Vector space models of meaning in natural language processing

Georgiana Dinu
IBM T.J. Watson Research Center
gdinu@us.ibm.com

ESSENCE 2015

Part of the slides adapted from earlier materials prepared in collaboration with Marco Baroni

http://clic.cimec.unitn.it/marco/

Outline

- Introduction to distributional semantics
- Distributed meaning representations
- Word meaning representations in NLP tasks

Break

- Compositional distributional semantics
- Beyond sentence similarity
    - Decomposition, plausibility, morphology
    - Cross-lingual and cross-modal applications

Distributional semantics

Distributional semantics

Distributional hypothesis: Words that occur in similar contexts have similar meanings.

We found a little hairy wampimuk sleeping behind the tree.

Co-occurrence to meaning

```
he curtains open and the moon shining in on the barely
ars and the cold , close moon " . And neither of the w
rough the night with the moon shining so brightly , it
made in the light of the moon . It all boils down , wr
 surely under a crescent moon , thrilled by ice-white
sun , the seasons of the moon ? Home , alone , Jay pla
m is dazzling snow , the moon has risen full and cold
un and the temple of the moon , driving out of the hug
 in the dark and now the moon rises , full and amber a
bird on the shape of the moon over the trees in front
 But I could n't see the moon or the stars , only the
rning , with a sliver of moon hanging among the stars
 they love the sun , the moon and the stars . None of
the light of an enormous moon . The plash of flowing w
man 's first step on the moon ; various exhibits , aer
 the inevitable piece of moon rock . Housing The Airsh
oud obscured part of the moon . The Allied guns behind
```

Distributional semantics in a nutshell

1. Represent words through vectors recording their co-occurrence counts with context elements in a corpus

Optional

Apply a re-weighting scheme to the results co-occurrence matrix

Optional

Apply dimensionality reduction to the co-occurrence matrix

4. Measure geometric distance of word vectors in “distributional space” as proxy to semantic similarity/ relatedness

1. Co-occurrence

	Doc1	Doc2	Doc3
• stars	38	45	2

	● $\xleftarrow{\text{dobj}}$ see	● $\xrightarrow{\text{mod}}$ bright	● $\xrightarrow{\text{mod}}$ shiny
• stars	38	45	44

	The nearest ● to Earth	stories of ● and their
• stars	12	10

Occurrence in document

… in syntactic relations

… in context pattern

Co-occurrence. More generally

Variation in ways to collect co-occurrence counts

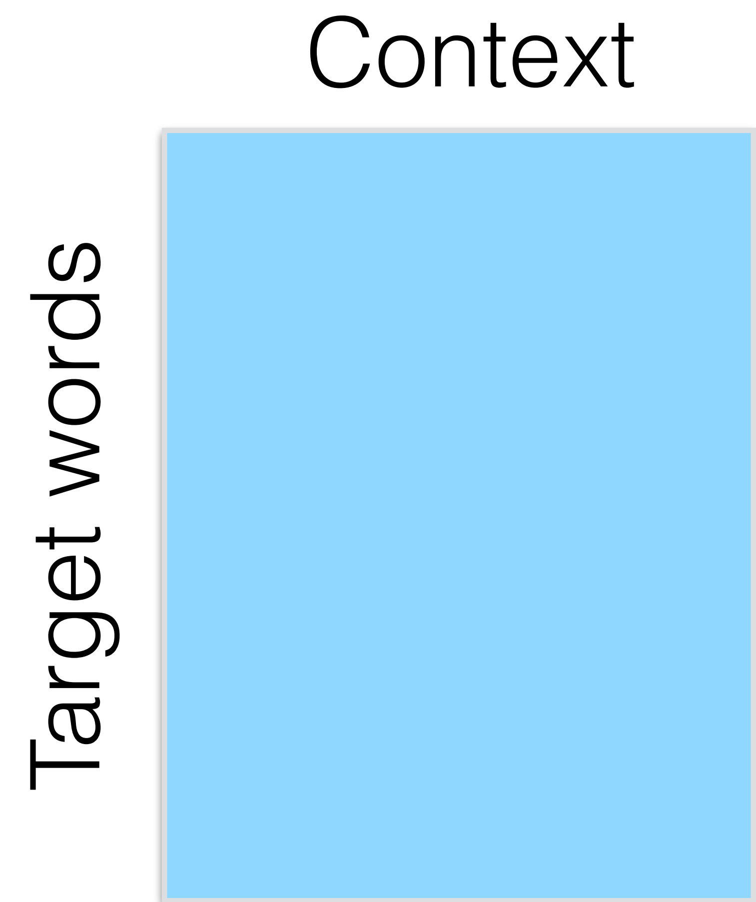
- E.g. co-occurrence with words, window of size 2, scaling by distance to target word

… two [intensely bright stars in the] night sky …

	intensely	bright	in	the
stars	0.5	1	1	0.5

Co-occurrence matrix

	…	bright	in	sky	…
stars	…	8	10	6	…
sun	…	10	15	4	…
hyrax	…	0	20	1	…



2. Re-weighting

1. Extract co-occurrence counts

2. Apply a re-weighting scheme on the resulting co-occurrence matrix

Re-weigh the counts using corpus-level statistics to reflect co-occurrence significance.

Point-wise mutual information (PMI)

$$\mathrm{PMI}(target, ctxt) = \log \frac{\mathrm{P}(target, ctxt)}{\mathrm{P}(target)\mathrm{P}(ctxt)}$$

	…	bright	in	sky	…
stars	…	80	300	61	…
stars	…	3.1	1.2	2.4	…

Raw counts

PMI scores

- Other weighting schemes:
  - Tf-idf, Local mutual information, Log-Likelihood Ratio

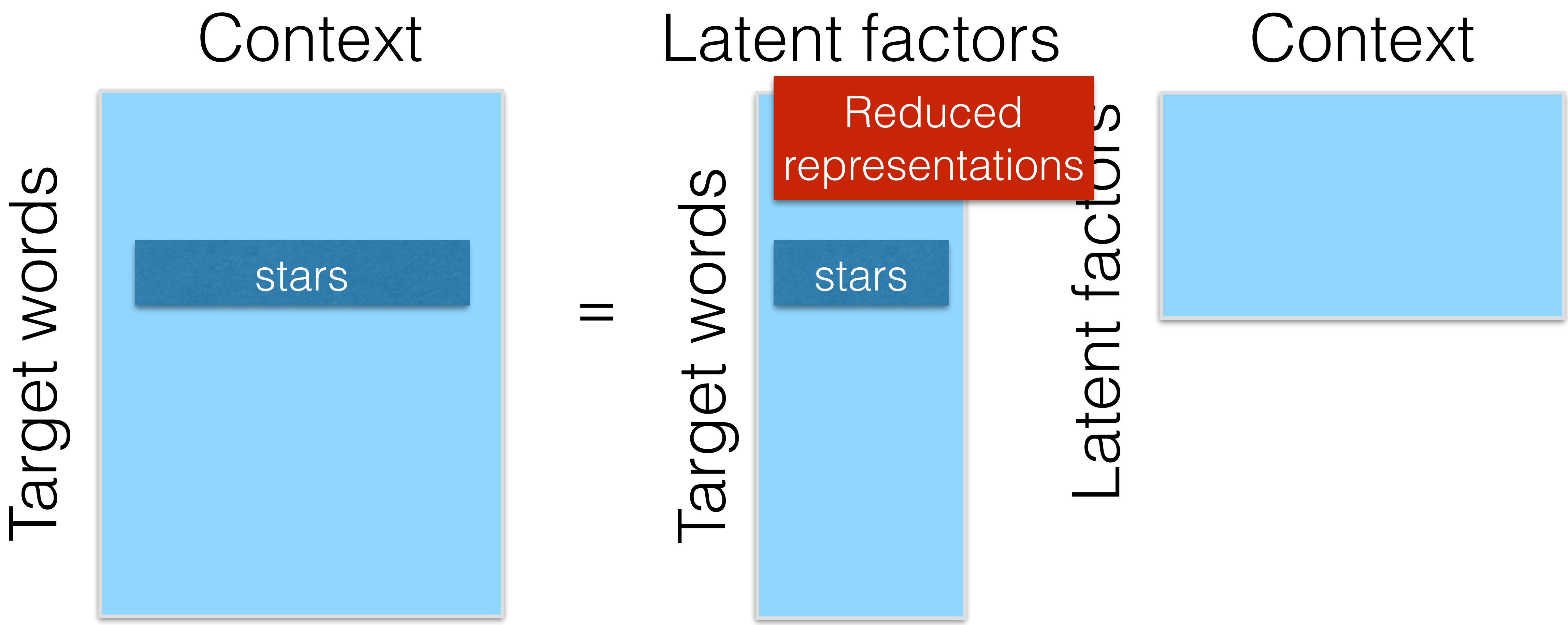
3. Dimensionality reduction

1. Extract co-occurrence counts
2. Apply a re-weighting scheme on the resulting co-occurrence matrix
3. Apply dimensionality reduction

- Vector spaces often range from tens of thousands to millions of context dimensions
- Some dimensionality reduction methods:
  - Select contexts based on various relevance criteria

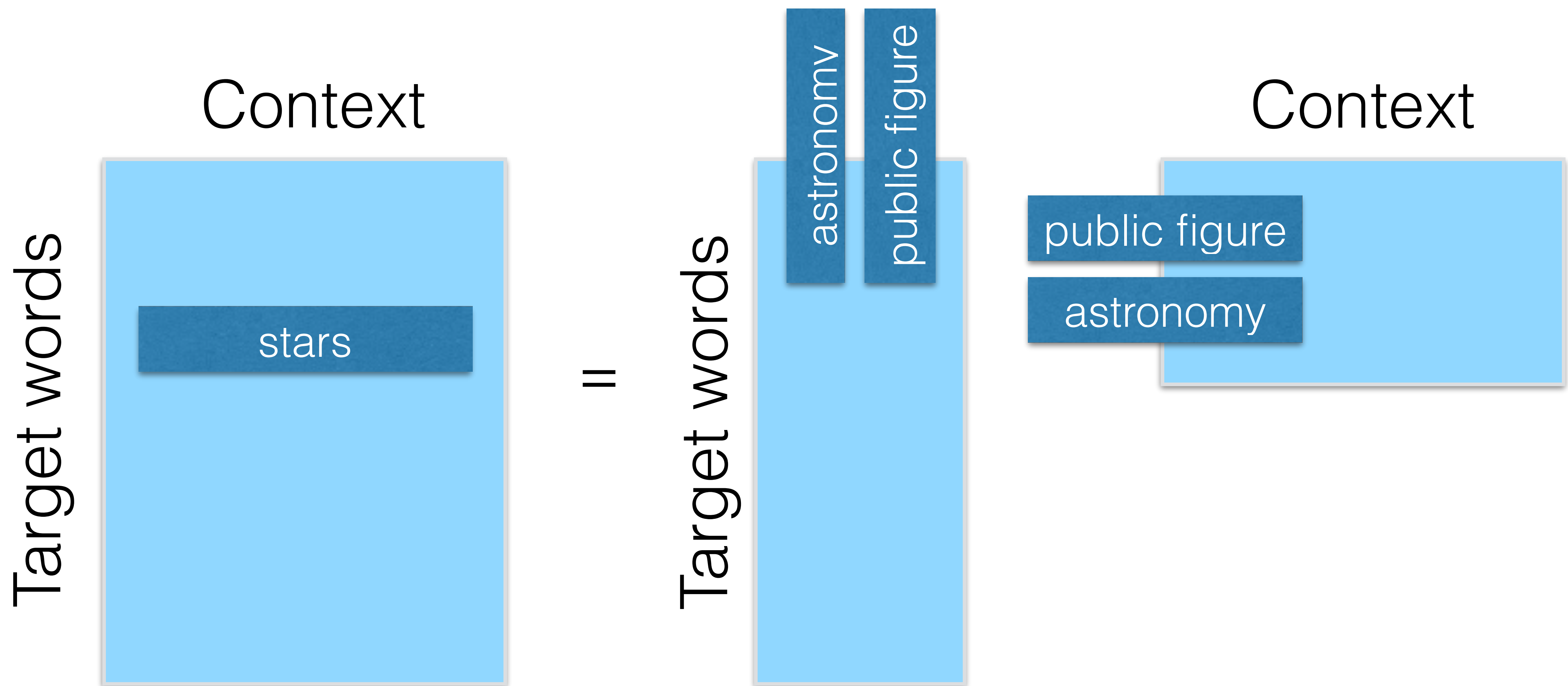
  - Having also a beneficial *smoothing* effect: Singular Value Decomposition, Probabilistic Latent Semantic Analysis, Latent Dirichlet Allocation

Dimensionality reduction



Factorize the co-occurrence counts as linear combinations over latent factors

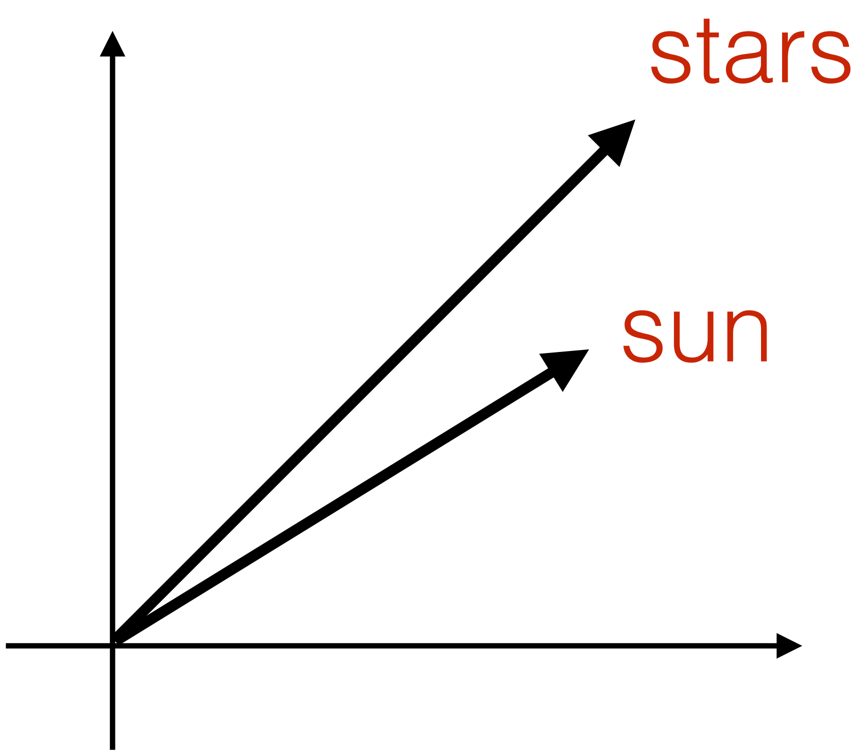
Dimensionality reduction



Latent factors can be more (e.g. topic models) or less (e.g. SVD) interpretable.

