COS 598C Advanced Topics in Computer Science:
Deep Learning for Natural Language Processing

Word Embeddings

Winter 2020

Most popular topics

- #1: Adversarial examples (63% chose “I love to”)
- #2: Bias in Language (53%)
- #3: Dialogue I (50%)
- #4: Interpretability
- #5: Generalization
- #6: Reading Comprehension

Least popular topics

- #1: Coreference resolution (47% chose “I am not really interested”)
- #2: Annotation artifacts (43%)
- #3: Semantic parsing (43%)

Suggested topics

Sentence embedding, compositionally, language + vision
No additional topics, but would love to spend more time on general linguistic intelligence!
Danqi's current working projects.
Translation, but I know that's not an option :)
No
Perhaps Machine translation, but I believe a good amount of other material is already covered

Overview

- (Mikolov et al, 2013) **Distributed Representations of Words and Phrases and their Compositionality**

- (Baroni et al, 2014) **Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors**

Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov
Google Inc.
Mountain View
mikolov@google.com

Ilya Sutskever
Google Inc.
Mountain View
ilyasu@google.com

Kai Chen
Google Inc.
Mountain View
kai@google.com

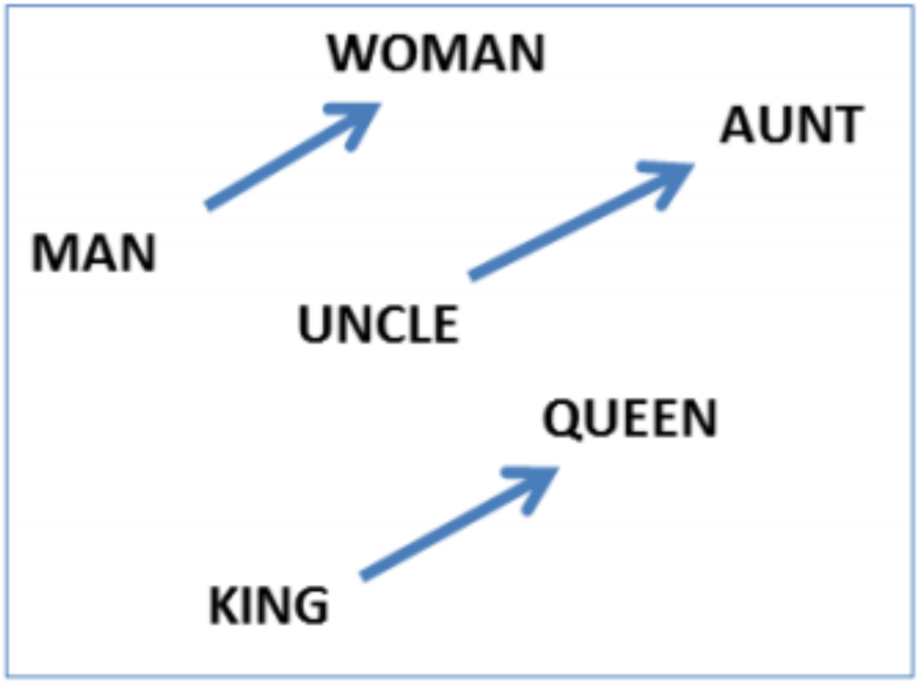
Greg Corrado
Google Inc.
Mountain View
gcorrado@google.com

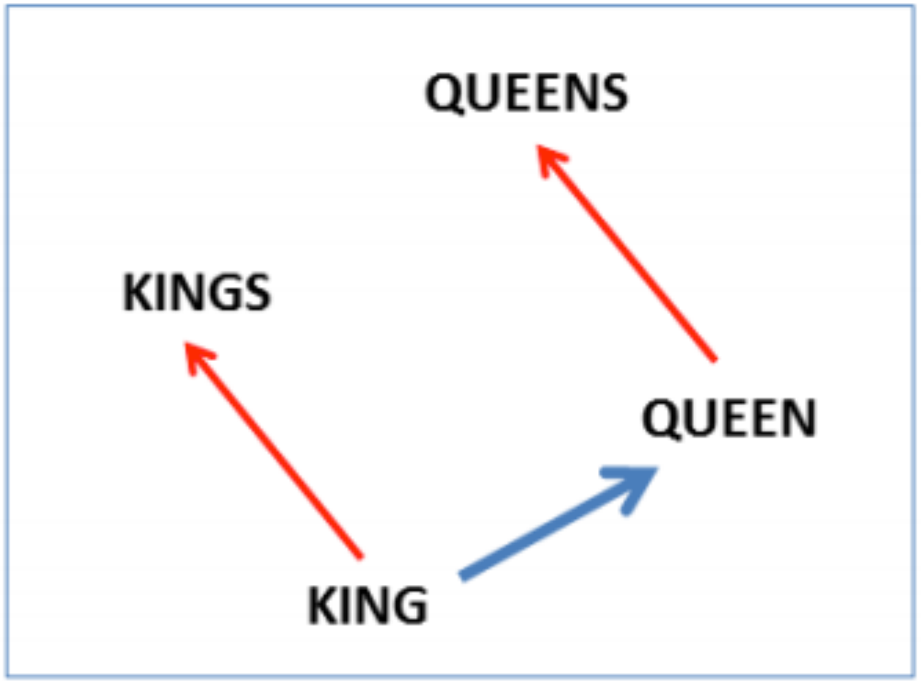
Jeffrey Dean
Google Inc.
Mountain View
jeff@google.com

