Distributional Semantics

The distributional hypothesis

- The meaning of a word is the set of contexts in which it occurs in texts (or)

Important aspects of the meaning of a word are a function of (can be approximated by) the set of contexts in which it occurs in texts

- Words that have similar contexts are likely to have similar meaning

Meaning from context

Consider the example from J&M

- a bottle of *tango* is on the table
- everybody likes *tango*
- *tango* makes you drunk
- we make *tango* out of corn

Distributional Semantics

- The study of statistical patterns of human words usage to extract semantics
- “.. If we consider words or morphemes A and B to be more different in meaning than A and C, then we will often find that the distributions of A and B are more different than the distributions of A and C. In other words, difference in meaning correlates with difference of distribution.” (Zellig Harris, 1954)
- what is the type of distributional semantics?
- How they capture that and how can I use that for certain meaningful applications?

Basic Idea

- **Distributions** are vectors in a multidimensional semantic space, that is, objects with a magnitude (length) and a direction.
- The **semantic space** has dimensions which correspond to possible contexts

Vector Space Model

- Represent each word w_i as a vector of its contexts – distributional semantic models also called vector-space models.

Ex: each dimension is a context word; $= 1$ if it co-occurs with w_i, otherwise 0.

	pet	bone	fur	run	brown	screen	mouse	fetch
$w_1 =$	1	1	1	1	1	0	0	1
$w_2 =$	1	0	1	0	1	0	1	0
$w_3 =$	0	0	0	1	0	1	1	0

Small Dataset

An automobile is a wheeled motor vehicle used for transporting passengers .
A car is a form of transport , usually with four wheels and the capacity to carry around five passengers .
Transport for the London games is limited , with spectators strongly advised to avoid the use of cars .
The London 2012 soccer tournament began yesterday , with plenty of goals in the opening matches .
Giggs scored the first goal of the football tournament at Wembley , North London .
Bellamy was largely a passenger in the football match , playing no part in either goal .

Target words: ⟨automobile, car, soccer, football⟩
Term vocabulary: ⟨wheel, transport, passenger, tournament, London, goal, match⟩

Distributional vector

- Count how many times each target word occurs in a certain context
  - Context window- doc , sentence, # words, POS based (eg. Only N), Dependency tree relation.
- Build vectors out of (a function of) these context occurrence counts (co-occurance)
- Similar words will have similar vectors

Computing Similarity

	wheel	transport	passenger	tournament	London	goal	match
automobile	1	1	1	0	0	0	0
car	1	2	1	0	1	0	0
soccer	0	0	0	1	1	1	1
football	0	0	1	1	1	2	1

Using simple vector product

automobile . car = 4
automobile . soccer = 0
automobile . football = 1
car . soccer = 1
car . football = 2
soccer . football = 5

The notion of context

- Word windows (unfiltered): n words on either side of the lexical item. Example: n=2 (5 words window): *... the prime* ***minister*** *acknowledged that ...*
- Word windows (filtered): n words on either side removing some words (e.g. some very frequent content words). Stop-list or by POS-tag. Example: n=2 (5 words window): ... the prime **minister** acknowledged that ...
- Lexeme window (filtered or unfiltered); as above but using stems.

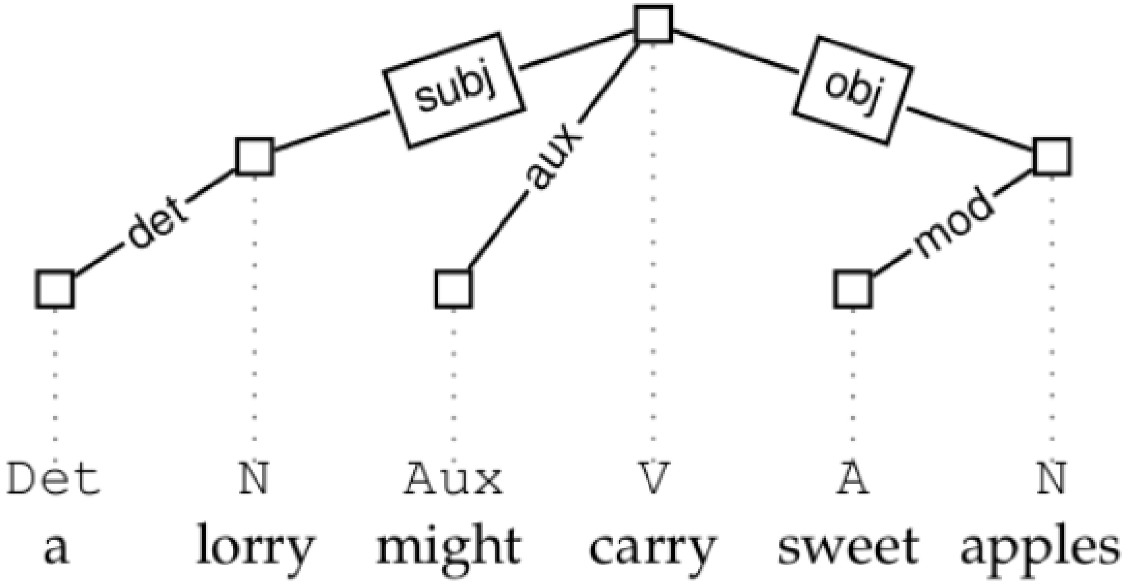
Same corpus (BNC), different contexts (window sizes)

