Reminder: previous lecture

EVENT SEMANTICS

- This idea is called event semantics (or Davidsonian semantics; Davidson, 1967).
  - 'Mark drove' -> ∃d(drive(d,mark) & past(d))
  - 'Mark drives quickly' -> ∃d(drive(d,mark) & quickly(d))
- Parsons (1990) proposed an updated form where thematic roles (or semantic roles) are explicit.
- In 'Brutus stabbed Caesar' (active voice)
  - The subject is Brutus, and the object is Caesar
- In 'Caesar was stabbed by Brutus' (passive voice)
  - The subject is Caesar, and the object is Brutus
- Even if the grammatical roles have switched, the semantic roles have not!

Reminder: previous lecture

SEMANTIC ROLES

- Every event has a set of semantic or thematic roles.
- The exact set of roles is debated, but here are some candidates:
  - Agent: the performer of the action (e.g., 'Brutus stabs')
  - Patient: the thing undergoing the action that changes state (e.g., 'Brutus stabs Caesar')
  - Theme: the thing undergoing the action but does not change state (e.g., 'I gave them the food')
  - Instrument
  - Location
  - Time
  - etc...

Reminder: previous lecture

SEMANTIC ROLE LABELING

- Semantic role labeling is the task of identifying the semantic roles for a given sentence and verb.
  - Sometimes the verb is not specified.
- Example input:

  'The batter hit the ball yesterday.'
- Example output:

  '[$_{\text{agent}}$The batter] hit [$_{\text{patient}}$the ball] [$_{\text{time}}$yesterday].'
- Not really full compositional semantic parsing,
  - But a "shallow" form of it.
  - Can be thought of as something between syntactic and semantic parsing.

Reminder: previous lecture

EVENT SEMANTICS

- 'Caesar was stabbed by Brutus'
  - Caesar is the patient (or theme)
  - Brutus is the agent
- In Davidsonian semantics, we would write this as:
  - ∃e.stab(e,brutus,caesar)
  - But how would we write the meaning of 'Caesar was stabbed'?
- In neo-Davidsonian semantics, we separate the semantic roles explicitly:
  - ∃e(stab(e) & agent(e,brutus) & patient(e,caesar))
  - ∃e(stab(e) & arg1(e,brutus) & arg2(e,caesar))
  - ∃e(stab(e) & arg1(e)=brutus & arg2(e)=caesar)

Reminder: previous lecture

LOGICAL FORMALISMS

- First-order logic is not just one monolithic logical formalism.
  - There are different ways we can use FOL to represent meaning.
  - Each with advantages and disadvantages.
- What makes a good logical formalism/meaning representation?
  - Coverage:
    - If there are two sentences with different meanings, they should have different logical forms.
    - If there are multiple readings/interpretations of the same sentence, there should be a logical form for each reading.

Reminder: previous lecture

SEMANTIC AMBIGUITY

- We have seen ambiguity in syntax:
  - 'Sally caught a butterfly with a net.'
  - 'Sally caught a butterfly with a stripe.'
- We can also have ambiguity in semantics:
  - 'The trophy didn't fit in the suitcase because it's too big.'
  - 'The trophy didn't fit in the suitcase because it's too small.'
  - These sentences have the same syntactic structure.
  - Coreference resolution:
    - Does 'it' refer to the same thing as 'trophy'?
    - Or to the same thing as 'suitcase'?
- A good logical formalism will have 2 LFs for each sentence.

Reminder: previous lecture

SEMANTIC AMBIGUITY

- Lexical ambiguity is another source of ambiguity in semantics:
  - 'I went to the bank to withdraw cash.'
  - 'I went to the bank to catch some fish.'
- The word 'bank' refers to the financial institution sense in the first sentence.
  - Whereas it refers to a riverbank in the second.
- A good logical formalism will produce two logical forms for each of the above two sentences:
  - 1 LF for the financial institution sense,
  - And 1 LF for the riverbank sense.
- The logical formalism itself has no background knowledge of the world.
  - Maybe there's a possible world where you withdraw money near rivers.

Reminder: previous lecture

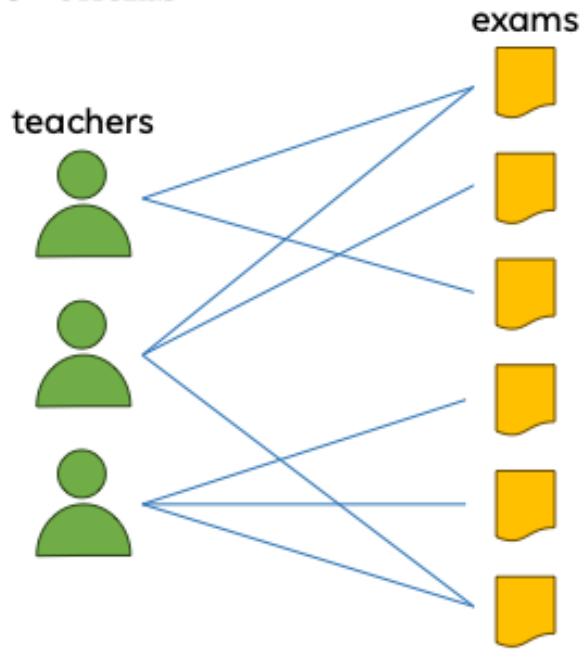
SEMANTIC AMBIGUITY

- Quantifier scope ambiguity is another source of ambiguity in semantics:
  - 'Every dog chases a cat.'
- There are two valid readings:
  - ∀d(dog(d) -> ∃c(cat(c) & chase(d,c)))
  - ∃c(cat(c) & ∀d(dog(d) -> chase(d,c)))
- The only difference is the order of the quantifiers,
  - But there is a stark difference in meaning.
- Negation scope ambiguity: 'All that glitters is not gold.'
  - ∀g(glitter(g) -> ¬gold(g))
  - ¬∀g(glitter(g) -> gold(g))

