CNRS - INPT - UPS - UT1 - UTM

Integrating Type Theory and Distributional Semantics

A case-study on adjective-noun compositions

Tim Van de Cruys

Joint work with Marta Abrusan & Nicholas Asher
CNRS & IRIT, Toulouse

Wednesday, 21 September 2016

Introduction

- Formal semantics provides an elaborate framework for logical inference
  - Lambda-calculus
  - Quantification
  - Negation
- But: mostly theoretical, few data-driven methods
- Symbolic approach: words are considered as atomic entities, no internal semantic structure
- $\rightarrow$ Formal semantics has relatively little to say about lexical semantics

Distributional semantics

- The induction of meaning from text is based on the DISTRIBUTIONAL HYPOTHESIS [Harris 1954]
  - Take a word and its contexts:
    - tasty *sooluceps*
    - sweet *sooluceps*
    - stale *sooluceps*
    - freshly baked *sooluceps*
- By looking at a word's context, one can infer its meaning

Distributional semantics

- The induction of meaning from text is based on the DISTRIBUTIONAL HYPOTHESIS [Harris 1954]

  - Take a word and its contexts:
    - tasty *sooluceps*
    - sweet *sooluceps*
    - stale *sooluceps*
    - freshly baked *sooluceps*

  $\Rightarrow$ **food**

- By looking at a word's context, one can infer its meaning

Distributional semantics

- The induction of meaning from text is based on the DISTRIBUTIONAL HYPOTHESIS [Harris 1954]

  - Take a word and its contexts:
    - tasty *sooluceps*
    - sweet *sooluceps*
    - stale *sooluceps*
    - freshly baked *sooluceps*

- By looking at a word's context, one can infer its meaning

Word vectors

- captures co-occurrence frequencies of two entities

	red	tasty	fast	second-hand
raspberry	2	1	0	0
strawberry	2	2	0	0
car	1	0	1	2
truck	1	0	1	1

Word vectors

- captures co-occurrence frequencies of two entities

	red	tasty	fast	second-hand
raspberry	7	9	0	0
strawberry	12	6	0	0
car	7	0	8	4
truck	2	0	3	4

Word vectors

- captures co-occurrence frequencies of two entities

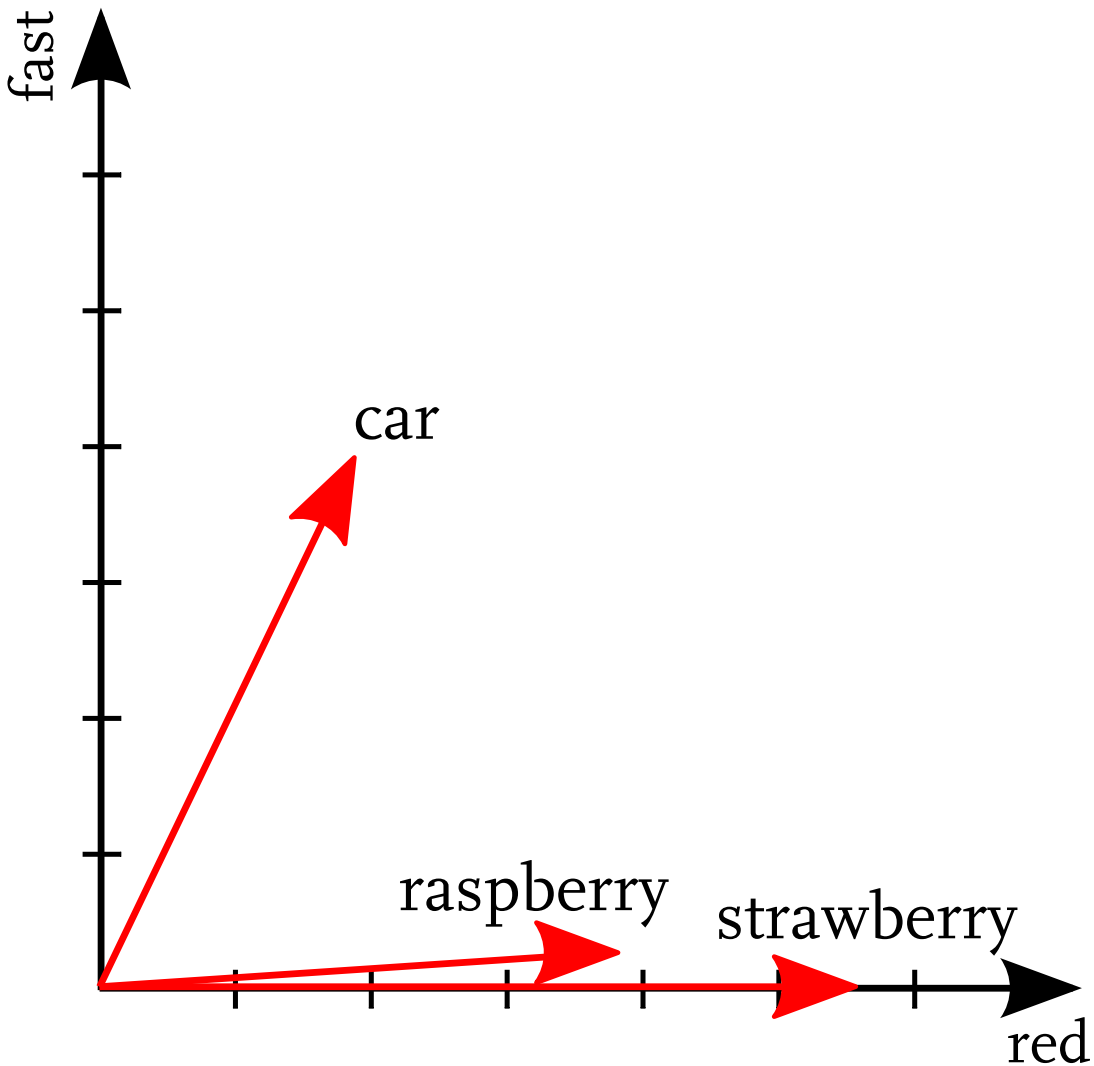
	red	tasty	fast	second-hand
raspberry	56	98	0	0
strawberry	44	34	0	0
car	23	0	31	39
truck	4	0	18	29