From vectors to similarity in meaning

1. Extract co-occurrence counts
2. Apply a re-weighting scheme on the resulting co-occurrence matrix
3. Apply dimensionality reduction
4. Vector similarity



Cosine similarity

$$cos(\vec{u}, \vec{v}) = \frac{\Sigma_i u_i v_i}{\sqrt{\Sigma_i u_i^2}\sqrt{\Sigma_i v_i^2}}$$

$$= \frac{< u, v >}{||u|| \times ||v||}$$

Other similarity measures: Euclidean, Lin

Semantic neighbours of words

rhino	fall	good
woodpecker	rise	bad
rhinoceros	increase	excellent
swan	fluctuation	superb
whale	drop	poor
ivory	decrease	improved
plover	reduction	perfect
elephant	logarithm	clever
bear	decline	terrific

http://clic.cimec.unitn.it/infomap-query/

Semantic neighbours of phrases

DIRT - Lin and Pantel, 2007

X's addiction to Y

Cosmos

N:gen:N<addiction>N:to:N

1 N:gen:N<**addiction**>N:nn:N
2 N:gen:N<**craving**>N:for:N
3 N:gen:N<**child**>N:about:N
4 N:gen:N<**money**<N:obj:V<**spend**>V:on:N
5 N:gen:N<**intake**>N:nn:N
6 N:gen:N<**zest**>N:for:N
7 N:gen:N<**winning**>N:nn:N
8 N:gen:N<**use**>N:nn:N
9 N:gen:N<**habit**>N:nn:N

X manufactures Y

Cosmos

N:subj:V<manufacture>V:obj:N

1 N:by:V<**manufacture**>V:obj:N
2 N:obj:V<**manufacture**>V:subj:N
3 N:subj:V<**produce**>V:obj:N
4 N:subj:V<**begin**>V:obj:N>**production**>N:of:N
5 N:subj:V<**export**>V:obj:N
6 N:subj:N<**supplier**>N:of:N
7 N:subj:V<**supply**>V:obj:N
8 N:subj:V<**sell**>V:obj:N
9 N:appo:N<**manufacturer**>N:nn:N

http://demo.patrickpantel.com/demos/lexsem/paraphrase.htm

General-purpose representations of meaning

- Synonymy
- Relatedness
- Concept categorization
- Selectional preferences
- Analogy
- Relation classification
- …

Similarity/relatedness

- WordSim-353, SimLex-999, MEN

chapel	church	0.45
eat	strawberry	0.33
jump	salad	0.06
bikini	pizza	0.01

- Evaluation: Correlation of model cosines with human similarity assessments (close to human performance on relatedness, difficulties on synonym detection)

Selectional preferences

- Pado 2007

eat	villager	obj	1.7
eat	pizza	obj	6.8
eat	pizza	subj	1.1

- Evaluation: Create prototype argument vector (average all OBJ vectors of *eat*), compute similarity of prototype with candidate argument (*pizza*)

Categorization

- ESSLLI 2008 Shared task, Almuhareb and Poesio 2006

VEHICLE	MAMMAL
helicopter	dog
motorcycle	elephant
car	cat

- Evaluation: Cluster word vectors, overlap between clusters and gold categories (close to 90% cluster purity with 6 categories)

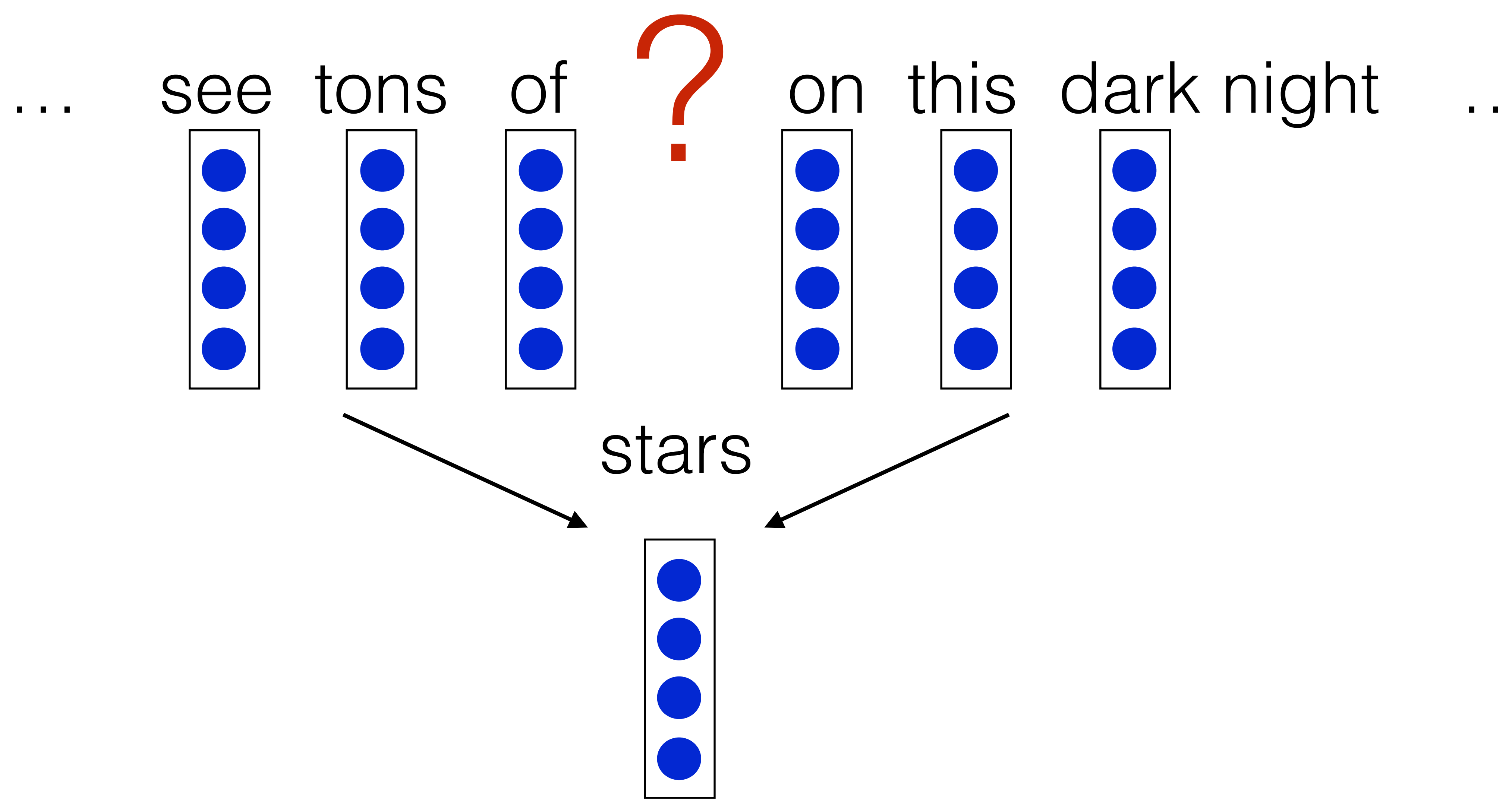
Distributional semantics: some references

- Overviews
  - Turney and Pantel 2010, Pado and Lapata 2007, Erk 2012, Baroni, Bernardi, Zamparelli - Frege in Space 2014
- Comparisons/evaluation
  - Agirre et al, 2009, Baroni and Lenci 2010, Bullinaria and Levy 2007, Bullinaria and Levi2012, Sahlgren 2006, Kiela et al 2014

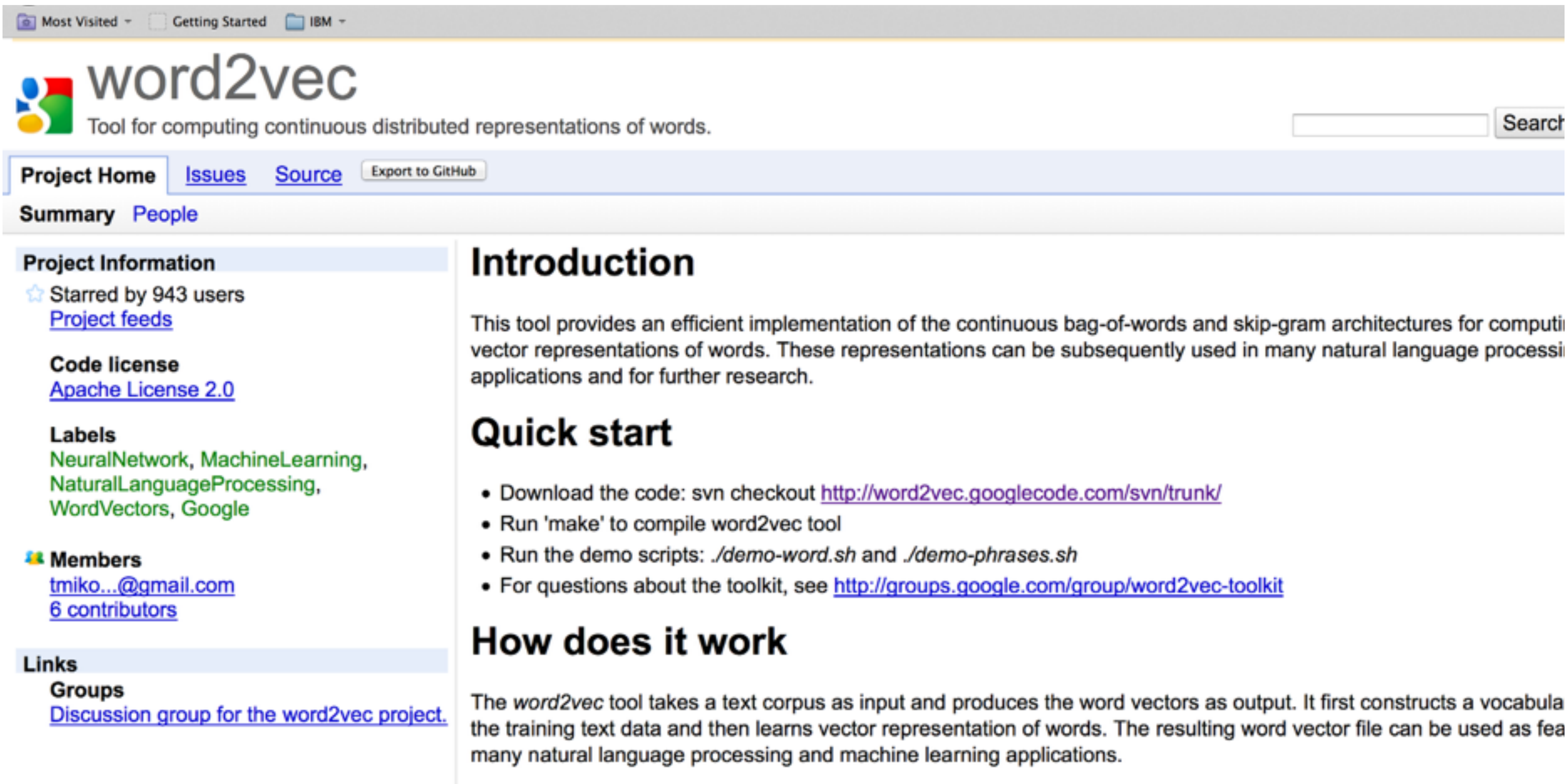
Other methods to obtain vectors?

Context-predicting objectives (a.k.a. distribut**ed** representations, embeddings)

Learn vector representations optimizing a context-prediction objective



word2vec



Most Visited Getting Started IBM

word2vec

Tool for computing continuous distributed representations of words.

Search

Project Home Issues Source Export to GitHub

Summary People

Project Information

Starred by 943 users

Project feeds

Code license

Apache License 2.0

Labels

NeuralNetwork, MachineLearning, NaturalLanguageProcessing, WordVectors, Google

Members

tmiko...@gmail.com

6 contributors

Links

Groups

Discussion group for the word2vec project.

Introduction

This tool provides an efficient implementation of the continuous bag-of-words and skip-gram architectures for computi vector representations of words. These representations can be subsequently used in many natural language processi applications and for further research.

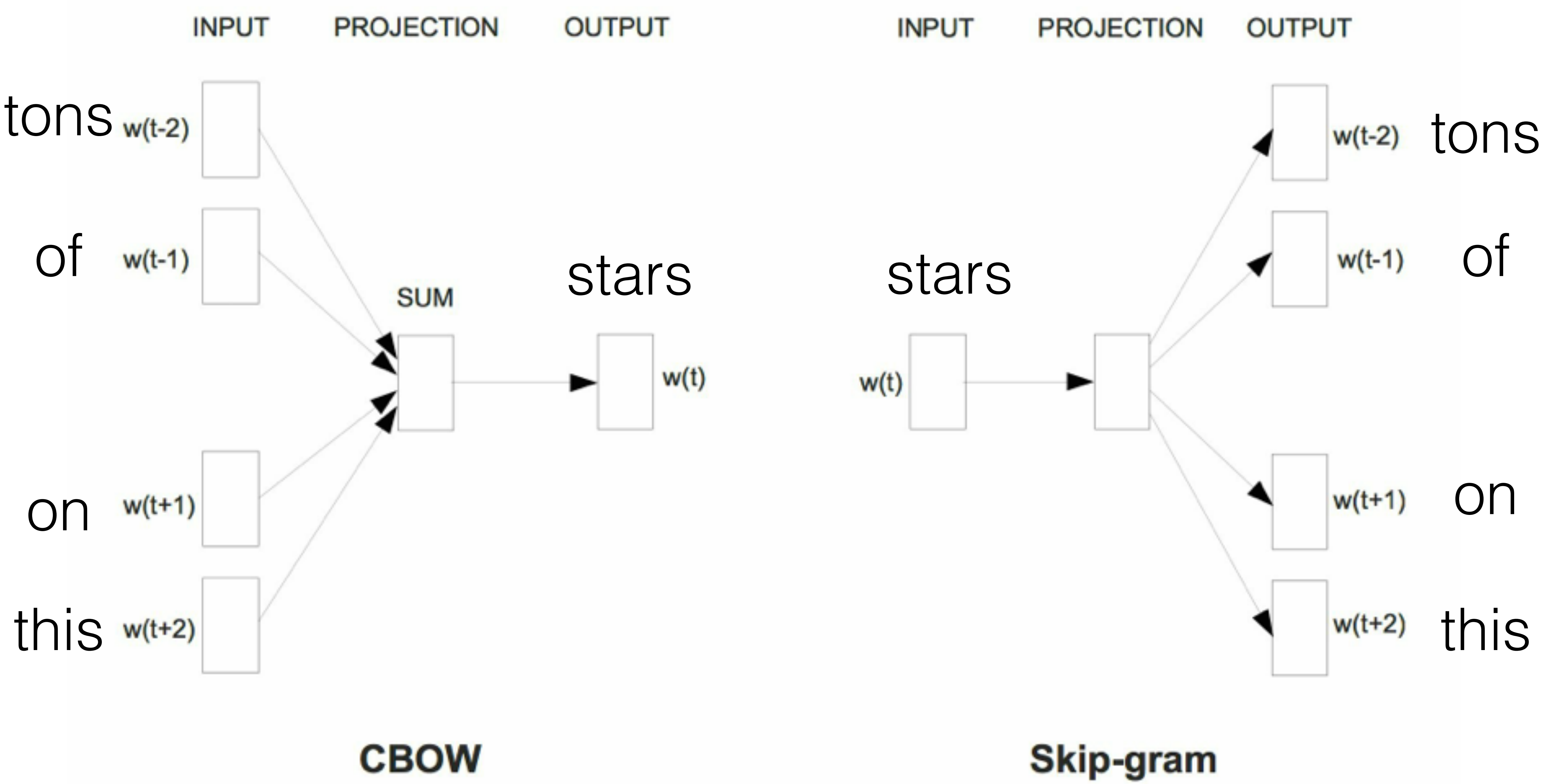
Quick start

- Download the code: svn checkout http://word2vec.googlecode.com/svn/trunk/
- Run 'make' to compile word2vec tool
- Run the demo scripts: *./demo-word.sh* and *./demo-phrases.sh*
- For questions about the toolkit, see http://groups.google.com/group/word2vec-toolkit

