EQUIPE MELODI

Distributional Semantics

The unsupervised modeling of meaning on a large scale

Tim Van de Cruys
IRIT, Toulouse

Tuesday 17 November 2015

Distributional similarity

- 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 similarity

- 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 similarity

- 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

Matrix

- 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

Matrix

- 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

Matrix

- captures co-occurrence frequencies of two entities

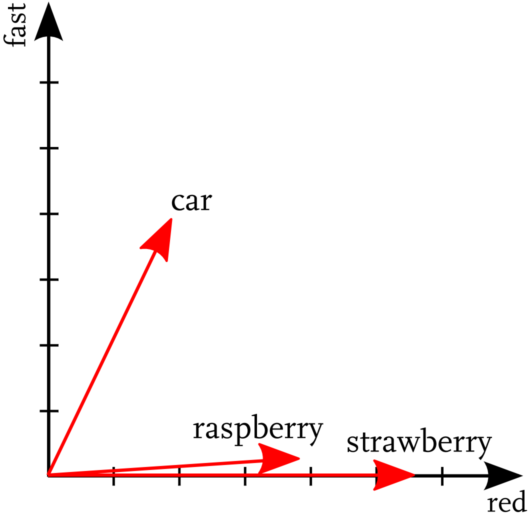
	red	tasty	fast	second-hand
raspberry	56	98	0	0
strawberry	44	34	0	0
car	23	0	31	39
truck	4	0	18	29

Matrix

- 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



Word-context matrix

	context1	context2	context3	context4
word1				
word2				
word3				
word4				

- Different notions of context
  - window around word
  - dependency-based features (extracted from parse trees)

He drove his second-hand **car** a couple of miles down the road .

Word-context matrix

	context1	context2	context3	context4
word1				
word2				
word3				
word4				

- Different notions of context
  - window around word (2 words)
  - dependency-based features (extracted from parse trees)

He drove [his second-hand **car** a couple] of miles down the road .

Word-context matrix

	context1	context2	context3	context4
word1				
word2				
word3				
word4				

- Different notions of context
  - window around word (sentence)
  - dependency-based features (extracted from parse trees)

[He drove his second-hand **car** a couple of miles down the road .]

Word-context matrix

	context1	context2	context3	context4
word1				
word2				
word3				
word4				

- Different notions of context
  - window around word
  - dependency-based features (extracted from parse trees)

obj

mod

He drove his second-hand **car** a couple of miles down the road .

Different kinds of semantic similarity

- **'tight', synonym-like similarity**: (near-)synonymous or (co-)hyponymous
- **loosely related, topical similarity**: more loose relationships, such as association and meronymy

Different kinds of semantic similarity

- **‘tight’, synonym-like similarity**: (near-)synonymous or (co-)hyponymous
- **loosely related, topical similarity**: more loose relationships, such as association and meronymy

Example

- **doctor**: *nurse*, *GP*, *physician*, *practitioner*, *midwife*, *dentist*, *surgeon*
- **doctor**: *medication*, *disease*, *surgery*, *hospital*, *patient*, *clinic*, *nurse*, *treatment*, *illness*

Relation context – similarity

- Different context leads to different kind of similarity
- Syntax, small window $\leftrightarrow$ large window, documents
- The former models induce **tight, synonymous similarity**
- The latter models induce **topical relatedness**

Computing similarity ...

- Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
- blackberry, blackcurrant, blueberry, raspberry, redcurrant, strawberry
- anthropologist, biologist, economist, linguist, mathematician, psychologist, physicist, sociologist, statistician
- drought, earthquake, famine, flood, flooding, storm, tsunami

...on a large scale

- Frequency matrices are extracted from very large corpora
- Large collections of newspapers, Wikipedia, documents crawled from the web, ...
- $>$ 100 billion words
- Large demands with regard to computing power and memory
- Matrices are very sparse $\rightarrow$ use of algorithms and storage formats that take advantage of the sparseness

...on a large scale

- Take advantage of parallel computations
- Many algorithms can be implemented within a map-reduce framework
  - Collection of frequency matrices
  - Matrix transformations
  - Syntactic parsing
- Make use of IRIT's high performance computing cluster OSIRIM (10 nodes, 640 cores in total)
- Huge speedup

Dimensionality reduction

Two reasons for performing dimensionality reduction:

- Intractable computations
  - When number of elements and number of features is too large, similarity computations may become intractable
  - reduction of the number of features makes computation tractable again
- Generalization capacity
  - the dimensionality reduction is able to describe the data better, or is able to capture intrinsic semantic features
  - dimensionality reduction is able to improve the results (counter data sparseness and noise)