Distributed representation of words

$$\text{employees} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 10.109 \\ -0.542 \\ 0.349 \\ 0.271 \\ 0.487 \end{pmatrix}$$

Linguistic regularities





(Mikolov et al, 2013) Linguistic Regularities in Continuous Space Word Representations

Distributional hypothesis

Distributional hypothesis: words that occur in similar contexts tend to have similar meanings

J.R.Firth 1957

- "You shall know a word by the company it keeps"
- One of the most successful ideas of modern statistical NLP!

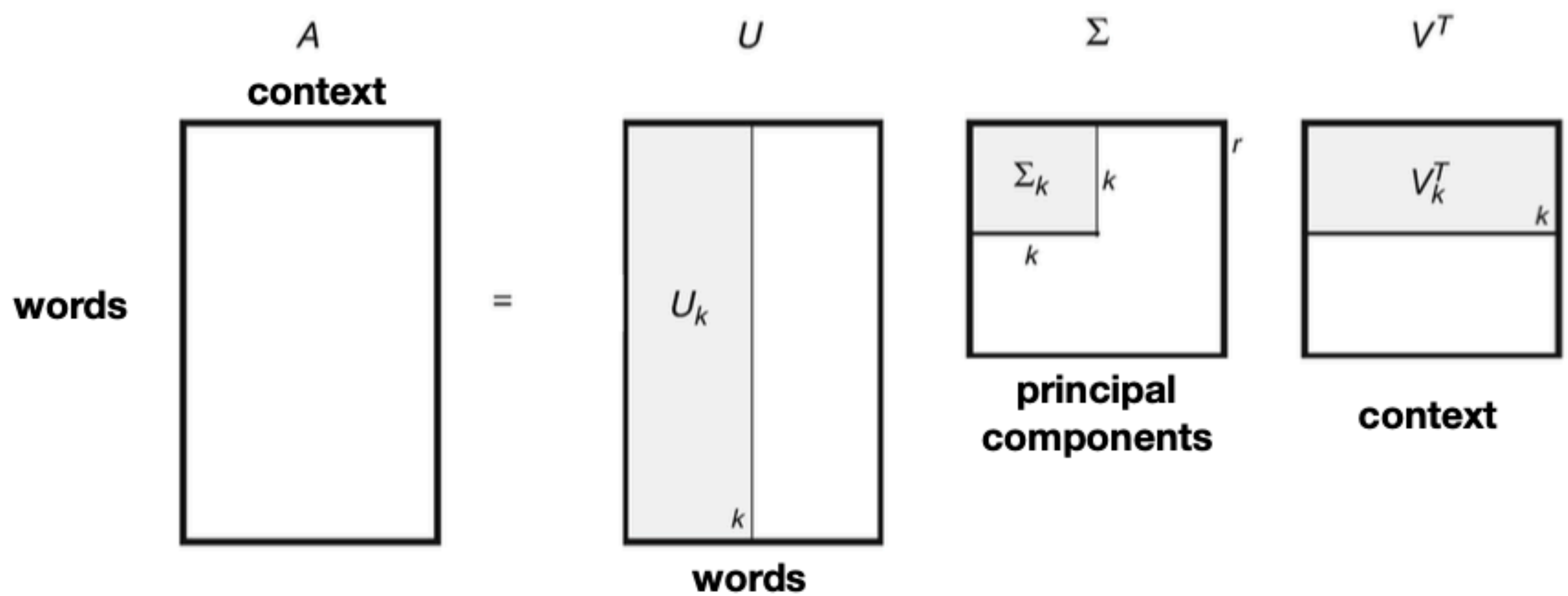
…government debt problems turning into	***banking***	*crises as happened in 2009…*
…saying that Europe needs unified	***banking***	*regulation to replace the hodgepodge…*
…India has just given its	***banking***	*system a shot in the arm…*

These context words will represent *banking*.

