Language in 3D

Multi-way factorization algorithms to tackle semantics

Tim Van de Cruys

University of Cambridge

NLIP seminar
Friday 3 June, 2011

Distributional similarity

Distributional similarity models are able to infer (lexical) semantics from text:

- semantically similar words (syntactic context, small window)
  - **train**: *bus*, *ferry*, *boat*, *coach*, *car*, *plane*, *vehicle*, *taxi*, *ship*, *truck*, ...
  - **doctor**: *nurse*, *GP*, *physician*, *practitioner*, *midwife*, *dentist*, *surgeon*, ...

Distributional similarity

Distributional similarity models are able to infer (lexical) semantics from text:

- topically related words (large window)
  - **train**: *bus*, *journey*, *railway*, *station*, *passenger*, *ride*, *stop*, *taxi*, *fare*, ...
  - **doctor**: *medication*, *GP*, *surgery*, *hospital*, *sufferer*, *clinic*, *nurse*, *treatment*, *illness*, ...

Two-way vs. three-way

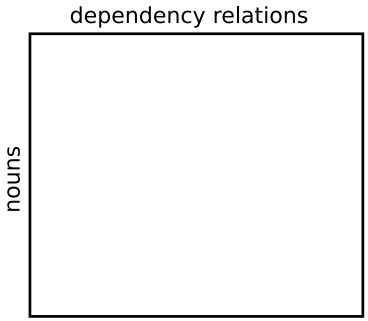
- all methods use two way co-occurrence frequencies $\longrightarrow$ matrix
- suitable for two-way problems
  - words $\times$ documents
  - nouns $\times$ dependency relations
- not suitable for *n*-way problems
  - words $\times$ documents $\times$ authors
  - verbs $\times$ subjects $\times$ direct objects

Two-way vs. three-way

- all methods use two way co-occurrence frequencies $\longrightarrow$ matrix
- suitable for two-way problems
  - words $\times$ documents
  - nouns $\times$ dependency relations
- not suitable for *n*-way problems $\longrightarrow$?
  - words $\times$ documents $\times$ authors
  - verbs $\times$ subjects $\times$ direct objects