Reminder: previous lecture

SEMANTIC AMBIGUITY

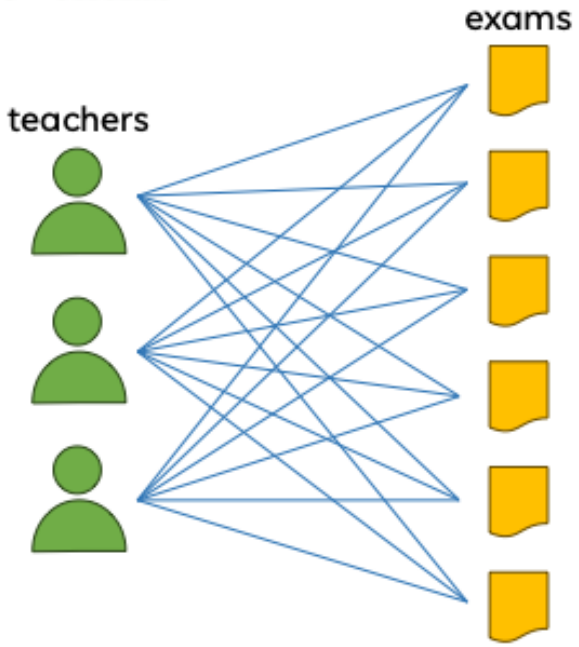
- More extreme example of quantifier scope ambiguity:
  - '3 teachers graded 6 exams'



Reminder: previous lecture

SEMANTIC AMBIGUITY

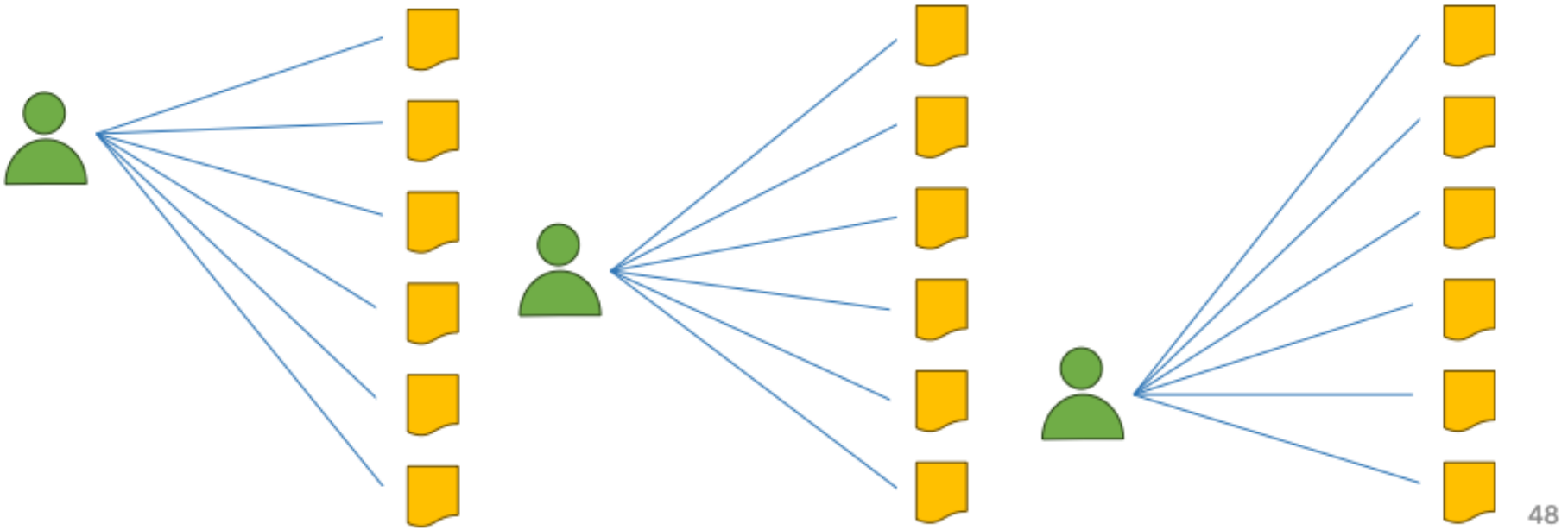
- More extreme example of quantifier scope ambiguity:
  - ‘3 teachers graded 6 exams’



Reminder: previous lecture

SEMANTIC AMBIGUITY

- More extreme example of quantifier scope ambiguity:
  - ‘3 teachers graded 6 exams’



SEMANTIC AMBIGUITY

"A woman gives birth to a child every thirty seconds.