Latent Semantic Analysis
(SVD-based methods)

"context-counting" vectors

word	dimensions (context)		
	police	owner	food
cat	10	120	170
dog	30	100	200
kill	100	50	20
murder	120	45	15



(Deerwester et al, 1990): Indexing by latent semantic analysis

Collobert & Weston vectors

Idea: a word and its context is a positive training sample; a random word in that sample context gives a negative training sample:

+ cat chills **on** a mat − cat chills Ohio a mat

How do we formalize this idea? Ask that

score(cat chills on a mat) > score(cat chills Ohio a mat)

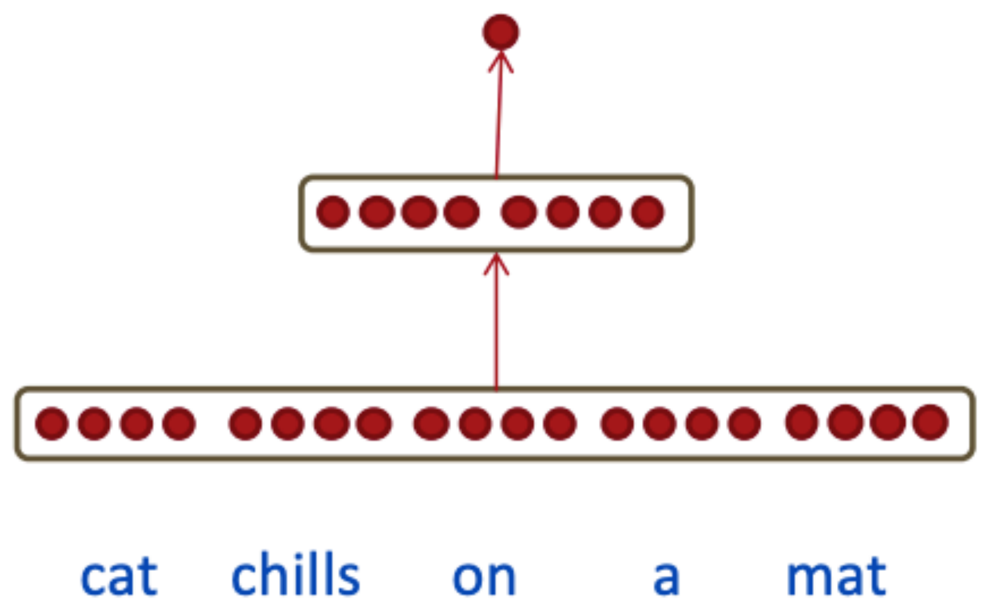
(Collobert et al, 2011) Natural Language Processing (Almost) from Scratch

Collobert & Weston vectors

$$
\begin{aligned}
s &= U^T a \\
a &= f(z) \\
z &= Wx + b
\end{aligned}
$$

$$x = [x_{cat} \quad x_{chills} \quad x_{on} \quad x_{a} \quad x_{mat}]$$

$$L \in \mathbb{R}^{n \times |V|}$$



s = score(cat chills on a mat)

s_c = score(cat chills Ohio a mat)

$$\boxed{J = \max(0, 1 - s + s_c)}$$

(Mikolov et al, 2013): Main Contributions

- An improved version of *skip-gram* algorithm
  - Negative sampling (vs hierarchical softmax in the earlier paper)
  - Subsampling of frequent words

- You can also learn good vector presentations for phrases!