Word vectors

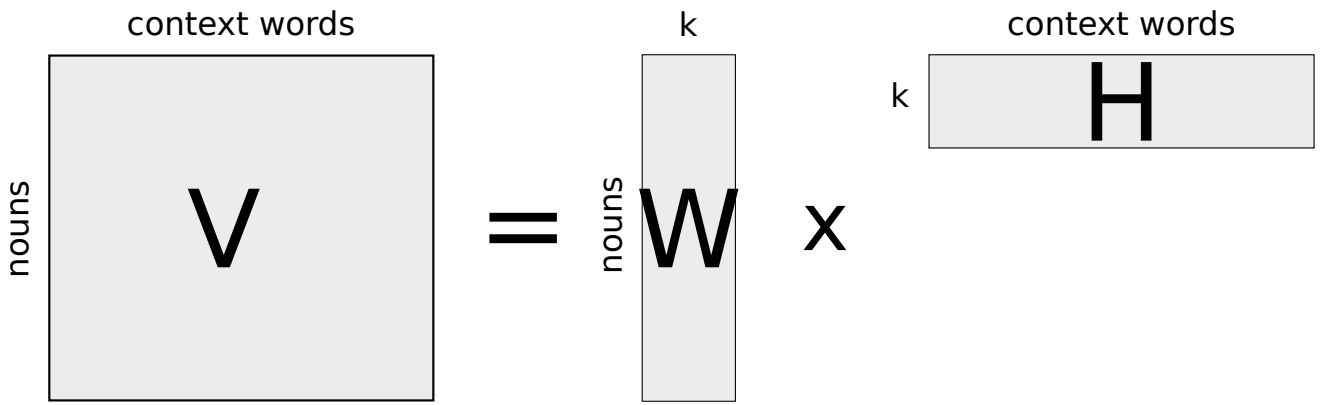
- captures co-occurrence frequencies of two entities

	red	tasty	fast	second-hand
raspberry	728	592	1	0
strawberry	1035	437	0	2
car	392	0	487	370
truck	104	0	393	293

Vector space model



Dimensionality reduction: latent semantics



Non-negative matrix factorization

- Given a non-negative matrix V, find non-negative matrix factors W and H such that:

$$\mathbf{V}_{n\times m} \approx \mathbf{W}_{n\times r}\mathbf{H}_{r\times m} \tag{1}$$