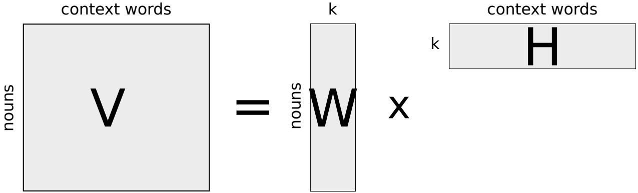
Non-negative matrix factorization

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

$$\mathbf{V}_{n\times m} \approx \mathbf{W}_{n\times r}\mathbf{H}_{r\times m} \quad (1)$$

- 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

Graphical Representation



Example dimensions

dim 9	**dim 12**	**dim 21**	**dim 24**
infection	fichiers	agneau	professeurs
respiratoire	windows	desserts	cursus
respiratoires	messagerie	miel	enseignants
maladies	téléchargement	boeuf	pédagogique
nerveux	serveur	veau	enseignant
artérielle	logiciel	pomme	universitaires
tumeurs	connexion	saumon	scolarité
lésions	via	canard	étudiants
cardiaque	internet	poire	étudiant
métabolisme	html	fumé	formateurs

Word meaning in context

- Standard word space models are good at capturing general, 'global' word meaning
  $\leftrightarrow$ Words have different senses
  $\leftrightarrow$ Meaning of individual word instances differs significantly
- Context is determining factor for construction of individual word meaning

(1) Jack is listening to a **record**
(2) Jill updated the **record**

Word meaning in context

- Standard word space models are good at capturing general, 'global' word meaning
  $\leftrightarrow$ Words have different senses
  $\leftrightarrow$ Meaning of individual word instances differs significantly

- Context is determining factor for construction of individual word meaning

  (1) Jack is listening to a **record**
  (2) Jill updated the **record**

Word meaning in context

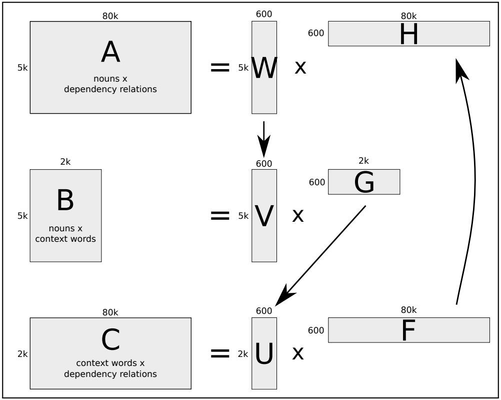
- Standard word space models are good at capturing general, 'global' word meaning
  $\leftrightarrow$ Words have different senses
  $\leftrightarrow$ Meaning of individual word instances differs significantly
- Context is determining factor for construction of individual word meaning

(1) Jack is listening to a **record**
(2) Jill updated the **record**

Word meaning in context

- 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

Calculating word meaning in context

- NMF can be interpreted probabilistically
- $p(\mathbf{z}|C) = \frac{\sum_{c_i \in C} p(\mathbf{z}|c_i)}{|C|}$ – 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. Jack is listening to a **record**.
   - $p(\textbf{topic}|listen_{pc(to)}) \rightarrow p(\textbf{feature}|record_N, listen_{pc(to)})$
   - **record**$_N$: *album*, *song*, *recording*, *track*, *cd*

2. Jill updated the **record**.
   - $p(\textbf{topic}|update_{obj}) \rightarrow p(\textbf{feature}|record_N, update_{obj})$
   - **record**$_N$: *file*, *data*, *document*, *database*, *list*

Evaluation

- Evaluated using an established lexical substitution task
- find appropriate substitutes in context
- Model performs significantly better than competing models
- Moreover, model performs well at paraphrase induction (inducing lexical substitutes from scratch) whereas other models only perform paraphrase ranking (rank a limited set of candidate substitutes)

Compositionality within a distributional model

- principle of semantic **compositionality** [Frege 1892]

meaning of a complex expression	=	meaning of its parts	+	the way those parts are combined

- fundamental principle that allows people to understand sentences they have never heard before

Compositionality within a distributional model

- principle of semantic **compositionality** [Frege 1892]

meaning of a complex expression	=	meaning of its parts	+	the way those parts are combined

standard distributional model

- fundamental principle that allows people to understand sentences they have never heard before

Compositionality within a distributional model

- principle of semantic **compositionality** [Frege 1892]

meaning of a complex expression	=	meaning of its parts	+	the way those parts are combined
		standard distributional model		tensor-based factorization model

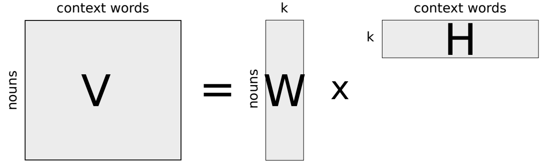
- fundamental principle that allows people to understand sentences they have never heard before

Compositionality within a distributional model

- model for joint composition of verb with subject and direct object
- allows us to compute semantic similarity between simple transitive sentences
- key idea: compositionality is modeled as a multi-way interaction between latent factors, automatically constructed from corpora
- implemented using tensor algebra