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

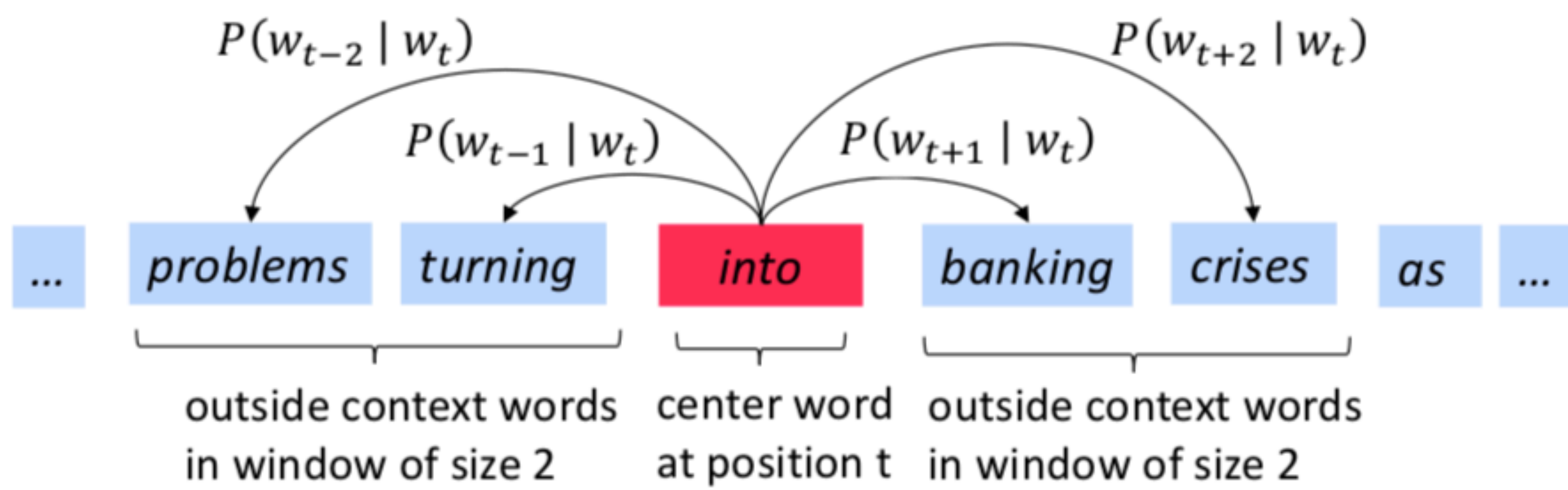
Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

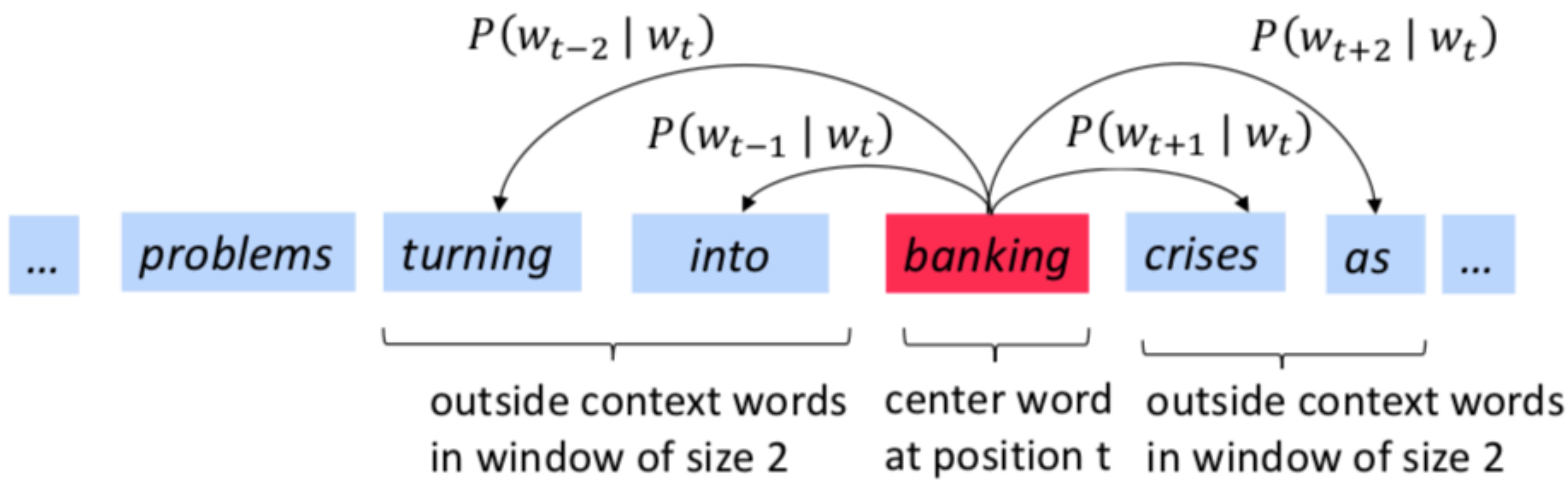
The Skip-gram model

- The idea: we want to use words to **predict** their context words
- Context: a fixed window of size $2m$



The Skip-gram model

- The idea: we want to use words to **predict** their context words
- Context: a fixed window of size $2m$



Skip-gram: objective function

- For each position $t = 1,2,\ldots T$, predict context words within context size m, given center word w_j:

all the parameters to be optimized

$$\mathcal{L}(\theta) = \prod_{t=1}^{T} \prod_{-m \le j \le m, j \ne 0} P(w_{t+j} \mid w_t; \theta)$$

- The objective function $J(\theta)$ is the (average) negative log likelihood:

$$J(\theta) = -\frac{1}{T} \log \mathcal{L}(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \le j \le m, j \ne 0} \log P(w_{t+j} \mid w_t; \theta)$$

How to define $P(w_{t+j} \mid w_t; \theta)$?