- Choosing $r \ll n, m$ reduces data
- Constraint on factorization: all values in three matrices need to be *non-negative values* (≥ 0)
- Constraint brings about a *parts-based* representation: only additive, no subtractive relations are allowed
- Particularly useful for finding topical, thematic information

Example dimensions (*k*=300)

dim 60	**dim 88**	**dim 89**	**dim 120**
rail	journal	filename	bathroom
bus	book	null	lounge
ferry	preface	integer	bedroom
train	anthology	string	kitchen
freight	author	parameter	WC
commuter	monograph	String	ensuite
tram	article	char	fireplace
airport	magazine	boolean	room
Heathrow	publisher	default	patio
Gatwick	pamphlet	int	dining

Compositionality

- Formal semantics
  - 'broad' compositionality at the level of the sentence
  - has little to say about 'co-composition': how does the lexical meaning of words change when they interact?
- $\leftrightarrow$ Distributional semantics
  - Good at capturing lexical semantics
  - Adequate models for computation of co-compositional meaning interaction
  - Representation of higher level representation (full sentence, quantification, negation, ...) is arguably more difficult
- Combine strengths of both approaches

Type Composition Logic (TCL)

- Detailed formal model of interaction between composition and lexical meaning
- Meaning of original words shifts in composition
- *heavy traffic* : $\lambda x.(\mathcal{O}(\text{heavy})(x) \wedge \mathcal{M}(\text{traffic})(x))$
- $\mathcal{O}$ and $\mathcal{M}$ are functors induced by the compositional counterpart
- adjective-noun combination decomposed into conjunction of two properties that represent the contextual contributions of noun and adjective
- But: no method for constructing functors or lexical meaning

TCL augmented with distributional semantics

- TCL distinguishes two types of semantic content:
  - internal/conceptual content
  - external/referential content
- distributional methods use vectors to represent word meaning
- represent TCL's internal content as distributional vectors
- TCL's functors correspond to vector transformations within a distributional model

Meaning shifts

- Meaning shifts occur when composition occurs
- Meaning shifting compositions: **co-composition**
- Ambiguity might be modeled as disjoint types
  - e.g. *traffic*: *internet traffic* vs. *rush hour traffic*
  - INFORMATION VS. VEHICLE
- What about *heavy* ?
  - *heavy appliance*
  - *heavy rain*
  - *heavy sea*
  - *heavy smoker*
- Impossible to define a disjunct type for each occurrence of *heavy*
- We need a flexible model of meaning shift

Adjective-noun composition schema

$$\lambda x\,(\mathcal{O}_A(N(x)) \wedge \mathcal{M}_N(A(x)))$$

- $\rightarrow$ meaning of both nouns and adjectives change according to the words they combine with
- How to define the functors $\mathcal{O}_A$ and $\mathcal{M}_N$?
- Nouns and adjectives are vectors in distributional space
- Functors are vector transformations influenced by co-occurring arguments

Distributional models for compositionality

- A number of distributional models of compositionality exist
  - additive model [Mitchell and Lapata 2008]
  - multiplicative model [Mitchell and Lapata 2008]
  - Lexical function model [Baroni and Zamparelli 2010]