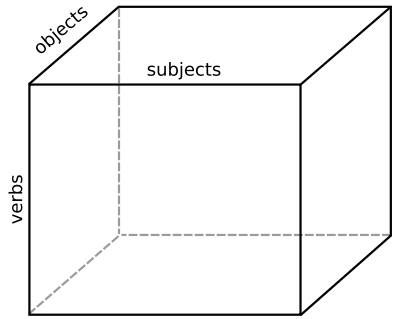
Two-way vs. three-way

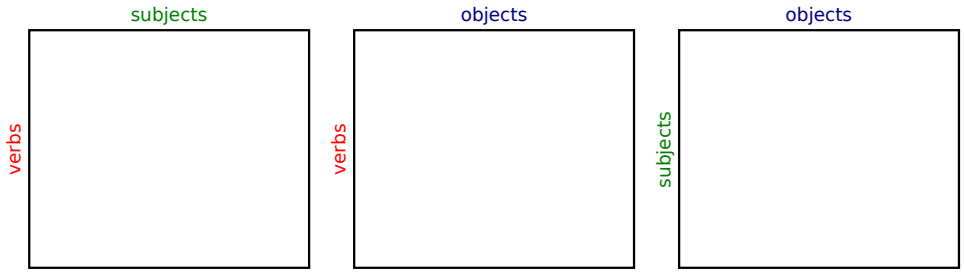
two-way



matrix

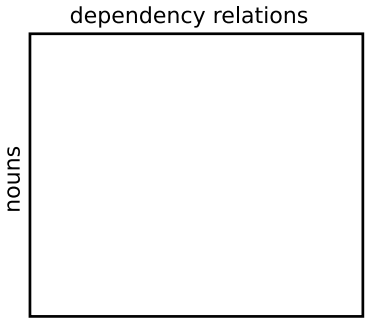
three-way





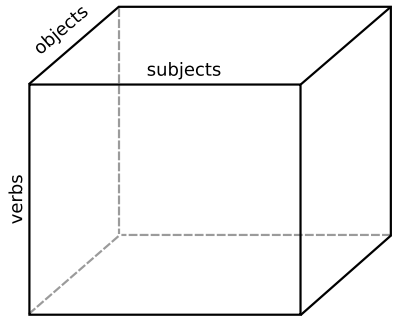
Two-way vs. three-way

two-way

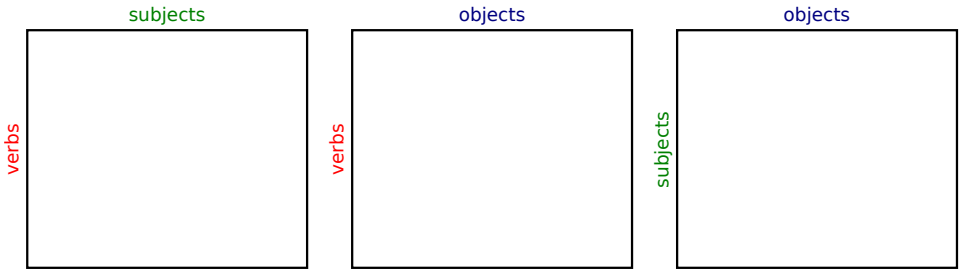


matrix

three-way

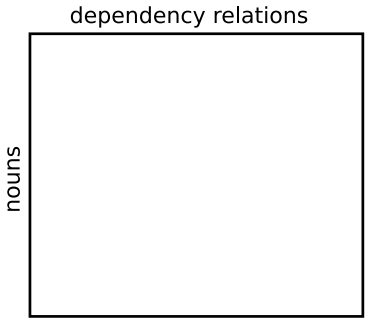


tensor



Two-way vs. three-way

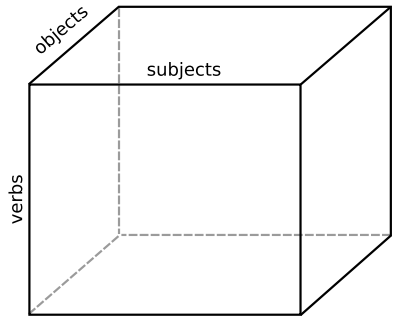
two-way



matrix

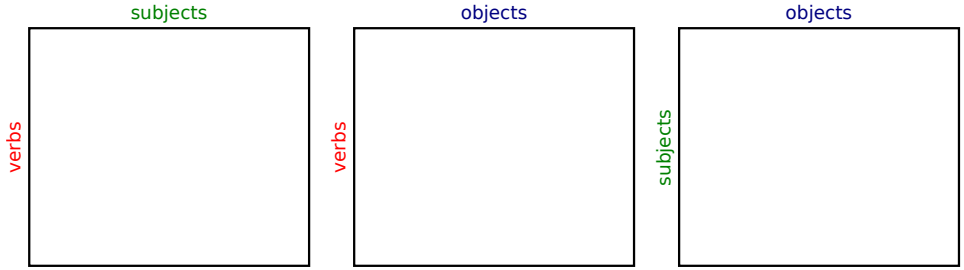
$\longleftrightarrow$

three-way



tensor

$\longleftrightarrow$



pairwise matrices

Factorization

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)

Different flavours

- Principal component analysis
- Latent semantic analysis (singular value decomposition)
- Probabilistic latent semantic analysis
- Topic models – latent dirichlet allocation
- Non-negative matrix factorization

