Semantic Knowledge Acquisition using Frequency Based Patterns

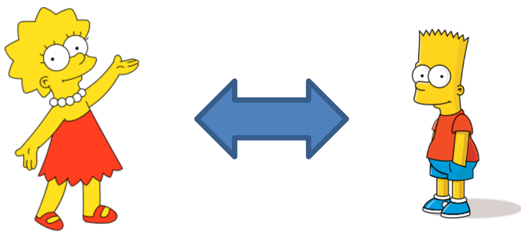
Roy Schwartz and Ari Rappoport

School of Computer Science and Engineering, The Hebrew University of Jerusalem, February 2015

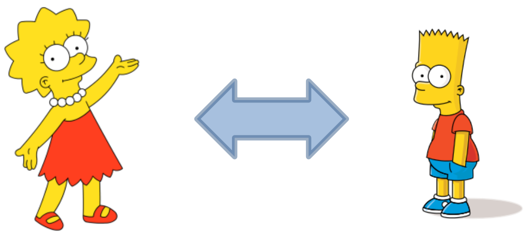
The Catalonia-Israel Symposium on Lexical Semantics and Grammatical Structure

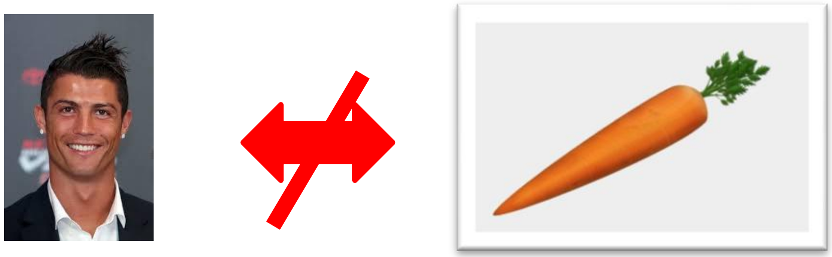
The Goal: Acquire (Lexical) Semantic Knowledge

The Goal: Acquire (Lexical) Semantic Knowledge

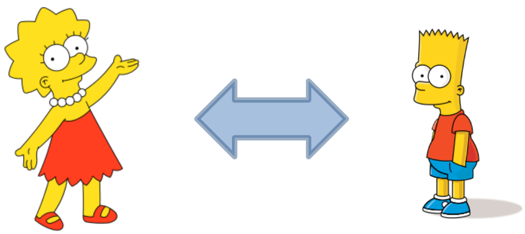


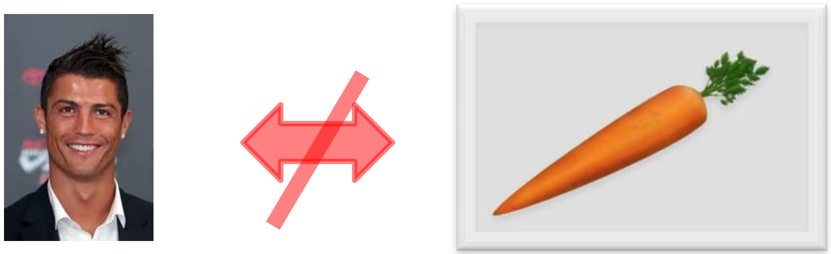
The Goal: Acquire (Lexical) Semantic Knowledge

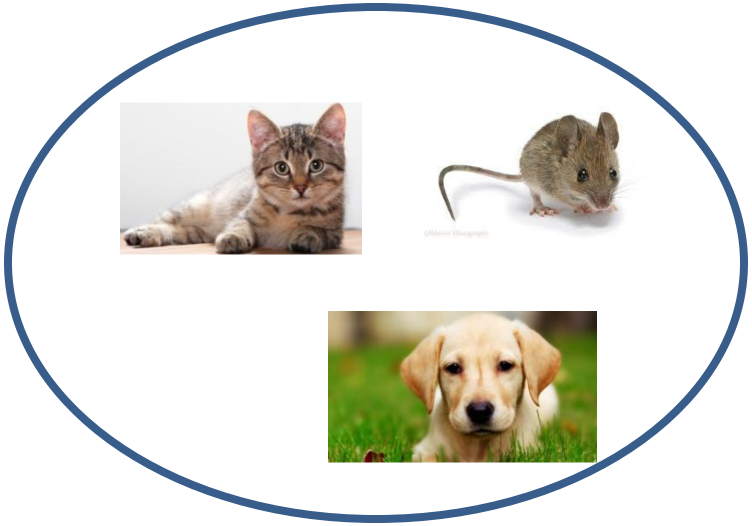




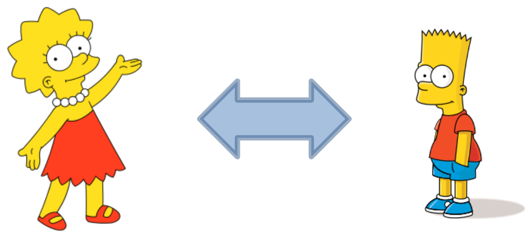
The Goal: Acquire (Lexical) Semantic Knowledge

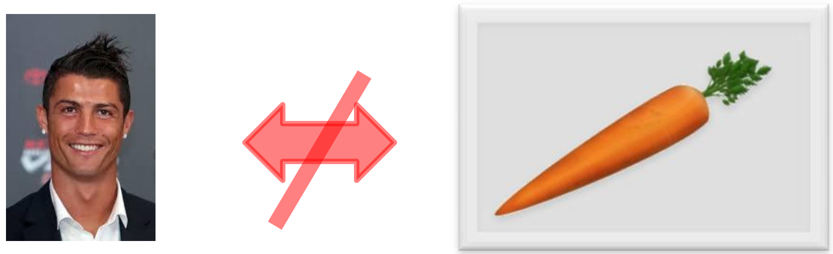


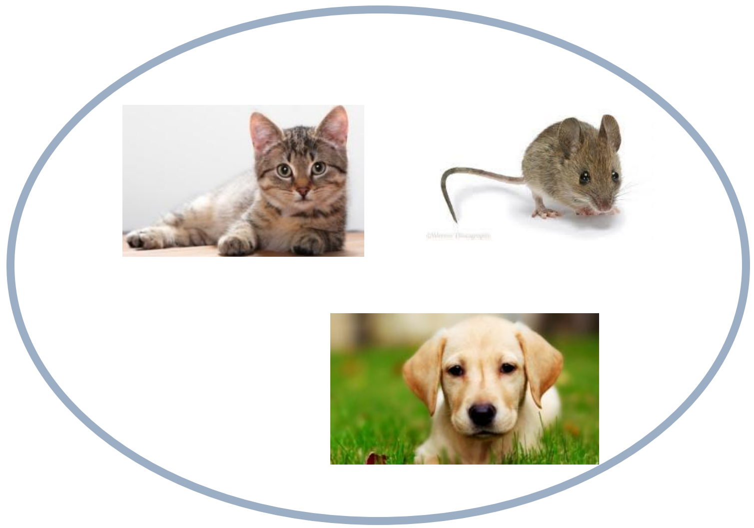


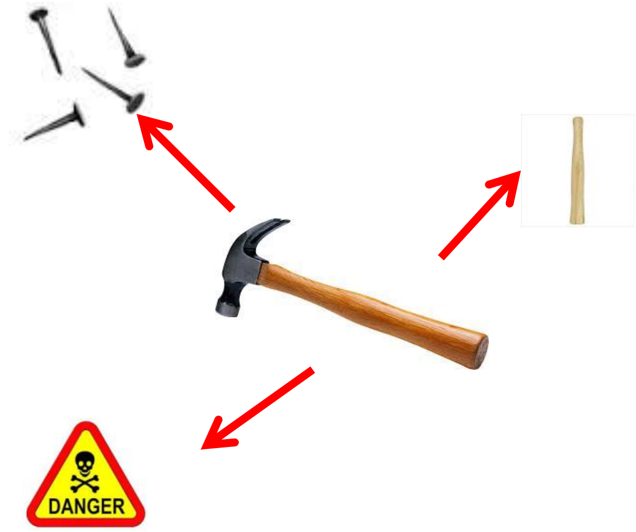


The Goal: Acquire (Lexical) Semantic Knowledge









Toolkit

Toolkit

Toolkit


words
speech
machine
language
task
evaluation
systems
NLP
learning
word
data
human
input
natural
processing
recognition
tasks
text
algorithms
rules
statistical
sentence
models
languages
example
possible
translation
sentences
understanding
research
computer
determine
information
morphology
standard
number
common
especially
hand-written
compared
difficult
English
analysis
complexity
automatic

