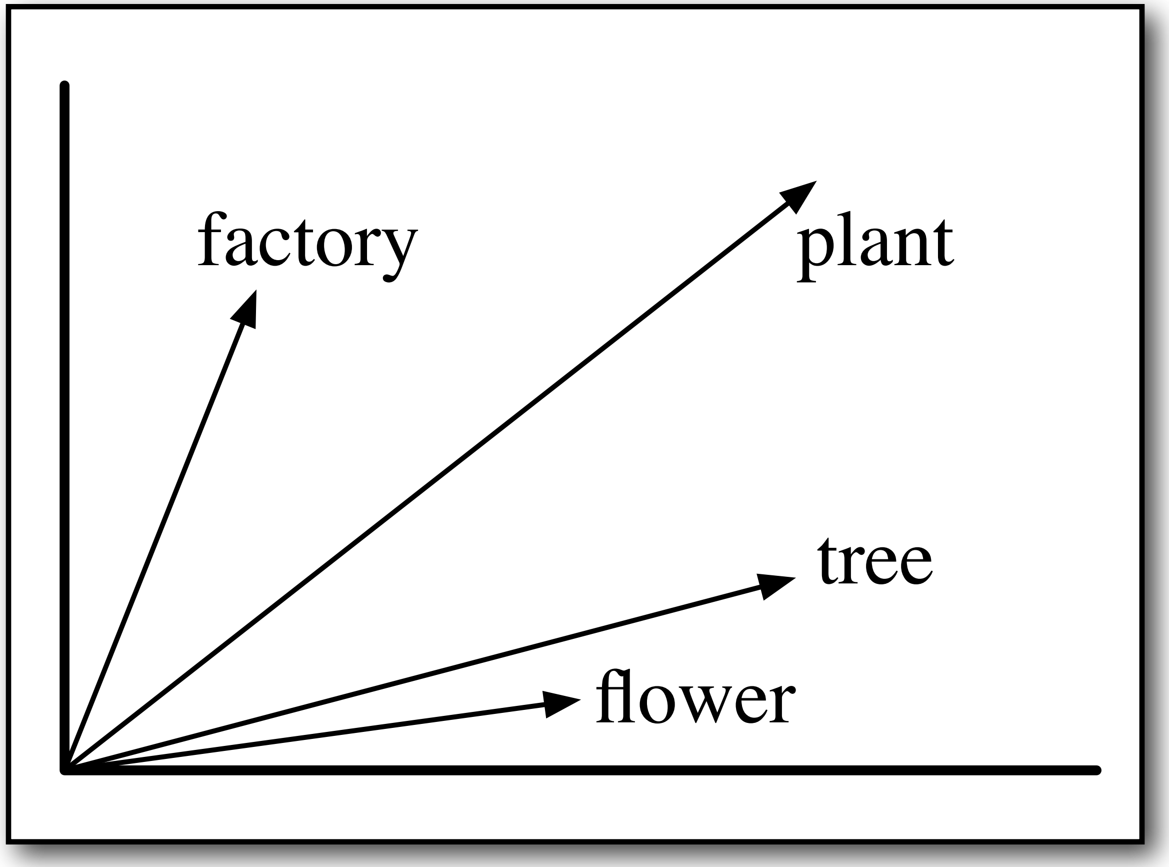
Vector space model

	factory	flower	tree	plant	water	fork
grow	15	147	330	517	106	3
garden	5	200	198	316	118	17
worker	279	0	5	84	18	0
production	102	6	9	130	28	0
wild	3	216	35	96	30	0

1 dimension per context word
(here: 6 dimensions)

Picture simplifies to 2 dimensions, is only schematic.

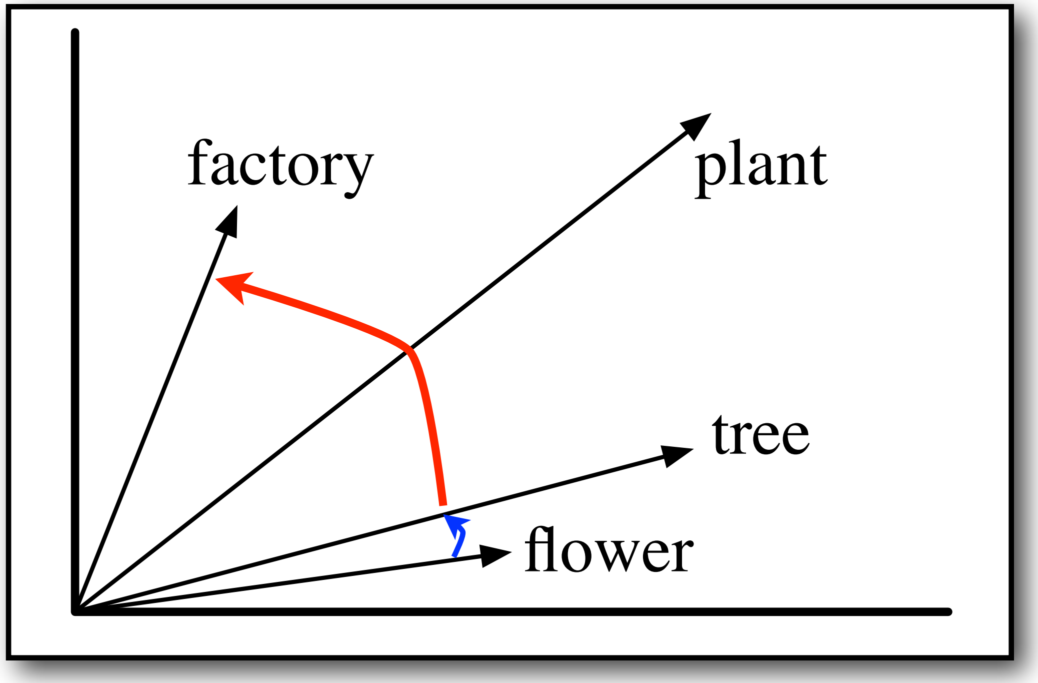
Vectors in a high-dimensional vector space



Cosine similarity

- Take *angle* between vectors as measure of similarity.
  - (correctly) ignores length of vectors = frequency of words
  - similar angle = similar proportion of context words

- Cosine of angle is easy to compute.
  - cos = 1 means angle = 0°, i.e. very similar
  - cos = 0 means angle = 90°, i.e. very dissimilar



$$\cos(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{n} v_i \cdot w_i}{\sqrt{\sum_{i=1}^{n} v_i^2} \cdot \sqrt{\sum_{i=1}^{n} w_i^2}}$$