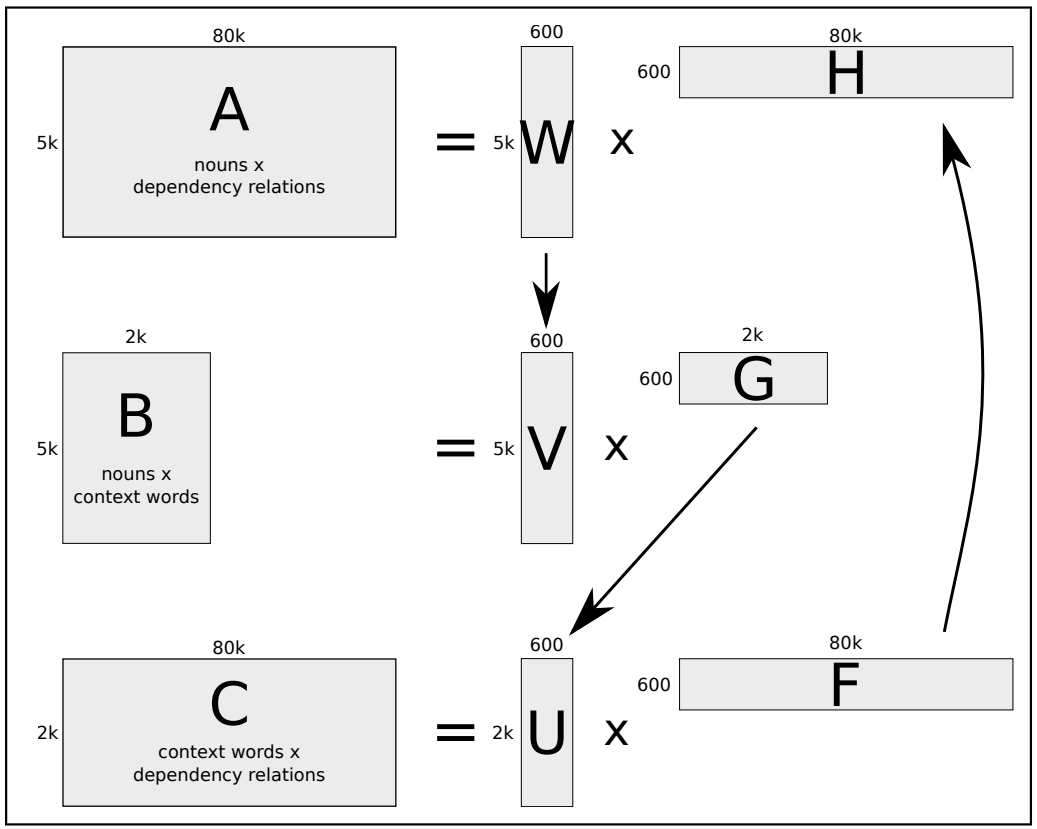
Two distributional models

- Two distributional models that are able to provide us with contextified vectors
  - LATENT VECTOR WEIGHTING: based on non-negative matrix factorization [Van de Cruys et al. 2011]
  - TENSOR-BASED WEIGHTING: based on non-negative tensor factorization

Latent vector weighting

- Can we combine 'topical' similarity and tight, synonym-like similarity to disambiguate meaning of word in a particular context?
- Goal: classification of nouns according to both window-based context (with large window) and syntactic context
- $\Rightarrow$ Construct three matrices capturing co-occurrence frequencies for each mode
  - nouns cross-classified by dependency relations
  - nouns cross-classified by (bag of words) context words
  - dependency relations cross-classified by context words
- $\Rightarrow$ Apply NMF to matrices, but interleave the process
- Result of former factorization is used to initialize factorization of the next one

Graphical representation



Example dimension 44

nouns	context words	dependency relations
building/NN	building/NN	dobj-1#redevelop/VB
factory/NN	construction/NN	conj_and/cc#warehouse/NN
center/NN	build/VB	prep_of/in-1#redevelopment/NN
refurbishment/NN	station/NN	prep_of/in-1#refurbishment/NN
warehouse/NN	store/NN	conj_and/cc#dock/NN
store/NN	open/VB	prep_by/in-1#open/VB
station/NN	center/NN	nn#refurbishment/NN
construction/NN	industrial/JJ	prep_of/in-1#ft/NN
complex/NN	Street/NNP	amod#multi-storey/JJ
headquarters/NN	close/VB	prep_of/in-1#opening/NN

Example dimension 89

words	context words	dependency relations
virus/NN	security/NN	amod#malicious/JJ
software/NN	Microsoft/NNP	nn-1#vulnerability/NN
security/NN	Internet/NNP	conj_and/cc#worm/NN
firewall/NN	Windows/NNP	nn-1#worm/NN
spam/NN	computer/NN	nn-1#flaw/NN
Security/NNP	network/NN	nn#antivirus/NN
vulnerability/NN	attack/NN	nn#IM/NNP
system/NN	software/NN	prep_of/in#worm/NN
Microsoft/NNP	protect/VB	nn#Trojan/NNP
computer/NN	protection/NN	conj_and/cc#virus/NN

Example dimension 319

words	context words	dependency relations
virus/NN	brain/NN	dobj-1#infect/VB
disease/NN	animal/NN	nsubjpass-1#infect/VB
bacterium/NN	disease/NN	rcmod#infect/VB
infection/NN	human/JJ	nsubj-1#infect/VB
human/NN	blood/NN	prep_with/in-1#infect/VB
rat/NN	cell/NN	conj_and/cc#rat/NN
cell/NN	cancer/NN	prep_of/in#virus/NN
animal/NN	skin/NN	amod#infected/JJ
mouse/NN	scientist/NN	prep_of/in#flu/NN
cancer/NN	drug/NN	nn#monkey/NN