We must stop that woman!"

$$\forall t(\text{30-sec-interval}(t) \rightarrow \exists w(\text{Woman}(w) \wedge \text{GivesBirth}(w, t)))$$

$$\exists w(\text{Woman}(w) \wedge \forall t(\text{30-sec-interval}(t) \rightarrow \text{GivesBirth}(w, t)))$$

Deep dive into lexical semantics

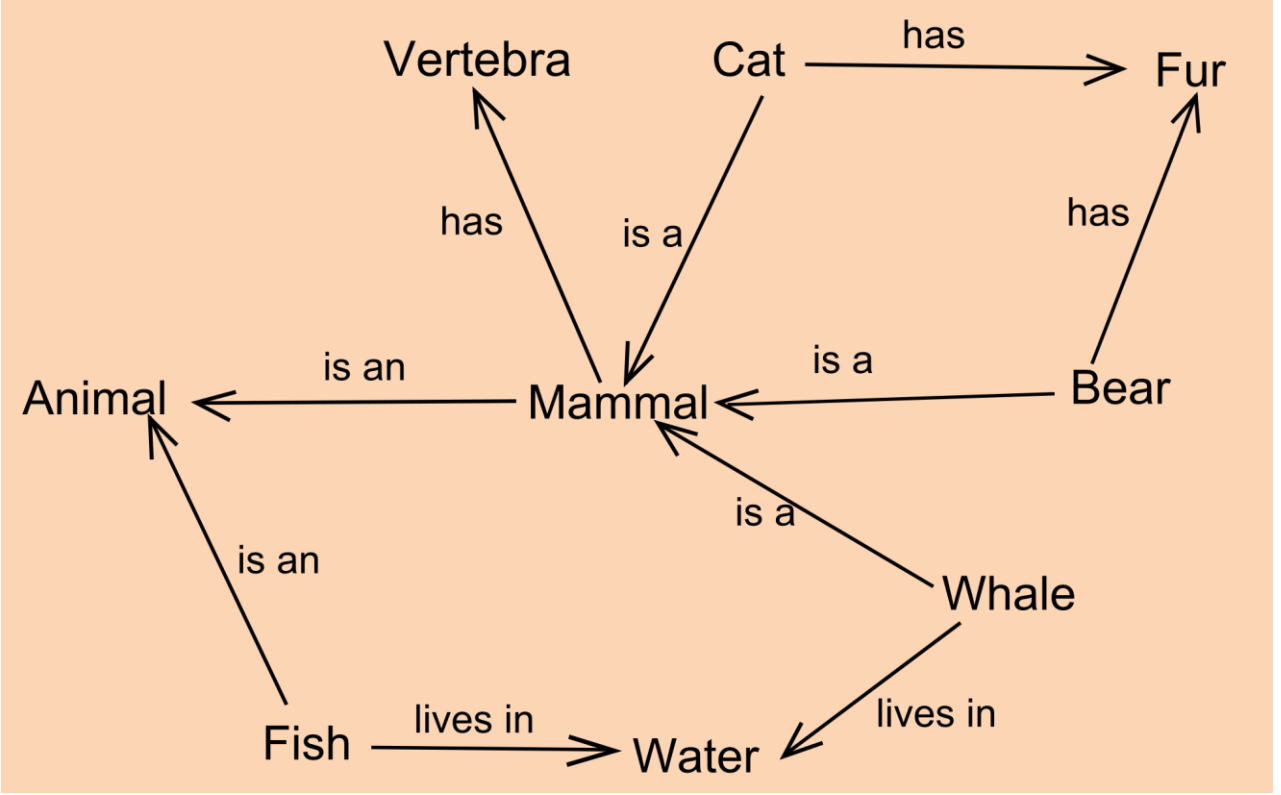
*We make the distinction between **lexical semantics**, the meaning of **words** and their relationships, and **compositional semantic,** capturing how meaning of complex NL expressions are composed from the meaning of their parts.*

***Lexical semantics** extends beyond a purely linguistic questions.* **It captures our understanding of the world.**

- *What are the entities the "world" is comprised of?*
- *What are the relationships between these entities?*

Lexical semantics captures how this information maps to the way we use language, specifically – individual words.

Lexical Semantics



The Secret Lives of Words!

- First some definitions –
- **Word form** : inflected word appearing in text
- **Lemma**: stem of the word

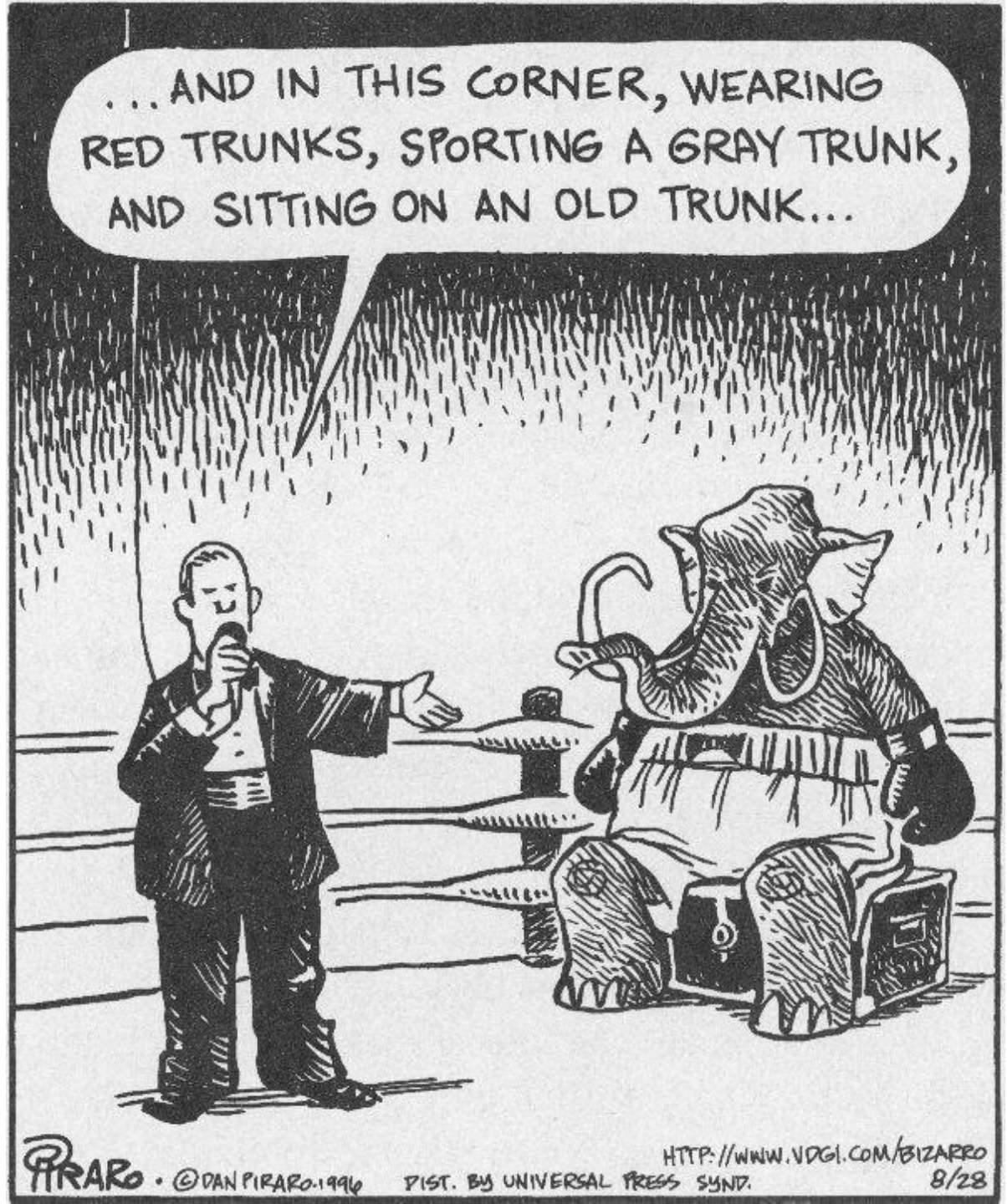
- *Several word forms will have the same lemma*
  - *Banking, Banked, Banks*