Toolkit

X gave Y to Z

Disclaimer

- We present a highly effective **computational** method
- We do not attempt to make any **linguistic** or **cognitive** claim
  - Nevertheless, there are some related issues, e.g., in **construction grammar** theories

Overview

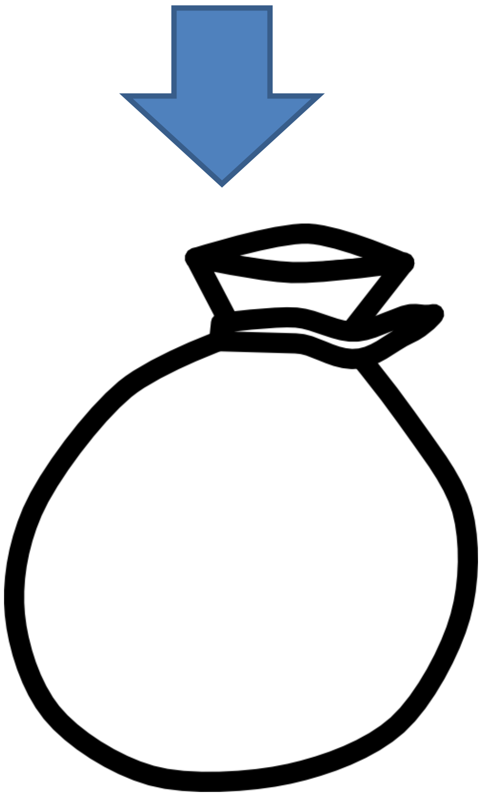
- Introduction
  - Bag of words models
  - Lexico-syntactic Patterns
  - Lexico-syntactic Patterns 2.0: Flexible Patterns
- Latest results
  - Interpretable Word Embeddings Using Patterns Features (**Schwartz**, Reichart and Rappoport, under review)

Bag-of-Words Models

John gave a present to Mary

Bag-of-Words Models

John gave a present to Mary



Bag-of-Words Models

John gave a present to Mary

Bag-of-Words Models

John gave a present to Mary

Distributional Semantics

(Harris, 1954)

Words that occur in similar ***context*** *are likely to have similar* ***meanings***

Bag-of-Words Applications

- Represent words using their surrounding (word) contexts
  - Word similarity / association
  - Word clustering / classification
  - …
- Represent phrases / sentences by the words that they contain
  - Sentiment analysis
  - Spam filters

Missing: Context

John gave a present to Marry

Missing: Context

John gave a $\boxed{\text{present}}$ to Marry

Missing: Context

$\boxed{\text{John}}$ gave a $\boxed{\text{present}}$ to $\boxed{\text{Marry}}$

Missing: Context

John gave a present to Marry

John's car broke down

John and Mary got married

Workers like John are an asset to every organization

Missing: Context

John gave a present to Marry

John's car broke down

John and Mary got married

Workers like John are an asset to every organization

Missing: Context

John gave a present to Marry

John's car broke down

John and Mary got married

Workers like John are an asset to every organization

Missing: Context

John gave a present to Marry

John's car broke down

John and Mary got married

Workers like John are an asset to every organization

Lexico-syntactic Patterns

Hearst, 1992

- Patterns of the form "***X*** *is a country*", "***X*** *such as* ***Y***", etc.

Lexico-syntactic Patterns

Hearst, 1992

- Patterns of the form "***X** is a country*", "***X** such as **Y***", etc.

- Patterns potentially capture the **context** in which a word participates

Lexico-syntactic Patterns

Hearst, 1992

- Patterns of the form "***X** is a country*", "***X** such as **Y***", etc.
- Patterns potentially capture the **context** in which a word participates
- For example:
  - A *dog* participates in patterns (contexts) such as:
  - "X barks", "X has a tail", "X and cats", …

Semantic Knowledge Acquisition using Patterns

- Extracting country names
  - "X is a country"

Semantic Knowledge Acquisition using Patterns

- Extracting country names
  - "X is a country"
  - ***Canada is a country*** *in north America*
  - *There's a sense in America that* ***France is a country*** *of culture*

Semantic Knowledge Acquisition using Patterns

- Extracting country names
  - "X is a country"
  - ***Canada is a country*** *in north America*
  - *There's a sense in America that* ***France is a country*** *of culture*
- Extracting hyponymy relations
  - "X such as Y"

Semantic Knowledge Acquisition using Patterns

- Extracting country names
  - "X is a country"
  - ***Canada is a country*** *in north America*
  - *There's a sense in America that* ***France is a country*** *of culture*
- Extracting hyponymy relations
  - "X such as Y"
  - *Cut the stems of boxed* ***flowers such as roses***
  - *I am responsible for preparing a range of* ***fruits such as apples***

Pattern Applications

- Acquiring the semantics of **single words**
  - Building semantic lexicons (Riloff and Shepherd, 1997; Roark and Charniak, 1998)
  - Semantic class learning (Kozareva et al., 2008)

- Acquiring the semantics of **relationships** between words
  - Discovering hyponymy (Hearst, 1992)
  - Discovering meronymy (Berland and Charniak, 1999)
  - Discovering antonymy (Lin et al., 2003)

Symmetric Patterns (SPs)

- X *and* Y
  - cats and dogs , dogs and cats
  - France and England, England and France

- X *as well as* Y
  - friends as well as colleagues, colleagues as well as friends
  - apples and oranges , oranges and apples

Symmetric Patterns (SPs)

- X *and* Y
  - cats and dogs , dogs and cats
  - France and England, England and France
- X *as well as* Y
  - friends as well as colleagues, colleagues as well as friends
  - apples and oranges , oranges and apples
- Words that co-occur in symmetric patterns are likely to be **similar** to one another
  - Widdows and Dorow, 2002; Dorow et al., 2005; Davidov et al., 2006, **Schwartz** et al., 2014

Limitations of Patterns

- The early works that adopted lexico-syntactic patterns used a set of **manually created** patterns
  - Require **human (experts) labor**
  - Language-specific

Patterns 2.0: **Flexible** Patterns

- Patterns that are extracted **automatically**

Patterns 2.0: **Flexible** Patterns

- Patterns that are extracted **automatically**

- Instead of defining a set of **fixed** patterns, we define **meta-patterns**
  - **Structures** of (potential) patterns
  - High frequency words (HFWs) are used instead of fixed words
  - Content words (CWs) appear as wildcards
  - E.g., "HFW_1 X HFW_2 Y"

Patterns 2.0: **Flexible** Patterns

- Patterns that are extracted **automatically**
- Instead of defining a set of **fixed** patterns, we define **meta-patterns**
  - **Structures** of (potential) patterns
  - High frequency words (HFWs) are used instead of fixed words
  - Content words (CWs) appear as wildcards
  - E.g., "HFW_1 X HFW_2 Y"

Frequent and informative patterns are automatically selected

Extracted Flexible Patterns

"HFW_1 X HFW_2 Y"

- as X as Y
- the X the Y
- an X from Y
- from X to Y
- a X has Y
- to X big Y
- in X the Y
- an X do Y
- to X and Y
- …

Extracted Flexible Patterns

"HFW_1 X HFW_2 Y"

- as X as Y
- the X the Y
- an X from Y
- from X to Y
- a X has Y
- to X big Y
- in X the Y
- an X do Y
- to X and Y
- …

Benefits of using Flexible Patterns

- Flexible patterns are computed **automatically**
  - Based solely on **word frequencies**
  - Do not require expert knowledge
  - Language and domain independent
  - Large coverage

Automatic Discovery of Symmetric Patterns
Davidov and Rappoport, ACL 2006

- An algorithm for extraction symmetric patterns from plain text (**symmetric** flexible patterns)
  - *“X and Y”, “X as well as Y”, “neither X nor Y”*
  - **Input**: a large corpus of plain text
  - **Output**: a set of symmetric patterns