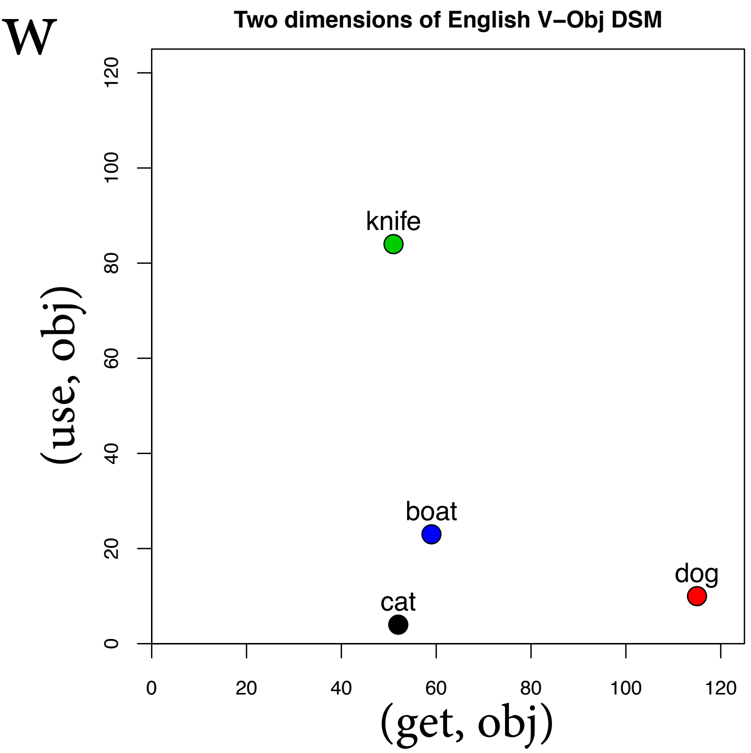
cos(tree, flower) = 0.75, i.e. 40°
cos(tree, factory) = 0.05, i.e. 85°

More complex features

- Co-occurrence in string can over-estimate whether the two words really belong together.

the Qataris had watched Abu Dhabi grow like a hot-house flower, but decided

- Fix this with more complex features which e.g. capture grammatical relations between words (Lin 98).
  - instead of counting "*flower* appears in window of length 7 around *Abu Dhabi*",
  - count "*flower* occurs as subject of *grow*"

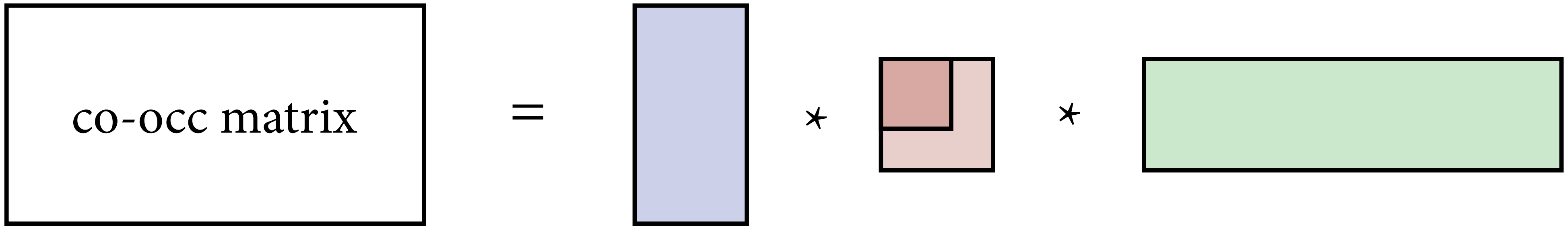


Dimensionality reduction

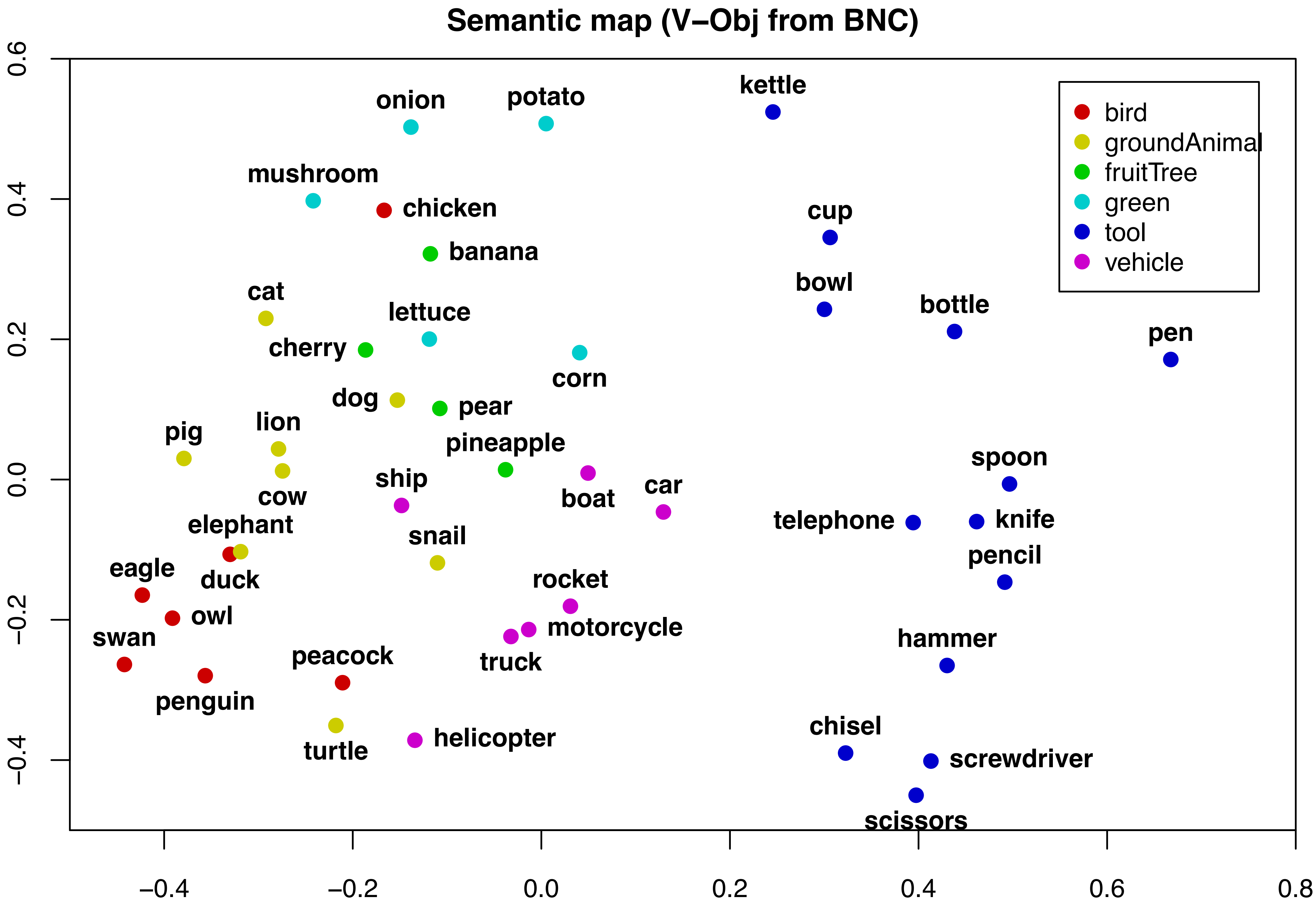
- Raw co-occurrence vectors have very high dimension (one for each context word).