- Nearest neighbours of dog

2-word window	30-word window
cat	kennel
horse	puppy
fox	pet
pet	bitch
rabbit	terrier
pig	rottweiler
animal	canine
mongrel	cat
sheep	to bark
pigeon	Alsatian

The notion of context

- Dependencies: syntactic or semantic (directed links between heads and dependents). Context for a lexical item is the dependency structure it belongs to (Padó and Lapata, 2007)



a	[Det,det,N]	lorry
lorry	[N,subj,V]	carry
might	[Aux,aux,V]	carry
apples	[N,obj,V]	carry
sweet	[A,mod,N]	apples

Document as context

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
against	0	0	0	1	0	0	3	2	3	0
age	0	0	0	1	0	3	1	0	4	0
agent	0	0	0	0	0	0	0	0	0	0
ages	0	0	0	0	0	2	0	0	0	0
ago	0	0	0	2	0	0	0	0	3	0
agree	0	1	0	0	0	0	0	0	0	0
ahead	0	0	0	1	0	0	0	0	0	0
ain't	0	0	0	0	0	0	0	0	0	0
air	0	0	0	0	0	0	0	0	0	0
aka	0	0	0	1	0	0	0	0	0	0

Words as context

	against	age	agent	ages	ago	agree	ahead	ain.t	air	aka	al
against	2003	90	39	20	88	57	33	15	58	22	24
age	90	1492	14	39	71	38	12	4	18	4	39
agent	39	14	507	2	21	5	10	3	9	8	25
ages	20	39	2	290	32	5	4	3	6	1	6
ago	88	71	21	32	1164	37	25	11	34	11	38
agree	57	38	5	5	37	627	12	2	16	19	14
ahead	33	12	10	4	25	12	429	4	12	10	7
ain't	15	4	3	3	11	2	4	166	0	3	3
air	58	18	9	6	34	16	12	0	746	5	11
aka	22	4	8	1	11	19	10	3	5	261	9
al	24	39	25	6	38	14	7	3	11	9	861

How to weight the context words?

- Binary model: if context c co-occurs with word w, value of vector w for dimension c is 1, 0 otherwise.

  ... [a long long long ***example*** *for a distributional semantics] model...* (n=4) ...
  {dog 0} {long 1} {sell 0} {semantics 1}...

- Basic frequency model: the value of vector w for dimension c is the number of times that c co-occurs with w.

  ... [a long long long example for a distributional semantics] model... (n=4) ...
  {{dog 0} {long 3} {sell 0} {semantics 1}...

- Rectangle or tringle window

Other methods (Weight)

- Reduce the effect of high frequency words by applying a weighting scheme
  - Pointwise mutual information (PMI), TF-IDF
- Smoothing by dimensionality reduction
  - Singular value decomposition (SVD), principal component analysis (PCA), matrix factorization methods

One-hot Vector

- We have vocabulary of size V.
- w_i =[0 0 0 … 1 … 0 0]; 1 at i^{th} position in the vector
- Meaning of word??-> dot product of words will be zero

Words are treated as atomic symbol

Summary

- Linguistic step:
  - corpus must be tokenized
  - POS tagging, lemmatization, dependency parsing. . .
  - Define the target and context
- Mathematical step:
  - Count the target- context co-occurrence
  - Weight the context (optional)
  - Build the distribution matrix
  - Reduce the matrix dimensions (optional)
  - Compute the vector distances on the (reduced) matrix