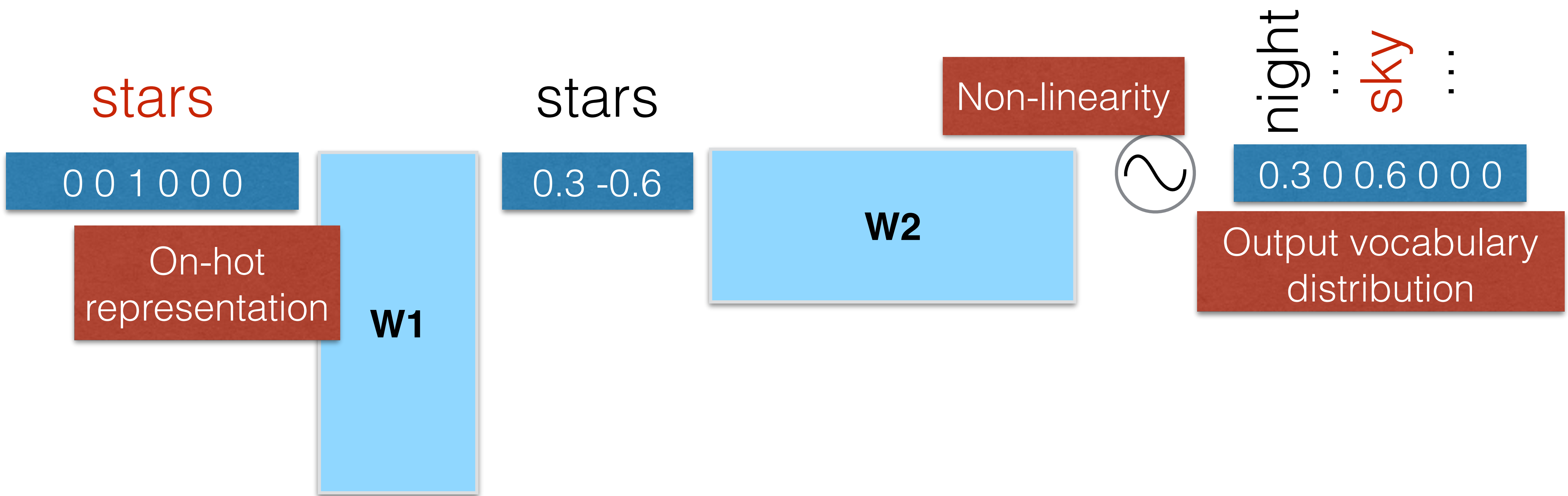
How does it work

The *word2vec* tool takes a text corpus as input and produces the word vectors as output. It first constructs a vocabula the training text data and then learns vector representation of words. The resulting word vector file can be used as fea many natural language processing and machine learning applications.

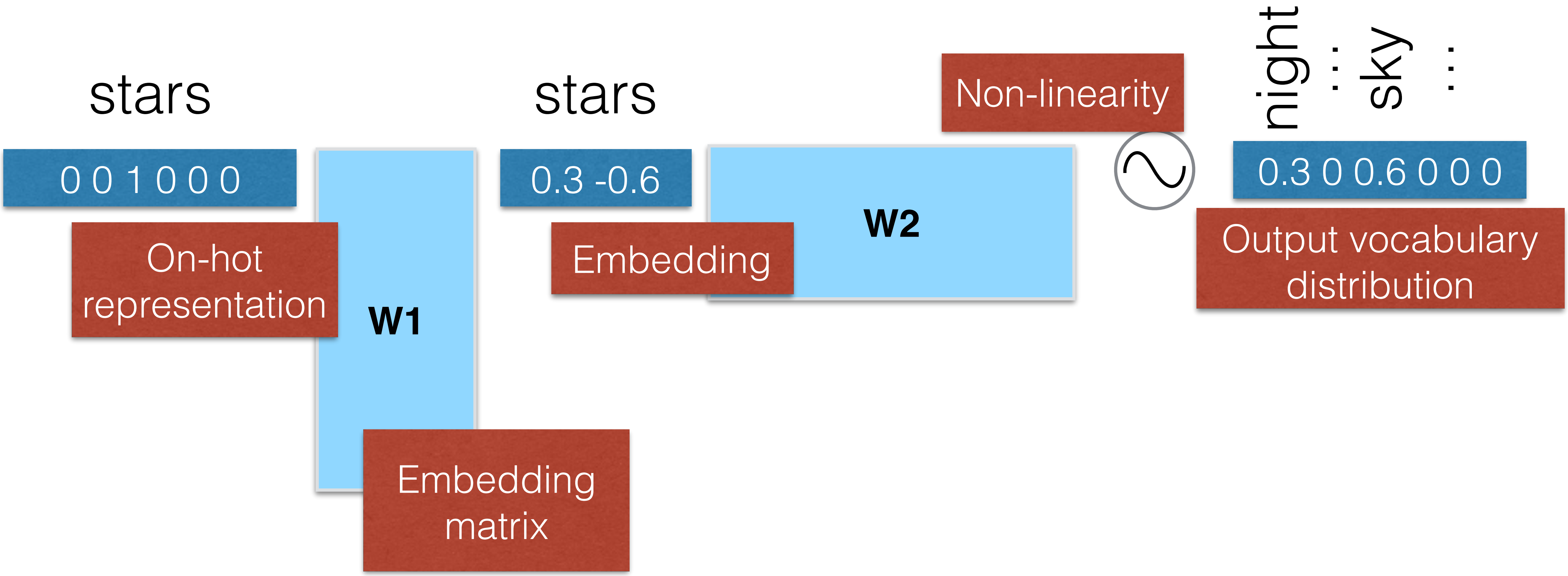
word2vec architectures



Skip-gram - In more detail



Skip-gram - In more detail



$$\frac{1}{T}\sum_{1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log P(w_{t+j}|w_t)$$

- Trained with stochastic gradient descent (parameters: W1, W2)
- Weigh context by distance to target, subsample frequent words

Distribut*ed* vs. distribution*al*

- Objective of skip-gram very similar to factorizing a co-occurrence matrix with PMI weighting (Into W1 and W2 matrices)(Levy and Goldberg 2014)

- However, word2vec has some advantages:
  - easy to use (takes a corpus of line-separated sentences as input)
  - fast (billions of tokens in up to several hours)
  - no need to explicitly compute and store large count matrices

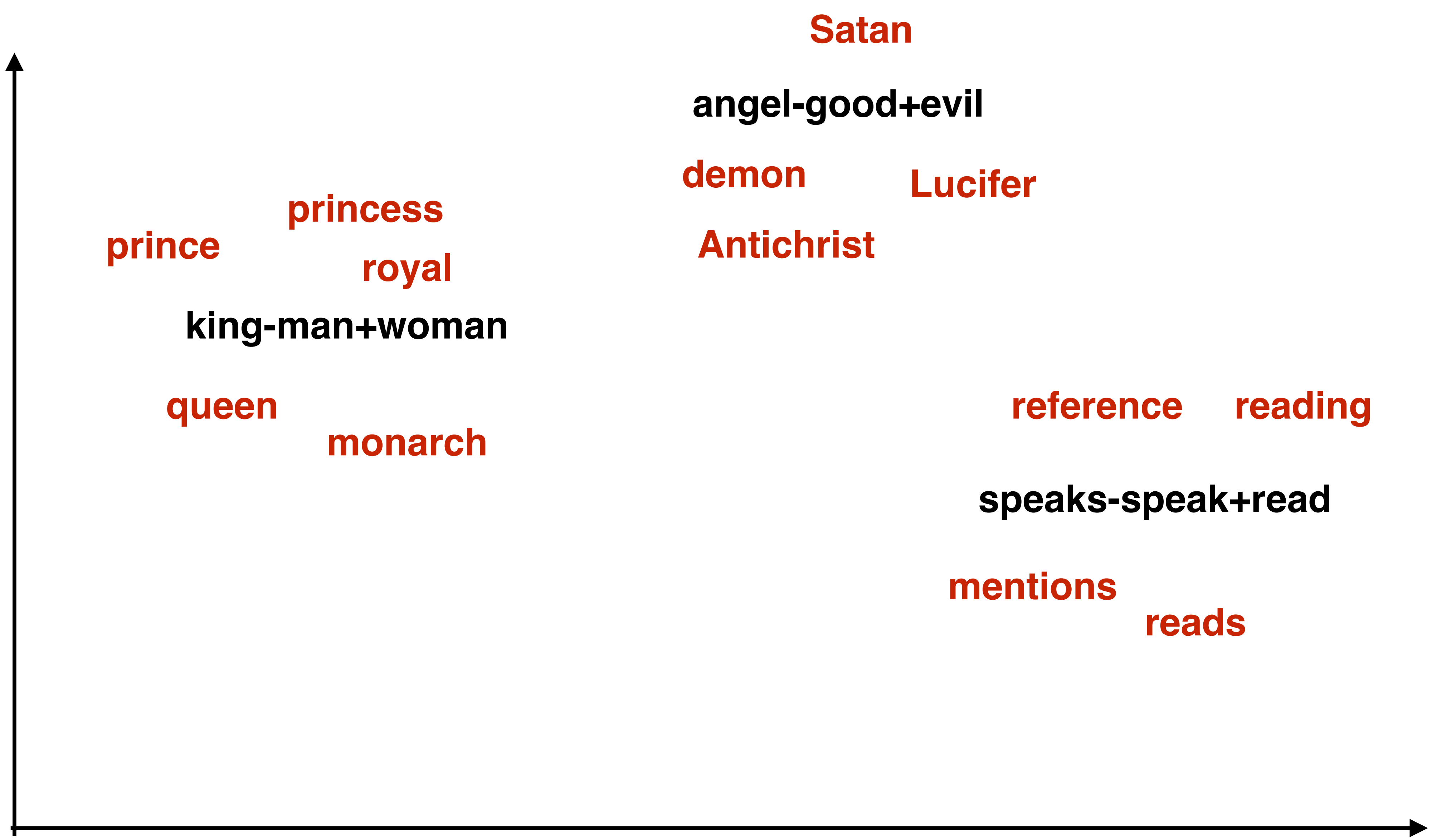
And, it can do:

king-man+woman = queen

Analogy data-set (Mikolov et al 2013):
~20K syntactic and semantic questions

a	b	c	d
man	woman	king	?
Spain	Madrid	France	?
good	better	rough	?
see	sees	return	?

Vector arithmetic



Evaluation:

- return the nearest neighbor of c-a+b (in the entire vocabulary)
- ~70% Top 1 accuracy with 300K vocabulary

Vector arithmetic

sushi-Japan+Italy

gelato

How does it compare to traditional distribution*al* approaches?

word2vec vs. distributional models on similarity benchmarks

	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

- State-of-the art performance in many similarity benchmarks (From Baroni et al, 2014)

References

Distribution-al/-ed (count/predict) comparisons

- Huang et al 2012, Blacoe and Lapata 2013, Baroni et al 2014

Before word2vec

- Neural network language models (predict the *next* word given the history):
  - Bengio et al 2003, 2006
  - Collobert and Weston 2008
  - Mikolov et al 2010, 2011
  - …

Distributional/distributed representations

- Robust, knowledge-lean methods
- Used in applications that require word similarity computations (thesaurus construction, question answering, information retrieval, machine translation)
- Word vectors used *directly* as features in various NLP tasks (parsing, part of speech tagging, information extraction tasks)

Information extraction

Extract information from unstructured text (named entity recognition, mention detection, co-reference, relation extraction)

Jethro Exum Sumner (c. 1733 – c. March 18, 1785) was a North Carolina landowner and businessman, and an officer in the Continental Army during the American Revolutionary War. Born in Virginia, Sumner's military service began in the French and Indian War as a member of the state's Provincial forces. After the conclusion of that conflict, he moved to Bute County, North Carolina, where he acquired a substantial area of land and operated a tavern. He served as Sheriff of Bute County, but with the coming of the American Revolution, he became a strident Patriot, and was elected to North Carolina's Provincial Congress.

Sumner was named the commanding officer of the 3rd North Carolina Regiment of the North Carolina Line, a formation of the Continental Army, in 1776, and served in both the Southern theater and Philadelphia campaign. He was one of five brigadier generals from North Carolina in the Continental Army, in which capacity he served between 1779 and 1783. He served with distinction in the battles of Stono Ferry and Eutaw Springs, but recurring bouts of poor health often forced him to play an administrative role, or to convalesce in North Carolina. Following a drastic reduction in the number of North Carolinians serving with the Continental Army, Sumner became a general in the state's militia but resigned in protest after the North Carolina Board of War awarded overall command of the militia to William Smallwood, a Continental Army general from Maryland. After the end of the war in 1783, Sumner helped to establish the North Carolina Chapter of the Society of the Cincinnati, and became its first president. He died in 1785 with extensive landholdings and 35 slaves.

Sumner was born in Nansemond County, Virginia, in 1733 to Jethro and Margaret Sullivan Sumner. His family had originally settled in Nansemond County in 1691.[1] Between 1758 and 1761, during the French and Indian War, he was a lieutenant in the Virginia Provincial forces in under the force

Mention detection

Mrs. Chisholm, an actress who suffered a nervous breakdown

PERSON PERSON DISEASE

and married a dull guy from Dayton , warns Gabby, "Don't

PERSON LOCATION PERSON

Standard approach

*.. a dull guy from **Dayton**, warns Gabby, ..*

LOCATION

- Use annotated data to train a classifier with features such as: current word, words before/after (unigrams and n-grams), capitalization information, word length, etc.