Automatic Discovery of Symmetric Patterns

Davidov and Rappoport, ACL 2006

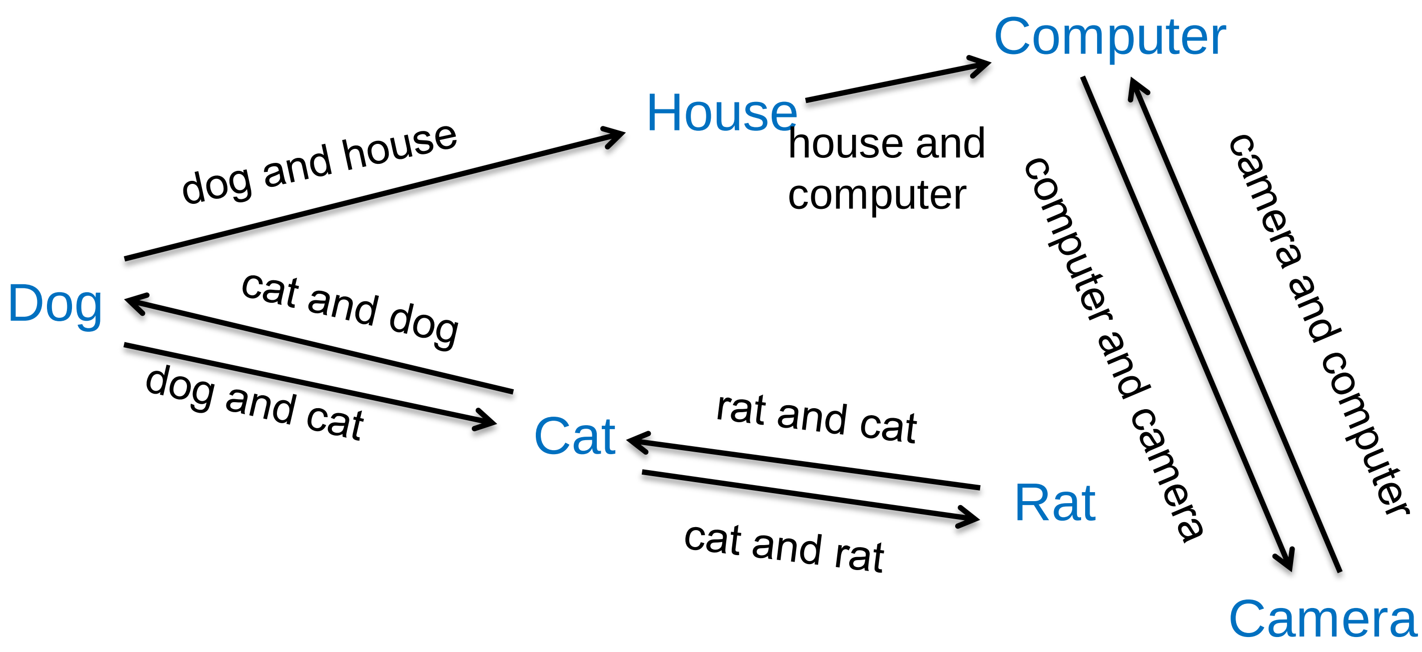
- An algorithm for extraction symmetric patterns from plain text (**symmetric** flexible patterns)
  - *"X and Y", "X as well as Y", "neither X nor Y"*
  - **Input**: a large corpus of plain text
  - **Output**: a set of symmetric patterns

- Application: cluster nouns into meaningful semantic groups

  Discovered categories include chemical elements, university names, languages, fruits, fishing baits…