Step 1: construction of latent noun factors

- Construction of a latent model for nouns using non-negative matrix factorization

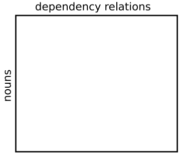


Step 1: example noun factors (*k*=300)

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

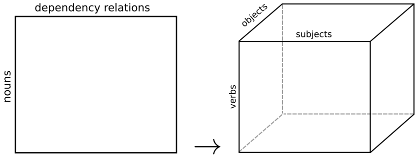
Step 2: Modeling multi-way interactions

- Standard distributional similarity methods model two-way interactions $\rightarrow$ matrix
  - words $\times$ context words
  - words $\times$ dependency relations
- not suitable for multi-way interactions
  - nouns $\times$ adjectives $\times$ context words
  - **verbs** $\times$ **subjects** $\times$ **objects**



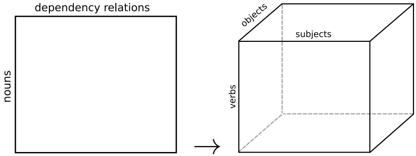
Step 2: Modeling multi-way interactions

- Standard distributional similarity methods model two-way interactions $\rightarrow$ matrix
  - words $\times$ context words
  - words $\times$ dependency relations
- not suitable for multi-way interactions $\rightarrow$ tensor
  - nouns $\times$ adjectives $\times$ context words
  - **verbs** $\times$ **subjects** $\times$ **objects**



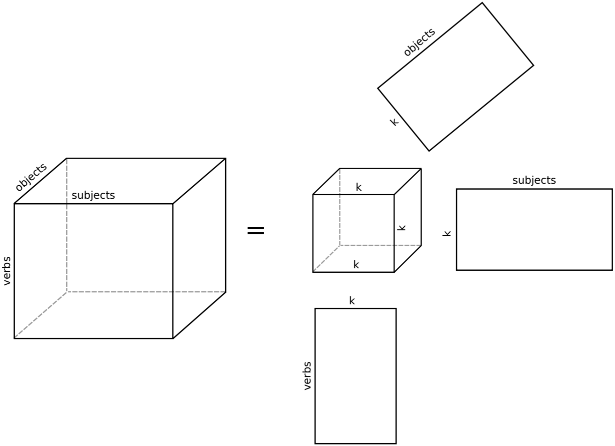
Step 2: Modeling multi-way interactions

- Standard distributional similarity methods model two-way interactions $\rightarrow$ matrix
  - words $\times$ context words
  - words $\times$ dependency relations
- not suitable for multi-way interactions $\rightarrow$ tensor
  - nouns $\times$ adjectives $\times$ context words
  - **verbs** $\times$ **subjects** $\times$ **objects**



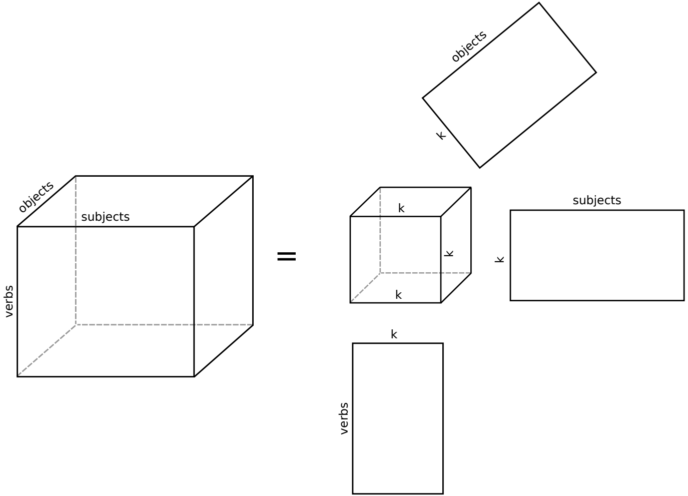
$\rightarrow$ build a latent model of multi-way interactions