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

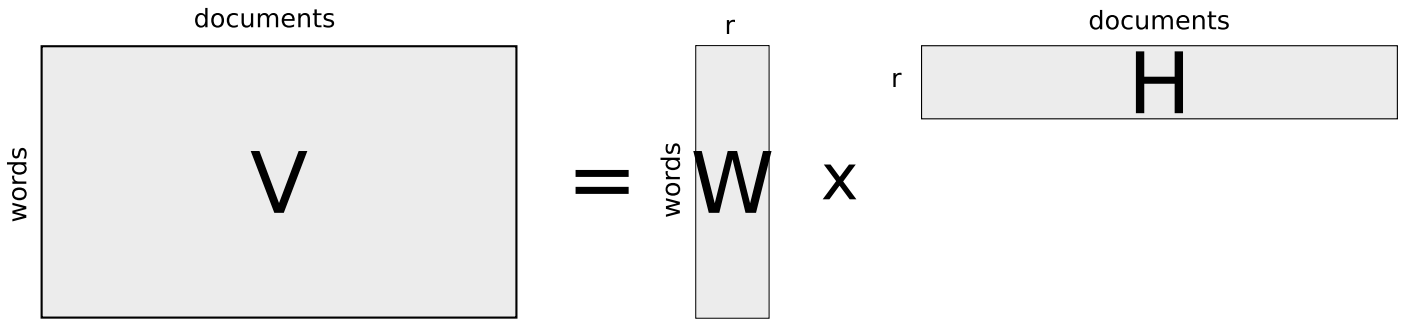
Non-negative matrix factorization

- Different kinds of NMF's that minimize different cost functions:
  - Square of Euclidean distance (L1-norm)
  - Kullback-Leibler Divergence (L2-norm)
    $\Rightarrow$ better suited for language phenomena
- To find NMF is to minimize $D(V\|WH)$ with respect to W and H, subject to the constraints $W, H \geq 0$
- This can be done with *update rules*

$$\mathbf{H}_{a\mu} \leftarrow \mathbf{H}_{a\mu} \frac{\sum_i \mathbf{W}_{ia} \frac{\mathbf{V}_{i\mu}}{(\mathbf{WH})_{i\mu}}}{\sum_k \mathbf{W}_{ka}} \quad \mathbf{W}_{ia} \leftarrow \mathbf{W}_{ia} \frac{\sum_\mu \mathbf{H}_{a\mu} \frac{\mathbf{V}_{i\mu}}{(\mathbf{WH})_{i\mu}}}{\sum_v \mathbf{H}_{av}} \quad (2)$$

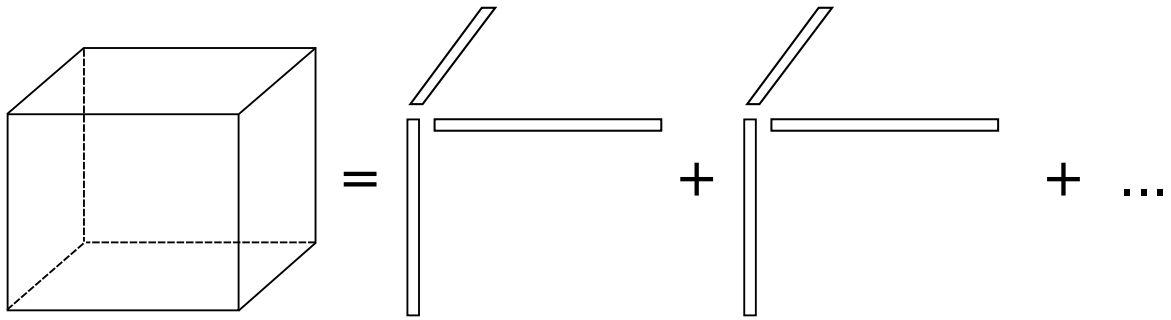
- these update rules converge to a *local optimum* in the minimization of KL divergence

