

Semantic Similarity Frontiers: From Concepts to Documents



David Jurgens
Stanford University

Mohammad Taher Pilehvar
Sapienza University of Rome



ERC grant 259234

Semantic Similarity's Key Question: How similar are two lexical items?



How similar are two sentences?

The boss fired the worker

The boss fired the employee

How similar are two sentences?

The boss fired the worker

The boss fired the employee

The supervisor let the employee go

How similar are two sentences?

The boss fired the worker

The boss fired the employee

The supervisor let the employee go

The supervisor reprimanded the
worker

The boss promoted the worker

very similar

very similar

somewhat

similar

related

**Don't we already have a
bunch of solutions for this?**

Lots of work on all types of text and concept input

Allison and Dix (1986)

Gusfield (1997)

Wise (1996)

Keselj et al. (2003)

***50+ Approaches from
SemEval***

2012, 2013, 2014

Sussna (1993, 1997)

Wu and Palmer (1994)

Resnik (1995)

Jiang and Conrath (1997)

Lin (1998)

Hirst and St-Onge (1998)

Leacock and Chodorow (1998)

Patwardan (2003)

Salton and McGill (1983)

Landauer et al. (1998)

Banerjee and Pederson (2003)

Turney (2007)

Gabrilovich and Markovitch (2007)

Ramage et al. (2009)

Yeh et al. (2009)

Radinsky et al. (2011)

**We refer to these as
Linguistic Levels**



Sentence

Word

Sense

Lots of work on all types of text and concept input

Allison and Dix (1986)
Gusfield (1997)
Wise (1996)
Keselj et al. (2003)
*50+ Approaches from
SemEval
2012, 2013, 2014*

**Not to mention
word embeddings...**



Sussna (1993, 1997)
Wu and Palmer (1994)
Resnik (1995)
Jiang and Conrath (1997)
Lin (1998)
Hirst and St-Onge (1998)
Jacobs and Chodorow (1998)
Patwardan (2003)
Banerjee and Pederson (2003)

Gab

7)

Ramage et al. (2009)
Yeh et al. (2009)
Radinsky et al. (2011)

**We refer to these as
Linguistic Levels**

Sentence

Word

Sense

Why do we have so many similarity methods?!

New resources or machine learning methods become available

~20 embeddings papers at EMNLP alone

New datasets reveal weaknesses in previous methods

SOA is a moving target

Need to adapt for new types of input or domains

Microtext, Biomedical, Multilingual

Application-specific similarity functions

Do we still need *more* methods?

Semantic similarity itself is not an end-task,
but rather a component

Applications can select the similarity method that
yields the best performance.

Performance on new benchmarks is still not
satisfactory

Low hanging similarity fruit is solved, but *many*
challenging cases still remain

New techniques yield much better
performance

Tutorial Objectives

Make sense of current Semantic Similarity state of the art!

- Task formulation and requisite resources

- Standard and state-of-the-art algorithms

- Current evaluation metrics

Provide practical knowledge

- What open source tools and data are available

- What are the current open problems

Target audience: we assume no knowledge of any machine learning or lexical semantics

Stop us to ask questions at any time!

Tutorial *non*-Objectives

Provide gory details of methodologies

We focus more on the landscape and knowing *which* methods matter

But feel free to ask questions on details if interested!

Covering all work on a similarity task

Course materials provide an extended bibliography

We focus on the most exciting ideas (to us)

**You should leave feeling comfortable
knowing what papers to read next, why,
and roughly what they're about!**

Quick outline of the morning

Foundations in Semantic Similarity

Concepts, Terminology, and Examples

State of the Art Overviews

Similarity when comparing Concepts, Words, Phrases,
Sentences, Paragraphs, *or* Documents

Cross-Level Semantic Similarity

Open source Tools and Resources

Current Challenges and Future Work

Quick outline of the morning

Foundations in Semantic Similarity

Concepts, Terminology, and Examples

State of the Art Overviews

Similarity when comparing Concepts, Words, Phrases,
Sentences, Paragraphs, *or* Documents

Cross-Level Semantic Similarity

Open source Tools and Resources

Current Challenges and Future Work



Coffee Break happens in here!
10:30 - 11:00

Foundations

Semantic similarity can be defined on many **linguistic levels**

Word senses (concepts)

Words

Phrases

Sentences

Paragraphs

Documents

For the most part, different algorithms are used for each kind of item being compared.

Typically, two main resources for measuring similarity



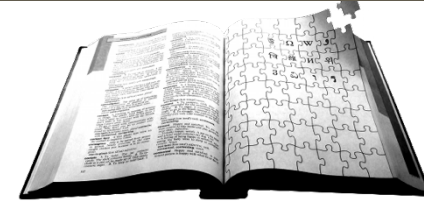
**Massive corpora
of text documents**

Typically, two main resources for measuring similarity



**Massive corpora
of text documents**

WordNet
A lexical database for English



WIKTIONARY
the free dictionary



BabelNet

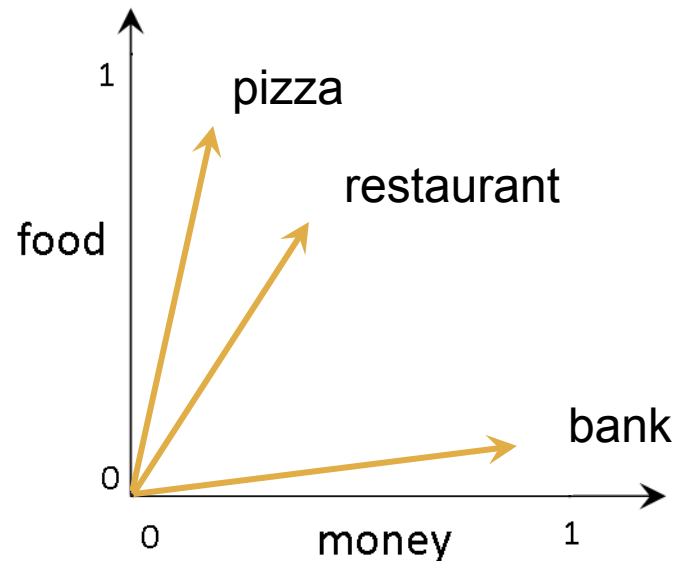


WIKIPEDIA
The Free Encyclopedia

**Semantic resources
and knowledge bases**

Many methods represent semantics using a vector space model (VSM)

Representing meaning in a machine-interpretable or mathematical format



Vector Space Models

- Simple representation based on linear algebra
- Easy comparison of different items based on a continuous scale of similarity
- Supported by studies in Cognitive science
- Flexible way of adjusting the degree of complication through setting the number of dimensions

Vector Space Models

- **Explicit**

Individual dimensions denote specific linguistic items,
e.g., words

Usually higher in dimension

The vector is interpretable

- **Continuous**

Dimensions do not correspond to explicit concepts

Usually lower in dimension

Vector Space Models

Vector comparison techniques

Kullback–Leibler (KL) divergence

$$D_{KL} (\mathcal{S}_1 \parallel \mathcal{S}_2) = \sum_{h \in H} \log_e \left(\frac{\mathcal{S}_1^h}{\mathcal{S}_2^h} \right) \mathcal{S}_1^h$$

Jensen–Shannon (JS) divergence

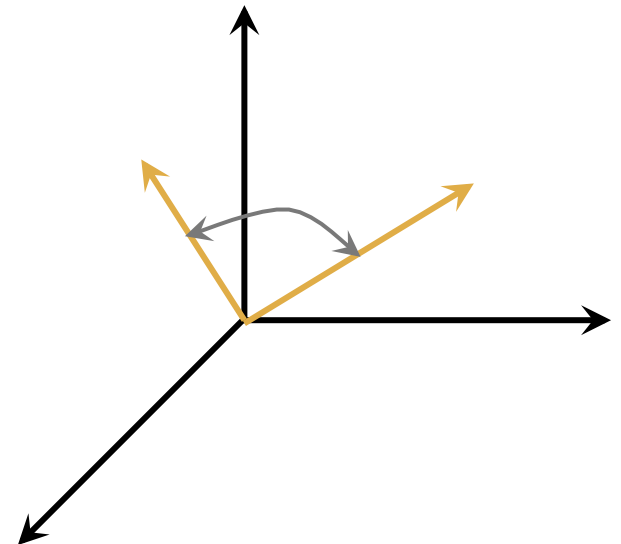
$$D_{JS} (\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{2} D_{KL} \left(\mathcal{S}_1 \parallel \frac{\mathcal{S}_1 + \mathcal{S}_2}{2} \right) + \frac{1}{2} D_{KL} \left(\mathcal{S}_2 \parallel \frac{\mathcal{S}_1 + \mathcal{S}_2}{2} \right)$$

Vector Space Models

Vector comparison techniques

Cosine distance

$$Sim_{Cos} (S_1, S_2) = \frac{S_1 \cdot S_2}{\|S_1\| \|S_2\|}$$



Vector Space Models

Vector comparison techniques

Tanimoto similarity (1957)

$$f(A, B) = \frac{A \cdot B}{|A|^2 + |B|^2 - A \cdot B}$$

Vector Space Models

Rank-based Vector comparison techniques

Rank-Biased Overlap (RBO)

$$RBO(S_1, S_2) = (1 - p) \sum_{d=1}^{|H|} p^{d-1} \frac{|H_d|}{d}$$

The set of overlapping dimensions between the top- d elements

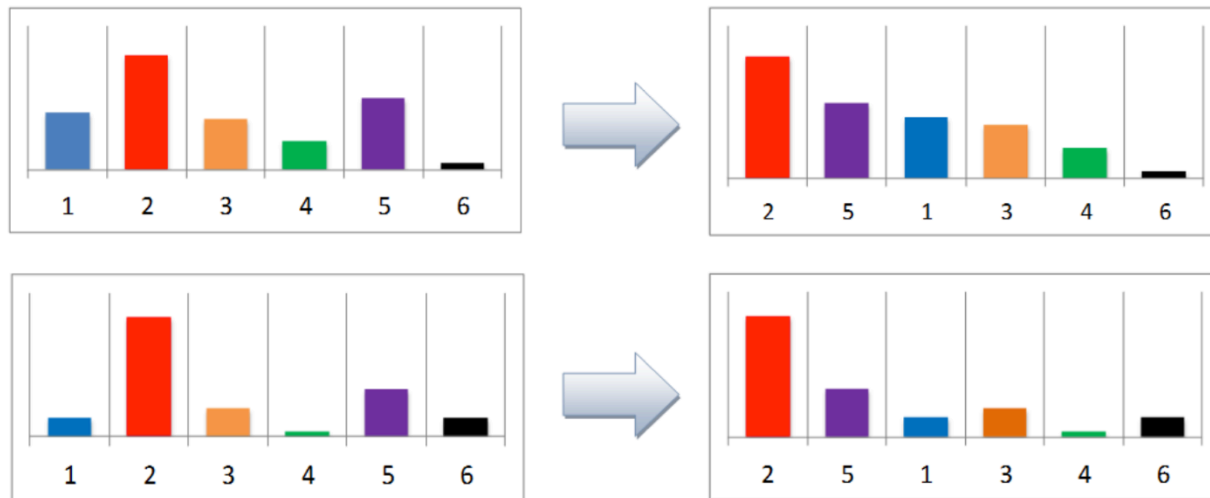
A parameter that determines the relative importance of the top elements.

Vector Space Models

Rank-based Vector comparison techniques

Weighted Overlap

$$\text{Sim}_{WO}(\mathcal{S}_1, \mathcal{S}_2) = \frac{\sum_{h \in H} (r_h(\mathcal{S}_1) + r_h(\mathcal{S}_2))^{-1}}{\sum_{i=1}^{|H|} (2i)^{-1}}$$



A few notes on the phenomenon of similarity

Similarity is *graded*

car vs. automobile -> 1.0

car vs. vehicle -> 0.6

car vs. tire -> 0.2

car vs. street -> 0.1

Similarity is *graded*

car vs. automobile -> 1.0

car vs. vehicle -> 0.6

car vs. tire -> 0.2

car vs. street -> 0.1



Relatedness is not the same as similarity

Similarity has psychological quirks

Nontransitive

Cuba vs. Jamaica

Cuba vs. China

Jamaica vs. China

Asymmetric

North Korea vs. China

China vs. North Korea

**These are ignored by nearly all approaches,
but see Gawron (2014)**

Desiderata for a Semantic Similarity Method

Consistently interpretable similarity scores
with explanations of why similar

Works well for different types of text
(news, web, social media, ...)

Applicable to multiple linguistic types
(words, phrases sentences)

Semantic Similarity: State of the Art

Many approaches incorporate techniques from more specific linguistic levels

Word senses (concepts)

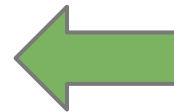
Words

Phrases

Sentences

Paragraphs

Documents



**Start here and
work our way to
bigger ideas!**

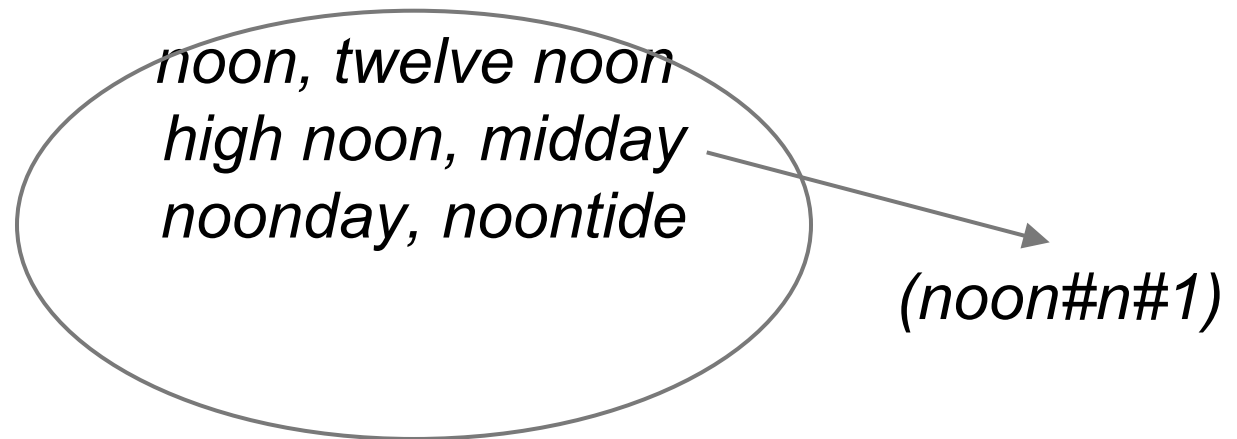
Semantic Similarity

between word senses

Concepts vs. senses

A WordNet synset (concept):

the middle of the day



Applications - general

Lowest (most fine-grained) level of semantic similarity

Can be extended to applications that require higher levels of similarity

MT evaluation, paraphrases recognition, textual entailment, information retrieval, question answering, text summarization, lexical substitution or simplification, query expansion

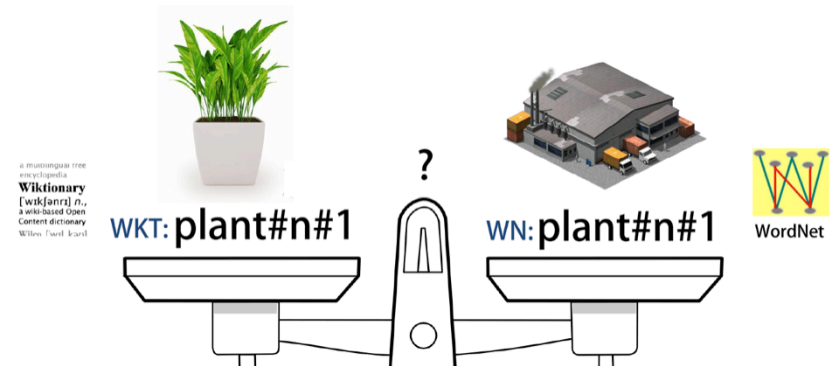
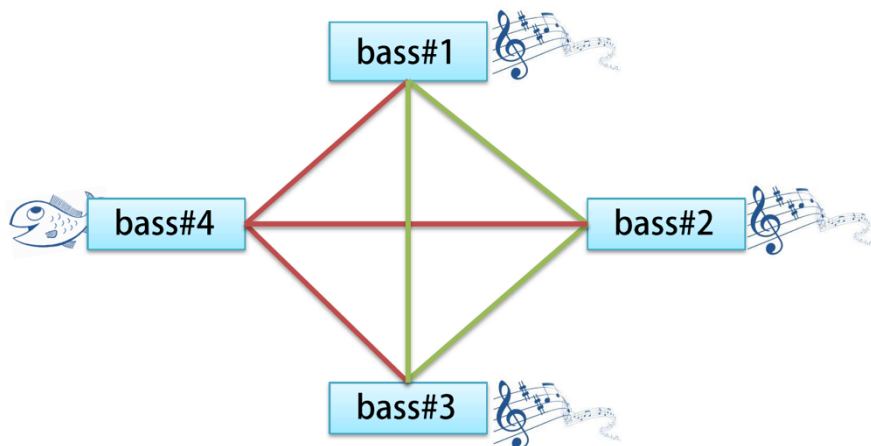
Applications - specific

WSD

install the updated **application**

- software application?
- application for a job?
- practical usage?

Coarsening Alignment



Sense Similarity Techniques

Tied to sense inventories

- Graph distance-based

 - WordNet-based

 - Thesauri-based

 - Dictionary-based

Explicit sense representation

- Simple gloss-based

- Random walk-based

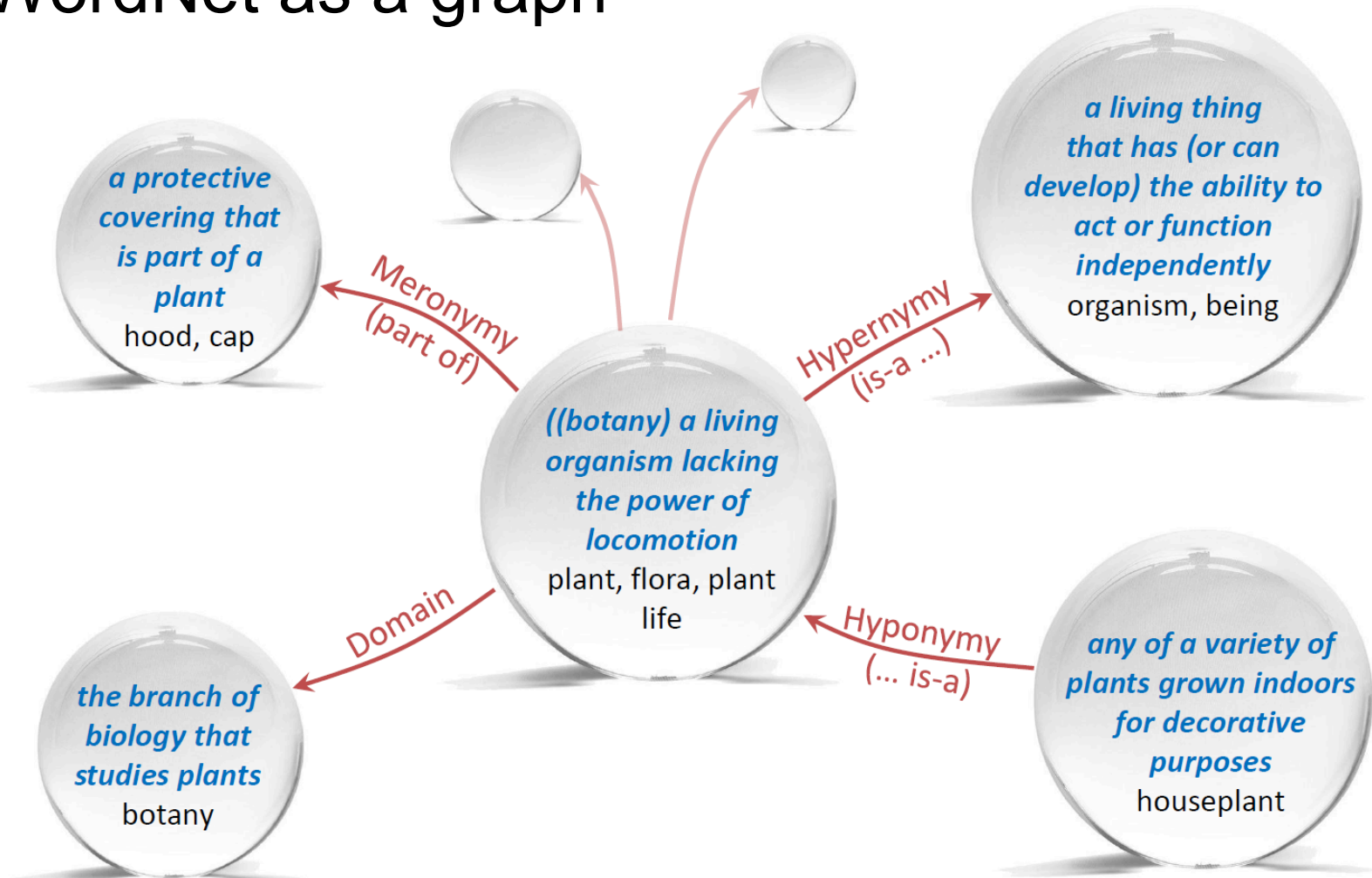
- Distributional

Not tied to sense inventories

Sense Similarity Techniques

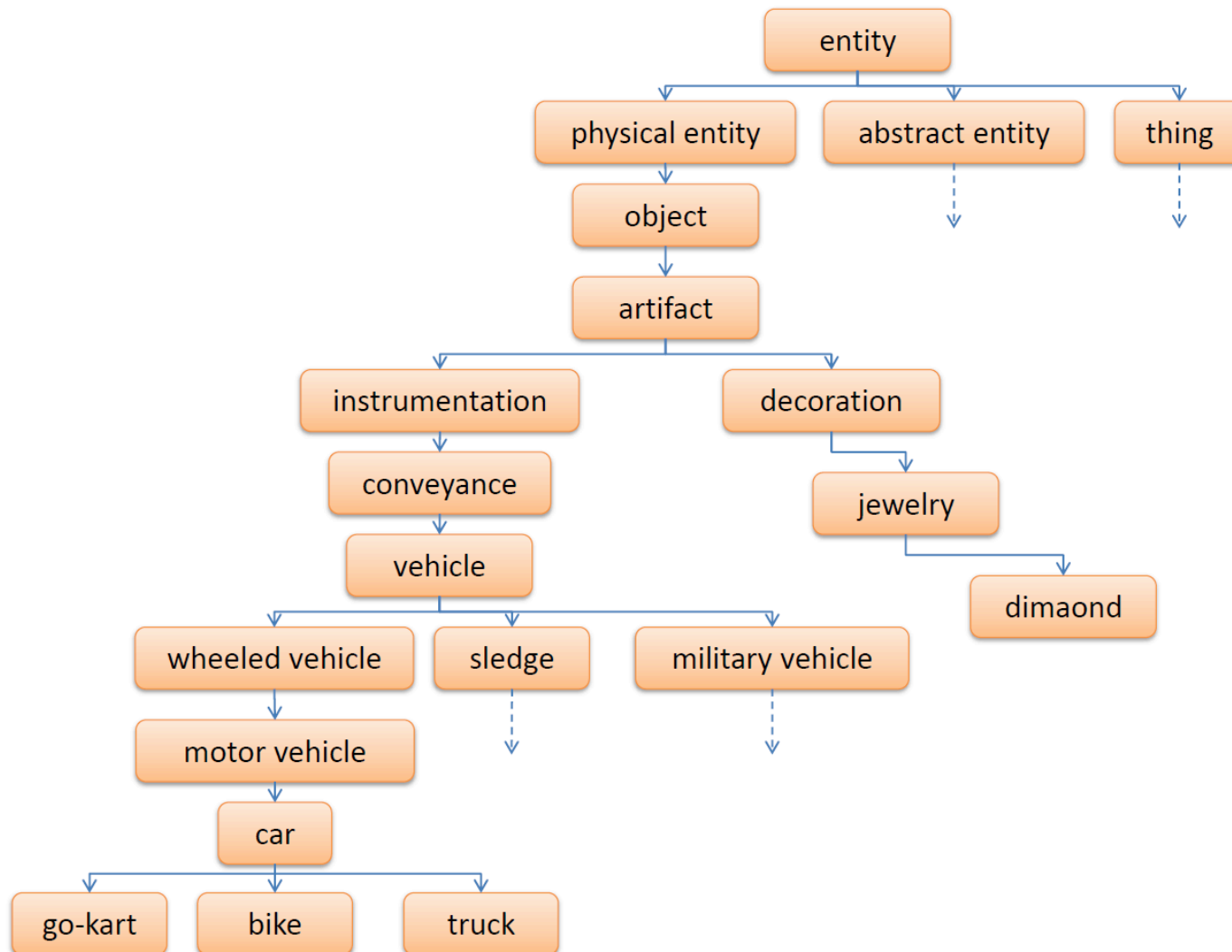
Tied to sense inventories: graph distance

WordNet as a graph



Sense Similarity Techniques

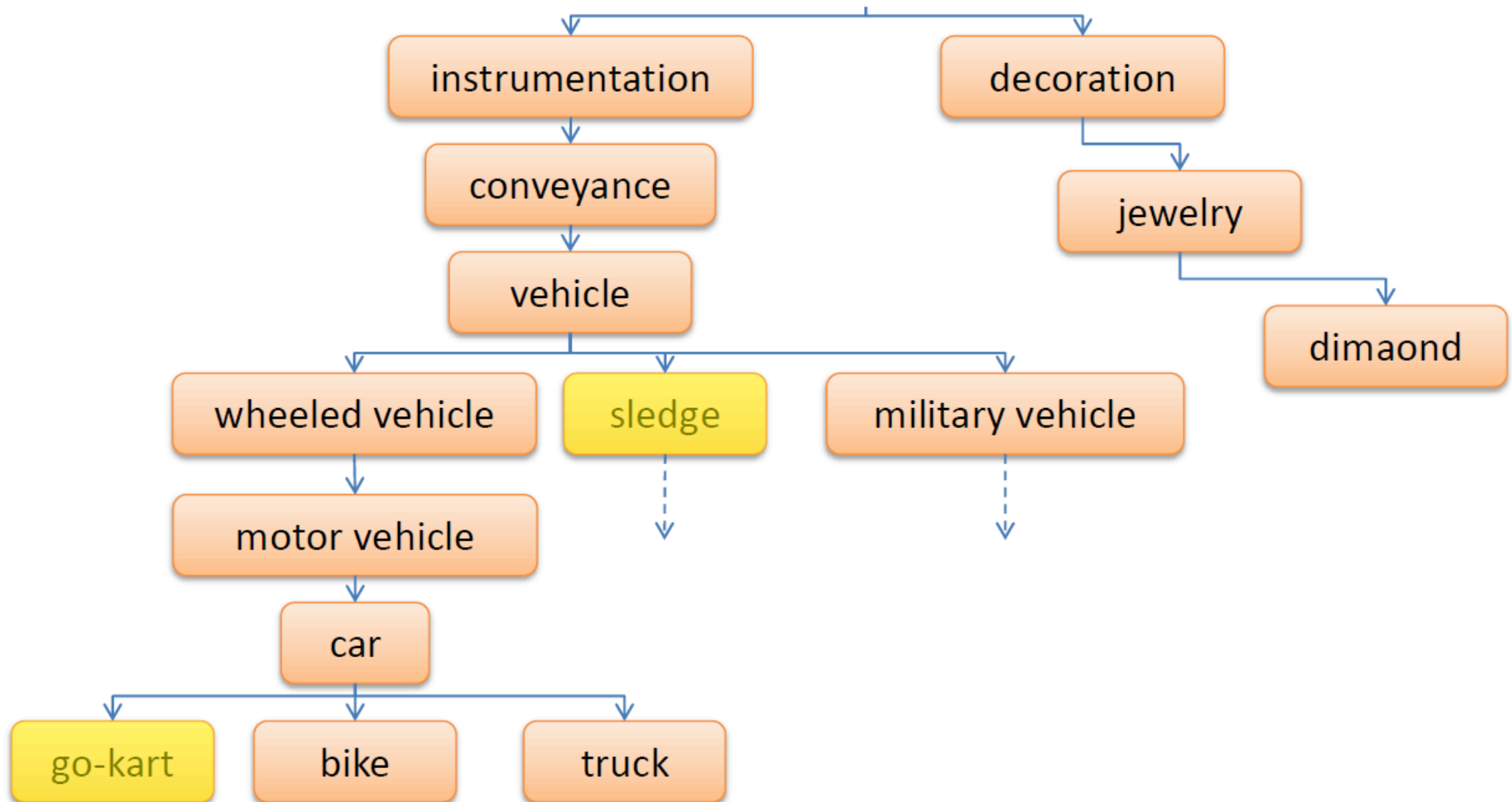
Tied to sense inventories: WordNet graph distance