Step 2: Non-negative Tucker decomposition

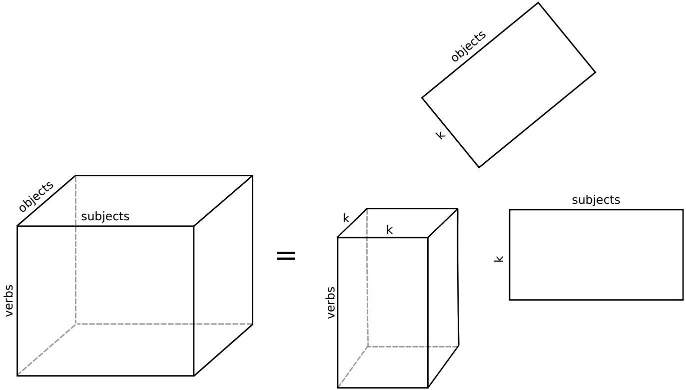


$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

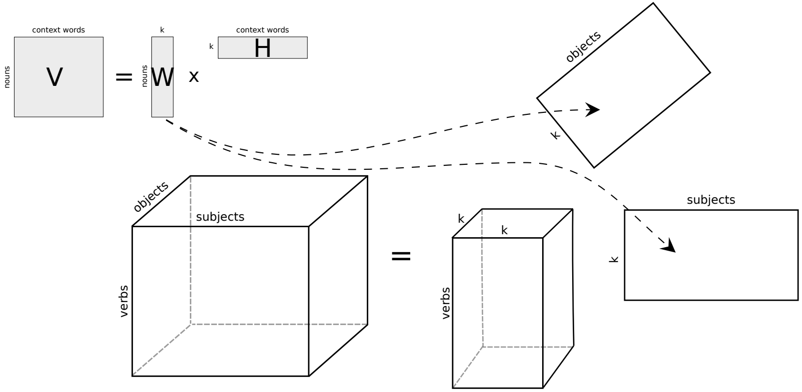
Step 2: Reconstructing a Tucker model from two-way factors



Step 2: Reconstructing a Tucker model from two-way factors

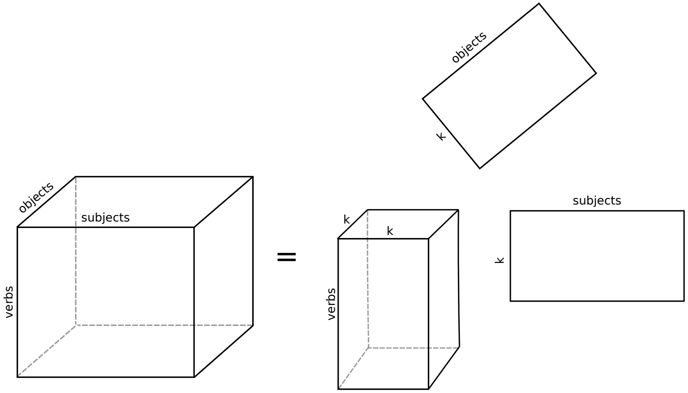


Step 2: Reconstructing a Tucker model from two-way factors

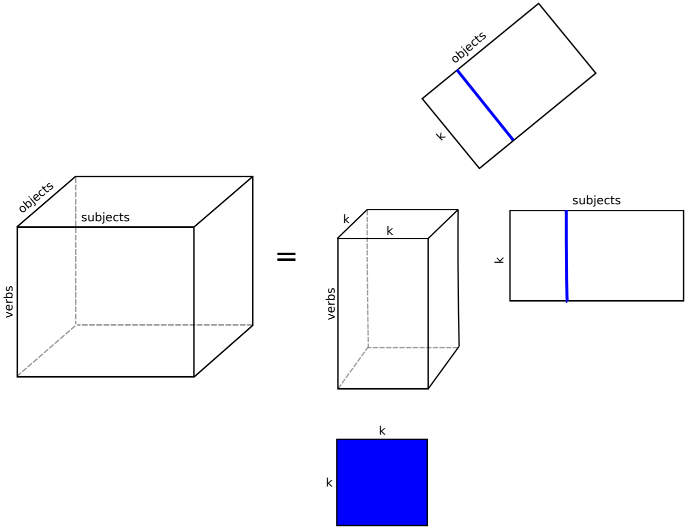


$$\mathcal{G} = \boldsymbol{\mathcal{X}} \times_2 \mathbf{W}^T \times_3 \mathbf{W}^T$$