- *Do all of these have the same **meaning**?*

The Secret Lives of Words!

- **Lemmas can mean different things –**
  - John waited by the river bank.
  - John waited by the River bank.

- The word “bank” has different **senses**
- A sense is a discrete representation of the words meaning.
- **Homonymy:** words that share a form but have unrelated meanings


...AND IN THIS CORNER, WEARING RED TRUNKS, SPORTING A GRAY TRUNK, AND SITTING ON AN OLD TRUNK...
PIRARO • ©DAN PIRARO·1996
HTTP://WWW.VDGI.COM/BIZARRO
DIST. BY UNIVERSAL PRESS SYND.
8/28

Ok, so what?

the spirit is willing but the flesh is weak

Russian

English

The Vodka is good, but the meat is rotten.

Disclaimer: "MT Myth", but still a nice example..

The Secret Lives of Words!

- “The bank is the oldest building in Lafayette. It opened in 1852”
- “The bank refused John’s loan”

- Polysemy: word that has several **related** meanings
- Happens systematically:
  - Building-organization, Food-animal, author-book

The Secret Lives of Words!

- “I sat of the sofa, it was big”
- “I sat on the couch, it was large”

- Words that have similar meaning are **synonyms**
- **There are often nuanced differences:**
  - “Garbage can” vs. “Rubbish bin”
  - “Water” vs. H_2O
  - “Big” vs. “large” (My *big/large* brother)

The Secret Lives of Words!

- Words with opposite meanings are **antonyms**.
  - Short – Long
  - Big – Small
- Words are **hyponyms** if one word is a subclass of the other.
  - *Car is a **hyponym** of vehicle.*
- The other direction is called a **hypernym**
  - *Vehicle is a **hypernym** of a car.*

The Secret Lives of Words!

- Hyponyms define a IS-A hierarchy
  - A cat is-a mammal is-an animal.
  - Hyponyms are transitive:

  If cats are mammals **AND** mammals are animals

  Then a cats are animals.
- A very useful resource: WordNet
  - A comprehensive hierarchy of concepts
  - *New York is-a city*

https://wordnet.princeton.edu/

WordNet

- Lexical database organized hierarchically
- Defines the possible senses of each word
- WordNet provides a ***SynSet*** for each word

Noun

- S: (n) **bank** (sloping land (especially the slope beside a body of water)) *"the pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"*
- S: (n) depository financial institution, **bank**, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) *"he cashed a check at the bank"; "that bank holds the mortgage on my home"*
- S: (n) **bank** (a long ridge or pile) *"a huge bank of earth"*

WordNet Noun Relations

Relation	Also called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	*breakfast*^{1} → *meal*^{1}
Hyponym	Subordinate	From concepts to subtypes	*meal*^{1} → *lunch*^{1}
Member Meronym	Has-Member	From groups to their members	*faculty*^{2} → *professor*^{1}
Has-Instance		From concepts to instances of the concept	*composer*^{1} → *Bach*^{1}
Instance		From instances to their concepts	*Austen*^{1} → *author*^{1}
Member Holonym	Member-Of	From members to their groups	*copilot*^{1} → *crew*^{1}
Part Meronym	Has-Part	From wholes to parts	*table*^{2} → *leg*^{3}
Part Holonym	Part-Of	From parts to wholes	*course*^{7} → *meal*^{1}
Antonym		Opposites	*leader*^{1} → *follower*^{1}

Lexical Semantics

- It's convenient to think about WordNet as "ground truth"
- We can define Lexical semantics tasks, with respect to WordNet:
- Given a sentence, can you:
  - **Determine the right sense of each word?**
  - **Answer questions?**
    - *Identify synonyms, or other relations*

Word Similarity

- It's often more realistic to discuss word **similarity** instead of synonyms.
  - **Synonym**: binary relationship
  - **Similarity**: "soft" assignment
    - Sim(w1,w2) ~ 1 if words are synonyms
    - Sim(w1,w2) > 0, if words are related
- For example – IR engines need to identify similarity between content and query terms.