Sense Similarity Techniques

Tied to sense inventories: WordNet graph distance

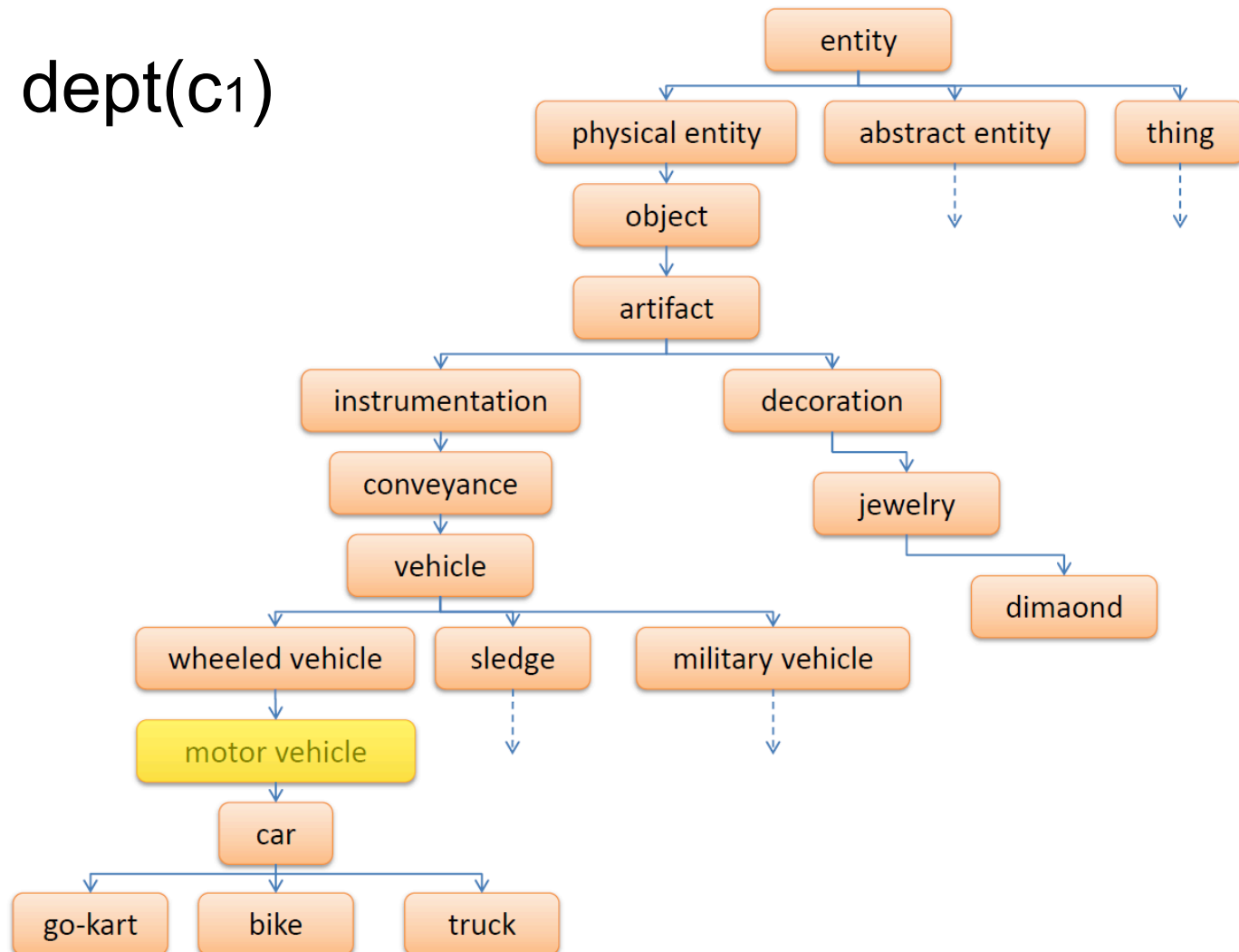
$\text{len}(c_1, c_2)$



Sense Similarity Techniques

Tied to sense inventories: WordNet graph distance

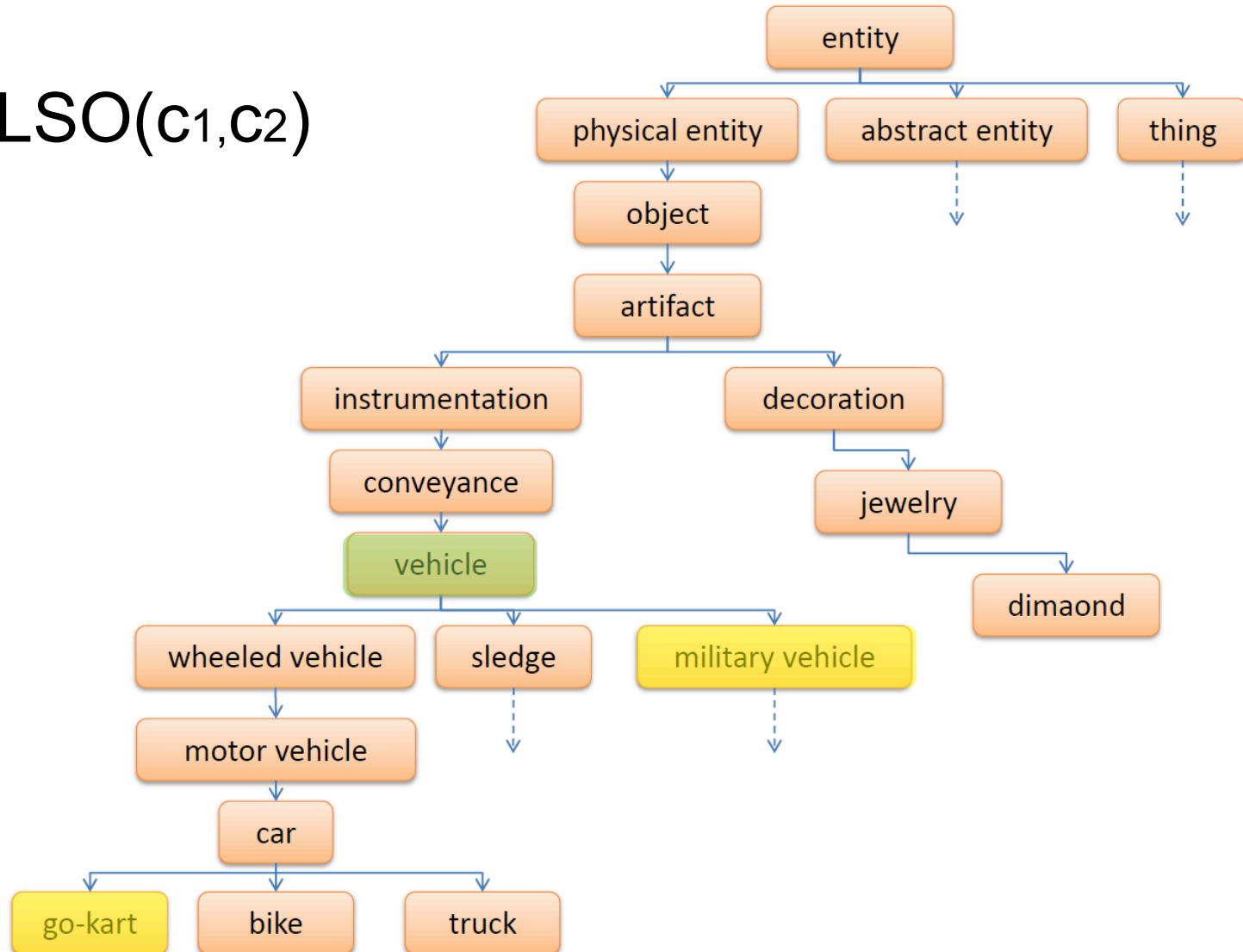
dept(c₁)



Sense Similarity Techniques

Tied to sense inventories: WordNet graph distance

$LSO(c_1, c_2)$



Sense Similarity Techniques

Tied to sense inventories: WordNet graph distance

Conventional WordNet-based techniques

Survey: Budanitsky and Hirst (2006)

WordNet structure only

Hirst and St-Onge (1998)

Sussna's Depth-relative Scaling (1993, 1997)

Wu and Palmer (1994)

Leacock and Chodorow's (1998)

Combined with statistics from corpora

Jiang and Conrath's Measure (1997)

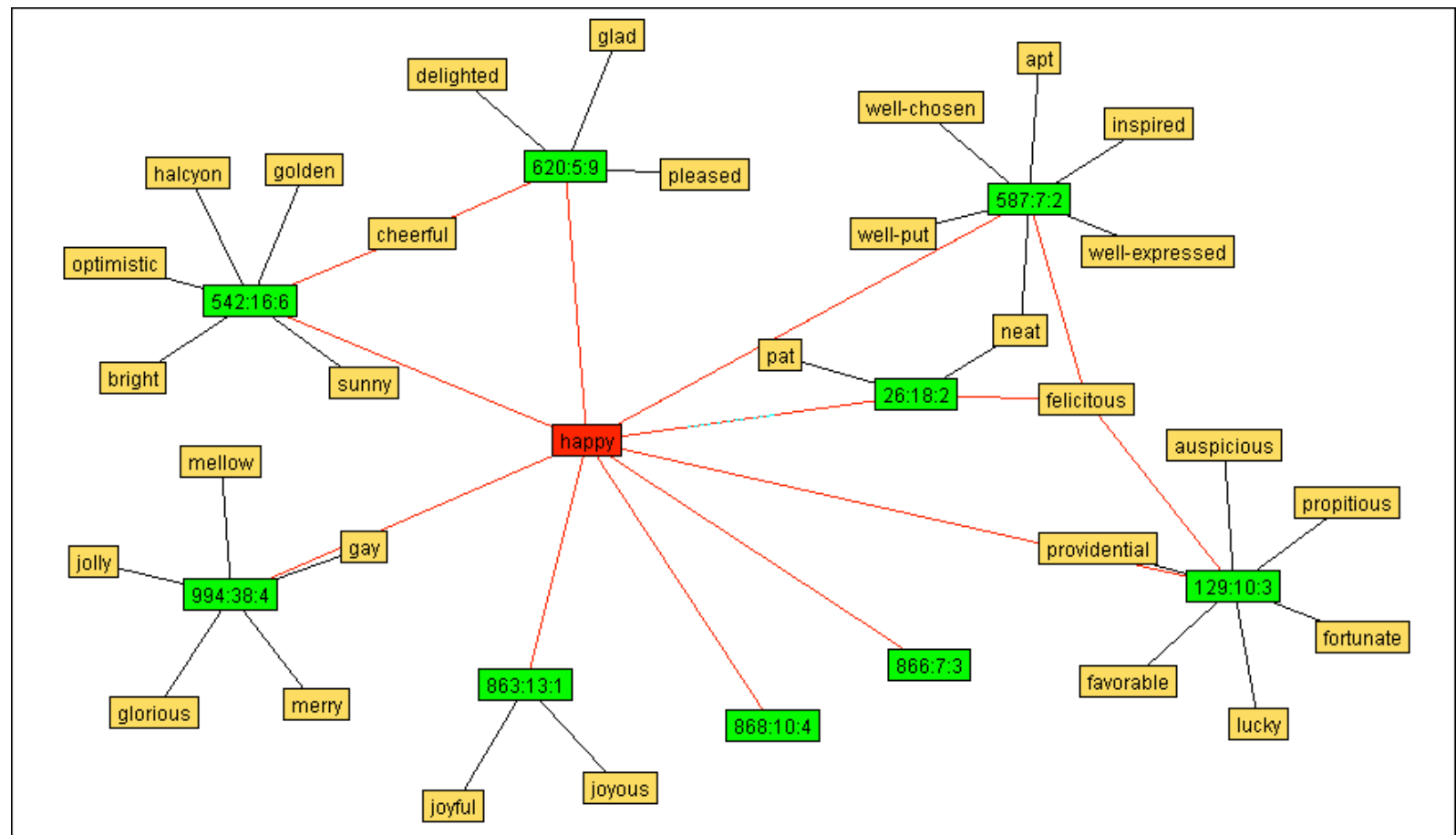
Resnik (1995)

Lin's Measure (1998)

Sense Similarity Techniques

Tied to sense inventories: Thesauri-based

Roget's thesaurus: Morris and Hirst (1991), Jarmasz and Szpakowicz (2003)



Sense Similarity Techniques

Tied to sense inventories: Dictionary-based

Longman Dictionary (LDOCE): Kozima and Furugori (1993), Kozima and Ito (1997)

- Constructs a semantic network from a subset of the dictionary, 2851 nodes, called Paradigme
- Computes similarity by spreading the activation in the network

Sense Similarity Techniques

Tied to sense inventories

Explicit semantic representation

Sense Similarity Techniques

Tied to sense inventories: Explicit semantic representation

Simple gloss-based: Exploiting WordNet's content

application#n#2 --

a verbal or written request for assistance or employment or admission to a school

application#n#4 --

a program that gives a computer instructions that provide the user with tools to accomplish a task

example:

Meerkat Mafia – Kashyap et al (2014)

@ SemEval-2014 Task-3: CLSS

Sense Similarity Techniques

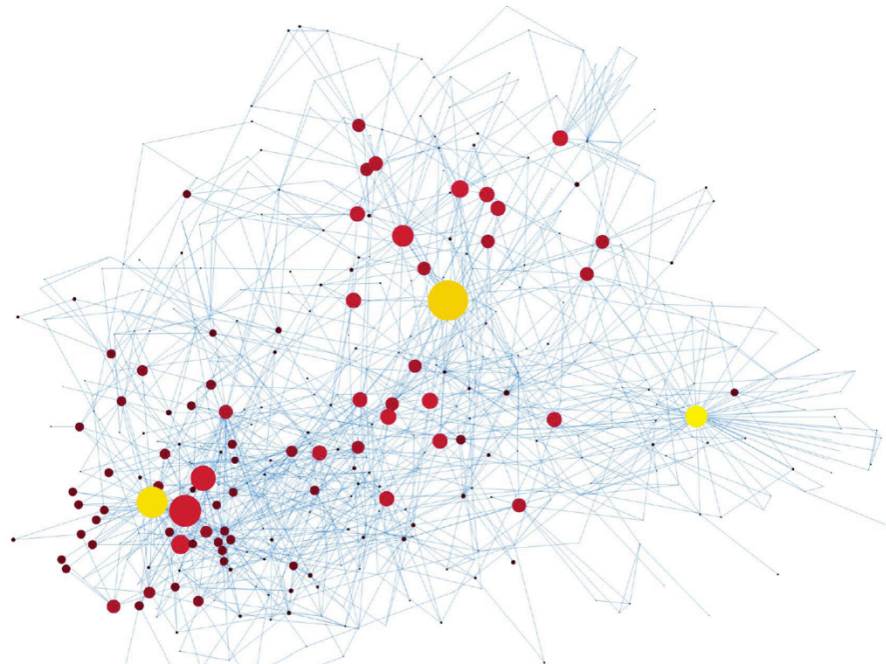
Tied to sense inventories: Explicit semantic representation

Random walks of semantic networks

The Personalized PageRank algorithm

Semantic similarity: Pilehvar et al (2013)

WSD: Agirre et al (CL 2014)



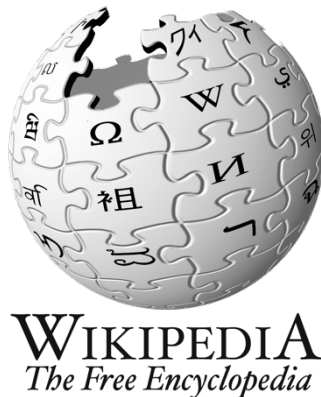
Sense Similarity Techniques

Tied to sense inventories: Explicit semantic representation

Distributional

SensEmbed - word2vec sense embeddings

Iacobacci et al (2015)



+

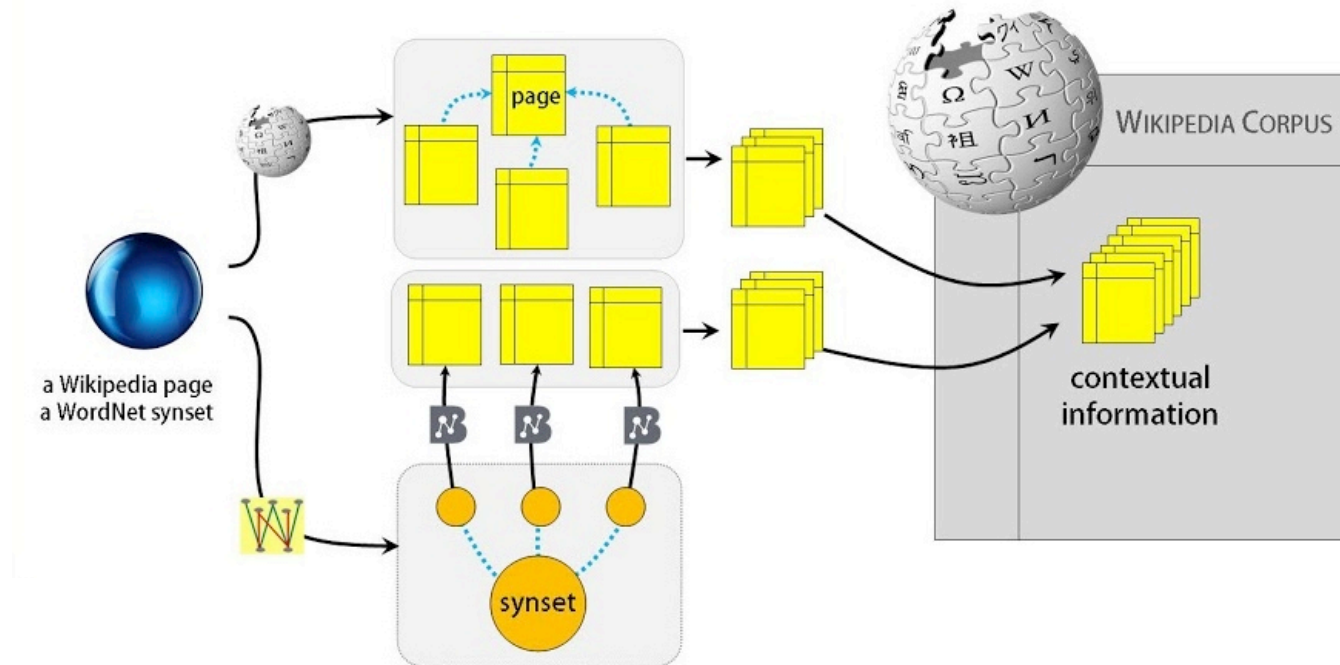


Sense Similarity Techniques

Tied to sense inventories: Explicit semantic representation

Distributional

NASARI and MUFFIN - Camacho-collados et al (2015)



Sense Similarity Techniques

Tied to sense inventories: Explicit semantic representation

Distributional

- Chen et al (emnlp 2014)

 - Joint word sense representation and disambiguation

- Learn word representations (word2vec skip-gram)
- Use them for sense representation (average gloss)
- Automatically disambiguate large amounts of text
- Modify the objective of Skip-gram to learn sense representations

Sense Similarity Techniques

Tied to sense inventories: Explicit semantic representation

Distributional

- Rothe and Schutze (acl 2015)

Extends word embeddings (word2vec) to embeddings of other data types: WordNet synsets and word senses

Constructs an auto-encoder

learns these representations based on WordNet constraints (word/synset is the summation of its lexemes + WN relations)

Sense Similarity Techniques

Not Tied to sense inventories

Also called

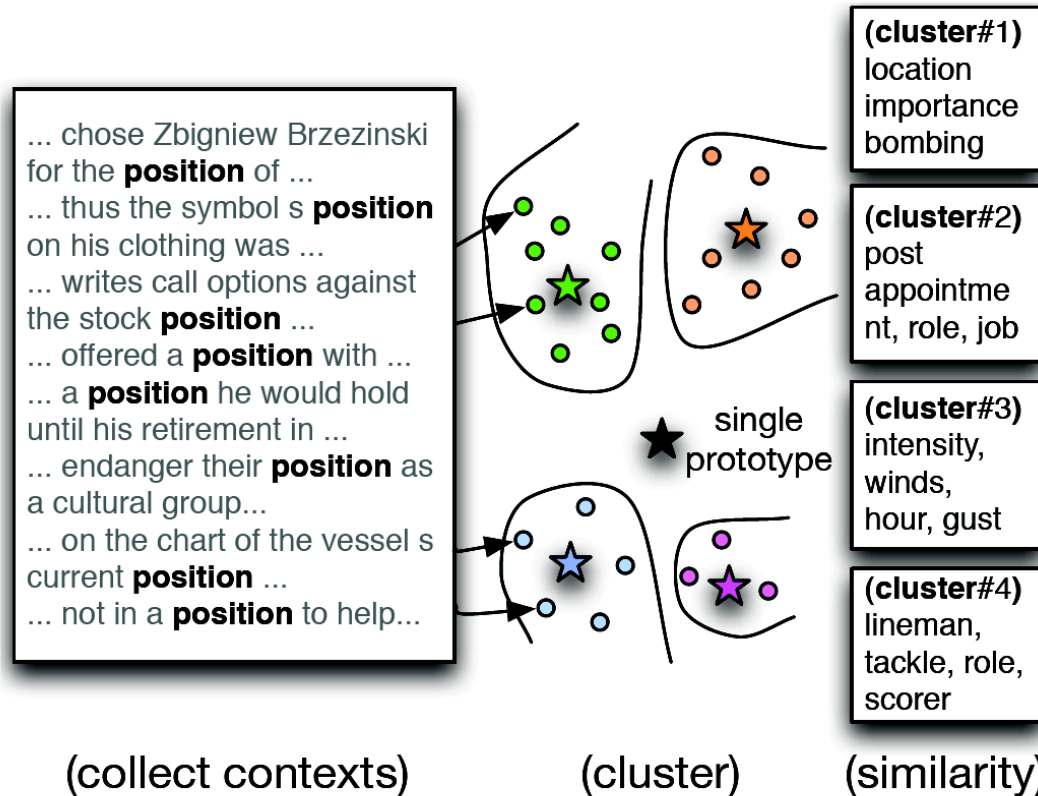
multi-prototype or topic-based representations

Usually based on clustering

Sense Similarity Techniques

Not Tied to sense inventories

Reisinger and Mooney (2010)



Sense Similarity Techniques

Not Tied to sense inventories

Reisinger and Mooney (2010)

Measuring similarity - isolated words:

$$\text{AvgSim}(w, w') \stackrel{\text{def}}{=} \frac{1}{K^2} \sum_{j=1}^K \sum_{k=1}^K d(\pi_k(w), \pi_j(w'))$$

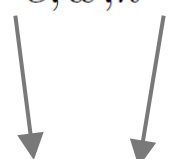
$$\text{MaxSim}(w, w') \stackrel{\text{def}}{=} \max_{1 \leq j \leq K, 1 \leq k \leq K} d(\pi_k(w), \pi_j(w'))$$

Sense Similarity Techniques


Not Tied to sense inventories

Reisinger and Mooney (2010)

Measuring similarity - words in contexts:

$$\text{AvgSimC}(w, w') \stackrel{\text{def}}{=} \frac{1}{K^2} \sum_{j=1}^K \sum_{k=1}^K d_{c,w,k} d_{c',w',j} d(\pi_k(w), \pi_j(w'))$$


likelihood of the cluster given the context

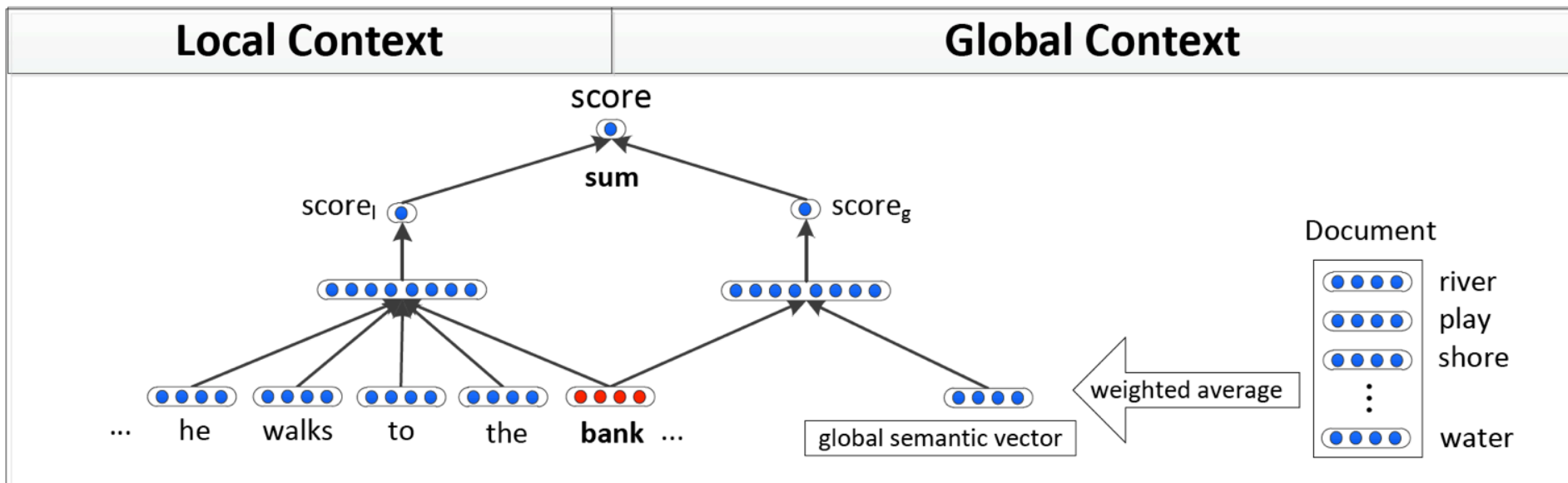

$$\text{MaxSimC}(w, w') \stackrel{\text{def}}{=} d(\hat{\pi}(w), \hat{\pi}(w'))$$

Sense Similarity Techniques

Not Tied to sense inventories

Huang et al (2012)

- Learns word embeddings with local and global objectives
- Then clusters the contexts of a word and learns multi-prototype representations



Sense Similarity Techniques

Not Tied to sense inventories

Neelakantan et al (emnlp 2014)

Multi-Sense Skip-gram (MSSG) model

(fixed number of senses)

Sense discrimination and learning embeddings are performed jointly

by disambiguating a word using current parameters

Non-parametric MSSG model

(varying number of senses per word)

Different in the sense discrimination phase

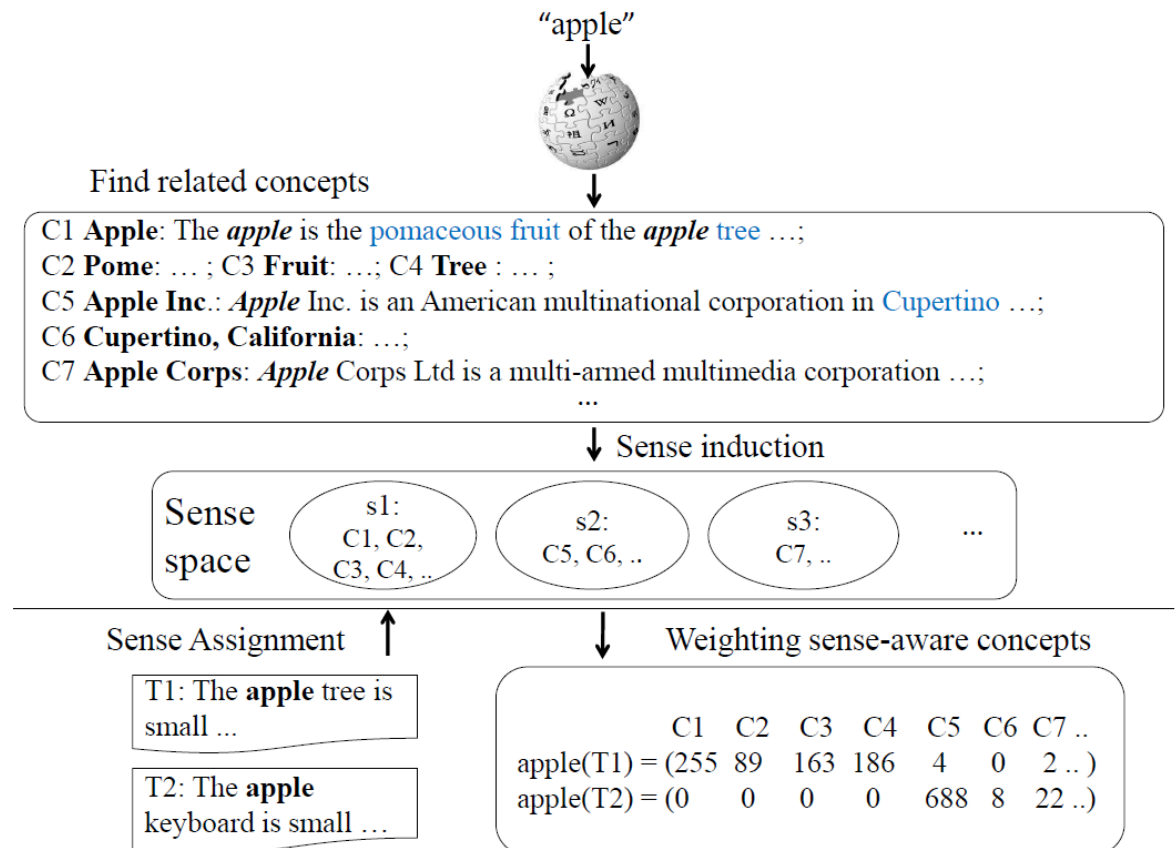
Online non-parametric clustering

Sense Similarity Techniques

Not Tied to sense inventories

SaSA - Sense-aware Semantic Analysis

Wu and Giles (AAAI 2015)



Sense Similarity Techniques

Not Tied to sense inventories

Topical Word Embeddings - Liu et al (AAAI 2015)

Different senses of a word can overlap
-> soft clustering

Uses LDA to learn representations for <word,topic> pairs

$$\mathcal{L}(D) = \frac{1}{M} \sum_{i=1}^M \sum_{-k \leq c \leq k, c \neq 0} \log \Pr(w_{i+c} | w_i) + \log \Pr(w_{i+c} | z_i).$$

Sense Similarity

Evaluation benchmarks

- Word similarity
 - and all other word-level applications
- Sense merging
- Word Sense Disambiguation
- Stanford's Contextual Word Similarities (SCWS)
- Cross Level Semantic Similarity
(more details to follow)

Word Similarity

Word similarity is a lot like sense similarity

He went to the **ATM** to deposit the money.