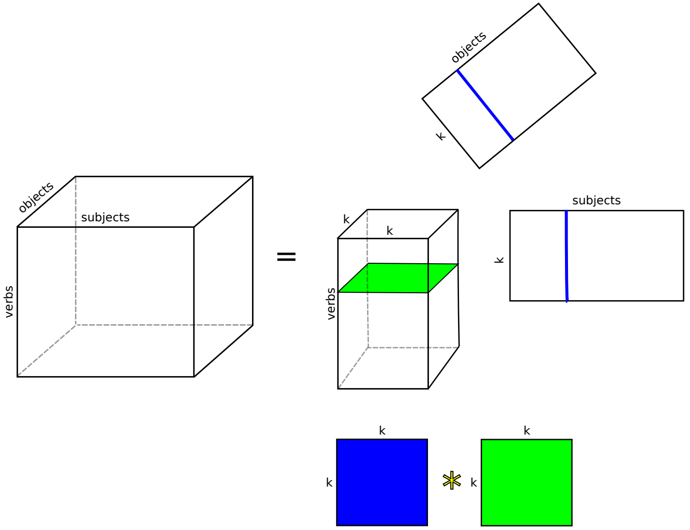
Step 3: composition of *svo* triples



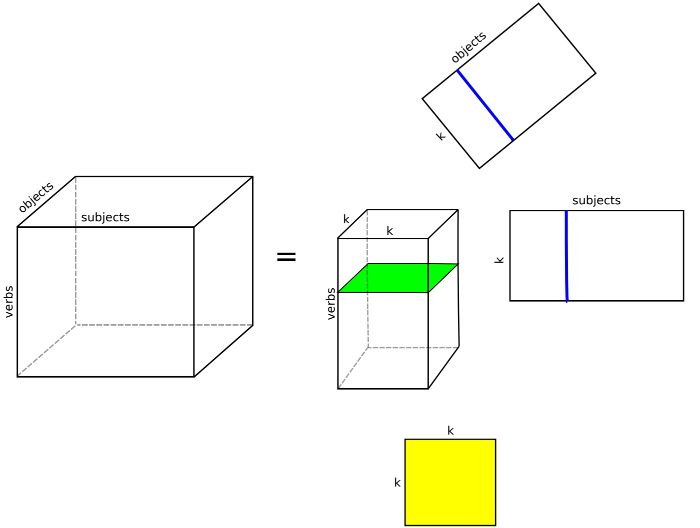
Step 3: composition of *svo* triples



Step 3: composition of *svo* triples



Step 3: composition of *svo* triples



Example

- *athlete runs race*
  - $\mathbf{Y}_{\langle athlete,race \rangle} = \mathbf{v}_{athlete} \circ \mathbf{u}_{race}$
  - $\mathbf{Z}_{run,\langle athlete,race \rangle} = \mathbf{G}_{run} * \mathbf{Y}_{\langle athlete,race \rangle}$
- *user runs command*
  - $\mathbf{Y}_{\langle user,command \rangle} = \mathbf{v}_{user} \circ \mathbf{u}_{command}$
  - $\mathbf{Z}_{run,\langle user,command \rangle} = \mathbf{G}_{run} * \mathbf{Y}_{\langle user,command \rangle}$

Example

- $\mathbf{Y}_{\langle athlete,race \rangle} = \mathbf{v}_{athlete} \circ \mathbf{u}_{race}$

top factors	top words on factor
195	*people*, *child*, *adolescent*
119	*cup*, *championship*, *final*
25	*hockey*, *poker*, *tennis*
119	*cup*, *championship*, *final*
90	*professionalism*, *teamwork*, *confidence*
119	*cup*, *championship*, *final*
28	*they*, *pupil*, *participant*
119	*cup*, *championship*, *final*

Example

- $\mathbf{Y}_{\langle athlete, race \rangle} = \mathbf{v}_{athlete} \circ \mathbf{u}_{race}$

top factors	top words on factor
195	**people**, **child**, **adolescent**
119	*cup*, *championship*, *final*
25	*hockey*, *poker*, *tennis*
119	*cup*, *championship*, *final*
90	*professionalism*, *teamwork*, *confidence*
119	*cup*, *championship*, *final*
28	**they**, **pupil**, **participant**
119	*cup*, *championship*, *final*

Example

- $\mathbf{Y}_{\langle athlete,race\rangle} = \mathbf{v}_{athlete} \circ \mathbf{u}_{race}$

top factors	top words on factor
195	*people*, *child*, *adolescent*
119	*cup*, *championship*, *final*
25	**hockey**, **poker**, **tennis**
119	*cup*, *championship*, *final*
90	**professionalism**, **teamwork**, **confidence**
119	*cup*, *championship*, *final*
28	*they*, *pupil*, *participant*
119	*cup*, *championship*, *final*

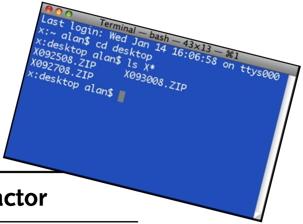
Example

- $\mathbf{Y}_{\langle athlete,race \rangle} = \mathbf{v}_{athlete} \circ \mathbf{u}_{race}$

top factors	top words on factor
195	*people*, *child*, *adolescent*
119	**cup**, **championship**, **final**
25	*hockey*, *poker*, *tennis*
119	**cup**, **championship**, **final**
90	*professionalism*, *teamwork*, *confidence*
119	**cup**, **championship**, **final**
28	*they*, *pupil*, *participant*
119	**cup**, **championship**, **final**

Example

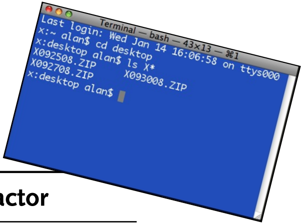
- $\mathbf{Y}_{\langle user, command \rangle} = \mathbf{v}_{user} \circ \mathbf{u}_{command}$



top factors	top words on factor
7	*password, login, username*
89	*filename, null, integer*
40	*anyone, reader, anybody*
89	*filename, null, integer*
195	*people, child, adolescent*
89	*filename, null, integer*
45	*website, Click, site*
89	*filename, null, integer*