- We have two sets of vectors for each word in the vocabulary

$\mathbf{u}_i \in \mathbb{R}^d$: embedding for center word *i*

$\mathbf{v}_{i'} \in \mathbb{R}^d$: embedding for context word *i'*

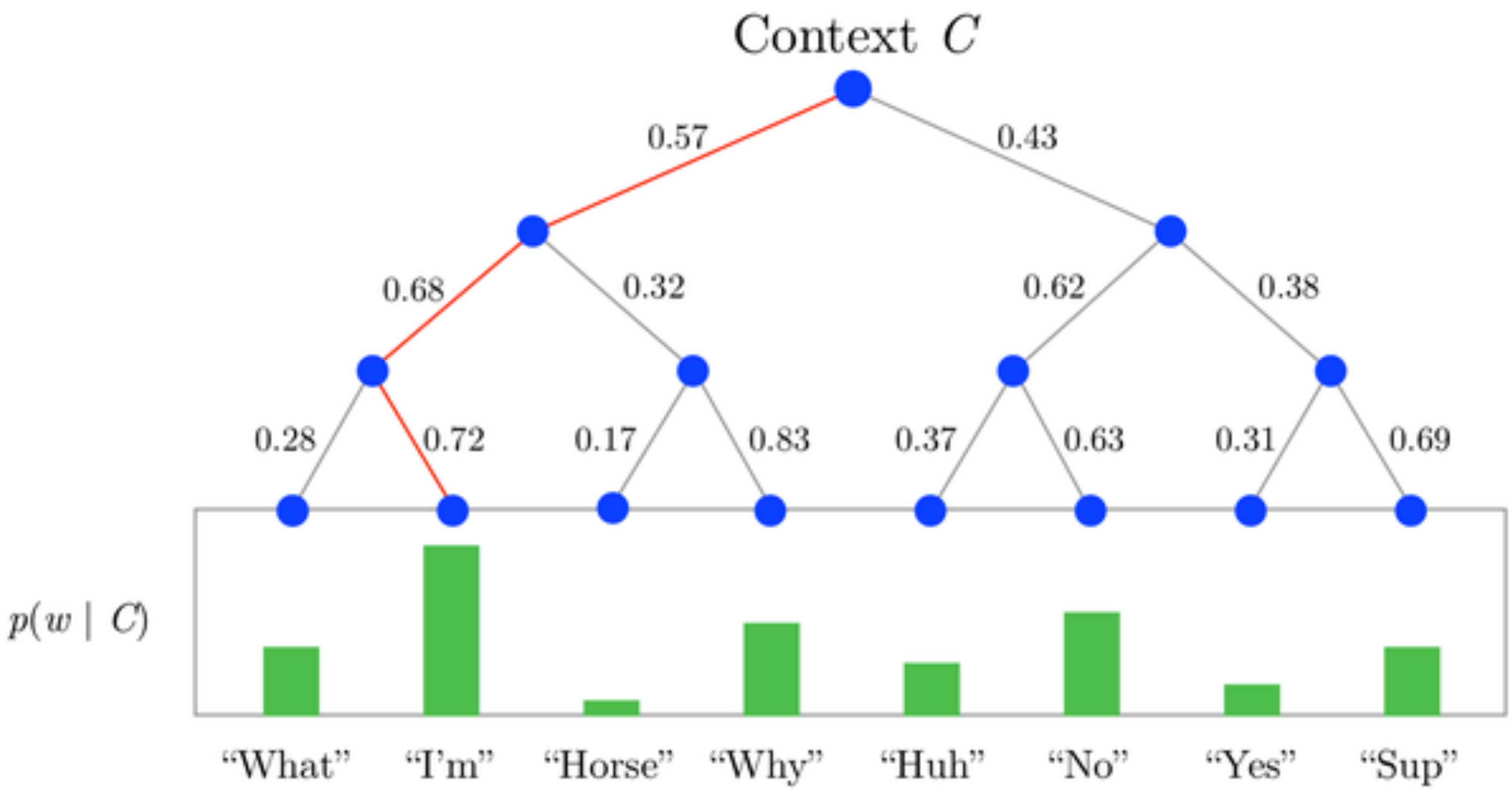
- Use inner product $\mathbf{u}_i \cdot \mathbf{v}_{i'}$ to measure how likely word *i* appears with context word *i'*, the larger the better

$$P(w_{t+j} \mid w_t) = \frac{\exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_{w_{t+j}})}{\sum_{k \in V} \exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_k)}$$

$\theta = \{\{\mathbf{u}_k\}, \{\mathbf{v}_k\}\}$ are all the parameters in this model!