She goes to the **bank** to withdraw cash.

Word similarity is a lot like sense similarity ... except for ambiguity

He went to the **ATM** to deposit the money.

She goes to the **bank** to withdraw cash.

She goes to the **shore** near the silt deposit.

Word similarity is a lot like sense similarity ... except for ambiguity

He went to the **ATM** to deposit the money.

She goes to the **bank** to withdraw cash.

She goes to the **shore** near the silt deposit.

**Most approaches measure
similarity completely out of context.**

Word similarity lets you easily build to larger linguistic level's similarities

The boy sailed the boat over the ocean.



The girl navigates the sailboat across the sea.

Many applications benefit from having word representations that encode similarity or having effective word similarity functions.

Text classification (Baker and McCallum, 1998)

Document classification (Sebastiani et al, 2002)

Question answering (Tellex et al, 2003)

IR (Sanderson, 1994), Manning et al (2008)

Textual entailment (Baroni, 2014 - SICK)

Named entity recognition (Turian et al, 2010, Passos et al, 2014)

Dependency parsing (Bansal et al, 2014)

Chunking (Turian et al, 2010, Dhillon and Ungar, 2011)

Paraphrase detection (Socher et al, 2011)

**Ideal references for comparing
impact of new approaches**

Most approaches evaluate on similarity benchmarks, rather than tasks

Numeric Word-Pair Similarity Tests

- Rubenstein & Goodenough, 1965 (RG)
- WordSim-353 (Finkelstein et al., 2001)
- Rare Words (Luong et al., 2013)
- MEN (Bruni et al., 2012)
- Radinsky et al., (2010)

Word Choice Tests

- TOEFL, ESL, Reader's Digest

TOEFL Synonymy recognition

enormous?
☐ appropriate
☐ unique
☒ tremendous
☐ decided

RG-65 judgement correlation

autograph	shore	0.06
coast	forest	0.85
midday	noon	3.94

Stanford Rare Word (RW) judgement correlation

dispossess	deprive	6.83
entrapping	capture	8.00
ruralist	advocate	0.67
acoustical	remedy	0.14
quieten	hush	9.38

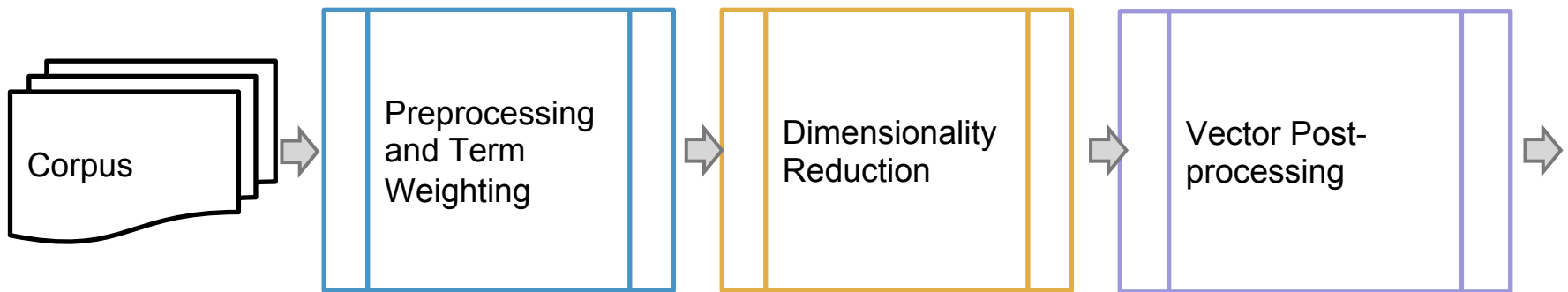
**What if we know nothing
about the words?**



**You shall know a word by
the company it keeps**

-- Firth (1957)

High-level schematic of corpus-based distributional learning approaches



Three Typical Setups: Term-Term, Term-Context or Term-Document Matrix

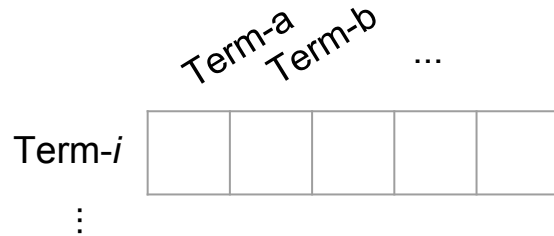
Cells record the number of times...

	Term-a	Term-b	...		
Term- i					
\vdots					

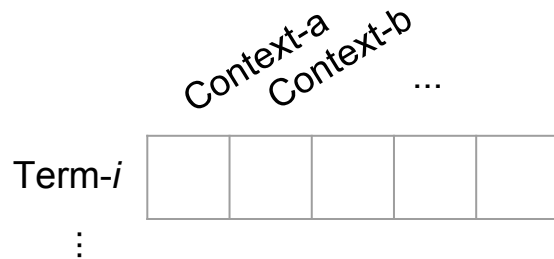
term j occurs in the context window of term i .

Three Typical Setups: Term-Term, Term-Context or Term-Document Matrix

Cells record the number of times...



term j occurs in the context window of term i .

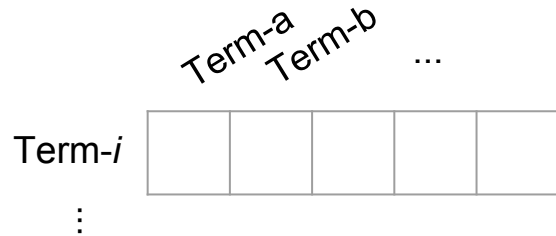


term i occurs in a particular context type

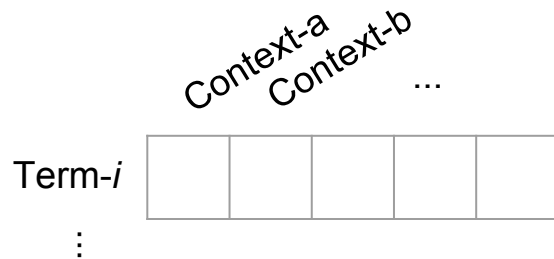
- $w_{-2}, w_{-1}, w, w_1, w_2$
- or similar, with dependencies

Three Typical Setups: Term-Term, Term-Context or Term-Document Matrix

Cells record the number of times...

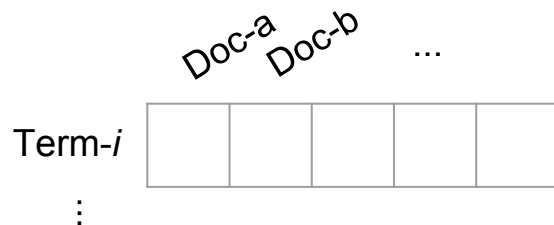


term j occurs in the context window of term i .



term i occurs in a particular context type

- $W_{-2}, W_{-1}, w, W_1, W_2$
- or similar, with dependencies



term i occurs in document j .

Raw word co-occurrence is rarely satisfactory as a representation

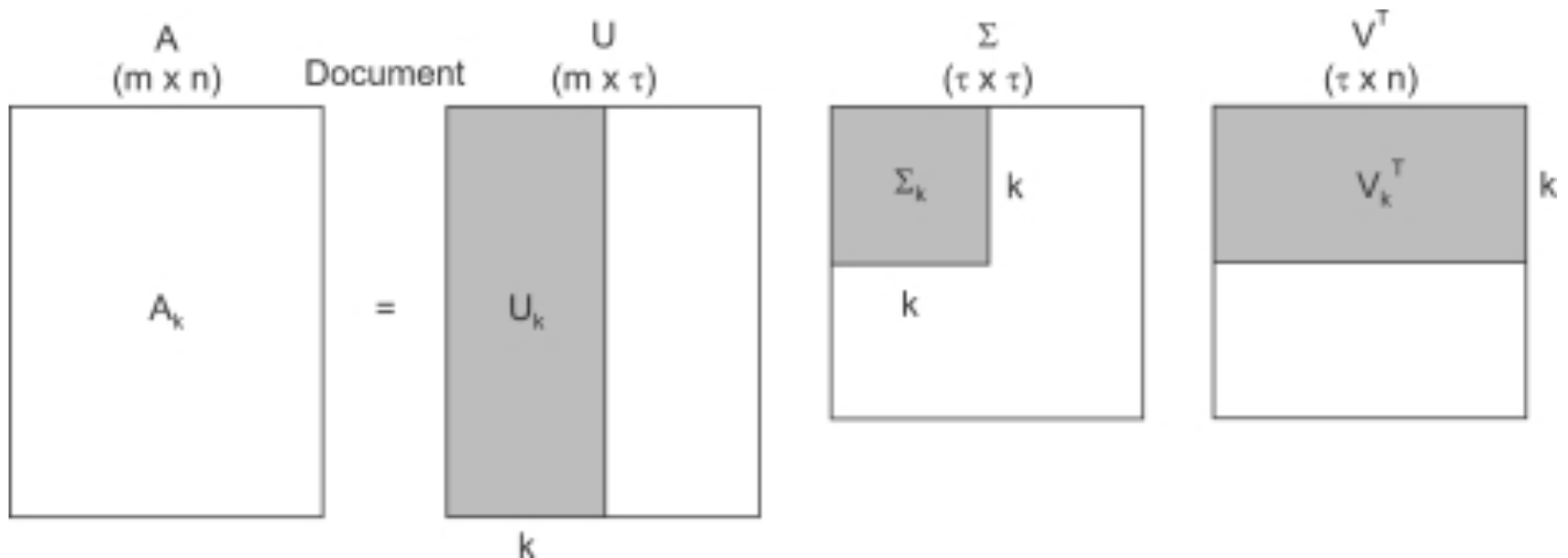
All words are treated as equally informative
the, big, metallic, biophosphorescence

Vector length is proportional to vocabulary
size

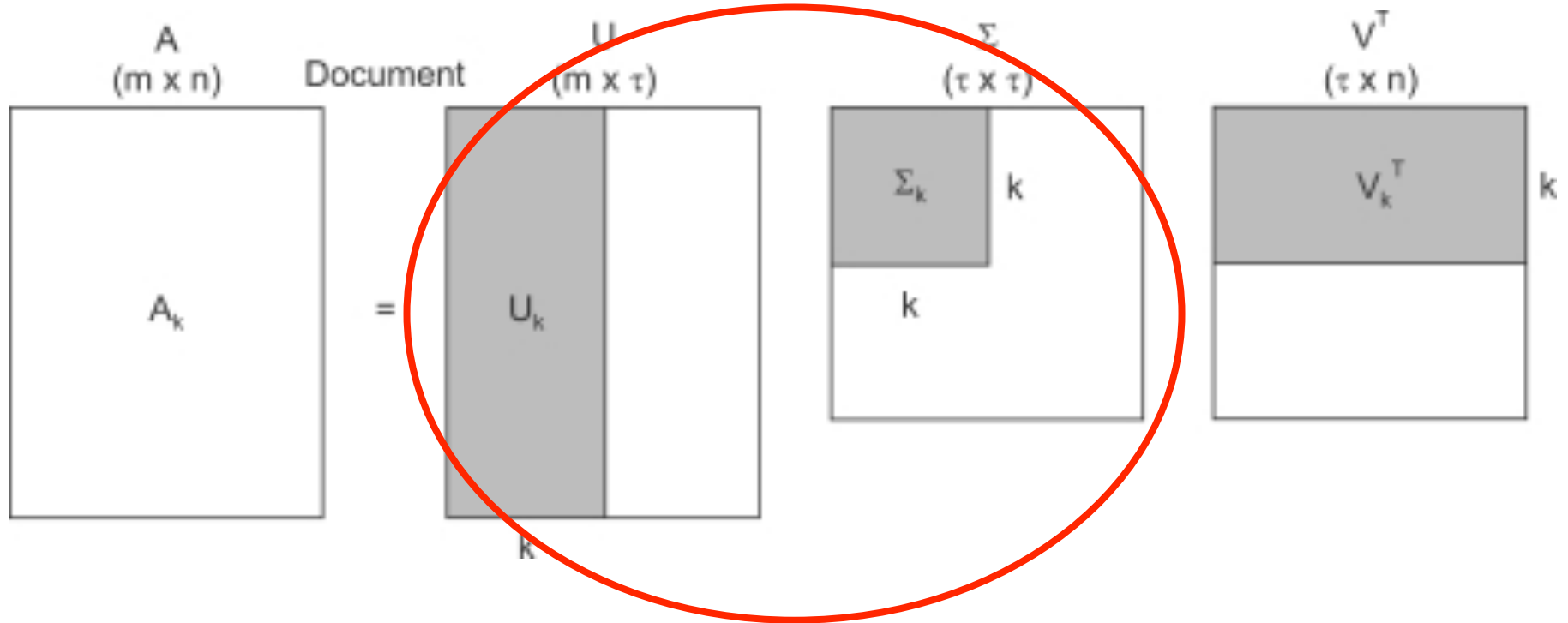
Eventually issues with computation and space

Infrequent words have overly-sparse vectors

Standard Approach: Reduce the dimensionality using the Singular Value Decomposition (SVD)



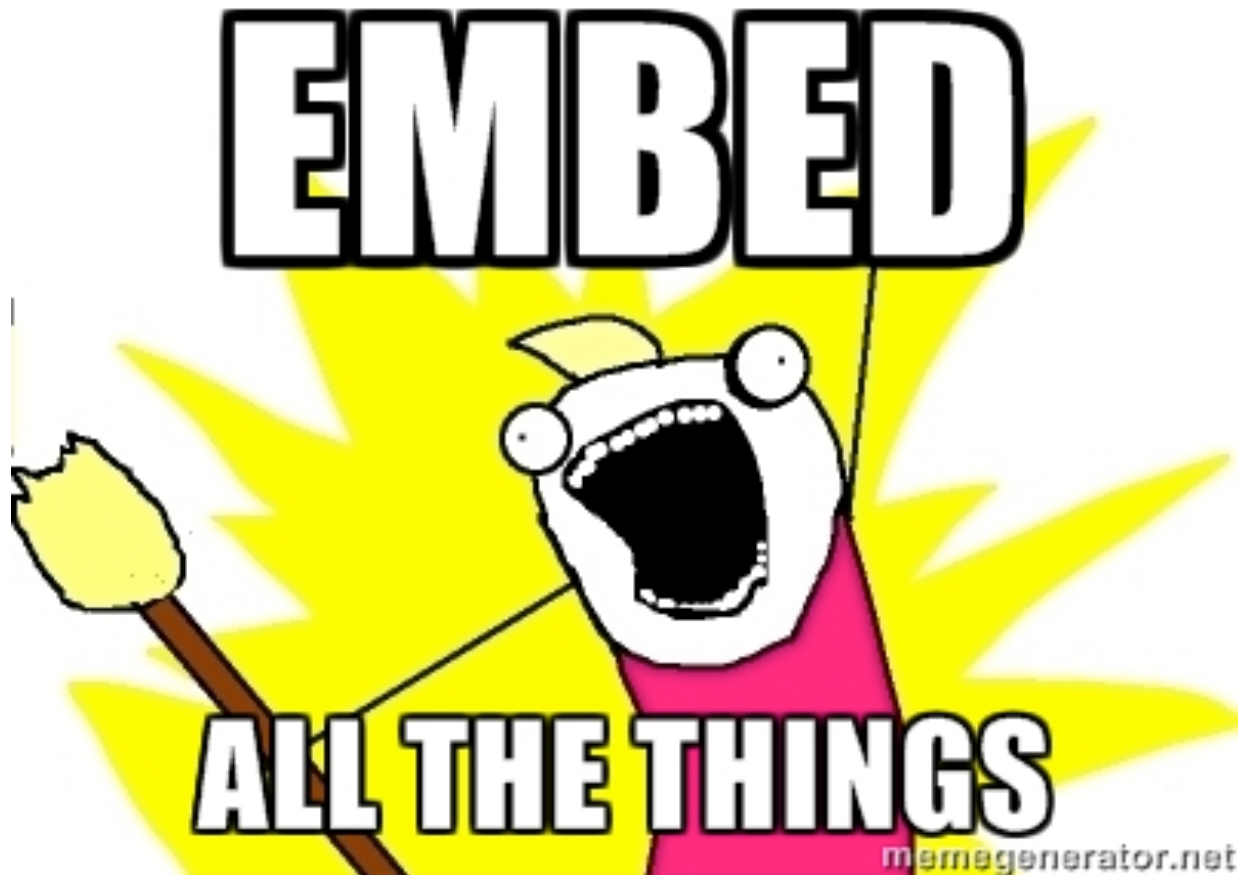
Standard Approach: Reduce the dimensionality using the Singular Value Decomposition (SVD)



Typically, $U * \Sigma$ is used as the vector space.

State of the Art: Reduce dimensionality with Neural Embeddings (word2vec)

also known as



word2vec

More a software system than an algorithm

Training methods

- Negative Sampling**

- Hierarchical Softmax

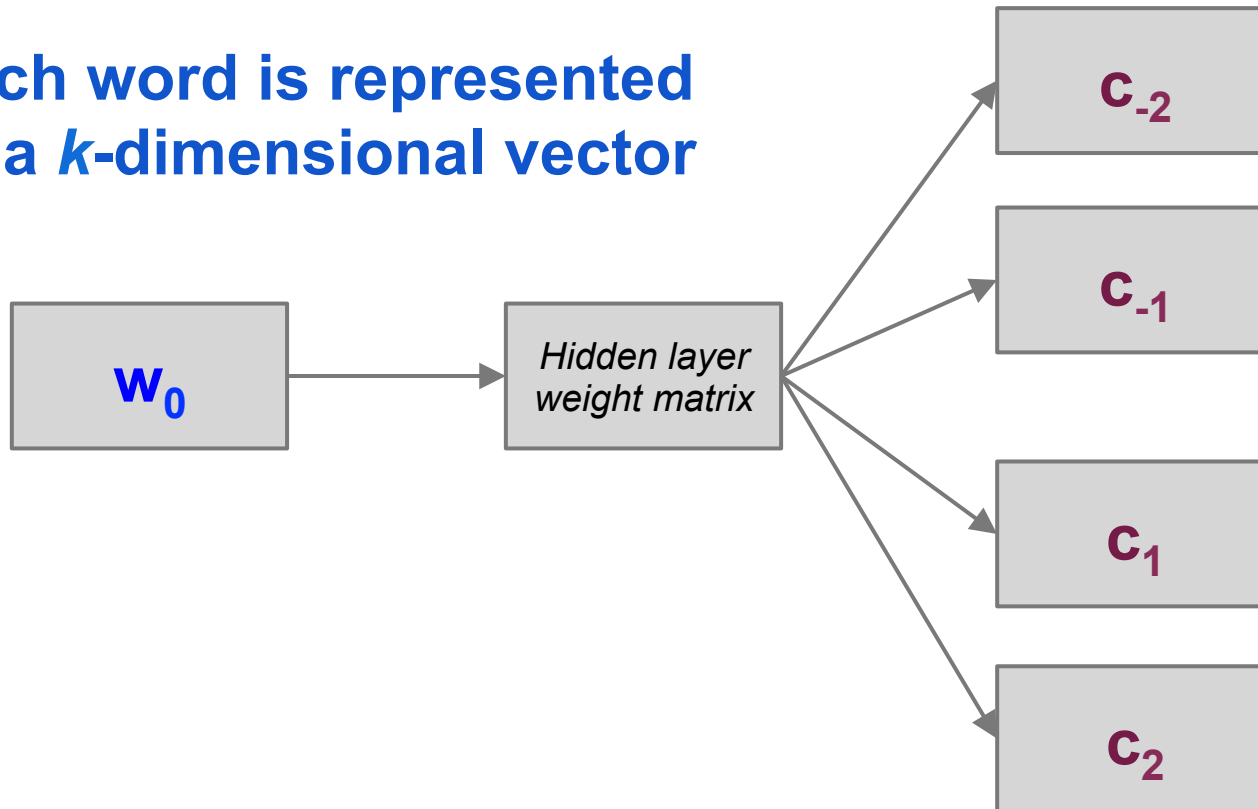
Context representations

- Continuous Bag of Words (CBow)

- Skip grams**

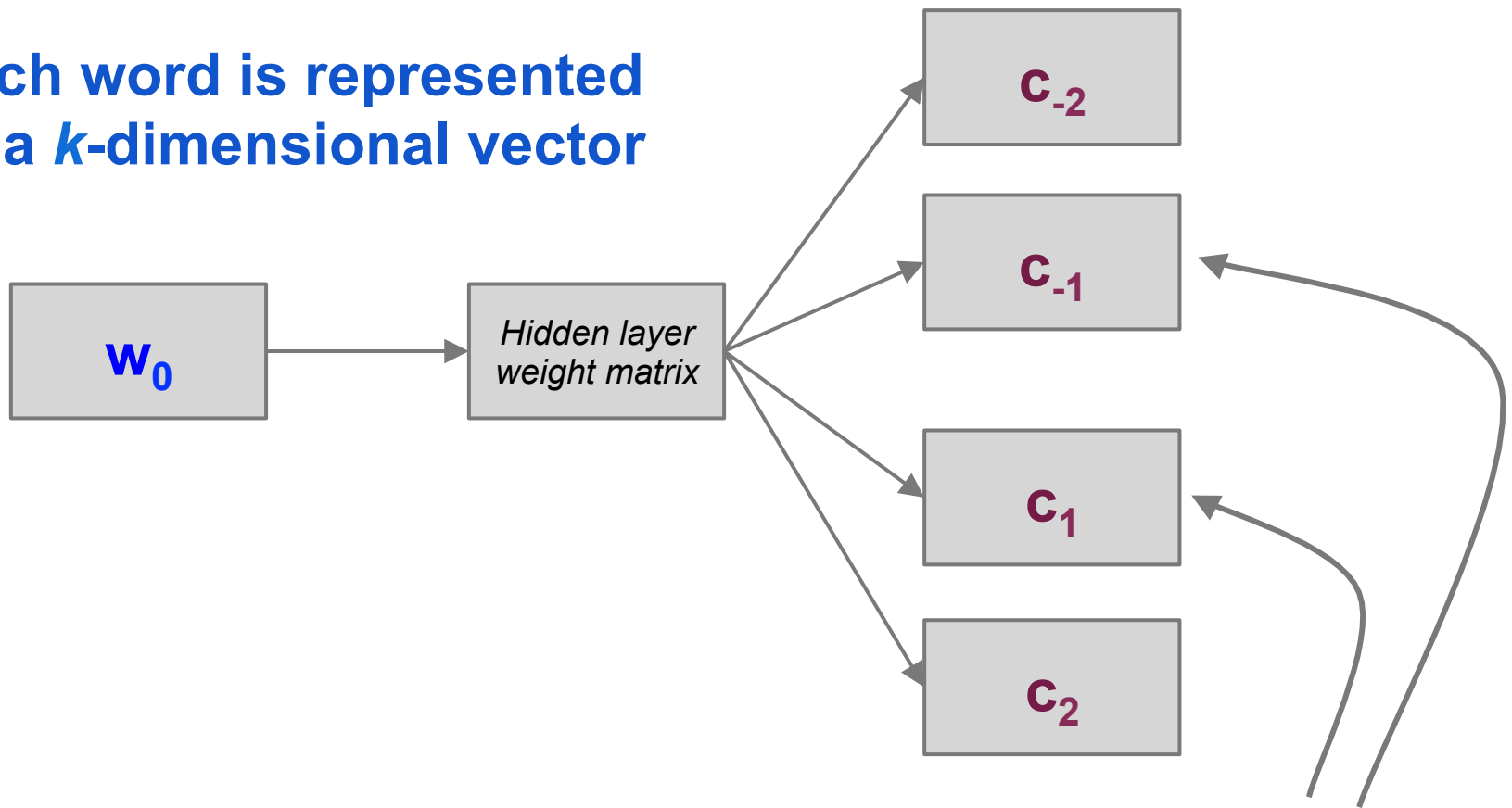
word2vec: a neural look

Each word is represented
as a k -dimensional vector



word2vec: a neural look

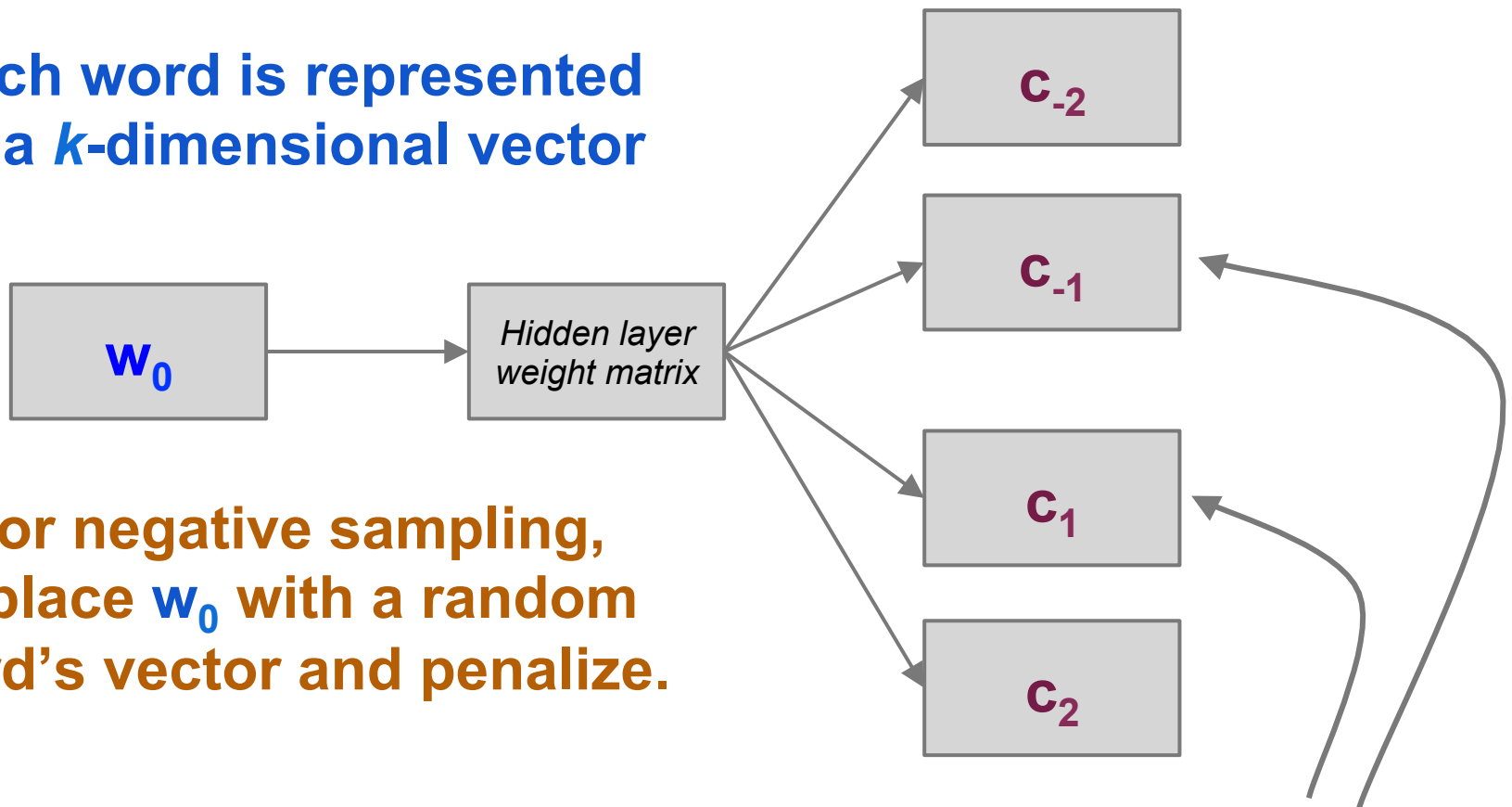
Each word is represented
as a k -dimensional vector



The system is trained to predict the representations
for context words before and after

word2vec: a neural look

Each word is represented
as a k -dimensional vector



For negative sampling,
replace w_0 with a random
word's vector and penalize.

The system is trained to predict the representations
for context words before and after

word2vec \cong implicitly factorizing PMI-weighted word-context matrix

**Key Implication: word2vec is building upon existing
techniques by using a new decomposition**

Huge gains from using embeddings!

	RG	WordSim	MEN	TOEFL
PMI+SVD	.70	.70	.72	.76
word2vec	.83	.78	.80	.86

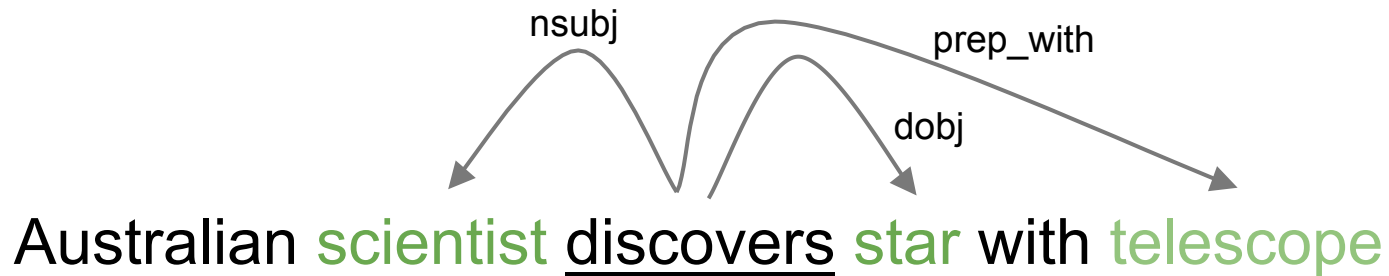
Performance improvement over SVD-based methods is consistent across many tasks*

Could we get better performance with syntactic contexts?