Example

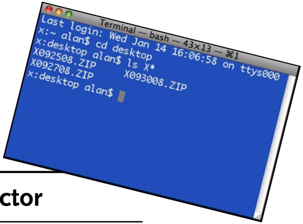
- $\mathbf{Y}_{\langle user,command \rangle} = \mathbf{v}_{user} \circ \mathbf{u}_{command}$



top factors	top words on factor
7	*password*, *login*, *username*
89	*filename*, *null*, *integer*
40	**anyone**, **reader**, **anybody**
89	*filename*, *null*, *integer*
195	**people**, **child**, **adolescent**
89	*filename*, *null*, *integer*
45	*website*, *Click*, *site*
89	*filename*, *null*, *integer*

Example

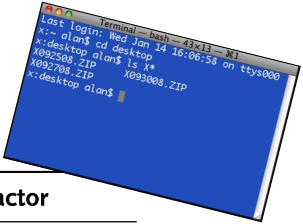
- $\mathbf{Y}_{\langle user,command \rangle} = \mathbf{v}_{user} \circ \mathbf{u}_{command}$



top factors	top words on factor
7	**password**, **login**, **username**
89	*filename*, *null*, *integer*
40	*anyone*, *reader*, *anybody*
89	*filename*, *null*, *integer*
195	*people*, *child*, *adolescent*
89	*filename*, *null*, *integer*
45	**website**, **Click**, **site**
89	*filename*, *null*, *integer*

Example

- $\mathbf{Y}_{\langle user, command \rangle} = \mathbf{v}_{user} \circ \mathbf{u}_{command}$



top factors	top words on factor
7	*password*, *login*, *username*
89	**filename**, **null**, **integer**
40	*anyone*, *reader*, *anybody*
89	**filename**, **null**, **integer**
195	*people*, *child*, *adolescent*
89	**filename**, **null**, **integer**
45	*website*, *Click*, *site*
89	**filename**, **null**, **integer**

Example

- $\mathbf{G}_{run}$

top factors	top words on factor
128	*Mathematics*, *Science*, *Economics*
181	*course*, *tutorial*, *seminar*
293	*organization*, *association*, *federation*
181	*course*, *tutorial*, *seminar*
60	*rail*, *bus*, *ferry*
140	*third*, *decade*, *hour*
268	*API*, *Apache*, *Unix*
268	*API*, *Apache*, *Unix*

Example

- $\mathbf{G}_{run}$

top factors	top words on factor
128	**Mathematics, Science, Economics**
181	**course, tutorial, seminar**
293	**organization, association, federation**
181	**course, tutorial, seminar**
60	*rail, bus, ferry*
140	*third, decade, hour*
268	*API, Apache, Unix*
268	*API, Apache, Unix*

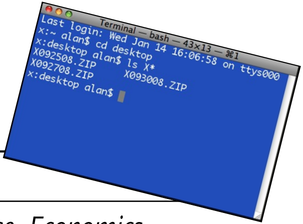
Example

- $\mathbf{G}_{run}$

top factors	top words on factor
128	*Mathematics, Science, Economics*
181	*course, tutorial, seminar*
293	*organization, association, federation*
181	*course, tutorial, seminar*
60	**rail, bus, ferry**
140	**third, decade, hour**
268	*API, Apache, Unix*
268	*API, Apache, Unix*

Example

- $\mathbf{G}_{run}$



top factors	top words on factor
128	*Mathematics*, *Science*, *Economics*
181	*course*, *tutorial*, *seminar*
293	*organization*, *association*, *federation*
181	*course*, *tutorial*, *seminar*
60	*rail*, *bus*, *ferry*
140	*third*, *decade*, *hour*
268	**API**, **Apache**, **Unix**
268	**API**, **Apache**, **Unix**

Example

- *athlete runs race*
  - $\mathbf{Y}_{\langle athlete,race \rangle} = \mathbf{v}_{athlete} \circ \mathbf{u}_{race}$
  - $\mathbf{Z}_{run,\langle athlete,race \rangle} = \mathbf{G}_{run} * \mathbf{Y}_{\langle athlete,race \rangle}$
  - **finish** (.29), **attend** (.27), **win** (.25)
- *user runs command*
  - $\mathbf{Y}_{\langle user,command \rangle} = \mathbf{v}_{user} \circ \mathbf{u}_{command}$
  - $\mathbf{Z}_{run,\langle user,command \rangle} = \mathbf{G}_{run} * \mathbf{Y}_{\langle user,command \rangle}$
  - **execute** (.42), **modify** (.40), **invoke** (.39)
- *man damages car*
  - **crash** (.43), **drive** (.35), **ride** (.35)
- *car damages man*
  - **scare** (.26), **kill** (.23), **hurt** (.23)

Evaluation

- sentence similarity task for transitive sentences
- compute correlation of model's judgements with human judgements

p	target	subject	object	landmark	sim
19	meet	system	criterion	visit	1
21	write	student	name	spell	6

- Model achieves a significant improvement compared to related models

Selectional preference induction

- Predicates often have a semantically motivated preference for particular arguments

(1) The vocalist sings a ballad.