Graphical Representation

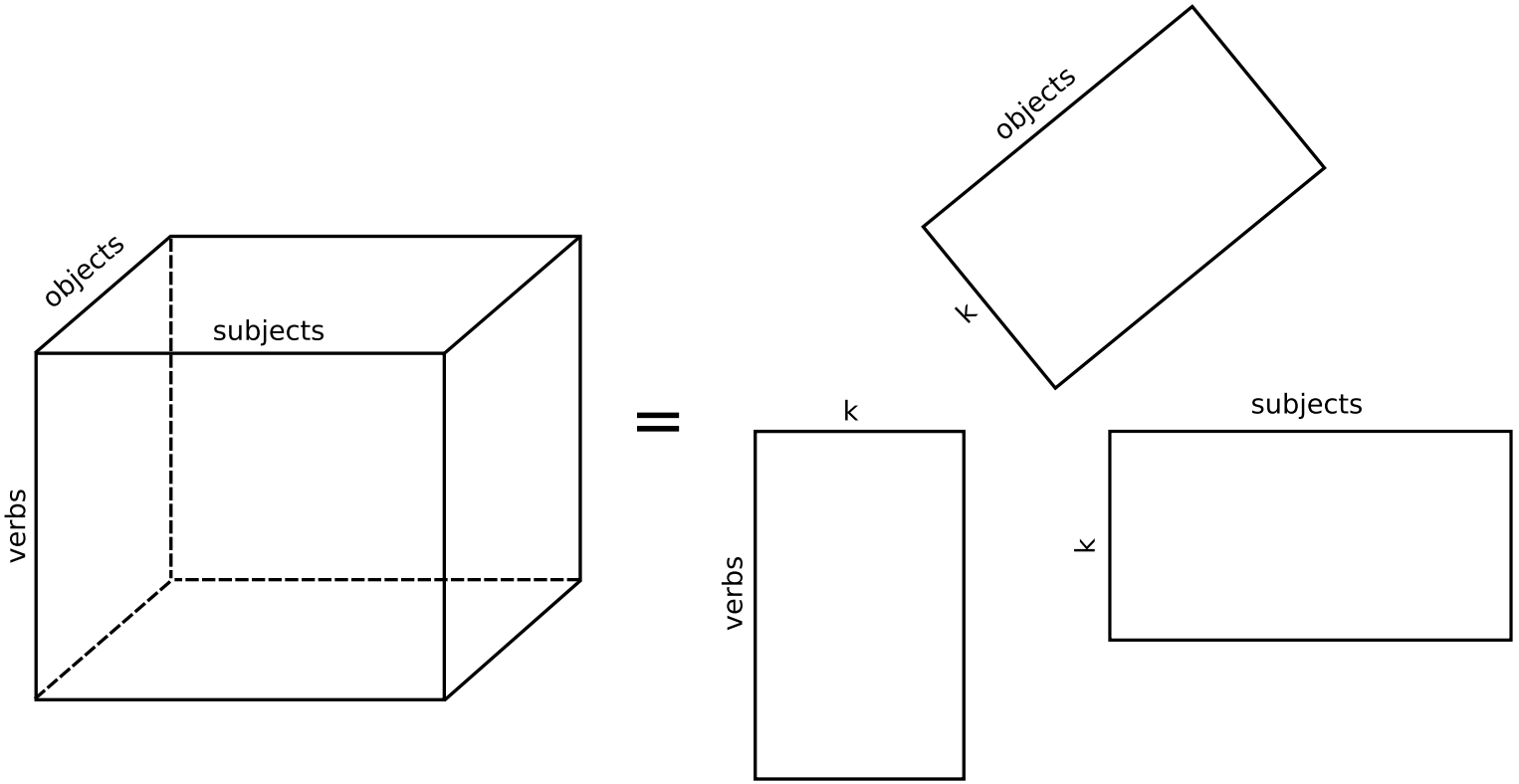


Technique

- Idea similar to non-negative matrix factorization
- Calculations are different
- $\min_{x_i \in \mathbb{R}^{D1}_{\geq 0}, y_i \in \mathbb{R}^{D2}_{\geq 0}, z_i \in \mathbb{R}^{D3}_{\geq 0}} \| T - \sum_{i=1}^{k} x_i \circ y_i \circ z_i \|_F^2$



Graphical representation



Introduction 1/2

- Standard selectional preference models: two-way co-occurrences
- Keeping track of single relationships
- But: two-way selectional preference models are not sufficiently rich
- Compare:
  - *The skyscraper is playing coffee.*
  - *The turntable is playing the piano.*

Introduction 2/2

- *The skyscraper is playing coffee.*
  - $(play, su, scyscraper)$ $\downarrow$
  - $(play, obj, coffee)$ $\downarrow$
- *The turntable is playing the piano.*
  - $(play, su, turntable)$ $\uparrow$
  - $(play, obj, piano)$ $\uparrow$
  - $(play, su, turntable, obj, piano)$ $\downarrow$

Methodology

- Three-way extraction of selectional preferences
- Approach applied to Dutch, using TWENTE NIEUWS CORPUS (500M words of newspaper texts)
- parsed with Dutch dependency parser ALPINO
- three-way co-occurrence of verbs with subjects and direct objects extracted
- adapted with extension of pointwise mutual information
- Resulting tensor 1K verbs $\times$ 10K subjects $\times$ 10K direct objects
- reduction to k dimensions ($k = 50, 100, 300$)