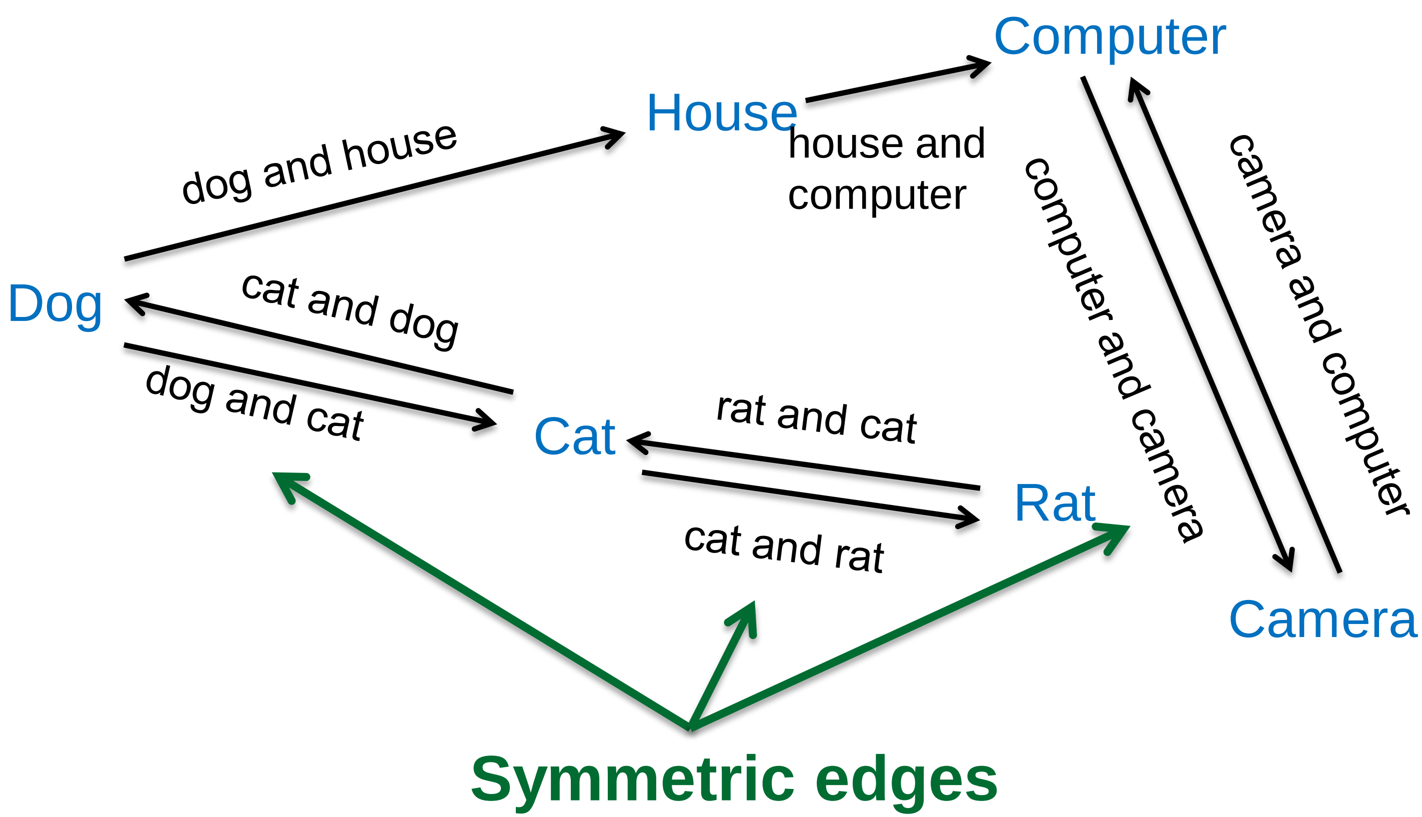
DR06 Example

X *and* Y



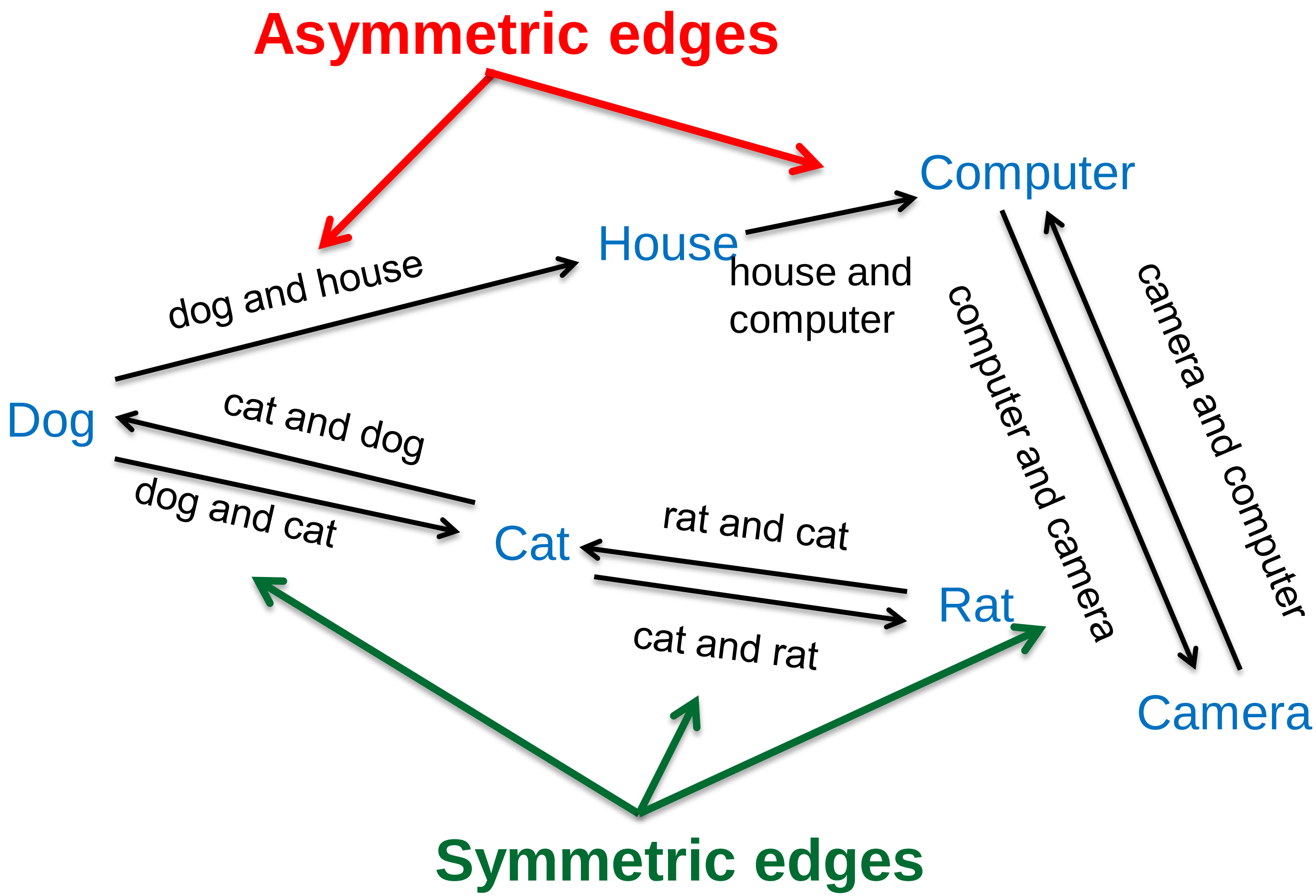
DR06 Example

X *and* Y



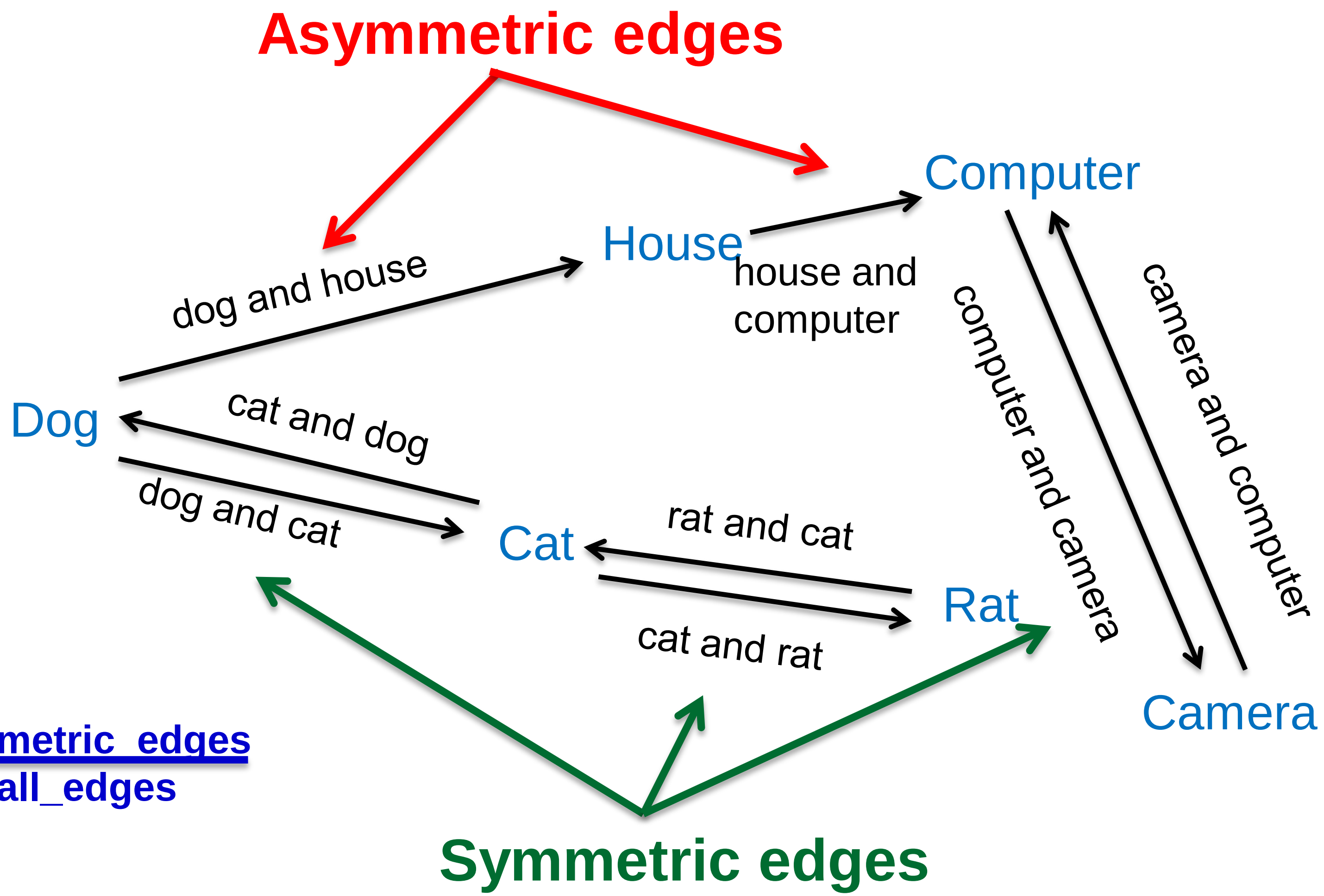
DR06 Example

X *and* Y



DR06 Example

X *and* Y



M = #symmetric_edges / #all_edges

Resulting Set of Patterns

- *“X and Y”*
- *“X or Y”*
- *“X as well as Y”*
- *“X nor Y”*

- *“X and the Y”*
- *“X or the Y”*
- *“X or a Y”*
- *“X and one Y”*

- *“from X to Y”*
- *“X rather than Y”*

Minimally Supervised Noun Classification

Schwartz, Reichart and Rappoport, Coling 2014

- Classify nouns into semantic categories
  - Animals, edibles, tools, …

Minimally Supervised Noun Classification

Schwartz, Reichart and Rappoport, Coling 2014