Compute co-composition

- NMF can be interpreted probabilistically
- $p(\mathbf{z}|C)$ – the probability distribution over latent factors given the context ('semantic fingerprint')
- $p(\mathbf{d}|C) = p(\mathbf{z}|C)p(\mathbf{d}|\mathbf{z})$ – probability distribution over dependency features given the context
- $p(\mathbf{d}|w_i, C) = p(\mathbf{d}|w_i) \cdot p(\mathbf{d}|C)$ – weight each dependency feature of the original noun vector according to its prominence given the context
- Using the distribution over latent senses, it is possible to calculate the precise meaning of a word in context

Example

1. *explosive device*
   - $p(\textbf{topic}|\mathit{explosive}_A) \rightarrow p(\textbf{feature}|\mathit{device}_N, \mathit{explosive}_A)$
   - $\textbf{device}_N$: *device*, *ammunition*, *firearm*, *weapon*, *missile*
2. *electrical device*
   - $p(\textbf{topic}|\mathit{electrical}_A) \rightarrow p(\textbf{feature}|\mathit{device}_N, \mathit{electrical}_A)$
   - $\textbf{device}_N$: *device*, *equipment*, *sensor*, *system*, *technology*

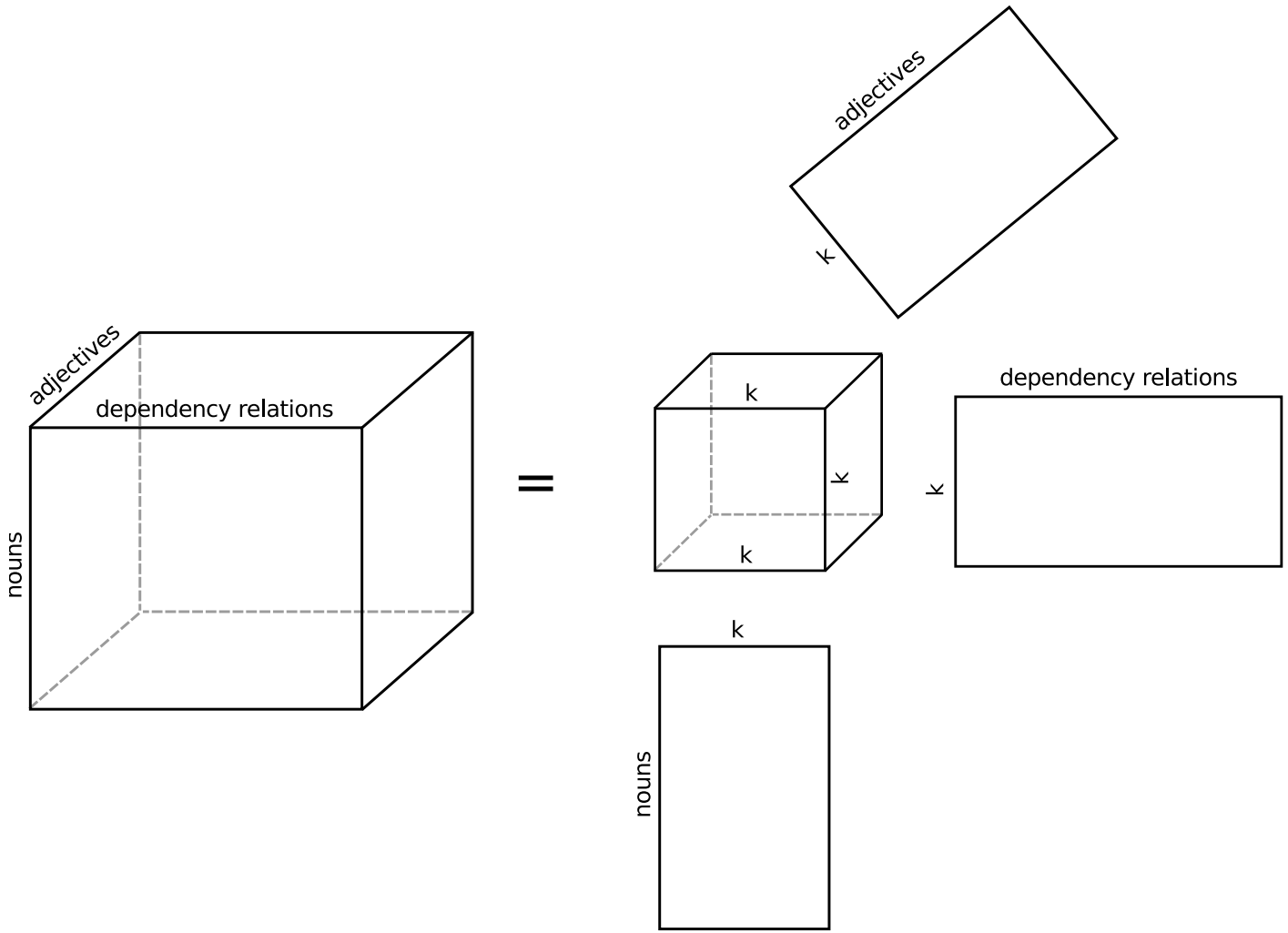
Tensor factorization

- Even richer and more flexible interaction between adjectives and nouns
- Factorize three-way tensor that contains multi-way co-occurrences of nouns, adjectives and various dependency relations
- Non-negative Tucker decomposition
- For a three-mode tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times L}$, the model is defined as

$$\begin{aligned} \mathcal{X} &= \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \\ &= \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r \end{aligned}$$

- computationally intensive, but efficient, sparse implementation

Graphical representation



Compute co-composition

- Multiply vector for noun ($\mathbf{a}_i$) and adjective ($\mathbf{b}_j$) into core tensor ($\mathcal{G}$) to determine latent factors of composition

$$\mathbf{h} = \mathcal{G} \times_1 \mathbf{a}_i \times_2 \mathbf{b}_j$$

- Multiply latent vector with transpose of $\mathbf{C}$ to compute vector $\mathbf{d}$, representing importance of each dependency feature given adjective-noun composition