Word Similarity

Two broad approaches:

- **Thesaurus-based algorithms**.
  - Assume a comprehensive knowledge base (e.g., wordnet)
  - Do words appear nearby in the hypernym hiearchy? Similar definition?
- **Distributional algorithms.**
  - Assume a large collection of text (***not annotated!***)
  - Do the words appear in similar contexts?

Hypernym path based Similarity

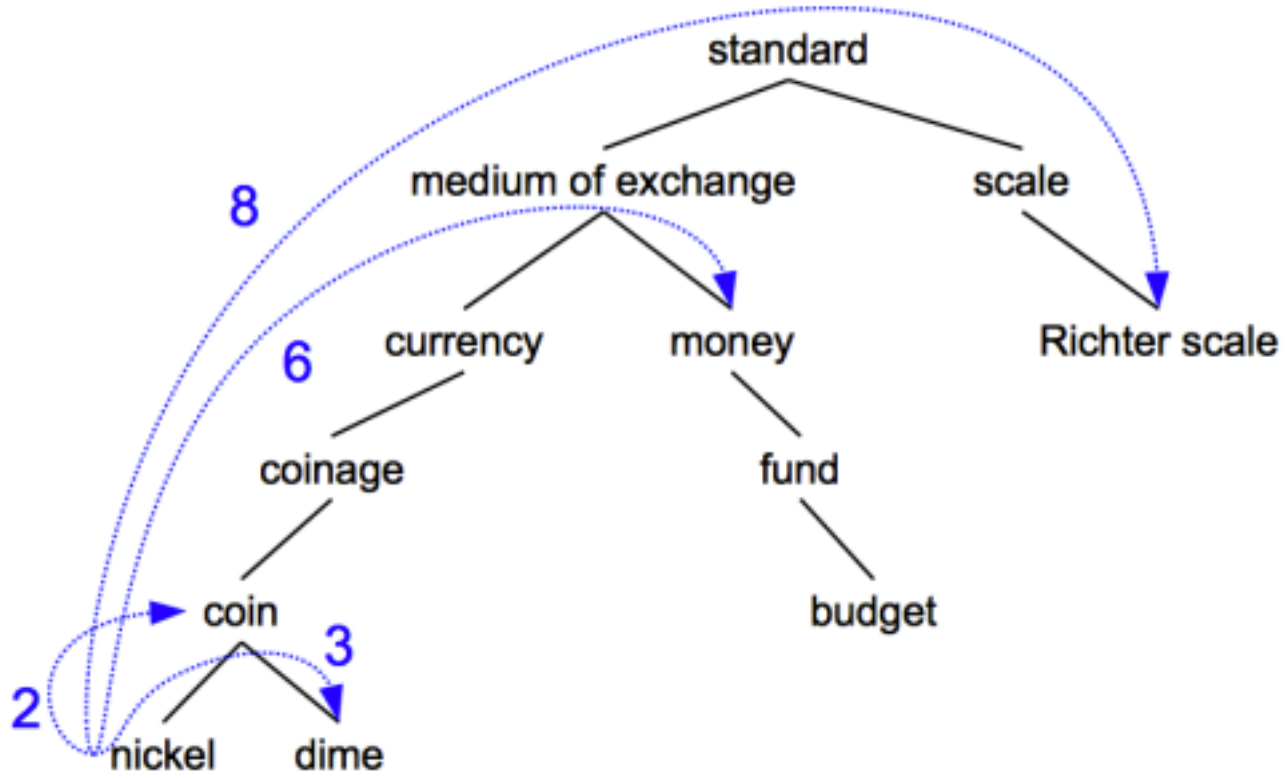


Image taken from D Jurafsky Slides on Lexical similarity

Hypernym path based Similarity

- The simple heuristic assumes uniform cost for each hop.
  - Nodes higher in the hierarchy are more abstract.
  - Ignore content of word definition
- Several works looked into improving it :
  - Resnik’95, Lin’98, Lesk’s algorithm.

Hypernym path based Similarity

- **Pros**
  - Simple, exploits existing knowledge
  - Tends to have ***high precision***.
- **Cons**
  - Depends on language specific knowledge
  - Does not evolve with language (***low recall***)

tesgüino

Distributional Similarity

A bottle of tesgüino is on the table
Everybody likes tesgüino
Tesgüino makes you drunk
We make tesgüino out of corn.

Question:
What is tesgüino?

Firth 1957:
"You shall know a word by the company it keeps"

Example from Jurafsky and Martin

Distributional Models

- **Key idea**: *word meaning is defined by it's context.*
- This method, also known as the **Vector Space** model, maintain a vector of context words, for each word.

$$w = (f_1, f_2, f_3, f_4, \ldots f_n)$$

- Given a large corpus, maintain the context words counts for each word.
  - Define context window size.

Distributional Models

$$w = (f_1, f_2, f_3, f_4, \ldots f_n)$$

- Instead of the raw counts, we prefer to have a *normalized score*.
- ***Positive Point-wise Mutual Information***

$$\mathbf{PMI(x,y)} = \log \; P(x,y) \,/\, P(x)\,P(y)$$

  - ***Intuition****: are words x,y more likely to appear together than independently?*

- ***Positive PMI****: round all negative scores to 0.*

Distributional Similarity

A bottle of tesgüino is on the table
Everybody likes tesgüino
Tesgüino makes you drunk
We make tesgüino out of corn.

