Distributional Semantics - Introduction

Pawan Goyal

CSE, IIT Kharagpur

Week 7, Lecture 1

Introduction

What is Semantics?

Introduction

What is Semantics?

The study of meaning: Relation between symbols and their denotata.

Introduction

What is Semantics?

The study of meaning: Relation between symbols and their denotata.
John told Mary that the train moved out of the station at 3 o'clock.

Computational Semantics

Computational Semantics

Computational Semantics

The study of how to automate the process of constructing and reasoning with meaning representations of natural language expressions.

Computational Semantics

Computational Semantics

The study of how to automate the process of constructing and reasoning with meaning representations of natural language expressions.

Methods in Computational Semantics generally fall in two categories:

- **Formal Semantics:** Construction of precise mathematical models of the relations between expressions in a natural language and the world.

Computational Semantics

Computational Semantics

The study of how to automate the process of constructing and reasoning with meaning representations of natural language expressions.

Methods in Computational Semantics generally fall in two categories:

- **Formal Semantics:** Construction of precise mathematical models of the relations between expressions in a natural language and the world.
  John chases a bat $\rightarrow \exists x[bat(x) \wedge chase(john, x)]$

Computational Semantics

Computational Semantics

The study of how to automate the process of constructing and reasoning with meaning representations of natural language expressions.

Methods in Computational Semantics generally fall in two categories:

- **Formal Semantics:** Construction of precise mathematical models of the relations between expressions in a natural language and the world.
  John chases a bat $\rightarrow \exists x[bat(x) \wedge chase(john, x)]$
- **Distributional Semantics:** The study of statistical patterns of human word usage to extract semantics.

Distributional Hypothesis

Distributional Hypothesis

Distributional Hypothesis: Basic Intuition

"The meaning of a word is its use in language." (Wittgenstein, 1953)

"You know a word by the company it keeps." (Firth, 1957)

Distributional Hypothesis

Distributional Hypothesis: Basic Intuition

> *"The meaning of a word is its use in language." (Wittgenstein, 1953)*

> *"You know a word by the company it keeps." (Firth, 1957)*

→ Word meaning (whatever it might be) is reflected in linguistic distributions.

Distributional Hypothesis

Distributional Hypothesis: Basic Intuition

"The meaning of a word is its use in language." (Wittgenstein, 1953)

"You know a word by the company it keeps." (Firth, 1957)

→ Word meaning (whatever it might be) is reflected in linguistic distributions.

"Words that occur in the same contexts tend to have similar meanings." (Zellig Harris, 1968)

Distributional Hypothesis

Distributional Hypothesis: Basic Intuition

> *"The meaning of a word is its use in language." (Wittgenstein, 1953)*

> *"You know a word by the company it keeps." (Firth, 1957)*

→ Word meaning (whatever it might be) is reflected in linguistic distributions.

> *"Words that occur in the same contexts tend to have similar meanings." (Zellig Harris, 1968)*

→ Semantically similar words tend to have similar distributional patterns.

Distributional Semantics: a linguistic perspective

"If linguistics is to deal with meaning, it can only do so through distributional analysis." (Zellig Harris)

Distributional Semantics: a linguistic perspective

"If linguistics is to deal with meaning, it can only do so through distributional analysis." (Zellig Harris)

"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, "Distributional Structure")

Distributional Semantics: a linguistic perspective

“If linguistics is to deal with meaning, it can only do so through distributional analysis.” (Zellig Harris)

“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, “Distributional Structure”)

Differential and not *referential*

Distributional Semantics: a cognitive perspective

Contextual representation

A word's contextual representation is an abstract cognitive structure that accumulates from encounters with the word in various linguistic contexts.

Distributional Semantics: a cognitive perspective

Contextual representation

A word's contextual representation is an abstract cognitive structure that accumulates from encounters with the word in various linguistic contexts.

We learn new words based on contextual cues

Distributional Semantics: a cognitive perspective

Contextual representation

A word's contextual representation is an abstract cognitive structure that accumulates from encounters with the word in various linguistic contexts.

We learn new words based on contextual cues

He filled the **wampimuk** with the substance, passed it around and we all drunk some.

Distributional Semantics: a cognitive perspective

Contextual representation

A word's contextual representation is an abstract cognitive structure that accumulates from encounters with the word in various linguistic contexts.

We learn new words based on contextual cues

He filled the **wampimuk** with the substance, passed it around and we all drunk some.

We found a little **wampimuk** sleeping behind the tree.

Distributional Semantic Models (DSMs)

- Computational models that build contextual semantic repesentations from corpus data

Distributional Semantic Models (DSMs)

- Computational models that build contextual semantic repesentations from corpus data
- DSMs are models for semantic representations
  - The semantic content is represented by a vector
  - Vectors are obtained through the statistical analysis of the linguistic contexts of a word

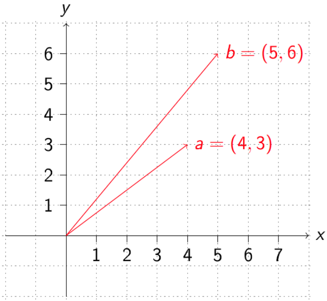
Distributional Semantic Models (DSMs)

- Computational models that build contextual semantic repesentations from corpus data
- DSMs are models for semantic representations
  - The semantic content is represented by a vector
  - Vectors are obtained through the statistical analysis of the linguistic contexts of a word
- Alternative names
  - corpus-based semantics
  - statistical semantics
  - geometrical models of meaning
  - vector semantics
  - word space models

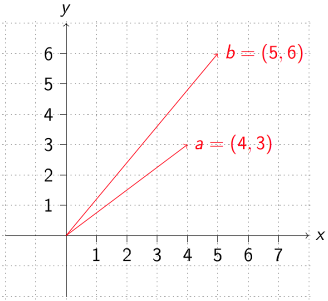
Distributional Semantics: The general intuition

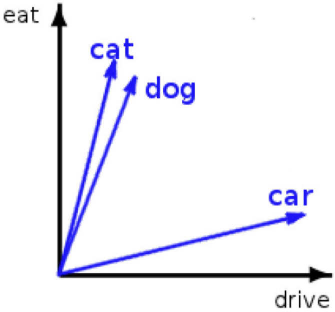
- **Distributions** are vectors in a multidimensional semantic space, that is, objects with a magnitude and a direction.
- The **semantic space** has dimensions which correspond to possible contexts, as gathered from a given corpus.