Common errors

- Word features are too sparse, lack of generalization
- Some features (words) are never seen before

PERSON PERSON

Mrs. Chisholm, an actress who suffered a ***nervous breakdown***

PERSON PERSON DISEASE

PERSON ORGANIZATION PERSON

and married a dull guy from ***Dayton*** *, warns Gabby, "Don't*

PERSON LOCATION PERSON

Distributional vectors to the rescue

Dayton	breakdown
Akron**	**breakdowns***
Fairborn	break-down
Evendale	disconnection
Chesterland	attenuation
SYLVANIA	**deterioration***
Reynoldsburg	wreck
Youngstown	disintegration
Cincinnati*	**loss***
Ashtabula	disconnect

***** - seen in training

Embeddings for Named Entity Recognition

- Improvements by using different embedding types and combinations: (Miller et al 2004, Ratinov and Roth 2009, Lin and Wu, Turian et al 2010, Guo et al 2014)

- However, while some improvements are almost guaranteed, there is a considerable amount of engineering required

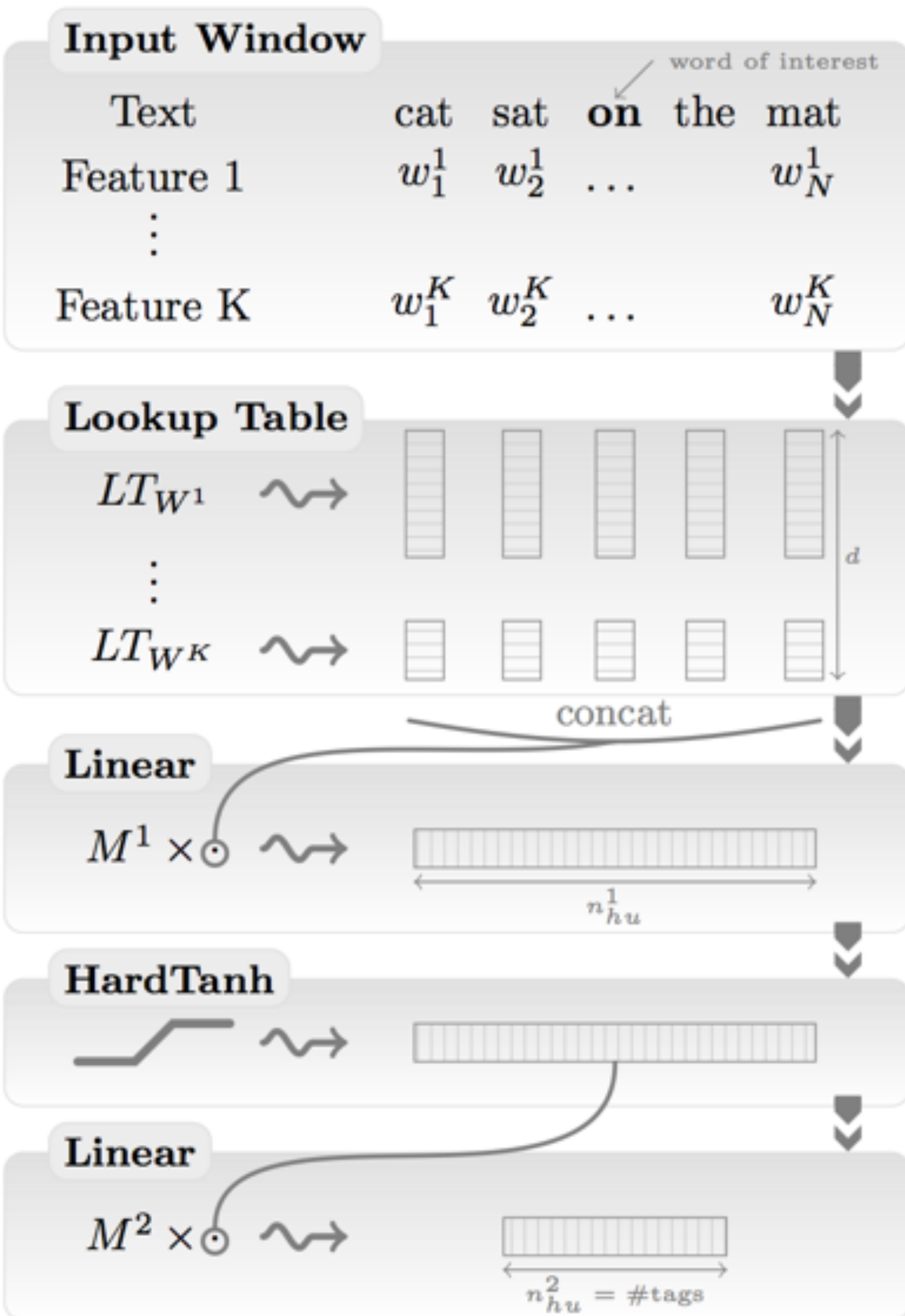
Going even further:
NLP (almost) from scratch, Collobert et al 2010

- Standard NLP tasks (part of speech tagging, parsing…) require hand-crafted features. Optimal features vary for different tasks.

Proposal

- Embedding-based neural network classifier with no additional features!

NLP from scratch



One-hot vectors

Embedding layer (initialized with pre-trained values)

Distribution over labels (Entity names, Part of speech tags…)

NLP from scratch

- State-of-the-art performance with no other features!

NLP from scratch

- Relevant word properties are implicitly modeled

Nearest neighbours in semantic space:

1983	Alphabet	ALPHABET	1
1985	Meatball	CUCUMBER	2
1982	Old-Fashioned	KITE	3
1986	Mummies	OATMEAL	4
1981	Vaudeville	NOODLE	6
1987	Travelin	BANNER	5
1978	Hairy	HAUNTING	7

NLP from scratch

- More robust, no need to engineer combinations with other features (word embeddings - *the main* feature)
- Can the state-of-the-art be significantly advanced? (Train task-specific embeddings, learn how to embed features, etc)

Outline

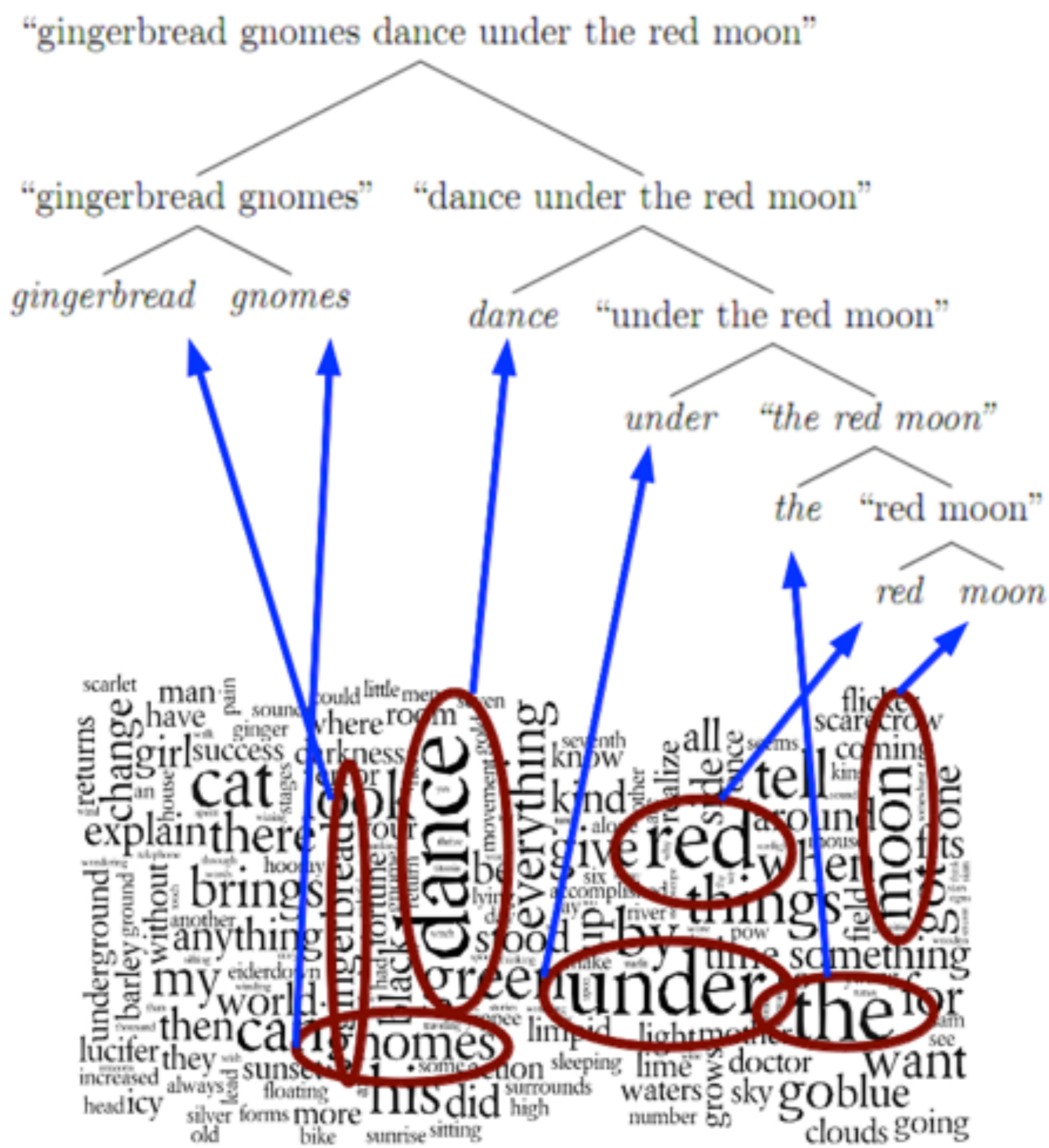
- Introduction to distributional semantics
- Distributed meaning representations
- Word meaning representations in NLP tasks

Break

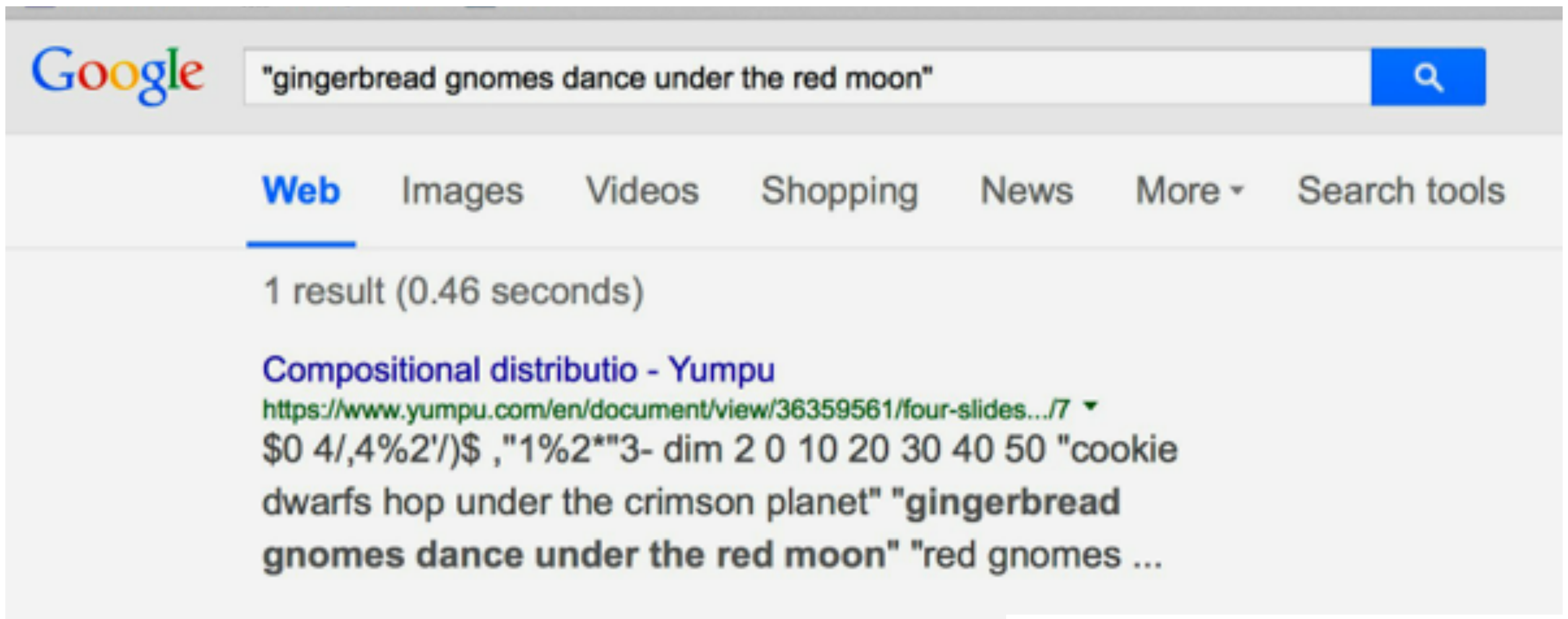
- Compositional distributional semantics
- Beyond sentence similarity
  - Decomposition, plausibility, morphology
  - Cross-lingual and cross-modal applications

Compositionality

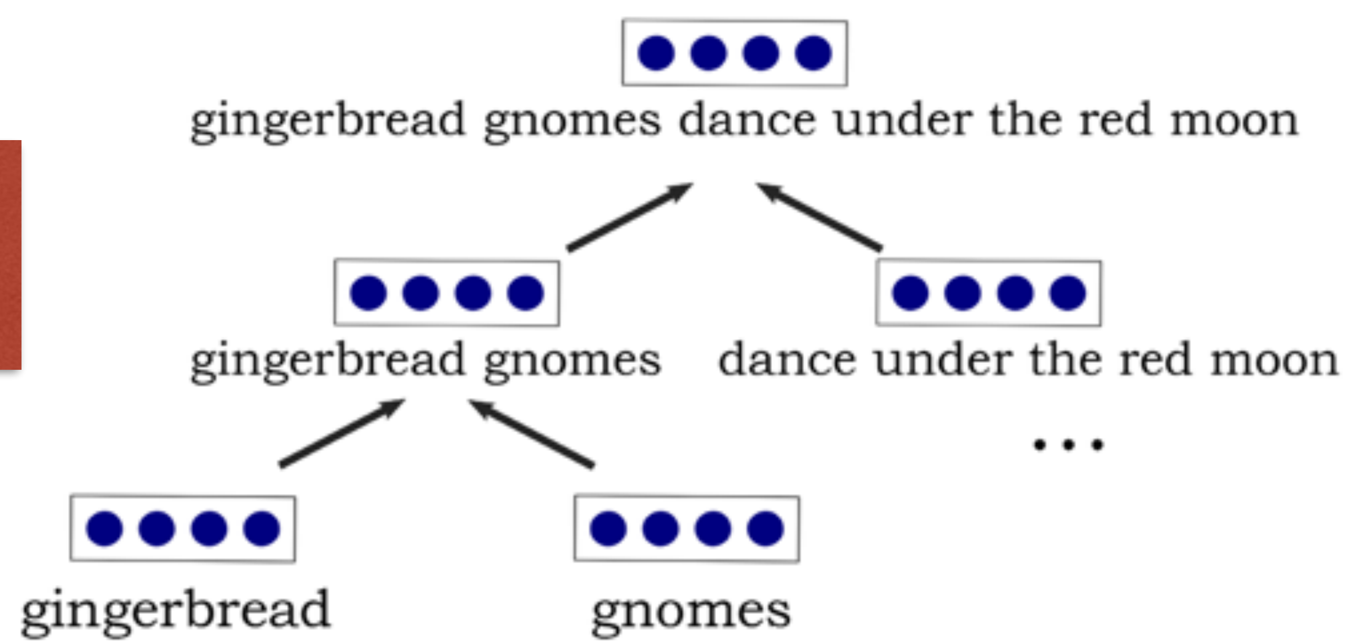
- The meaning of an utterance is a function of the meanings of its parts and their composition rules



Compositionality in distributional semantics?