- Typical approach: dimensionality reduction.
    - improves efficiency; can filter out random noise

- For instance, *Latent Semantic Analysis* reduces dimensionality via singular value decomposition.

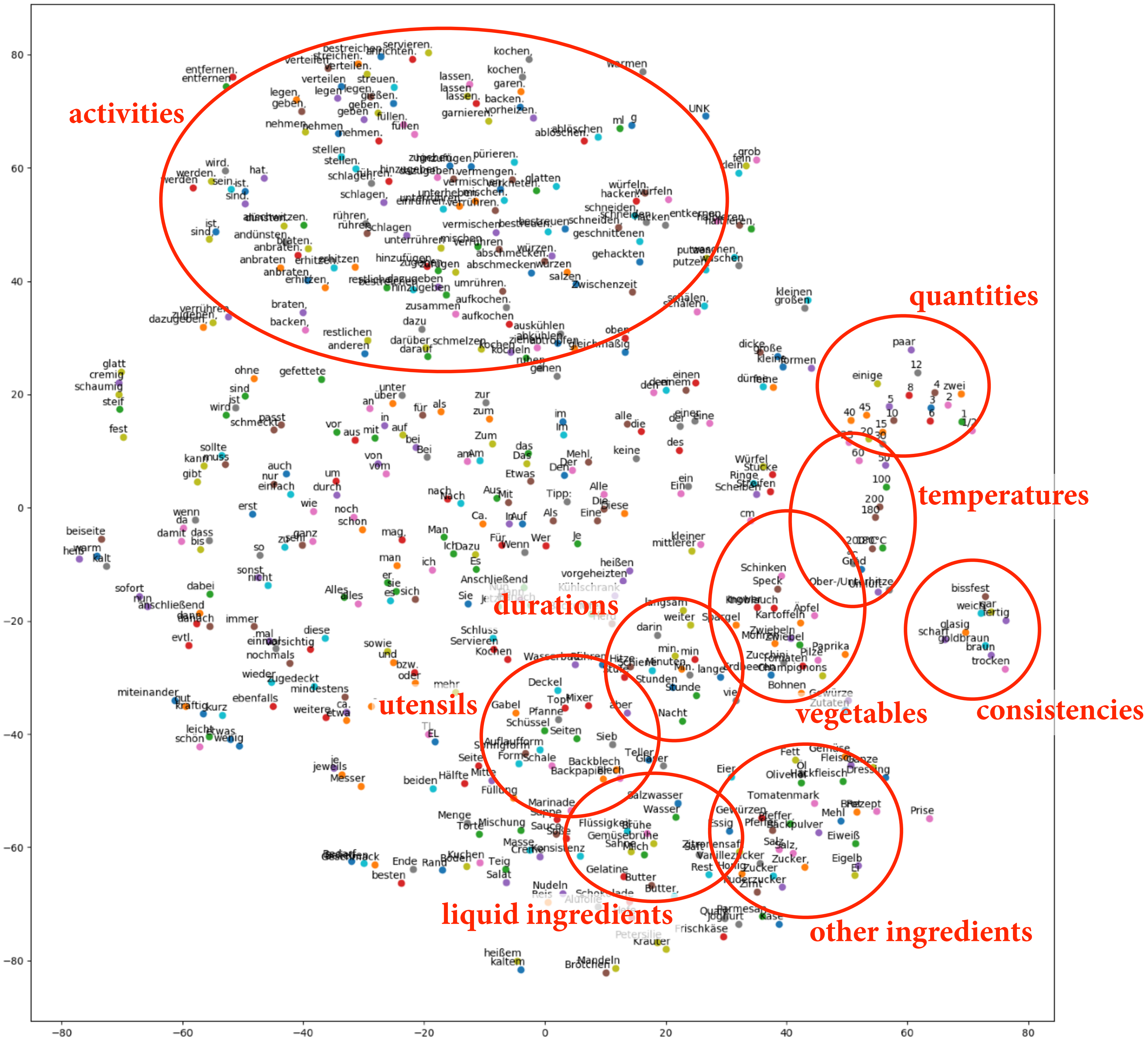


Results



(Evert, NAACL tutorial 2010)

German cooking words



word2vec embeddings, Theresa Schmidt's BSc thesis, 2020

Results

hope (N):
optimism 0.141, chance 0.137, expectation 0.136, prospect 0.126, dream 0.119, desire 0.118, fear 0.116, effort 0.111, confidence 0.109, promise 0.108

hope(V):
would like 0.158, wish 0.140, plan 0.139, say 0.137, believe 0.135, think 0.133, agree 0.130, wonder 0.130, try 0.127, decide 0.125

brief (N):
legal brief 0.139, affidavit 0.103, filing 0.098, petition 0.086, document 0.083, argument 0.083, letter 0.079, rebuttal 0.078, memo 0.077, article 0.076

brief (A):
lengthy 0.256, hour-long 0.191, short 0.173, extended 0.163, frequent 0.162, recent 0.158, short-lived 0.155, prolonged 0.149, week-long 0.149, occasional 0.146

(results of Lin 98, from J&M)