Australian scientist discovers star with telescope

Could we get better performance with syntactic contexts?

Australian scientist discovers star with telescope



Dependency-based embeddings capture functional information

Target Word	BoW5	BoW2	DEPS
batman	nightwing aquaman catwoman superman manhunter	superman superboy aquaman catwoman batgirl	superman superboy supergirl catwoman aquaman
hogwarts	dumbledore hallows half-blood malfoy snape	evernight sunnydale garderobe blandings collinwood	sunnydale collinwood calarts greendale millfield
turing	nondeterministic non-deterministic computability deterministic finite-state	non-deterministic finite-state nondeterministic buchi primality	pauling hotelling heting lessing hamming

No quantitative results on standard benchmarks

Glove: capture the ratio of co-occurrence probabilities

word2vec: $\vec{w} \cdot \vec{c}^T = \text{pmi}(w, c) - \log k$

GloVe: $\vec{w} \cdot \vec{c}^T \cdot b_w \cdot b_c = \log(\#(w, c))$

Key insight: the context vector provides insight into so a word representation is $w + c$

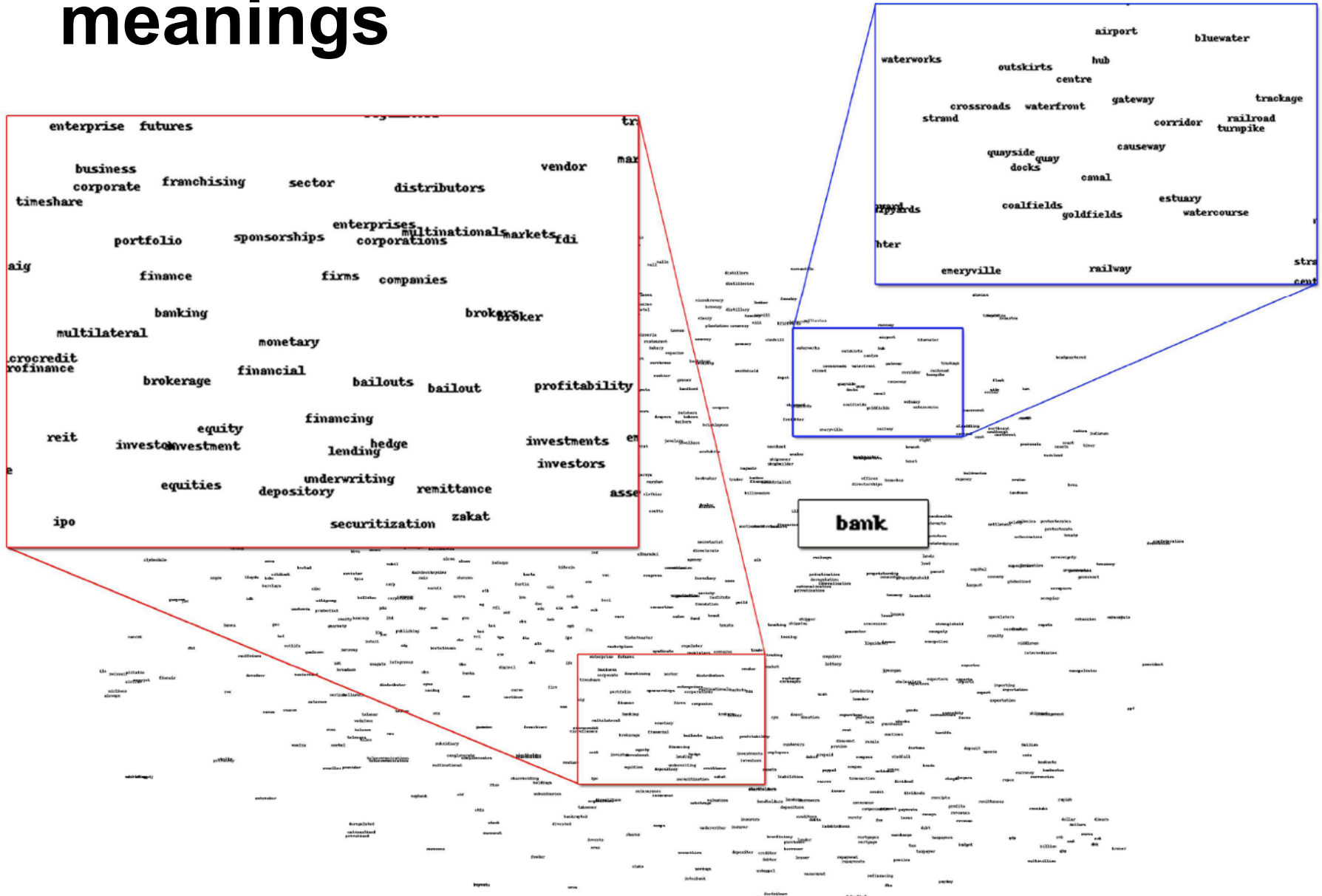
GloVe had initially impressive performance at word similarity

	MC	RG	SCWS	Rare Words
SVD	.727	.751	.565	.370
word2vec	.652	.697	.581	.372
GloVe	.727	.778	.529	.381

However under equivalent conditions, word2vec performs better

	Word Sim	MEN	Rad	Rare Words	SimLex
PPMI	.755	.745	.686	.462	.393
PMI+SVD	.793	.778	.666	.514	.432
word2vec	.793	.774	.693	.470	.438
GloVe	.725	.729	.632	.403	.398

Regular embeddings still conflate meanings



Incorporating senses* seems to improve performance

	SCWS	RG	MEN	SimLex
word2vec	.657	.694	.707	.311
Gaussian Embeddings (Vilnis and McCallum, 2015)		.710	.713	.322
TWI (Liu et al. 2015)	.681			

But results vary based on test setup

	SCWS	RG	MEN	SimLex
word2vec	.657	.694	.707	.311
Gaussian Embeddings (Vilnis and McCallum, 2015)		.710	.713	.322
TWI (Liu et al. 2015)	.681			

	SCWS	WordSim	MEN	SimLex
PMI+SVD		.793	.778	.432
word2vec	.581	.793	.774	.438

**Many other sense-based embeddings
never evaluate on similarity**

(Pennington et al., 2014;
Levy and Goldberg, 2015)

Results suggest that more dimensions in word vectors can compensate for conflating meanings

	NER	Semantic Relatedness	Sentiment
word-embeddings (50 dims)	.852	.748	.747
sense-embeddings (50 dims)	.854	.762	.750
word-embeddings (100 dims)	.867	.770	.763

Learning-Approach Recap

Nothing magic in the representation
similar to SVD with PMI-weighted matrix

word2vec state of the art for most use cases

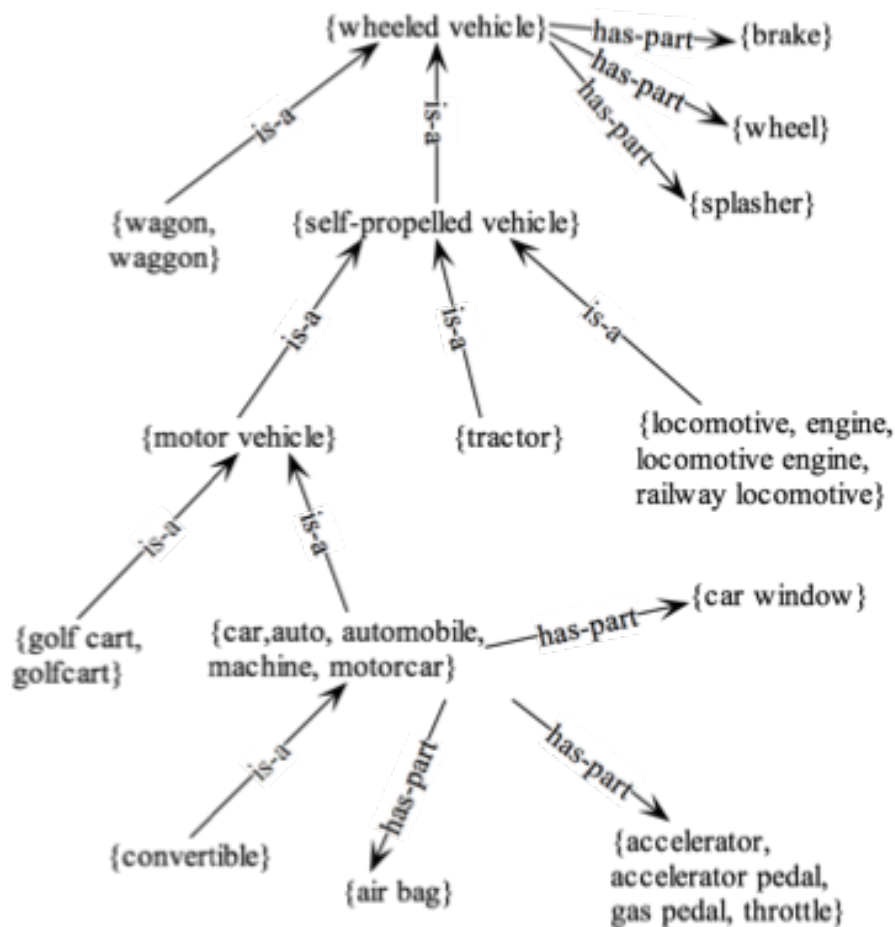
But dependency-based relations may be useful in
some circumstances

Also, one of the fastest to train

Sense-aware representations have a yet to
show a clear benefit

**What if we already know
something about the
words?**

The structure of WordNet, Wikipedia, and other knowledge bases can be used to measure word similarity



Great for when you need a similarity value

Not as great when you need a representation to use, unless you create one

Wikipedia links create a knowledge graph with edges between related pages

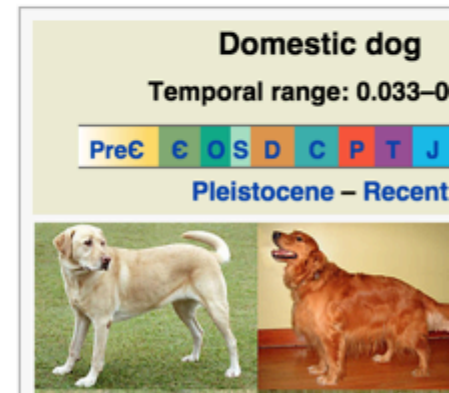
Dog

From Wikipedia, the free encyclopedia

This article is about the domestic dog. For related species known as "dogs", see [Canidae](#). For other uses, see [Dog \(disambiguation\)](#). "Doggie" redirects here. For the Danish artist, see [Doggie \(artist\)](#).

The **domestic dog** (*Canis lupus familiaris* or *Canis familiaris*) is a [domesticated canid](#) which has been selectively bred for millennia for various behaviors, sensory capabilities, and physical attributes.^[2]

Although initially thought to have originated as a manmade variant of an extant canid species (variously supposed as being the [dhole](#),^[3] [golden jackal](#),^[4] or [gray wolf](#)^[5]), extensive genetic studies undertaken during the 2010s indicate that dogs diverged from other [wolf-like canids](#) in [Eurasia](#) 40,000 years ago.^[6] Being [the oldest domesticated animals](#), their long association with people has allowed dogs to be uniquely attuned to human behavior,^[7] as well as thrive on a [starch-rich](#) diet which would be inadequate for other canid species.^[8]



**Ideal for path-based measures of similarity
and for random walks!**

WikiRelate: Apply WordNet measures on Wikipedia's graph

Best results with Leacock & Codorow's method:

$-\log(\text{path_length}(\text{page}_1, \text{page}_2) / \text{max_depth})$

	RG	MC	WordSim-353
L&C (Wikipedia)	.41	.54	.48

WikiRelate: Apply WordNet measures on Wikipedia's graph

Best results with Leacock & Codorow's method:
 $-\log(\text{path_length}(\text{page}_1, \text{page}_2) / \text{max_depth})$

	RG	MC	WordSim-353
L&C (Wikipedia)	.41	.54	.48
L&C (WordNet)	.82	.86	.34

Large amount of noise in Wikipedia's graph creates issues for similarity-specific calculations. I.e., difficult to tell edges and nodes are important?

Idea: Identify important pages in Wikipedia using Personalized PageRank

- Given a page p , find all wiki-linked pages to p and initialize the PPR vector to these pages
 - Optionally prune (a) pages with spaces in the name and (b) pages account for fewer than $x\%$ of the links
- Run PPR and compare vectors

Idea: Identify important pages in Wikipedia using Personalized PageRank

- Given a page p , find all wiki-linked pages to p and initialize the PPR vector to these pages
 - Optionally prune (a) pages with spaces in the name and (b) pages account for fewer than $x\%$ of the links
- Run PPR and compare vectors

	MC	WordSim-353
PPR	.60	.45
WikiRelate	.54	.48

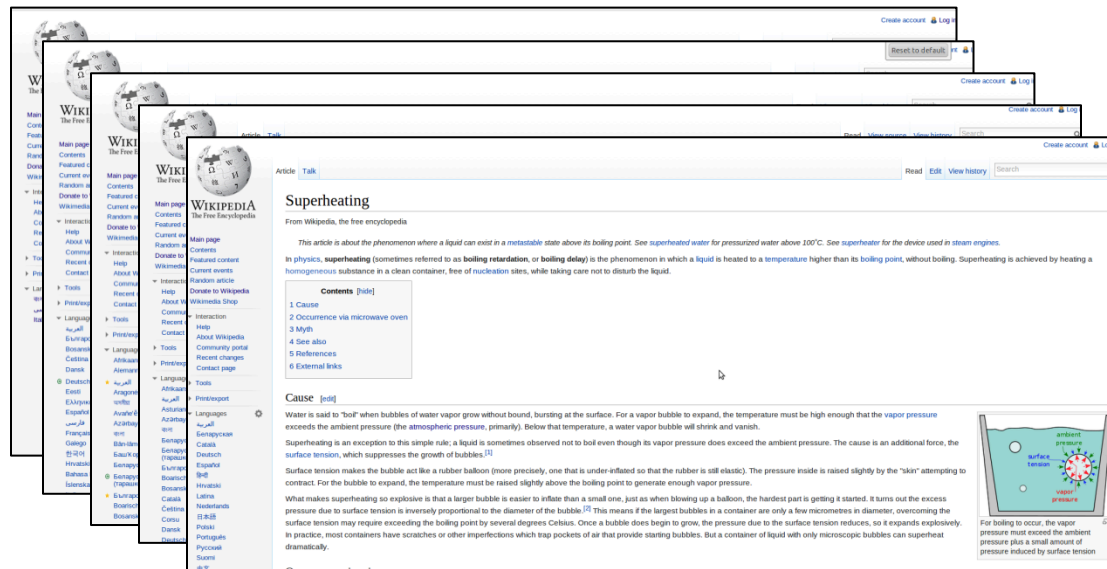
Idea: Identify important pages in Wikipedia using Personalized PageRank

- Given a page p , find all wiki-linked pages to p and initialize the PPR vector to these pages
 - Optionally prune (a) pages with spaces in the name and (b) pages account for fewer than $x\%$ of the links
- Run PPR and compare vectors

	MC	WordSim-353
PPR	.60	.45
WikiRelate	.54	.48
ESA	.72	.75

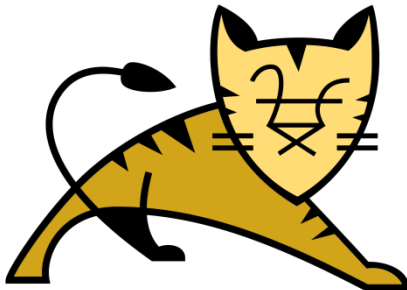
(Still) State of the Art for Wikipedia: Explicit Semantic Analysis

- Consider each Wikipedia article as a concept



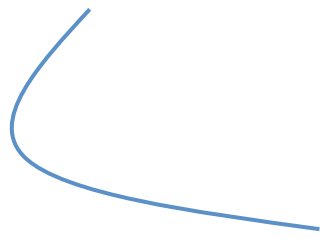
- $\{c_1, c_2, \dots, c_N\}$ where N is the number of articles in Wikipedia

articles for Tomcat



Explicit Semantic Analysis (ESA)

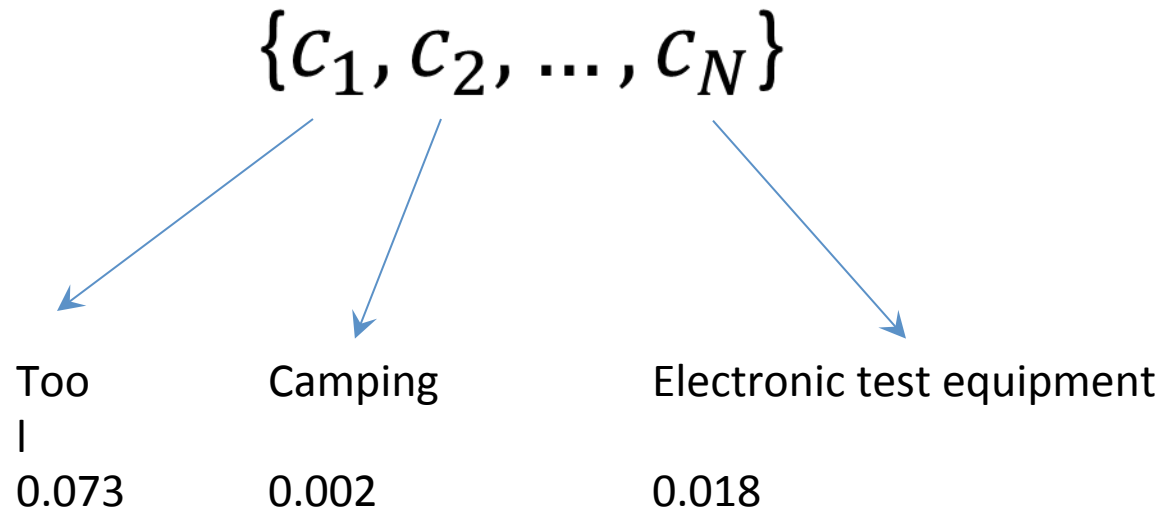
- For a given word (e.g., *equipment*) calculate **inverted index entry** to all the N documents:


$$T[i, j] = tf(t_i, d_j) \cdot \log \frac{n}{df_i},$$

$$tf(t_i, d_j) = \begin{cases} 1 + \log count(t_i, d_j), & \text{if } count(t_i, d_j) > 0 \\ 0, & \text{otherwise} \end{cases}$$

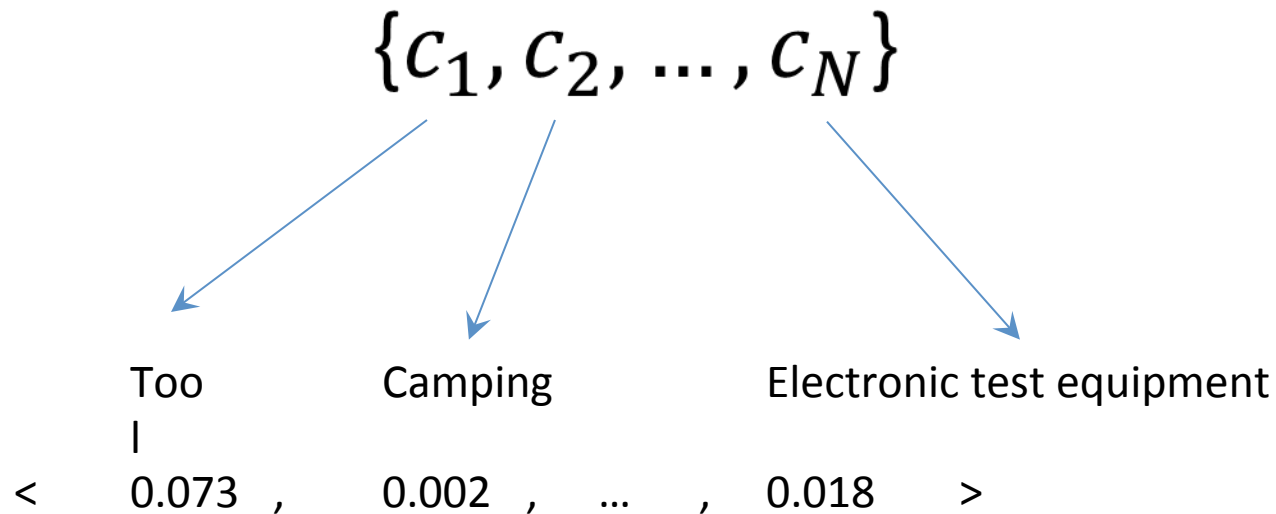
Explicit Semantic Analysis (ESA)

- For a given word (e.g., *equipment*) calculate **inverted index entry** to all the N documents:



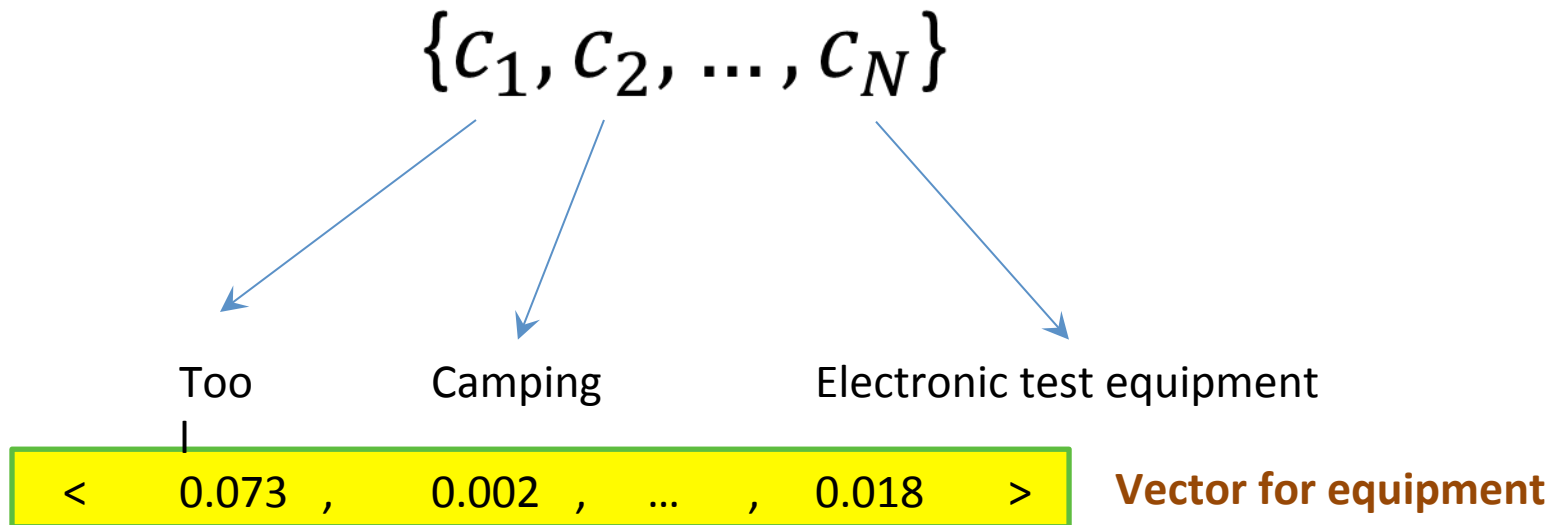
Explicit Semantic Analysis (ESA)

- For a given word (e.g., *equipment*) calculate **inverted index entry** to all the N documents:

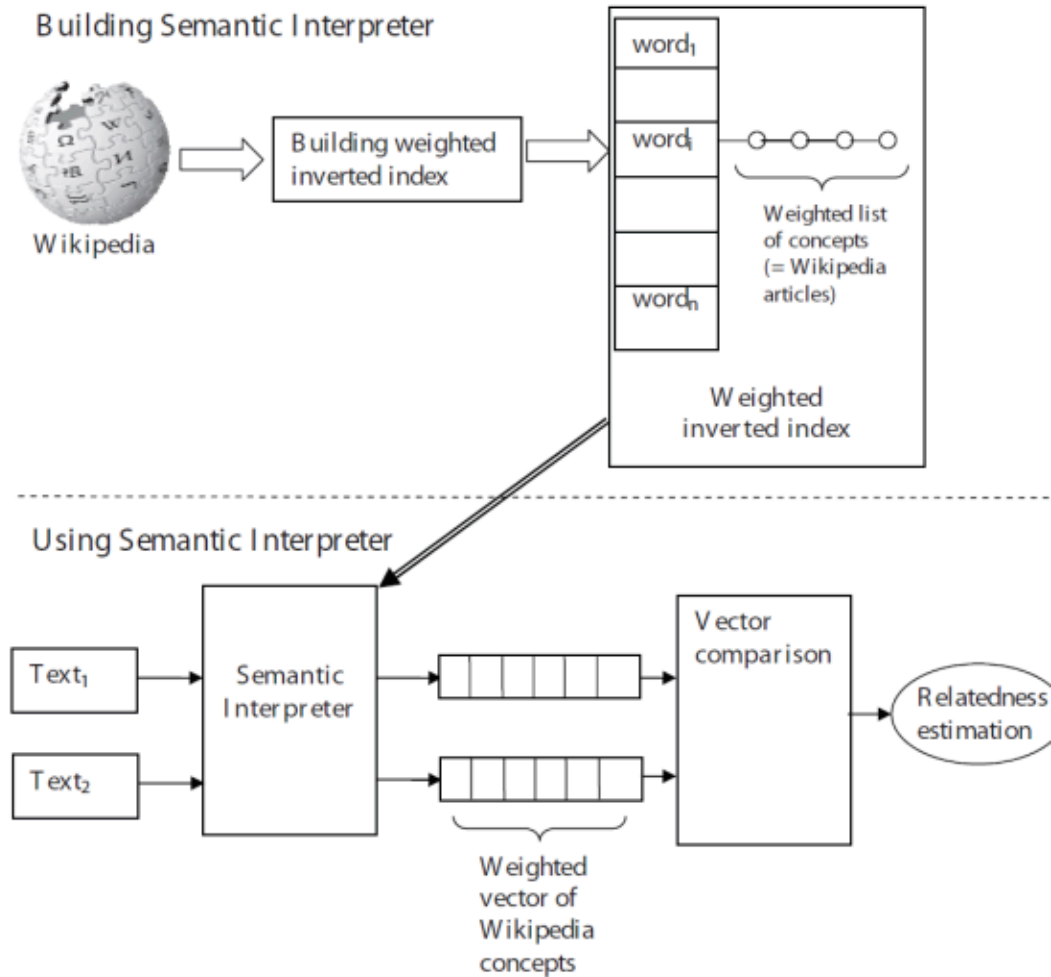


Explicit Semantic Analysis (ESA)

- For a given word (e.g., *equipment*) calculate **inverted index entry** to all the N documents:



ESA pipeline



ESA (example)

#	Input: “ <i>equipment</i> ”	Input: “ <i>investor</i> ”
1	Tool	Investment
2	Digital Equipment Corporation	Angel investor
3	Military technology and equipment	Stock trader
4	Camping	Mutual fund
5	Engineering vehicle	Margin (finance)
6	Weapon	Modern portfolio theory
7	Original equipment manufacturer	Equity investment
8	French Army	Exchange-traded fund
9	Electronic test equipment	Hedge fund
10	Distance Measuring Equipment	Ponzi scheme

Table 1: First ten concepts in sample interpretation vectors.

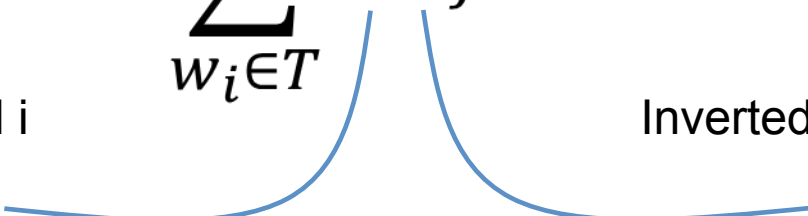
ESA: text modeling

- Centroid of the vectors representing the individual words
- Can be weighted:

$$\sum_{w_i \in T} v_i k_j$$

TFIDF weight of word i
in the text

Inverted index for word i



Wiktionary provides links with more semantic structure

English [\[edit\]](#)

Alternative forms [\[edit\]](#)

- **darg**, **dawg** (*dialectal*); **doggie**, **doggy** (*childish*)

Noun [\[edit\]](#)

dog (*plural* **dogs**)

1. A mammal, *Canis lupus familiaris*, that has been **domesticated** for thousands of years, of highly variable appearance due to human breeding. [\[quotations ▼\]](#)

*The **dog** barked all night long.*

2. A male dog, **wolf** or **fox**, as opposed to a **bitch** (often attributive). [\[quotations ▼\]](#)

3. (*derogatory*) A dull, unattractive **girl** or **woman**.