Compose vector representations



Outline

- Introduction to distributional semantics
- Distributed meaning representations
- Word meaning representations in NLP tasks

Break

- Compositional distributional semantics
- Beyond sentence similarity
    - Decomposition, plausibility, morphology
    - Cross-lingual and cross-modal applications

Composition through vector mixtures

- Mitchell and Lapata, 2008, 2009, 2010
- Additive/multiplicative models: $\vec{p} = \vec{u} + \vec{v}$ $\vec{p} = \vec{u} \times \vec{v}$

	music	solution	craft	reasonable
practical	0	6	10	4
difficulty	1	8	4	0
practical + difficulty	1	14	14	4
practical x difficulty	0	48	40	0

Composition through vector mixtures. Evaluation

Human-assigned scores for similarity of phrases/small sentences:

Verb-Object	face difficulty	pose problem	7
	sell property	hold meeting	2
Adjective-Noun	left arm	elderly woman	1
	action programme	care plan	6
Subject-Verb	symptom subside	symptom lessen	6
	skin glow	skin burn	2

Evaluation: Correlation of (model-assigned) phrase cosines with human scores.

Composition through vector mixtures

Blacoe and Lapata 2012: close to state-of-the-art performance on sentence paraphrase identification (Microsoft Research Paraphrase Corpus, Dolan et al 2004):

- *Former company chief financial officer Franklyn M. Bergonzi pleaded guilty to one count of conspiracy on June 5 and agreed to cooperate with prosecutors.*
- *Last week, former chief financial officer Franklyn Bergonzi pleaded guilty to one count of conspiracy and agreed to cooperate with the government's investigation.*

Evaluation: Composed sentence vectors used as features in a classifier to predict YES/NO classes

What can compositional distributional semantics do?

Yes:

- blue pen

No:

- kick the bucket

Maybe:

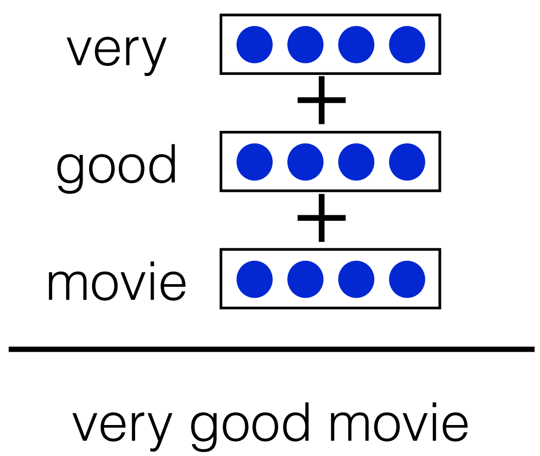
- some child, red face, former president
- pandas eat bamboo vs. bamboo eats panda

Beyond vector addition

1. More complex composition functions?
2. Can we *learn* how to compose?

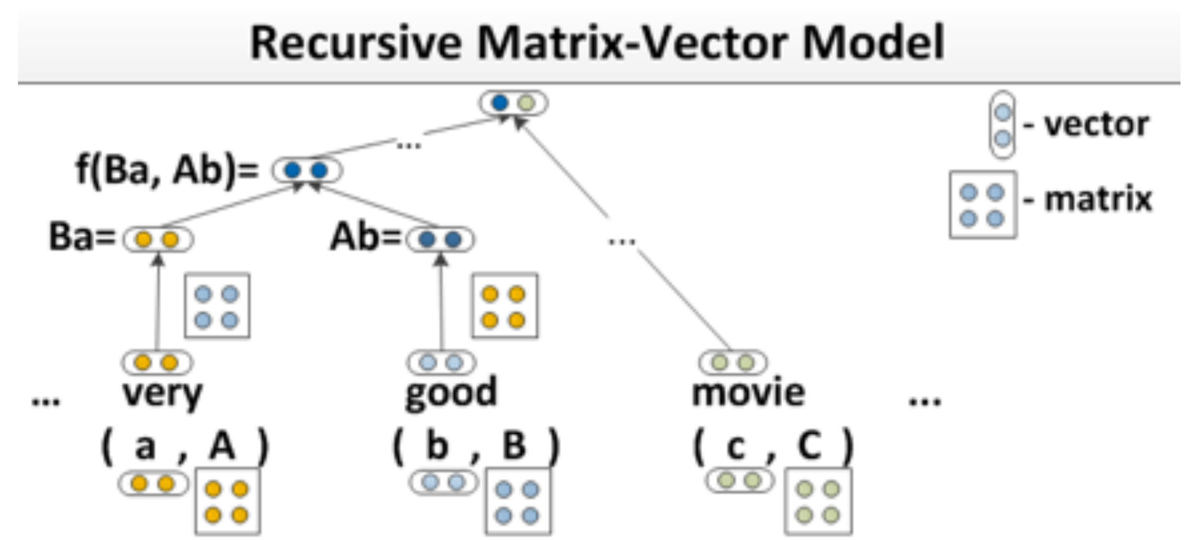
Overview

From



- Distributional word vectors as input
- No learning necessary (no, or few parameters)

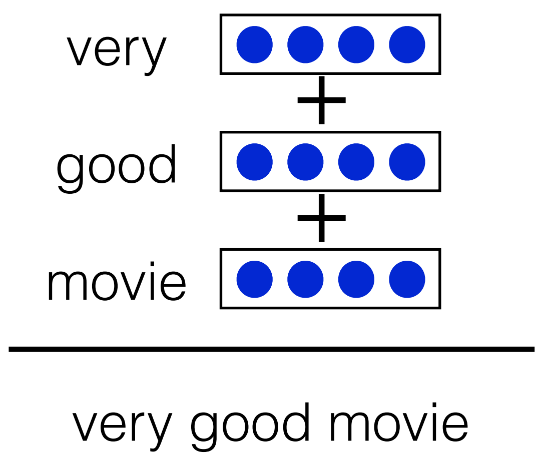
To



- Various composition models (with word vectors/composition functions as parameters)
- Learning objectives:
    - Corpus-extracted phrase vectors
    - Reconstruction error
    - Task-specific supervision

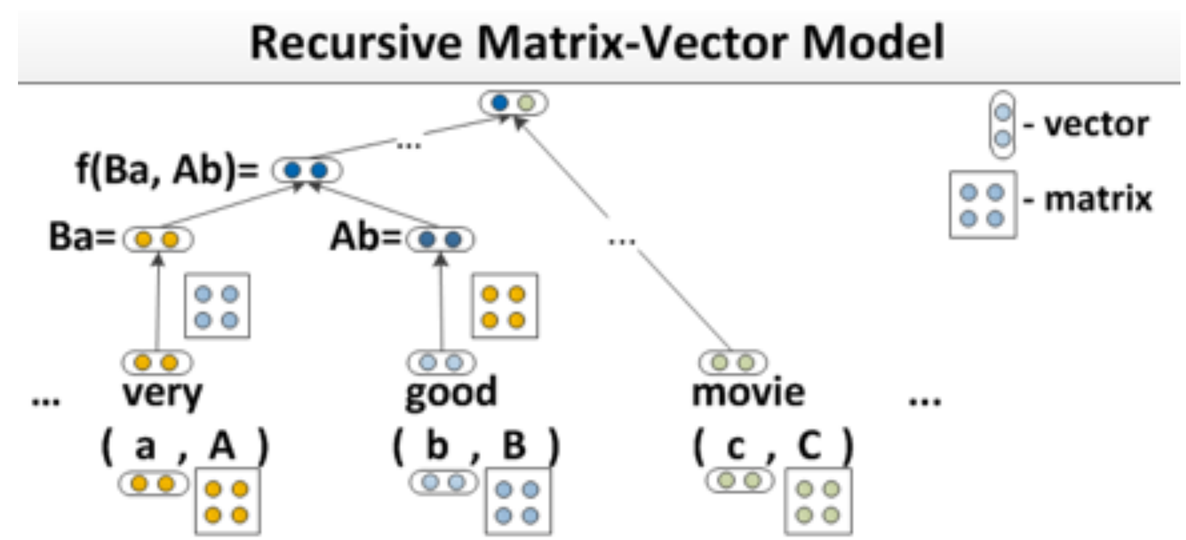
Overview

From



- Distributional (count) word vectors as input
- No learning necessary (no, or few parameters)

To



- Various composition models (with word vectors/composition functions as parameters)
- Learning objectives:
  - Corpus-extracted phrase vectors
  - Reconstruction error
  - Task-specific supervision

Baroni and Zamparelli 2010

Function application in vector space

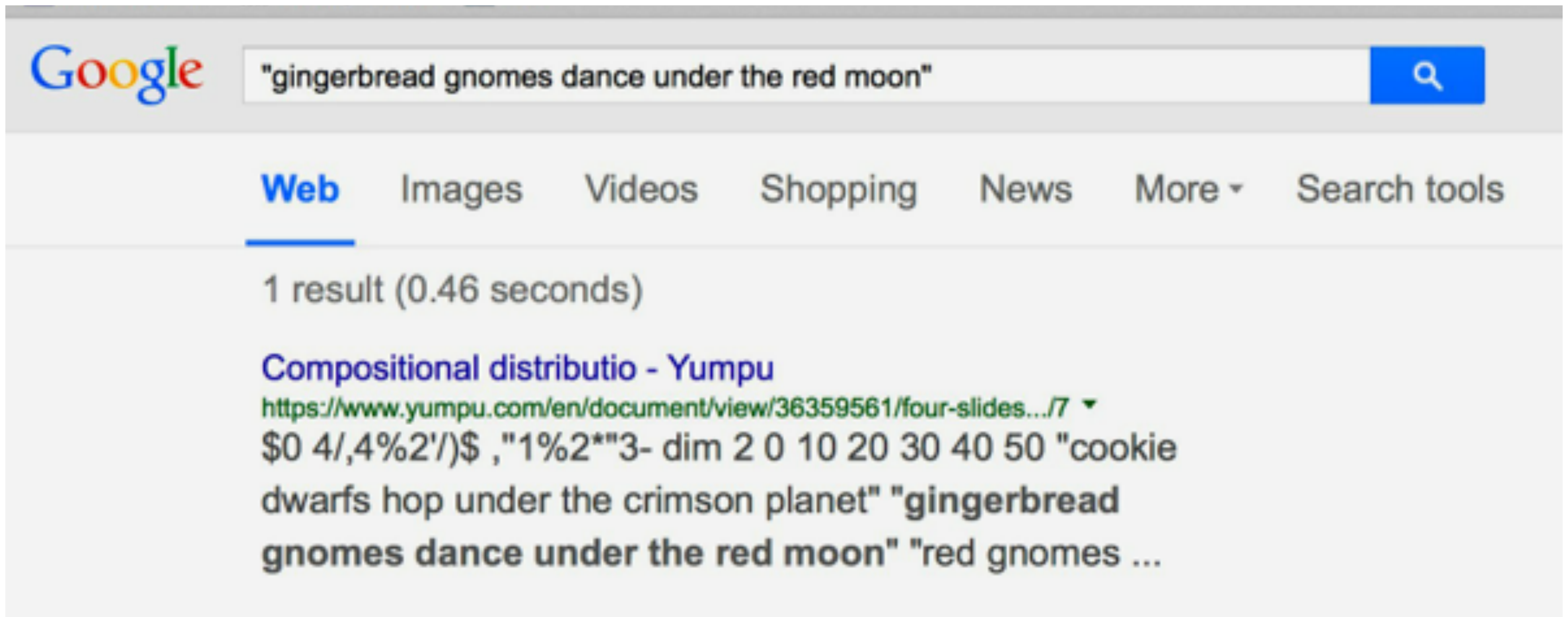
Baroni and Zamparelli, 2010

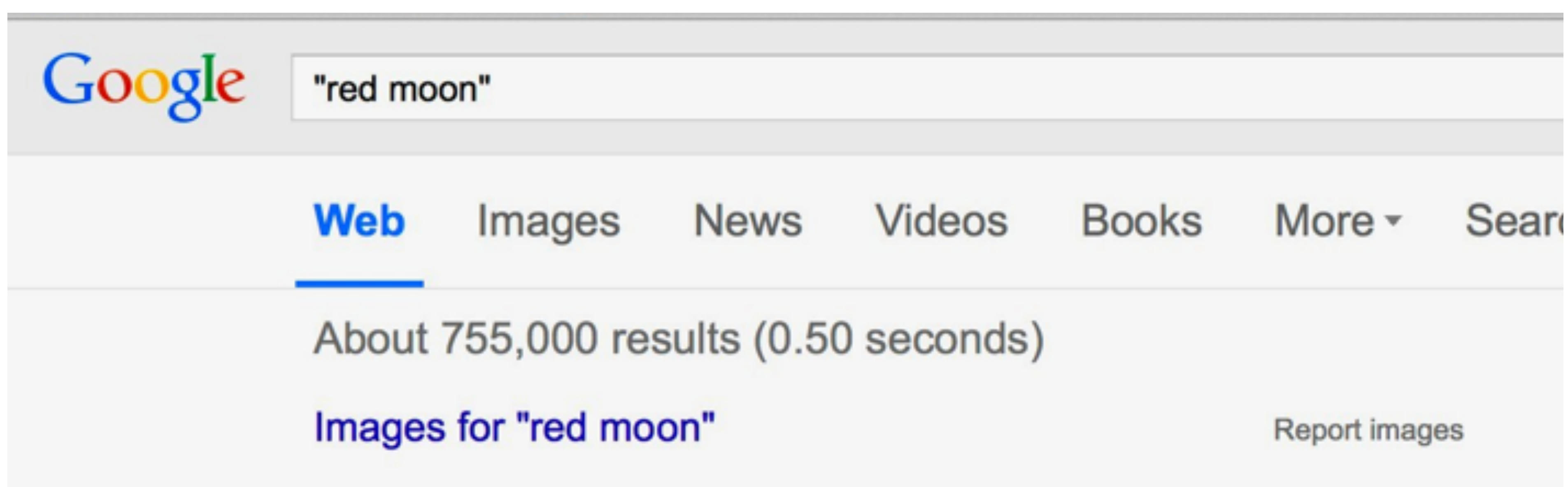
Distributional composition: distributional functions (e.g. adjectives, verbs, determiners) applied on distributional vectors (e.g. nouns)

- Adjectives are linear functions
- Nouns are vectors
- Linear functions are matrices, function application is function-vector multiplication

