Distributional Semantics: Composition Models and Applications to Typology

Denis Paperno

TyLex, 07.09.2017

DSM Creation

when dogs bark
a dog today a cat tomorrow
have you walked the dog
there was a dog in the park
a dog is a man's best friend
a sad old dog was sleeping
have you walked the dog

context	count
bark	1843
friend	2540
walk	4366
...	...

weighting

$$\log\left(\frac{f(w,c)N}{f(c)f(w)}\right)$$

context	dim
bark	2.4
friend	0.3
walk	0.6
...	...

SVD, NMF, SkipGram, Glove...

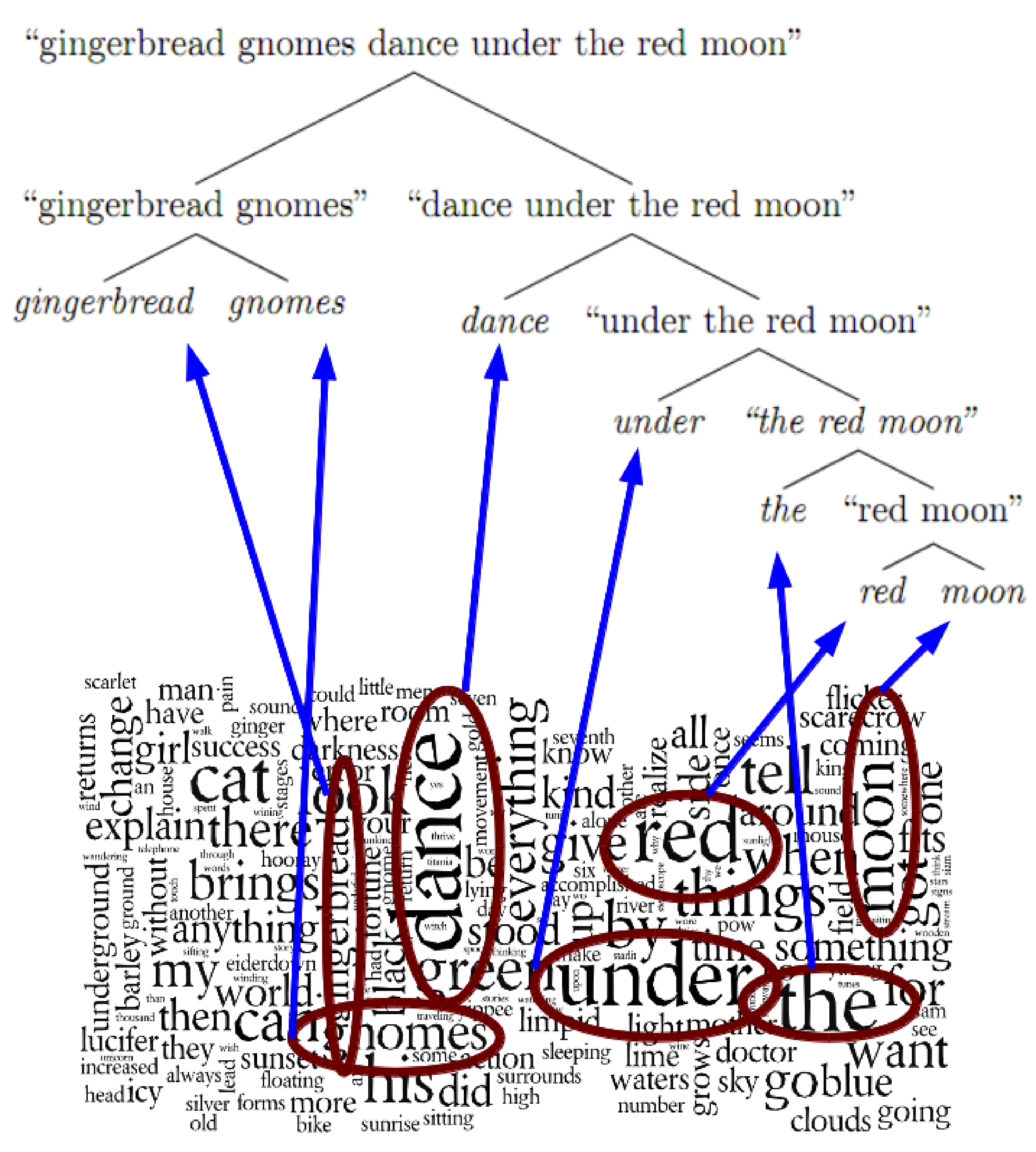
Dimentionality reduction

dog	-1.78	-1.62	-1.9	1.16	0.86	0.16	1.2	-1.34	1.1

Compositionality

- Programmatic article:

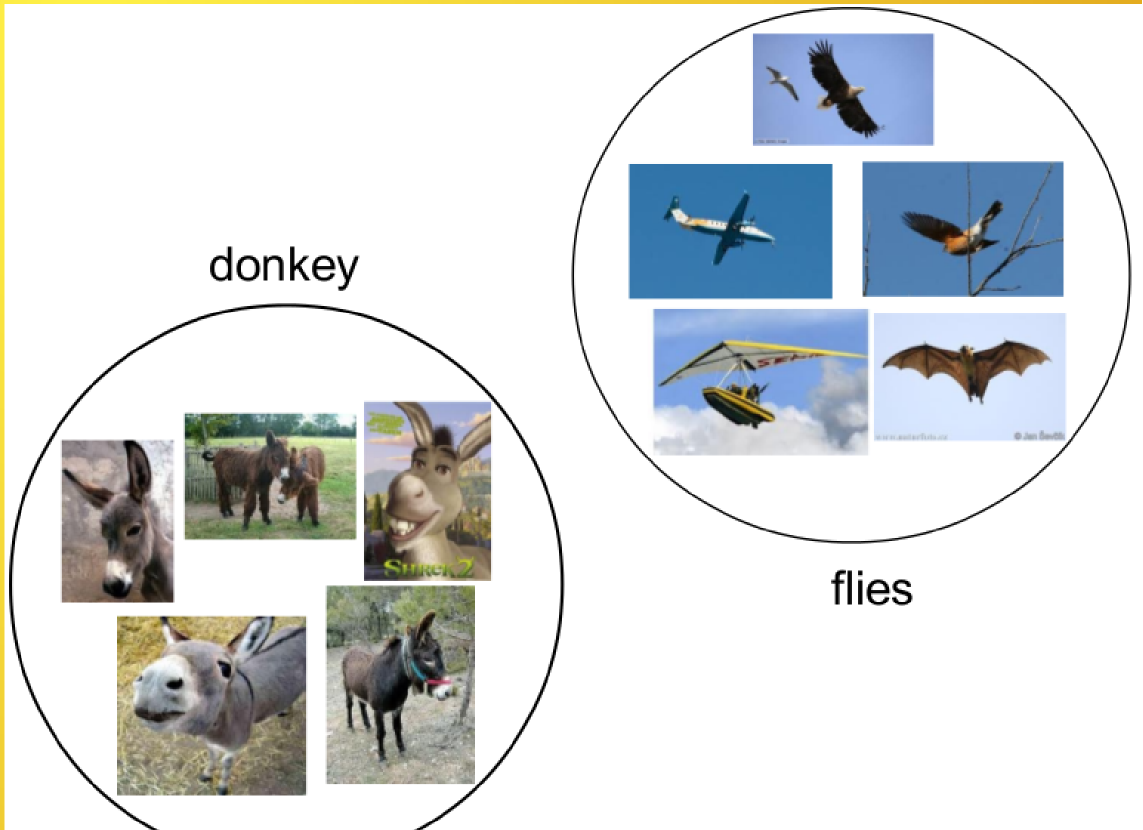
M. Baroni, R. Bernardi and R. Zamparelli. 2014. Frege in space: A program for compositional distributional semantics. Linguistic Issues in Language Technologies 9(6): 5-110.