*She's a real **dog**.*

Synonyms [\[edit\]](#)

- (animal): taxonomic names: *Canis familiaris*, *Canis domesticus*, *Canis familiarus domesticus*, *Canis canis*, *Canis aegyptius*, *Canis familiarus aegyptius*, *Canis melitaeus*, *Canis familiarus melitaeus*, *Canis molossus*, *Canis familiarus molossus*, *Canis saultor*, *Canis familiaris saultor*
- (animal): **domestic dog**, **hound**, **canine**; *see also* Wikisaurus:dog
- (male): **stud**, **sire**
- (man): **bloke** (*British*), **chap** (*British*), **dude**, **fellow**, **guy**, **man**; *see also* Wikisaurus:man
- (morally reprehensible person): **cad**, **bounder**, **blackguard**, **fool**, **hound**, **heel**, **scoundrel**
- (mechanical device): **click**, **detent**, **pawl**
- (metal support for logs): **andiron**, **firedog**, **dogiron**

Coordinate terms [\[edit\]](#)

- (male adult dog): **bitch**, **pup**, **puppy**

Hyponyms [\[edit\]](#)

- (animal): **Afghan hound**, **bloodhound**, **chihuahua**, **coonhound**, **dachshund**, **deerhound**, **foxhound**, **gazehound**, **German shepherd**, **greyhound**, **hound**, **Irish Wolfhound**, **Norwegian Elkhound**, **otterhound**, **pointer**, **poodle**, **retriever**, **Russian Wolfhound**, **scenthound**, **setter**, **sheepdog**, **shepherd**, **sighthound**, **spaniel**, **staghound**, **terrier**, **wolfhound**

Hypernyms [\[edit\]](#)

- (animal): **canid**

**Ideal for path-based
measures of similarity
and for random walks!**

Random Walks are still useful if you use a semantically structured resource

	RG
ADW w/ Wiktionary (Pilehvar and Navigli, 2015)	.920
ADW w/ WordNet (Pilehvar et al. 2013)	.868
PPR w/ WordNet (Hughes and Ramage, 2007)	.838
PPR w/ WordNet (Agirre et al., 2009)	.830
ESA (Gabrilovich and Markovitch, 2007)	.749
WikiRelate (Strube and Ponzetto, 2006)	$r = 0.53$

Word vectors don't need to be learned either!

Idea: create binary vectors of whether a word satisfies a set properties from knowledge bases

- WordNet: is hypernym of x

- FrameNet: evokes frame x

- Sentiment: evokes emotion or sentiment

- ~172K features total

Optionally compress vectors using an SVD

Word vectors don't need to be distributional either!

	RG	SimLex	WordSim-353
word2vec	.728	.436	.656
GloVe	.766	.369	.605
LSA	.770	.496	.673
Ling (full)	.778	.566	.446
Ling (with SVD)	.670	.576	.454

Significant gains in similarity just by encoding knowledge bases in a vector format

Word vectors don't need to be distributional either!

	RG	SimLex	WordSim-353
word2vec	.728	.436	.656
GloVe	.766	.369	.605
LSA	.770	.496	.673
Ling (full)	.778	.566	.446
Ling (with SVD)	.670	.576	.454
ADW	.868		

There may still be better ways to encode knowledge though

(Faruqui and Dyer, 2015)

**What if we knew
something but still
wanted to learn?**

Idea: modify vectors learning (or representations) to match desired properties of knowledge bases

Impose constraints such as

$\text{Sim}(\text{word}, \text{synonym}) > \text{Sim}(\text{word}, \text{antonym})$

Similarity is greater when concepts are more categorically related (e.g., using hypernyms)

Constraints could be added during learning or could be used to retrofit already-learned vectors

Idea: modify vectors learning (or representations) to match desired properties of knowledge bases

	Where is knowledge added?	RG	TOEFL	WordSim-353
word2vec	N/A	.728	83.75	.709
Li et al., (2015)	Learning		87.5	.727
Faruqui et al., (2015)	Representation	.778	100	.700
Iacobacci et al., (2015)	Similarity	.871		.779

Significant opportunities to add knowledge at different stages, with the ability to tune the representation or how it is used for a specific task

(Iacobacci et al., 2015; Liu et al., 2015; Faruqui et al, 2015)

Phrase similarity

Compositionality

Moving from words to phrases, sentences,
and larger pieces of texts

How would we compare...

“the usual morning cup of joe”

“drip coffee with freshly-ground arabica beans”

“must do our utmost”

“must make every effort”

Measuring the similarity of the phrases requires understanding each item as a whole.

We need compositionality!

**Initial idea: compose from
existing word representations**

Compositionality Techniques

Combining individual words' vectors

Mitchell and Lapata (2008)

Simple average: $p_i = u_i + v_i$

Weighted average: $p_i = \alpha u_i + \beta v_i$

including one or more
distributional neighbors: $\mathbf{p} = \mathbf{u} + \mathbf{v} + \sum \mathbf{n}$

Multiplicative: $p_i = u_i \cdot v_i$

Combined multiplication
and addition:

$$p_i = \alpha u_i + \beta v_i + \gamma u_i v_i$$

Better at distinguishing
high and low
semantic similarity

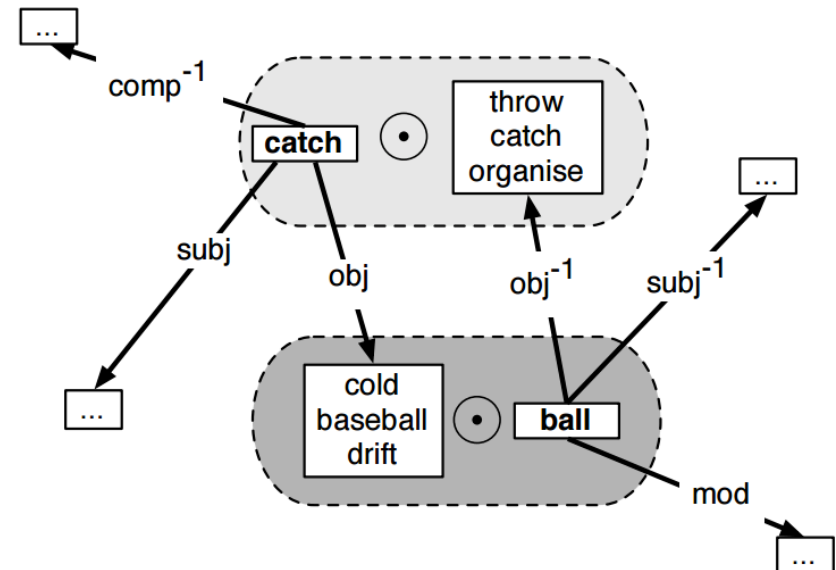
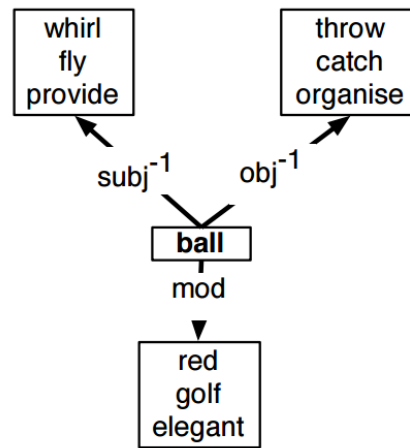
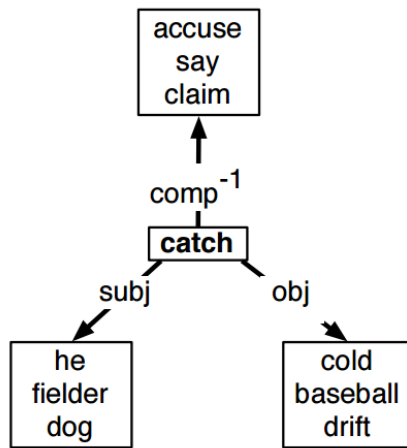
Compositionality Techniques

Erk and Pado (2008)

- Integrates lexical information with selectional preferences
- Computes the meaning of a word a in the context of the word b (disambiguates the meaning of a word in the context of another)
- Takes syntactic structure into account

Compositionality Techniques

Erk and Pado (2008)



Compositionality Techniques

Zanzotto et al (2010)

$$\odot(\mathbf{s}) = \vec{z} = A\vec{x} + B\vec{y}$$

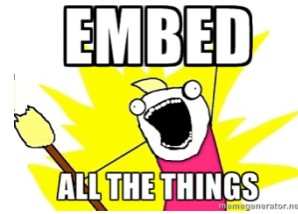
square matrices

- Estimating A and B is a regression problem with multiple dependent variables. Proposes an algorithm for estimating the two matrices.
- Uses a dictionary to build a training data containing positive and negative examples

Training examples, triples: $(\vec{z}, \vec{x}, \vec{y})$

expected vector of the composition of x and y

Learn phrase representations during embedding!



Directly learns word2vec representations for phrases

- First detects phrases in the training corpus by using a simple frequency-based approach
- Treating these phrases as single tokens, obtains phrase-specific representations

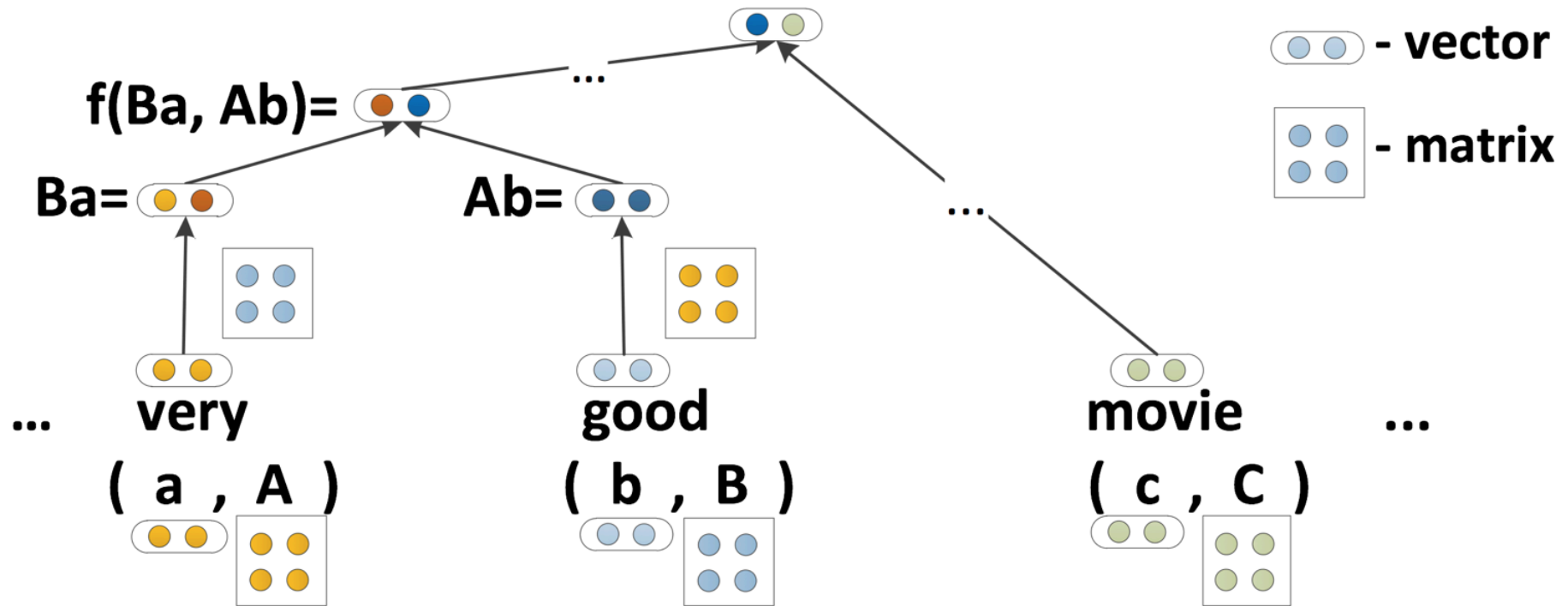
Additive Compositionality

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

**Incapable of handling syntactic dependencies
or related phrasal constructions**

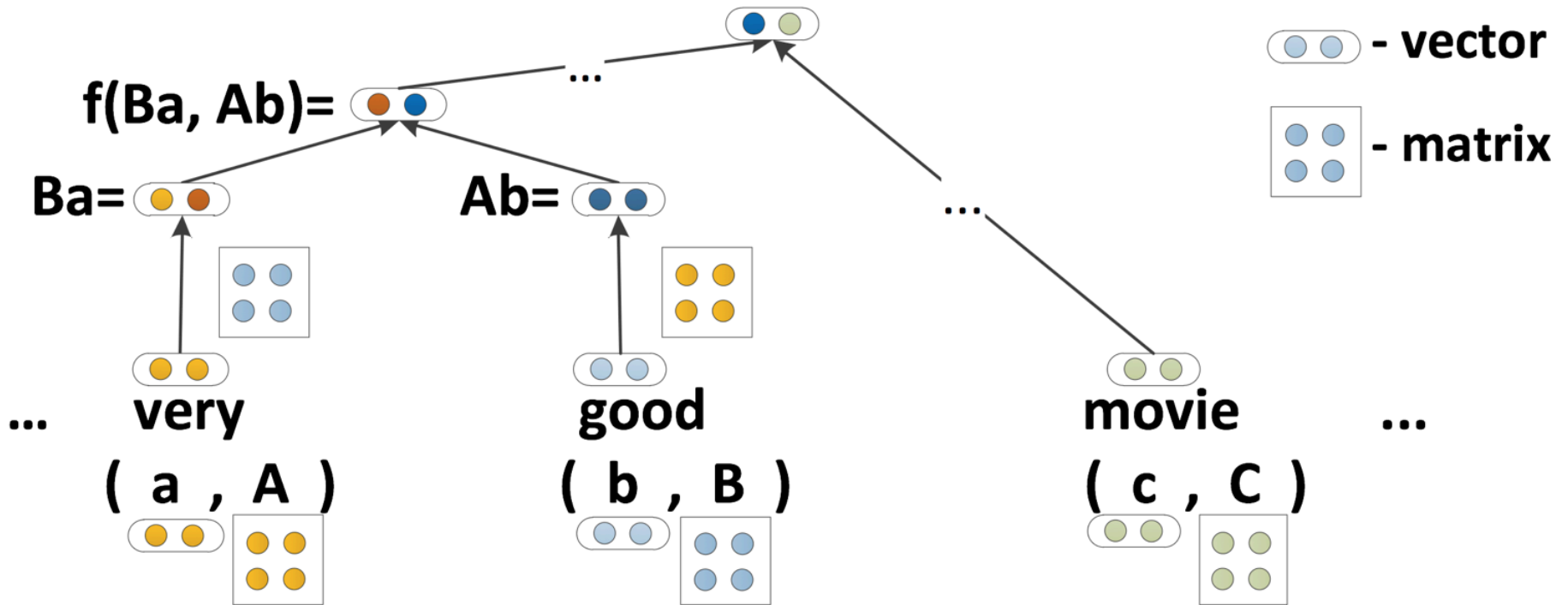
(Mikolov et al., 2013)

Compose with a recursive neural net



Note: Requires data be parsable.

Compose with a recursive neural net



Not ideal performance in compositionality-specific tasks (Blacoe and Lapata, 2012; Hashimoto et al., 2014) -- partly because the model isn't trained for compositionality!

(Socher et al., 2012)

Idea: Design an RNN with a cost function based on good paraphrase

Create a paraphrase ranking corpus from PPDB (Ganitkevitch et al., 2013)

Modify the RNN from Socher et al. (2014) so that the loss function penalizes similar representations of bad paraphrase examples

Initialize with word2vec, but tune the vectors

Idea: Design an RNN with a cost function based on good paraphrase

		M&L Bigrams	M&L Paraphrase	Annotated PPDB
word2vec	additive	.39	.36	.20
paragram	additive	.42	.46	.32
paragram	RNN	.47	.52	.40
Hashimoto et al. (2014)		.47	.41	-
Mitchell and Lapata (2010)		.44	-	-

A supervised RNN provides significant benefits over representing phrases using vector addition.

Sentence Similarity

Sentence similarity is one of the most active areas

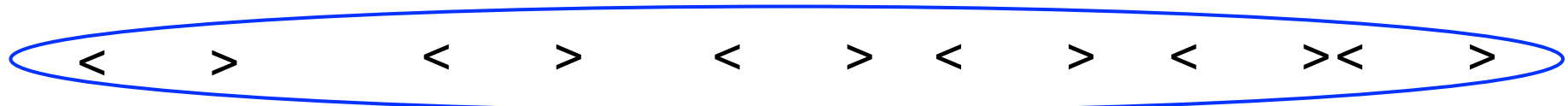
Many applications benefit:

- Paraphrasing
- Textual entailment
- Machine translation
- Question Answering

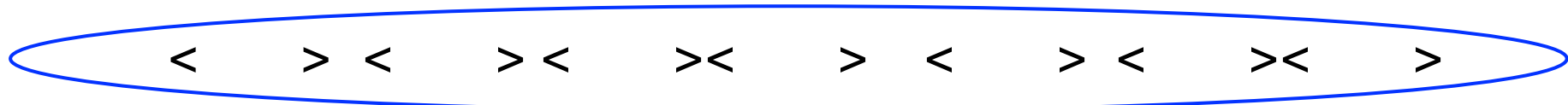
Easy to build models using combinations of string similarity and word-semantics similarity!

Sentence Similarity Techniques

Basic idea: Average vectors of the words in a sentence



Indonesia passenger plane wreckage located in remote Papua



Indonesia Plane Debris Found in Remote Papua Area

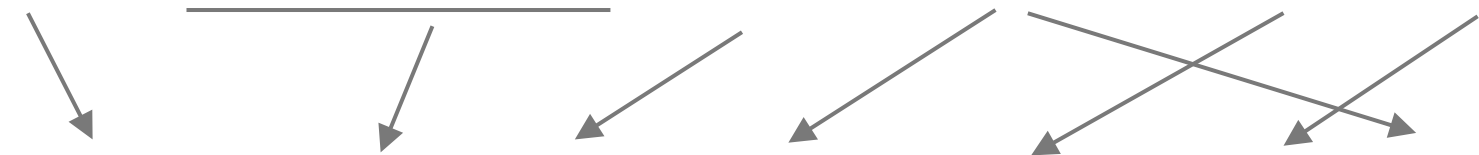
Sentence Similarity Techniques

Alignment

Aggregate the similarities of the closest pairs of words: Corley and Mihalcea (2005)

$$\text{sim}(T_1, T_2) = \frac{\sum_{w \in T_1} \max \text{Sim}(w, T_2) \text{idf}(w)}{\sum_{w \in T_1} \text{idf}(w)}$$

Indonesia passenger plane wreckage located in remote Papua



Indonesia Plane Debris Found in Remote Papua Area

Sentence Similarity Techniques

Usually feature-based regression models

e.g., UKP (best system in STS-12)

String-based similarity: character n-gram,
GST, etc.

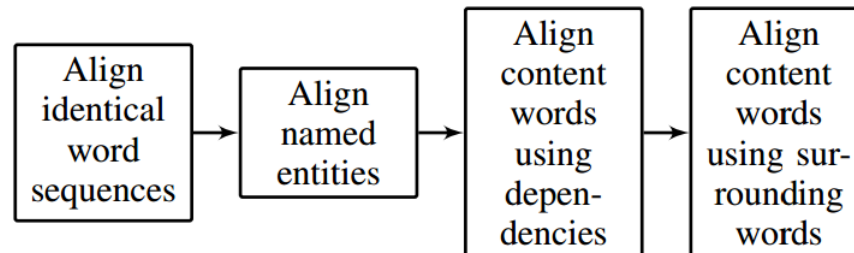
semantic similarity: WordNet-based approaches,
ESA, etc.

Other features: POS n-gram, SMT, etc.

Sentence Similarity Techniques

Monolingual alignment

Sultan et al (2014): best system in STS-14 and -15
DLS@CU



The similarity score is computed as a function of the proportions of aligned content words in the two input sentences.

Sentence Similarity Techniques

Soft cardinality

Jimenez et al (2010)

Uses only surface text information, a stop-word remover, and a stemmer
ranked 3rd in STS-12

$$SIM(A, B) = \frac{|A \cap B| + bias}{\alpha \max(|A|, |B|) + (1 - \alpha) \min(|A|, |B|)}$$

Coffee Break

30 minutes

Paragraph Similarity

Paragraphs represent large thematic, topical units -- more than just a sequence of sentence

The Lisbon region is the wealthiest region in Portugal and it is well above the European Union's GDP per capita average – it produces 45% of the Portuguese GDP. Lisbon's economy is based primarily on the tertiary sector. Most of the headquarters of multinationals operating in Portugal are concentrated in the Grande Lisboa Subregion, specially in the Oeiras municipality. The Lisbon Metropolitan Area is heavily industrialized, especially the south bank of the Tagus river (Rio Tejo).

Little evaluation directly on paragraph similarity

Often used as the unit of text for applications

- Plagiarism detection
- Summarization
- Essay grading
- Scientific abstracts
- Document chunking

Simplest Idea: Model paragraphs as a bag of words (BoW)

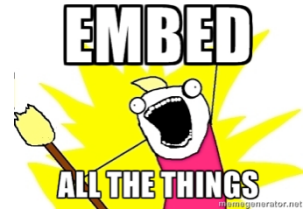
Paragraph BoW representations run into all the same issues as with words

- huge dimensionality makes them cumbersome
- ignores word semantics

Paragraphs also include word ordering and sentence ordering

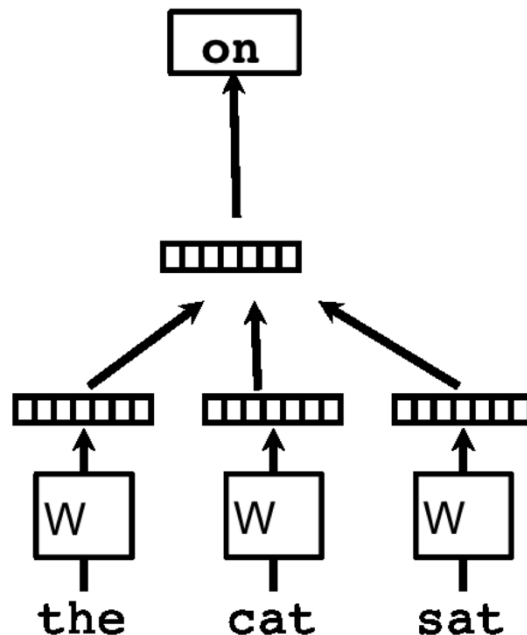
- The topic sentence can matter!

Current state of the art: doc2vec



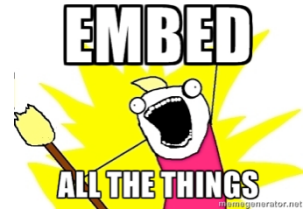
Tackles two problems with bag-of-words and topic modeling approaches:

- They lose the ordering of the words
- They ignore semantics of the words

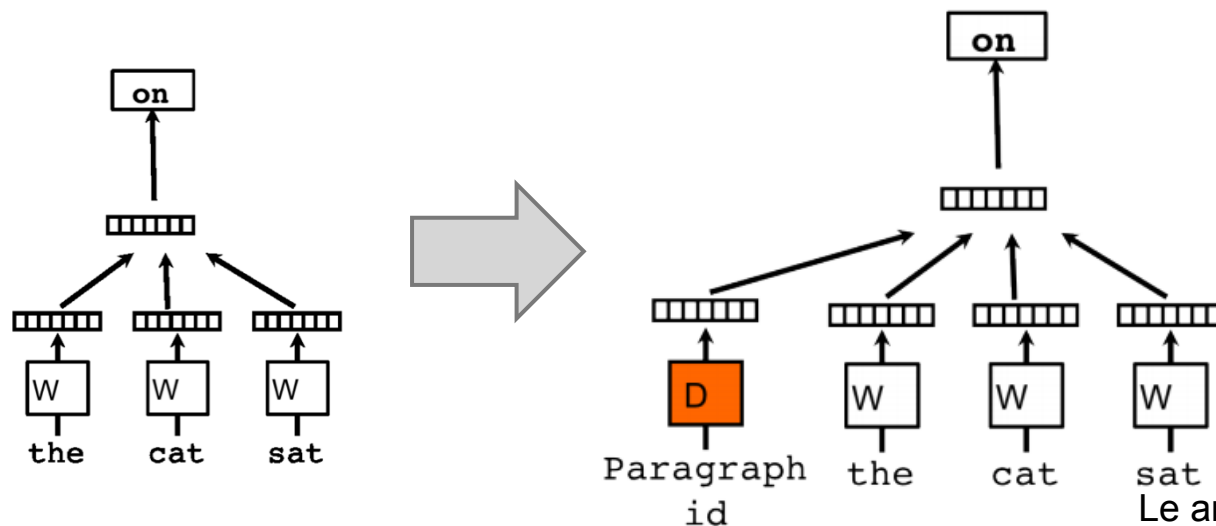


Base model is a prediction task to predict the next word in a sequence

Current state of the art: doc2vec



- Incorporate paragraph structure explicitly by adding a paragraph vector to the predictive model
 - Every paragraph is mapped to a unique vector
 - A paragraph is thought of as another word that remembers what is missing from the current context

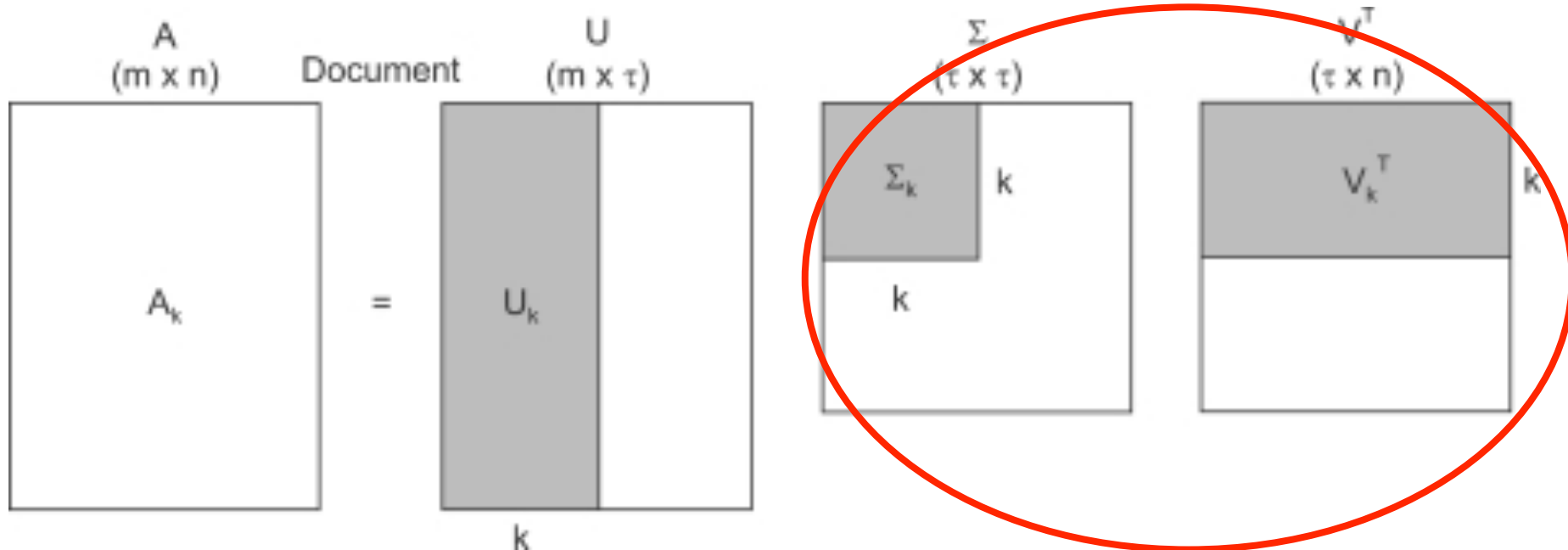


Le and Mikolov (2014)

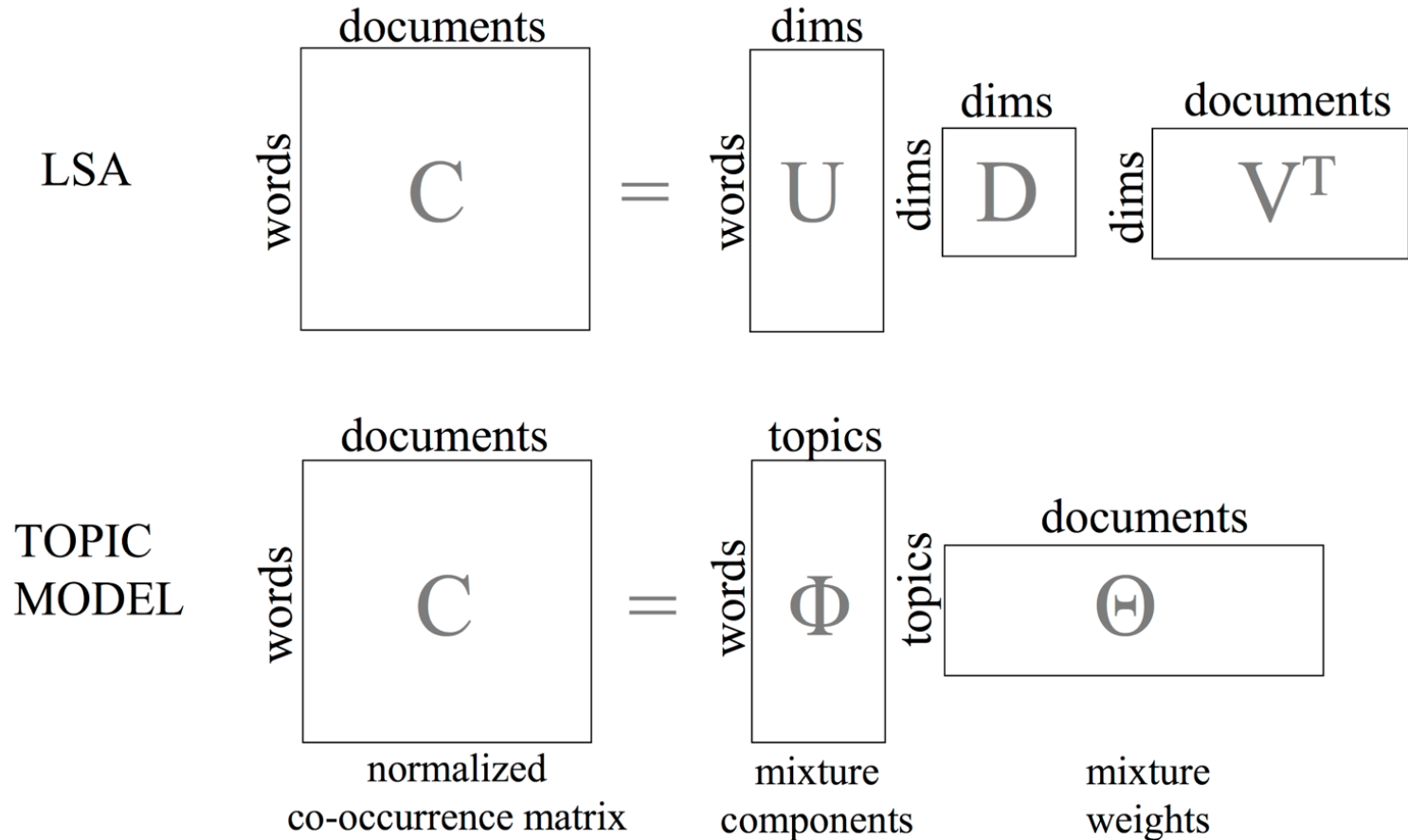
Document Similarity

Early document similarity techniques used vector space models

Latent Semantic Indexing (LSI, aka LSA) developed by Deerwester (1988) to address already-discussed issues with VSMs.