Matrix type		Weighting		Dimensionality reduction		Vector comparison
word × document		probabilities		LSA		Euclidean
word × word		length normalization		PLSA		Cosine
word × search proximity	×	TF-IDF	×	LDA	×	Dice
adj. × modified noun		PMI		PCA		Jaccard
word × dependency rel.		Positive PMI		IS		KL
verb × arguments		PPMI with discounting		DCA		KL with skew
⋮		⋮		⋮		⋮

Weight of context

- Weight α Term frequency (f_{ij})
- Weight α 1/len of document

  |d1|=1000, |d2|=20, freq(d1,w1)=50, freq(d2,w1)=10
- Weight α inverse document frequency($1/N_j$)
  - w1 occur in 3 doc and w2 occur in 100 document, w1 seems to be more informative than w2 (which seems to be a very general term)

- Tf-idf = $f_{ij} * \log\left(\frac{N}{N_j}\right)$ where ith word and jth document

Associative Measures

basic intuition

word1	word2	freq(1,2)	freq(1)	freq(2)
dog	small	855	33,338	490,580
dog	domesticated	29	33,338	918

- Associative measures are used to give more weight to contexts that are more significantly associated with a target word
  - The less frequent the target and context element are, the higher the weight given to their co-occurrence count should be
    - co-occurrence with frequent context element *small* is less informative than co-occurrence with rare *domesticated*
  - Different measures- Mutual information Log likelihood ratio, etc.

Mutual Information

Two random variables X, Y are **independent** iff their joint distribution is equal to the product of their individual distributions:

$$p(X, Y) = p(X)p(Y)$$

That is, for all outcomes x, y:

$$p(X{=}x, Y{=}x) = p(X{=}x)p(Y{=}y)$$

$I(X;Y)$, the **mutual information** of two random variables X and Y is defined as

$$I(X;Y) = \sum_{X,Y} p(X=x, Y=y) \log \frac{p(X=x, Y=y)}{p(X=x)p(Y=y)}$$

Pointwise Mutual Information (PMI)

Recall that two **events** x, y are **independent** if their joint probability is equal to the product of their individual probabilities:

x,y are independent iff $p(x,y) = p(x)p(y)$

x,y are independent iff $p(x,y)/p(x)p(y) = 1$

In NLP, we often use the pointwise mutual information (PMI) of two outcomes/events (e.g. words):

$$PMI(x,y) = \log \frac{p(X=x, Y=y)}{p(X=x)p(Y=y)}$$

Using PMI to find related words

Find pairs of words w_i, w_j that have high pointwise mutual information:

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

Different ways of defining $p(w_i, w_j)$ give different answers.

Example

Context: ± 7 words

sugar, a sliced lemon, a tablespoonful of	**apricot**	preserve or jam, a pinch each of,
their enjoyment. Cautiously she sampled her first	**pineapple**	and another fruit whose taste she likened
well suited to programming on the digital	**computer**.	In finding the optimal R-stage policy from
for the purpose of gathering data and	**information**	necessary for the study authorized in the

Resulting word-word matrix:

f(w, c) = how often does word w appear in context c:
"information" appeared six times in the context of "data"

	aardvark	computer	data	pinch	result	sugar	…
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0	2	1	0	1	0	
information	0	1	6	0	4	0	

- f_{ij} is number of times w_i occurs in context c_j

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \qquad p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \qquad p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}} \qquad ppmi_{ij} = \begin{cases} pmi_{ij} & \text{if } pmi_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

Count(w,context)

	computer	data	pinch	result	sugar
apricot	0	0	1	0	1
pineapple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

p(w=information,c=data) = 6/19 = .32

p(w=information) = 11/19 = .58

p(c=data) = 7/19 = .37

$$p(w_i) = \frac{\sum_{j=1}^{C} f_{ij}}{N} \qquad p(c_j) = \frac{\sum_{i=1}^{W} f_{ij}}{N}$$

p(w,context) **p(w)**

	computer	data	pinch	result	sugar	p(w)
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*} p_{*j}}$$

p(w,context) / **p(w)**

	computer	data	pinch	result	sugar	p(w)
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

- pmi(information,data) = $\log_2$ (.32 / (.37*.58)) = .57

PPMI(w,context)

	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

Problem with PPMI

- Infrequent events: P(w1))=P(W2)=P(w1,w2)
  - PMI=log(1/P(w1)); PMI will be high if P(w1) is very low i.e. it is very rare and hence it will get very high value (Noise)