(2) *The exception sings a tomato.

$\rightarrow$ known as *selectional preferences*

Selectional preference induction

- majority of language utterances occur very infrequently
- models of selectional preference need to properly generalize
- Earlier approaches:
  - hand-crafted resources (WordNet)
  - latent variable models
  - distributional similarity metrics
- this research: *neural network model*

Model overview

- Inspired by recent advances of neural network models for NLP applications [Collobert and Weston 2008]
- Train a neural network to discriminate between felicitous and infelicitous arguments for a particular predicate
- Entirely unsupervised: preferences are learned from corpus data
    - positive instances constructed from attested corpus data
    - negative instances constructed from randomly corrupted data
- two network architectures: for both two-way and multi-way preferences

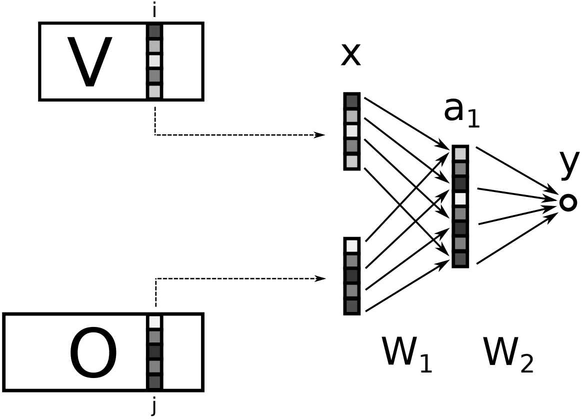
Neural network architecture

- feed-forward neural network architecture with one hidden layer
- tuple (i, j) is represented as concatenation of vectors $\mathbf{v}_i$ and $\mathbf{o}_j$, extracted from embedding matrices $\mathbf{V}$ and $\mathbf{O}$ (learned during training)
- Vector $\mathbf{x}$ then serves as input vector to our neural network.

$$
\begin{aligned}
\mathbf{x} &= [\mathbf{v}_i, \mathbf{o}_j] \\
\mathbf{a}_1 &= f(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \\
y &= \mathbf{W}_2 \mathbf{a}_1
\end{aligned}
$$

- $\mathbf{a}_1$: activation of hidden layer
- $\mathbf{W}_1$ and $\mathbf{W}_2$: first and second layer weights
- $\mathbf{b}_1$: first layer's bias
- $f(\cdot)$: element-wise activation function tanh

Graphical representation



Training

- Proper estimation of neural network's parameters requires large amount of training data
- Create unsupervised training data by corrupting actual attested tuples
- Cost function that learns to discriminate between good and bad examples (margin of at least one)

$$\sum_{j' \in J} \max(0, 1 - g[(i,j)] + g[(i,j')])$$

- Compute derivative of the cost with respect to the model's parameters
- Update parameters through backpropagation

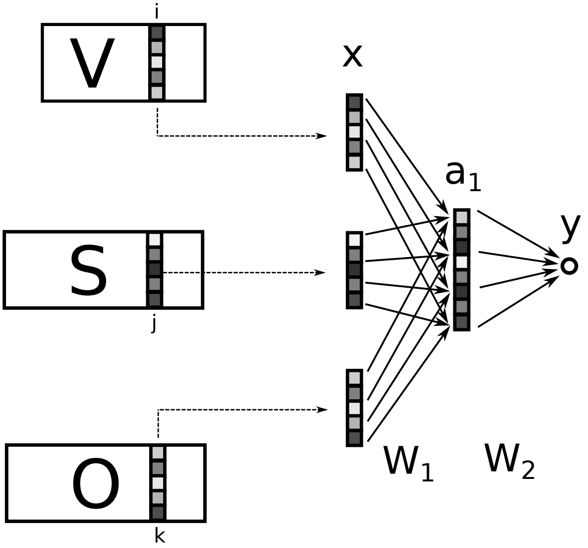
Multi-way selectional preferences

- Similar to two-way case, but one extra embedding matrix for each extra argument
- E.g., for *subject-verb-object* tuples, input vector is

$$\mathbf{x} = (\mathbf{v}_i, \mathbf{s}_j, \mathbf{o}_k)$$

- Rest of the network architecture stays the same

Graphical representation



Training

- Adapted version of training objective
- Given attested tuple (i, j, k), discriminate the correct tuple from corrupted tuples (i, j, k'), (i, j', k), (i, j', k')

$$
\begin{aligned}
&\sum_{k' \in K} \max(0, 1 - g[(i, j, k)] + g[(i, j, k')]) \\
+ &\sum_{j' \in J} \max(0, 1 - g[(i, j, k)] + g[(i, j', k)]) \\
+ &\sum_{\substack{j' \in J \\ k' \in K}} \max(0, 1 - g[(i, j, k)] + g[(i, j', k')])
\end{aligned}
$$