Examples: police action

subjects	su_s	verbs	v_s	objects	obj_s
politie 'police'	.99	*houd_aan* 'arrest'	.64	*verdachte* 'suspect'	.16
agent 'policeman'	.07	*arresteer* 'arrest'	.63	*man* 'man'	.16
autoriteit 'authority'	.05	*pak_op* 'run in'	.41	*betoger* 'demonstrator'	.14
Justitie 'Justice'	.05	*schiet_dood* 'shoot'	.08	*relschopper* 'rioter'	.13
recherche 'detective force'	.04	*verdenk* 'suspect'	.07	*raddraaier* 'instigator'	.13
marechaussee 'military police'	.04	*tref_aan* 'find'	.06	*overvaller* 'raider'	.13
justitie 'justice'	.04	*achterhaal* 'overtake'	.05	*Roemeen* 'Romanian'	.13
arrestatieteam 'special squad'	.03	*verwijder* 'remove'	.05	*actievoerder* 'campaigner'	.13
leger 'army'	.03	*zoek* 'search'	.04	*hooligan* 'hooligan'	.13
douane 'customs'	.02	*spoor_op* 'track'	.03	*Algerijn* 'Algerian'	.13

Examples: legislation

subjects	su_s	**verbs**	v_s	**objects**	obj_s
meerderheid 'majority'	.33	*steun* 'support'	.83	*motie* 'motion'	.63
VVD	.28	*dien_in* 'submit'	.44	*voorstel* 'proposal'	.53
D66	.25	*neem_aan* 'pass'	.23	*plan* 'plan'	.28
Kamermeerderheid	.25	*wijs_af* 'reject'	.17	*wetsvoorstel* 'bill'	.19
fractie 'party'	.24	*verwerp* 'reject'	.14	*hem* 'him'	.18
PvdA	.23	*vind* 'think'	.08	*kabinet* 'cabinet'	.16
CDA	.23	*aanvaard* 'accepts'	.05	*minister* 'minister'	.16
Tweede Kamer	.21	*behandel* 'treat'	.05	*beleid* 'policy'	.13
partij 'party'	.20	*doe* 'do'	.04	*kandidatuur* 'candidature'	.11
Kamer 'Chamber'	.20	*keur_goed* 'pass'	.03	*amendement* 'amendment'	.09

Examples: exhibition

subjects	su_s	verbs	v_s	objects	obj_s
tentoonstelling 'exhibition'	.50	*toon* 'display'	.72	*schilderij* 'painting'	.47
expositie 'exposition'	.49	*omvat* 'cover'	.63	*werk* 'work'	.46
galerie 'gallery'	.36	*bevat* 'contain'	.18	*tekening* 'drawing'	.36
collectie 'collection'	.29	*presenteer* 'present'	.17	*foto* 'picture'	.33
museum 'museum'	.27	*laat* 'let'	.07	*sculptuur* 'sculpture'	.25
oeuvre 'oeuvre'	.22	*koop* 'buy'	.07	*aquarel* 'aquarelle'	.20
Kunsthal	.19	*bezit* 'own'	.06	*object* 'object'	.19
kunstenaar 'artist'	.15	*zie* 'see'	.05	*beeld* 'statue'	.12
dat 'that'	.12	*koop_aan* 'acquire'	.05	*overzicht* 'overview'	.12
hij 'he'	.10	*in huis heb* 'own'	.04	*portret* 'portrait'	.11

Examples: quality count

- 44 dimensions contain similar, framelike semantics
- 43 dimensions contain less clear-cut semantics
  - single verbs account for one dimension
  - verb senses are mixed up
- 13 dimensions based on syntax rather than semantics
  - fixed expressions
  - pronomina

Evaluation: methodology

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

s	v	o	s'	o'
jongere	*drink*	*bier*	*coalitie*	*aandeel*
'youngster'	'drink'	'beer'	'coalition'	'share'
werkgever	*riskeer*	*boete*	*doel*	*kopzorg*
'employer'	'risk'	'fine'	'goal'	'worry'
directeur	*zwaai*	*scepter*	*informateur*	*vodka*
'manager'	'sway'	'sceptre'	'informer'	'wodka'