Compositionality in formal semantics

- SOME: function, takes 2 sets A and B
  - Returns True if A and B share an element
  - Returns False otherwise

- Some donkey flies



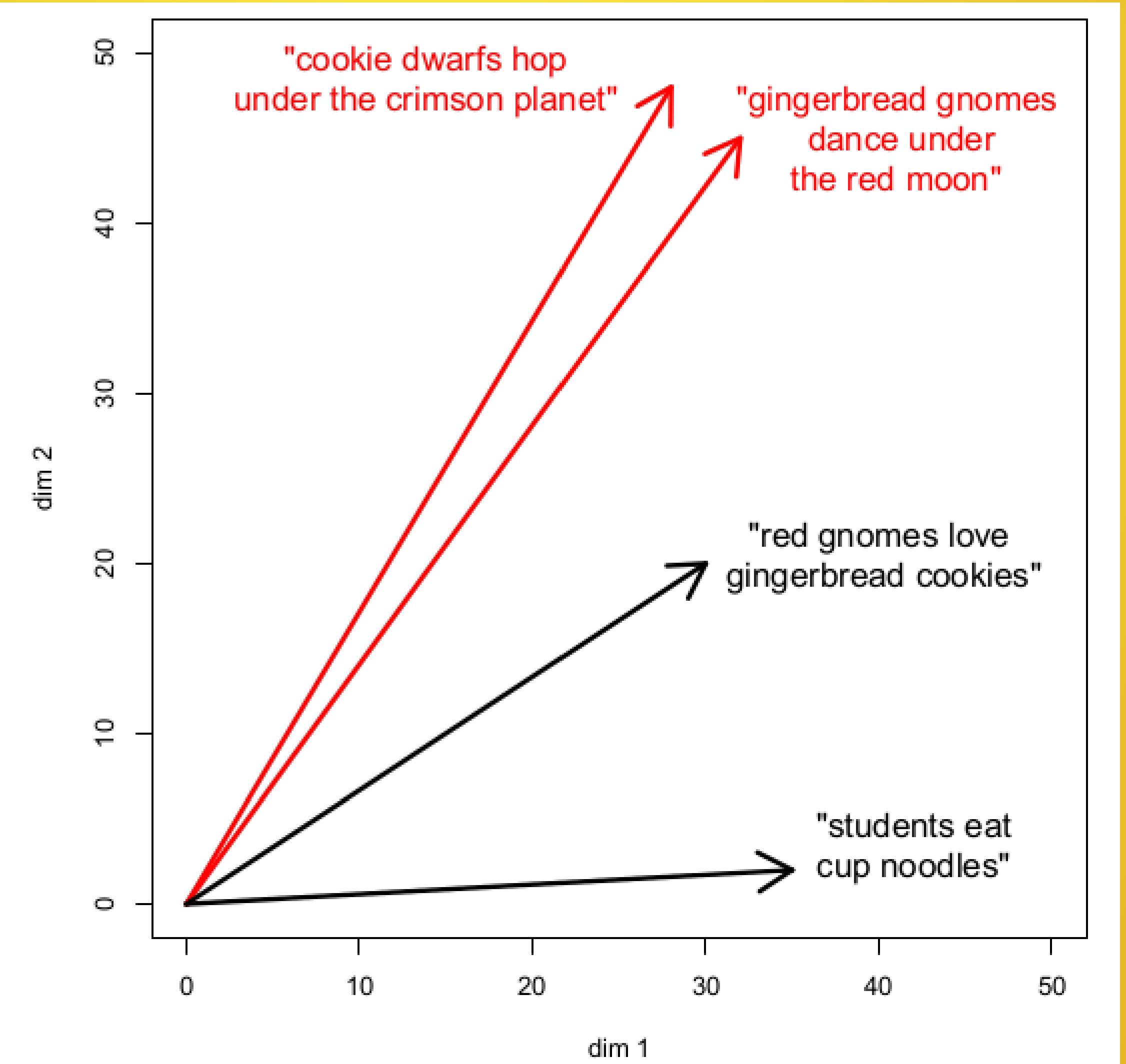
Composition models

- Needed if we want to use vectors for phrases, sentences, etc.:

	planet	night	full	blood	shine
moon	10	22	43	3	29
red moon	12	21	40	20	28
the red moon shines	11	23	21	15	45