Evaluation

- pseudo-disambiguation task to test generalization capacity (standard automatic evaluation for selectional preferences)

v	s	o	s'	o'
win	*team*	*game*	*diversity*	*egg*
publish	*government*	*document*	*grid*	*priest*
develop	*company*	*software*	*breakfast*	*landlord*

- state-of-the art results

Examples

DRINK	PROGRAM	INTERVIEW	FLOOD
SIP	RECOMPILE	RECRUIT	INUNDATE
BREW	UNDELETE	PERSUADE	RAVAGE
MINCE	CODE	INSTRUCT	SUBMERGE
FRY	IMPORT	PESTER	COLONIZE

Examples

PAPER	RASPBERRY	SECRETARY	DESIGNER
BOOK	COURGETTE	PRESIDENT	PLANNER
JOURNAL	LATTE	MANAGER	PAINTER
ARTICLE	LEMONADE	POLICE	SPECIALIST
CODE	OATMEAL	EDITOR	SPEAKER

Examples

WALL	PARK	LUNCH	THESIS
FLOOR	STUDIO	DINNER	QUESTIONNAIRE
CEILING	VILLAGE	MEAL	DISSERTATION
ROOF	HALL	BUFFET	PERIODICAL
METRE	MUSEUM	BREAKFAST	DISCOURSE

Examples

- Separate word representations for subject and object position
- Allows model to capture specific characteristics for words given their argument position
  - *virus*
    - subject slot: similar to active words like *animal*
    - object slot: similar to passive words like *cell, device*
  - *mouse*
    - subject slot: similar to *animal, rat*
    - object slot: similar to *web, browser*

Conclusion

- By using text corpora on a large scale, we are able to efficiently model meaning
- Global word meaning can be computed by accumulating word context vectors
- Individual word meaning can be modeled by adapting the word's original feature vector based on the latent dimensions determined by the context
- compositionality can be modeled as a multi-way interaction between latent factors, using tensor algebra
- Machine learning algorithms (neural networks) are helpful for capturing semantic phenomena

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Gottlob Frege. 1892. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50.

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

Tim Van de Cruys. 2009. A non-negative tensor factorization model for selectional preference induction. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 83–90, Athens, Greece, March. Association for Computational Linguistics.

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.

Tim Van de Cruys. 2014. A neural network approach to selectional preference acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 26–35.

Lexical substitution: Evaluation

- Evaluated with SEMEVAL 2007 lexical substitution task
- find appropriate substitutes in context
- 200 target words (50 for each pos), 10 sentences each
- Paraphrase **ranking**: rank possible candidates, standard evaluation for unsupervised methods
  - Kendall's τ_b ranking coefficient
  - Generalized average precision
- Paraphrase **induction**: find candidates from scratch, not carried out before for unsupervised methods
  - Recall
  - Precision out-of-ten

Lexical substitution: Paraphrase ranking

model	τ_b	GAP
random	-0.61	29.98
vector_{dep}	16.57	45.08
EP09	–	32.2 ▼
EP10	–	39.9 ▼
TFP	–	45.94▼
DL	16.56	41.68
$\text{NMF}_{context}$	20.64**	47.60**
NMF_{dep}	22.49**	48.97**
NMF_{c+d}	**22.59****	**49.02****

Lexical substitution: Paraphrase induction

model	R_{best}	P_{10}
vector$_{dep}$	8.78	30.21
DL	1.06	7.59
NMF$_{context}$	8.81	**30.49**
NMF$_{dep}$	7.73	26.92
NMF$_{c+d}$	**8.96**	29.26

Compositionality Evaluation: results

model	contextualized	non-contextualized
baseline	.23	
multiplicative	**.32**	.34
categorical	**.32**	.35
latent	**.32**	**.37**
upper bound	.62	

Results: two-way selectional preference induction

model	accuracy ($\mu \pm \sigma$)
[Rooth et al. 2009]	$.720 \pm .002$
[Erk et al. 2010]	$\mathbf{.887} \pm .004$
2-way neural network	$.880 \pm .001$

- Slightly better result of model based on distributional similarity
- But: Erk et al.'s model is very slow, neural network model is very fast

Results: three-way selectional preference induction

model	accuracy ($\mu \pm \sigma$)
[Van de Cruys 2009]	$.868 \pm .001$
3-way neural network	**.889** $\pm .001$

- Neural network approach reaches state-of-the-art results for multi-way selectional preference induction