$$\mathbf{d} = \mathbf{h}\mathbf{C}^T$$

- Last step: weight the original noun vector according to the importance of each dependency feature given the adjective-noun composition (pointwise multiplication)

$$\mathbf{v}'_d = \mathbf{d}_d . \mathbf{v}_d$$

Example

1. muddy bank
   - $\mathbf{h}_{muddy\_bank} = \mathcal{G} \times_1 \mathbf{a}_{bank} \times_2 \mathbf{b}_{muddy}$
   - $\mathbf{d}_{muddy\_bank} = \mathbf{h}_{muddy\_bank}\mathbf{C}^T$
   - $\mathbf{v}'_{bank} = \mathbf{d}_{muddy\_bank}.\mathbf{v}_{bank}$
   - $\mathbf{bank}_N$: *hillside*, *slope*, *ledge*, *cliff*, *ridge*

2. financial bank
   - $\mathbf{h}_{financial\_bank} = \mathcal{G} \times_1 \mathbf{a}_{bank} \times_2 \mathbf{b}_{financial}$
   - $\mathbf{d}_{financial\_bank} = \mathbf{h}_{financial\_bank}\mathbf{C}^T$
   - $\mathbf{v}'_{bank} = \mathbf{d}_{financial\_bank}.\mathbf{v}_{bank}$
   - $\mathbf{bank}_N$: *bank*, *broker*, *insurer*, *firm*, *banker*

Evaluation

- Test set of $\pm$ 250 adjective-noun combinations
- Compute top 10 most similar nouns to both adjective and noun in context, using 4 different models
  - unmodified: original adjective and noun vectors
  - lexical function model (LEXFUNC)
  - latent vector weighting (LVW)
  - tensor factorization (TENSOR)

Evaluation

- Evaluation of the models with regard to a number of formal semantic criteria:
  - **meaning shift**: does the approach predict a meaning shift?
  - **subsectivity and intersectivity**: for composition of A and N,
    - subsectivity: does composition predict individual noun meaning, $AN(x) \rightarrow N(x)$
    - intersectivity: does composition predict individual adjective meaning, $AN(x) \rightarrow A(x)$
  - **entailment**: for top most similar words Y, does $AN(x)$ defeasibly entail $Y(x)$
  - **semantic coherence**: for top most similar words, is $AN(x)$ semantically related to $Y(x)$; semantic relations included part-whole, subtype, typical localization, causal, semantic alternative, and antonym relations
- Note: first two criteria can be evaluated automatically, last two need to be evaluated manually (two annotators)