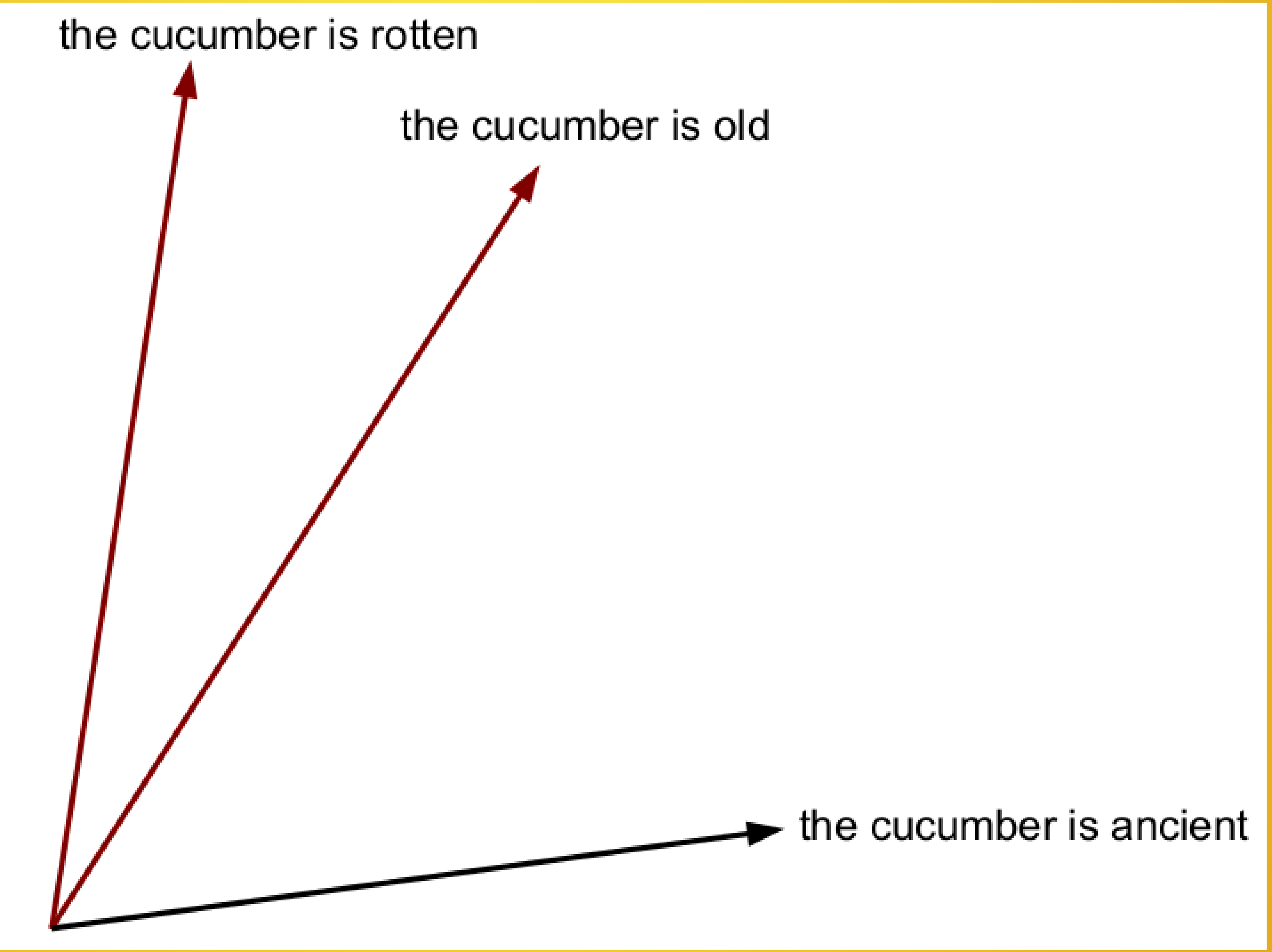
Possible applications

- Paraphrasing



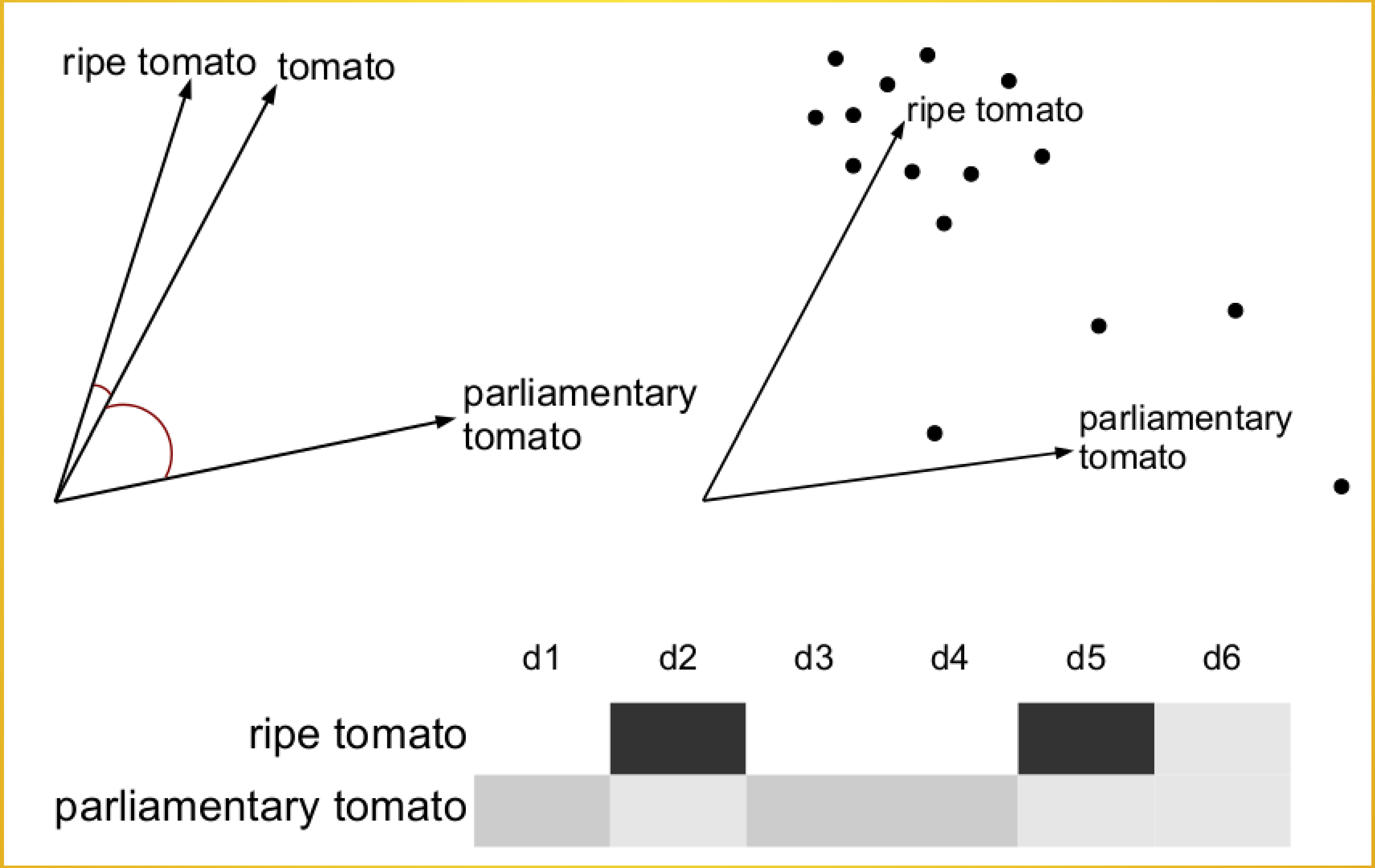
Possible applications

- Contextual disambiguation



Possible applications

- Semantic plausibility



Semantic composition: how?