V is large: 10^5-10^7. Computing probabilities is very expensive!

Hierarchical softmax



$$\boxed{p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w,j+1) = \mathrm{ch}(n(w,j))]\!] \cdot {v'_{n(w,j)}}^{\top} v_{w_I} \right)}$$

(Morin and Bengio et al, 2005) Hierarchical probabilistic neural language model

Hierarchical softmax

- Huffman tree:

word	count
fat	3
fridge	2
zebra	1
potato	3
and	14
in	7
today	4
kangaroo	2

$\longrightarrow$

Negative sampling

- SGNS = Skip-gram with negative sampling
- Intuition: for each (w, c) pair, we sample k negative pairs (w, c'):

$$P(D = 1 \mid w, c) = \frac{1}{1 + \exp(-\mathbf{u}_w \cdot \mathbf{v}_c)}$$

$$P(D = 0 \mid w, c') = \frac{\exp(-\mathbf{u}_w \cdot \mathbf{v}_{c'})}{1 + \exp(-\mathbf{u}_w \cdot \mathbf{v}_{c'})}$$

$$\boxed{\log \sigma({v'_{w_O}}^\top v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-{v'_{w_i}}^\top v_{w_I}) \right]}$$

$$P_n(w) = \frac{U(w)^{3/4}}{Z}$$

is: $0.9^{3/4} = 0.92$
Constitution: $0.09^{3/4} = 0.16$
bombastic: $0.01^{3/4} = 0.032$

Noise Contrastive Estimation (NCE)

- Recommended reading: **(Dyer, 2014) Notes on Noise Contrastive Estimation and Negative Sampling**

- “They are superficially similar, NCE is a general parameter estimation technique that is asymptotically unbiased, while negative sampling is best understood as a family of binary classification models that are useful for learning word representations but not as a general-purpose estimator.”

Hierarchical softmax vs Negative sampling

- Pros and Cons

Subsampling of Frequent Words

- Probability of discarding a word:

$$\boxed{P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}}$$

- $t = 10^{-5}$

Experimental setup

- Google dataset: 1 billion words
- Vocabulary size: 692K
- Context size: 5
- Dimension: 300
- "Our experiments indicate that values of k in the range 5–20 are useful for small training datasets, while for large datasets the k can be as small as 2–5."

- Pre-trained word vectors: 100 billion words, 300-dimensional vectors for 3 million words and phrases.

Evaluation: analogical reasoning

Word analogy

man: woman $\approx$ king: ?

$$\arg\max_i \left(\cos(\mathbf{u}_i, \mathbf{u}_b - \mathbf{u}_a + \mathbf{u}_c)\right)$$

semantic

Chicago:Illinois$\approx$Philadelphia: ?

syntactic

bad:worst $\approx$ cool: ?

More examples at

http://download.tensorflow.org/data/questions-words.txt

Evaluation: analogical reasoning

Method	Time [min]	Syntactic [%]	Semantic [%]	Total accuracy [%]
NEG-5	38	63	54	59
NEG-15	97	63	58	**61**
HS-Huffman	41	53	40	47
NCE-5	38	60	45	53
The following results use 10^{-5} subsampling				
NEG-5	14	61	58	60
NEG-15	36	61	61	**61**
HS-Huffman	21	52	59	55

Table 1: Accuracy of various Skip-gram 300-dimensional models on the analogical reasoning task as defined in [8]. NEG-k stands for Negative Sampling with k negative samples for each positive sample; NCE stands for Noise Contrastive Estimation and HS-Huffman stands for the Hierarchical Softmax with the frequency-based Huffman codes.

No word similarity evaluation!