- 10-fold cross validation ($\pm$ 300,000 co-occurrences)

Evaluation: models

- Evaluation of 4 different models
- 2 matrix models
  $\longrightarrow$ 1K verbs $\times$ (10K subjects + 10K direct objects)
  - singular value decomposition ($\mathbb{R}$)
  - non-negative matrix factorization ($\mathbb{R}_{\geq 0}$)
- 2 tensor models
  $\longrightarrow$ 1K verbs $\times$ 10K subjects $\times$ 10K direct objects
  - parallel factor analysis ($\mathbb{R}$)
  - non-negative tensor factorization ($\mathbb{R}_{\geq 0}$)

Evaluation: results

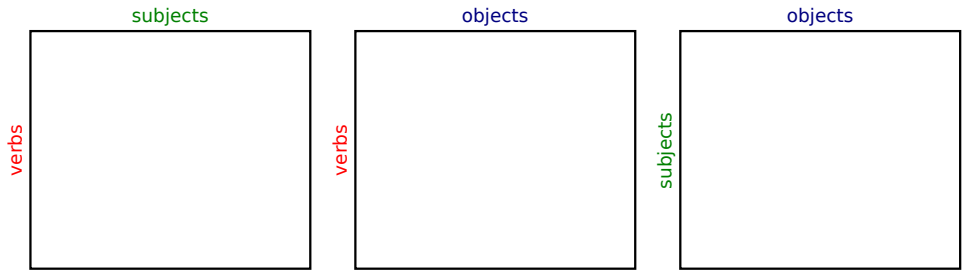
	dimensions		
	50 (%)	100 (%)	300 (%)
SVD	69.60 ± 0.41	62.84 ± 1.30	45.22 ± 1.01
NMF	81.79 ± 0.15	78.83 ± 0.40	75.74 ± 0.63
PARAFAC	85.57 ± 0.25	83.58 ± 0.59	80.12 ± 0.76
NTF	**89.52** ± 0.18	**90.43** ± 0.14	**90.89** ± 0.16

What if tensor factorization is infeasible?

- Significant space requirements, not feasible for large dataset

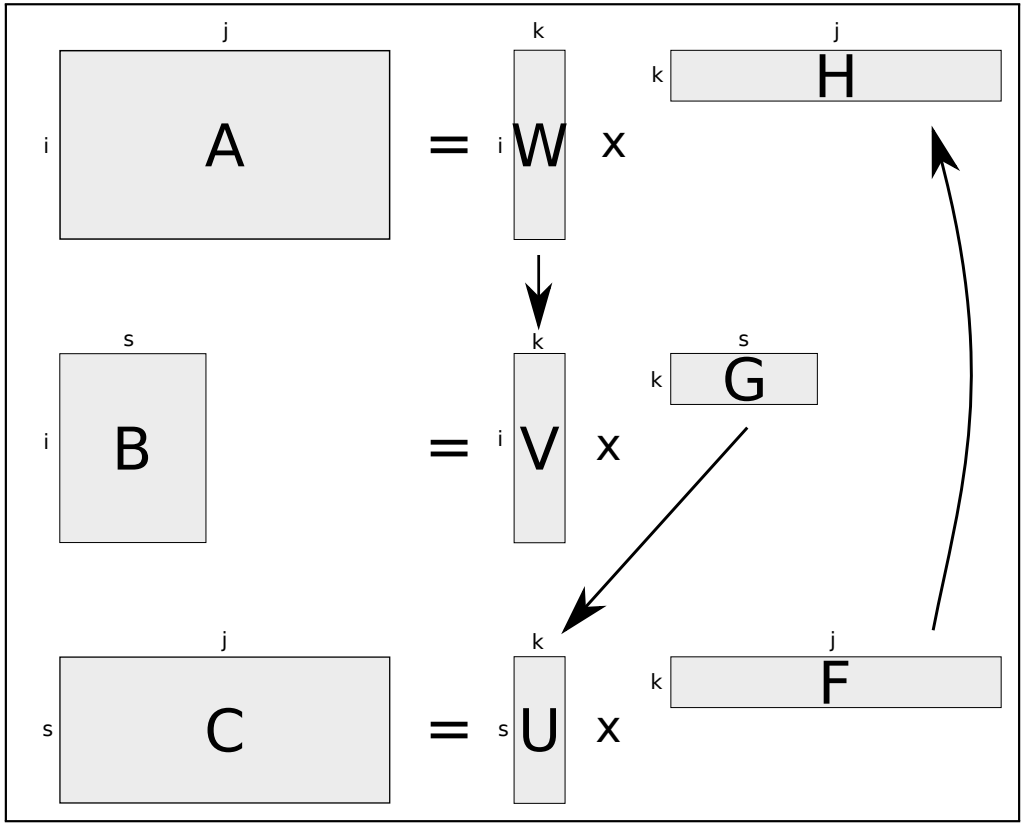
What if tensor factorization is infeasible?