- Pointwise addition and multiplication:

	planet	night	space	color	blood	brown
red	15.3	3.7	2.2	24.3	19.1	20.2
moon	24.3	15.2	20.1	3.0	1.2	0.5
red+moon	39.6	18.9	22.3	27.3	20.3	20.7
red$\odot$moon	371.8	56.2	44.2	72.9	22.9	10.1
red(moon)	24.6	19.3	12.4	22.6	23.9	7.1

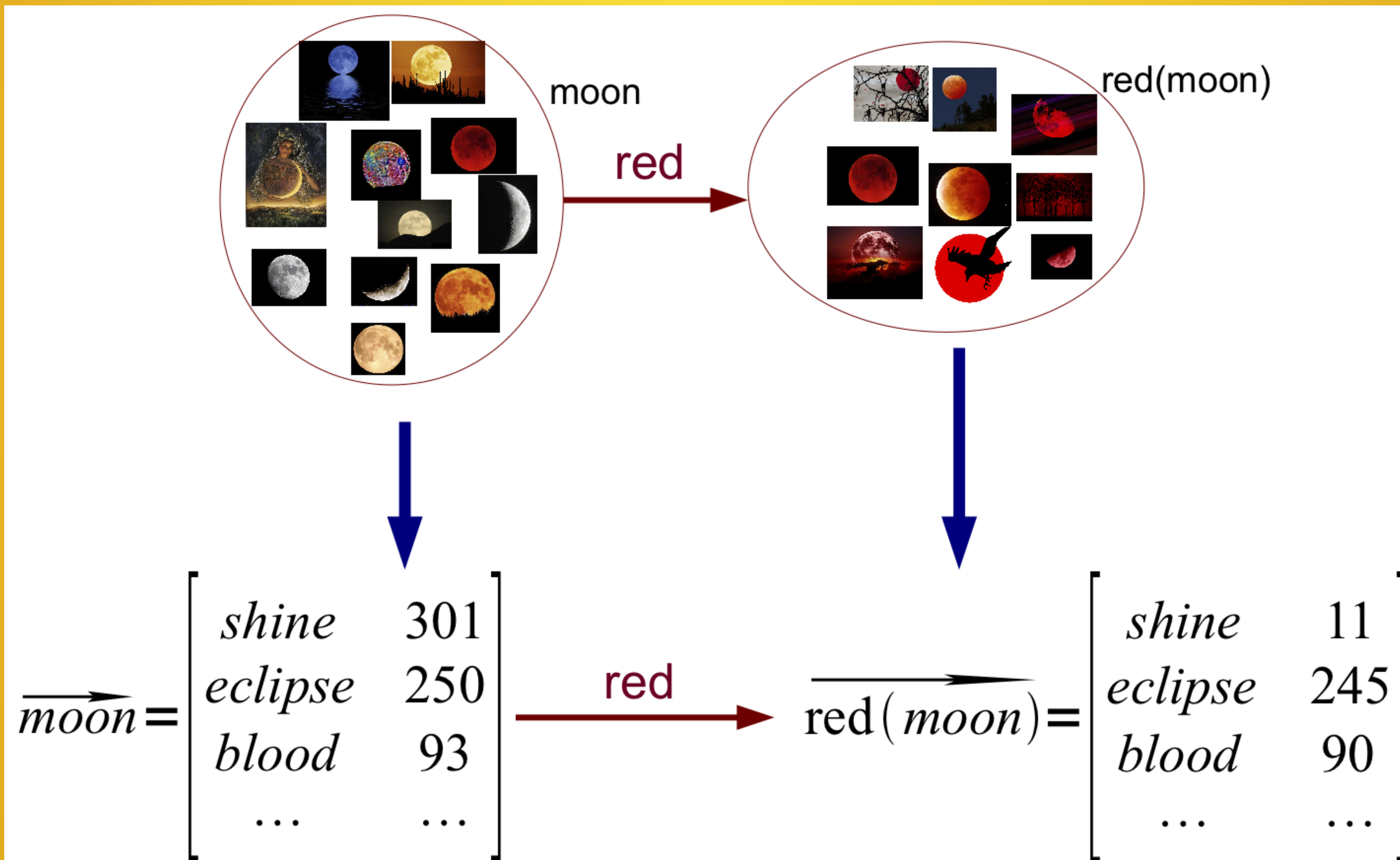
Weighted addition

- (Mitchell and Lapata 2010)