Vector Space

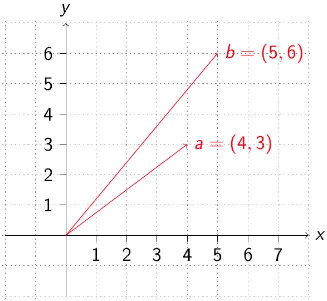


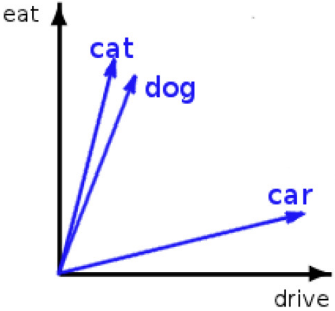
Vector Space





Vector Space





In practice, many more dimensions are used.
cat = [...dog 0.8, eat 0.7, joke 0.01, mansion 0.2,...]

Word Space

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⟩

Constructing Word spaces

Informal algorithm for constructing word spaces

- Pick the words you are interested in: **target words**

Constructing Word spaces

Informal algorithm for constructing word spaces

- Pick the words you are interested in: **target words**
- Define a **context window**, number of words surrounding target word

Constructing Word spaces

Informal algorithm for constructing word spaces

- Pick the words you are interested in: **target words**
- Define a **context window**, number of words surrounding target word
  - The context can in general be defined in terms of documents, paragraphs or sentences.

Constructing Word spaces

Informal algorithm for constructing word spaces

- Pick the words you are interested in: **target words**
- Define a **context window**, number of words surrounding target word
  - The context can in general be defined in terms of documents, paragraphs or sentences.
- Count number of times the target word co-occurs with the context words: **co-occurrence matrix**

Constructing Word spaces

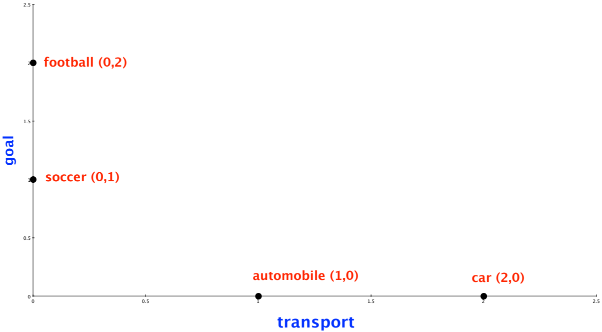
Informal algorithm for constructing word spaces

- Pick the words you are interested in: **target words**
- Define a **context window**, number of words surrounding target word
  - The context can in general be defined in terms of documents, paragraphs or sentences.
- Count number of times the target word co-occurs with the context words: **co-occurrence matrix**
- Build vectors out of (a function of) these co-occurrence counts

Constructing Word spaces: distributional vectors

distributional matrix = targets X contexts

	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


football (0,2)
soccer (0,1)
automobile (1,0)
car (2,0)
goal
transport
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5

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

Distributional Models of Semantics

Pawan Goyal

CSE, IIT Kharagpur

Week 7, Lecture 2

Vector Space Model without distributional similarity

Words are treated as atomic symbols

Vector Space Model without distributional similarity

Words are treated as atomic symbols

One-hot representation

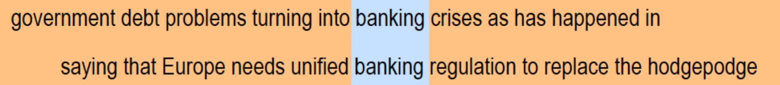
motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0] = 0

Distributional Similarity Based Representations

You know a word by the company it keeps

Distributional Similarity Based Representations

You know a word by the company it keeps

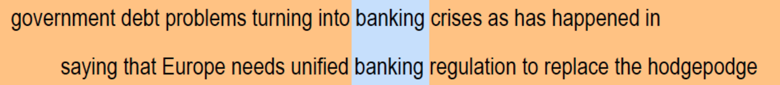


government debt problems turning into banking crises as has happened in

saying that Europe needs unified banking regulation to replace the hodgepodge

Distributional Similarity Based Representations

You know a word by the company it keeps



government debt problems turning into banking crises as has happened in

saying that Europe needs unified banking regulation to replace the hodgepodge

These words will represent banking

Building a DSM step-by-step

The "linguistic" steps

Pre-process a corpus (to define targets and contexts)

$\Downarrow$

Select the targets and the contexts

Building a DSM step-by-step

The "linguistic" steps

Pre-process a corpus (to define targets and contexts)

⇓

Select the targets and the contexts

The "mathematical" steps

Count the target-context co-occurrences

⇓

Weight the contexts (optional)

⇓

Build the distributional matrix

⇓

Reduce the matrix dimensions (optional)

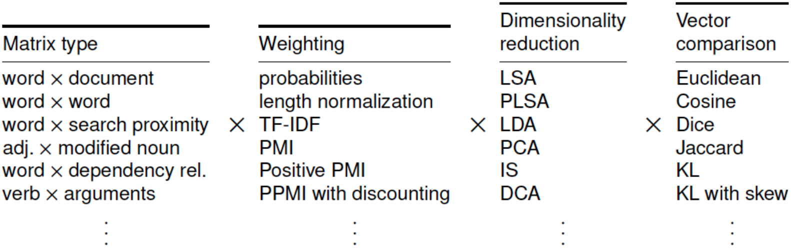
⇓

Compute the vector distances on the (reduced) matrix

Many design choices

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
⋮		⋮		⋮		⋮

Many design choices



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
⋮		⋮		⋮		⋮

General Questions

- How do the rows (words, ...) relate to each other?
- How do the columns (contexts, documents, ...) relate to each other?

The parameter space

A number of parameters to be fixed

- Which type of context?
- Which weighting scheme?
- Which similarity measure?
- ...

A specific parameter setting determines a particular type of DSM (e.g. LSA, HAL, etc.)

