Qual Presentation

Daniel Khashabi

Outline

- My own line of research
- Papers:
  - Fast Dropout training, ICML, 2013
  - Distributional Semantics Beyond Words: Supervised Learning of Analogy and Paraphrase, TACL, 2013.

Outline

- My own line of research
- Papers:
  - Fast Dropout training, ICML, 2013
  - Distributional Semantics Beyond Words: Supervised Learning of Analogy and Paraphrase, TACL, 2013.

Motivation

- Developing tools for word-similarity
- Useful in many applications

Motivation

- Developing tools for word-similarity
- Useful in many applications
  - For example paraphrase detection:

Motivation

- Developing tools for word-similarity
- Useful in many applications
  - For example paraphrase detection:

The Iraqi foreign minister warned of disastrous consequences if Turkey launched an invasion.

Motivation

- Developing tools for word-similarity
- Useful in many applications
  - For example paraphrase detection:

The Iraqi foreign minister warned of disastrous consequences if Turkey launched an invasion.

Iraq has warned that a Turkish incursion would have disastrous results.

Motivation

- Developing tools for word-similarity
- Useful in many applications
  - For example paraphrase detection:

The Iraqi foreign minister warned of disastrous consequences if Turkey launched an invasion.

Iraq has warned that a Turkish incursion would have disastrous results.

Motivation

- Developing tools for word-similarity
- Useful in many applications
  - For example paraphrase detection:

The Iraqi foreign minister warned of disastrous consequences if Turkey launched an invasion.

Iraq has warned that a Turkish incursion would have disastrous results.

I can be there on time.

Motivation

- Developing tools for word-similarity
- Useful in many applications
  - For example paraphrase detection:

The Iraqi foreign minister warned of disastrous consequences if Turkey launched an invasion.

Iraq has warned that a Turkish incursion would have disastrous results.

✓

I can be there on time.

I can't be there on time.

Motivation

- Developing tools for word-similarity
- Useful in many applications
  - For example paraphrase detection:

The Iraqi foreign minister warned of disastrous consequences if Turkey launched an invasion.
Iraq has warned that a Turkish incursion would have disastrous results.

I can be there on time.
I can't be there on time.

Motivation (2)

- Developing tools for word-similarity

Motivation (2)

- Developing tools for word-similarity
- We need to solve easier problem

Motivation (2)

- Developing tools for word-similarity
- We need to solve easier problem
  - For example, SAT test:

Stem:		word:language
Choices:	(1)	paint:portrait
	(2)	poetry:rhythm
	(3)	note:music
	(4)	tale:story
	(5)	week:year
Solution:	(3)	note:music

  - Very important for understanding hierarchies of word semantics

Motivation (3)

- Developing tools for word-similarity
- We need to solve easier problem
  - ↓

Motivation (3)

- Developing tools for word-similarity
- We need to solve easier problem
  - Compositional behavior of the word semantics

Motivation (3)

- Developing tools for word-similarity
- We need to solve easier problem
  - Compositional behavior of the word semantics

Search

Motivation (3)

- Developing tools for word-similarity
- We need to solve easier problem
  - Compositional behavior of the word semantics

Search

Motivation (3)

- Developing tools for word-similarity
- We need to solve easier problem
  - Compositional behavior of the word semantics

Search   Engine

Motivation (3)

- Developing tools for word-similarity
- We need to solve easier problem
  - Compositional behavior of the word semantics

Search

Engine

Motivation (3)

- Developing tools for word-similarity
- We need to solve easier problem
  - Compositional behavior of the word semantics

Search + Engine =

Motivation (3)

- Developing tools for word-similarity
- We need to solve easier problem
  - Compositional behavior of the word semantics

Search + Engine = Search Engine

Google

YAHOO!

bing

Motivation (4)

- Developing tools for word-similarity
- We need to solve easier problem
  - Compositional behavior of the word semantics
  - ↓

Motivation (4)

- Developing tools for word-similarity
- We need to solve easier problem
  - Compositional behavior of the word semantics
  - For example: understanding noun-modifier questions

Stem:		fantasy world
Choices:	(1)	fairyland
	(2)	fantasy
	(3)	world
	(4)	phantasy
	(5)	universe
	(6)	ranter
	(7)	souring
Solution:	(1)	fairyland

Designing semantic features

- Feature engineering
  - An important step in semantic modeling of words

Designing semantic features

- Feature engineering
  - An important step in semantic modeling of words
- The rest is just learning the task in a fully supervised fashion

Designing semantic features

- Feature engineering
  - An important step in semantic modeling of words
- The rest is just learning the task in a fully supervised fashion
- Type of the features:
  - Log-Frequency $$LF(x_i) = \log(freq(x_i) + 1)$$

Designing semantic features

- Feature engineering
  - An important step in semantic modeling of words
- The rest is just learning the task in a fully supervised fashion
- Type of the features:
  - Log-Frequency $$LF(x_i) = \log(freq(x_i) + 1)$$
  - PPMI (Positive Pointwise Mutual Information)
  -
  -

Designing semantic features

- Feature engineering
  - An important step in semantic modeling of words
- The rest is just learning the task in a fully supervised fashion
- Type of the features:
  - Log-Frequency $$LF(x_i) = \log(freq(x_i) + 1)$$
  - PPMI (Positive Pointwise Mutual Information)
  - Semantic Similarity
  - Functional Similarity

Feature generation

Feature generation

- Features
  - Log-Frequency
  - PPMI (Positive Pointwise Mutual Information)
  - Semantic Similarity
  - Functional Similarity

Feature generation

- Features
  - Log-Frequency
  - PPMI (Positive Pointwise Mutual Information)
  - Semantic Similarity
  - Functional Similarity
- All features are generated on a collection of documents
  - Of size 5×10^{10} words

Feature generation

- Features
  - Log-Frequency
  - PPMI (Positive Pointwise Mutual Information)
  - Semantic Similarity
  - Functional Similarity
- All features are generated on a collection of documents
  - Of size 5×10^{10} words
- Definition
  - Word-Context:

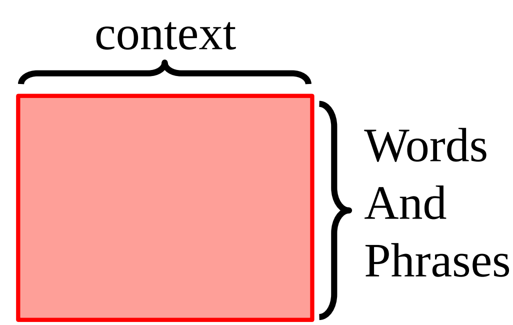
(Left) Context | Word | (Right) Context

Feature generation

- Features
  - Log-Frequency
  - PPMI (Positive Pointwise Mutual Information)
  - Semantic Similarity
  - Functional Similarity
- All features are generated on a collection of documents
  - Of size 5×10^{10} words
- Definition
  - Word-Context:

(Left) Context | Word | (Right) Context

- Three word-context matrices
  - Rows correspond to words/phrases in Wordnet



Pointwise Mutual Information

Pointwise Mutual Information

- PMI (Pointwise Mutual Information):

$$PMI(a,b) = \log \frac{p(a,b)}{p(a)p(b)}$$

Pointwise Mutual Information

- PMI (Pointwise Mutual Information):

$$PMI(a,b) = \log \frac{p(a,b)}{p(a)\,p(b)}$$

- PPMI(Positive PPMI)

$$PPMI(a,b) = \max\left(0, PMI(a,b)\right)$$

Pointwise Mutual Information

- PMI (Pointwise Mutual Information):

$$PMI(a,b) = \log \frac{p(a,b)}{p(a)\,p(b)}$$

- PPMI(Positive PPMI)

$$PPMI(a,b) = \max\left(0, PMI(a,b)\right)$$

- One useful definition for probabilities
  - The ratio of the times a context appears with a words

Pointwise Mutual Information (2)

Pointwise Mutual Information (2)

- Only the words or phrases that exist in the Wordnet

Pointwise Mutual Information (2)

- Only the words or phrases that exist in the Wordnet
- And appear with frequency more than 100 in the corpus

Pointwise Mutual Information (2)

- Only the words or phrases that exist in the Wordnet
- And appear with frequency more than 100 in the corpus
- Find words to the left and right of the word (context) in phrases:

Table shows forty paradigm words

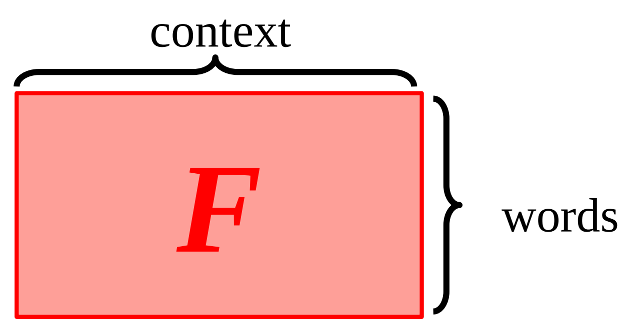
-
-

Pointwise Mutual Information (2)

- Only the words or phrases that exist in the Wordnet
- And appear with frequency more than 100 in the corpus
- Find words to the left and right of the word (context) in phrases:

Table shows forty paradigm words

- Create word-context frequency matrix ***F***:

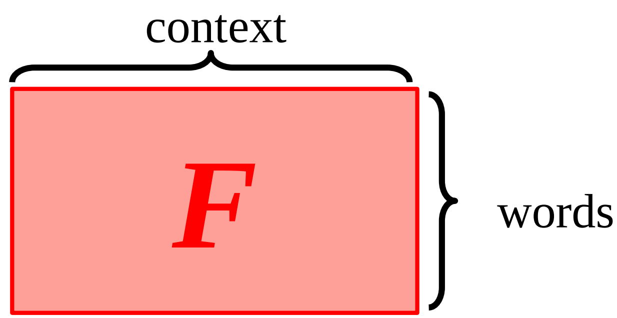


Pointwise Mutual Information (2)

- Only the words or phrases that exist in the Wordnet
- And appear with frequency more than 100 in the corpus
- Find words to the left and right of the word (context) in phrases:

Table shows forty paradigm words

- Create word-context frequency matrix ***F***:

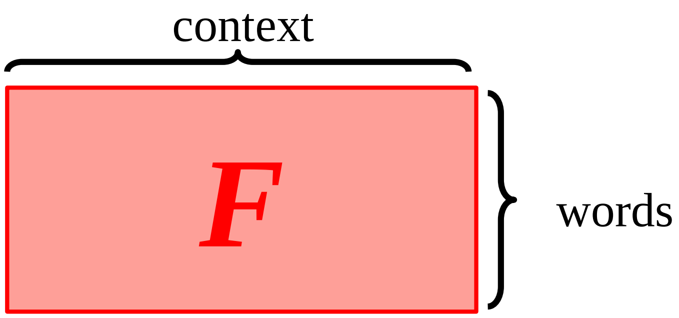


- f_{ij} is the number of times w_i appear in context c_j .

Pointwise Mutual Information (3)

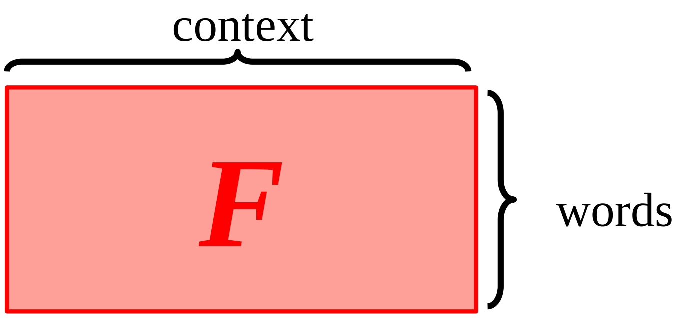
Pointwise Mutual Information (3)

- Create word-context frequency matrix ***F***:



Pointwise Mutual Information (3)

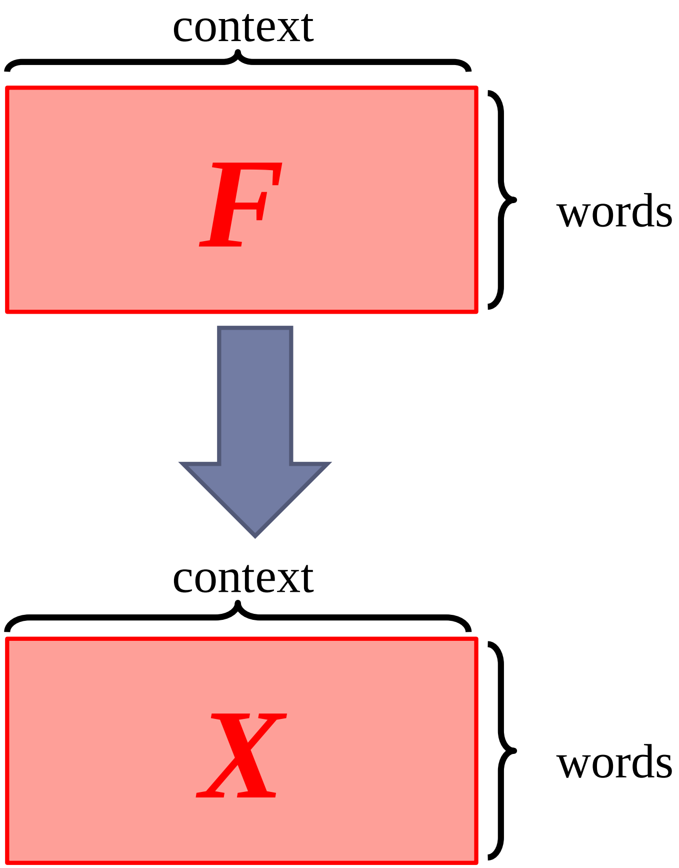
- Create word-context frequency matrix ***F***:



- Create PPMI matrix:

Pointwise Mutual Information (3)

- Create word-context frequency matrix ***F***:
- Create PPMI matrix:

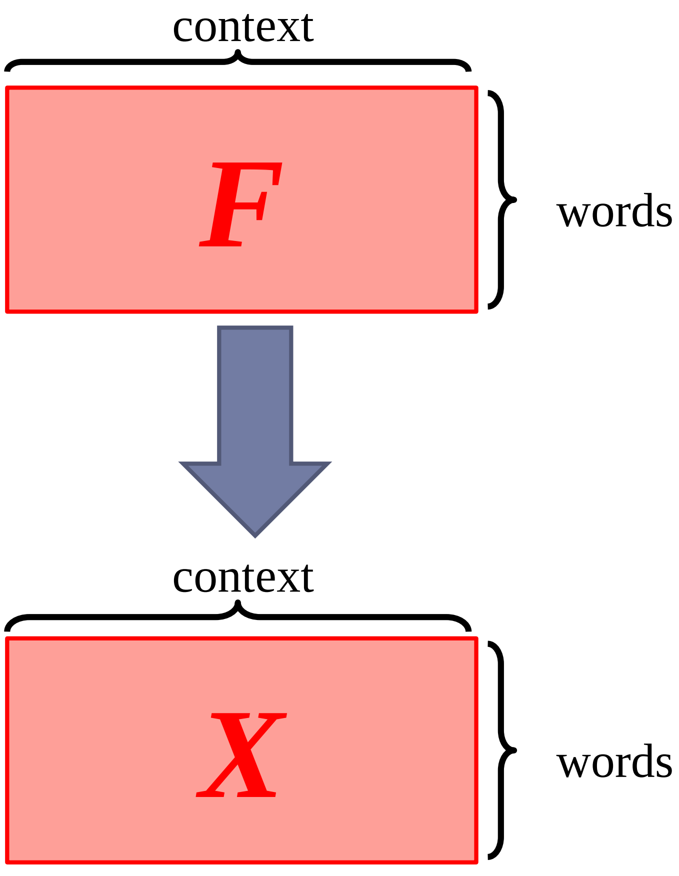


Pointwise Mutual Information (3)

- Create word-context frequency matrix ***F***:

- Create PPMI matrix:

$$p_{ij} = f_{ij} \Big/ \sum_{j=1}^{n_r} \sum_{i=1}^{n_c} f_{ij}$$

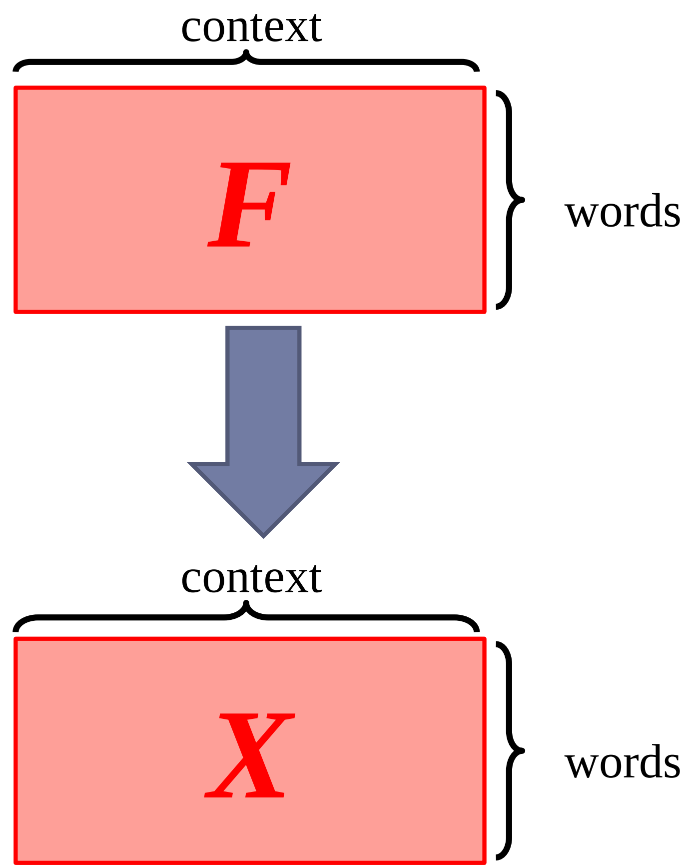


Pointwise Mutual Information (3)

- Create word-context frequency matrix ***F***:
- Create PPMI matrix:

$$p_{ij} = f_{ij} \Big/ \sum_{j=1}^{n_r} \sum_{i=1}^{n_c} f_{ij}$$

$$p_{*j} = \sum_{i=1}^{n_c} f_{ij} \Big/ \sum_{j=1}^{n_r} \sum_{i=1}^{n_c} f_{ij}$$



Pointwise Mutual Information (3)

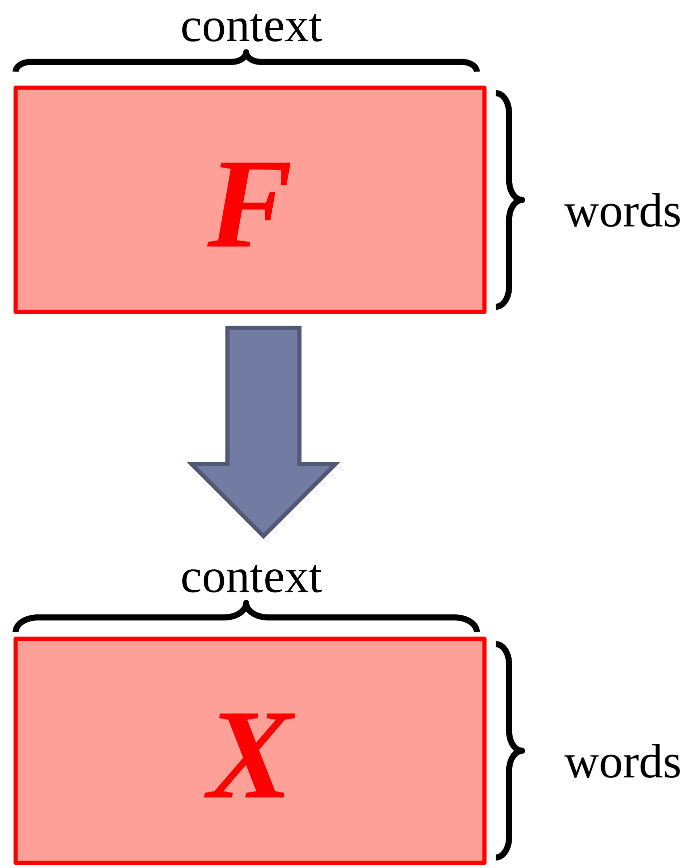
- Create word-context frequency matrix ***F***:

- Create PPMI matrix:

$$p_{ij} = f_{ij} \Big/ \sum_{j=1}^{n_r} \sum_{i=1}^{n_c} f_{ij}$$

$$p_{*j} = \sum_{i=1}^{n_c} f_{ij} \Big/ \sum_{j=1}^{n_r} \sum_{i=1}^{n_c} f_{ij}$$

$$p_{i*} = \sum_{j=1}^{n_c} f_{ij} \Big/ \sum_{j=1}^{n_r} \sum_{i=1}^{n_c} f_{ij}$$



Pointwise Mutual Information (3)

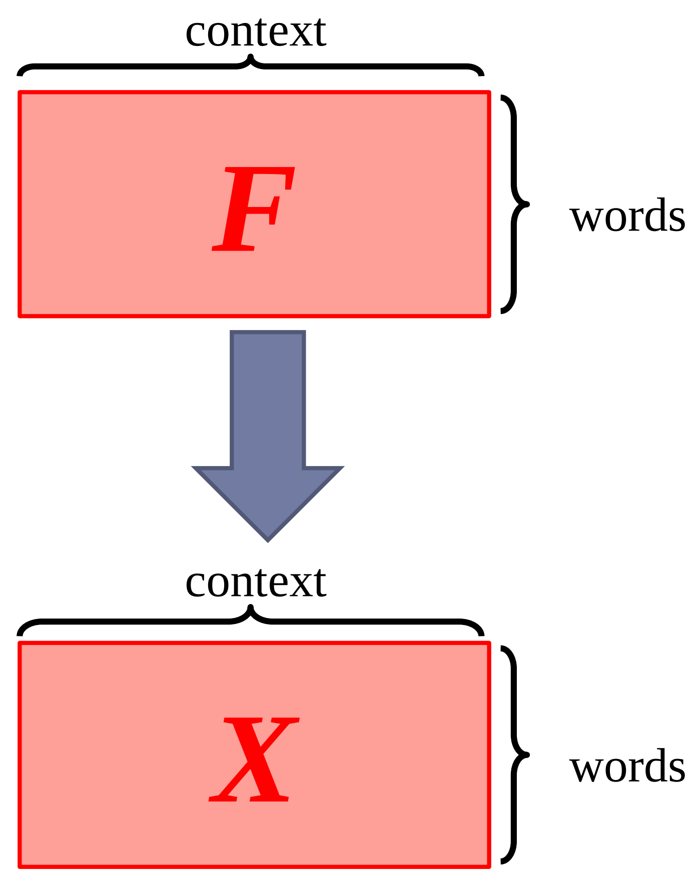
- Create word-context frequency matrix ***F***:
- Create PPMI matrix:

$$p_{ij} = f_{ij} \Big/ \sum_{j=1}^{n_r} \sum_{i=1}^{n_c} f_{ij}$$

$$p_{*j} = \sum_{i=1}^{n_c} f_{ij} \Big/ \sum_{j=1}^{n_r} \sum_{i=1}^{n_c} f_{ij}$$

$$p_{i*} = \sum_{j=1}^{n_c} f_{ij} \Big/ \sum_{j=1}^{n_r} \sum_{i=1}^{n_c} f_{ij}$$

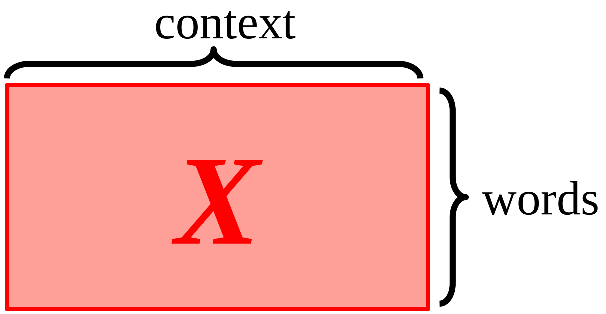
$$x_{ij} = \max\left(\log(\frac{p_{ij}}{p_{i*} p_{*j}}), 0 \right)$$



Pointwise Mutual Information (4)