- Two solutions:

•Give rare words slightly higher probabilities

•Use add-one smoothing (which has a similar effect)

Giving rare context words slightly higher probability

- Raise the context probabilities to $\alpha = 0.75$:

$$\text{PPMI}_\alpha(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P_\alpha(c)}, 0)$$

$$P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha}$$

- This helps because $P_\alpha(c) > P(c)$ for rare c
- Consider two events, P(a) = .99 and P(b)=.01 (here we use probability to show the effect)
- $P_\alpha(a) = \frac{.99^{.75}}{.99^{.75} + .01^{.75}} = .97 \quad P_\alpha(b) = \frac{.01^{.75}}{.99^{.75} + .01^{.75}} = .03$

Add-n smoothing

Add-2 Smoothed Count

	computer	data	pinch	result	sugar
apricot	2	2	3	2	3
pineapple	2	2	3	2	3
digital	4	3	2	3	2
information	3	8	2	6	2

	p(w,context) [add-2]					**p(w)**
	computer	data	pinch	result	sugar	
apricot	0.03	0.03	0.05	0.03	0.05	0.20
pineapple	0.03	0.03	0.05	0.03	0.05	0.20
digital	0.07	0.05	0.03	0.05	0.03	0.24
information	0.05	0.14	0.03	0.10	0.03	0.36
p(context)	0.19	0.25	0.17	0.22	0.17	

Applications

Query Expansion

Term mismatch problem in IR (semantically similar but at surface level different)

- Stems from the word independence assumption during document indexing
- User query: *insurance cove which pays for long term care*
- A relevant document may contain terms different from the actual user query
- Some relevant words concerning this query (medicare, premiums, insurers)

Use DSMs for Query Expansion

Given a user query, reformulate it using related terms to enhance the retrieval performance

- The distributional vectors for query terms are computed
- Expanded query is obtained by a linear combination or a functional combination of these vectors

TREC Topic 104: catastrophic health insurance

Query Representation: surtax:1.0 hcfa:0.97 medicare:0.93 hmos:0.83 medicaid:0.8 hmo:0.78 beneficiaries:0.75 ambulatory:0.72 premiums:0.72 hospitalization:0.71 hhs:0.7 reimbursable:0.7 deductible:0.69

- Broad expansion terms: **medicare, beneficiaries, premiums** ...
- Specific domain terms: **HCFA** (Health Care Financing Administration), **HMO** (Health Maintenance Organization), **HHS** (Health and Human Services)

TREC Topic 355: ocean remote sensing

Query Representation: radiometer:1.0 landsat:0.97 ionosphere:0.94 cnes:0.84 altimeter:0.83 nasda:0.81 meterology:0.81 cartography:0.78 geostationary:0.78 doppler:0.78 oceanographic:0.76

- Broad expansion terms: **radiometer, landsat, ionosphere** ...
- Specific domain terms: **CNES** (Centre National dÉtudes Spatiales) and **NASDA** (National Space Development Agency of Japan)

Similarity Measures

- Let X and Y denote the binary distributional vector for words X and Y
- Dice coefficient : $\frac{2|X \cap Y|}{|X|+|Y|}$
- Jaccard coefficient : $\frac{|X \cap Y|}{|X \cup Y|}$
- Overlap coefficient : $\frac{|X \cap Y|}{\min(|X|,|Y|)}$

- Jaccard coefficient penalizes small number of shared entities, while overlap coefficient uses the concept of inclusion

Similarity Measure for Vector Space

- Let $\vec{X}\ and\ \vec{Y}$ denoted the distributional vectors for word X and Y (n-dimenaions)
- Cosine similarity : $\cos(\vec{X}, \vec{Y}) = \frac{\vec{X}.\vec{Y}}{\overrightarrow{|X}||\vec{Y}|}$
- Euclidean distance:$\left|\vec{X} - \vec{Y}\right| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

Attributional similarity and Relational Similarity

- Attributional similarity between two words a and b depends on the degree of correspondence between the properties of a and b
- e.g. Dog and wolf

- Relational similarity is based on similar relations present with pair of words (a,b) and (c,d)
- e.g: dog:bark and cat:meow both pairs are relationally similar

Relational Similarity: Pair-pattern matrix

Pair- pattern Matrix:

- Row vectors corresponds to pair of words, such as mason:stone and carpenter:wood
- Columns corresponds to the patterns in which the pairs occur, e.g. *X cuts Y and X works with Y*
- Compute the similarity of rows to find similar pairs