Documents as context: Word × document

	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: Word × Word

	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

Words as contexts

Parameters

- Window size
- Window shape - rectangular/triangular/other

Words as contexts

Parameters

- Window size
- Window shape - rectangular/triangular/other

Consider the following passage

Suspected communist rebels on 4 July 1989 killed Col. Herminio Taylo, police chief of Makati, the Philippines major financial center, in an escalation of street violence sweeping the Capitol area. The gunmen shouted references to the rebel New People's Army. They fled in a commandeered passenger jeep. The military says communist rebels have killed up to 65 soldiers and police in the Capitol region since January.

Words as contexts

Parameters

- Window size
- Window shape - rectangular/triangular/other

5 words window (unfiltered): 2 words either side of the target word

Suspected communist rebels on 4 July 1989 killed Col. Herminio Taylo, police chief of Makati, the Philippines major financial center, in an escalation of street violence sweeping the Capitol area. The gunmen shouted references to the rebel New People's Army. They fled in a commandeered passenger jeep. The military says communist rebels have killed up to 65 soldiers and police in the Capitol region since January.

Words as contexts

Parameters

- Window size
- Window shape - rectangular/triangular/other

5 words window (filtered): 2 words either side of the target word

Suspected communist rebels on 4 July 1989 killed Col. Herminio Taylo, police chief of Makati, the Philippines major financial center, in an escalation of street violence sweeping the Capitol area. The gunmen shouted references to the rebel New People's Army. They fled in a commandeered passenger jeep. The military says communist rebels have killed up to 65 soldiers and police in the Capitol region since January.

Context weighting: documents as context

Context weighting: documents as context

Indexing function F: Essential factors

- **Word frequency (f_{ij}):** How many times a word appears in the document? $F \propto f_{ij}$
- **Document length ($|D_i|$):** How many words appear in the document? $F \propto \frac{1}{|D_i|}$
- **Document frequency (N_j):** Number of documents in which a word appears. $F \propto \frac{1}{N_j}$

Context weighting: documents as context

Indexing function F: Essential factors

- **Word frequency (f_{ij}):** How many times a word appears in the document? $F \propto f_{ij}$
- **Document length ($|D_i|$):** How many words appear in the document? $F \propto \frac{1}{|D_i|}$
- **Document frequency (N_j):** Number of documents in which a word appears. $F \propto \frac{1}{N_j}$

Indexing Weight: tf-Idf

- $f_{ij} * log(\frac{N}{N_j})$ for each term, normalize the weight in a document with respect to L_2-norm.

Context weighting: words as context

Context weighting: words as context

basic intuition

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

Context weighting: words as context

basic intuition

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

Association measures are used to give more weight to contexts that are more significantly associted with a targer word.

Context weighting: words as context

basic intuition

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

Association measures are used to give more weight to contexts that are more significantly associted with a targer word.

- The less frequent the target and context element are, the higher the weight given to their co-occurrence count should be.

Context weighting: words as context

basic intuition

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

Association measures are used to give more weight to contexts that are more significantly associted with a targer 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 rarer *domesticated*.

Context weighting: words as context

basic intuition

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

Association measures are used to give more weight to contexts that are more significantly associted with a targer word.

- The less frequent the target and context element are, the higher the weight given to their co-occurrence count should be.
  $\Rightarrow$ Co-occurrence with frequent context element *small* is less informative than co-occurrence with rarer *domesticated*.
- different measures - e.g., Mutual information, Log-likelihood ratio

Pointwise Mutual Information (PMI)

$$PMI(w_1, w_2) = log_2 \frac{P_{corpus}(w_1, w_2)}{P_{ind}(w_1, w_2)}$$

Pointwise Mutual Information (PMI)

$$PMI(w_1, w_2) = log_2 \frac{P_{corpus}(w_1, w_2)}{P_{ind}(w_1, w_2)}$$

$$PMI(w_1, w_2) = log_2 \frac{P_{corpus}(w_1, w_2)}{P_{corpus}(w_1) P_{corpus}(w_2)}$$

Pointwise Mutual Information (PMI)

$$PMI(w_1, w_2) = log_2 \frac{P_{corpus}(w_1, w_2)}{P_{ind}(w_1, w_2)}$$

$$PMI(w_1, w_2) = log_2 \frac{P_{corpus}(w_1, w_2)}{P_{corpus}(w_1) P_{corpus}(w_2)}$$

$$P_{corpus}(w_1, w_2) = \frac{freq(w_1, w_2)}{N}$$

$$P_{corpus}(w) = \frac{freq(w)}{N}$$

PMI: Issues and Variations

Positive PMI

All PMI values less than zero are replaced with zero.

PMI: Issues and Variations

Positive PMI

All PMI values less than zero are replaced with zero.

Bias towards infrequent events

Consider w_j having the maximum association with w_i,
$P_{corpus}(w_i) \approx P_{corpus}(w_j) \approx P_{corpus}(w_i, w_j)$

PMI: Issues and Variations

Positive PMI

All PMI values less than zero are replaced with zero.

Bias towards infrequent events

Consider w_j having the maximum association with w_i,
$P_{corpus}(w_i) \approx P_{corpus}(w_j) \approx P_{corpus}(w_i, w_j)$
PMI increases as the probability of w_i decreases.

PMI: Issues and Variations

Positive PMI

All PMI values less than zero are replaced with zero.

Bias towards infrequent events

Consider w_j having the maximum association with w_i,
$P_{corpus}(w_i) \approx P_{corpus}(w_j) \approx P_{corpus}(w_i, w_j)$
PMI increases as the probability of w_i decreases.
Also, consider a word w_j that occurs once in the corpus, also in the context of w_i.

PMI: Issues and Variations

Positive PMI

All PMI values less than zero are replaced with zero.