Question:

What is tesgüino?

Tesgüino = (Bottle = 123, Table =54, drunk = 141, Corn = 91, …)

Bourbon = (Bottle = 231, Table =41, drunk =231, corn =121, …)

Vodka = (Bottle = 311, Table =82, drunk =321, corn =0, …)

Distributional Similarity

Given the vector based representation of words
we can compute their similarity easily -

$$\cos(v, w) = \frac{v \cdot w}{|v|\ |w|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

Using Positive PMI, ensure that Cosine similarity
will have non-negative values

Word Embedding

- Basic idea: represent words in a continuous vector space.
  - Similar idea as using PMI
- **Key difference:**
  - Find low dimensional **dense** representation
  - **Instead of counting co-occurrence, use discriminative learning methods**
    - Predict surrounding words

Word2Vec

" AI fields such as NLP, machine learning , vision, have increased in popularity in recent years"

- For each word, predict other words in window C
- **Training Objective**: maximize the probability of context word, given the current word.

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

Efficient Implementation

- For non-trivial vocabulary, the normalization factor is too costly to compute accurately.

$$p(w_o|w_i) = \frac{\exp(v'_{w_o} v_{w_i})}{\sum_{w=1}^{W} \exp(v'_w v_{w_i})}$$

- **Skip-gram with negative sampling**
  - Binary logistic regression for a small subset:
    - True pair, small subset of negative examples.

Skip-gram with Negative Sampling

- New objective function:

$$\log \sigma(v'_{w_o} v_{w_I}) + \sum_{i=1}^{k} \mathop{\mathbb{E}}_{w_i \sim P_n(w)} \left[\log \sigma(-v'_{w_i}{}^T v_{w_I})\right]$$

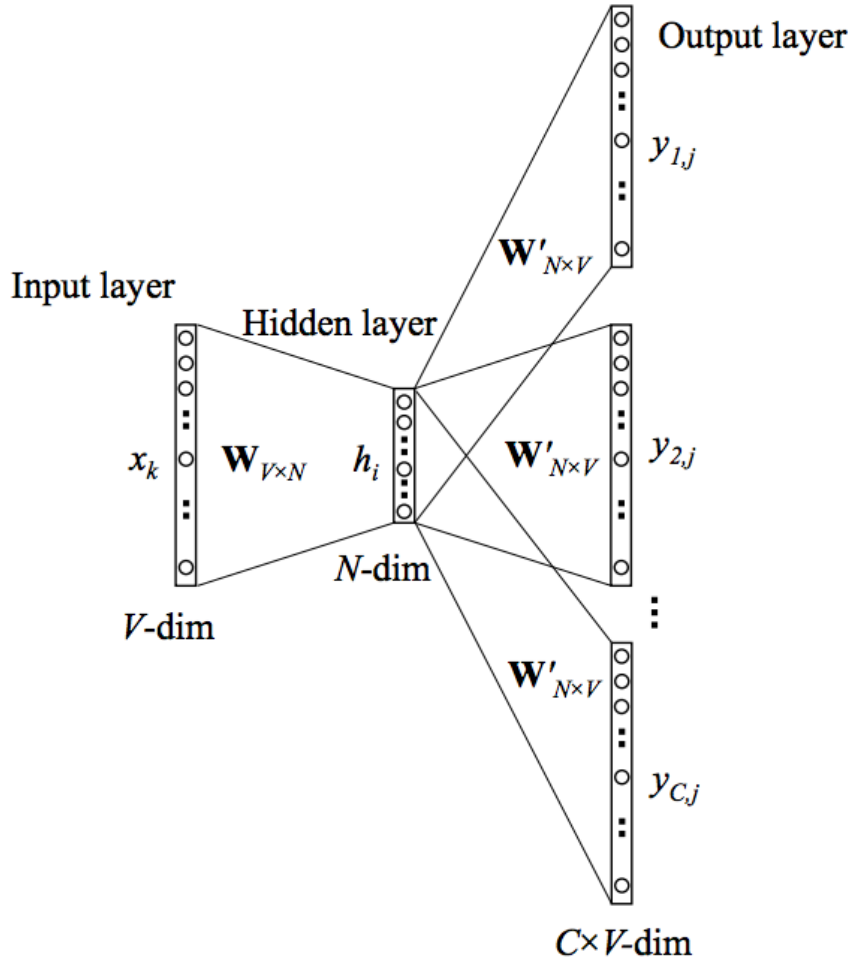
Maximize the probability of center + context words

Minimize the probability of random words

Note:

- Only pick a **small subset** of negative examples
- samples are drawn from a distribution: $P_n(w)$
- $P_n(w)$ captures unigram statistics, modified to increase the probability of sampling low frequency words.