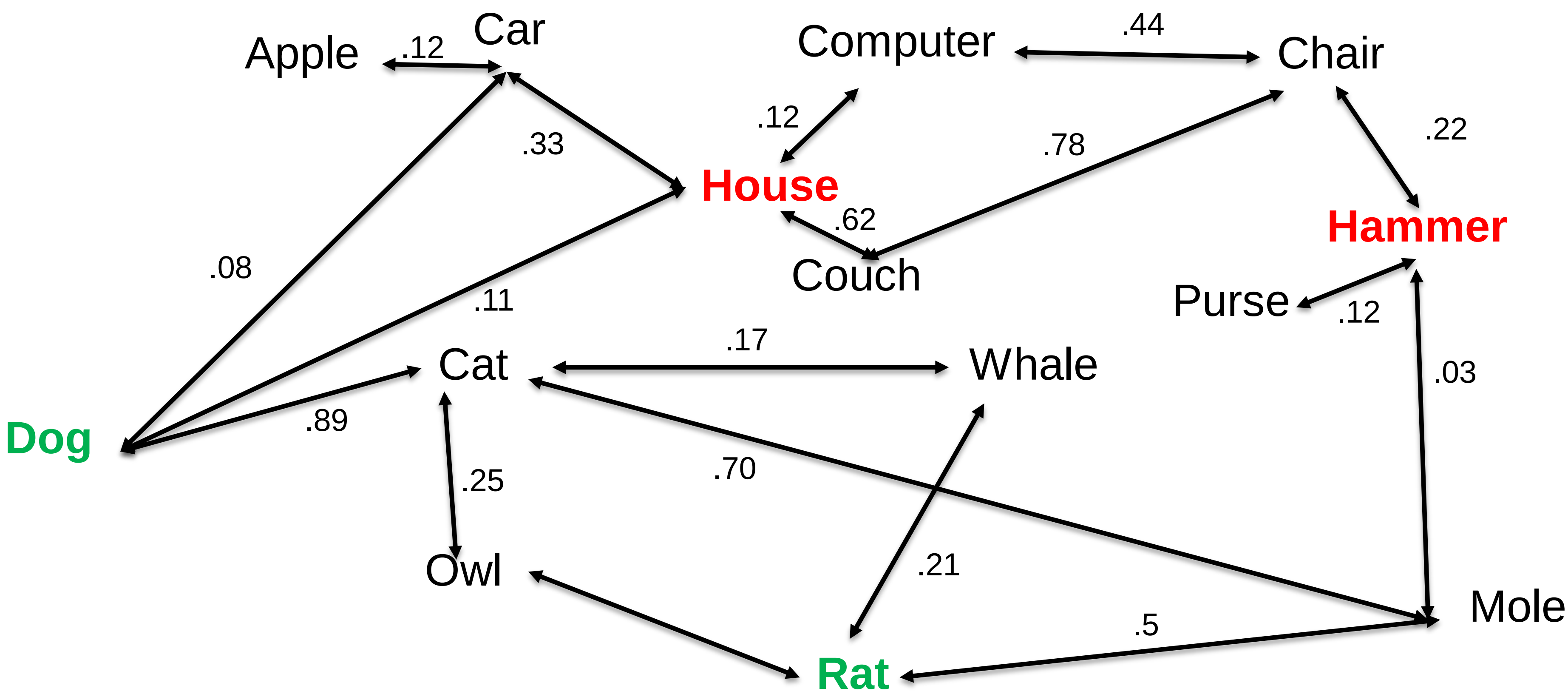
- Classify nouns into semantic categories
  - Animals, edibles, tools, …
- For each semantic category, start with a small set of positive and negative examples
  - Typically only two positive examples and two negative examples

Minimally Supervised Noun Classification

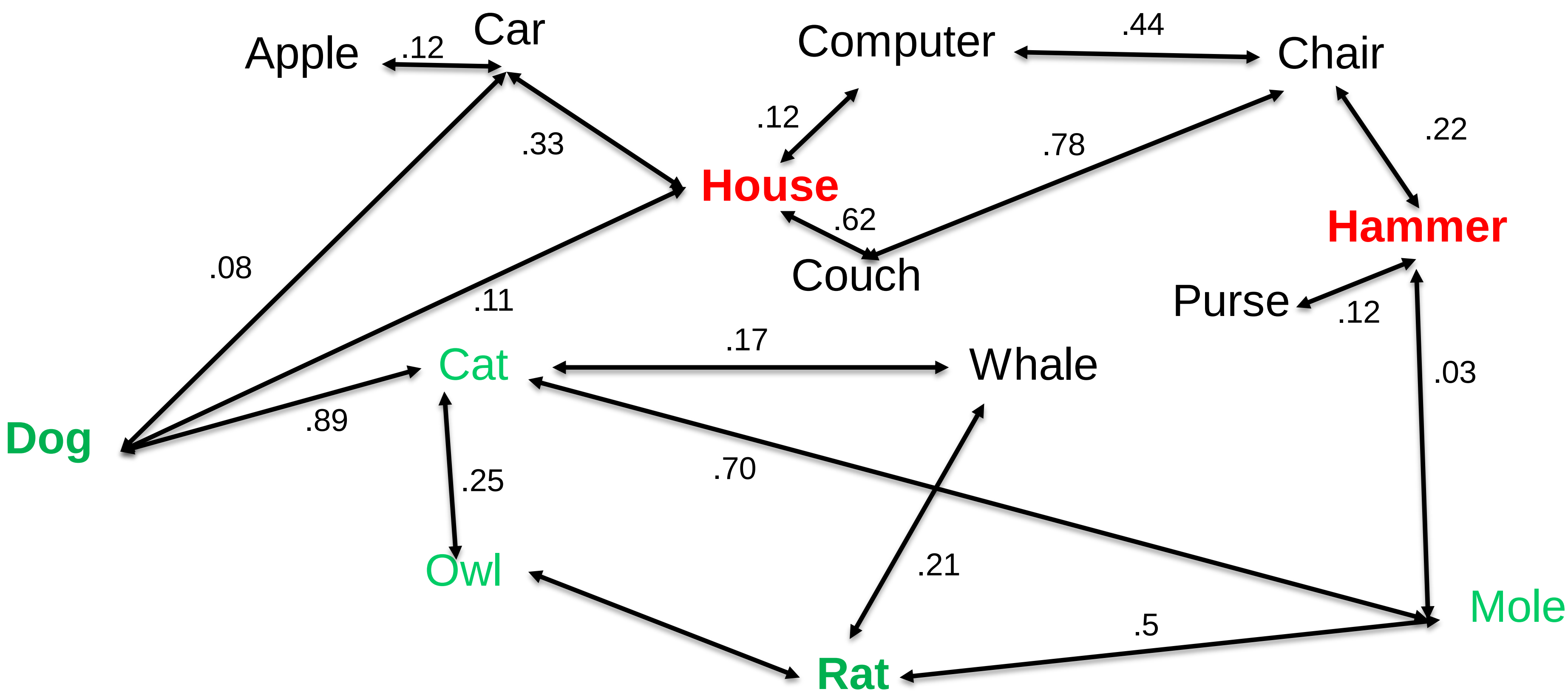
Schwartz, Reichart and Rappoport, Coling 2014

- Classify nouns into semantic categories
  - Animals, edibles, tools, …
- For each semantic category, start with a small set of positive and negative examples
  - Typically only two positive examples and two negative examples
- Link words that co-occur in **symmetric flexible patterns**