Extended Distributional Hypothesis: Lin and Pantel

Patterns that co-occur with similar pairs tend to have similar meanings. This matrix can also be used to measure the semantic similarity of patterns

Given a pattern such as “X solves Y”, you can use this matrix to find similar patterns, such as “ Y is solved by X”, “ Y is resolved in X”, “ X resolves Y”

Structured DSM

Basic Issue:

- Words may not be basic context units anymore
- How to capture and represent syntactic information? – X solves Y

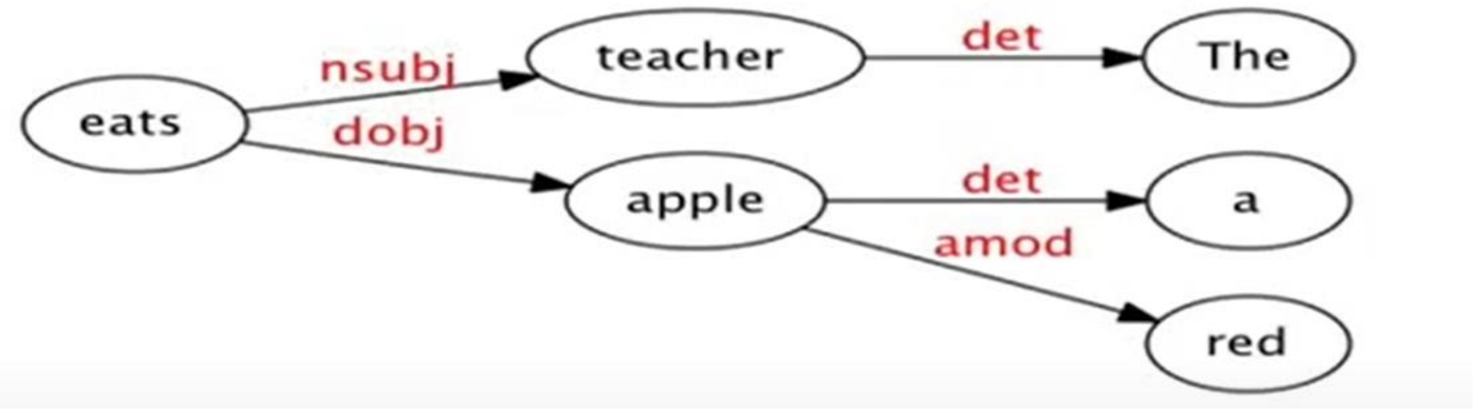
An Ideal Formalism

- Should mirror semantic relations as close as possible
- Incorporate word-based information and syntactic analysis
- Should be applicable to different languages

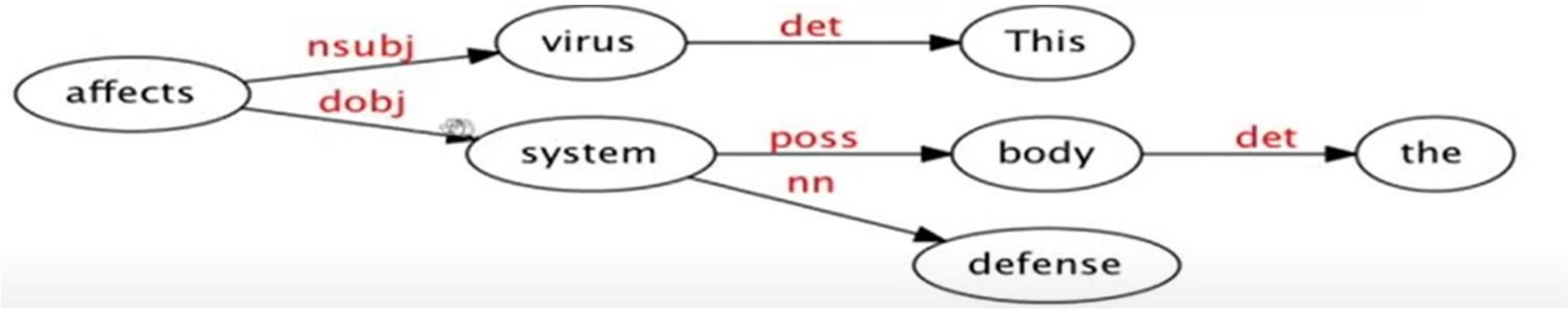
Use Dependency grammar framework

Structured DSM

- Using Dependency structure- “ The teacher eats a red apple.”