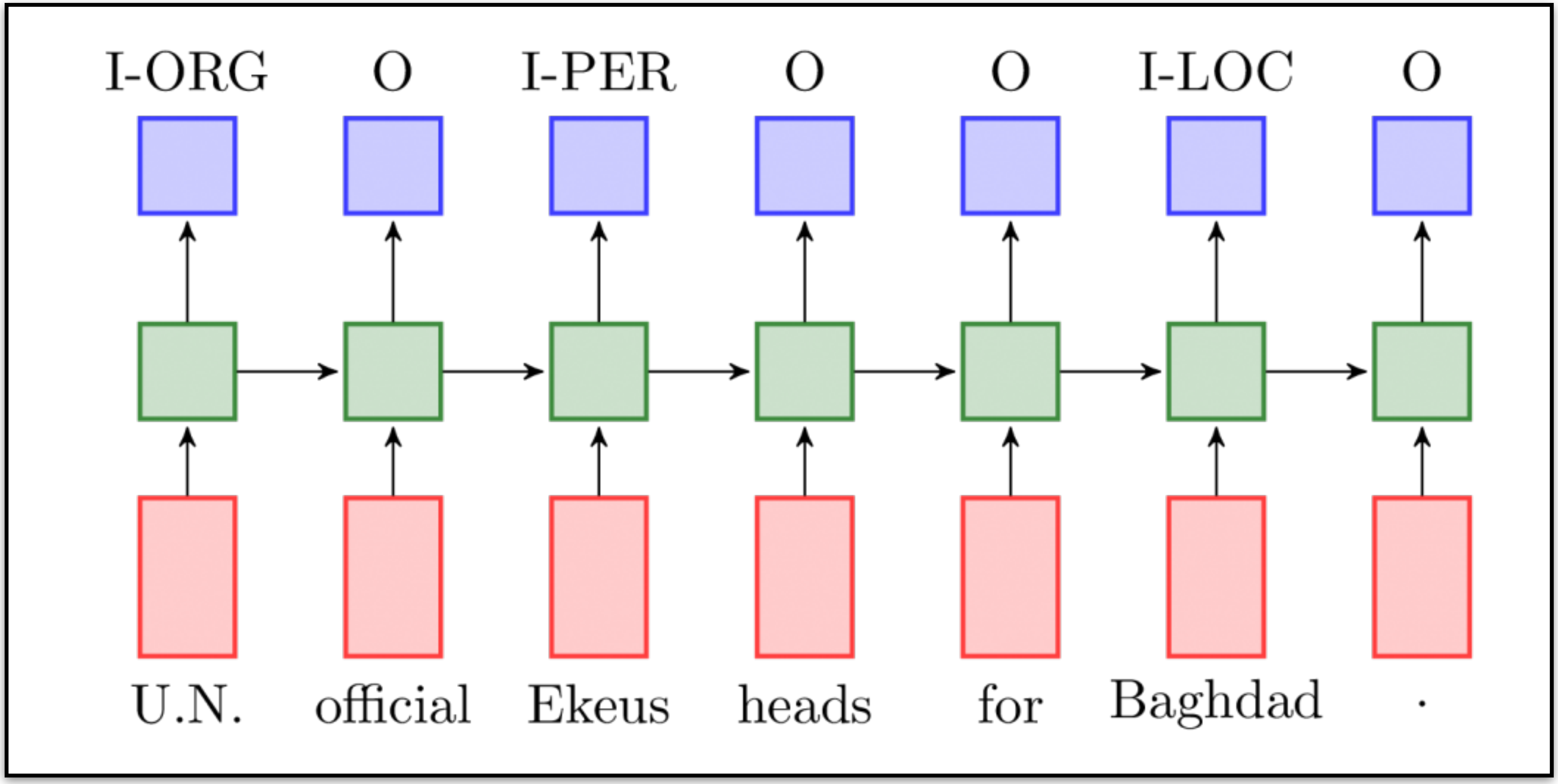
Problems

- Similarity = synonymy?
    - Antonyms are basically as distributionally similar as synonyms:

brief (A): lengthy 0.256, hour-long 0.191, short 0.173, extended 0.163, frequent 0.162, recent 0.158, short-lived 0.155, prolonged 0.149, week-long 0.149, occasional 0.146

- Distributional similarity is not referential similarity. Distinguishing synonyms from antonyms is notoriously hard problem.

Word embeddings for NNs



neural named entity recognizer (schematic)

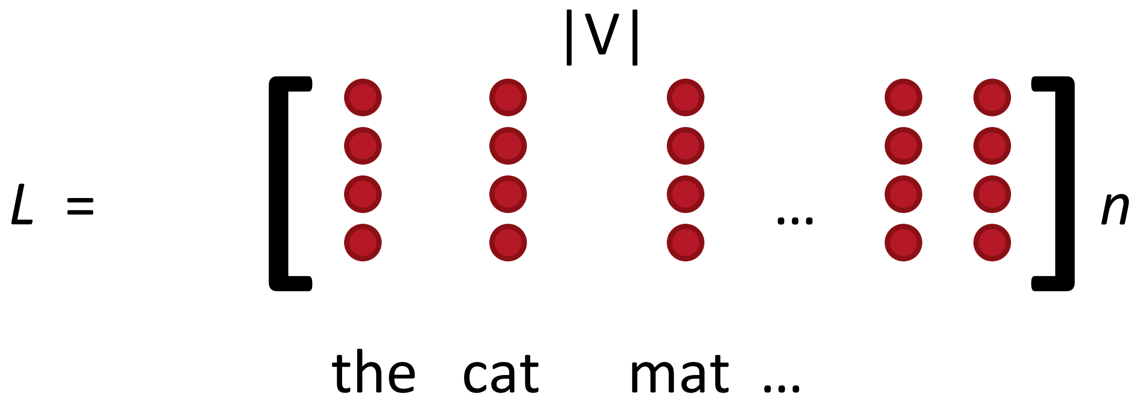
Neural networks cannot directly read words.
Need to map each word to a vector, called a *word embedding*.

Word embeddings

- One-hot encoding: Every word a 0-1 vector.