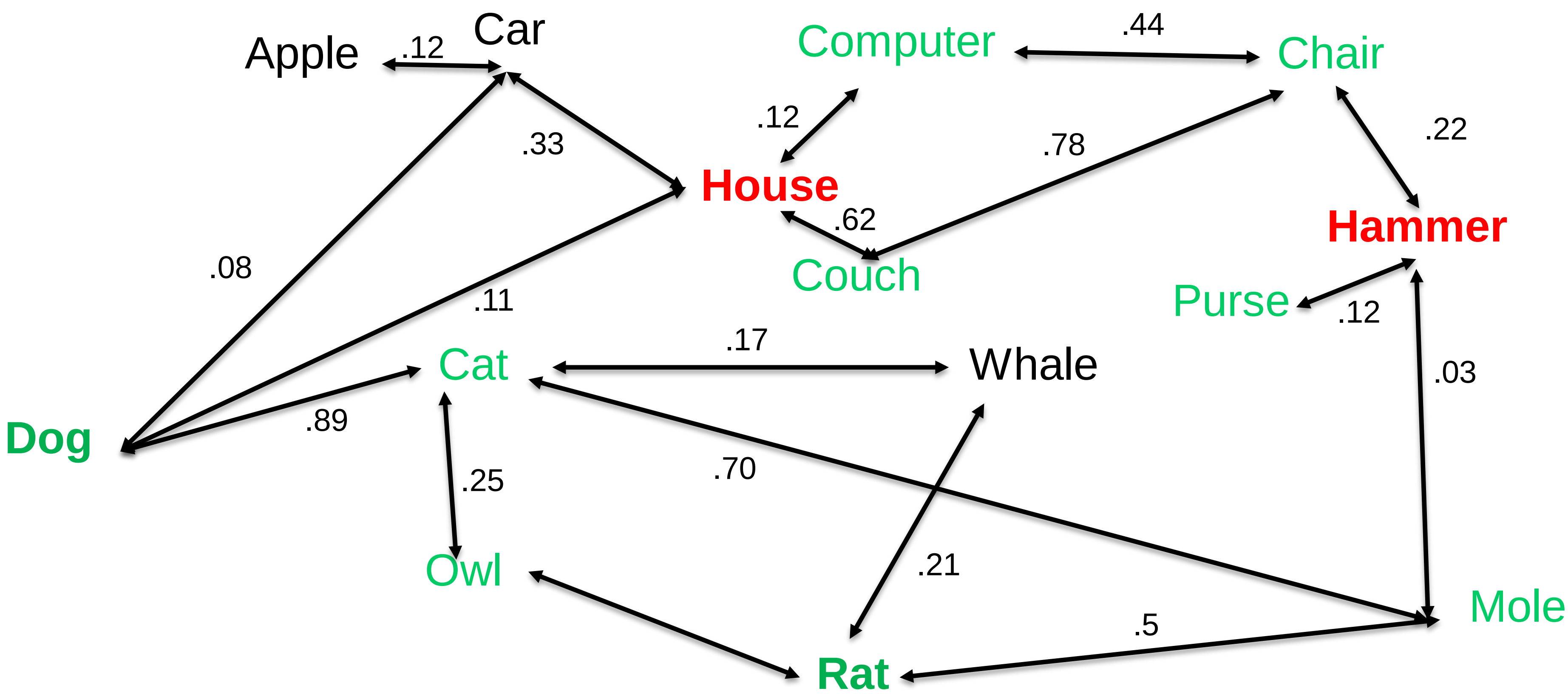
Example: Animate class



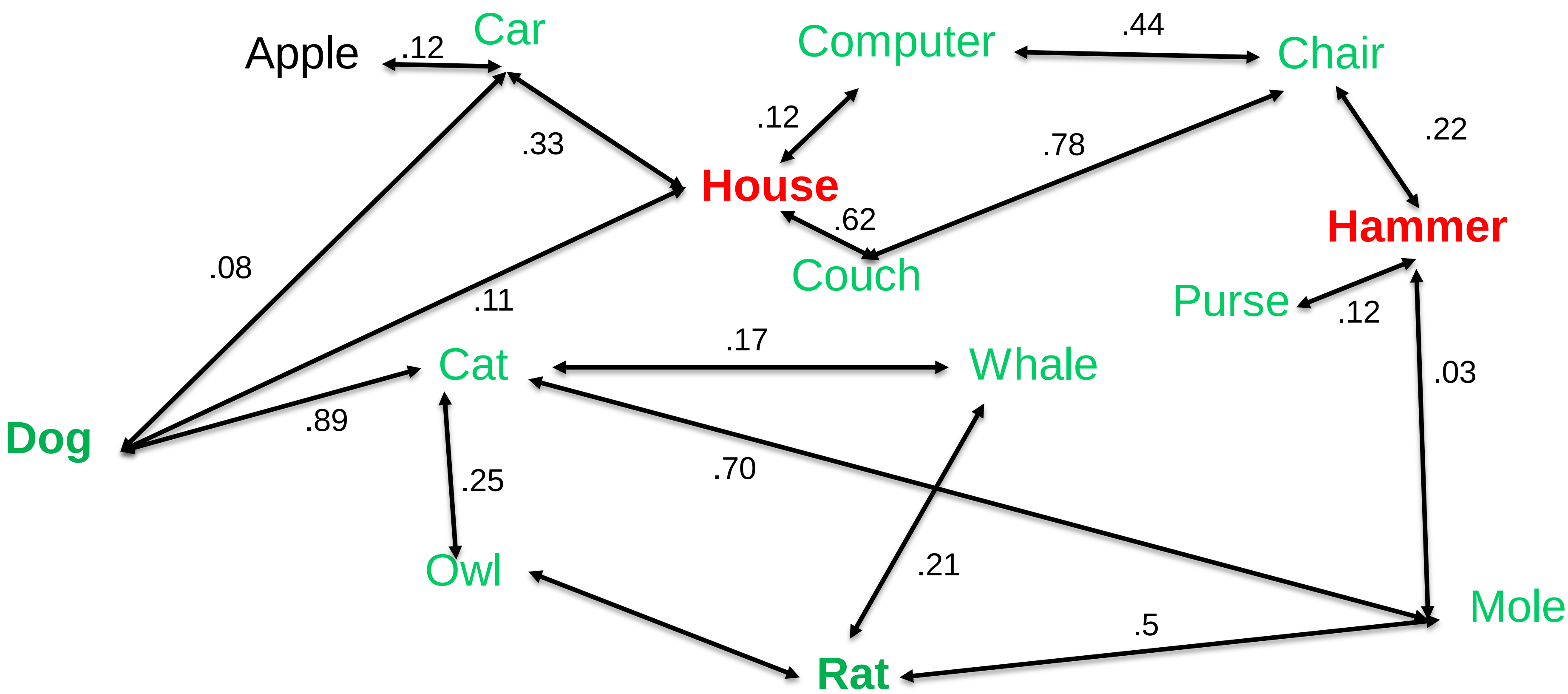
Example: Animate class



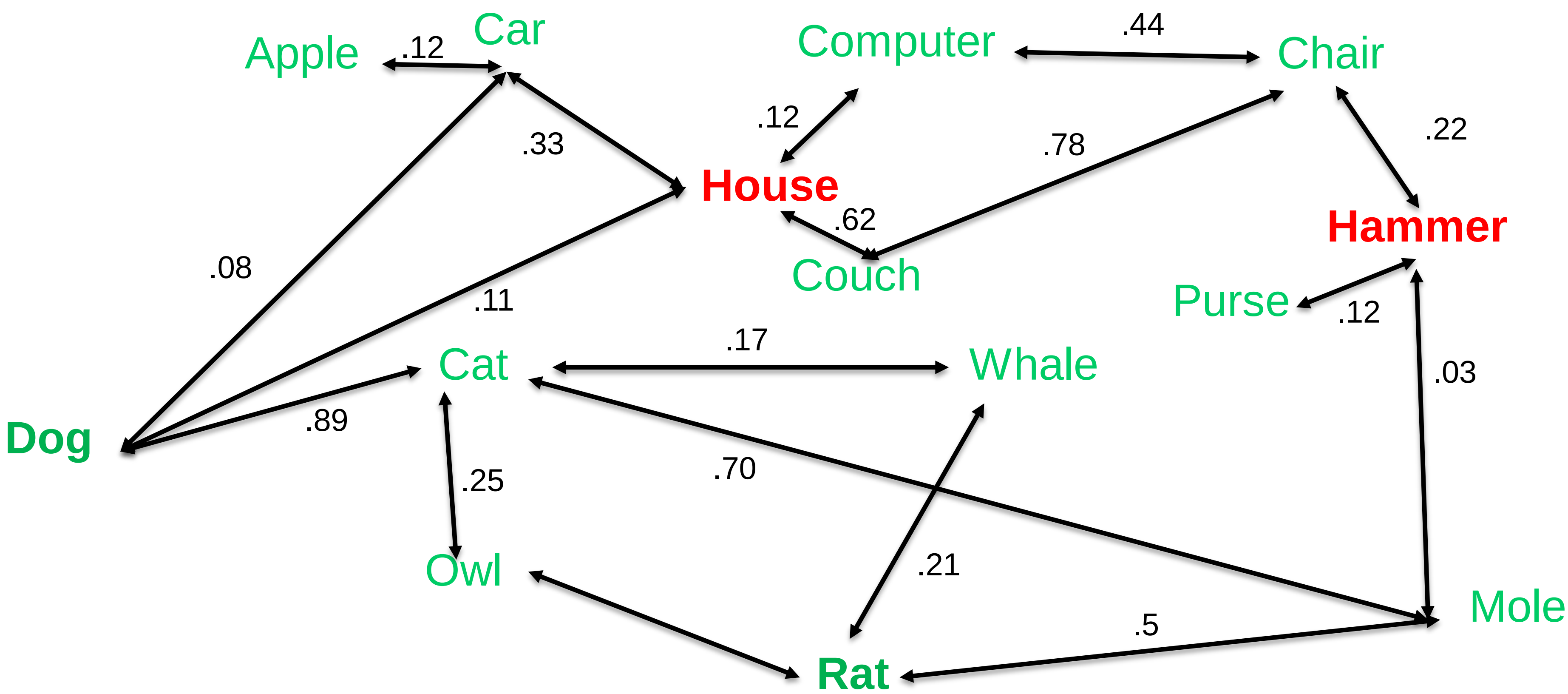
Example: Animate class



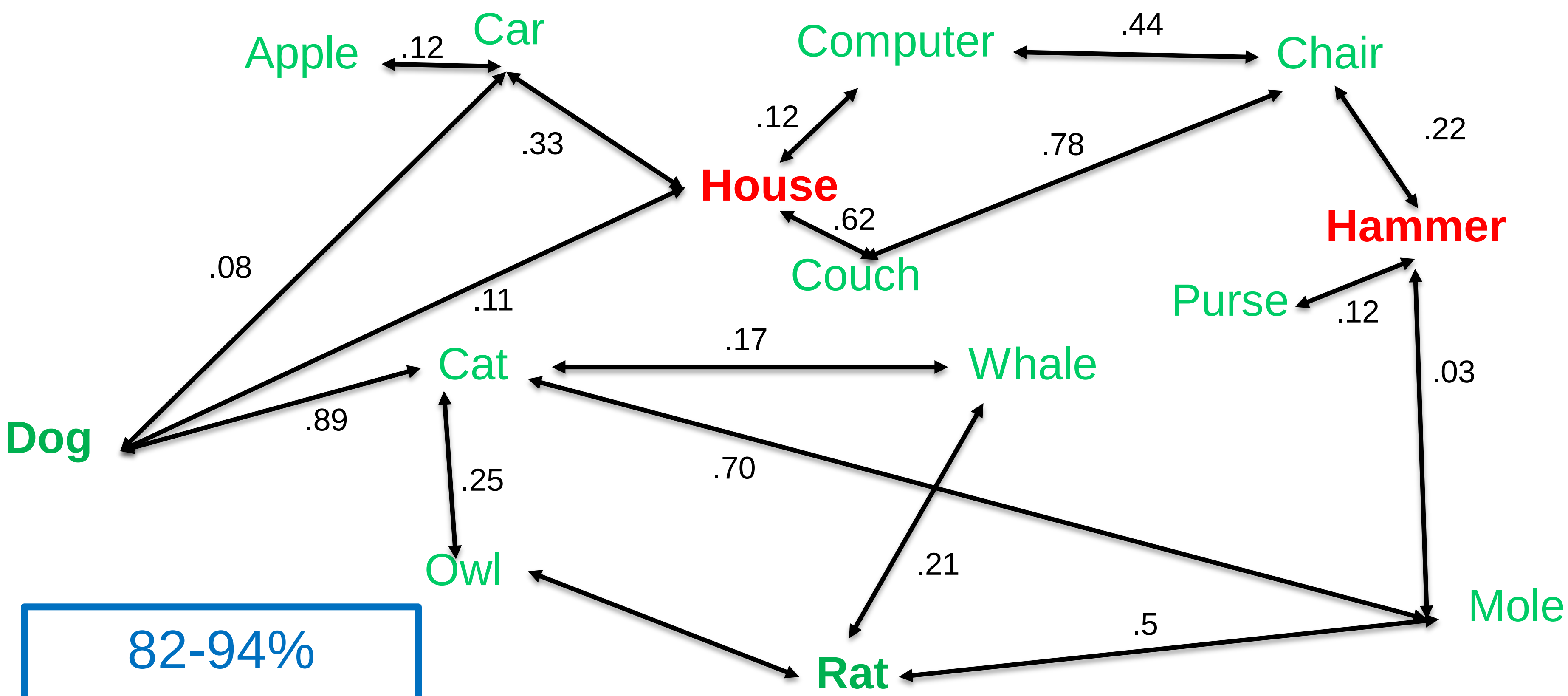
Example: Animate class



Example: Animate class



Example: Animate class



82-94% accuracy on 450 words

Interpretable Word Embeddings Using Pattern Features

Roy Schwartz, Roi Reichart and Ari Rappoport

(Under Revision)

Vector Space Models

- Representations of words as vectors of features (numbers)

$$V_{dog} = \begin{pmatrix} 0 \\ 0.5 \\ 0.76 \\ -0.12 \\ 0.76 \\ 0 \\ 0 \\ -0.51 \\ . \\ . \\ . \end{pmatrix}$$

Vector Space Models

- Representations of words as vectors of features (numbers)

- Features are usually bag-of-words counts
  - Directly or via some mathematical transformation

$$V_{dog} = \begin{pmatrix} 0 \\ 0.5 \\ 0.76 \\ -0.12 \\ 0.76 \\ 0 \\ 0 \\ -0.51 \\ . \\ . \\ . \end{pmatrix}$$

Vector Space Models

- Representations of words as vectors of features (numbers)

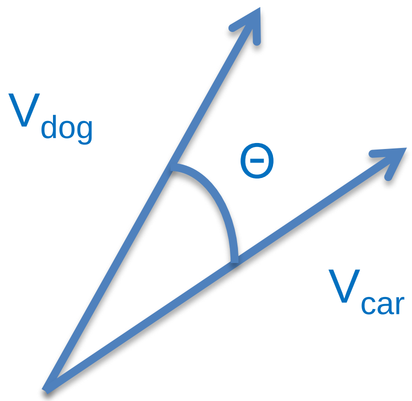
- Features are usually bag-of-words counts
  - Directly or via some mathematical transformation