Topic Modeling: Viewing document contents as a mixture of topics



Topic Modeling: Viewing document contents as a mixture of topics

Topics

gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	

Documents

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson at Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

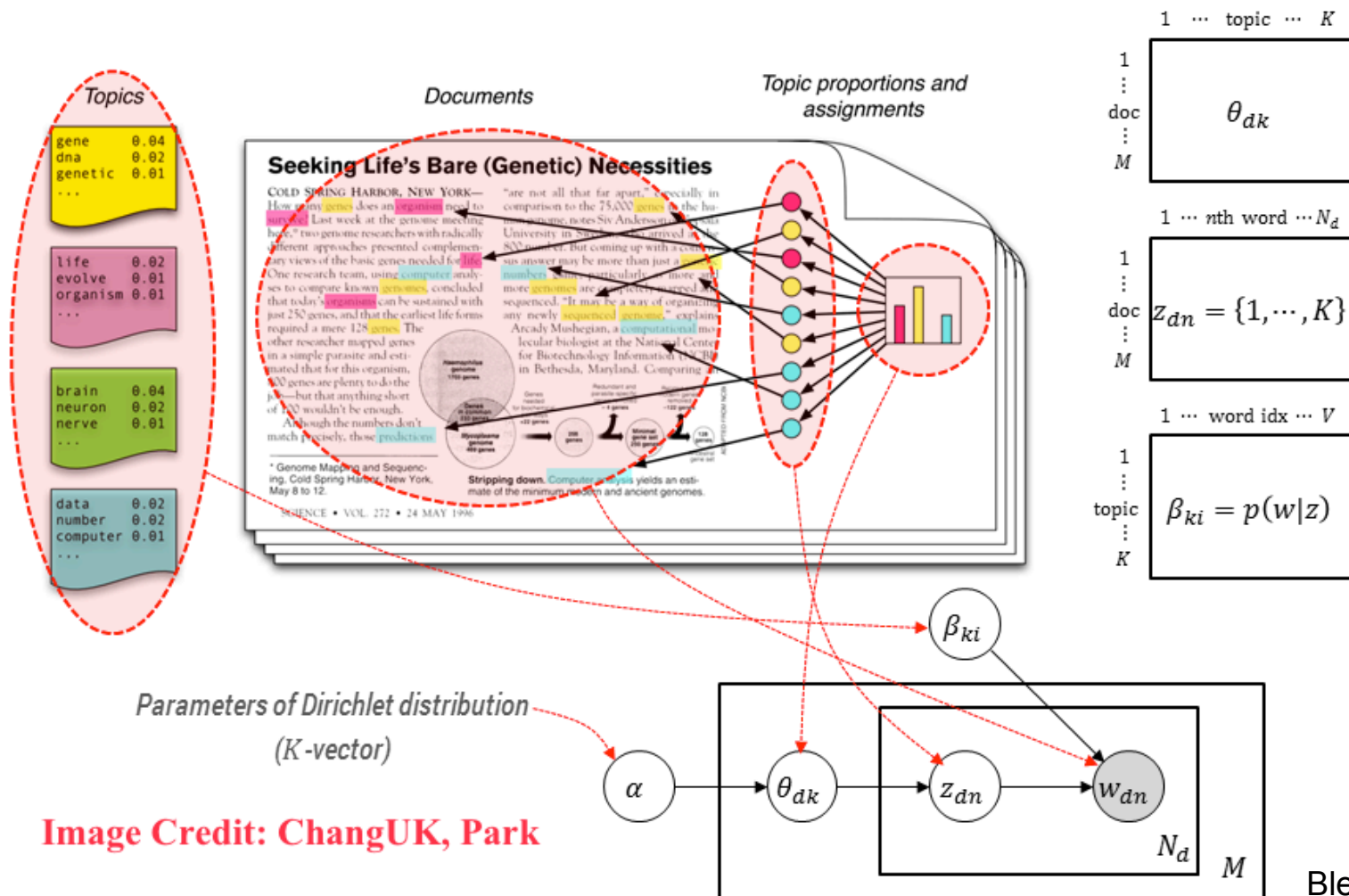
SCIENCE • VOL. 272 • 24 MAY 1996

Topic proportions and assignments



Document Similarity Techniques

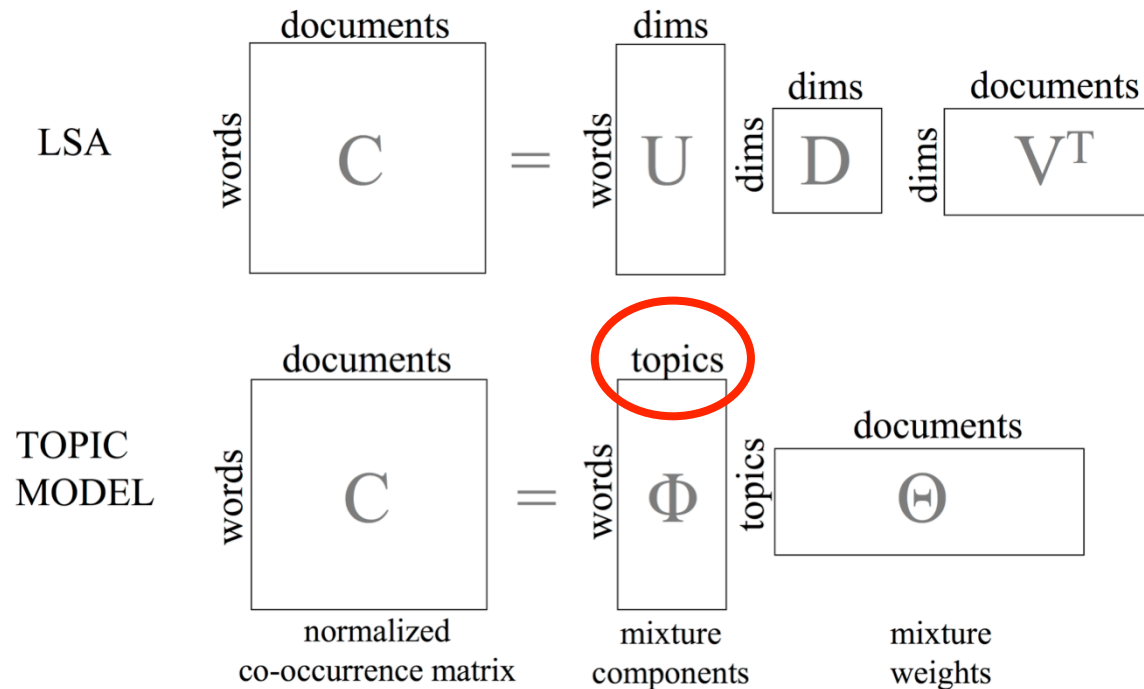
Latent Dirichlet Allocation



Key points for using topic distributions as document representations

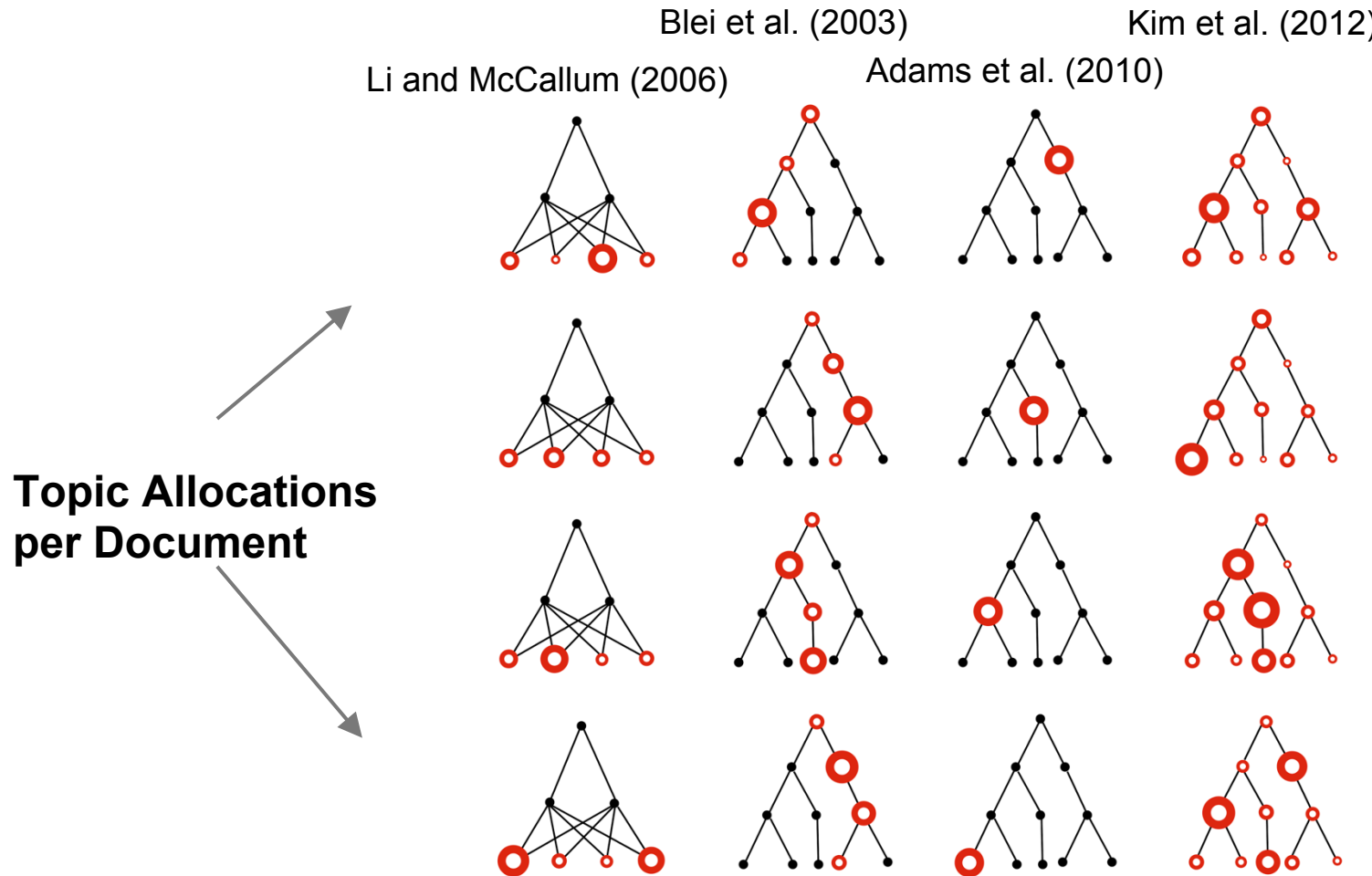
- Selecting the number of topics
- Identify relationships between topics
- Moving beyond token-topic assignments

How many topics should you use?



Let a Hierarchical Dirichlet Process (HDP) model decide for you.

Introducing structure into the topics



Hierarchical topic organizations can potentially yield more informative document representations

Incorporating Multi-Word Expressions into topics

Pre-process the corpus to glob MWEs together, e.g., “white house” -> white_house
Not feasible for domain-specific MWEs

Incorporating Multi-Word Expressions into topics

Pre-process the corpus to glob MWEs together, e.g., “white house” -> white_house
Not feasible for domain-specific MWEs

Learn the MWEs on the fly by looking at topic-assignment sequences

TurboTopics (Blei and Lafferty, 2009)

TurboTopics example phrases

Huffington Post				Physics arXiv				n-gram topics
movie	barack obama	marriage	hillary clinton	mass	model	lattice qcd	phase transitions	
the film	obamas	state	campaign	star formation	point	mass	model	
hollywood	campaign	in california	bill clinton	stars	monte carlo simulations	dirac operator	symmetry	
director	sen barack obama	gay	shes	masses	fixed point	chiral perturbation theory	point	
first	democratic	decision	the clinton	black hole	results	operators	quantum	
character	the illinois senator	court	hillarys	stellar	lattice	quarks	systems	
documentary	michelle	law	president	star	scaling	limit	phase transition	
theater	recent	supreme court	sen clinton	black holes	numerical	theta	phase diagram	
best	speech	couples	mark penn	massive	ising model	quark	system	
sex and the city	choice	ruling	politics	msun	two	mev	order	
hbo	sen clinton	rights	sexism	solar masses	we study the	simulations	field	
scene	david axelrod	equality	the first	stellar mass	models	lattice spacing	order parameter	
to make	president	legal	her campaign	black hole mass	quantum monte carlo	chiral symmetry breaking	critical	
release	camp	to marry	supporters	the stellar	interactions	results	two	
screen	the huffington post	married	made	young	numerical simulations	effects	transitions in	
actor	endorsed	samesex couples	fight	the mass	simulation	small	models	
made	seen	states	called	times	dimensions	baryon	different	
stars	attacks	gay marriage	mrs clinton	myr	analytical	in the continuum limit	symmetry breaking	
indiana jones	political	sexual orientation	political	imf	phase	physical	first order	
seen	gave	the california supreme court	hillary rodham clinton	supermassive black holes	spin glass	quenched	phenomena	

film	obama	california	clinton	mass	carlo	lattice	phase	unigram topics
movie	barack	marriage	hillary	black	monte	qcd	transitions	
films	obamas	gay	clintons	star	simulations	chiral	phases	
movies	sen	court	campaign	stellar	point	theory	transition	
hollywood	campaign	state	bill	stars	model	mass	quantum	
documentary	senator	couples	shes	masses	results	quark	critical	
director	democratic	supreme	president	hole	fixed	finite	symmetry	
jones	illinois	decision	hillarys	massive	critical	quenched	field	
screen	president	married	supporters	formation	study	perturbation	point	
character	presidential	samesex	penn	holes	two	limit	model	
cannes	recent	rights	politics	msun	lattice	quarks	order	
festival	political	marry	sexism	function	dimensions	results	diagram	
city	speech	law	political	young	scaling	potential	systems	
theater	huffington	ruling	rodham	supermassive	numerical	staggered	two	
star	politics	states	democratic	accretion	simulation	chemical	theory	
hbo	michelle	equality	first	rate	transition	masses	system	
scene	voters	legal	say	solar	ising	simulations	study	
actor	supporters	lesbian	sen	initial	phase	theta	breaking	
played	candidacy	equal	mrs	galactic	twodimensional	continuum	spin	
indiana	choice	appeals	presidency	central	temperature	volume	first	

Incorporating Multi-Word Expressions into topics

Pre-process the corpus to glob MWEs together, e.g., “white house” -> white_house
Not feasible for domain-specific MWEs

Learn the MWEs on the fly by looking at topic-assignment sequences
TurboTopics (Blei and Lafferty, 2009)

Learn the MWEs *during* topic modeling
Most scalable approach is Top-Min (El-Kishky et al., 2014)

TopMine example phrases

	<i>Topic 1</i>	<i>Topic 2</i>	<i>Topic 3</i>	<i>Topic 4</i>
1-grams	plant nuclear environmental energy year waste department power state chemical	church catholic religious bishop pope roman jewish rev john christian	palestinian israeli israel arab plo army reported west bank state	bush house senate year bill president congress tax budget committee
n-grams	energy department environmental protection agency nuclear weapons acid rain nuclear power plant hazardous waste savannah river rocky flats nuclear power natural gas	roman catholic pope john paul john paul catholic church anti semitism baptist church united states lutheran church episcopal church church members	gaza strip west bank palestine liberation organization united states arab reports prime minister yitzhak shamir israel radio occupied territories occupied west bank	president bush white house bush administration house and senate members of congress defense secretary capital gains tax pay raise house members committee chairman

Documents can contain much more than just text

The Application of the PSO Based Community Discovery Algorithm in Scientific Paper Management SNS Platform*

Ruixin Ma¹, Guishi Deng², Xiao Wang², and Ailinna²

¹ School of Software, Dalian University of Technology
teacher_mrx@126.com

² School of Management, Dalian University of Technology
denggs@dlut.edu.cn

Abstract. The development of SNS provides a new platform and application prospect for the realization of Personalized Recommendation System. It is becoming a fire new research hot spot in social science and e-commerce about how to apply community discovery algorithm (CDA) to find community structures in large network and to effectively conduct personalized recommendation. In terms of our recent research, we applied PSO based CDA as principle to divide social communities and based on this to conduct personalized recommendation in a scientific paper management system. The system's running results proved that by applying PSO based CDA, the accuracy of PR and the popularity of the platform had been both improved greatly.

Keywords: SNS, community structure, Personalized Recommendation, PSO based CDA.

1 Introduction

Social network lays emphasis on erpersonal interaction and reflects user's essential need for internet service. In recent years, SNS has become an increasing concern because it allows users to create and to manage things independently by themselves. This concern not only comes from the need to analyze network users' behavior model but also is dominated by the request to maintain the network cohesion. As a result of sparse communication and identity conceals, the effects of PR are not very efficient in normal network.

However, the emergence of SNS brings network users with a safer and more reliable platform[1] which enables users to interact with others more convenient and relieved. This paper's results sufficiently proved that SNS optimized the consequent of social communication and constructed the foundation to realize PRS.

The rest of this paper is organized as follows. In section 2, we introduce the current development of SNS. In section 3, common community discovery algorithms are presented. Section 4 introduces the application of PSO based CDA in scientific paper management system. Subsequently, running comparison is presented and finally, section 6 gives conclusion.

* Supported by "the Fundamental Research Funds for the Central Universitie.

Chapter Two

First Steps in My New Country

My father gave me 50 piasters – that was the name of the sub-division of the pound. My mother made me take my awful, rough Bulgarian coat, and wearing my freshly washed but creased white dress, I made my way towards the main gate.

The policeman, in shirt sleeves, was sitting on a chair at the entrance to his little booth, reading a newspaper. He looked up as I came nearer and smiled:

'Bulgaria?'

I nodded. He spoke to me for a while, and although I didn't understand a word of what he said, I realised that he was talking about me. His winks and side-smiles, made that quite clear, but I didn't much like to think what the subject matter of his discourse was. He pointed to himself eventually, and said:

'Polonia.'

I understood that he was telling me that he was Polish.

'Yes, Warsaw.' I said.

'No, no Krakow!'

It was all Poland to me, but obviously, to him, it made a



Lots of work on structured document similarity

Cross-Level Semantic Similarity

Semantic Similarity

Mostly focused on similar types of lexical items

Sentence Level

Semantic Textual Similarity (STS)

- SemEval 2012
- *SEM 2013
- SemEval 2014

Word Level

Rubenstein and Goodenough (1965)

Miller and Charles (1993)

Landauer and Dumais (1997)

Turney et al. (2001)

Sense Level

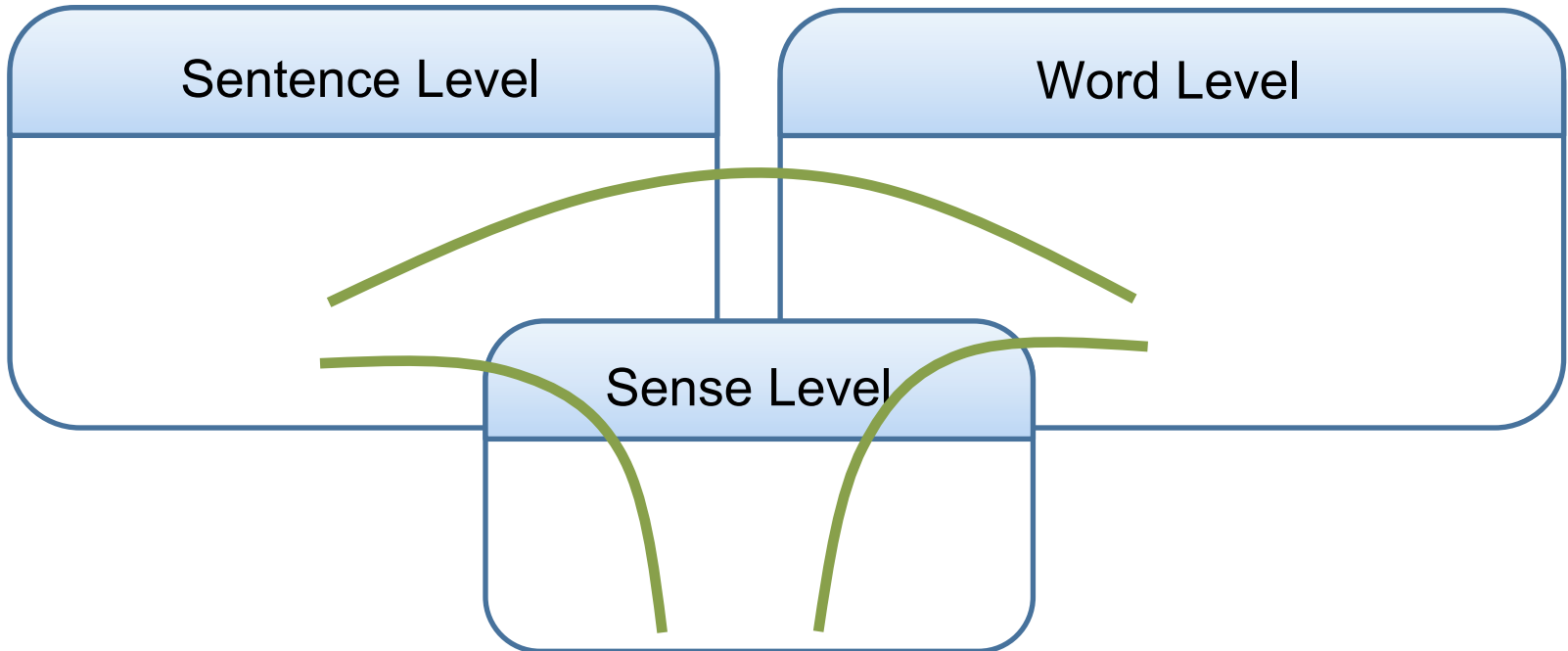
Kilgarrriff (2001)

Navigli (2006)

Semantic Similarity

What if we have different types of inputs?

a large, expensive house
mansion



CLSS: Cross-Level Semantic Similarity

A new type of similarity task

Paragraph

Sentence

Phrase

Word

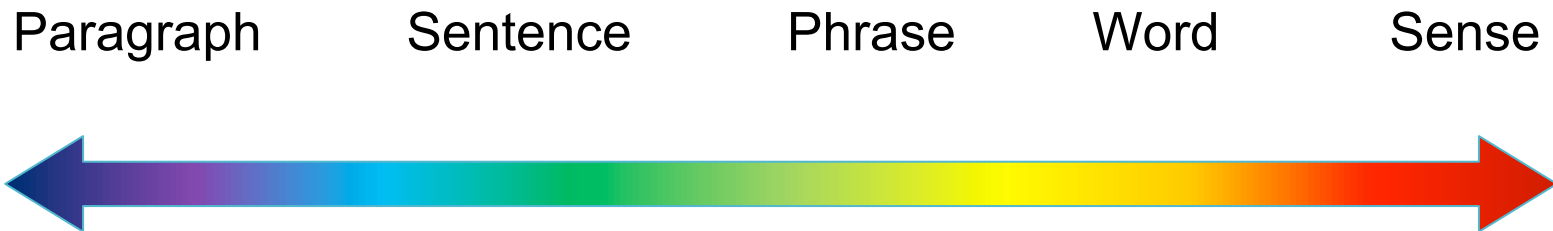
Sense



**Different types of inputs,
from different lexical
levels**

CLSS: Cross-Level Semantic Similarity

A new type of similarity task



- Multiple **types** of comparison
- Incorporate **multiple genres** of text
- Push towards computing the similarity of **anything**

CLASS: Comparison Types

Paragraph to Sentence



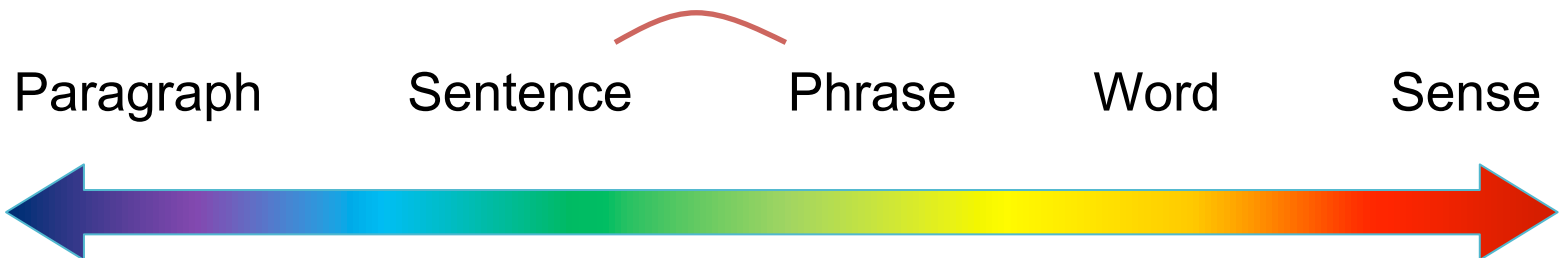
CLSS: Comparison Types

Paragraph to Sentence

Sentence to Phrase

The 30-year-old woman has had no contact with the outside world.