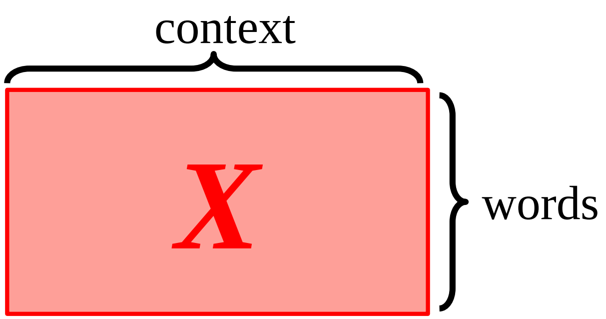
Pointwise Mutual Information (4)

- Given PPMI matrix:



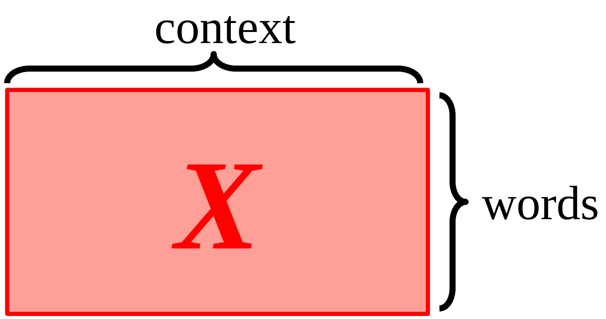
Pointwise Mutual Information (4)

- Given PPMI matrix:
- Word w_i in the *i*-th **row**



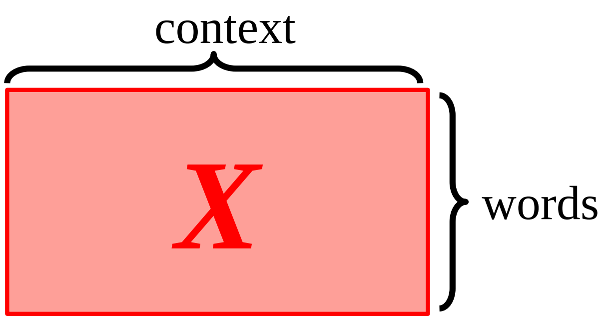
Pointwise Mutual Information (4)

- Given PPMI matrix:
- Word w_i in the i-th **row**
- Word w_j in the j-th **column**



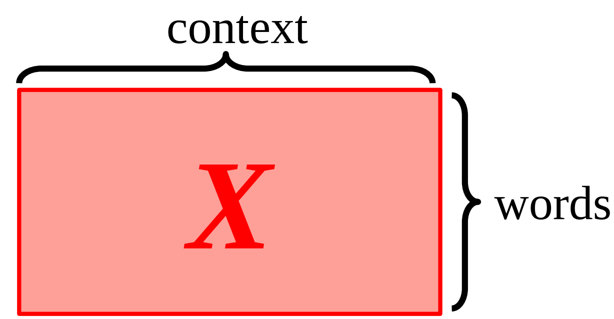
Pointwise Mutual Information (4)

- Given PPMI matrix:
- Word w_i in the i-th **row**
- Word w_j in the j-th **column**
- $$\begin{cases} \mathrm{PPMI}(w_i, w_j, left) = x_{ij}^{left} \\ \\ \\ \end{cases}$$

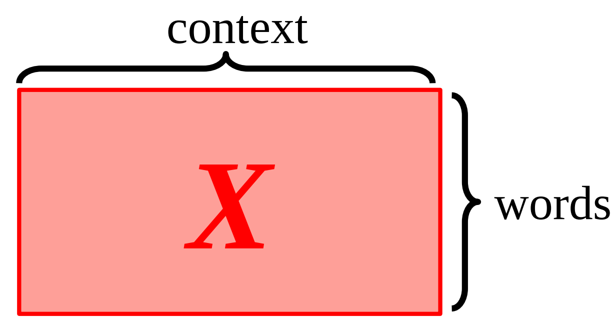


Pointwise Mutual Information (4)

- Given PPMI matrix:
- Word w_i in the i-th **row**
- Word w_j in the j-th **column**
- $$\begin{cases} \mathrm{PPMI}(w_i, w_j, \mathit{left}) = x_{ij}^{\mathit{left}} \\ \mathrm{PPMI}(w_i, w_j, \mathit{right}) = x_{ij}^{\mathit{right}} \end{cases}$$



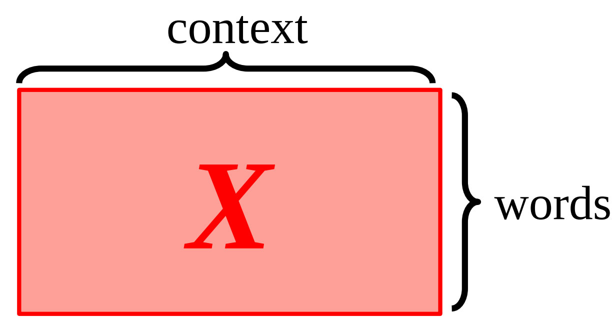
Pointwise Mutual Information (4)



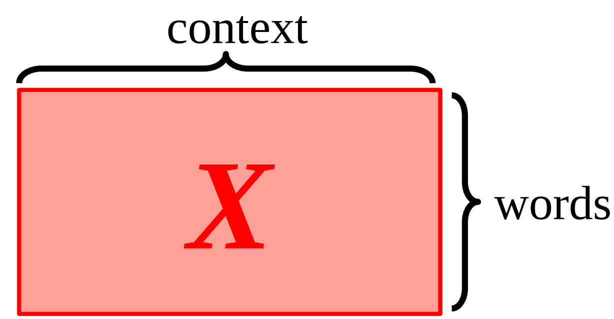
- Given PPMI matrix:
- Word w_i in the i-th **row**
- Word w_j in the j-th **column**
- $$\begin{cases} \text{PPMI}(w_i, w_j, left) = x_{ij}^{left} \\ \text{PPMI}(w_i, w_j, right) = x_{ij}^{right} \\ \text{PPMI}(w_j, w_i, left) = x_{ji}^{left} \end{cases}$$

Pointwise Mutual Information (4)

- Given PPMI matrix:
- Word w_i in the i-th **row**
- Word w_j in the j-th **column**
- $$\begin{cases} \mathrm{PPMI}(w_i, w_j, left) = x_{ij}^{left} \\ \mathrm{PPMI}(w_i, w_j, right) = x_{ij}^{right} \\ \mathrm{PPMI}(w_j, w_i, left) = x_{ji}^{left} \\ \mathrm{PPMI}(w_j, w_i, right) = x_{ji}^{right} \end{cases}$$



Pointwise Mutual Information (4)



- Given PPMI matrix:
- Word w_i in the i-th **row**
- Word w_j in the j-th **column**
- $$\begin{cases} \mathrm{PPMI}(w_i, w_j, left) = x_{ij}^{left} \\ \mathrm{PPMI}(w_i, w_j, right) = x_{ij}^{right} \\ \mathrm{PPMI}(w_j, w_i, left) = x_{ji}^{left} \\ \mathrm{PPMI}(w_j, w_i, right) = x_{ji}^{right} \end{cases}$$
- For an n-tuple one can generate $2n(n-1)$ PPMI features

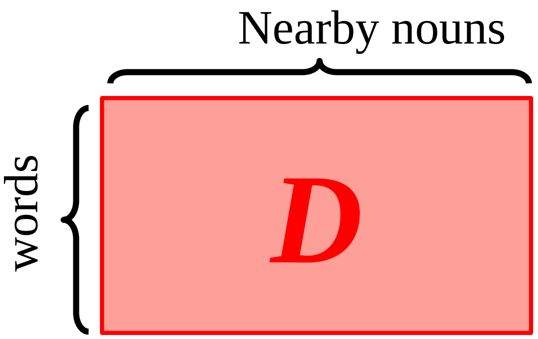
A vector space for domain similarity

A vector space for domain similarity

- Designed to capture the topic of a word.