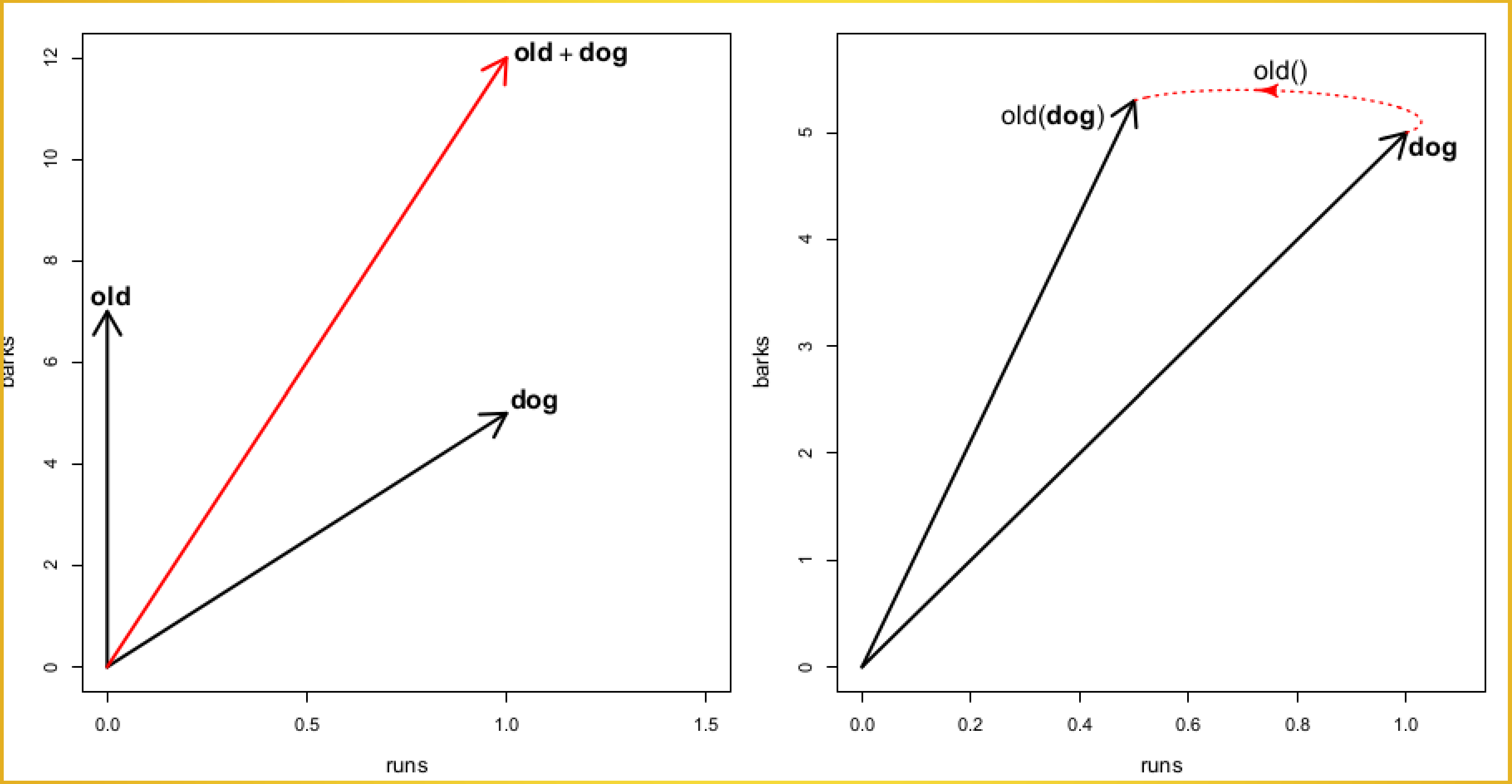
$$\vec{p} = \alpha\vec{u} + \beta\vec{v}$$

	music	solution	economy	craft	reasonable
practical	0	6	2	10	4
difficulty	1	8	4	4	0
practical + difficulty	1	14	6	14	4
0.4×practical + 0.6×difficulty	0.6	5.6	3.2	6.4	1.6

Lexical function



Addition vs. lexical function



Linear Mapping

- “Adjectives are matrices” (Baroni and Zamparelli 2012)

$$\overrightarrow{c} = \mathbf{A}\overrightarrow{n}$$

In practice: collect phrase vectors

```
n and the moon shining i
 with the moon shining s
rainbowed moon . And the
 crescent moon , thrille
in a blue moon only , wi
now , the moon has risen
d now the moon rises , f
y at full moon , get up
 crescent moon . Mr Angu

f a large red moon , Campana
, a blood red moon hung over
 glorious red moon turning t
The round red moon , she 's
l a blood red moon emerged f
n rains , red moon blows , w
```

	shine	blood	Soviet
moon	301	93	1
red moon	11	90	0
army	2	454	20
red army	0	22	18

...and estimate the matrix for the adjective

- Input:
  - N: matrix of noun vectors
  - C: matrix of adj-noun observed phrase vectors

$$\begin{array}{llll} c^1 & \overrightarrow{\text{red.moon}} & n^1 & \overrightarrow{\text{moon}} \\ c^2 & \overrightarrow{\text{red.army}} & n^2 & \overrightarrow{\text{army}} \\ c^3 & \overrightarrow{\text{red.car}} & n^3 & \overrightarrow{\text{car}} \end{array}$$

- Estimate:
  - Matrix of function word ($\mathbf{A}_{\mathbf{red}}$)

Application to Typology: Semantic Typology of Adjectives

Ryzhova, Kyuseva and Paperno. Typology of Adjectives Benchmark for Compositional Distributional Models. Proceedings of Language Resources and Evaluation, 2016

Semantic Maps

- “attempts to visually represent cross-linguistical regularity in semantic structure”