- Significant space requirements, not feasible for large dataset
- Solution: use pairwise co-occurrences and combine matrices in factorization



- $\Rightarrow$ Apply NMF to matrices, but interleave the process
- Result of former update step is used to initialize the next one

Graphical Representation



Ambiguity

- **Problem**: ambiguity
- BAR

Ambiguity

- **Problem**: ambiguity
- BAR

Ambiguity

- **Problem**: ambiguity
- BAR

$\longleftrightarrow$

Ambiguity

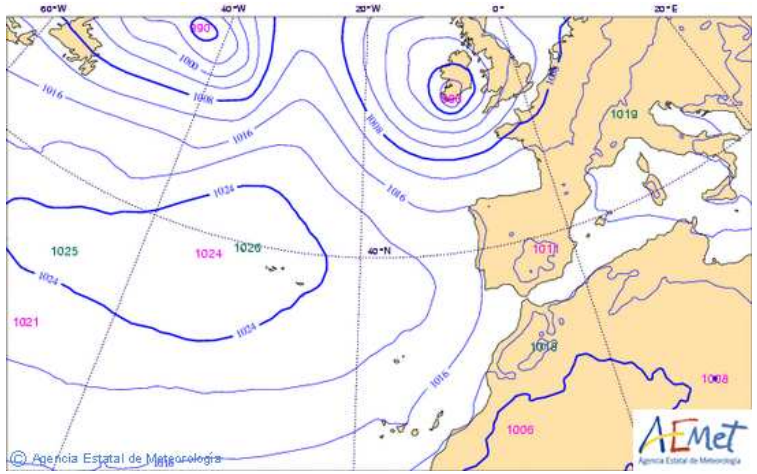
- **Problem**: ambiguity
- BAR

$\longleftrightarrow$

Ambiguity

- **Problem**: ambiguity
- BAR

$\longleftrightarrow$



Ambiguity

- **Problem**: ambiguity
- BAR

$\longleftrightarrow$

Ambiguity

- **Problem**: ambiguity
- BAR

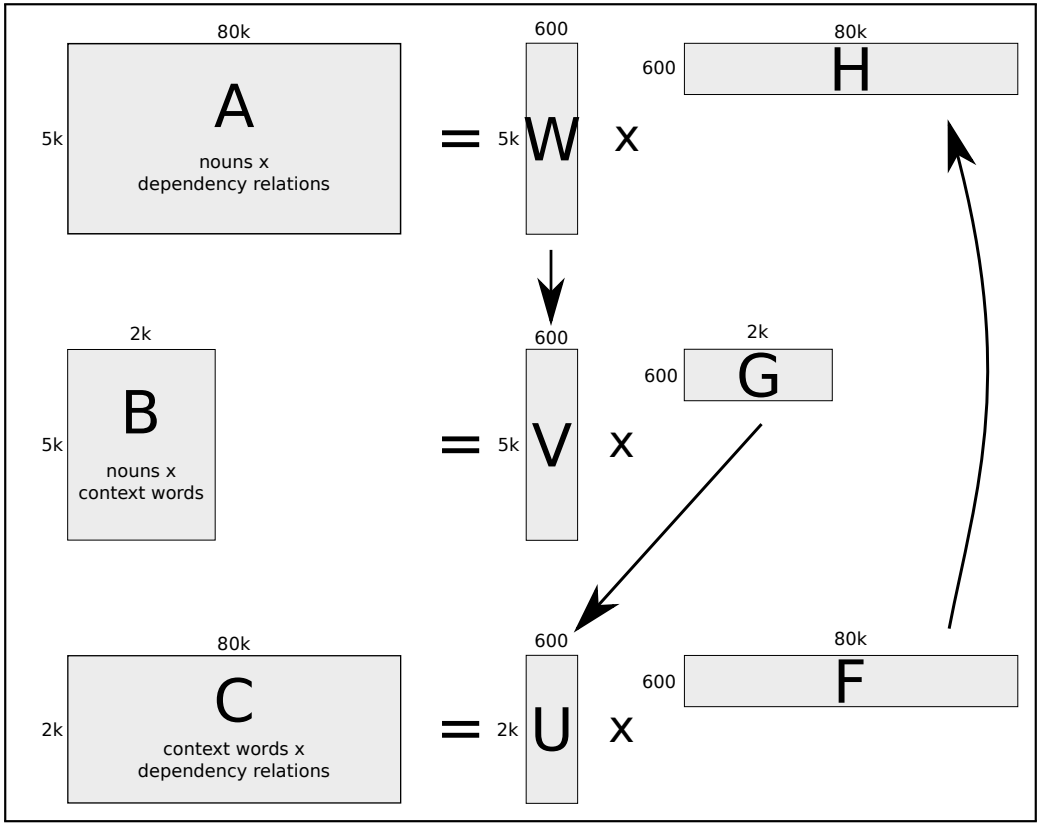
$\leftrightarrow$

- Main research question: can 'topical' similarity and tight, synonym-like similarity be combined to compute meaning of word in a particular context?

Methodology

- 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



Word meaning in context

- NMF can be interpreted probabilistically
  - Matrix $\mathbf{W} \rightarrow p(w_i|\mathbf{z})$
  - Matrix $\mathbf{H} \rightarrow p(\mathbf{z}|d_j), p(\mathbf{d}|\mathbf{z})$
  - Matrix $\mathbf{G} \rightarrow p(\mathbf{z}|c_i)$
- $p(\mathbf{z}|C) = \frac{\sum_{c_i \in C} p(\mathbf{z}|c_i)}{|C|}$ – 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 according to the importance given the context

Example

1. Jack is listening to a **record**. $\rightarrow C_1 = \{listen^{-1}_{prep(to)}\}$
   - $p(\mathbf{z}|C_1) \rightarrow p(\mathbf{d}|C_1) \rightarrow p(\mathbf{d}|w_i, C_1)$