30-year-old female recluse



CLSS: Comparison Types

Paragraph to Sentence

Sentence to Phrase

Phrase to Word

a large, expensive house

mansion

Paragraph

Sentence

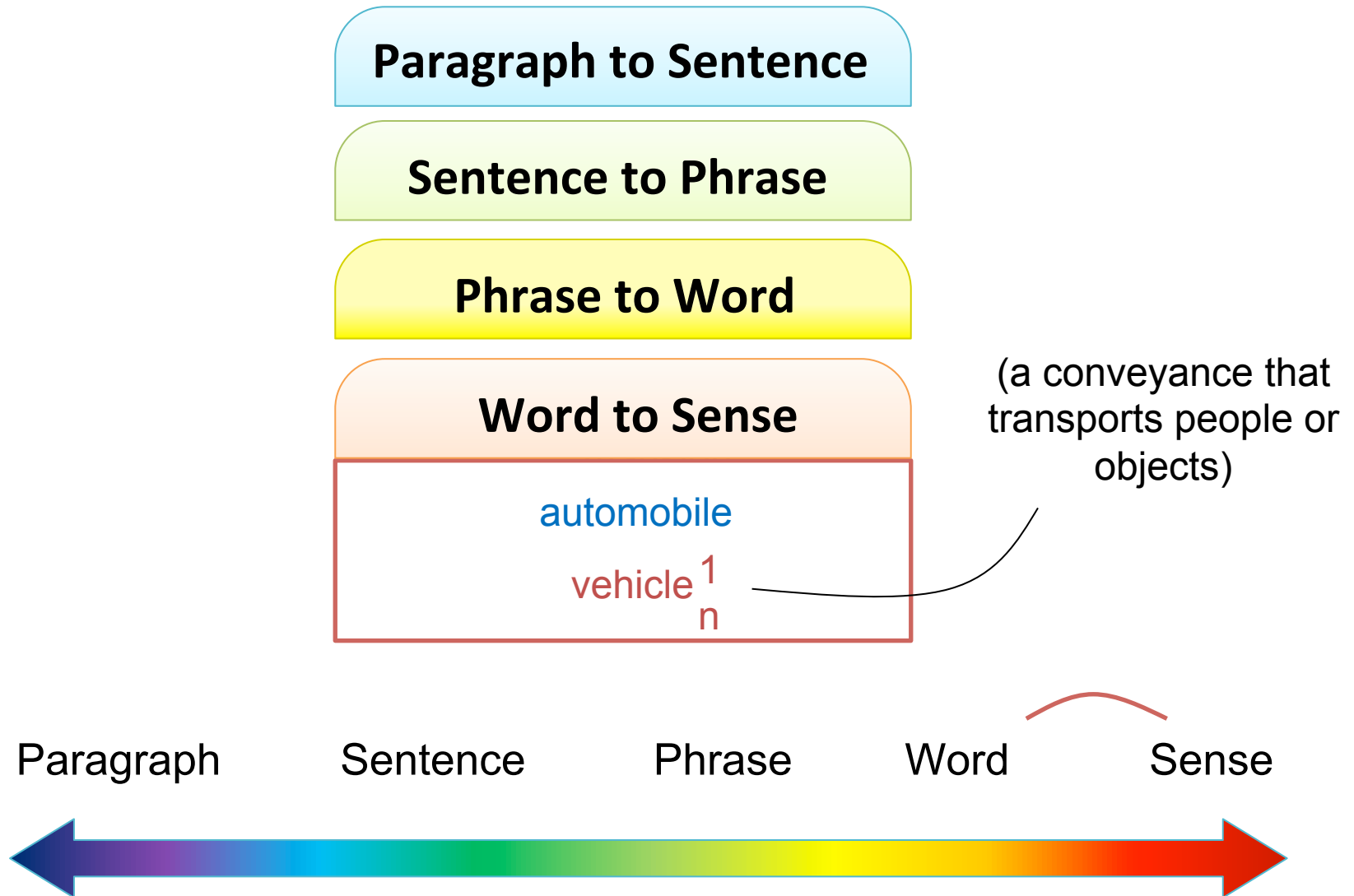
Phrase

Word

Sense



CLSS: Comparison Types



Task Data

Picking Corpora

News

MSR Paraphrase
WikiNews
Reuters News
SMT Europarl
PPDB

Domain-specific

Travel
Guides
Amazon Food Reviews
Wikipedia Science Articles

Yahoo! Answers
Wikipedia Image Captions
Web Search Queries
Wikipedia glosses

Fables
Idioms
Slang

Challenge

Varied writing styles

Picking word-to-sense pairs

“Regular”

“central” vs. essential#a#1

“car” vs. automobile#n#1

Word not in WordNet

“zombify” vs. resurrect#v#3

“drank” vs. opiate#n#1

Sense not in WordNet

“red” vs. communist#a#1

“shiraz” vs. grape#n#1

Picking word-to-sense pairs

“Regular”

“car” vs. automobile#n#1

Challenge

“dog” vs. doggy#n#1

“bite” vs. slang#n#1

Word not in WordNet

“zombify” vs. resurrect#v#3

“drank” vs. opiate#n#1

Sense not in WordNet

“red” vs. communist#a#1

“shiraz” vs. grape#n#1

Rating Scale



4 -- *Nearly* identical

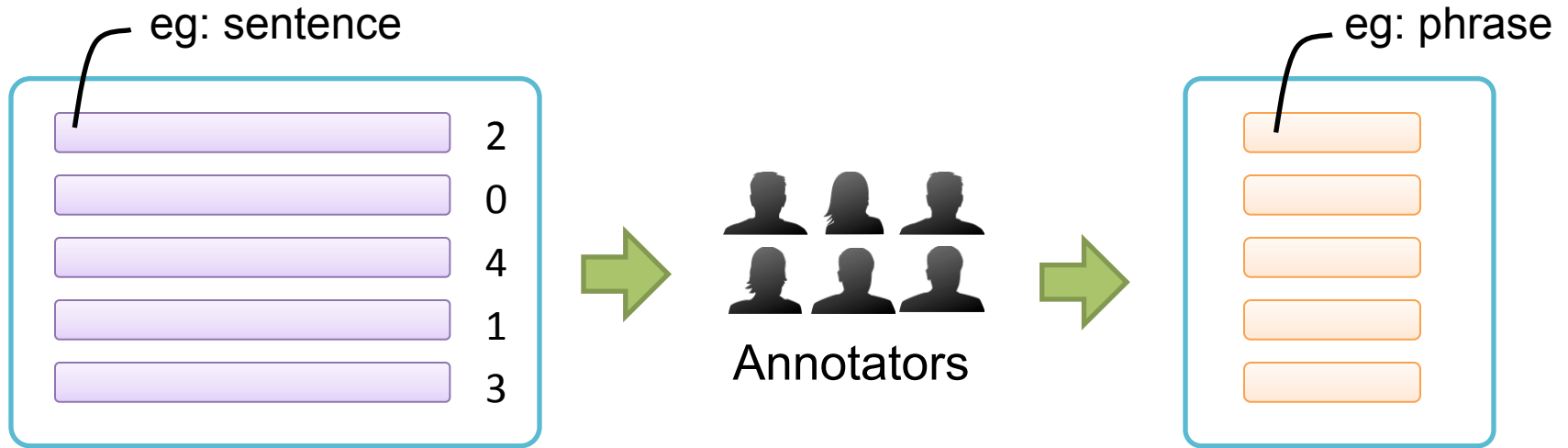
3 -- Similar, but not identical

2 -- Related but not similar

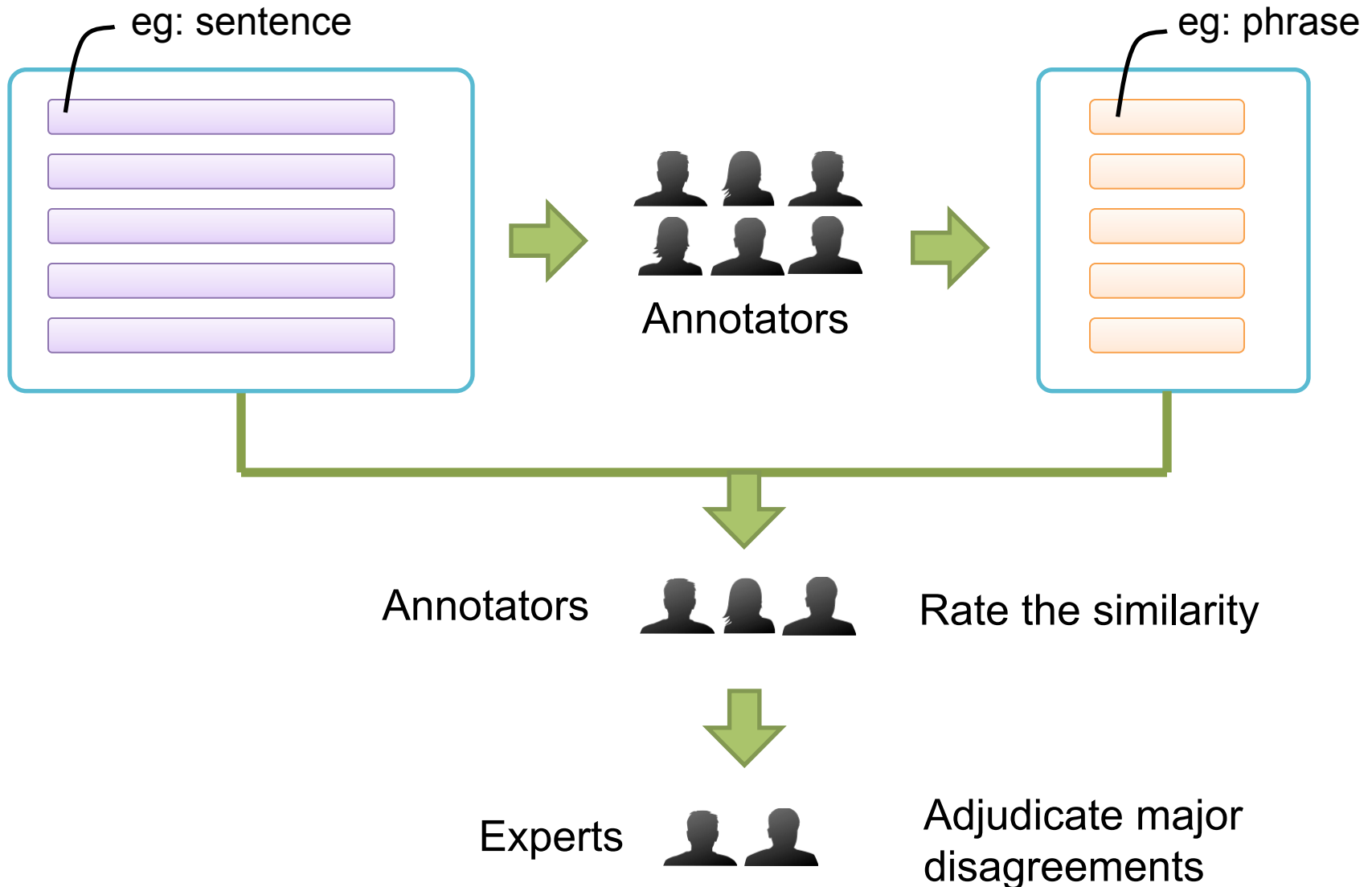
1 -- On the same topic, but not closely related

0 -- Completely unrelated

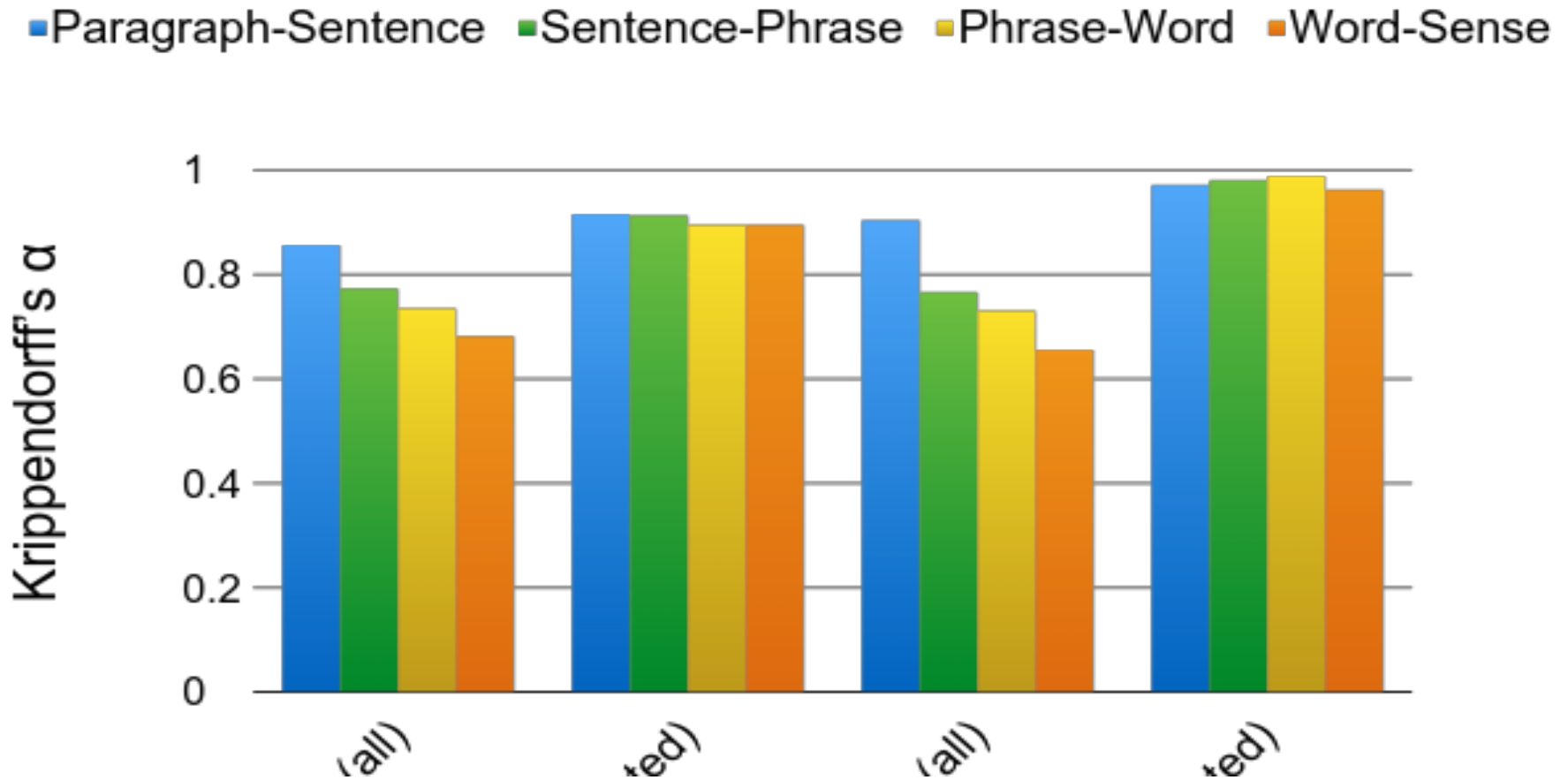
Crafting an idealized similarity distribution



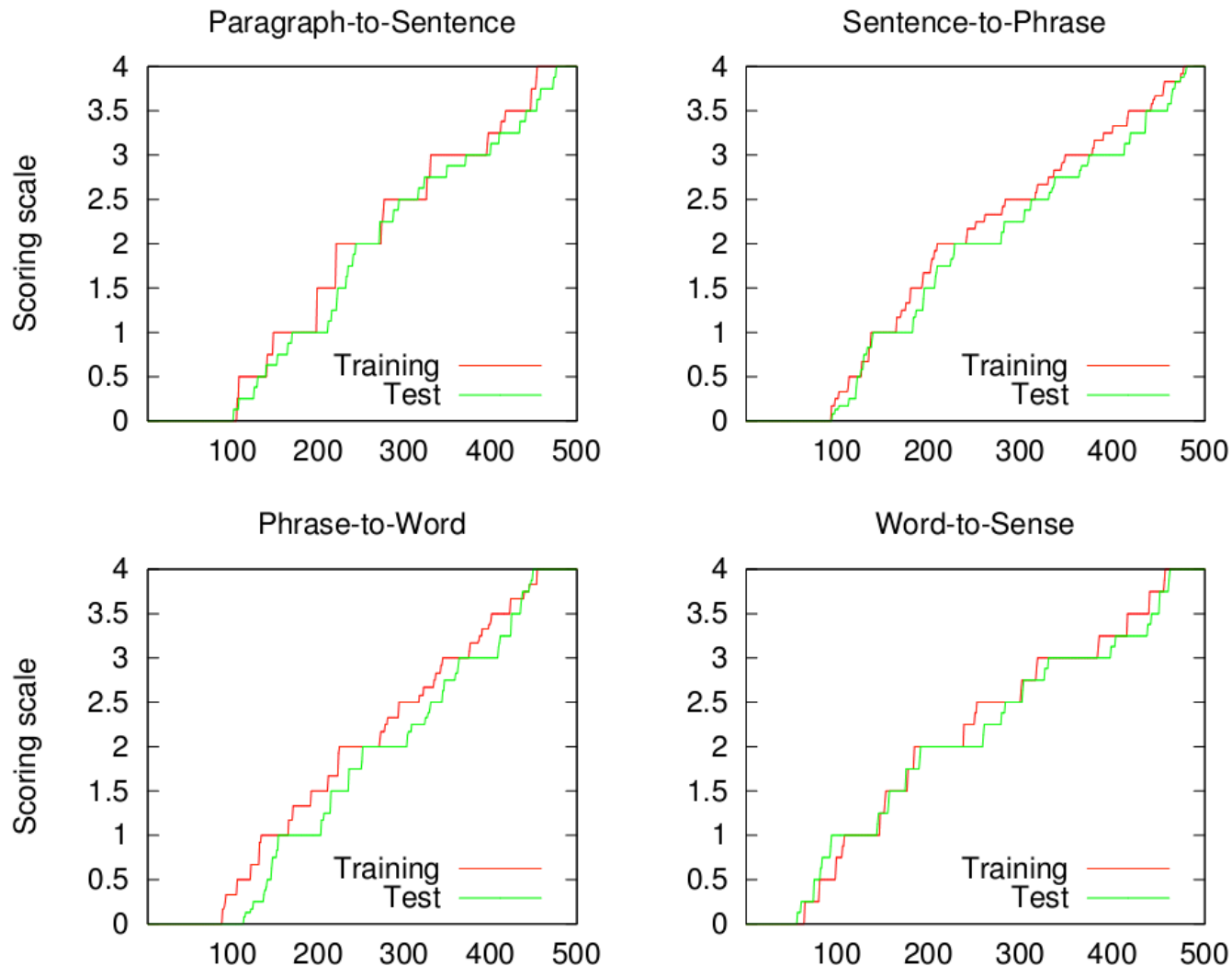
Crafting an idealized similarity distribution



Test and Training data IAA



The annotation procedure produces a balanced rating distribution



Comparison Baselines

- Longest Common Substring (LCS)

The quick **brown fox**

The **brown fox** was quick

- Greedy String Tiling (GST)

The quick brown fox

The brown foxes was **quick**

Evaluation Setup

500 pairs per type

Paragraph to Sentence

Sentence to Phrase

Phrase to Word

Word to Sense

Training set

500 pairs per type

Paragraph to Sentence

Sentence to Phrase

Phrase to Word

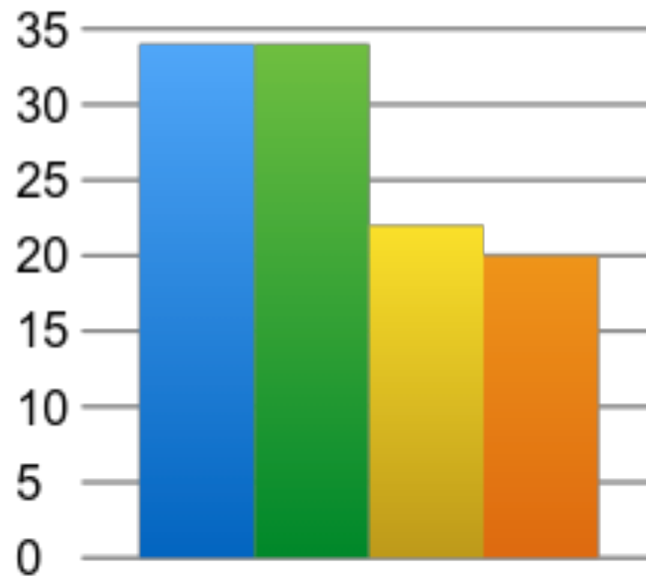
Word to Sense

Test set

Evaluation measure: Pearson correlation

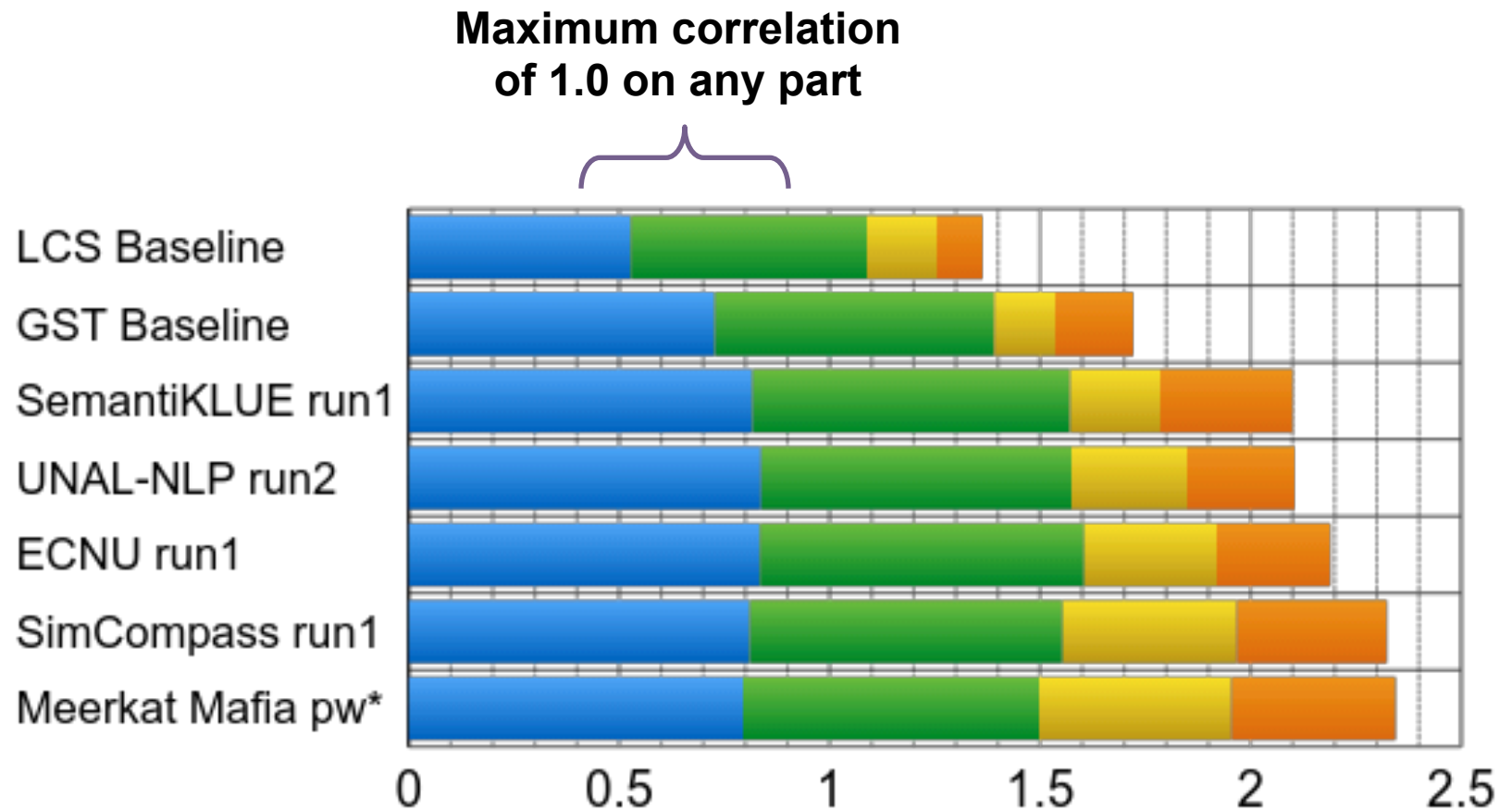
Number of participants

- Paragraph-Sentence
- Sentence-Phrase
- Phrase-Word
- Word-Sense



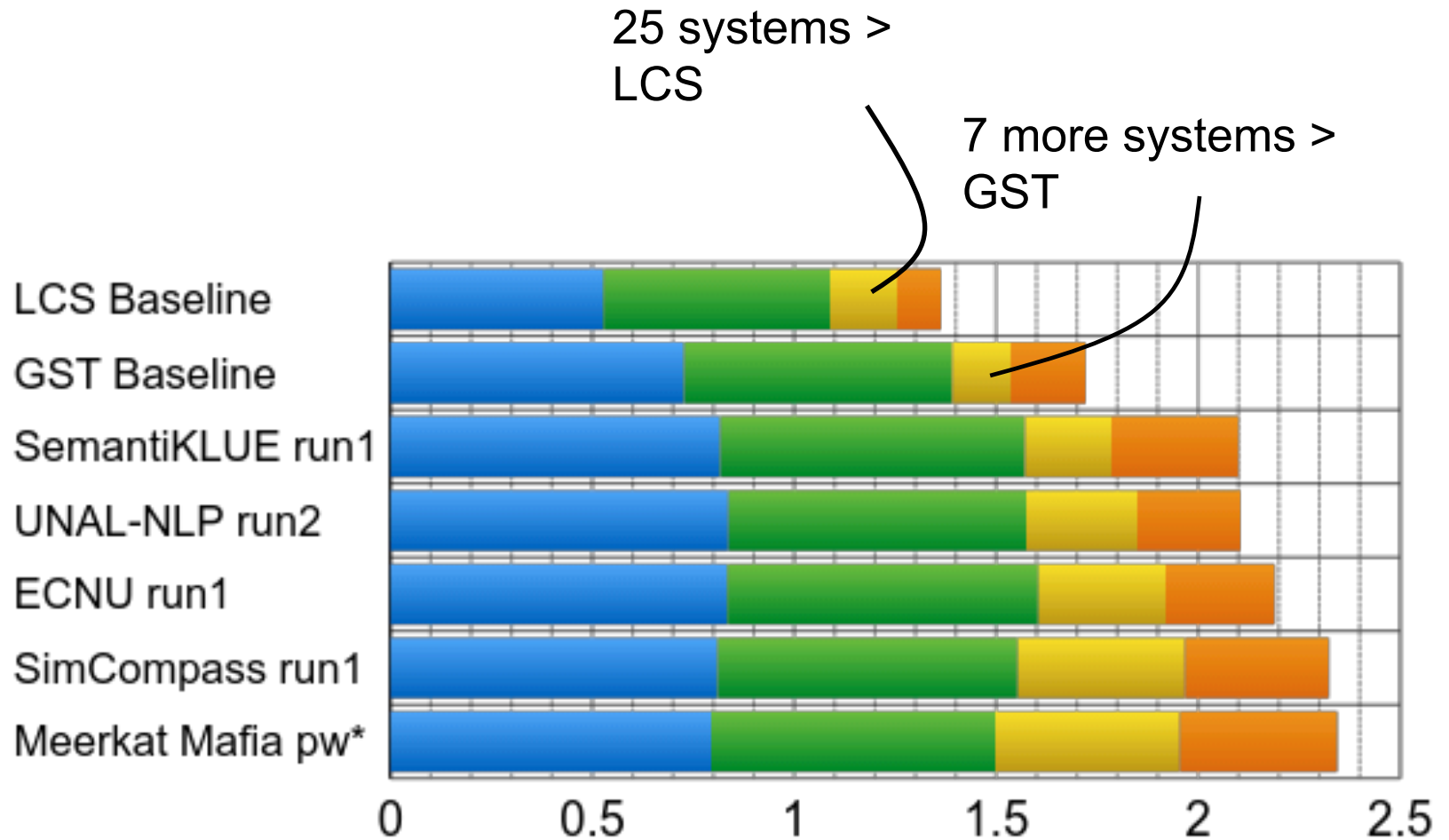
38 Systems total from 19 teams

Top 5 Systems and Baselines



■ paragraph-sentence

Where do baselines stand?



■ paragraph-sentence

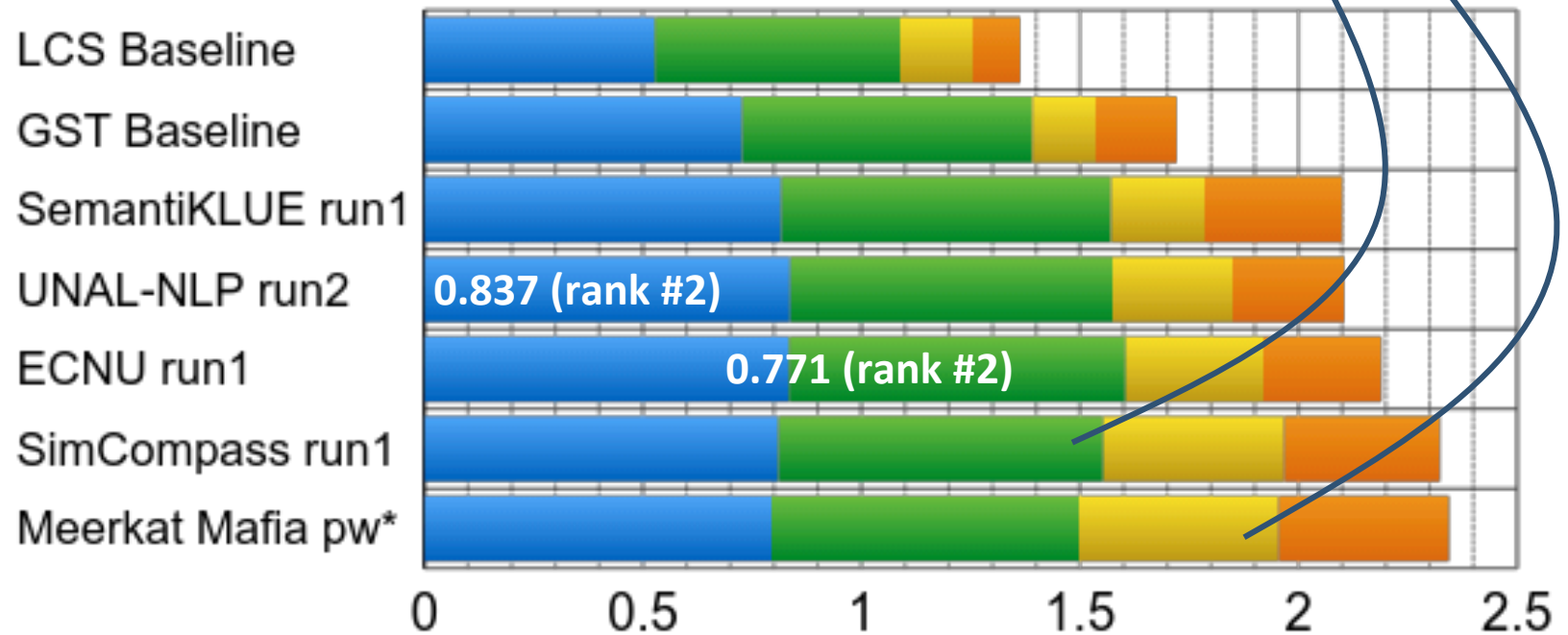
Best Results per Comparison Type

Best word-to-sense: **0.389**

Best phrase-to-word: **0.458**

Best sentence-to-phrase: **0.777 (not shown)**

Best paragraph-to-sentence: **0.845 (not shown)**



■ paragraph-sentence

SimCompass - Banea et al (2014)

Highest overall performance among all competing systems.

Multi-feature regression model:

- Knowledge-based
 - Different WordNet-based measures
- Corpus-based
 - Deep Learning Word Embeddings, Skip-gram (Mikolov et al, 2013)

Other novel features:

- Transform texts to a sets of topic centroids; then check for closest topics

ENCU - Zhu and Lan (2014)

Among the top three systems

Multi-feature regression model:

- String-based
- Knowledge-based
 - Different WordNet-based measures
- Corpus-based
 - LSA
- Syntactic-based

Other novel features:

Using metrics for Machine Translation evaluation for semantic similarity, e.g., TER, METEOR, BLEU, etc.