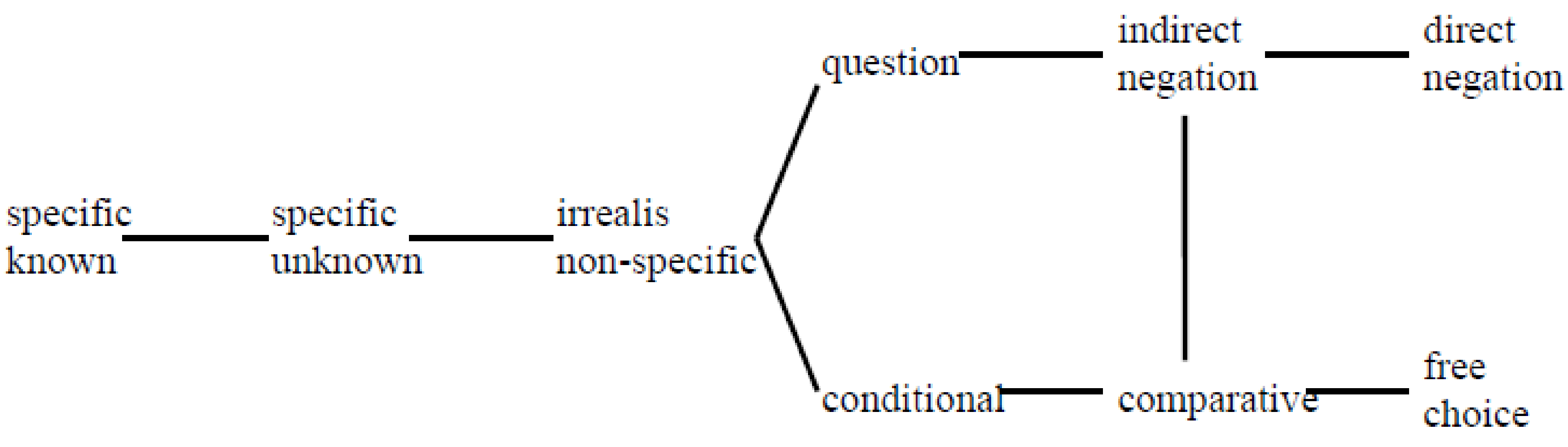
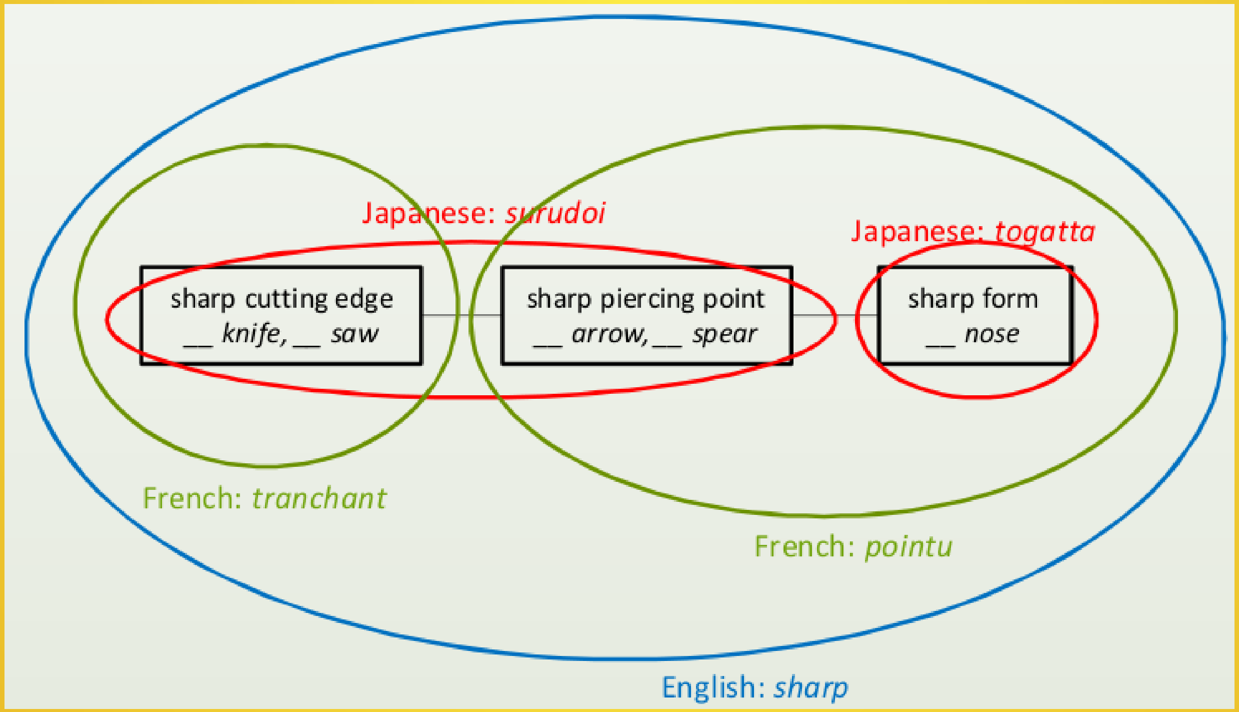


Figure 1
Haspelmath's (1997) semantic map of indefinite pronouns

Lexical Semantic Variation



Typological vector space

- 1: lexical item covers a given usage
- 0: lexical item does not cover a given usage

	English *sharp*	Japanese *surudoi*	Japanese *togatta*	French *tranchant*	French *pointu*
… knife	1	1	0	1	0
… saw	1	1	0	1	0
… arrow	1	1	0	0	1
… spear	1	1	0	0	1
… nose	1	0	1	0	1

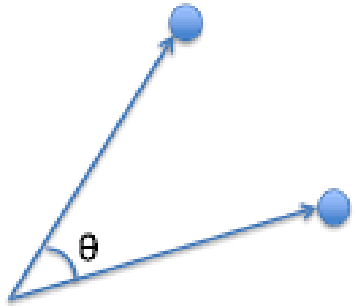
$\mathbf{w}_i$ = 1 iff *w* is covered by the lexeme l_i of some language in the database.

Typological Closeness

Typological closeness of two word usages w, w’ is a measure of how likely an arbitrary lexical item of an arbitrary human language that covers one of them is to cover both.

Typological closeness can be quantified as the **cosine of typological vectors** of word usages.

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



Simplified example

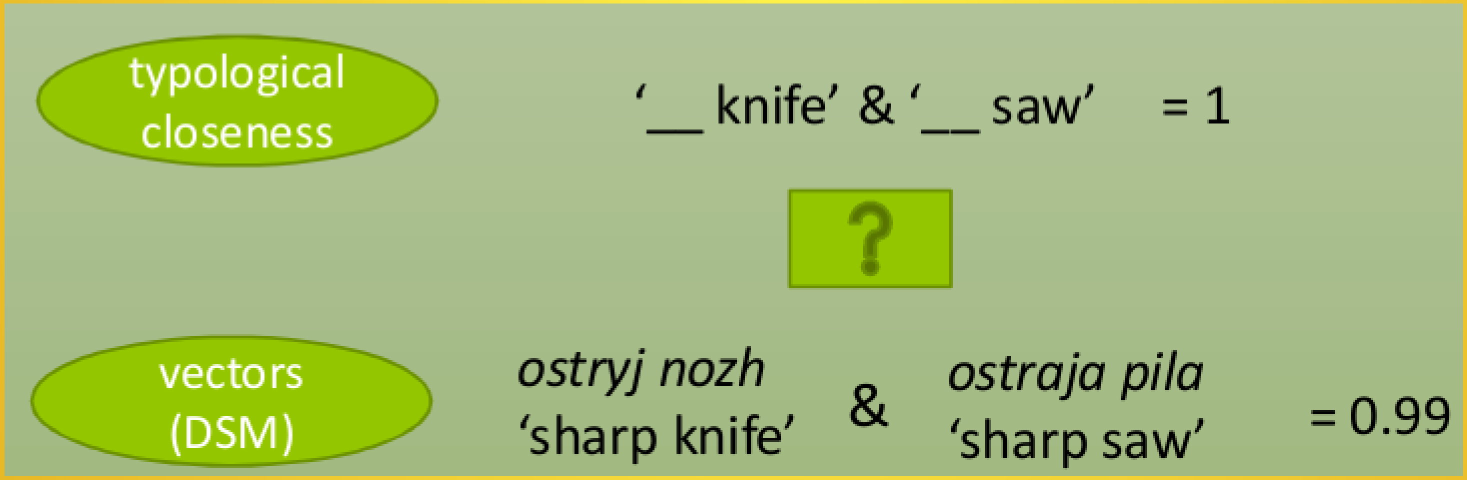
... saw	1	1	0	1	0
... arrow	1	1	0	0	1
... spear	1	1	0	0	1