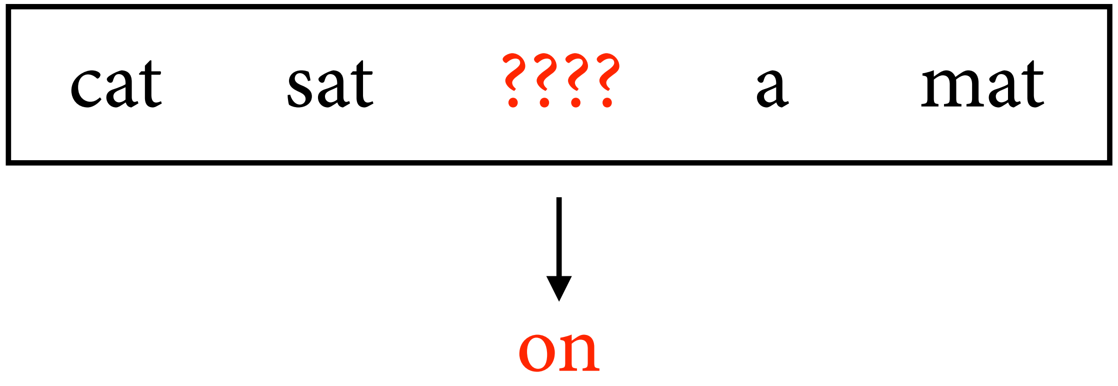
$$\begin{array}{cccccc} \text{the} & \text{cat} & \text{mat} & \text{sat} & \text{a} & \\ \begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix} & \begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix} & \begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix} & \begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix} & \begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix} & \updownarrow |V| \end{array}$$

- Word embeddings: n x |V| matrix L which maps each one-hot encoding into n-dimensional vector.



Word embeddings

- Idea: try to predict the missing word in a given context.



cat sat ???? a mat

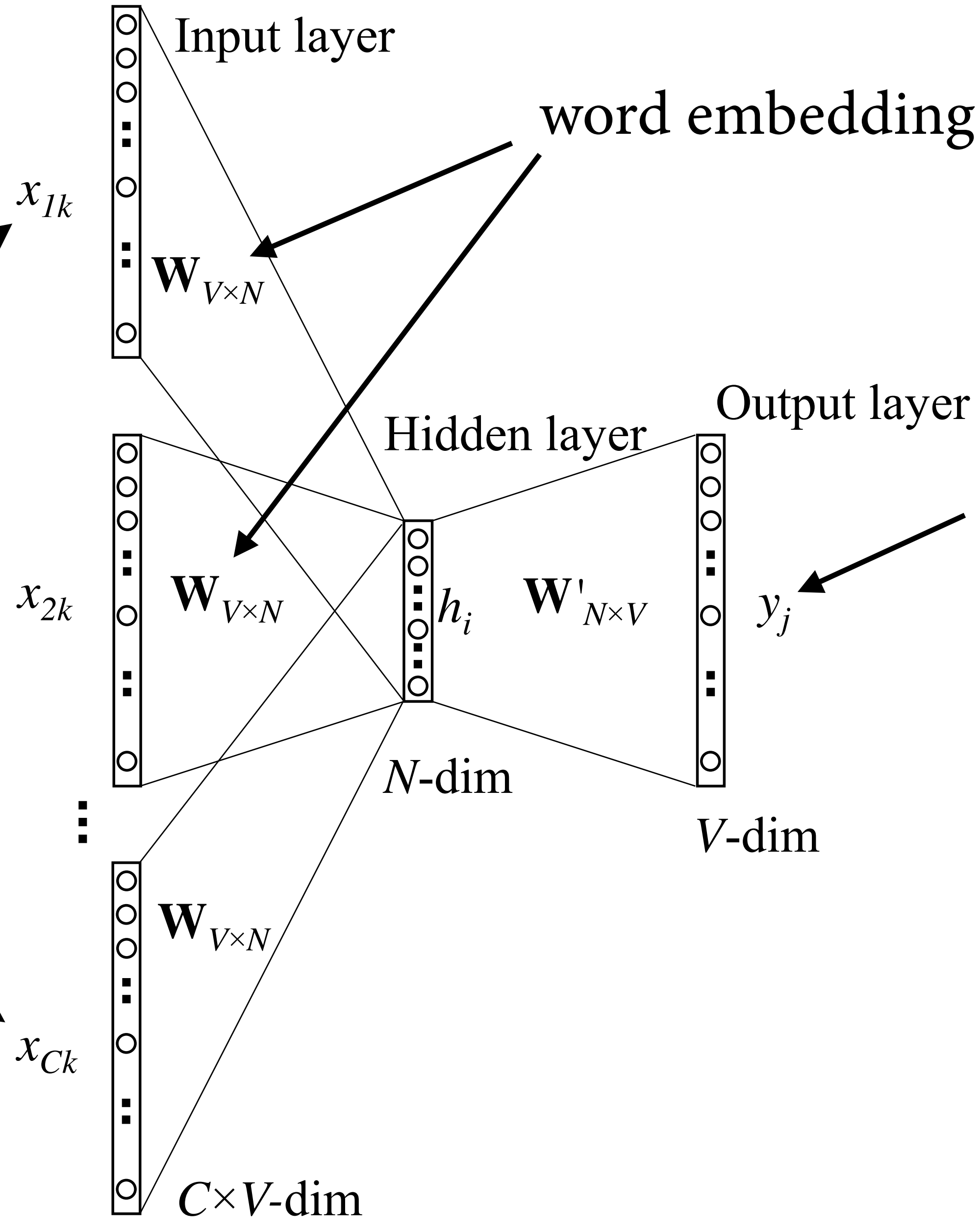
on

- Train neural network to learn “distributional” vectors for all the words.