Using dependency Structure: find relation between pair of words



1) Forget dependency relations <system, affects>
2) <system, dobj-affects>
3) <system:affects, dobj>

Structured DSMs for Selectional Preferences

Selectional Preferences for Verbs

- Most verbs prefer arguments of particular type. This regularity is known as selectional preference

	obj-carry	obj-buy	obj-drive	obj-eat	obj-store	sub-fly	...
car	0.1	0.4	0.8	0.02	0.2	0.05	...
vegetable	0.3	0.5	0	0.6	0.3	0.05	...
biscuit	0.4	0.4	0	0.5	0.4	0.02	...
...	...	...	...	...	...	...	...

Selectional Preferences

- Suppose we want to compute the selectional preferences of the nouns as object of verb ‘eat’.
- N nouns having highest weight in the dimension ‘obj-eat’ are selected. Let {vegetable, biscuit,…} be the set of these n nouns
- The complete vectors of these n nouns are used to obtain an ‘object prototype’ of the verb
- Object prototype will indicate various attributes such as these nouns can be consumed, bought, carried, stored, etc.
- Similarity of a noun to this object prototype is used to denote the plausibility of that noun being an object of verb ‘eat’

Word Embedding

Distributional Representation

- Take a vector with several hundred dimensions
- Each word is represented by a distribution of weights across those elements
- So instead of one-to one mapping between an element in the vector and a word, the representation of a word is spread across all of the elements in the vector, and
- Each element in the vector contributes to the definition of many words

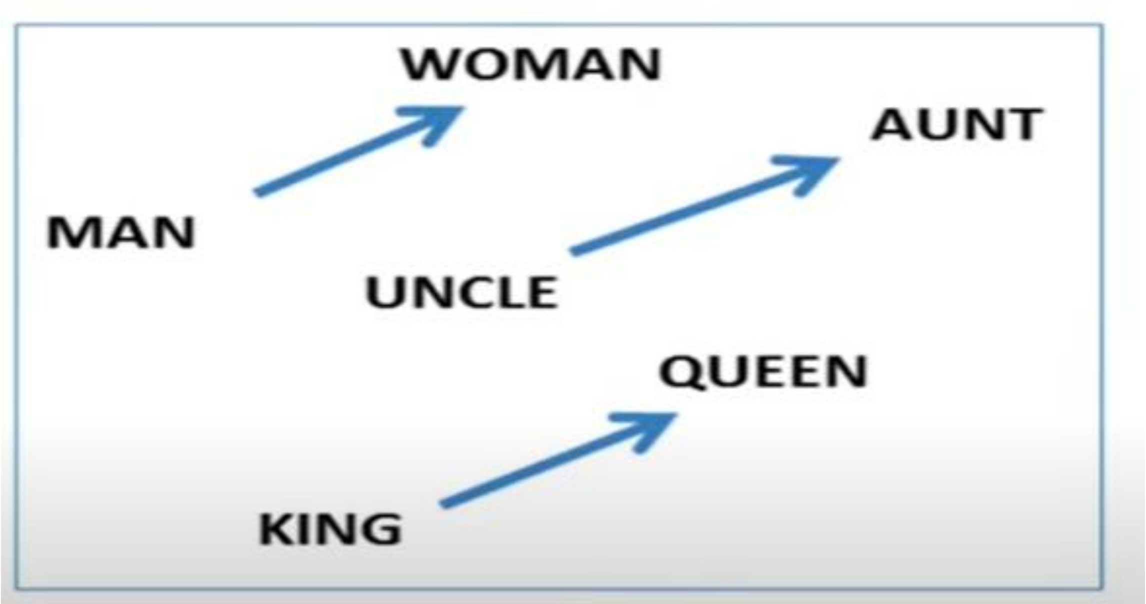
Distributional Representation: illustration

- If we label the dimension in a hypothetical word vector, it might look a bit like this- latent representation