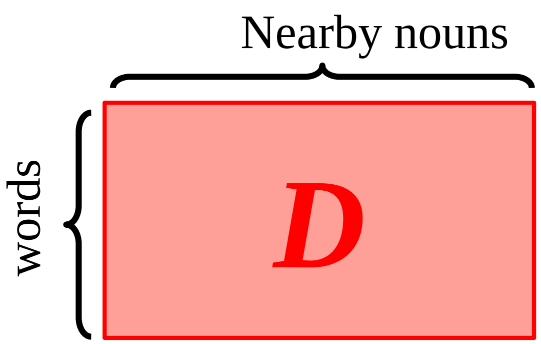
A vector space for domain similarity

- Designed to capture the topic of a word.
- Construct a frequency matrix:
  - Rows: correspond to words in Wordnet
  - Columns: Nearby nouns



A vector space for domain similarity

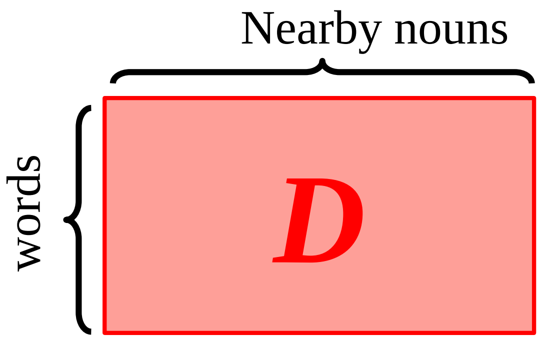
- Designed to capture the topic of a word.
- Construct a frequency matrix:
  - Rows: correspond to words in Wordnet
  - Columns: Nearby nouns



- Given a term x_i search the corpus for it

A vector space for domain similarity

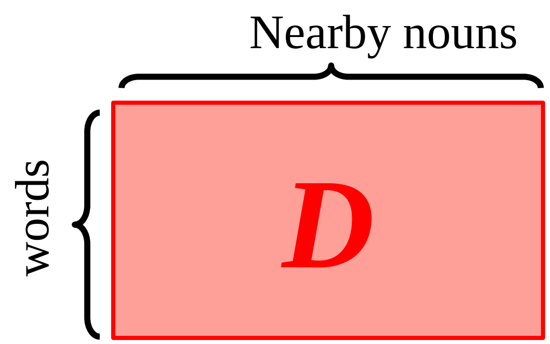
- Designed to capture the topic of a word.
- Construct a frequency matrix:
  - Rows: correspond to words in Wordnet
  - Columns: Nearby nouns



- Given a term x_i search the corpus for it
- Choose x_j a "noun" closest to the right/left of

A vector space for domain similarity

- Designed to capture the topic of a word.
- Construct a frequency matrix:
  - Rows: correspond to words in Wordnet
  - Columns: Nearby nouns



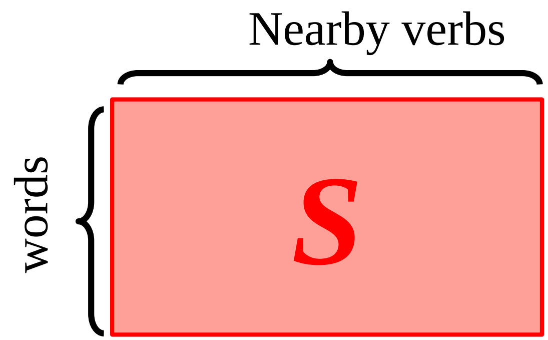
- Given a term x_i search the corpus for it
- Choose x_j a "noun" closest to the right/left of
- And increment d_{ij} by one.

A vector space for functional similarity

- Exactly the same as the domain similarity measures
  - Except that it is made using the "verbal" context

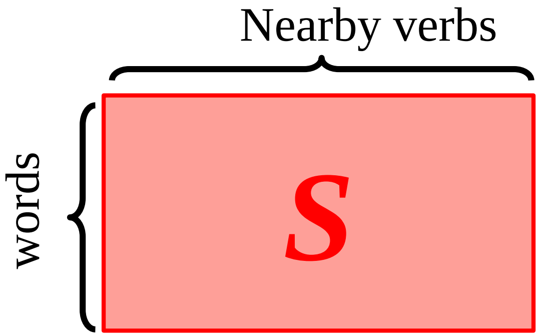
A vector space for functional similarity

- Exactly the same as the domain similarity measures
  - Except that it is made using the "verbal" context
- Construct a frequency matrix:



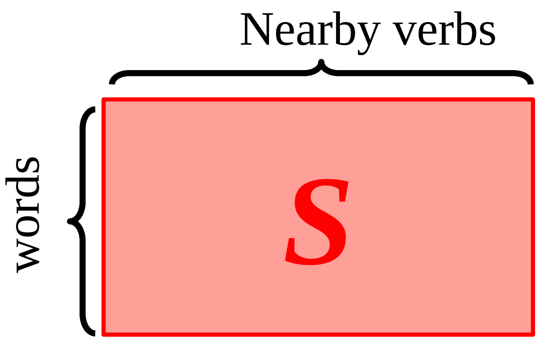
A vector space for functional similarity

- Exactly the same as the domain similarity measures
  - Except that it is made using the "verbal" context
- Construct a frequency matrix:
  - Rows: correspond to terms in Wordnet
  - Columns: Nearby verbs



A vector space for functional similarity

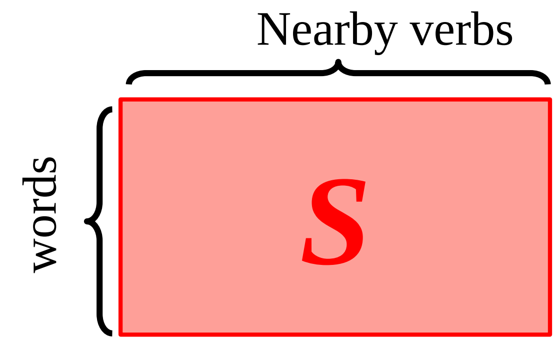
- Exactly the same as the domain similarity measures
  - Except that it is made using the "verbal" context
- Construct a frequency matrix:
  - Rows: correspond to terms in Wordnet
  - Columns: Nearby verbs



- Given a term x_i search the corpus for it
- ↓
- ↓

A vector space for functional similarity

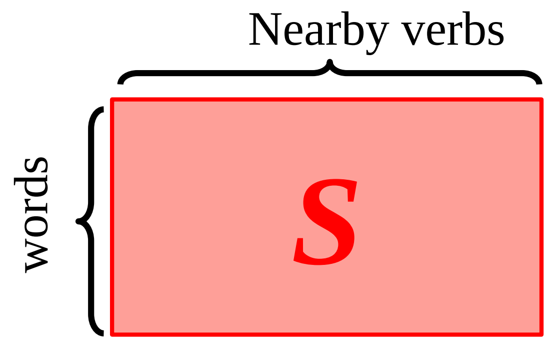
- Exactly the same as the domain similarity measures
  - Except that it is made using the "verbal" context
- Construct a frequency matrix:
  - Rows: correspond to terms in Wordnet
  - Columns: Nearby verbs



- Given a term x_i search the corpus for it
- Choose x_j a "verbs" closest to the right/left of

A vector space for functional similarity

- Exactly the same as the domain similarity measures
  - Except that it is made using the "verbal" context
- Construct a frequency matrix:
  - Rows: correspond to terms in Wordnet
  - Columns: Nearby verbs

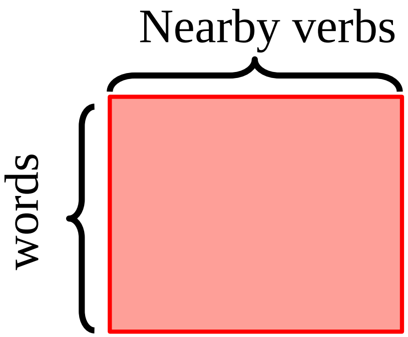


- Given a term x_i search the corpus for it
- Choose x_j a "verbs" closest to the right/left of
- And increment d_{ij} by one.