$$sim(sharp\_arrow,sharp\_saw)=2/(3*3)^{1/2}=2/3=0.67$$

$$sim(sharp\_arrow,sharp\_spear)=3/(3*3)^{1/2}=3/3=1$$

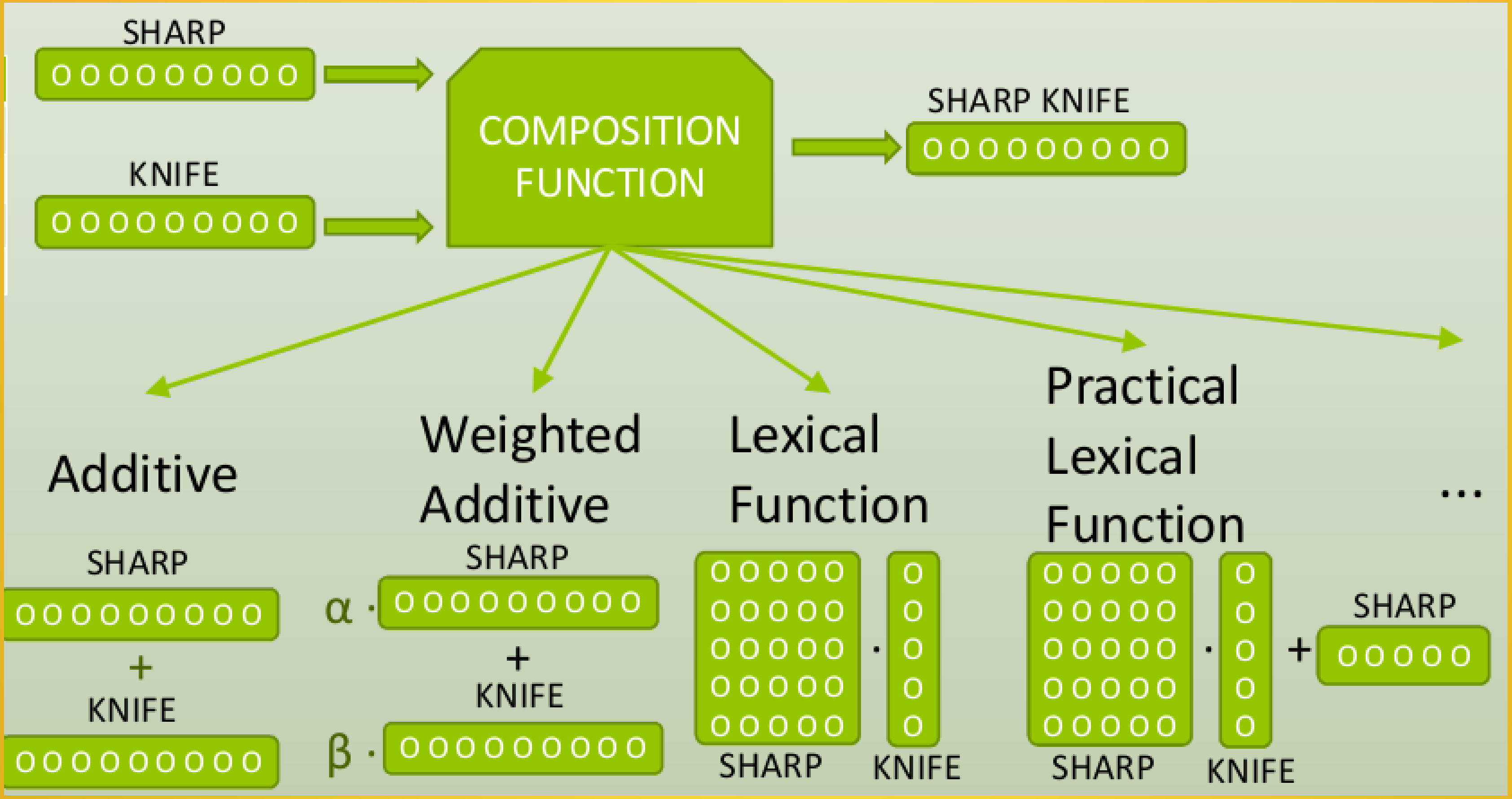
Evaluation

- How well is typological closeness correlated with closeness in a DSM?



Compositional DSM

- What is the vector for *sharp knife?*



Test data

semantic field 'sharp',

direct and figurative meanings

(9019 pairs of rows)

only direct meanings (528 pairs)

'smooth',

direct and figurative meanings (1992 pairs)

only direct meanings (561 pairs)

Results

	corpus	vectors	composition model	sharp	sharp:direct	smooth	smooth:direct
1	RNC	none	noun only	0.092	0.12	0.121	0.196
2	RNC	ppmi	noun only	0.139	0.167	0.237	0.21
3	RNC	ppmi,SVD	noun only	0.167	0.268	0.274	0.244
4	RNC	ppmi	none	0.097	0.194	0.134	0.154
5	RNC	ppmi	Additive	0.36	0.654	0.589	0.74
6	RNC	ppmi	Multiplicative	0.253	0.421	0.585	0.7
7	RNC	ppmi	Dilation	0.19	0.222	0.379	0.443
8	RNC	ppmi	Dilation w/ Training	0.207	0.35	0.249	0.313
9	RNC	ppmi, SVD	LexFunc	0.112	0.336	0.263	0.349
10	RNC	ppmi, SVD	LexFunc, Ridge	0.116	0.345	0.443	0.703
11	RNC	ppmi, SVD	PrLexFunc	0.389	0.765	0.444	0.931
12	RNC	ppmi, SVD	PrLexFunc, Ridge	0.39	**0.766**	0.449	**0.946**
13	RNC	none	WeightedAdd	0.443	0.754	0.589	0.849
14	RNC	plog	WeightedAdd	0.387	0.76	0.477	0.765
15	RNC	epmi	WeightedAdd	**0.462**	0.763	0.59	0.865
16	RNC	ppmi	WeightedAdd	0.42	0.764	**0.604**	0.905
17	RNC	plmi	WeightedAdd	0.443	0.762	0.603	0.791
18	all	ppmi	WeightedAdd	0.418	0.764	0.564	0.899
19	all	plmi	WeightedAdd	0.438	0.763	0.549	0.712
20	RNC	ppmi, SVD	Additive	0.269	0.443	0.404	0.566
21	RNC	ppmi, SVD	Dilation w/ Training	0.388	0.766	0.448	0.936
22	RNC	ppmi, SVD	WeightedAdd	0.388	0.717	0.421	0.682
23	RNC	ppmi, SVD	Dilation	0.231	0.519	0.374	0.512
24	RNC	ppmi, SVD	Multiplicative	0.062	0.41	0.194	0.228

Further experiment: Questionnaire construction

- Sharp knife, sharp sword, sharp saw
- Sharp needle, sharp arrow, sharp nail
- Sharp nose, sharp mountain, sharp elbow
- ….
- Sharp line, sharp photo, sharp contrast
- Sharp mind, sharp gaze, sharp girl
- ….