Words**2**Vec: Skip-gram model



Mikolov et-al 2013

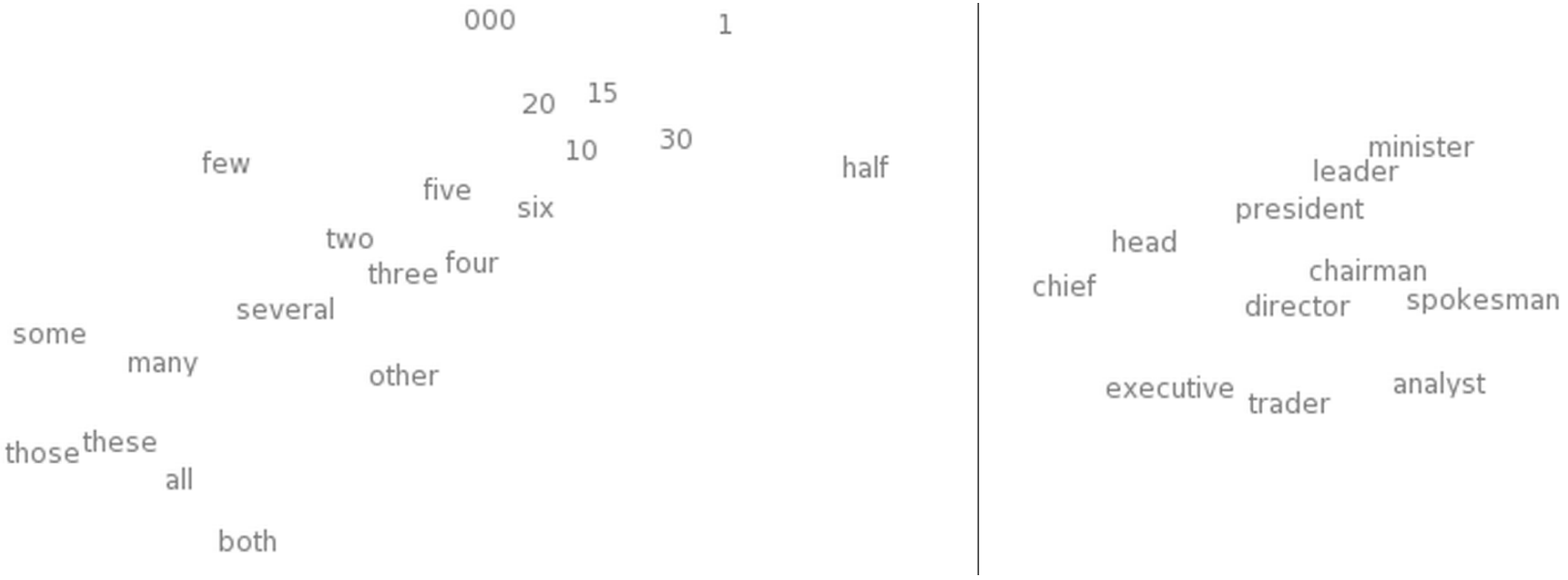
Continuous Bag-of-Words

- Very similar idea:
  - Instead of predicting context words, based on center word,
  - ***Predict center word using context words***
    - Sum up the surrounding words vectors

- Resulting word vectors capture similar information.

Word Embedding

- **Word embedding**: *move to a low dimensional, real valued dense representation of the input*
  - Key idea: similar words should have similar vectors



Turian et-al 2010

Word Embedding Evaluation

- **How should we evaluate word embeddings?**
  - What properties “good” word embeddings will have?
- Similarity
- Word relationships
  - E.g., Hyponym/Hypernymy
- Semantic relationships (e.g., analogy)
  - Paris to France as Rome is to ___

Similarity Evaluation

- Evaluation over multiple datasets using-
  - PPMI/SVD
  - WE: SGNS/Glove

Method	WordSim Similarity	WordSim Relatedness	Bruni et al. MEN	Radinsky et al. M. Turk	Luong et al. Rare Words	Hill et al. SimLex
PPMI	.755	**.697**	.745	.686	.462	.393
SVD	**.793**	.691	**.778**	.666	**.514**	.432
SGNS	**.793**	.685	.774	**.693**	.470	**.438**
GloVe	.725	.604	.729	.632	.403	.398

Levy 2015

Word2Vec

```
Enter word or sentence (EXIT to break): Chinese river

                  Word       Cosine distance
------------------------------------------
       Yangtze_River                0.667376
             Yangtze                0.644091
      Qiantang_River                0.632979
   Yangtze_tributary                0.623527
    Xiangjiang_River                0.615482
       Huangpu_River                0.604726
      Hanjiang_River                0.598110
       Yangtze_river                0.597621
         Hongze_Lake                0.594108
             Yangtse                0.593442
```

Mikolov et-al 2013

Word Representation Arithmetic

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

Mikolov et-al 2013

Word Representation Arithmetic

Paris - France + Italy = Rome

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

Mikolov et-al 2013

Evaluation Analogies

Method	Google Add / Mul	MSR Add / Mul
PPMI	.553 / .679	.306 / .535
SVD	.554 / .591	.408 / .468
SGNS	.676 / **.688**	.618 / **.645**
GloVe	.569 / .596	.533 / .580