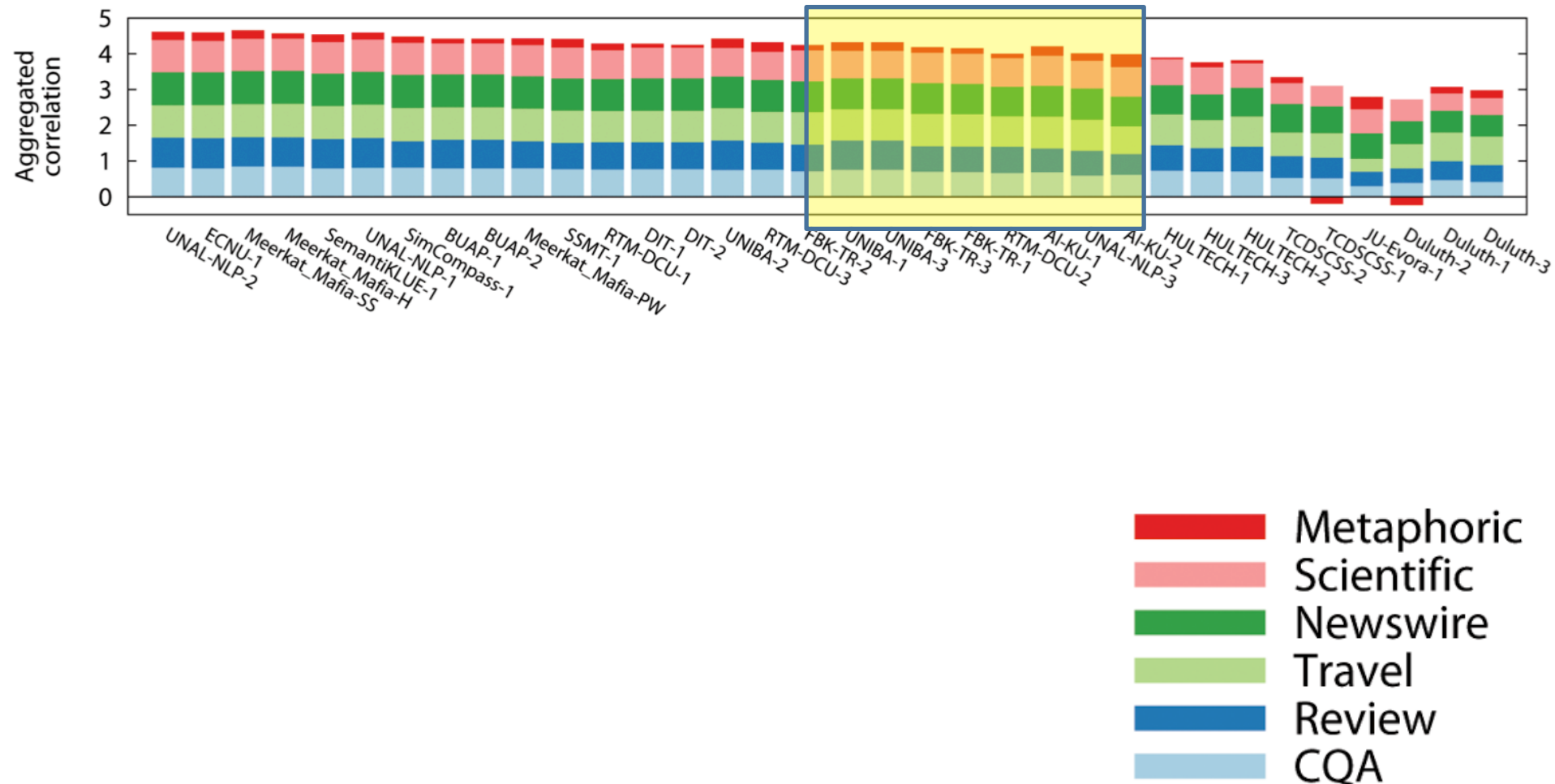
UNAL-NLP - Jimenez et al (2014)

Third best system overall

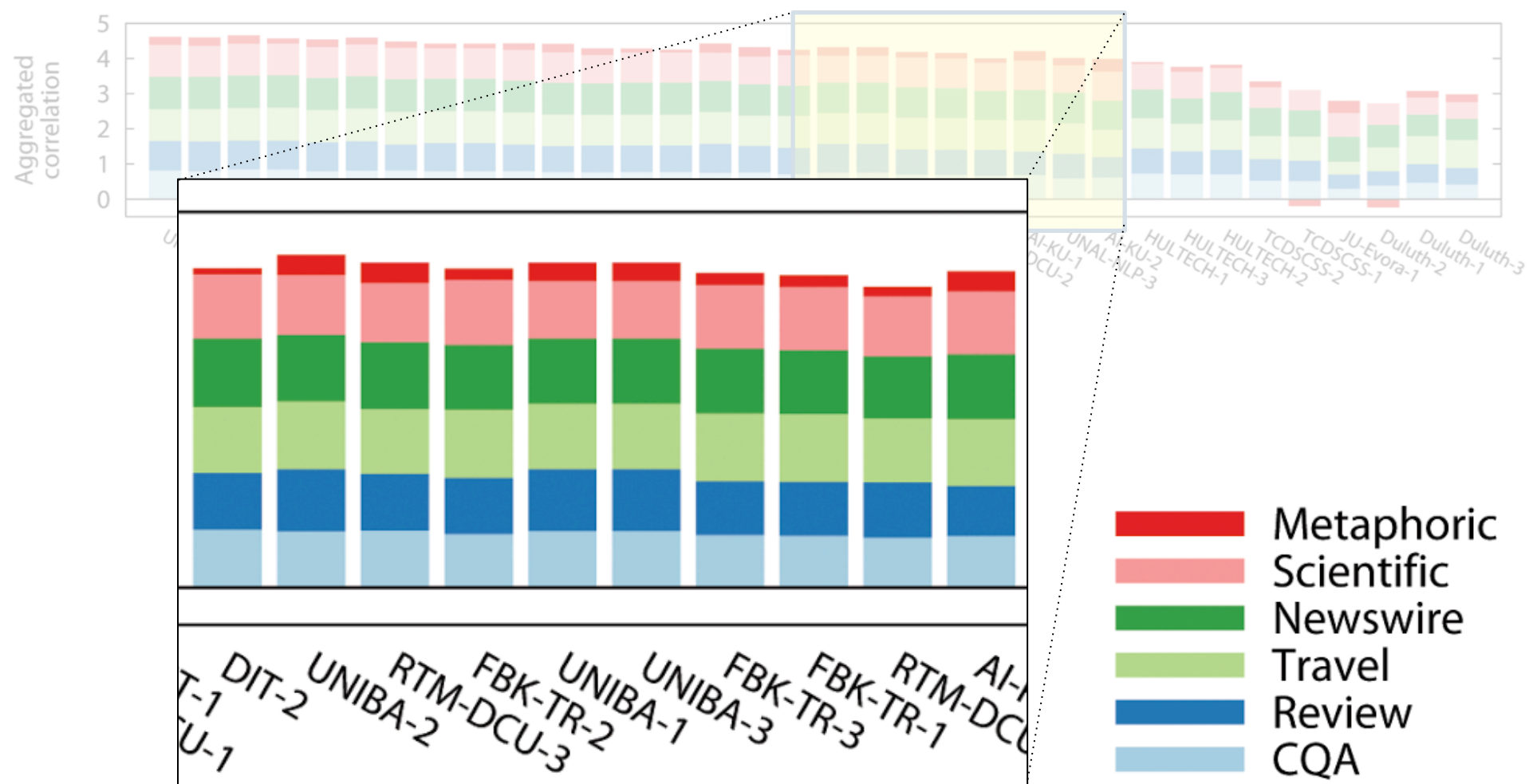
Utilizes only a set of simple string-similarity features based on soft cardinality (Jimenez et al, 2010).

UNAL-NLP *run1*, ranked 5th, is unsupervised: mirroring the potential for unsupervised semantic similarity measured seen in the recent work of Sultan et al (2014, 2015).

Correlation per genre paragraph-to-sentence

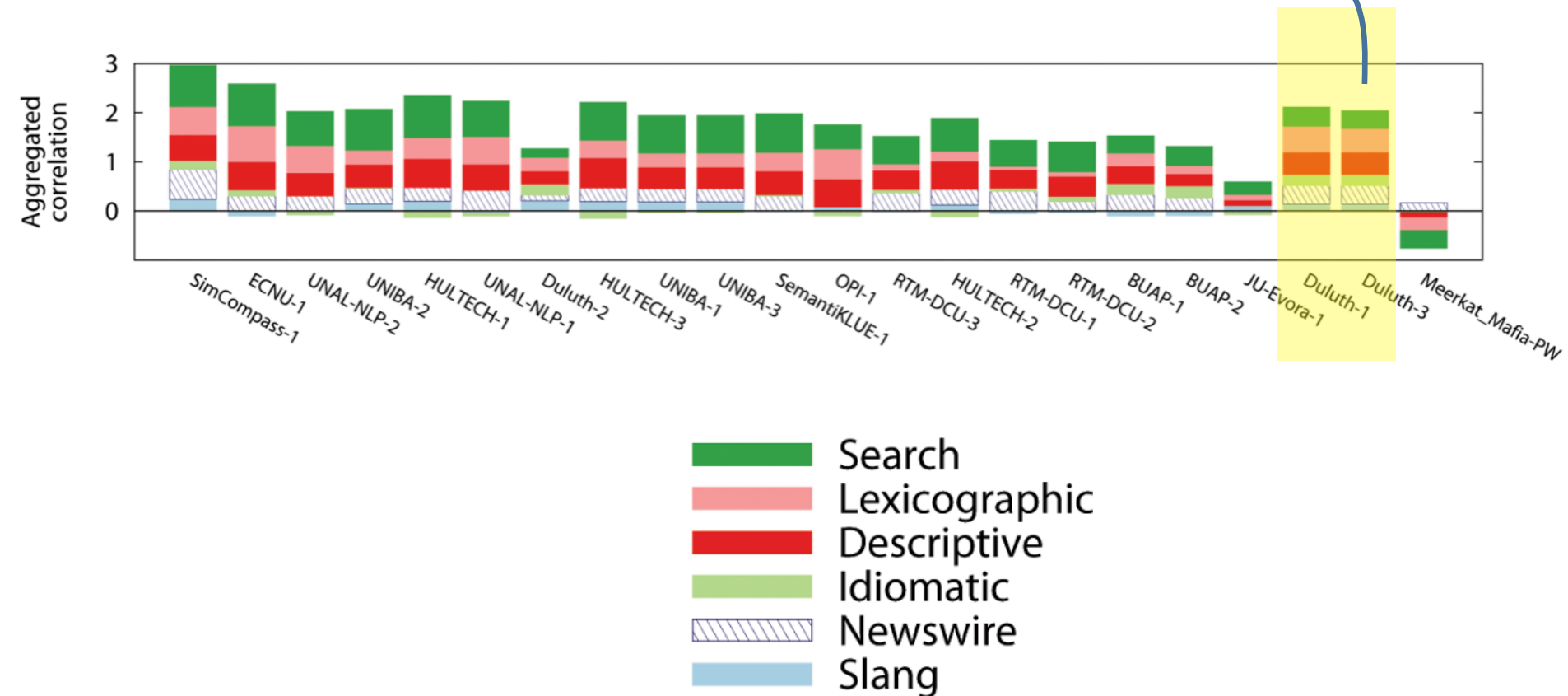


Correlation per genre paragraph-to-sentence



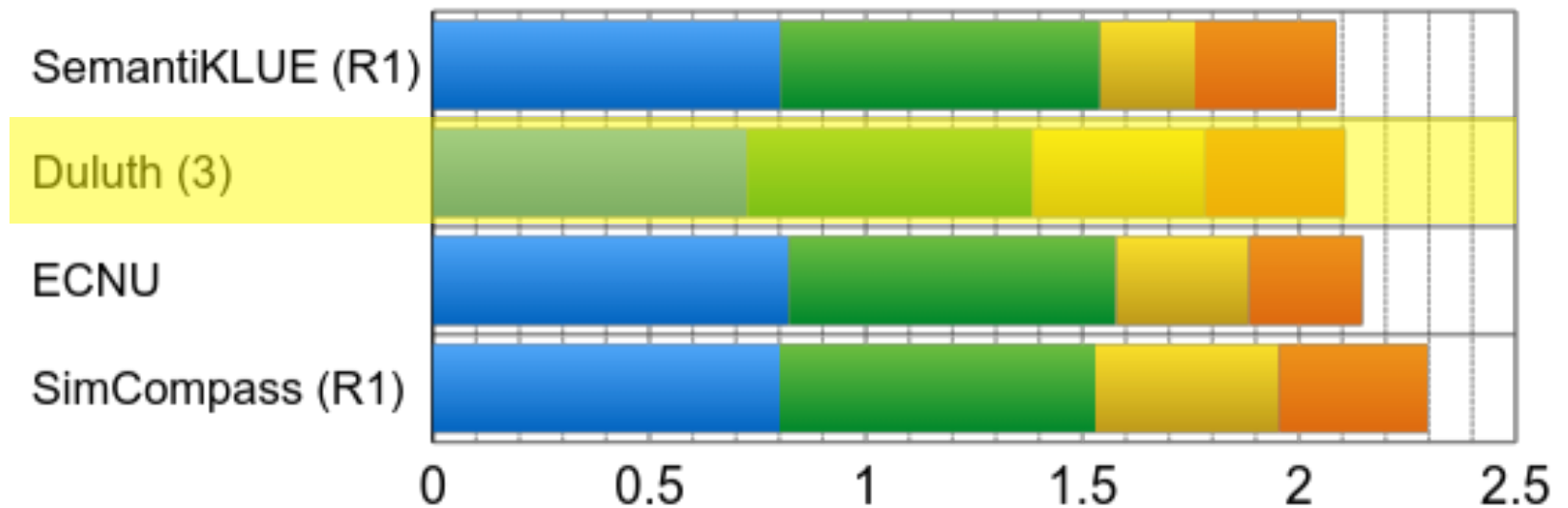
Correlation per genre phrase-to-word

30th (of 38) according to Pearson but 2nd according to Spearman



Ranking by similarity was easier for some systems

Switch from Pearson correlation to Spearman's ρ
(Ranked #30/38 according to Pearson's r)



What makes the task difficult?

Handling OOV words and novel usages

How often do draik eggs come in Merifoods in Meridell?

frequency of draik eggs in Merifoods

Brown-headed Cowbird in Arlington, Massachusetts

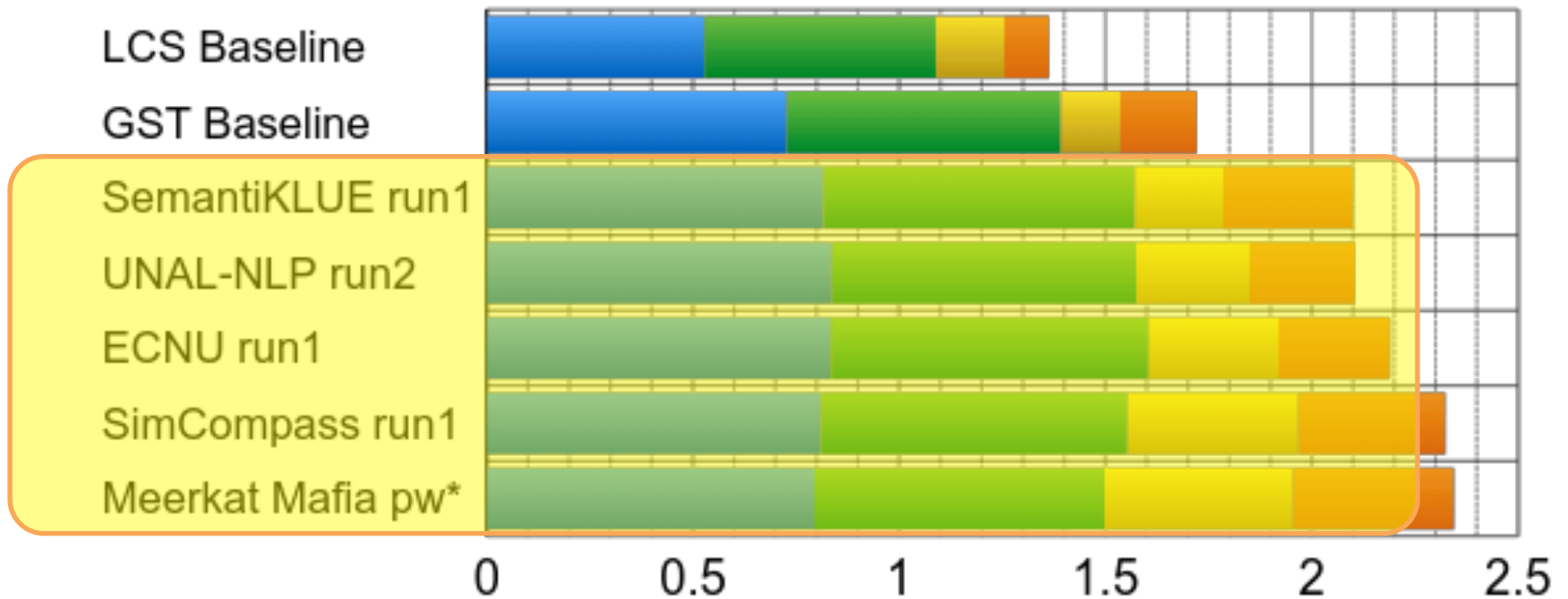
aviary

Hard feelings

grudge

WordNet alone is too limited

Include multiple dictionaries or
use distributional methods



■ paragraph-sentence

Dealing with *social media* text

can i [watch](#) 4od bbc iplayer etc with 10GB useage allowence?

online television streaming for bbc

Can d Internet companies see which websyts ive bin visiting?

internet provider's knowledge of my actions

Fables

A Groom used to spend whole days in currycombing and rubbing down his Horse, but at the same time stole his oats and sold them for his own profit. “Alas!” said the Horse, “if you really wish me to be in good condition, you should groom me less, and feed me more.”

Horses need food to look their best.

Fables in real world

The Fields Medals are regarded as mathematics' Nobel Prize, and are awarded every four years. All the previous 52 winners of the Fields have been men since its inception in 1936.

Mathematics is a male-dominated research area.

Open Source Tools for Semantic Similarity

Tools

WordNet::Similarity

- Word and sense similarity (Ted Pederson)

in Perl

but also available in Java, by Hideki Shima

WS4J: <http://code.google.com/p/ws4j/>

Leacock & Chodorow (1998)

Jiang & Conrath (1997)

Resnik (1995)

Lin (1998)

Hirst & St-Onge (1998)

Wu & Palmer (1994)

The extended gloss overlap measure by Banerjee and Pedersen (2002)

Two measures based on context vectors by Patwardhan (2003).

Tools

Align, Disambiguate and Walk: ADW (ACL 2013)

- Multi-level similarity
from word senses to texts
- Implicit word sense disambiguation
- Publicly available in Java

<https://github.com/pilehvar/adw>

Tools

Align, Disambiguate and Walk: ADW

Online demo at

<http://lcl.uniroma1.it/adw/>

Input the two lexical items ?

fire#v

Input type: Detect automatically ?

terminate#v

Input type: Detect automatically ?

Alignment-based disambiguation? ☒ Yes ☐ No ?

Calculate similarity

Tools

DKProSimilarity

<https://github.com/dkpro/dkpro-similarity>

- Open source framework for text similarity, Java
- Best system STS-12
- Several similarity measures, including:

algorithms.lexical	GreedyStringTiling, Levenshtein, NGramBased, ...
algorithms.lsr	Based on WordNet or Wikipedia
algorithms.style	FunctionWordFrequency, MTLD, TypeTokenRatio
algorithms.vsm	Vector-space models, e.g. ESA
algorithms.wikipedia	Special Wikipedia measures, e.g., WikipediaLinkMeasure

Tools

TakeLab

<http://takelab.fer.hr/sts/>

- Open source framework for text similarity, Python
- Among the top five in STS-12
- Several similarity measures, including:

Lexical	WordNet-based measures from NLTK
Knowledge-based	GreedyStringTiling, Levenshtein, NGramBased, etc
Corpus-based	LSA
Syntactic	Syntactic role similarity, syntactic dependency similarity
Other	Normalized differences, number overlap, etc.

Tools

S-Space

<https://github.com/fozziethebeat/S-Space>

- Open source framework for word distributions
- Written in Java
- Support for common weighting (e.g., PMI) and matrix factorizations (e.g., SVD)
- Implements many common algorithms in a single interface
 - LSA, word2vec, COALS, random indexing
- Integrated pre-processing support using Stanford CoreNLP

Tools

DISSECT

<http://clic.cimec.unitn.it/composes/toolkit/>

- Open source framework for word distributions
- Written in Python
- Support for common weighting (e.g., PMI) and matrix factorizations (e.g., SVD)
- Designed around compositionality
 - Easy to build representation for larger units of text

Tools

Gensim

<https://radimrehurek.com/gensim/>

- Originally written for high-performance LSA
- Now includes support for many kinds of topic modeling and word2vec
 - Usually where new algorithms get first implemented
- Fast and written in Python

Tools

Word2vec

<https://code.google.com/p/word2vec/>

- Tomas Mikolov (in C)
- Efficient implementation of the continuous bag-of-words and skip-gram architectures for **word** representation
- Also available in
 - Java: DL4J, Deep Learning 4 Java
<http://deeplearning4j.org/word2vec.html>
 - Spark MLlib: <https://spark.apache.org/docs/latest/mllib-feature-extraction.html#word2vec>
 - Python: as a part of gensim <http://radimrehurek.com/2013/09/deep-learning-with-word2vec-and-gensim/>

Tools

Doc2vec

<https://radimrehurek.com/gensim/models/doc2vec.html>

- As a part of gensim
- Efficient implementation of the continuous bag-of-words and skip-gram architectures for **paragraph** representation

Tools

NLTK

<http://www.nltk.org/>

- A large NLP package with support for many kinds of operations on text
- Integrated with WordNet with easy support for most sense- and word-similarity measures
- Written in Python

Tools

<http://mallet.cs.umass.edu/>

MALLET - MACHine Learning for Language Toolkit

- A software package for building probabilistic models using text
- Scalable and fast support for LDA and the hierarchical Pachinko Allocation Model
- Written in Java

Tools

Other topic models

- Hugh list of topic modeling software available at http://www.cs.columbia.edu/~blei/topicmodeling_software.html with an active mailing list too
- Highlights include
 - LDA in C (fast!)
 - HDP in C
 - TurboTopics in Python

Resources for OOV words (1)

- MESH or other domain specific terminologies
- Wiktionary
- Wordnik: “world's biggest online English dictionary”
- Collaborative International Dictionary of English
- Moby Thesaurus II by Grady Ward
- The Jargon File
- The Free On-line Dictionary of Computing
- The CIA World Factbook 2002 (world02)

Resources for OOV words (2)

Tools

CROWN

<https://github.com/davidjurgens/crown>

- Extension of WordNet with new synsets and lexicalizations
 - 2X the size of WordNet
 - Slang, archaic forms, idioms, technical words, ...
- Released as stand-off dictionaries, so compatible with all WordNet libraries

Pre-trained Word Vectors

- Word2vec
 - <https://code.google.com/p/word2vec/>
 - <https://github.com/3Top/word2vec-api>
- Baroni and Lenci, Distributional memory
 - <http://clic.cimec.unitn.it/dm/>
- GloVe
 - <http://nlp.stanford.edu/projects/glove/>
- Faruqui and Dyer (ACL 2014)
 - <http://wordvectors.org/>
- Huang et al (2012), Improving Word Representations via Global Context and Multiple Word Prototypes
- Levy and Goldberg (2014), dependency-based word embeddings

Open Problems in Semantic Similarity

Open Problem: Irregular Language

can i watch 4od bbc iplayer etc with 10GB useage allowance?

online television streaming for bbc

Can d Internet companies see which websyts ive bin visiting?

internet provider's knowledge of my actions

Open Problem: Multi-word Expressions (MWEs)

- Most approaches either ignore MWEs or recognize those from fixed lists of MWEs
 - Problematic unless lemmatizing
 - Even more problematic with syntactic rearrangement

We need to **sort out** the problem

We need to **sort** the problem **out**

- New SemEval-2016 task on super-sense tagging seems like a promising direction for addressing this

Open Problem: Cross-Language Similarity

- Beneficial for Machine Translation evaluation or even applications like plagiarism detection
- Recent benchmarks by Camacho-collados et al. (2015) and Leviant and Reichart (2015)

Cross-lingual datasets constructed based on RG-65 (FR, DE, EN, FA, ES, and PT) and WS353 (EN, DE, IT, and RU)

<http://lcl.uniroma1.it/similarity-datasets/>

<http://technion.ac.il/~irakr/MultilingualVSMdata.html>

Open Problem: Syntax

- Syntax matters
 - “Man bites dog”
 - “Dog bites man”
 - “Pitbull bites man”
- Vector addition would fail in these cases**
- Compositionality can help here but more analysis is needed
 - Recent SICK benchmark designed to explicit test for compositional ability (Marelli et al., 2014)
 - Possible solution with Abstract Meaning Representations (AMRs)
 - Check out SemEval-2016’s task!

Open Problem: Punctuations!

A woman without her man is nothing.

A woman: without her, man is nothing.

Open Problem: Variable-Sized Input

The 30-year-old woman has had "no contact with the outside world." 30-year-old female recluse

Prius

A fuel-efficient hybrid car

An automobile powered by both an internal combustion engine and an electric motor, reducing its dependence on fossil fuels

Requires smarter compositionality

Open Problem: Ambiguity

- Multiple interpretations can wreak havoc when text is limited

The boss **fired** his
worker.

An employee was
terminated
from work by his boss.

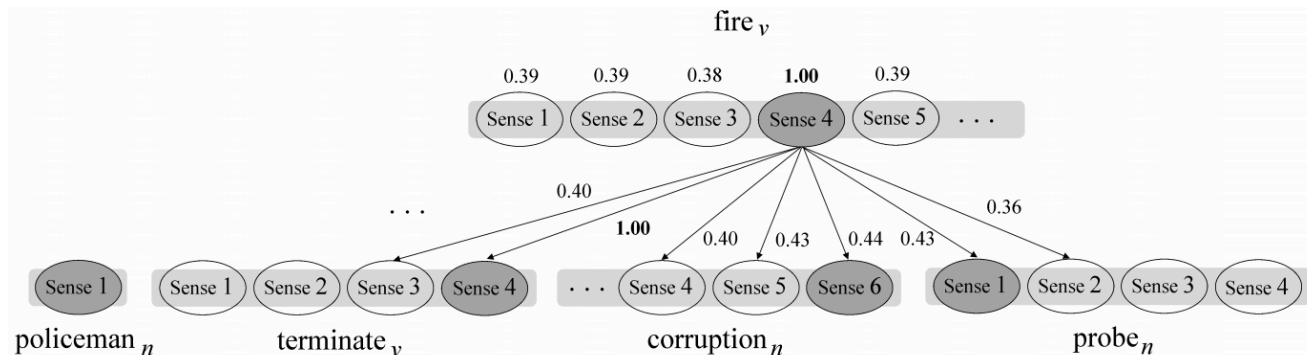


A worker was **shot**
by his boss.



Open Problem: Ambiguity

- Alignment-based disambiguation of ADW



- WSD is a solution, but is still a long way off

Babelfy

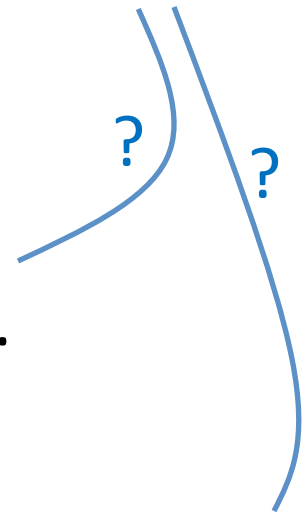


Open Problem: Subjectivity vs. objectivity

As of 2012, there are 2.1 million hybrids on U.S. roads.

Hybrid cars are getting quite popular in U.S.

US hybrid vehicle market share grew by 41% in 2012.



Open Problem: Uncovered words

- Words might not have been covered in the corpus or by the lexicon;
- For instance, some WordNet OOV words:
 - prequel#n
 - fanbase#n
 - screenshot#n
 - bookmark#v
 - programmatic#a
 - broadband#n
 - And many more regular terms
 - photoshop#v
 - space_cadet#n
 - homewrecker#n
 - And many more slang terms

Open Problem: Evaluation

Many evaluation tasks make it easy to pick-and-choose which results to report

20+ choices for work similarity!

What exactly is state of the art?

Similarity itself is not an end-task, yet most approaches are only tested on STS benchmarks, not in any application.

No easily-pluggable application-based tests

Semantic Similarity Frontiers: From Concepts to Documents

David Jurgens

jurgens@stanford.edu

Stanford University

Mohammad Taher Pilehvar

pilehvar@di.uniroma1.it

Sapienza University of Rome

Slides, bibliography, extended reading list,
and all other materials available at

<http://tiny.cc/similarity-tutorial>



ERC grant 259234

Bonus: Must-see papers at EMNLP!

- J. Li and D. Jurafsky: Do Multi-Sense Embeddings Improve Natural Language Understanding?
- H. He et al: Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks
- D. Kiela et al: Specializing Word Embeddings for Similarity or Relatedness
- J. Wieting and D. Roth: Latent Variable Regression for Text Similarity and Textual Entailment
- Sergienya and Schutze: Learning Better Embeddings for Rare Words Using Distributional Representations
- A. Gupta et al: Distributional vectors encode referential attributes