- In recent years, deep neural network models have been applied to generate accurate vector representations (aka **word embeddings**)
  - Bengio, 2003; Collobert, 2008 & 2011, word2vec (Mikolov 2013{a,b,c})

$$\mathbf{V}_{\mathbf{dog}} = \begin{pmatrix} 0 \\ 0.5 \\ 0.76 \\ -0.12 \\ 0.76 \\ 0 \\ 0 \\ -0.51 \\ . \\ . \\ . \end{pmatrix}$$

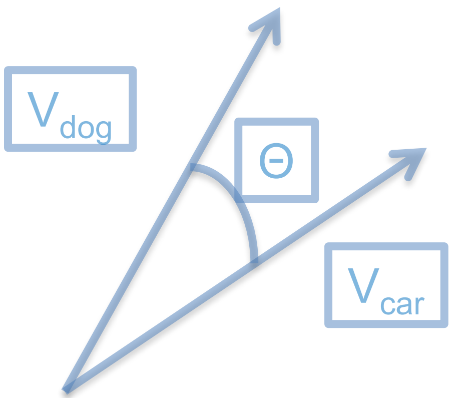
Word Embeddings (Cool!) Properties

- (accurate) Word similarity

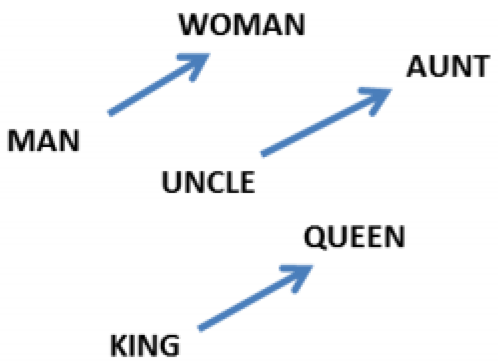


Word Embeddings (Cool!) Properties

- (accurate) Word similarity



- Word analogy



(Mikolov et al., 2013)

Word Embeddings Applications

- Information Retrieval
- Document Classification
- Question Answering
- Named Entity Recognition
- Parsing
- …

Word Embeddings Limitations

- Resulting vectors are highly **uninterpretable**
  - Sequences of several hundred numbers
  - Not clear what each number represents

Word Embeddings Limitations

- Resulting vectors are highly **uninterpretable**
  - Sequences of several hundred numbers
  - Not clear what each number represents

- Restricted to a limited set of relations
  - Similarity/Relatedness, some analogies
  - Other relations are not supported: hyponymy (animal → dog), antonymy (big/tall), etc.