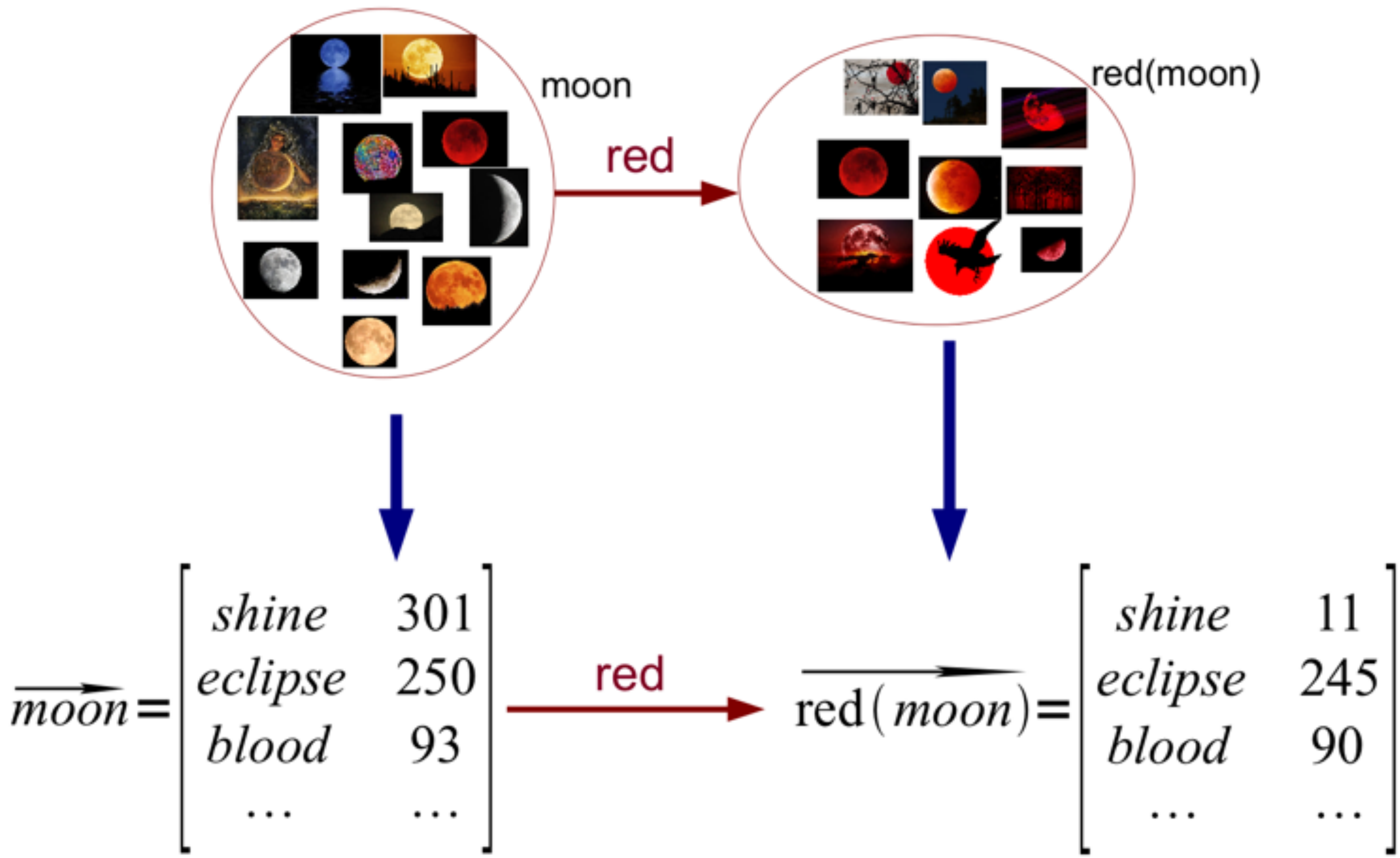
$$\vec{p} = \vec{n}\mathbf{A}$$

How to learn composition functions?





Baroni and Zamparelli, 2010: Learn composition from observed phrases



From Marco Baroni

- Coeke+Clark+Grefenstette+Sadrzadeh, Guevara, Socher et al, Zanzotto et al.

Baroni and Zamparelli, 2010: 2. Learn composition from observed phrases

- Nearest neighbors of *observed* phrases:

important route	important transport, important road, major road
historical map	topographical atlas, historical material
young husband	small son, small daughter, mistress

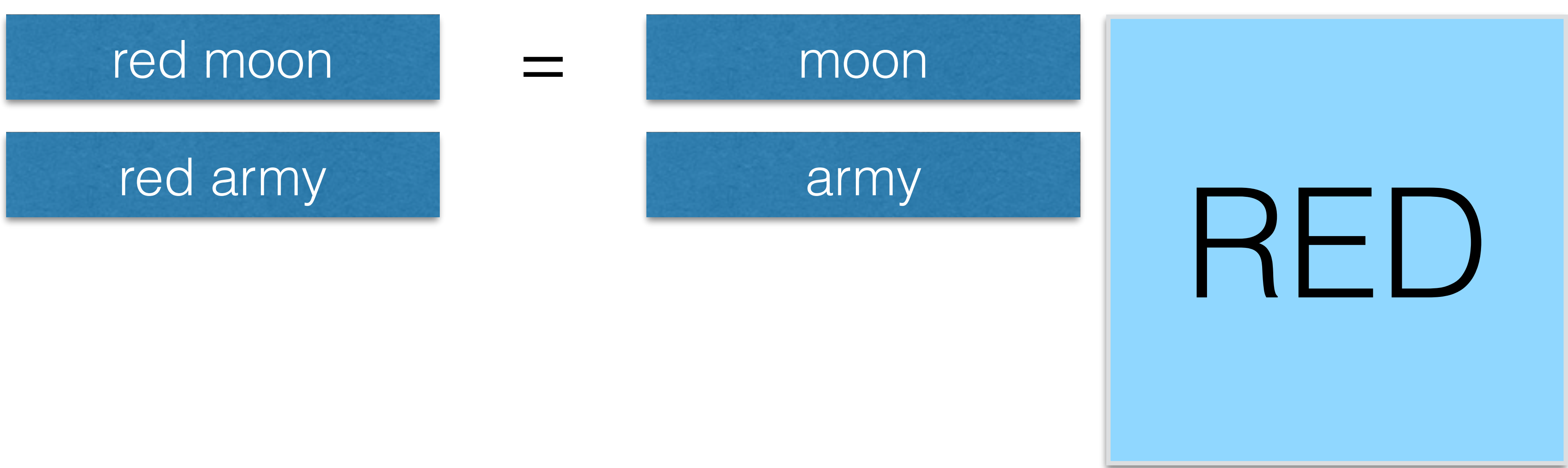
Learning composition functions

Training

- extract noun count vectors
- extract AN phrase vectors
- learn A matrix (e.g. ordinary least squares regression)

Observed phrases

	shine	blood	Soviet
red moon	11	90	0
red army	0	22	50



Learning composition functions

- Outperforms component-wise operations on small phrase/sentence similarity (verb-object, adjective-noun, subject-verb-object)

- Extend beyond one/two argument functions (n-argument functions become tensors or order n+1?) (Paperno et al 2014)

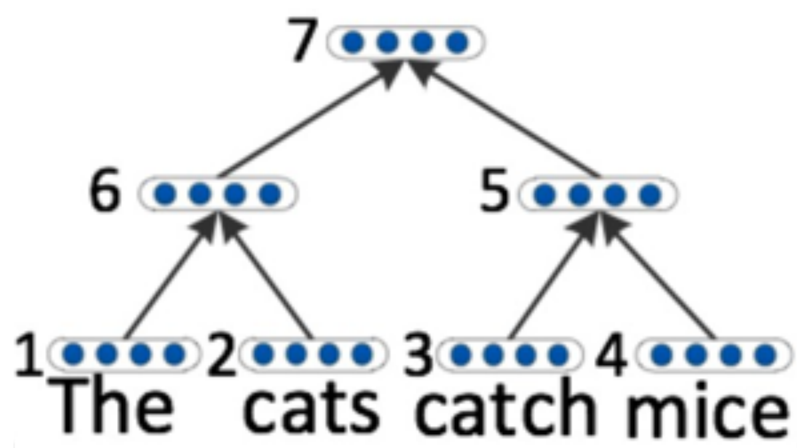
Socher et al 2011

Socher et al 2011

- Composition function: one standard neural network layer (input: the concatenation of two children, output: phrase vector)

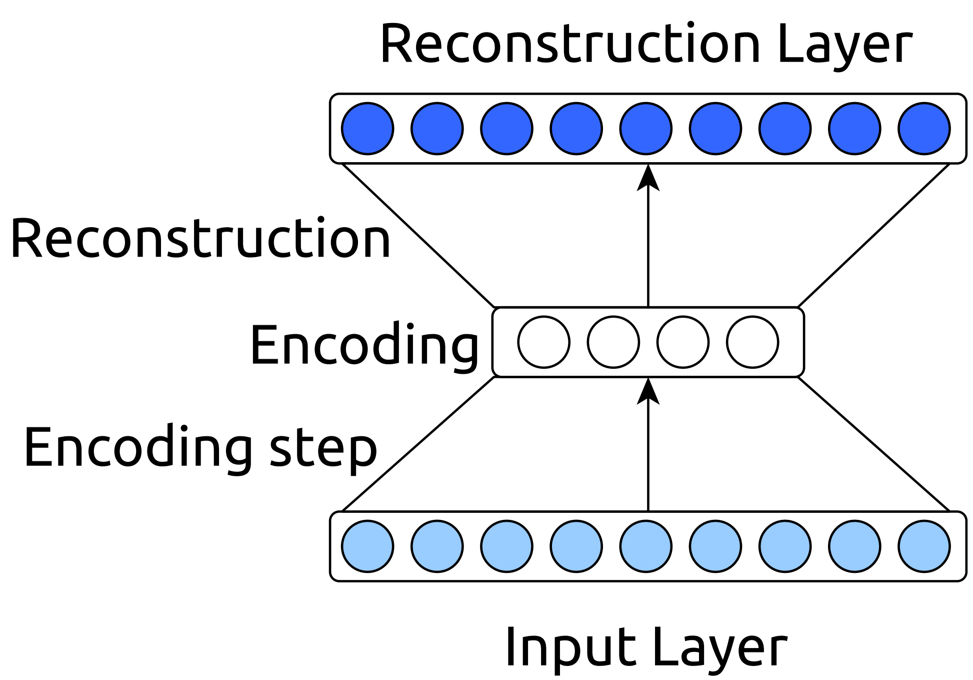
$$\vec{p} = g([\vec{c_1}; \vec{c_2}]\, W_e)$$

- Recursively compose vectors in syntactic trees



Autoencoder composition learning

- Learn encoding/decoding matrices in order to compress and decompress (reconstruct) the input



- Encode

$$\vec{p} = g([\vec{c_1}; \vec{c_2}]\, W_e)$$

- Decode

$$[\vec{c'_1}; \vec{c'_2}] = g(\vec{p}\, W_d)$$

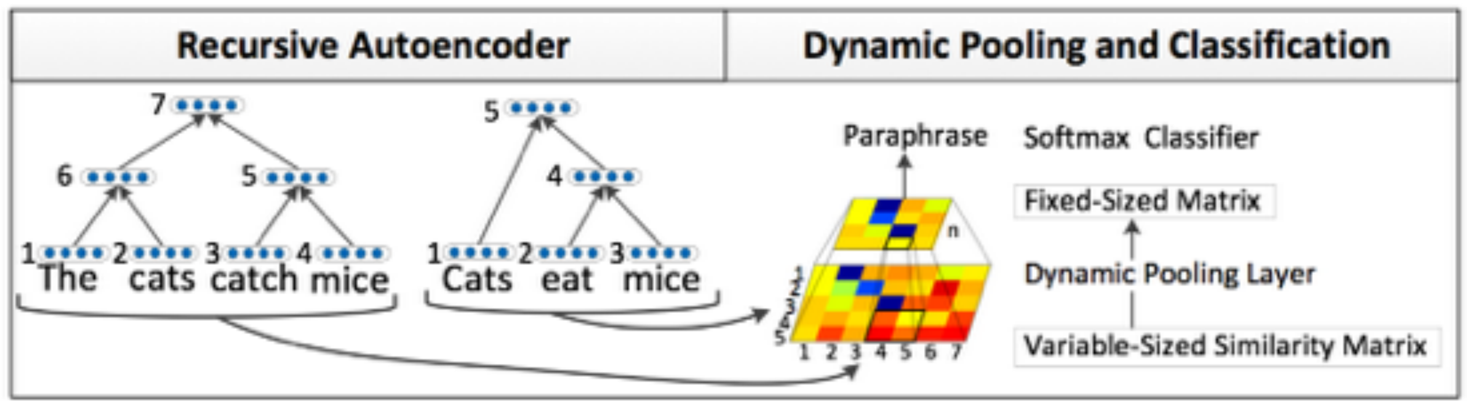
- Reconstruction error

$$||[\vec{c'_1}; \vec{c'_2}] - [\vec{c_1}; \vec{c_2}]||$$

Image from Greffenstette et al 2014

Similarity of composed representations

- Complete model for sentence similarity:



- Nearest neighbours of composed phrases:

the U.S.	the former U.S.
suffering low morale	suffering heavy casualties
conditions of his release	negotiations for their release
advance to the next round	advance to the semis

Adding task-specific supervision

- Compositional semantics for sentiment analysis (Socher et al 2011)

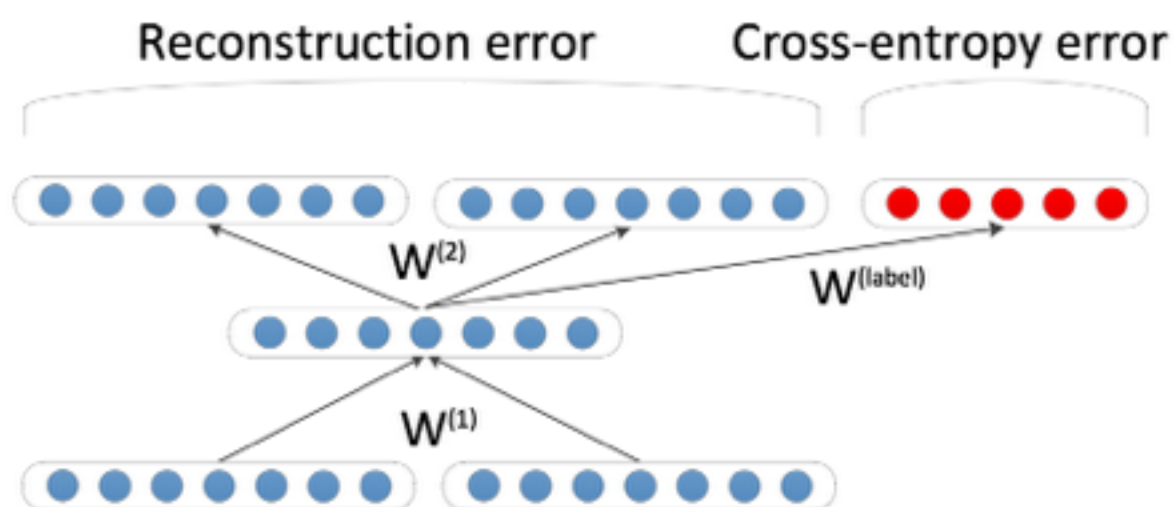
- Movie polarity:

Positive: *see it , see it again and when the dvd comes out , buy it , because a movie this hilarious will surely have outtakes to die for.*

Negative: *i'm willing to give director peter mettler credit for trying something different , but this particular experiment is not a success .*

Adding task-specific supervision

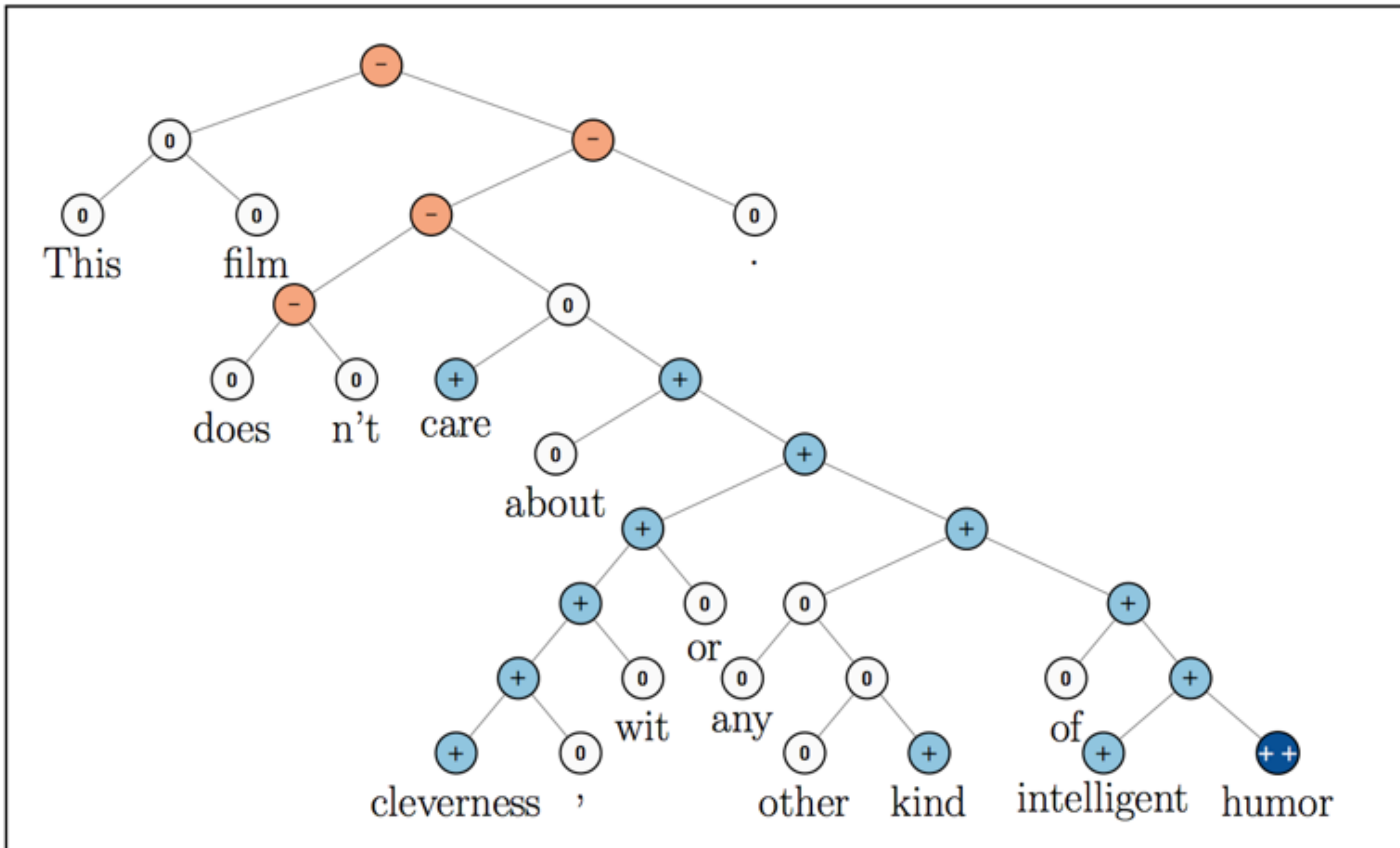
- Joint composition and label-prediction objective, Socher et al 2011



- Distributional representations are adapted to the task (*bad* is not similar to *good* anymore)

Semantic compositionality over a sentiment treebank: Socher et al 2014

- Sentiment changing through a parse tree (Recursive Neural Tensor Network)



Some references

- Kintsch 2001, Landauer and Dumais 1997, Mitchell and Lapata 2008, Mitchell and Lapata 2010, Baroni and Zamparelli 2010, Coecke et al 2010, Zanzotto et al 2010, Socher et al 2012, Socher et al 2013, Socher et al 2014, Dinu et al 2013, Li et al 2013, Grefenstette et al 2013, Polajnar et al 2014, Paperno et al 2014, Le and Mikolov 2014, Pham et al 2015, Tai et al 2015, Polajnar et al 2015, Fried et al 2015

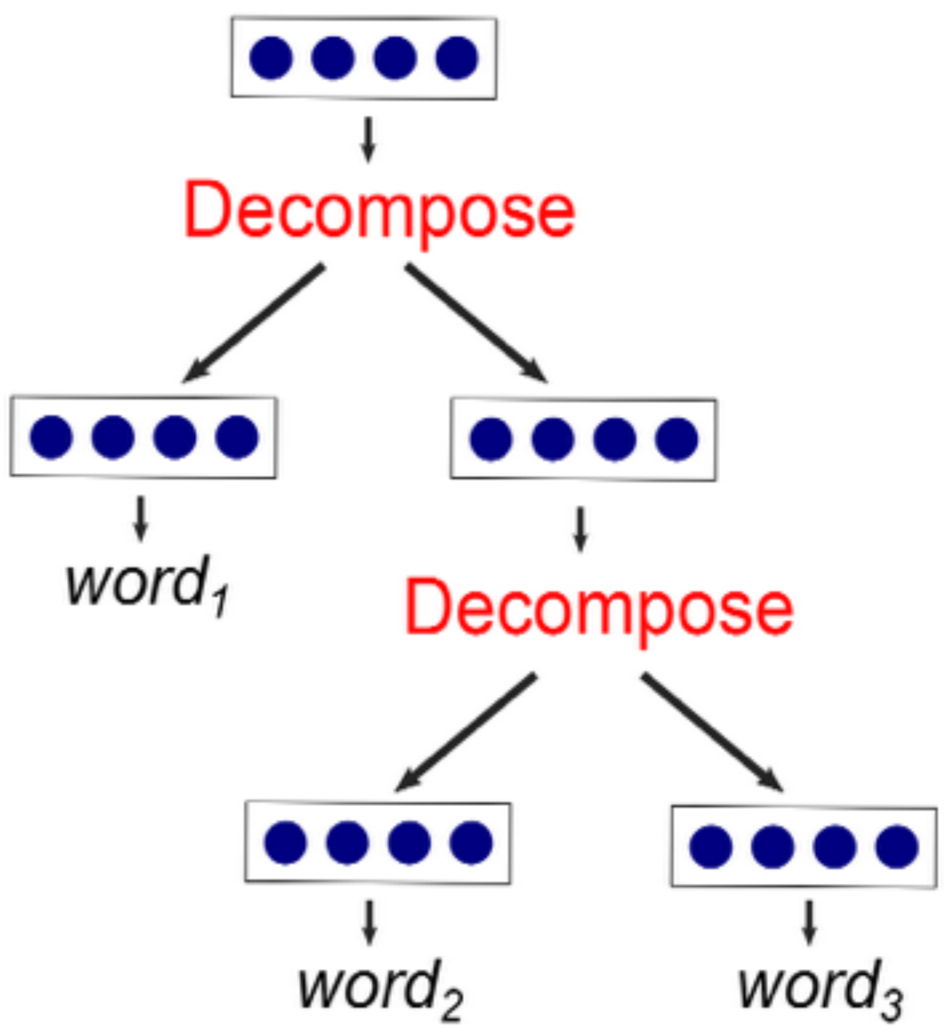
Outline

- Introduction to distributional semantics
- Distributed meaning representations
- Word meaning representations in NLP tasks

Break

- Compositional distributional semantics
- Beyond sentence similarity
  - Decomposition, plausibility, morphology
  - Cross-lingual and cross-modal applications

Phrase generation through de-compositional semantics



1. Decomposition
   - linear function

   $$[\vec{u}; \vec{v}] = \vec{p} \times \mathbf{W}_d$$

   - trained with observed phrase/word pair vectors
2. Nearest neighbour query

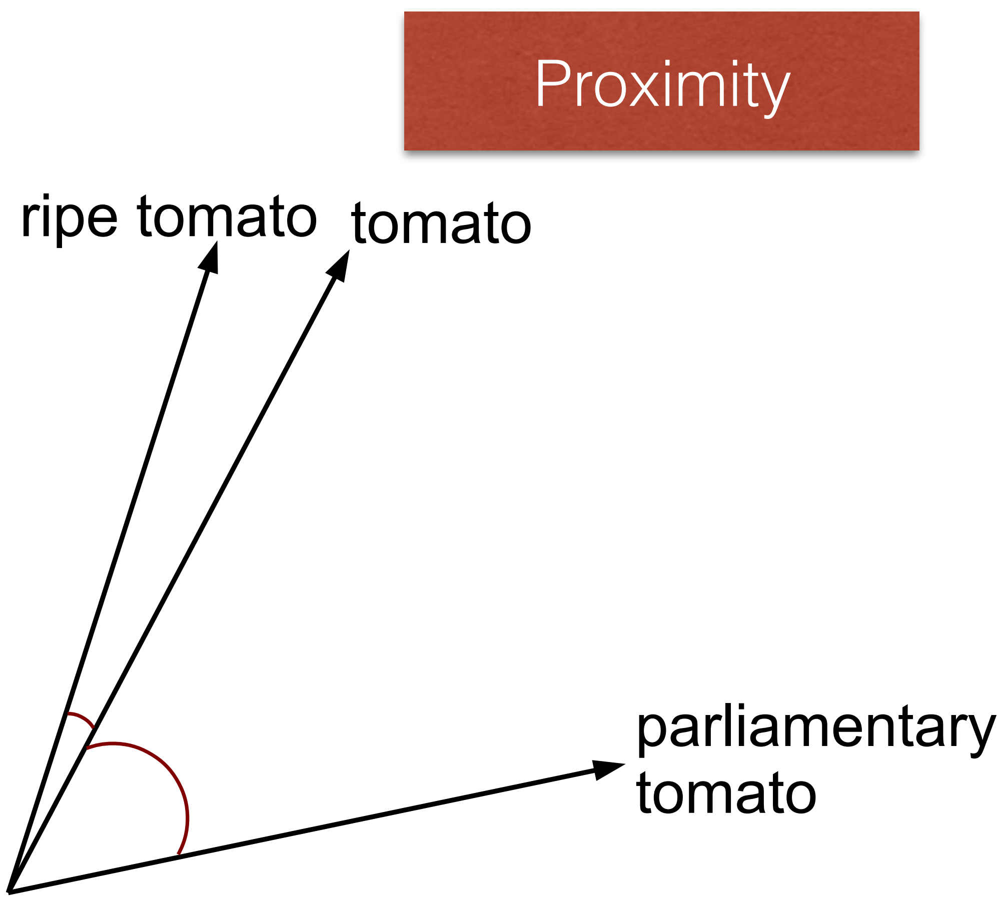
$$word_1 = \mathrm{NN}_{lex}(\vec{u})$$
$$word_2 = \mathrm{NN}_{lex}(\vec{v})$$

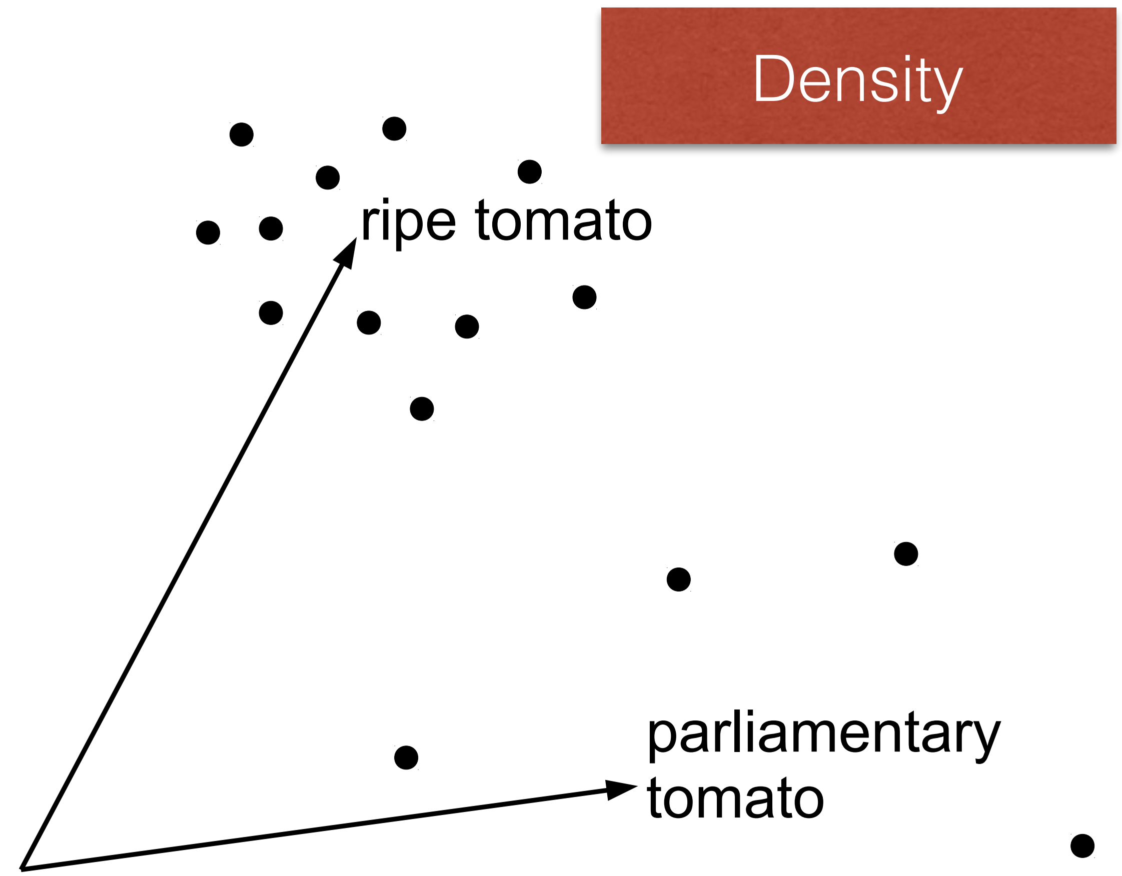
Phrase generation through de-compositional semantics