CBOW

= continuous bag of words

similarly, skip-gram model predicts context from word



word embedding matrix L

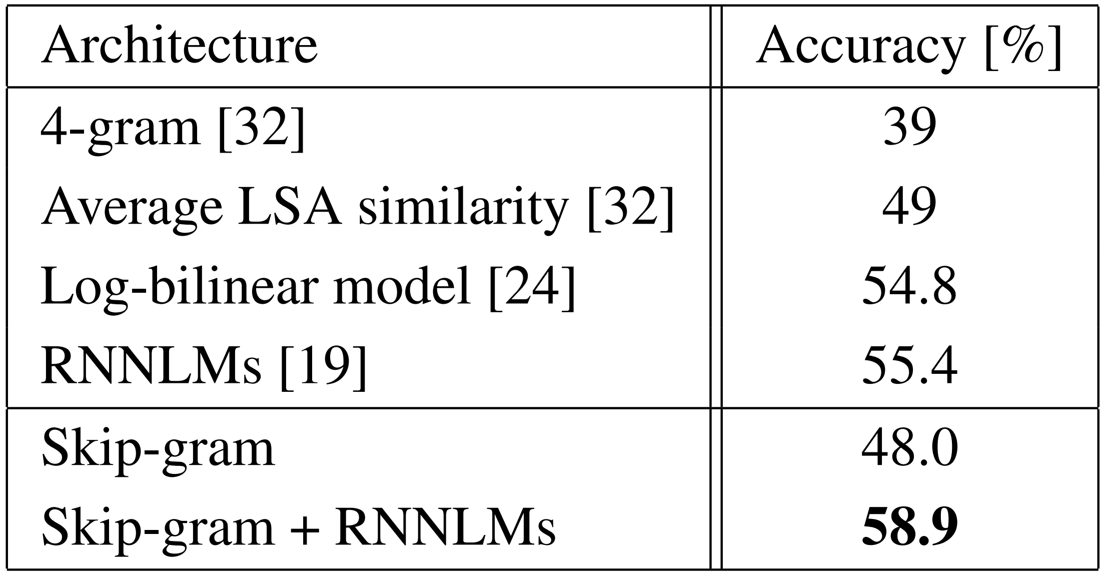
one-hot encodings
of words at positions
-2, -1, +1, +2

“one-hot encoding”
of word at position 0
(see next slide)

(word2vec; Mikolov et al. 2013; Levy & Goldberg 2014: math. parallels to SVD)

Results

Accuracy on MS Sentence Completion Task



Architecture	Accuracy [%]
4-gram [32]	39
Average LSA similarity [32]	49
Log-bilinear model [24]	54.8
RNNLMs [19]	55.4
Skip-gram	48.0
Skip-gram + RNNLMs	**58.9**

Was she his [client | musings | discomfigure | choice | opportunity], his friend, or his mistress?

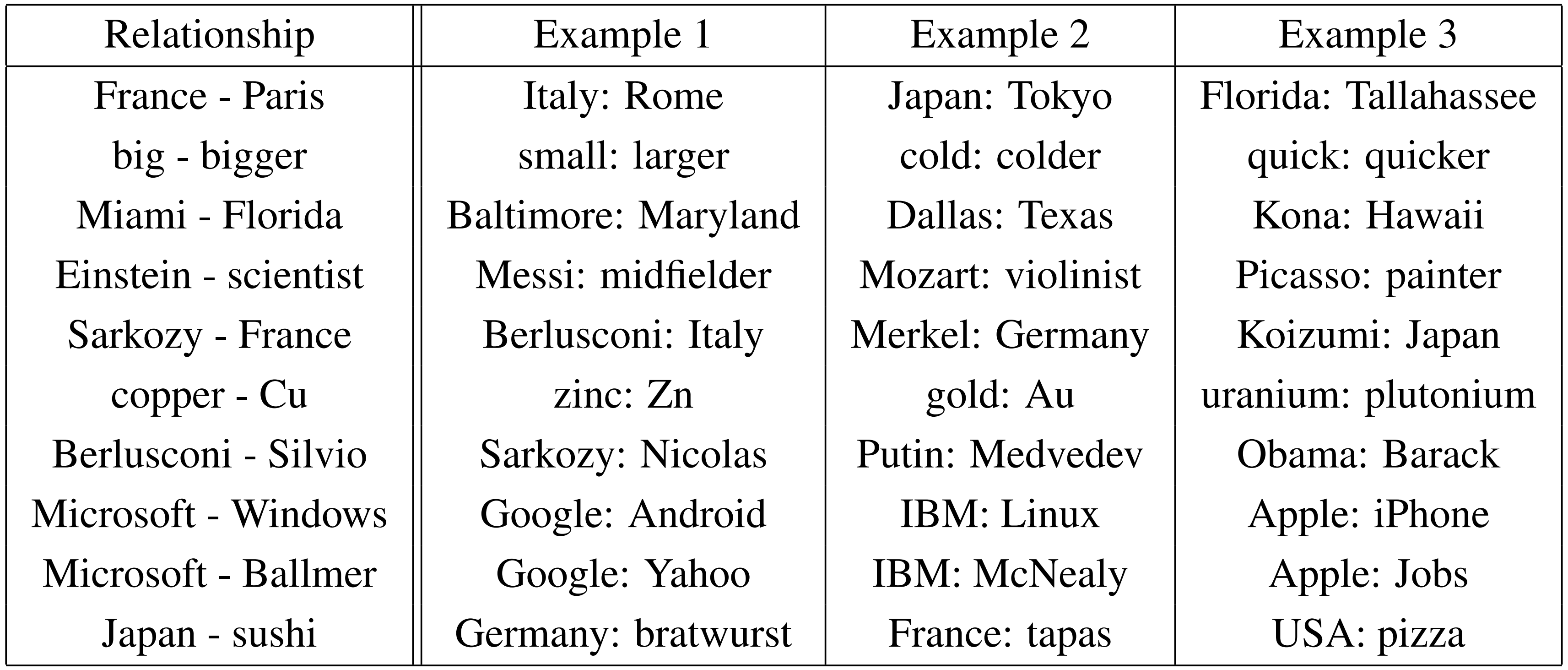
All red-headed men who are above the age of [800 | seven | twenty-one | 1,200 | 60,000] years are eligible.

That is his [generous | mother's | successful | favorite | main] fault, but on the whole he's a good worker.

(Mikolov et al. 2013)