Symmetric Patterns to Word Embeddings

- Input: a large corpus C (8G words)

Symmetric Patterns to Word Embeddings

- Input: a large corpus C (8G words)

  Extract a set of SPs *P* using the DR06 algorithm

Symmetric Patterns to Word Embeddings

- Input: a large corpus C (8G words)

  Extract a set of SPs *P* using the DR06 algorithm

- Traverse C, extract all instances of all *p in P*
  - **cats** *and* **dogs**
  - **House** *and the* **rooms**
  - *from* **France** *to* **England**
  - …

Symmetric Patterns to Word Embeddings (2)

- For each word *w* in the lexicon, build a count vector (V_w) of all other words that co-occur with *w* in SPs

Symmetric Patterns to Word Embeddings (2)

- For each word *w* in the lexicon, build a count vector (V_w) of all other words that co-occur with *w* in SPs

- orange
  1. … apples and oranges …
  2. … oranges as well as grapes

     …

  K. … neither banana nor orange

- China
  1. … Japan or China …
  2. … China rather than Korea

     …

  M. … Vietnam and China …

Symmetric Patterns to Word Embeddings (3)

- Compute the Positive Pointwise Mutual Information (PPMI) between each pair of words

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

$$PPMI(w_i, w_j) = \begin{cases} PMI(w_i, w_j) < 0 : 0 \\ otherwise : PMI(w_i, w_j) \end{cases}$$

The Result: **Interpretable** Word Embeddings based on Symmetric Patterns

$$V^{sp}_{\mathbf{dog}} = \begin{pmatrix} \text{PPMI(dog,house)} \\ \text{PPMI(dog,mouse)} \\ \text{PPMI(dog,zebra)} \\ \text{PPMI(dog,wine)} \\ \text{PPMI(dog,cat)} \\ \text{PPMI(dog,dolphin)} \\ \text{PPMI(dog,bottle)} \\ \text{PPMI(dog,pen)} \\ . \\ . \\ . \end{pmatrix}$$

The Result: **Interpretable** Word Embeddings based on Symmetric Patterns

$$V^{sp}_{\mathbf{dog}} = \begin{pmatrix} \text{PPMI(dog,house)} \\ \text{PPMI(dog,mouse)} \\ \text{PPMI(dog,zebra)} \\ \text{PPMI(dog,wine)} \\ \text{PPMI(dog,cat)} \\ \text{PPMI(dog,dolphin)} \\ \text{PPMI(dog,bottle)} \\ \text{PPMI(dog,pen)} \\ . \\ . \\ . \end{pmatrix}$$

$$|V^{SP}_{w}| = \sim 500K$$

$$E_w(|nonzero(V^{SP}_{w})|) = \sim 50$$

Interpretability

- Our model is interpretable in two different manners

Interpretability

- Our model is interpretable in two different manners

- We **understand** what each feature represents
  - The similarity score between the target word and another **word *w***

Interpretability

- Our model is interpretable in two different manners

- We **understand** what each feature represents
  – The similarity score between the target word and another **word *w***

- We understand how the value of each feature is **generated**
  – The co-occurrence score of the target word and ***w*** in **symmetric patterns**

Interpretability

- Our model is interpretable in two different manners

- We **understand** what each feature represents
  - The similarity score between the target word and another **word *w***

- We understand how the value of each feature is **generated**
  - The co-occurrence score of the target word and ***w*** in **symmetric patterns**

- Interpretability can be exploited to **improve our model**

Antonyms

big / small

Antonyms

big / small

- Antonyms appear in similar contexts
  - Here is a X car
  - I live in a X house

Antonyms

big / small

- Antonyms appear in similar contexts
  - Here is a X car
  - I live in a X house

➔ In typical word embeddings, $\cos(V_{big}, V_{small})$ is high

Antonyms

big / small

- Some symmetric patterns are indicative of antonymy*
  - "*either* X *or* Y" (*either* big *or* small), "*from* X *to* Y" (*from* poverty *to* richness)

* Lin et al. (2003)

Antonyms

- A variant of our model that assigns dissimilar vectors to antonym pairs

Antonyms

- A variant of our model that assigns dissimilar vectors to antonym pairs

- For each word *w*, compute V_w^{AP} similarly to V_w^{SP}, but using the set of antonym patterns

$$V_w^{\mathrm{AP}'} = V_w^{\mathrm{SP}} - \beta \cdot V_w^{\mathrm{AP}}$$

❖ β is tuned using a development set

Experiments

- Word similarity task
  - Experiments with the SimLex999 dataset (Hill et al., 2014)
  - 999 word pairs, each assigned a similarity score by human annotators
  - $f_{<model>}(w_i,w_j) = \cos(V^{<model>}_{wi},V^{<model>}_{wj})$
  - Evaluation results is the Spearman's ρ score between model and human judgments
  - Numbers are average scores of 10 folds of 25% (dev) / 75 (test) partitions
  - Baselines: 2 interpretable baselines, 3 state-of-the-art, non-interpretable baselines

Interpretable?	Model	Spearman's ρ
Non-interpretable	GloVe	0.426
	CBOW	0.43
	skip-gram	0.462
Interpretable	BOW	0.423
	NNSE	0.455
	$SP^{(+)}$	**0.517**

Experiments

- Word similarity task
  - Experiments with the SimLex999 dataset (Hill et al., 2014)
  - 999 word pairs, each assigned a similarity score by human annotators
  - $f_{\text{<model>}}(w_i,w_j) = \cos(V^{\text{<model>}}_{wi}, V^{\text{<model>}}_{wj})$
  - Evaluation results is the Spearman's ρ score between model and human judgments
  - Numbers are average scores of 10 folds of 25% (dev) / 75 (test) partitions
  - Baselines: 2 interpretable baselines, 3 state-of-the-art, non-interpretable baselines

Interpretable?	Model	Spearman's ρ
Non-interpretable	GloVe	0.426
	CBOW	0.43
	skip-gram	0.462
Interpretable	BOW	0.423
	NNSE	0.455
	SP$^{(+)}$	**0.517**

Experiments

- Word similarity task
  - Experiments with the SimLex999 dataset (Hill et al., 2014)
  - 999 word pairs, each assigned a similarity score by human annotators
  - $f_{<model>}(w_i,w_j) = \cos(V^{<model>}_{wi},V^{<model>}_{wj})$
  - Evaluation results is the Spearman's ρ score between model and human judgments
  - Numbers are average scores of 10 folds of 25% (dev) / 75 (test) partitions
  - Baselines: 2 interpretable baselines, 3 state-of-the-art, non-interpretable baselines

Interpretable?	Model	Spearman's ρ
Non-interpretable	GloVe	0.426
	CBOW	0.43
	$\boxed{\text{skip-gram}}$	$\boxed{0.462}$
Interpretable	BOW	0.423
	NNSE	0.455
	$\boxed{\mathbf{SP}^{(+)}}$	$\boxed{\mathbf{0.517}}$

Antonyms

Word Pair	*SP*		SG
	+AN	-AN	
new - old	1	6	6
narrow - wide	1	7	8
necessary - unnecessary	2	2	9
bottom - top	3	8	10
absence - presence	4	7	9
receive - send	1	9	8
fail - succeed	1	8	6

Joint Model

$$f_{joint}(w_i, w_j) = \gamma \cdot f_{SP}(w_i, w_j) + (1-\gamma) \cdot f_{skip\text{-}gram}(w_i, w_j)$$

Interpretable?	Model	Spearman's ρ
Non-interpretable	GloVe	0.426
	CBOW	0.43
	skip-gram	0.462
Interpretable	BOW	0.423
	NNSE	0.455
	SP$^{(+)}$	**0.517**
Joint		**0.563**
Average Human Score		0.651

❖ γ determined using a development set

Joint Model

$$f_{joint}(w_i, w_j) = \gamma \cdot f_{SP}(w_i, w_j) + (1 - \gamma) \cdot f_{skip\text{-}gram}(w_i, w_j)$$

Interpretable?	Model	Spearman's ρ
Non-interpretable	GloVe	0.426
	CBOW	0.43
	skip-gram	0.462
Interpretable	BOW	0.423
	NNSE	0.455
	SP$^{(+)}$	**0.517**
Joint		**0.563**
Average Human Score		0.651

❖ γ determined using a development set

Patterns

- Highly effective **computational** tool
  - High quality results (either unsupervised or very weakly supervised)

- Simple to understand and implement
  - Can be implemented in computer hardware

Patterns and Language Acquisition

- Children learn linguistic structures, among others, through pattern-finding in their discourse interactions with others (Tomasello, 2003)

Summary

- Patterns are useful for extracting semantic information

- Symmetric patterns are as useful (actually more useful) as state-of-the-art word embeddings in modeling word similarity
  - 5–9.4 points gap

- Patterns can capture relations that word embeddings cannot
  - Antonymy

- SPs can be combined along with state-of-the-art embeddings to create an even more accurate representation
  - 10.1 points higher than state-of-the-art

Current Work: Asymmetry of Symmetric Patterns

- Symmetric patterns are not really symmetric
  - good or bad >> bad or good, more or less >> less or more
  - Order of binomials (Bunin Benor and Levy, 2006)
- In a large majority of the cases, positive word comes before negative
- Application: polarity induction

roys02@cs.huji.ac.il

http://www.cs.huji.ac.il/~roys02/