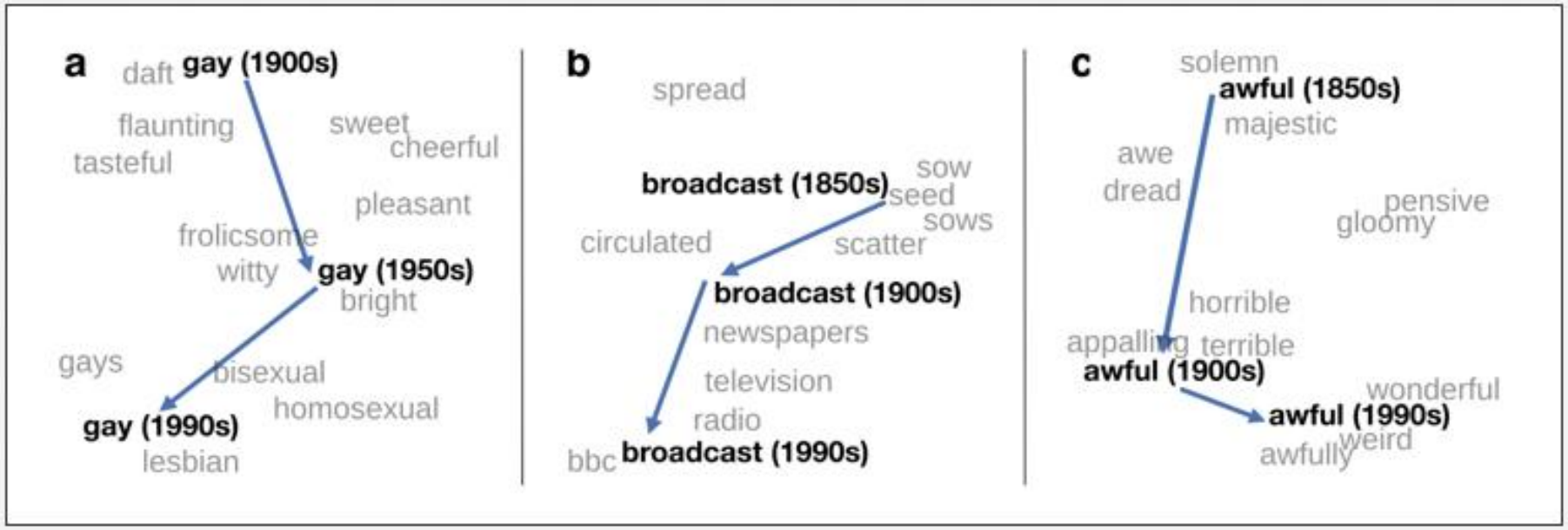
Mul = $\frac{\cos(b_2, a_2)\cos(b_2, b_1)}{\cos(b_2, a_1) + \epsilon}$

Maximizing for *b*: Add = $\cos(b, a_2 - a_1 + b_1)$

Levy et al. (2015)

Using Word Embeddings

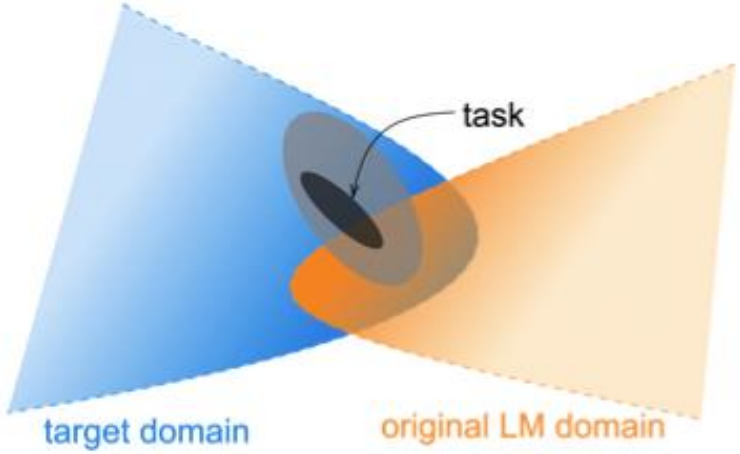
- **Option 1**: study word representation and connection using word embeddings.
  - **Can you think about possible applications?**



William L. Hamilton, et-al 2018

Using Word Embeddings

- **Option 2:** Use as input representation.
  - *Train directly over you task-specific data.*
  - *Pre-train over a large dataset, and keep fixed*
  - *Pre-train over a large dataset, fine tune to your dataset.*



	PT	News	Reviews	BioMed	CS
PT	100.0	54.1	34.5	27.3	19.2
News	54.1	100.0	40.0	24.9	17.3
Reviews	34.5	40.0	100.0	18.3	12.7
BioMed	27.3	24.9	18.3	100.0	21.4
CS	19.2	17.3	12.7	21.4	100.0

Useful way to think about it: make sure that your word representation captures the properties of the dataset you are working on

Gururangan et al 2020

Don’t stop pretraining

Domain	Task	RoBERTa	Additional Pretraining Phases		
			DAPT	TAPT	DAPT + TAPT
BioMed	ChemProt	$81.9_{1.0}$	$84.2_{0.2}$	$82.6_{0.4}$	$\mathbf{84.4}_{0.4}$
	†RCT	$87.2_{0.1}$	$87.6_{0.1}$	$87.7_{0.1}$	$\mathbf{87.8}_{0.1}$
CS	ACL-ARC	$63.0_{5.8}$	$75.4_{2.5}$	$67.4_{1.8}$	$\mathbf{75.6}_{3.8}$
	SciERC	$77.3_{1.9}$	$80.8_{1.5}$	$79.3_{1.5}$	$\mathbf{81.3}_{1.8}$
News	HyperPartisan	$86.6_{0.9}$	$88.2_{5.9}$	$\mathbf{90.4}_{5.2}$	$90.0_{6.6}$
	†AGNews	$93.9_{0.2}$	$93.9_{0.2}$	$94.5_{0.1}$	$\mathbf{94.6}_{0.1}$
Reviews	†Helpfulness	$65.1_{3.4}$	$66.5_{1.4}$	$68.5_{1.9}$	$\mathbf{68.7}_{1.8}$
	†IMDB	$95.0_{0.2}$	$95.4_{0.1}$	$95.5_{0.1}$	$\mathbf{95.6}_{0.1}$

Gururangan et al 2020

The need for **Context Dependent Embedding**

The movie was great
≈
The movie was good

This song is sick **vs**.
This patient is sick

This movie is sick **vs**.
This movie is sick

- The motivation for word embedding was to create a dense low dimensional representation for word meaning.
- Often, the meaning can vary depending on the context in which the word appears.