Bias towards infrequent events

Consider w_j having the maximum association with w_i,
$P_{corpus}(w_i) \approx P_{corpus}(w_j) \approx P_{corpus}(w_i, w_j)$
PMI increases as the probability of w_i decreases.
Also, consider a word w_j that occurs once in the corpus, also in the context of w_i. A discounting factor proposed by Pantel and Lin:

$$\delta_{ij} = \frac{f_{ij}}{f_{ij}+1} \frac{min(f_i, f_j)}{min(f_i, f_j)+1}$$

$PMI_{new}(w_i, w_j) = \delta_{ij} PMI(w_i, w_j)$

Distributional Vectors: Example

Normalized Distributional Vectors using Pointwise Mutual Information

petroleum	oil:0.032 gas:0.029 crude:0.029 barrels:0.028 exploration:0.027 barrel:0.026 opec:0.026 refining:0.026 gasoline:0.026 fuel:0.025 natural:0.025 exporting:0.025
drug	trafficking:0.029 cocaine:0.028 narcotics:0.027 fda:0.026 police:0.026 abuse:0.026 marijuana:0.025 crime:0.025 colombian:0.025 arrested:0.025 addicts:0.024
insurance	insurers:0.028 premiums:0.028 lloyds:0.026 reinsurance:0.026 underwriting:0.025 pension:0.025 mortgage:0.025 credit:0.025 investors:0.024 claims:0.024 benefits:0.024
forest	timber:0.028 trees:0.027 land:0.027 forestry:0.026 environmental:0.026 species:0.026 wildlife:0.026 habitat:0.025 tree:0.025 mountain:0.025 river:0.025 lake:0.025
robotics	robots:0.032 automation:0.029 technology:0.028 engineering:0.026 systems:0.026 sensors:0.025 welding:0.025 computer:0.025 manufacturing:0.025 automated:0.025

Distributional Semantics: Applications, Structured Models

Pawan Goyal

CSE, IIT Kharagpur

Week 7, Lecture 3

Application to Query Expansion: Addressing Term Mismatch

Term Mismatch Problem in Information Retrieval

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

Application to Query Expansion: Addressing Term Mismatch

Term Mismatch Problem in Information Retrieval

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

Using DSMs for Query Expansion

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

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

Query Expansion using Unstructured DSMs

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

Query Expansion using Unstructured DSMs

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)

Query Expansion using Unstructured DSMs

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

Query Expansion using Unstructured DSMs

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 for Binary Vectors

Let X and Y denote the binary distributional vectors for words X and Y.

Similarity Measures

Dice coefficient : $\frac{2|X \cap Y|}{|X|+|Y|}$

Similarity Measures for Binary Vectors

Let X and Y denote the binary distributional vectors for words X and Y.

Similarity Measures

Dice coefficient : $\frac{2|X \cap Y|}{|X|+|Y|}$

Jaccard Coefficient : $\frac{|X \cap Y|}{|X \cup Y|}$

Similarity Measures for Binary Vectors

Let X and Y denote the binary distributional vectors for words X and Y.

Similarity Measures

Dice coefficient : $\frac{2|X \cap Y|}{|X|+|Y|}$

Jaccard Coefficient : $\frac{|X \cap Y|}{|X \cup Y|}$

Overlap Coefficient : $\frac{|X \cap Y|}{min(|X|,|Y|)}$

Similarity Measures for Binary Vectors

Let X and Y denote the binary distributional vectors for words X and Y.

Similarity Measures

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 entries, while Overlap coefficient uses the concept of inclusion.

Similarity Measures for Vector Spaces

Let $\vec{X}$ and $\vec{Y}$ denote the distributional vectors for words X and Y.
$\vec{X} = [x_1, x_2, \ldots, x_n]$, $\vec{Y} = [y_1, y_2, \ldots, y_n]$

Similarity Measures for Vector Spaces

Let $\vec{X}$ and $\vec{Y}$ denote the distributional vectors for words X and Y.
$\vec{X} = [x_1, x_2, \ldots, x_n]$, $\vec{Y} = [y_1, y_2, \ldots, y_n]$

Similarity Measures

Cosine similarity : $cos(\vec{X}, \vec{Y}) = \frac{\bar{X} \cdot \vec{Y}}{|\vec{X}||\vec{Y}|}$

Similarity Measures for Vector Spaces

Let $\vec{X}$ and $\vec{Y}$ denote the distributional vectors for words X and Y.
$\vec{X} = [x_1, x_2, \ldots, x_n]$, $\vec{Y} = [y_1, y_2, \ldots, y_n]$

Similarity Measures

Cosine similarity : $cos(\vec{X}, \vec{Y}) = \frac{\bar{X} \cdot \vec{Y}}{|\vec{X}||\vec{Y}|}$

Euclidean distance : $|\vec{X} - \vec{Y}| = \sqrt{\Sigma_{i=1}^{n} (x_i - y_i)^2}$

Similarity Measure for Probability Distributions

Let p and q denote the probability distributions corresponding to two distributional vectors.

Similarity Measure for Probability Distributions

Let p and q denote the probability distributions corresponding to two distributional vectors.

Similarity Measures

KL-divergence : $D(p||q) = \Sigma_i p_i log \frac{p_i}{q_i}$

Information Radius : $D(p||\frac{p+q}{2}) + D(q||\frac{p+q}{2})$

L_1-norm : $\Sigma_i |p_i - q_i|$

Attributional Similarity vs. Relational Similarity

Attributional Similarity

The attributional similarity between two words a and b depends on the degree of correspondence between the properties of a and b.
Ex: dog and wolf

Relational Similarity

Two pairs(a,b) and (c,d) are relationally similar if they have many similar relations.
Ex: dog: bark and cat: meow

Relational Similarity: Pair-pattern matrix

Pair-pattern matrix

- Row vectors correspond to pairs of words, such as *mason: stone* and *carpenter: wood*
- Column vectors correspond 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

Relational Similarity: Pair-pattern matrix

Pair-pattern matrix

- Row vectors correspond to pairs of words, such as *mason: stone* and *carpenter: wood*
- Column vectors correspond 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.

Relational Similarity: Pair-pattern matrix

Pair-pattern matrix

- Row vectors correspond to pairs of words, such as *mason: stone* and *carpenter: wood*
- Column vectors correspond 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.