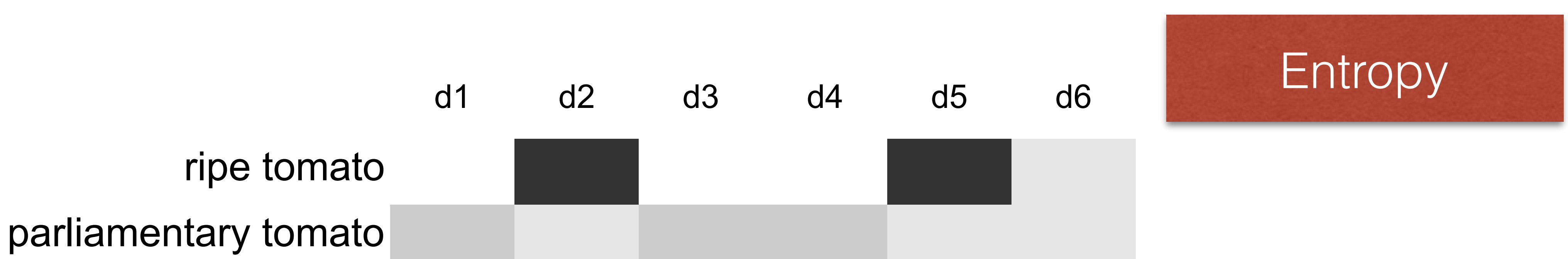
- Noun to Adj-Noun and Adj-Noun to Noun-Prep-Noun paraphrase generation

Compose	Generate	Gold
thunderstorm	thundery storm	electrical storm
reasoning	deductive thinking	abstract thought
jurisdiction	legal authority	legal power
superstition	old-fasion religion	superstitious notion
vitriol	political bitterness	sulfuric acid
mountainous region	region in highlands	region in mountains
inter-war years	years during 1930s	years between wars

Measuring phrase plausibility







Measuring phrase plausibility

- Proximity of composed vector to words is a good predictor of phrase acceptability (Vecchi et al 2011)

- Composed-vector plausibility measures can be used to predict bracketing of noun phrases (*miracle [home run]* vs. *[miracle home] run*) (Lazaridou et al 2013)

Morphology

Derivation as composition: Lazaridou et al 2013

- Affixes as functions from stems to derived words:

$$\vec{redo} = \vec{do} \times \mathbf{RE}$$

- Affix matrices learned from corpus-observed stem/derived word vectors (try/retry, climb/reclimb, open/reopen)

Morphology

- Nearest neighbours of composed words:

re+issue	original, expanded, long-awaited
re+touch	repair, refashion, reconfigure
re+sound	reverberate, clangorous, echo
type+ify	embody, characterize, essentially
nerve+ous	brochial, nasal, intestinal

- Unsupervised morphology induction: Soricut and Och 2015
    - Induce morphological transformations when supported by *regularities* in semantic space
    - e.g. suffix:ed:ing (substitute suffix *ed* with *ing*) is supported by the semantic regularities given by pairs: (bored, boring), (stopped, stopping), etc.

References

Decomposition

- Socher et al 2011, Andreas and Ghahramani 2013, Kalchbrenner and Blunsom 2013, Dinu and Baroni 2014

Plausibility

- Vecchi et al 2011, Lazaridou et al 2013

Morphology

- Guevara 2009, Luong et al 2013, Lazaridou et al 2013, Botha and Blunsom 2014, Marelli and Baroni 2015, Soricut and Och 2015

Outline

- Introduction to distributional semantics
- Distributed meaning representations
- Word meaning representations in NLP tasks

Break

- Compositional distributional semantics
- Beyond sentence similarity
  - Decomposition, plausibility, morphology
  - Cross-lingual and cross-modal applications

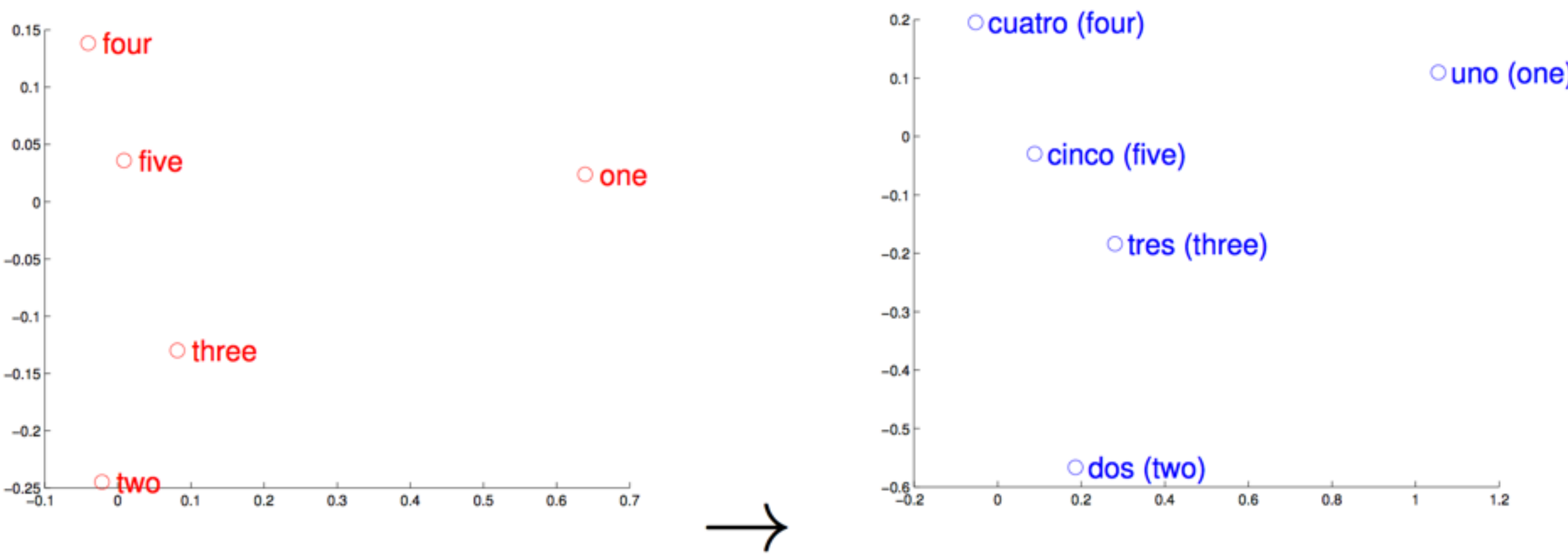
Bilingual lexicon induction

- Dictionaries can never be complete: new/rare/misspelled words
- Parallel data is limited
- Low-resource languages

Leverage monolingual data to translate new words?

Bilingual lexicon induction

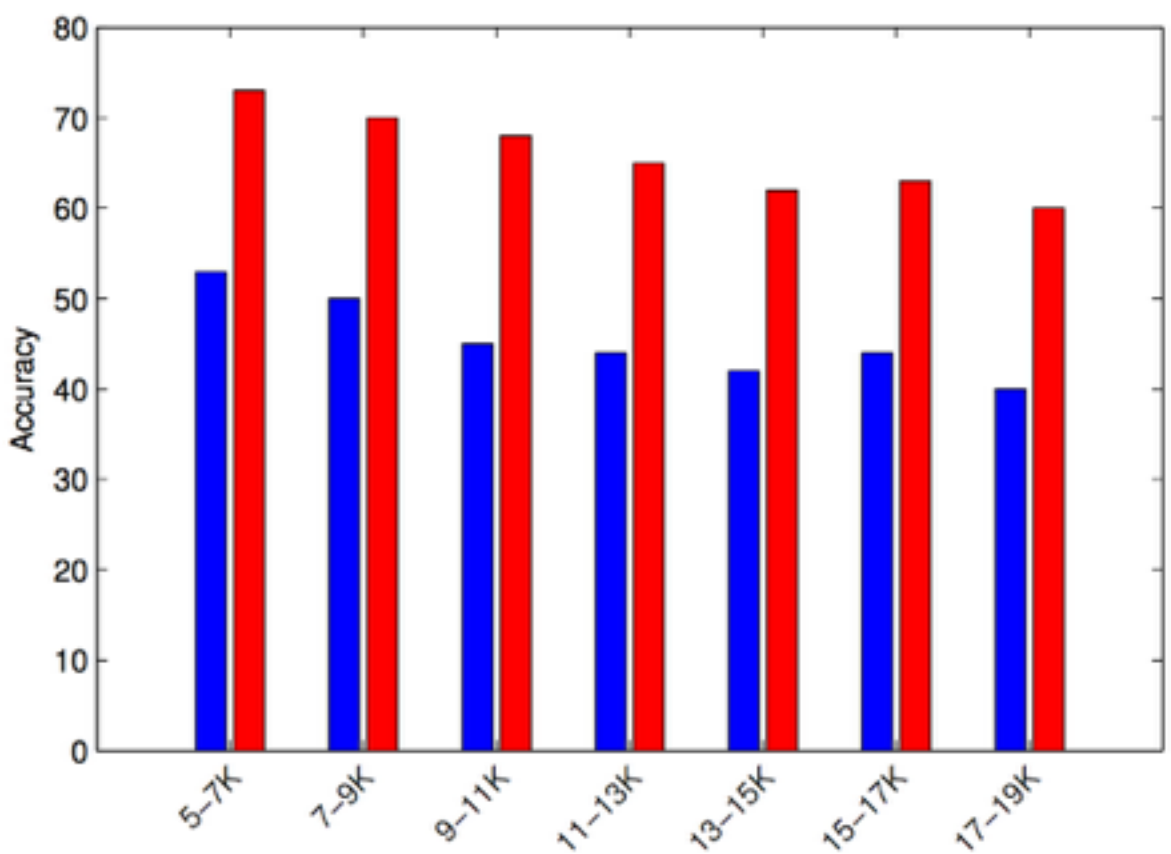
- Rapp 1995, Rapp 1999, Koehn and Knight 2002, Klementiev et al 2012
- Mikolov et al 2013: Words and their translations have similar geometric arrangements in English and Spanish



Bilingual lexicon induction: Mikolov et al 2013

- Learn individual semantic spaces from *monolingual* data
- Learn a linear transformation to map from one space to another

- Word translation accuracies for different frequency bins



- Translate small phrases by adding a decomposition layer

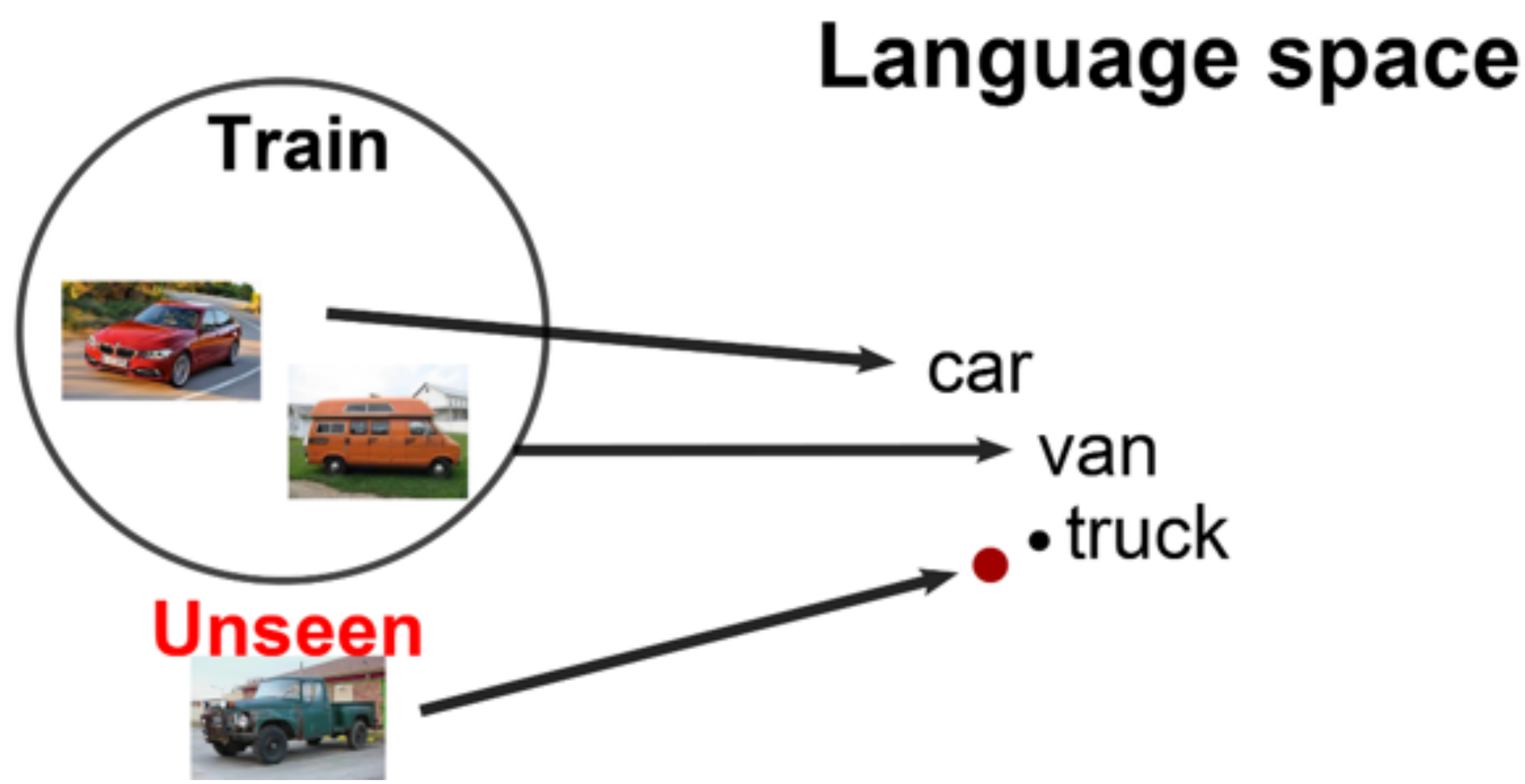
English	Italian
vicious killer	assasino feroce
black **tie**	**cravatta** nera
indissolubile **tie**	**alleanza** indissolubile

Cross-modally: Zero-shot learning in vision

- Object recognition limited to a set of categories (discrete labels)
- In reality: Unknown objects, ambigous/task-specific labels, multiple labels

Zero-shot learning in vision

- Exploit the correlation between visual similarity and text-based similarity to predict labels for unseen objects



Zero-shot learning in vision

- Zero-shot image labeling is much more difficult

	Lexicon induction	Image labeling
P@1	33%	0.5% / 5.6%*

* Lazaridou et al, 2015

Visual similarity

?

Text-based similarity

tarantula	highland
	
anteater	whisky
arachnid	lowland
spider	bagpipe
opossum	glen
scorpion	distillery

From Lazaridou et al 2015

References

Bilingual lexicon acquisition

- Rapp 1995, Koehn and Knight 2002, Klementiev et al 2012, Mikolov et al 2013, Dinu and Baroni 2014, Dinu et al 2014, Kiela et al 2015, Lazaridou et al 2015

Zero-shot object recognition

- Frome et al 2013, Socher et al 2013, Norouzi et al 2013, Lazaridou et al 2014

Thank you!