Results

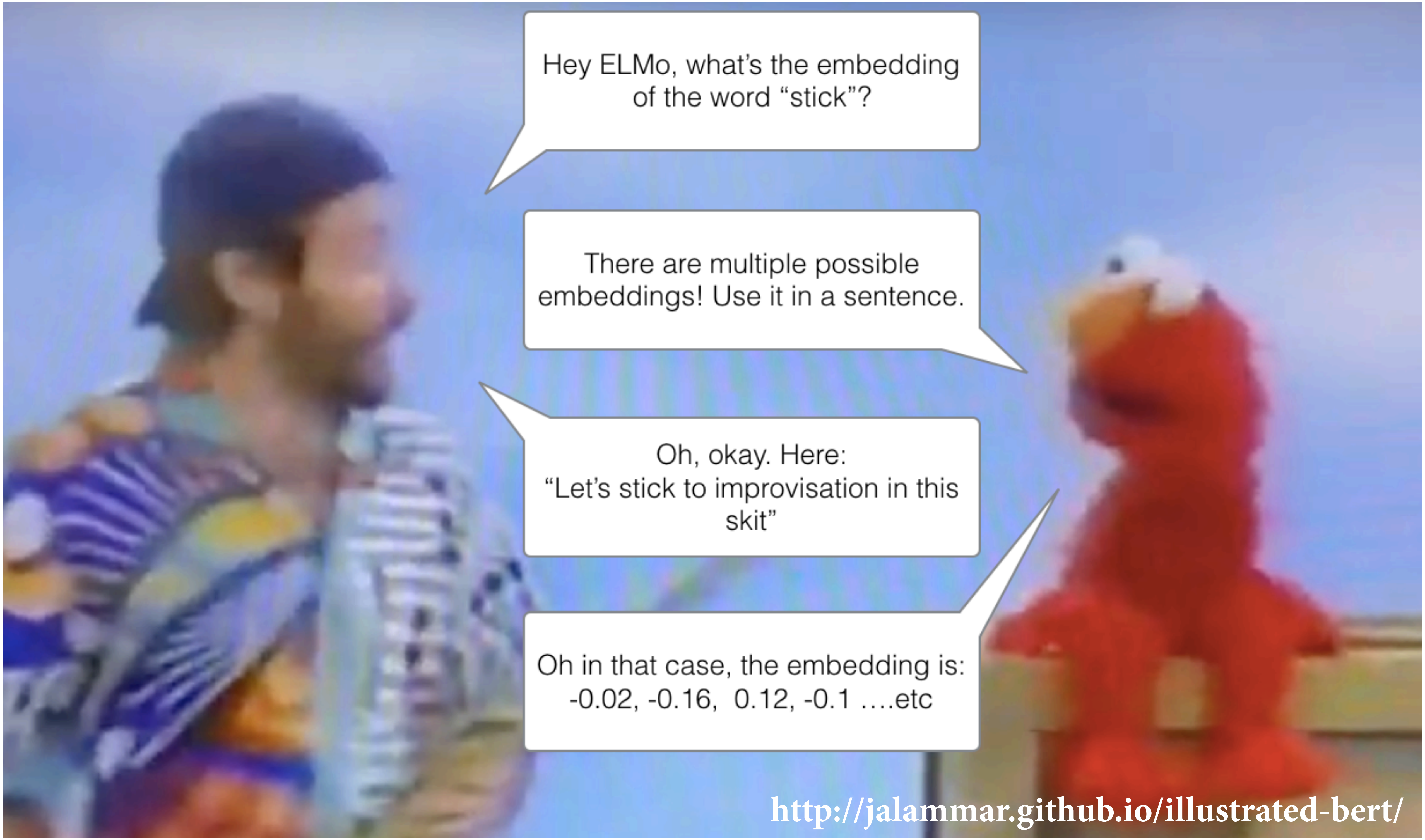
(on analogy task)



Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

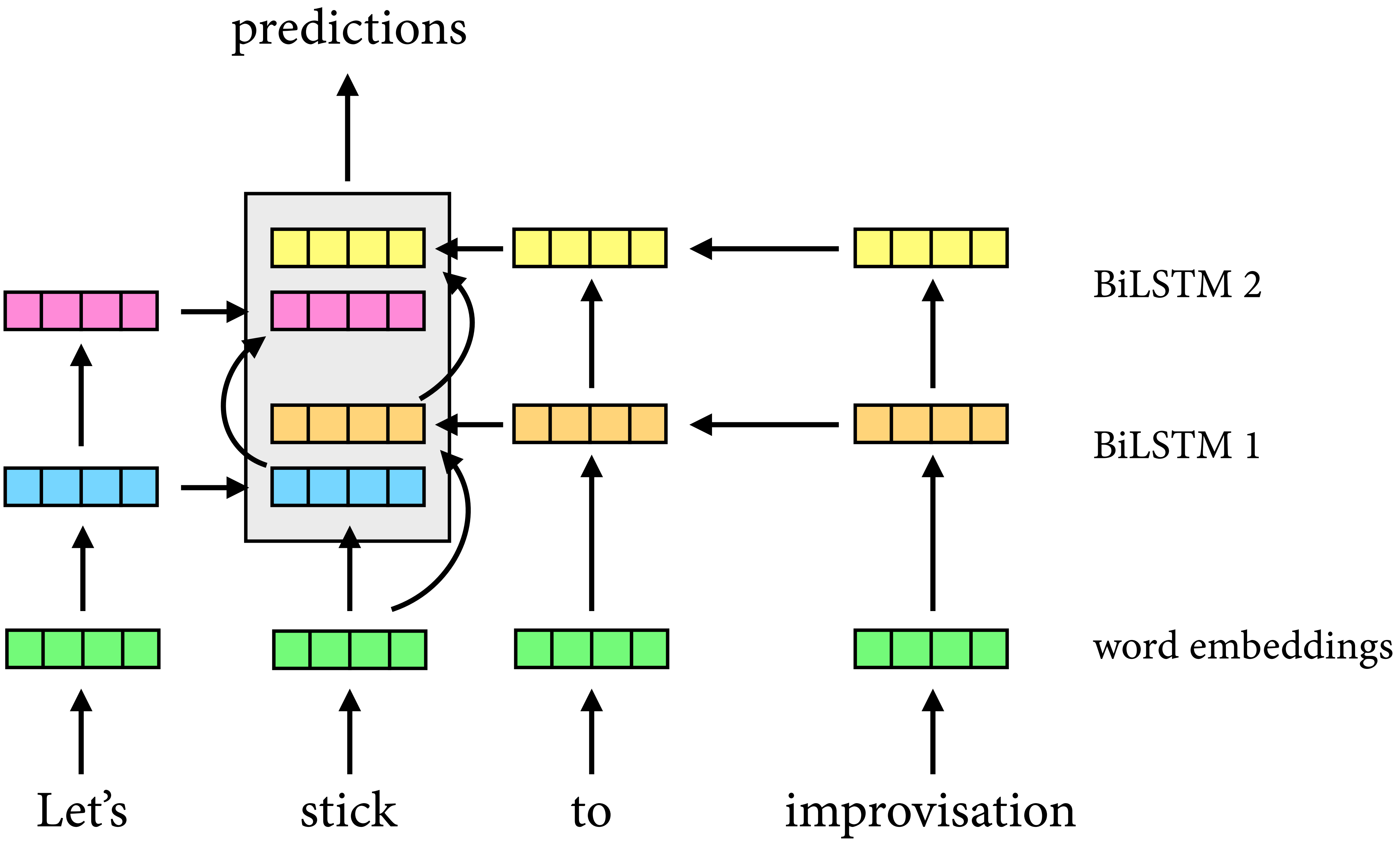
(vector with highest cosine similarity to L(Paris) - L(France) + L(Italy) etc.)