Relational Similarity: Pair-pattern matrix

Pair-pattern matrix

- Row vectors correspond to pairs of words, such as *mason: stone* and *carpenter: wood*
- Column vectors correspond 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 DSMs

Basic Issue

- Words may not be the basic context units anymore
- How to capture and represent syntactic information?
  X solves Y and *Y is solved by X*

Structured DSMs

Basic Issue

- Words may not be the basic context units anymore
- How to capture and represent syntactic information?
  X solves Y and *Y is solved by X*

An Ideal Formalism

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

Structured DSMs

Basic Issue

- Words may not be the basic context units anymore
- How to capture and represent syntactic information?
  X solves Y and *Y is solved by X*

An Ideal Formalism

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

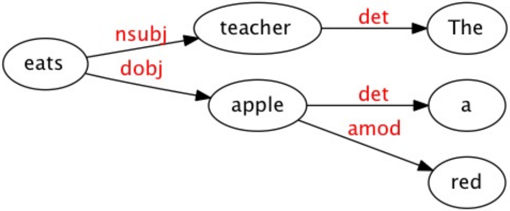
Use Dependency grammar framework

Structured DSMs

Structured DSMs

Using Dependency Structure: How does it help?

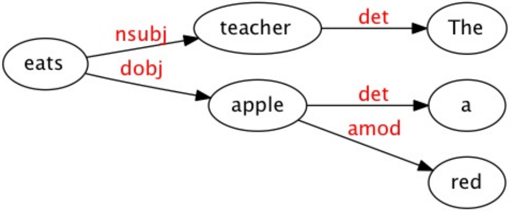
The teacher eats a red apple.



Structured DSMs

Using Dependency Structure: How does it help?

The teacher eats a red apple.



- 'eat' is not a legitimate context for 'red'.
- The 'object' relation connecting 'eat' and 'apple' is treated as a different type of co-occurrence from the 'modifier' relation linking 'red' and 'apple'.

Structured DSMs

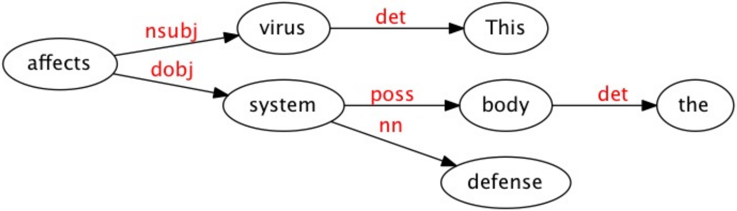
Structured DSMs: Words as ‘legitimate’ contexts

- Co-occurrence statistics are collected using parser-extracted relations.
- To qualify as context of a target item, a word must be linked to it by some (interesting) lexico-syntactic relation

Structured DSMs

Distributional models, as guided by dependency

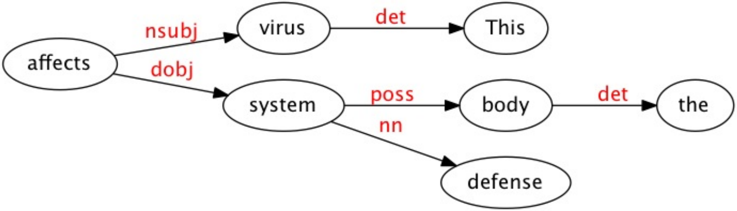
Ex: For the sentence '*This virus affects the body's defense system.*', the dependency parse is:



Structured DSMs

Distributional models, as guided by dependency

Ex: For the sentence '*This virus affects the body's defense system.*', the dependency parse is:



Word vectors

<system, dobj, affects> ...

Corpus-derived ternary data can also be mapped onto a 2-way matrix

2-way matrix

<system, dobj, affects>
<virus, nsubj, affects>

The dependency information can be dropped

- <system, dobj, affects> ⇒ <system, affects>
- <virus, nsubj, affects> ⇒ <virus, affects>

2-way matrix

<system, dobj, affects>
<virus, nsubj, affects>

The dependency information can be dropped

- <system, dobj, affects> ⇒ <system, affects>
- <virus, nsubj, affects> ⇒ <virus, affects>

Link and one word can be concatenated and treated as attributes

- *virus*={nsubj-affects:0.05,...},
- *system*={dobj-affects:0.03,...}

Structured DSMs for Selectional Preferences

Selectional Preferences for Verbs

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

Structured DSMs for Selectional Preferences

Selectional Preferences for Verbs

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

- From a parsed corpus, noun vectors are calculated as shown for ‘virus’ and ‘system’.

Structured DSMs for Selectional Preferences

Selectional Preferences for Verbs

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

- From a parsed corpus, noun vectors are calculated as shown for ‘virus’ and ‘system’.

	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	...
...	...	...	...	...	...	...	...

Structured DSMs for Selectional Preferences

Selectional Preferences

- Suppose we want to compute the selectional preferences of the nouns as object of verb ‘eat’.

Structured DSMs for Selectional Preferences

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,$\ldots\}$ be the set of these n nouns.

Structured DSMs for Selectional Preferences

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,$\ldots\}$ be the set of these n nouns.
- The complete vectors of these n nouns are used to obtain an ‘object prototype’ of the verb.

Structured DSMs for Selectional Preferences

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,$\ldots\}$ 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.

Structured DSMs for Selectional Preferences

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,$\ldots\}$ 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 Embeddings - Part I

Pawan Goyal

CSE, IIT Kharagpur

Week 7, Lecture 4

Word Vectors

- At one level, it is simply a vector of weights.

Word Vectors

- At one level, it is simply a vector of weights.
- In a simple 1-of-N (or ‘one-hot’) encoding every element in the vector is associated with a word in the vocabulary.

Word Vectors

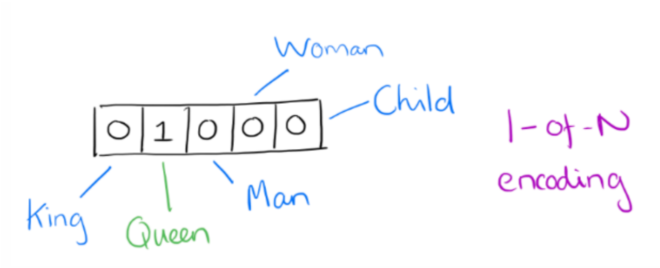
- At one level, it is simply a vector of weights.
- In a simple 1-of-N (or ‘one-hot’) encoding every element in the vector is associated with a word in the vocabulary.
- The encoding of a given word is simply the vector in which the corresponding element is set to one, and all other elements are zero.