Using semantic and functional similarity

Using semantic and functional similarity

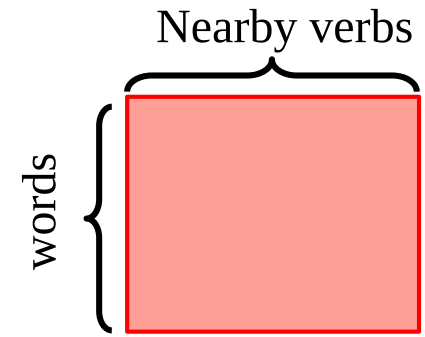
- Given the frequency matrix:



Using semantic and functional similarity

- Given the frequency matrix:
- Keep the lower-dimensional representation

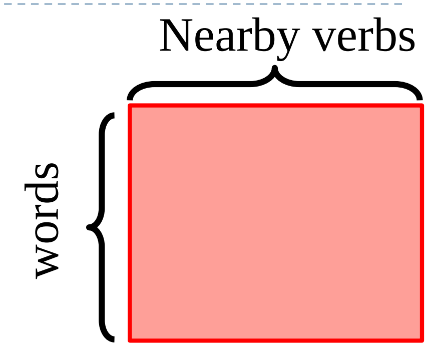
$$F = U\Sigma V$$



Using semantic and functional similarity

- Given the frequency matrix:
- Keep the lower-dimensional representation

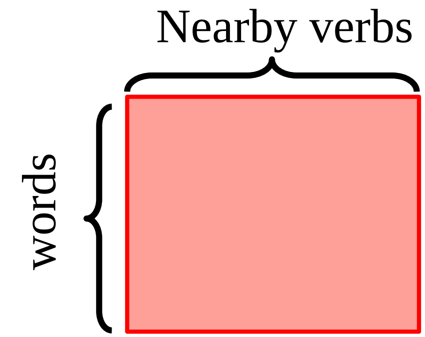
$$F = U\Sigma V$$



- Keep the values corresponding to the k biggest eigenvalues

$$F \approx U_k \Sigma_k V_k$$

Using semantic and functional similarity



- Given the frequency matrix:
- Keep the lower-dimensional representation

$$F = U \Sigma V$$

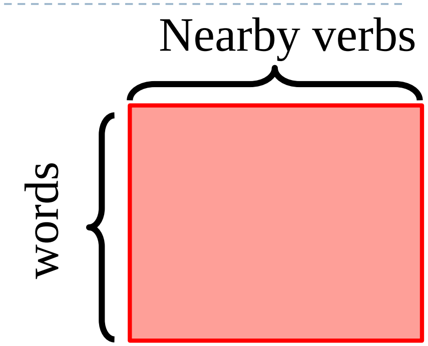
- Keep the values corresponding to the k biggest eigenvalues

$$F \approx U_k \Sigma_k V_k$$

- Given word w_i, $U_k \Sigma_k^p$ is the corresponding vector

Using semantic and functional similarity

- Given the frequency matrix:
- Keep the lower-dimensional representation



$$F = U\Sigma V$$

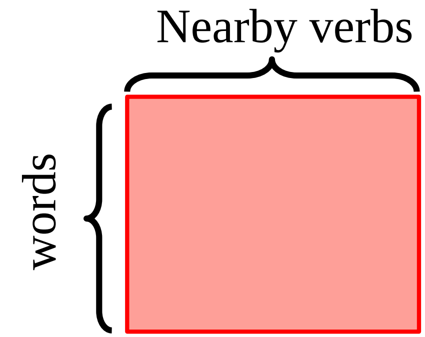
- Keep the values corresponding to the k biggest eigenvalues

$$F \approx U_k \Sigma_k V_k$$

- Given word w_i, $U_k \Sigma_k^p$ is the corresponding vector
  - $p \in [0,1]$ is used to tune sensitivity with respect to eigenvalues

Using semantic and functional similarity

- Given the frequency matrix:
- Keep the lower-dimensional representation



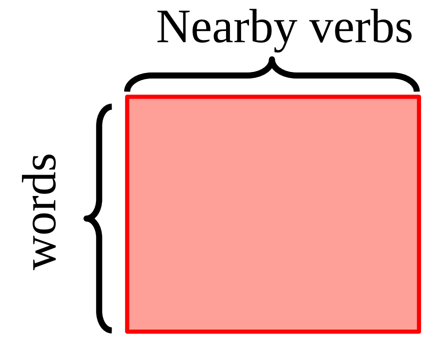
$$F = U\Sigma V$$

- Keep the values corresponding to the k biggest eigenvalues

$$F \approx U_k \Sigma_k V_k$$

- Given word w_i, $U_k \Sigma_k^p$ is the corresponding vector
- $p \in [0,1]$ is used to tune sensitivity with respect to eigenvalues
  - Given w_i and w_j to find: $Dom(w_i, w_j, k, p)$

Using semantic and functional similarity



- Given the frequency matrix:
- Keep the lower-dimensional representation

$$F = U\Sigma V$$

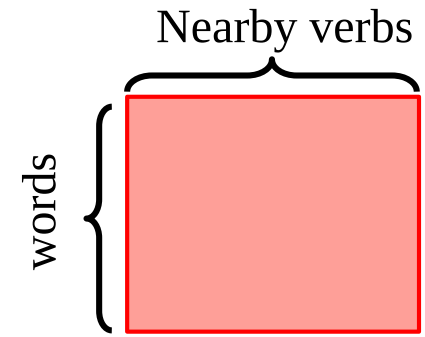
- Keep the values corresponding to the k biggest eigenvalues

$$F \approx U_k \Sigma_k V_k$$

- Given word w_i, $U_k \Sigma_k^p$ is the corresponding vector
- $p \in [0,1]$ is used to tune sensitivity with respect to eigenvalues
  - Given w_i and w_j to find: $Dom(w_i, w_j, k, p)$
  - *find corresponding vectors*

Using semantic and functional similarity

- Given the frequency matrix:



- Keep the lower-dimensional representation

$$F = U \Sigma V$$

- Keep the values corresponding to the k biggest eigenvalues

$$F \approx U_k \Sigma_k V_k$$

- Given word w_i, $U_k \Sigma_k^p$ is the corresponding vector
- $p \in [0,1]$ is used to tune sensitivity with respect to eigenvalues
  - Given w_i and w_j to find: $Dom(w_i, w_j, k, p)$
  - *find corresponding vectors*
  - *find cosine distance between the vectors*

5-choice SAT tests

- 374 five-choice SAT questions

Stem:		word:language
Choices:	(1)	paint:portrait
	(2)	poetry:rhythm
	(3)	note:music
	(4)	tale:story
	(5)	week:year
Solution:	(3)	note:music

5-choice SAT tests

- 374 five-choice SAT questions

Stem:		word:language
Choices:	(1)	paint:portrait
	(2)	poetry:rhythm
	(3)	note:music
	(4)	tale:story
	(5)	week:year
Solution:	(3)	note:music

- Could be converted into 5 4-tuples:

$$\langle \textit{word, language, note, music} \rangle$$

↓

5-choice SAT tests

- 374 five-choice SAT questions

Stem:		word:language
Choices:	(1)	paint:portrait
	(2)	poetry:rhythm
	(3)	note:music
	(4)	tale:story
	(5)	week:year
Solution:	(3)	note:music

- Could be converted into 5 4-tuples:

$$\langle word,\ language,\ note,\ music \rangle$$

- Each positive 4-tuple $\langle a, b, c, d \rangle$ could be converted to:

5-choice SAT tests

- 374 five-choice SAT questions

Stem:		word:language
Choices:	(1)	paint:portrait
	(2)	poetry:rhythm
	(3)	note:music
	(4)	tale:story
	(5)	week:year
Solution:	(3)	note:music

- Could be converted into 5 4-tuples:

$$\langle word,\ language,\ note,\ music \rangle$$

- Each positive 4-tuple $\langle a, b, c, d \rangle$ could be converted to:

$$\langle b, a, d, c \rangle,\ \ \langle c, d, a, b \rangle,\ \langle d, c, b, a \rangle$$

Results on 5-choice SAT

- The top ten results with the SAT analogy questions.

Algorithm	Reference	Correct
Know-Best	Veale (2004)	43.0
k-means	Biçici & Yuret (2006)	44.0
BagPack	Herdağdelen & Baroni (2009)	44.1
VSM	Turney & Littman (2005)	47.1
Dual-Space	Turney (2012)	51.1
BMI	Bollegala et al. (2009)	51.1
PairClass	Turney (2008b)	52.1
PERT	Turney (2006a)	53.5
SuperSim	—	54.8
LRA	Turney (2006b)	56.1
Human	Average college applicant	57.0

Results on 5-choice SAT

- The top ten results with the SAT analogy questions.

Algorithm	Reference	Correct
Know-Best	Veale (2004)	43.0
k-means	Biçici & Yuret (2006)	44.0
BagPack	Herdağdelen & Baroni (2009)	44.1
VSM	Turney & Littman (2005)	47.1
Dual-Space	Turney (2012)	51.1
BMI	Bollegala et al. (2009)	51.1
PairClass	Turney (2008b)	52.1
PERT	Turney (2006a)	53.5
SuperSim	—	54.8
LRA	Turney (2006b)	56.1
Human	Average college applicant	57.0

not significantly different according to Fisher's exact test at the 95% confidence level

Results on 5-choice SAT

- The top ten results with the SAT analogy questions.