Contextualized word embeddings


Hey ELMo, what's the embedding of the word "stick"?
There are multiple possible embeddings! Use it in a sentence.
Oh, okay. Here: "Let's stick to improvisation in this skit"
Oh in that case, the embedding is: -0.02, -0.16, 0.12, -0.1 ….etc

http://jalammar.github.io/illustrated-bert/

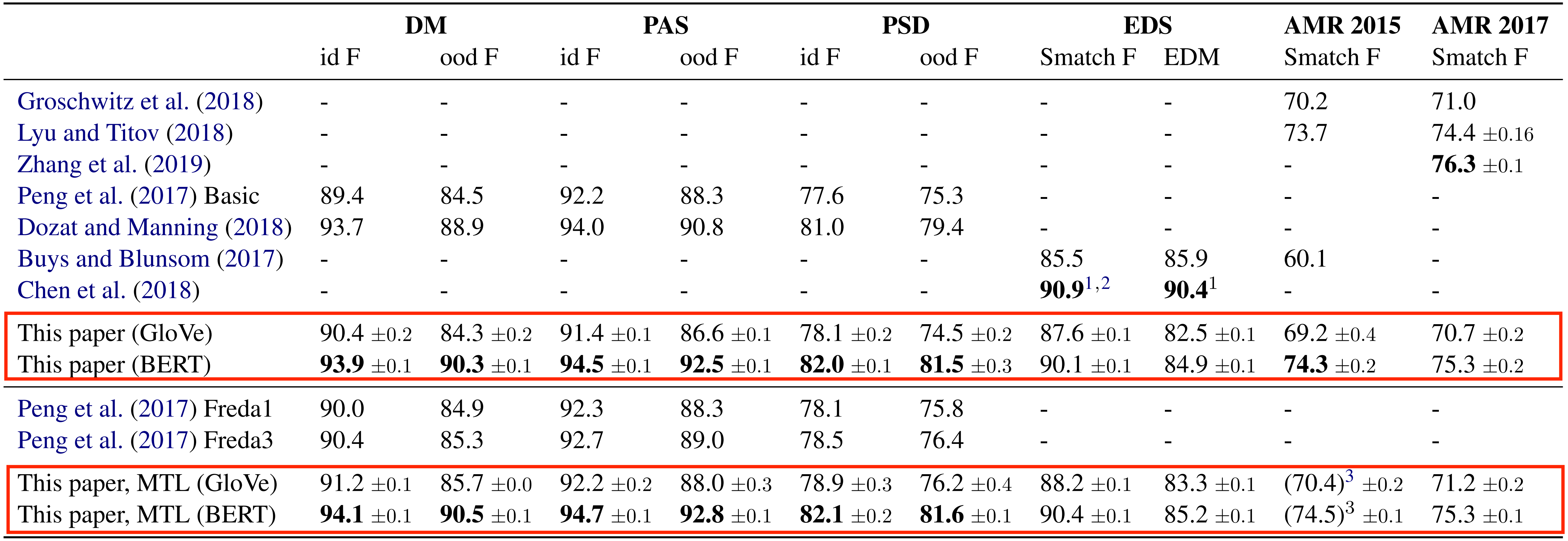
ELMo



Alternatively, use attention; currently everyone is using BERT.

(Peters et al. 2018)

Contextualization really helps



	DM		**PAS**		**PSD**		**EDS**		**AMR 2015**	**AMR 2017**
	id F	ood F	id F	ood F	id F	ood F	Smatch F	EDM	Smatch F	Smatch F
Groschwitz et al. (2018)	-	-	-	-	-	-	-	-	70.2	71.0
Lyu and Titov (2018)	-	-	-	-	-	-	-	-	73.7	74.4 ±0.16
Zhang et al. (2019)	-	-	-	-	-	-	-	-	-	**76.3** ±0.1
Peng et al. (2017) Basic	89.4	84.5	92.2	88.3	77.6	75.3	-	-	-	-
Dozat and Manning (2018)	93.7	88.9	94.0	90.8	81.0	79.4	-	-	-	-
Buys and Blunsom (2017)	-	-	-	-	-	-	85.5	85.9	60.1	-
Chen et al. (2018)	-	-	-	-	-	-	**90.9**[1,2]	**90.4**[1]	-	-
This paper (GloVe)	90.4 ±0.2	84.3 ±0.2	91.4 ±0.1	86.6 ±0.1	78.1 ±0.2	74.5 ±0.2	87.6 ±0.1	82.5 ±0.1	69.2 ±0.4	70.7 ±0.2
This paper (BERT)	**93.9** ±0.1	**90.3** ±0.1	**94.5** ±0.1	**92.5** ±0.1	**82.0** ±0.1	**81.5** ±0.3	90.1 ±0.1	84.9 ±0.1	**74.3** ±0.2	75.3 ±0.2
Peng et al. (2017) Freda1	90.0	84.9	92.3	88.3	78.1	75.8	-	-	-	-
Peng et al. (2017) Freda3	90.4	85.3	92.7	89.0	78.5	76.4	-	-	-	-
This paper, MTL (GloVe)	91.2 ±0.1	85.7 ±0.0	92.2 ±0.2	88.0 ±0.3	78.9 ±0.3	76.2 ±0.4	88.2 ±0.1	83.3 ±0.1	(70.4)[3] ±0.2	71.2 ±0.2
This paper, MTL (BERT)	**94.1** ±0.1	**90.5** ±0.1	**94.7** ±0.1	**92.8** ±0.1	**82.1** ±0.2	**81.6** ±0.1	90.4 ±0.1	85.2 ±0.1	(74.5)[3] ±0.1	75.3 ±0.1

- First semantic parser that does well across all six major graphbanks.
- Established new states of the art through use of pretrained BERT embeddings.
- Small improvements through multi-task learning on multiple graphbanks.

(Lindemann et al., ACL 2019)

Conclusion

- “Knowledge bottleneck” is a serious problem in computational semantics. Try to overcome by modeling information about word meaning.
- Classical task: word sense disambiguation (WSD).
- Distributional methods:
  - co-occurrence-based
  - neural (word embeddings, e.g. word2vec/GloVe)
  - latest cry: context-dependent word embeddings, e.g. ELMo, BERT