Learning phrases

- "New York Times" != "New" + "York" + "Times"
- "Air Canada" != "Air" + "Canada"

- A simple data-driven approach to select phrases:

$$\text{score}(w_i, w_j) = \frac{\text{count}(w_i w_j) - \delta}{\text{count}(w_i) \times \text{count}(w_j)}.$$

Evaluation: analogical reasoning for phrases

Newspapers			
New York	New York Times	Baltimore	Baltimore Sun
San Jose	San Jose Mercury News	Cincinnati	Cincinnati Enquirer
NHL Teams			
Boston	Boston Bruins	Montreal	Montreal Canadiens
Phoenix	Phoenix Coyotes	Nashville	Nashville Predators
NBA Teams			
Detroit	Detroit Pistons	Toronto	Toronto Raptors
Oakland	Golden State Warriors	Memphis	Memphis Grizzlies
Airlines			
Austria	Austrian Airlines	Spain	Spainair
Belgium	Brussels Airlines	Greece	Aegean Airlines
Company executives			
Steve Ballmer	Microsoft	Larry Page	Google
Samuel J. Palmisano	IBM	Werner Vogels	Amazon

Table 2: Examples of the analogical reasoning task for phrases (the full test set has 3218 examples). The goal is to compute the fourth phrase using the first three. Our best model achieved an accuracy of 72% on this dataset.

Evaluation: analogical reasoning for phrases

Method	Dimensionality	No subsampling [%]	10^{-5} subsampling [%]
NEG-5	300	24	27
NEG-15	300	27	42
HS-Huffman	300	19	**47**

Table 3: Accuracies of the Skip-gram models on the phrase analogy dataset. The models were trained on approximately one billion words from the news dataset.

Comparison to previous models

Model (training time)	Redmond	Havel	ninjutsu	graffiti	capitulate
Collobert (50d) (2 months)	conyers lubbock keene	plauen dzerzhinsky osterreich	reiki kohona karate	cheesecake gossip dioramas	abdicate accede rearm
Turian (200d) (few weeks)	McCarthy Alston Cousins	Jewell Arzu Ovitz	- - -	gunfire emotion impunity	- - -
Mnih (100d) (7 days)	Podhurst Harlang Agarwal	Pontiff Pinochet Rodionov	- - -	anaesthetics monkeys Jews	Mavericks planning hesitated
Skip-Phrase (1000d, 1 day)	Redmond Wash. Redmond Washington Microsoft	Vaclav Havel president Vaclav Havel Velvet Revolution	ninja martial arts swordsmanship	spray paint grafitti taggers	capitulation capitulated capitulating

Table 6: Examples of the closest tokens given various well known models and the Skip-gram model trained on phrases using over 30 billion training words. An empty cell means that the word was not in the vocabulary.

No quantitative evaluation!

No downstream evaluation

What is good about word2vec?

- Discussion
- .. vs Collobert & Weston?

Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors

Marco Baroni and **Georgiana Dinu** and **Germán Kruszewski**
Center for Mind/Brain Sciences (University of Trento, Italy)
(marco.baroni|georgiana.dinu|german.kruszewski)@unitn.it

Main contributions

- A systematic comparative evaluation of count and predict vectors.
- Main result: predict vectors >> count vectors.

Motivation: these silly deep learning people keep writing papers but don't compare to traditional distributional semantics models. So we will.

Conclusion: okay, those people are actually right.

Count vs predict models

- “Count” models: collect raw co-occurrence counts in a corpus, and transform them into vectors with dimensionality reduction (and reweighting)

- “Predict” models: estimate the word vectors directly by maximizing the probability of the contexts in which the word is observed in the corpus

Experimental setup

- Corpus: **2.8 billion** tokens (ukWaC, English Wikipedia, British National Corpus)
- Vocabulary: **300k** most frequent words.

- Count models
  - Context size: 2 or 5
  - Two weighting schemes: positive Pointwise Mutual information (PPMI), Local Mutual Information
  - SVD, two other non-negative matrix factorization methods
  - Dimensions: 200, 300, 400, 500

$$\mathrm{PPMI}(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P(c)}, 0)$$

Experimental setup

- Predict models: CBOW
  - Dimensions: 200, 300, 400, 500
  - Context size: 2, 5
  - Hierarchical softmax and negative sampling (k = 5 or 10)
  - Subsampling $t = 1e^{-5}$

- Out-of-the-box models
  - (Baroni and Lenci, 2010): count models relying on syntactic information
  - Collobert & Weston vectors