Algorithm	Reference	Correct
Know-Best	Veale (2004)	43.0
k-means	Biçici & Yuret (2006)	44.0
BagPack	Herdağdelen & Baroni (2009)	44.1
VSM	Turney & Littman (2005)	47.1
Dual-Space	Turney (2012)	51.1
BMI	Bollegala et al. (2009)	51.1
PairClass	Turney (2008b)	52.1
PERT	Turney (2006a)	53.5
SuperSim	—	54.8
LRA	Turney (2006b)	56.1
Human	Average college applicant	57.0

not significantly different according to Fisher's exact test at the 95% confidence level

- SuperSim answers the SAT questions in a few minutes
- LRA requires nine days

SAT with 10 choices

SAT with 10 choices

- Adding more negative instances

SAT with 10 choices

- Adding more negative instances
- In general if $\langle a, b, c, d \rangle$ is positive $\langle a, d, c, b \rangle$ is negative

SAT with 10 choices

- Adding more negative instances
- In general if $\langle a, b, c, d \rangle$ is positive $\langle a, d, c, b \rangle$ is negative
- For example: Positive: $\langle word,\ language,\ note,\ music \rangle$

SAT with 10 choices

- Adding more negative instances
- In general if $\langle a, b, c, d \rangle$ is positive $\langle a, d, c, b \rangle$ is negative
- For example: Positive: $\langle word,\ language,\ note,\ music \rangle$

  Negative: $\langle word,\ music,\ note,\ language \rangle$

SAT with 10 choices

- Adding more negative instances
- In general if $\langle a, b, c, d \rangle$ is positive $\langle a, d, c, b \rangle$ is negative
- For example: Positive: $\langle word,\ language,\ note,\ music \rangle$

  Negative: $\langle word,\ music,\ note,\ language \rangle$
- This generates 5 more negative instances

SAT with 10 choices

- Adding more negative instances
- In general if $\langle a, b, c, d \rangle$ is positive $\langle a, d, c, b \rangle$ is negative
- For example: Positive: $\langle word,\ language,\ note,\ music \rangle$
  Negative: $\langle word,\ music,\ note,\ language \rangle$
- This generates 5 more negative instances

	Features				
Algorithm	LF	PPMI	Dom	Fun	Correct
Dual-Space	0	0	1	1	47.9
SuperSim	1	1	1	1	52.7
SuperSim	0	1	1	1	52.7
SuperSim	1	0	1	1	52.7
SuperSim	1	1	0	1	45.7
SuperSim	1	1	1	0	41.7
SuperSim	1	0	0	0	5.6
SuperSim	0	1	0	0	32.4
SuperSim	0	0	1	0	39.6
SuperSim	0	0	0	1	39.3

SemEval-2012 Task 2

- Class and subclasses labels + examples

```
CLASS-INCLUSION,Taxonomic
50.0 "weapon:spear"
...
34.7 "vegetable:carrot"
...
-1.9 "mammal:porpoise"
...
-29.8 "pen:ballpoint"
...
-55.1 "wheat:bread"
```

- Gather using Mechanical Turk:
- 75 subcategories
- Average of 41 word-pairs per subcategories

SemEval-2012 Task 2

- SuperSim Trained on 5-choice SAT and tested on SemEval data
- It gives the best correlation coefficient

Algorithm	Reference	Spearman
BUAP	Tovar et al. (2012)	0.014
Duluth-V2	Pedersen (2012)	0.038
Duluth-V1	Pedersen (2012)	0.039
Duluth-V0	Pedersen (2012)	0.050
UTD-SVM	Rink & Harabagiu (2012)	0.116
UTD-NB	Rink & Harabagiu (2012)	0.229
RNN-1600	Mikolov et al. (2013)	0.275
UTD-LDA	Rink & Harabagiu (2013)	0.334
Com	Zhila et al. (2013)	0.353
SuperSim	—	0.408

Compositional similarity: the data

- Noun-modifier question based on WordNet

Stem:		fantasy world
Choices:	(1)	fairyland
	(2)	fantasy
	(3)	world
	(4)	phantasy
	(5)	universe
	(6)	ranter
	(7)	souring
Solution:	(1)	fairyland

Compositional similarity: the data

- Noun-modifier question based on WordNet

Stem:		fantasy world
Choices:	(1)	fairyland
	(2)	fantasy
	(3)	world
	(4)	phantasy
	(5)	universe
	(6)	ranter
	(7)	souring
Solution:	(1)	fairyland

- Create tuples of the form: $\langle a, b, c \rangle$

Compositional similarity: the data

- Noun-modifier question based on WordNet

Stem:		fantasy world
Choices:	(1)	fairyland
	(2)	fantasy
	(3)	world
	(4)	phantasy
	(5)	universe
	(6)	ranter
	(7)	souring
Solution:	(1)	fairyland

- Create tuples of the form: $\langle a, b, c \rangle$
  - Example: $\langle$*fantasy, world, fairyland*$\rangle$

Compositional similarity: the data

- Noun-modifier question based on WordNet

Stem:		fantasy world
Choices:	(1)	fairyland
	(2)	fantasy
	(3)	world
	(4)	phantasy
	(5)	universe
	(6)	ranter
	(7)	souring
Solution:	(1)	fairyland

- Create tuples of the form: $\langle a, b, c \rangle$
  - Example: $\langle \textit{fantasy, world, fairyland} \rangle$
- Any question gives one positive instance and six negative instance

Compositional similarity: 7-choices questions

- 680 questions for training
- 1,500 questions for testing

Total of 2,180 questions

Compositional similarity: 7-choices questions

- 680 questions for training
- 1,500 questions for testing
- Any question gives one positive instance and six negative instance

Total of 2,180 questions

Compositional similarity: 7-choices questions

- 680 questions for training
- 1,500 questions for testing
- Any question gives one positive instance and six negative instance
- And train a classifier given the tuple for probabilities

Total of 2,180 questions

Compositional similarity: 7-choices questions

- 680 questions for training
- 1,500 questions for testing

Total of 2,180 questions

- Any question gives one positive instance and six negative instance
- And train a classifier given the tuple for probabilities

Algorithm	7-choices
Vector addition	50.1
Element-wise multiplication	57.5
Dual-Space model	58.3
SuperSim	75.9
Holistic model	81.6

Compositional similarity: 7-choices questions

- 680 questions for training
- 1,500 questions for testing

Total of 2,180 questions

- Any question gives one positive instance and six negative instance
- And train a classifier given the tuple for probabilities

Algorithm	7-choices
Vector addition	50.1
Element-wise multiplication	57.5
Dual-Space model	58.3
SuperSim	75.9
Holistic model	81.6

- The holistic approach is noncompositional.
  - The stem bigram is represented by a single context vector
  - As if it were a unigram.

Compositional similarity:14-choices questions

Compositional similarity:14-choices questions

- Any question gives one positive instance and six negative instance

Compositional similarity: 14-choices questions

- Any question gives one positive instance and six negative instance
- Positive instance: $\langle a, b, c \rangle$

Compositional similarity: 14-choices questions

- Any question gives one positive instance and six negative instance
- Positive instance: $\langle a, b, c \rangle$
- Negative instance: $\langle b, a, c \rangle$ e.g. word fantasy $\neq$ wonderland

Compositional similarity: 14-choices questions

- Any question gives one positive instance and six negative instance
- Positive instance: $\langle a, b, c \rangle$
- Negative instance: $\langle b, a, c \rangle$ e.g. word fantasy $\neq$ wonderland
- This gives 7 more negative instances (14-choices)

Compositional similarity:14-choices questions

- Any question gives one positive instance and six negative instance
- Positive instance: $\langle a, b, c \rangle$
- Negative instance: $\langle b, a, c \rangle$ e.g. word fantasy $\neq$ wonderland
- This gives 7 more negative instances (14-choices)

	Correct	
Algorithm	7-choices	14-choices
Vector addition	50.1	22.5
Element-wise multiplication	57.5	27.4
Dual-Space model	58.3	41.5
SuperSim	75.9	68.0
Holistic model	81.6	—