Reasoning with word vector

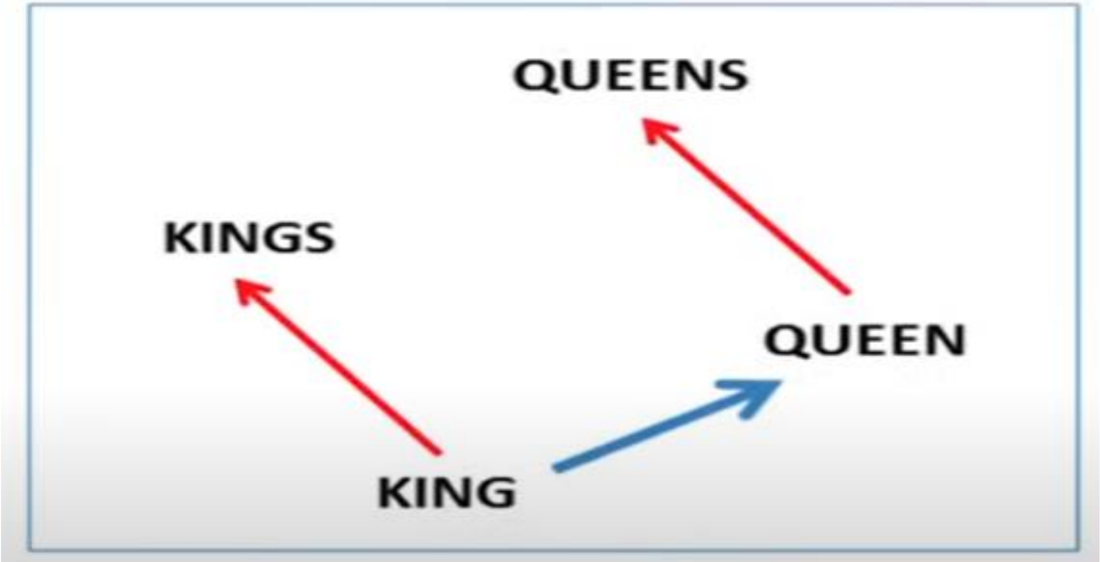
- Perhaps more surprisingly, we find that this is also the case for a variety of semantic relations
- Good at answering analogy questions-
  - *a is to b, as c is to ?*
  - Man is to woman as uncle is to ? (aunt)
- *A simple vector offset method based on cosine distance shows the relation*

Vector offset for gender relation



- Man-woman offset similar to uncle-aunt

Vector offset for Singular-Plural relation



Analogy Testing

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

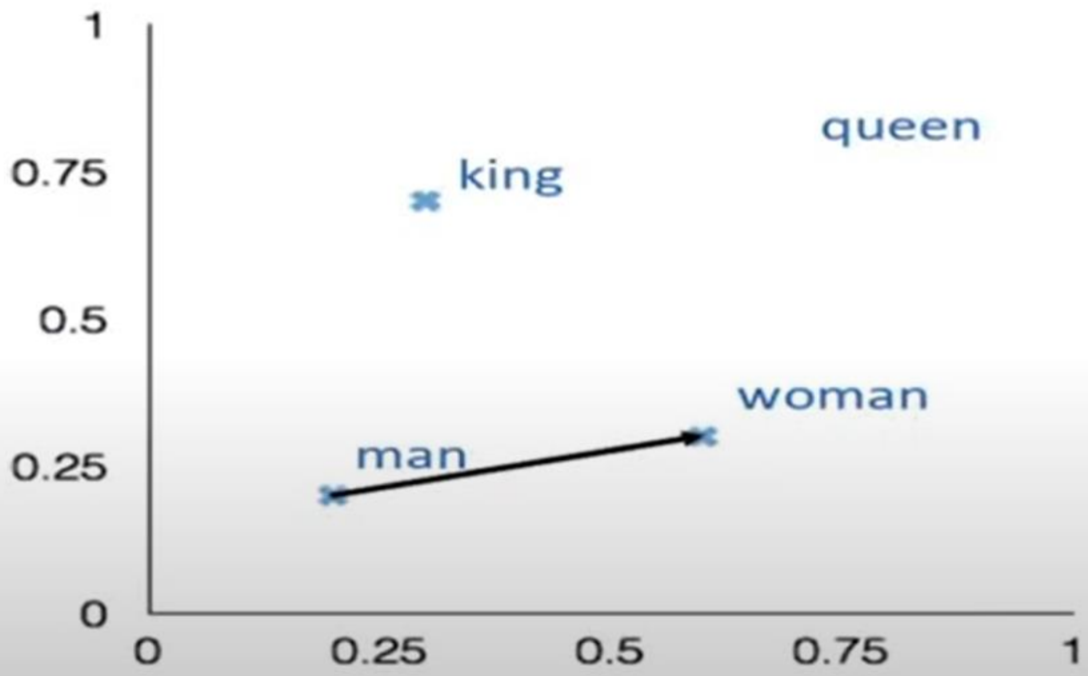
Analogy Testing

a:b :: c:?

$$d = \arg\max_{x} \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$$

man:woman :: king:?