One-hot representation

motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0] = 0

Word Vectors - One-hot Encoding

- Suppose our vocabulary has only five words: King, Queen, Man, Woman, and Child.
- We could encode the word ‘Queen’ as:



Limitations of One-hot encoding

Limitations of One-hot encoding

Word vectors are not comparable

Using such an encoding, there is no meaningful comparison we can make between word vectors other than equality testing.

Word2Vec – A distributed representation

Distributional representation – word embedding?

Any word w_i in the corpus is given a distributional representation by an embedding

$$w_i \in R^d$$

i.e., a $d-$dimensional vector, which is mostly learnt!

Word2Vec – A distributed representation

Distributional representation – word embedding?

Any word w_i in the corpus is given a distributional representation by an embedding

$$w_i \in R^d$$

i.e., a $d-$dimensional vector, which is mostly learnt!

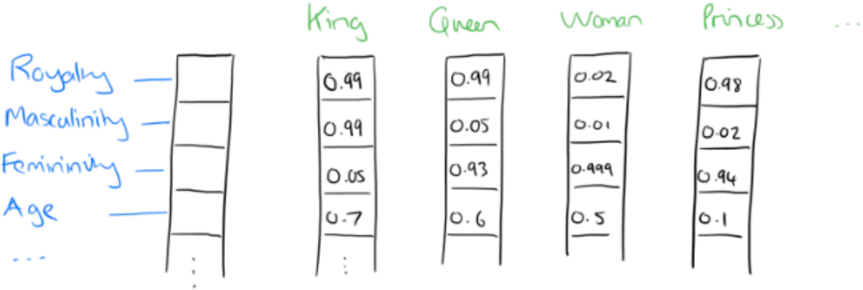
$$\textit{linguistics} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

Distributional Representation

- Take a vector with several hundred dimensions (say 1000).
- Each word is represented by a distribution of weights across those elements.
- So instead of a 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 dimensions in a hypothetical word vector (there are no such pre-assigned labels in the algorithm of course), it might look a bit like this:



	King	Queen	Woman	Princess	...
Royalty	0.99	0.99	0.02	0.98	
Masculinity	0.99	0.05	0.01	0.02	
Femininity	0.05	0.93	0.999	0.94	
Age	0.7	0.6	0.5	0.1	
...	⋮				

Such a vector comes to represent in some abstract way the 'meaning' of a word

Word Embeddings

- d typically in the range 50 to 1000
- Similar words should have similar embeddings

Word Embeddings

- d typically in the range 50 to 1000
- Similar words should have similar embeddings

SVD can also be thought of as an embedding method

Reasoning with Word Vectors

- It has been found that the learned word representations in fact capture meaningful syntactic and semantic regularities in a very simple way.

Reasoning with Word Vectors

- It has been found that the learned word representations in fact capture meaningful syntactic and semantic regularities in a very simple way.
- Specifically, the regularities are observed as constant vector offsets between pairs of words sharing a particular relationship.

Reasoning with Word Vectors

- It has been found that the learned word representations in fact capture meaningful syntactic and semantic regularities in a very simple way.
- Specifically, the regularities are observed as constant vector offsets between pairs of words sharing a particular relationship.

Case of Singular-Plural Relations

If we denote the vector for word i as x_i, and focus on the singular/plural relation, we observe that

Reasoning with Word Vectors

- It has been found that the learned word representations in fact capture meaningful syntactic and semantic regularities in a very simple way.
- Specifically, the regularities are observed as constant vector offsets between pairs of words sharing a particular relationship.

Case of Singular-Plural Relations

If we denote the vector for word i as x_i, and focus on the singular/plural relation, we observe that

$$x_{apple} - x_{apples} \approx x_{car} - x_{cars} \approx x_{family} - x_{families} \approx x_{car} - x_{cars}$$

and so on.

Reasoning with Word Vectors

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*)

Reasoning with Word Vectors

Perhaps more surprisingly, we find that this is also the case for a variety of semantic relations.

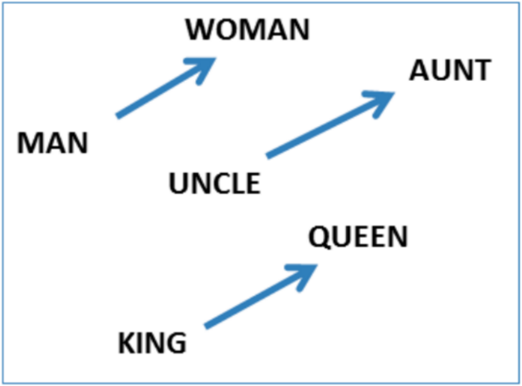
Good at answering analogy questions

a is to b, as c is to ?

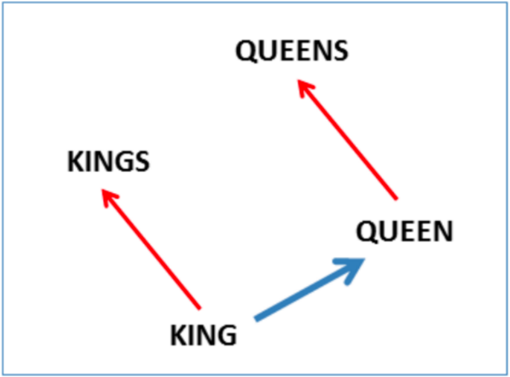
man is to *woman* as *uncle* is to ? (*aunt*)

A simple vector offset method based on cosine distance shows the relation.