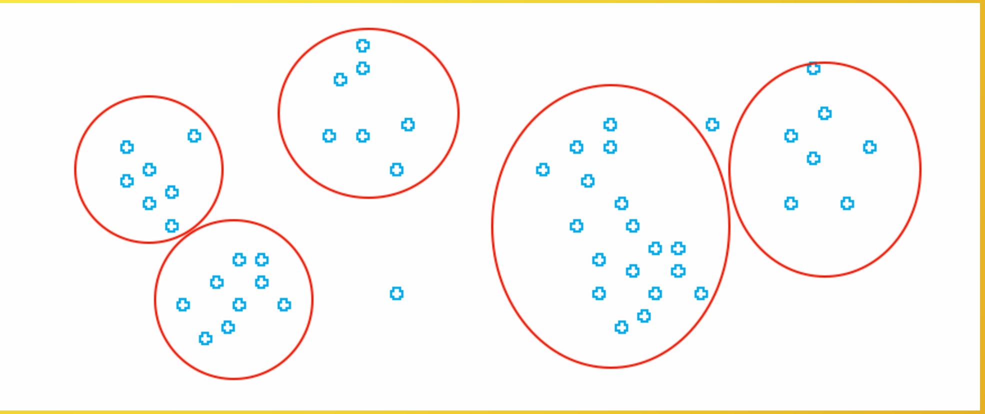
Procedure

1) Defining the lexemes that constitute the semantic field under investigation (with the help of translation dictionaries);

2) Collecting a list of contexts in which the target words occur (based on the data of the lemmatized RNC main subcorpus);

3) Creating a semantic vector space (with Distributional Semantic Models techniques);

4) Clustering the resulting space (with the hierarchical clustering algorithm);



5) Extracting three core elements from every cluster.

Examples of resulting clusters

<u>Example 1:</u>

prjamoj_stolb ‘straight pole’

prjamoj_dorozhka ‘straight path’

prjamoj_alleja ‘straight avenue’

<u>Example 2:</u>

prjamoj_potomok ‘direct descendant’

prjamoj_predshestvennik ‘direct predecessor’

prjamoj_nasledije ‘direct heritage’

Markup for evaluation

‘straight’: fragment of the dataset marked up by experts

prjamoj rjad ‘straight row’ 1

prjamaja linija ‘straight line’ 1

prjamoj udar ‘straight/direct blow’ 1|4

prjamoj dostup ‘direct access’ 6

prjamoj razgovor ‘direct/frank conversation’ 6|7

prjamaja ugroza ‘direct threat’ 7

Results

F = 2PR / (P+R)

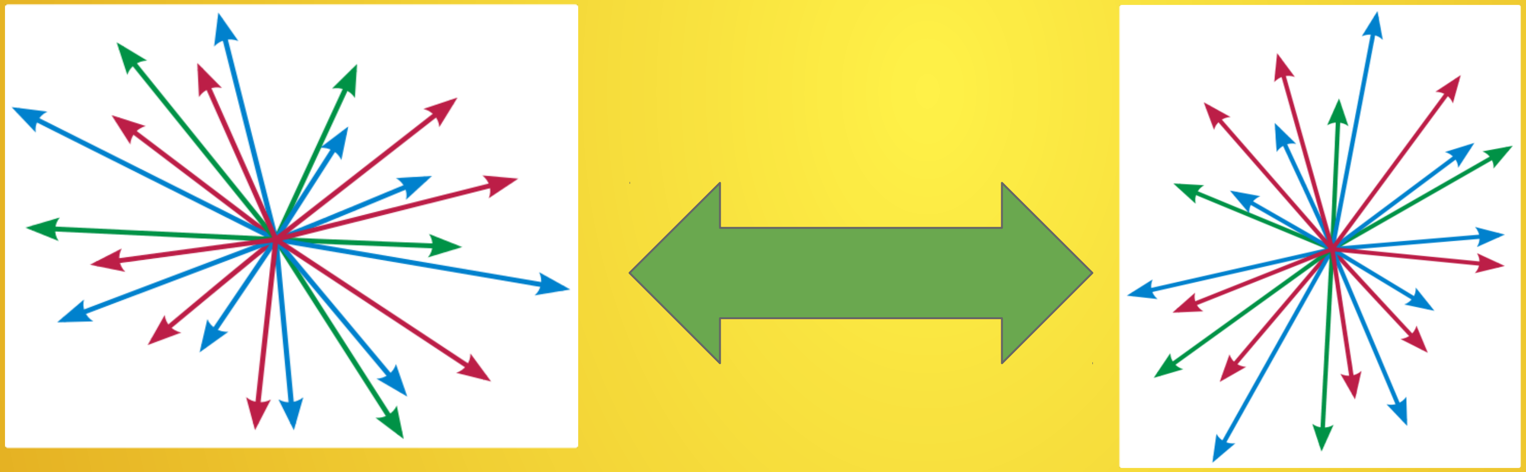
Recall, R: how many typological nodes are presented

Precision, P: (=purity), whether every cluster is homogenous or not

Best results: 0.903 for ‘sharp’, .884 for ‘straight’

Typological case: Conclusions

- Explicit notion of typological semantic space
- relation between typological and distributional spaces



- Inferred nodes in the typological space from CDSM.

Demo: the Dissect toolkit

- Would you like to do computations with distributional vector spaces?
- If you already use software packages for dealing with vector data (e.g. R, Matlab, numpy or Tensorflow), you are all set
- Otherwise you may try Dissect

  http://clic.cimec.unitn.it/composes/toolkit/
- Prerequisites: Linux system with Python 2.7

Programming not necessary

- Ready scripts for many basic operations

- Create vectors from cooccurrence data
- Compute and evaluate similarity scores
- Find nearest neighbors in a vector space
- Train and apply composition models

Thank you!

- And let's stay in touch

  denis.paperno@gmail.com

- Thanks to Daria Ryzhova for the help with reading assignments