   - $\mathbf{record}_N$, C_1: *album*, *song*, *recording*, *track*, *cd*
2. Jill updated the **record**. $\rightarrow C_2 = \{update^{-1}_{obj}\}$
   - $p(\mathbf{z}|C_2) \rightarrow p(\mathbf{d}|C_2) \rightarrow p(\mathbf{d}|w_i, C_2)$
   - $\mathbf{record}_N$, C_2: *file*, *datum*, *document*, *database*, *list*

Implementational details

- method applied to English and French
  - UKWaC corpus, parsed with MaltParser
  - French Wikipedia, parsed with FRMG
- one model per pos (noun,adjective,verb,adverb)
- NMF model: $K = 600$, 100 iterations
- interleaved NMF algorithm implemented in Matlab, preprocessing and vector computation in Python.

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

nouns	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 316

nouns	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

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

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
NMF$_{context}$	20.64**	47.60**
NMF$_{dep}$	22.49**	48.97**
NMF$_{c+d}$	**22.59****	**49.02****

Paraphrase induction

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

Conclusion

Beneficial to consider language as a multi-way co-occurrence problem

- Tensor space
  - novel model to investigate three-way (up to *n*-way) co-occurrence data
  - Possible to generalize over co-occurrence data with appropriate factorization models
  - Applicable to and beneficial for three-way selectional preference induction
- Pairwise matrices
  - 'makeshift' multi-way co-occurrence modeling
  - Useful when tensor approach is not feasible
  - Applicable to and beneficial for computation of word meaning in context