Vcctor Offset for Gender Relation



Vcctor Offset for Singular-Plural Relation



Encoding Other Dimensions of Similarity

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

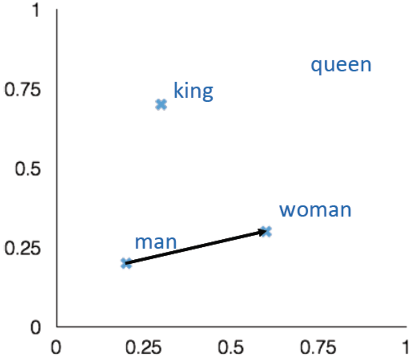
Analogy Testing

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 city relationships

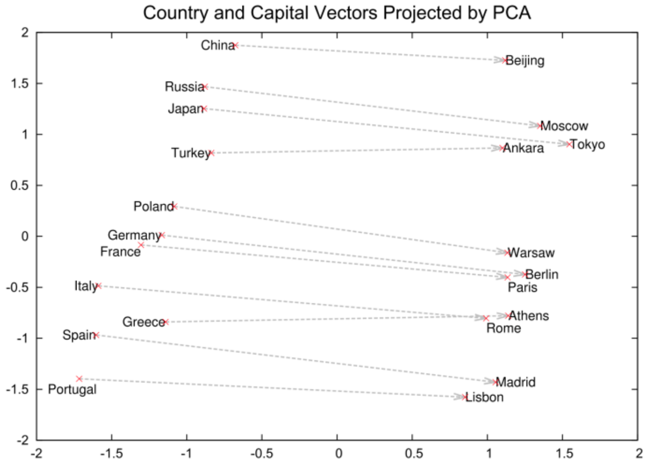
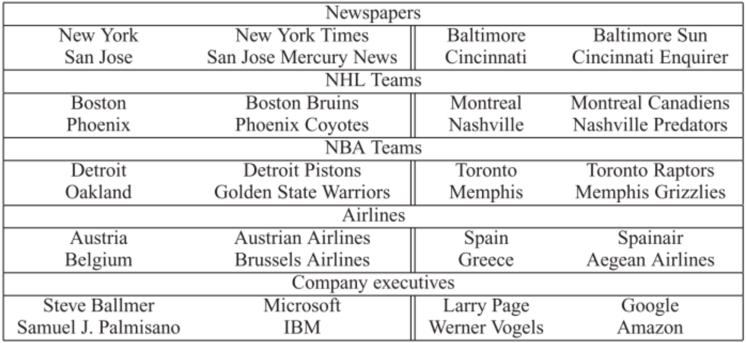


Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised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

More Analogy Questions

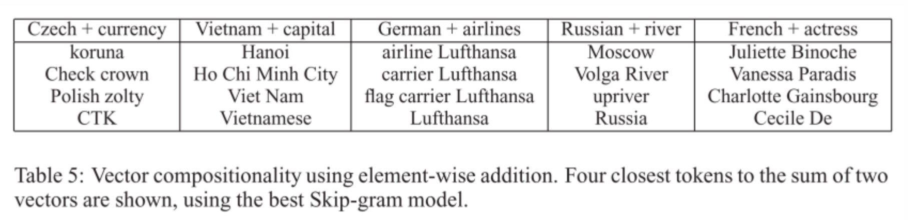


Newspapers			
New York	New York Times	Baltimore	Baltimore Sun
San Jose	San Jose Mercury News	Cincinnati	Cincinnati Enquirer
NHL Teams			
Boston	Boston Bruins	Montreal	Montreal Canadiens
Phoenix	Phoenix Coyotes	Nashville	Nashville Predators
NBA Teams			
Detroit	Detroit Pistons	Toronto	Toronto Raptors
Oakland	Golden State Warriors	Memphis	Memphis Grizzlies
Airlines			
Austria	Austrian Airlines	Spain	Spainair
Belgium	Brussels Airlines	Greece	Aegean Airlines
Company executives			
Steve Ballmer	Microsoft	Larry Page	Google
Samuel J. Palmisano	IBM	Werner Vogels	Amazon

Table 2: Examples of the analogical reasoning task for phrases (the full test set has 3218 examples). The goal is to compute the fourth phrase using the first three. Our best model achieved an accuracy of 72% on this dataset.

Element Wise Addition

We can also use element-wise addition of vector elements to ask questions such as ‘German + airlines’ and by looking at the closest tokens 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.

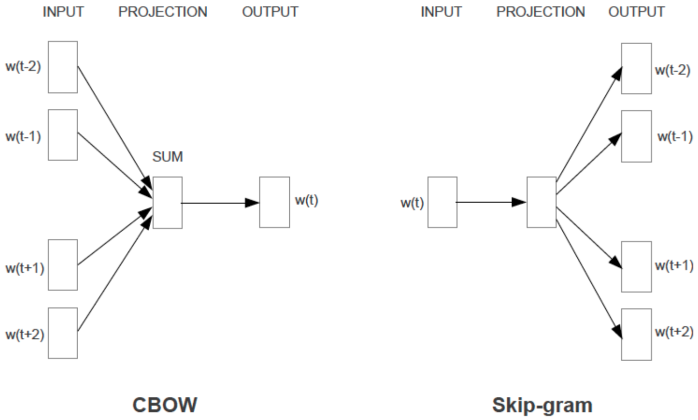
Learning Word Vectors

Basic Idea

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

Code as well as word-vectors: *https://code.google.com/p/word2vec/*

Two Variations: CBOW and Skip-grams



Word Embedings - Part II

Pawan Goyal

CSE, IIT Kharagpur

Week 7, Lecture 5

CBOW

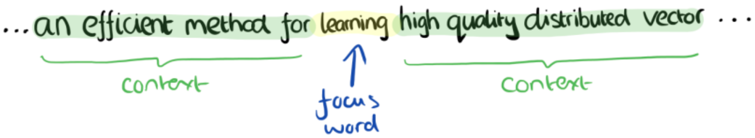
- Consider a piece of prose such as:
  "The recently introduced continuous Skip-gram model is an efficient method for learning high-quality distributed vector representations that capture a large number of syntatic and semantic word relationships."

CBOW

- Consider a piece of prose such as:
  “The recently introduced continuous Skip-gram model is an efficient method for learning high-quality distributed vector representations that capture a large number of syntatic and semantic word relationships.”
- Imagine a sliding window over the text, that includes the central word currently in focus, together with the four words that precede it, and the four words that follow it:

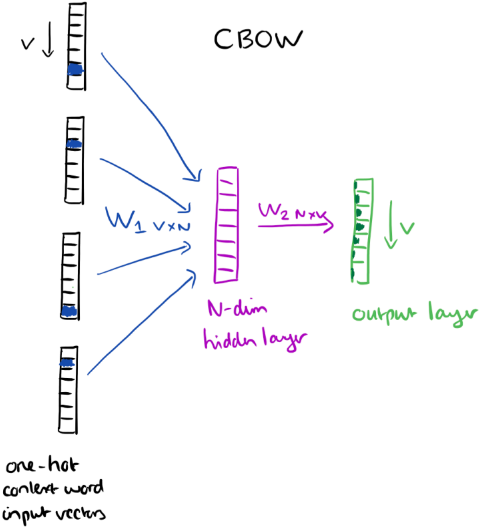
CBOW

- Consider a piece of prose such as:
  "The recently introduced continuous Skip-gram model is an efficient method for learning high-quality distributed vector representations that capture a large number of syntatic and semantic word relationships."
- Imagine a sliding window over the text, that includes the central word currently in focus, together with the four words that precede it, and the four words that follow it:


...an efficient method for learning high quality distributed vector ...
context
focus word
context

CBOW

The context words form the input layer. Each word is encoded in one-hot form. A single hidden and output layer.