Meaning shift

- Cosine similarity as a proxy for shift
- LVW: on average, $sim_{cos}(\vec{v}_{orig}, \vec{v}_{mod}) = 0.3$ for adjectives, 0.5 for nouns
- TENSOR: on average, $sim_{cos}(\vec{v}_{orig}, \vec{v}_{mod}) = 0.2$ for adjectives, 0.5 for nouns
- large shift for adjectives, moderate shift for nouns

Example (LVW):

1. **heavy**$_A$: *heavy*$_A$ (1.000), *torrential*$_A$ (.149), *light*$_A$ (.140), *thick*$_A$ (.127), *massive*$_A$ (.118), *excessive*$_A$ (.115), *soft*$_A$ (.107), *large*$_A$ (.107), *huge*$_A$ (.104), *big*$_A$ (.103)
2. **heavy**$_A$, traffic$_N$: *heavy*$_A$ (.293), *motorised*$_A$ (.231), *vehicular*$_A$ (.229), *peak*$_A$ (.181), *one-way*$_A$ (.181), *horse-drawn*$_A$ (.175), *fast-moving*$_A$ (.164), *articulated*$_A$ (.158), *calming*$_A$ (.156), *horrendous*$_A$ (.146)

Subsectivity and intersectivity

method	subsectivity	intersectivity
UNMODIFIED	**1.00**	**1.00**
LEXFUNC	.58	–
LVW	.99	**1.00**
TENSOR	.97	.47

- Subsectivity clearly holds for LVW and TENSOR, less so for LEXFUNC
- Intersectivity mixed: LVW favours intersectivity, less so for TENSOR

Entailment

	nouns		adjectives	
method	**ent**$_1$	**ent**$_{10}$	**ent**$_1$	**ent**$_{10}$
UNMODIFIED	.22	.13	.18	.12
LEXFUNC	.42	.23	–	–
LVW	**.59**	**.32**	**.32**	**.25**
TENSOR	.42	.30	.16	.13

- Cosine, as expected, not the best measure for entailment

Semantic coherence

	nouns				adjectives			
method	**sr**$_1$	**sr**$_{10}$	**ent+sr**$_1$	**ent+sr**$_{10}$	**sr**$_1$	**sr**$_{10}$	**ent+sr**$_1$	**ent+sr**$_{10}$
UNMODIFIED	.37	.20	.59	.33	.14	.09	.32	.21
LEXFUNC	.19	.15	.61	.38	–	–	–	–
LVW	.27	.20	.86	.52	.35	.24	**.67**	**.49**
TENSOR	**.52**	**.43**	**.94**	**.73**	**.38**	**.30**	.53	.43

- cosine provides combination of entailments and semantically related words
- TENSOR method scored better on semantic coherence for nouns

Integration

- Distributional methods (LVW and TENSOR) as algebraic counterpart of TCL's functors
- Each word is represented in a vector space of its syntactic/semantic contexts
- Co-occurring argument determines distribution over latent topics
- Used to shift meaning of original word according to its argument
- Co-composition may be modeled as vector transformations within distributional space
- To reinject information into the symbolic system, we need better detectors for semantic relations

Discussion

- Why mix statistical and logical information in one system?
- Can't we just do everything with statistics (algebraic methods, neural nets, ...)?
- Formal semantics has some attractive features
  - use of variables
  - discourse referents
  - scope-bearing operators
- Less straightforward with alternative methods
- Formal semantics explicitly linked to a theory of truth

Conclusion

- First step towards integration of formal and distributional semantics
- Internal content of words represent by distributional vectors
- Co-composition functors as transformations acting on vectors

Further work

- Investigation of verb-argument co-compositions
- Further integration of distributional methods with TCL's functor approach
- Make use of distributional information to inform type-theoretic content and construct logical form


Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA.


B. Coecke, M. Sadrzadeh, and S. Clark. 2010. Mathematical foundations for a compositional distributed model of meaning. *Lambek Festschrift, Linguistic Analysis, vol. 36*, 36.


Zellig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.


Daniel D. Lee and H. Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562.


Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. *proceedings of ACL-08: HLT*, pages 236–244.


Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2011. Latent vector weighting for word meaning in context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1012–1022, Edinburgh, Scotland, UK.


Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2013. A tensor-based factorization model of semantic compositionality. In *Conference of the North American Chapter of the Association of Computational Linguistics (HTL-NAACL)*, pages 1142–1151.