+	king	[0.30 0.70]
-	man	[0.20 0.20]
+	woman	[0.60 0.30]
	queen	[0.70 0.80]



Country –capital relation

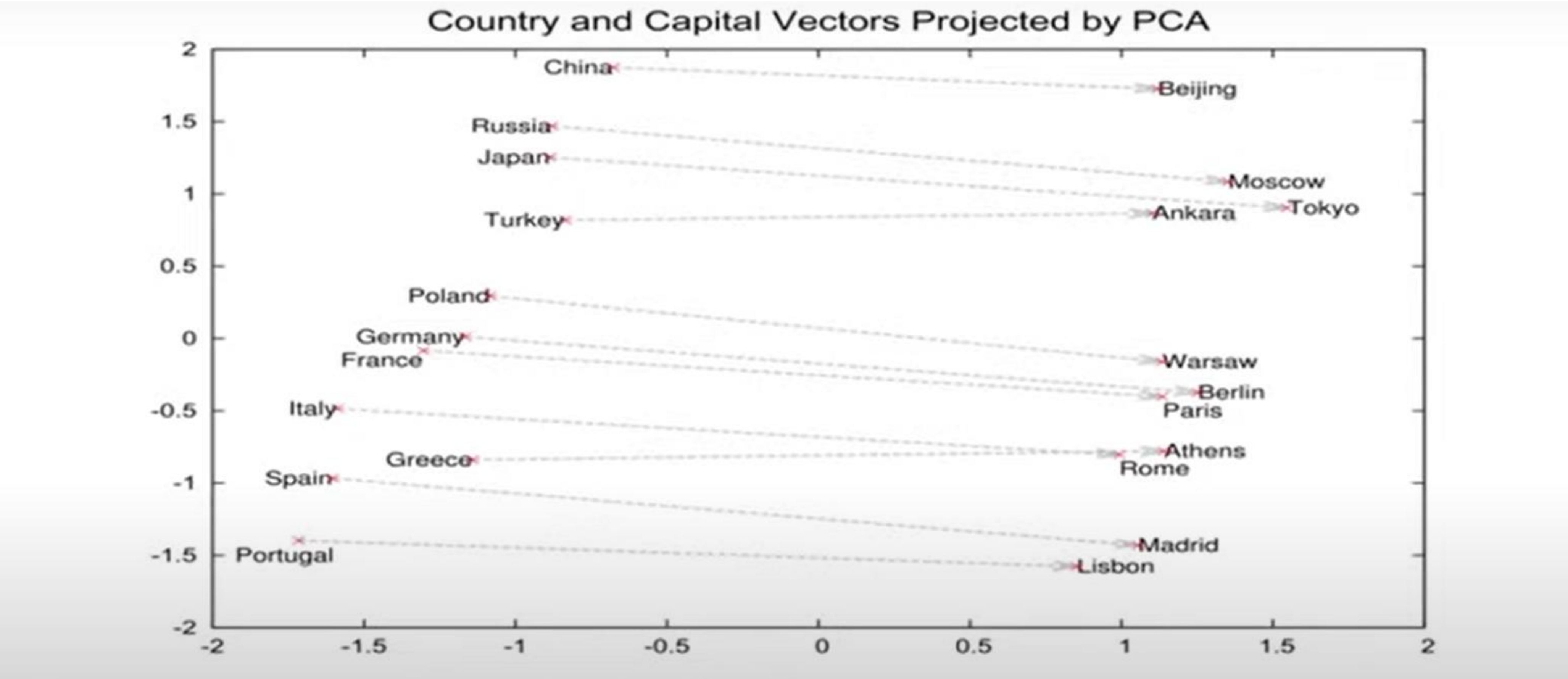


Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised information about what a capital city means.

Elementwise Addition

- We can also use element-wise addition of vector elements to ask questions such as ‘German+airlines’ and by looking at the closest token to the composite vector come up with impressive answers:

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

Table 5: Vector compositionality using element-wise addition. Four closest tokens to the sum of two vectors are shown, using the best Skip-gram model.

Learning Word Vectors

- Basic Idea
  - Instead of capturing co-occurrence counts directly, predict (using) surrounding words of every words

CBOW and Skip-grams