CBOW: Training Objective

- The training objective is to maximize the conditional probability of observing the actual output word (the focus word) given the input context words, with regard to the weights.
- In our example, given the input (“an”, “efficient”, “method”, “for”, “high”, “quality”, “distributed”, “vector”), we want to maximize the probability of getting “learning” as the output.

CBOW: Input to Hidden Layer

Since our input vectors are one-hot, multiplying an input vector by the weight matrix W_1 amounts to simply selecting a row from W_1.

$$\overset{\substack{\text{input}\\1\times V}}{\begin{bmatrix}0&1&0\end{bmatrix}}\underset{W_1}{\overset{\substack{W_1\\V\times N}}{\begin{bmatrix}a&b&c&d\\e&f&g&h\\i&j&k&l\end{bmatrix}}}=\overset{\substack{\text{hidden}\\1\times N}}{\begin{bmatrix}e&f&g&h\end{bmatrix}}$$

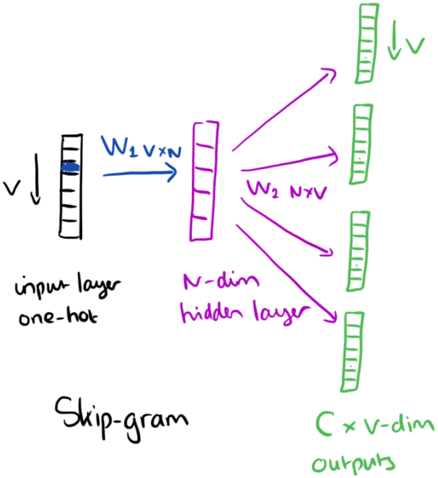
Given C input word vectors, the activation function for the hidden layer h amounts to simply summing the corresponding ‘hot’ rows in W_1, and dividing by C to take their average.

CBOW: Hidden to Output Layer

From the hidden layer to the output layer, the second weight matrix W_2 can be used to compute a score for each word in the vocabulary, and softmax can be used to obtain the posterior distribution of words.

Skip-gram Model

The skip-gram model is the opposite of the CBOW model. It is constructed with the focus word as the single input vector, and the target context words are now at the output layer:



Skip-gram Model: Training

- The activation function for the hidden layer simply amounts to copying the corresponding row from the weights matrix W_1 (linear) as we saw before.
- At the output layer, we now output C multinomial distributions instead of just one.
- The training objective is to mimimize the summed prediction error across all context words in the output layer. In our example, the input would be “learning”, and we hope to see (“an”, “efficient”, “method”, “for”, “high”, “quality”, “distributed”, “vector”) at the output layer.

Skip-gram Model

Details

Predict surrounding words in a window of length c of each word

Skip-gram Model

Details

Predict surrounding words in a window of length c of each word

Objective Function: Maximize the log probablility of any context word given the current center word:

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} log\ p(w_{t+j}|w_t)$$

Word Vectors

For $p(w_{t+j}|w_t)$ the simplest first formulation is

$$p(w_O|w_I) = \frac{exp({v'_{wO}}^T v_{WI})}{\sum_{w=1}^{W} exp({v'_w}^T v_{WI})}$$

Word Vectors

For $p(w_{t+j}|w_t)$ the simplest first formulation is

$$p(w_O|w_I) = \frac{exp({v'_{wO}}^T v_{WI})}{\sum_{w=1}^{W} exp({v'_w}^T v_{WI})}$$

where v and v' are “input” and “output” vector representations of w (so every word has two vectors)

Parameters θ

With $d-$dimensional words and V many words:

$$\theta = \begin{bmatrix} v_{aardvark} \\ v_a \\ \vdots \\ v_{zebra} \\ v'_{aardvark} \\ v'_a \\ \vdots \\ v'_{zebra} \end{bmatrix} \in \mathbb{R}^{2dV}$$

Gradient Descent for Parameter Updates

$$\theta_j^{new} = \theta_j^{old} - \alpha \frac{\partial}{\partial \theta_j^{old}} J(\theta)$$

Two sets of vectors

Best solution is to sum these up

$$L_{final} = L + L'$$

Two sets of vectors

Best solution is to sum these up

$$L_{final} = L + L'$$

A good tutorial to understand parameter learning:

https://arxiv.org/pdf/1411.2738.pdf

Two sets of vectors

Best solution is to sum these up

$$L_{final} = L + L'$$

A good tutorial to understand parameter learning:

https://arxiv.org/pdf/1411.2738.pdf

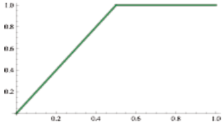
An interactive Demo

https://ronxin.github.io/wevi/

Glove

$$J = \frac{1}{2}\sum_{ij} f(P_{ij})\left(w_i \cdot \tilde{w}_j - \log P_{ij}\right)^2$$

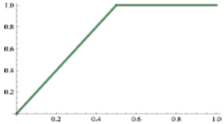
$f \sim$



Combine the best of both worlds – count based methods as well as direct prediction methods

Glove

$$J = \frac{1}{2} \sum_{ij} f(P_{ij}) \left(w_i \cdot \tilde{w}_j - \log P_{ij} \right)^2 \qquad f \sim$$



Combine the best of both worlds – count based methods as well as direct prediction methods

- Fast training
- Scalable to huge corpora
- Good performance even with small corpus, and small vectors

Code and vectors: http://nlp.stanford.edu/projects/glove/