Benchmarks

name	task	measure	source	soa
rg	relatedness	Pearson	Rubenstein and Goodenough (1965)	Hassan and Mihalcea (2011)
ws	relatedness	Spearman	Finkelstein et al. (2002)	Halawi et al. (2012)
wss	relatedness	Spearman	Agirre et al. (2009)	Agirre et al. (2009)
wsr	relatedness	Spearman	Agirre et al. (2009)	Agirre et al. (2009)
men	relatedness	Spearman	Bruni et al. (2014)	Bruni et al. (2014)
toefl	synonyms	accuracy	Landauer and Dumais (1997)	Bullinaria and Levy (2012)
ap	categorization	purity	Almuhareb (2006)	Rothenhäusler and Schütze (2009)
esslli	categorization	purity	Baroni et al. (2008)	Katrenko and Adriaans (2008)
battig	categorization	purity	Baroni et al. (2010)	Baroni and Lenci (2010)
up	sel pref	Spearman	Padó (2007)	Herdağdelen and Baroni (2009)
mcrae	sel pref	Spearman	McRae et al. (1998)	Baroni and Lenci (2010)
an	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013c)
ansyn	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013a)
ansem	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013c)

Table 1: Benchmarks used in experiments, with type of task, figure of merit (measure), original reference (source) and reference to current state-of-the-art system (soa).

Semantic relatedness

- Compare the correlation between the average scores that human subjects assigned to the pairs and cosine similarity between corresponding vectors.
- Similarity vs relatedness: “car” vs “vechicle” AND “car” vs “journey”

Word 1	Word 2	Human (mean)
tiger	cat	7.35
tiger	tiger	10
book	paper	7.46
computer	internet	7.58
plane	car	5.77
professor	doctor	6.62
stock	phone	1.62
stock	CD	1.31
stock	jaguar	0.92

Metric: Spearman rank correlation

Synonym detection

TOEFL test

- levied: imposed, believed, requested, correlated

Concept categorization

- “helicopters” “motorcycles”
- “elaphants” “mammal”

Selectional preferences

- Verb-noun pairs
- *People* received a high average score as **subject** of *to eat*, and a low score as **object** of the same verb.

Final performance

	rg	ws	wss	wsr	men	toefl	ap	esslli	battig	up	mcrae	an	ansyn	ansem
							best setup on each task							
cnt	74	62	70	59	72	76	66	84	98	41	27	49	43	60
pre	84	75	**80**	**70**	**80**	91	75	86	**99**	41	28	**68**	**71**	**66**
							best setup across tasks							
cnt	70	62	70	57	72	76	64	84	98	37	27	43	41	44
pre	83	73	78	68	**80**	86	71	77	98	41	26	67	69	64
							worst setup across tasks							
cnt	11	16	23	4	21	49	24	43	38	-6	-10	1	0	1
pre	74	60	73	48	68	71	65	82	88	33	20	27	40	10
							best setup on rg							
cnt	(74)	59	66	52	71	64	64	84	98	37	20	35	42	26
pre	(84)	71	76	64	79	85	72	84	98	39	25	66	70	61
							other models							
soa	**86**	**81**	77	62	76	**100**	**79**	**91**	96	**60**	**32**	61	64	61
dm	82	35	60	13	42	77	76	84	94	51	29	NA	NA	NA
cw	48	48	61	38	57	56	58	61	70	28	15	11	12	9

Table 2: Performance of count (cnt), predict (pre), dm and cw models on all tasks. See Section 3 and Table 1 for figures of merit and state-of-the-art results (soa). Since dm has very low coverage of the an* data sets, we do not report its performance there.

Top count models

window	weight	compress	dim.	mean rank
2	PMI	no	300K	35
5	PMI	no	300K	38
2	PMI	SVD	500	42
2	PMI	SVD	400	46
5	PMI	SVD	500	47
2	PMI	SVD	300	50
5	PMI	SVD	400	51
2	PMI	NMF	300	52
2	PMI	NMF	400	53
5	PMI	SVD	300	53

Top predict models

win.	hier. softm.	neg. samp.	subsamp.	dim	mean rank
5	no	10	yes	400	10
2	no	10	yes	300	13
5	no	5	yes	400	13
5	no	5	yes	300	13
5	no	10	yes	300	13
2	no	10	yes	400	13
2	no	5	yes	400	15
5	no	10	yes	200	15
2	no	10	yes	500	15
2	no	5	yes	300	16

Recommended reading

- **(Dyer, 2014) Notes on Noise Contrastive Estimation and Negative Sampling**
- **(Pennington et al, 2014) GloVe: Global Vectors for Word Representation**
- **(Levy et al, 2015): Improving Distributional Similarity with Lessons Learned from Word Embeddings**
    - "We reveal that much of the performance gains of word embeddings are due to certain system design choices and hyperparameter optimizations, rather than the embedding algorithms themselves."