Compositional similarity: ablation experiment

- Analyzing effect of each feature type on the 14-choice test

	Features				
Algorithm	LF	PPMI	Dom	Fun	Correct
Dual-Space	0	0	1	1	41.5
SuperSim	1	1	1	1	68.0
SuperSim	0	1	1	1	66.6
SuperSim	1	0	1	1	52.3
SuperSim	1	1	0	1	69.3
SuperSim	1	1	1	0	65.9
SuperSim	1	0	0	0	14.1
SuperSim	0	1	0	0	59.7
SuperSim	0	0	1	0	34.6
SuperSim	0	0	0	1	32.9

- PPMI features are the most important

Compositional similarity: closer look at PPMI

Compositional similarity: closer look at PPMI

- PPMI features for $\langle a, b, c\rangle$ into three subsets:

$$\langle a, b\rangle, \langle a, c\rangle, \langle b, c\rangle$$

Compositional similarity: closer look at PPMI

- PPMI features for $\langle a, b, c\rangle$ into three subsets:

$$\langle a, b\rangle, \langle a, c\rangle, \ \langle b, c\rangle$$

- For example for $\langle a, b\rangle$: $\boxed{\begin{array}{l}\mathrm{PPMI}(a, b, \mathit{left}), \ \mathrm{PPMI}(a, b, \mathit{right}) \\ \mathrm{PPMI}(b, a, \mathit{left}), \ \mathrm{PPMI}(b, a, \mathit{right})\end{array}}$

Compositional similarity: closer look at PPMI

- PPMI features for $\langle a, b, c\rangle$ into three subsets:

$$\langle a, b\rangle, \langle a, c\rangle, \ \langle b, c\rangle$$

- For example for $\langle a, b\rangle$: $\boxed{\begin{array}{l} \mathrm{PPMI}(a, b, \mathit{left}),\ \mathrm{PPMI}(a, b, \mathit{right}) \\ \mathrm{PPMI}(b, a, \mathit{left}),\ \mathrm{PPMI}(b, a, \mathit{right}) \end{array}}$

PPMI feature subsets			
$\langle a, b\rangle$	$\langle a, c\rangle$	$\langle b, c\rangle$	Correct
1	1	1	68.0
0	1	1	59.9
1	0	1	65.4
1	1	0	67.5
1	0	0	62.6
0	1	0	58.1
0	0	1	55.6
0	0	0	52.3

Compositional similarity: closer look at PPMI

- PPMI features for $\langle a, b, c\rangle$ into three subsets:

$$\langle a, b\rangle, \langle a, c\rangle, \langle b, c\rangle$$

- For example for $\langle a, b\rangle$: $\mathrm{PPMI}(a, b, \mathit{left})$, $\mathrm{PPMI}(a, b, \mathit{right})$ $\mathrm{PPMI}(b, a, \mathit{left})$, $\mathrm{PPMI}(b, a, \mathit{right})$

- $\langle a, b\rangle$ subset are more important.

PPMI feature subsets			
$\langle a, b\rangle$	$\langle a, c\rangle$	$\langle b, c\rangle$	Correct
1	1	1	68.0
0	1	1	59.9
1	0	1	65.4
1	1	0	67.5
1	0	0	62.6
0	1	0	58.1
0	0	1	55.6
0	0	0	52.3

Compositional similarity: holistic training

Compositional similarity: holistic training

- A holistic training data

Compositional similarity: holistic training

- A holistic training data

Stem:		search engine
Choices:	(1)	search_engine
	(2)	search
	(3)	engine
	(4)	search_language
	(5)	search_warrant
	(6)	diesel_engine
	(7)	steam_engine
Solution:	(1)	search_engine

Compositional similarity: holistic training

- A holistic training data
- Extract noun-modifier pairs from WordNet

Stem:		search engine
Choices:	(1)	search_engine
	(2)	search
	(3)	engine
	(4)	search_language
	(5)	search_warrant
	(6)	diesel_engine
	(7)	steam_engine
Solution:	(1)	search_engine

Compositional similarity: holistic training

- A holistic training data
- Extract noun-modifier pairs from WordNet
- Call a_b a pseudo-unigram and treat it as unigram

Stem:		search engine
Choices:	(1)	search_engine
	(2)	search
	(3)	engine
	(4)	search_language
	(5)	search_warrant
	(6)	diesel_engine
	(7)	steam_engine
Solution:	(1)	search_engine

Compositional similarity: holistic training

- A holistic training data
- Extract noun-modifier pairs from WordNet
- Call a_b a pseudo-unigram and treat it as unigram
- Use the components as distracters

Stem:		search engine
Choices:	(1)	search_engine
	(2)	search
	(3)	engine
	(4)	search_language
	(5)	search_warrant
	(6)	diesel_engine
	(7)	steam_engine
Solution:	(1)	search_engine

Compositional similarity: holistic training(2)

- Training on the holistic questions: "Holistic"
- Compared with the standard training
- Test is the standard testing

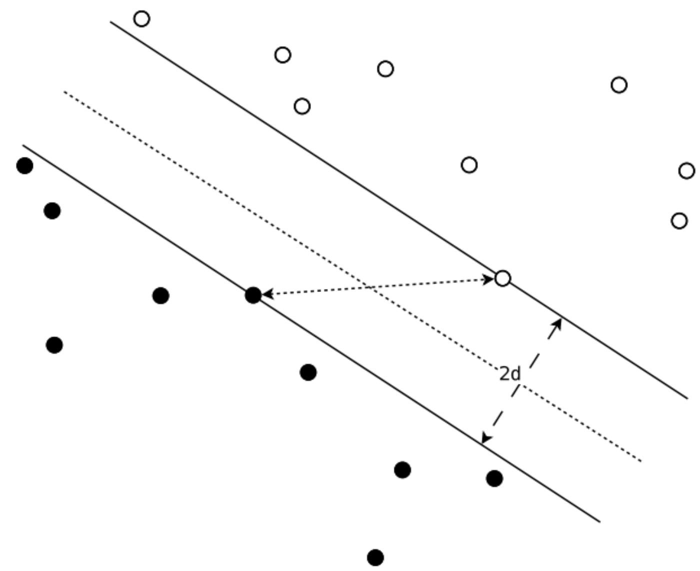
	Correct	
Training	7-choices	14-choices
Holistic	61.8	54.4
Standard	75.9	68.0

- There is a drop when training with the holistic samples
- Not very clear, but seems to be because of the nature of the

References

- Some figures from: http://nlp.cs.berkeley.edu/tutorials/variational-tutorial-slides.pdf
- Hinton, Geoffrey E., et al. "Improving neural networks by preventing co-adaptation of feature detectors." arXiv preprint arXiv:1207.0580 (2012).

SVM



- Primal form:

$$\begin{cases} \min_\beta \frac{1}{2}\|\beta\|^2 \\ y_i\left(\beta.\mathbf{x}_i\right)-1\geq 0 \end{cases}$$

- Relaxed form:

$$\begin{cases} \min_\beta \frac{1}{2}\|\beta\|^2 + C\sum_i \varepsilon_i \\ y_i\left(\beta.\mathbf{x}_i\right)-1\geq -\varepsilon_i \end{cases}$$

- Dual form:

$$\begin{cases} \max_\alpha \sum \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \left(\mathbf{x}_i.\mathbf{x}_j\right) \\ \sum_i \alpha_i y_i = 0, \quad 0\leq \alpha_i \leq C \end{cases}$$

Proof: Hinge regression closed form

- Suppose we want to minimize:

$$\left\|Ax - b\right\|^2 + \left\|\Gamma x\right\|^2$$

$$x = \left(A^T A + \Gamma^T \Gamma\right)^{-1} A^T b$$

Proof: Hinge regression closed form

- Proof:

$$L = \frac{1}{2}\left(Ax - b\right)^T \left(Ax - b\right)$$

$$\frac{dL}{dx} = A^T \left(Ax - b\right) = 0$$

$$x = \left(A^T A\right)^{-1} A^T b$$

Proof: Hinge regression closed form

- Proof:

$$L = \frac{1}{2}\left(Ax - b\right)^T\left(Ax - b\right) + \frac{1}{2}\left(\Gamma x\right)^T\left(\Gamma x\right)$$

$$\frac{dL}{dx} = A^T\left(Ax - b\right) + \Gamma^T\left(\Gamma x\right) = 0$$

$$x = \left(A^T A + \Gamma^T \Gamma\right)^{-1} A^T b$$