If you use anything from this talk, please cite it!
I have no time to write papers nowadays. Thanks!

A New Semantics for NLP: Toward Merging Propositional and Distributional Semantics

Eduard Hovy

USC Information Sciences Institute

www.isi.edu/~hovy

USC INFORMATION SCIENCES INSTITUTE

Overview

- The problem
- Merging Propositional and Distributional formalisms
- Recent work at ISI
- The problem of composition
- Some interesting thoughts
- Where next?

Propositional semantic reps

John attended the soccer Word Cup in South Africa in 2010

Logic:

(∃ e0) (attend e0 x0 x1 x2 x3)
(John x0) (soccer World Cup x1) (South Africa x2) (2010 x3)

Frame:

(e0 (:type attend)
(:agent John)
(:theme soccer-World-Cup)
(:loc South-Africa)
(:date 2010))

The green table is strong

Logic:

(∃ e0) (have-property e0 x0)
(table x0) (green x0) (strong x0)

Frame:

(x0 (:type table) (:color green) (:strength +5))

Content in semantic theories

- Semantics is expressed in *propositions* about *symbols*
- What is the meaning of the symbols?
  - De Saussure (1878) talks about the *signifier* (the signs) and the *signified* (the 'meaning')
  - Peirce (1867) talks about the representant (sign), the object (signified), and the 'meaning of the sign', represented separately (thirdness)
  - Theory of mediated reference (Frege, 1892): distinction between sense (intension) and reference (extension)
  - Theory of direct reference (Russell, 1905): meaning is equated with reference
- To date, semantic theories have focused on truth conditions and the calculation of the 'truth' or not of propositions
  - Frege, Tarski, Davidson, etc.
- But they have not really focused on representing explicitly the **elements that the propositions are <u>about</u>**
  - The propositions provide relationships among the symbols, but leave to the Denotational Model what the symbols 'mean'

The trouble with: 1) Intensions

Table —— Object

rel?: val?

rel?: val?

A term is defined by its properties (Aristotle…)

But…

- Have *you* ever tried to define a table? Anything else?
- Have you ever seen anyone's definition using this method?

The trouble with: 2) Extensions

- A term in the model is defined as the set of all real-world instances of it:

  Concept x = { *all instances of x in the world* }

- **Problem**: what if you change the instance set?

Representing content in AI today

- Formal, logic-based semantics
  - The meaning of *table* is **table'**
  - The meaning of *table* is a **collection of specific properties**
  - The meaning of *table* is the **set of all tables in the world**

- Frame semantics, implemented
  - The meaning of *table* is **whatever the system ontology contains and refers to** (sort-of intensional)
  - The meaning of *table15* is a **specific instance in the domain and its database** (sort-of extensional)

Problems with Propositional model

1. Symbols themselves are 'empty'
   - No content for symbols in the notation: one cannot *within the propositions* work with their content
   - For example, interactions between negation, modalities, etc., on particular aspects of content remains hidden
2. Symbols are discrete
   - Yet meanings are shaded, spread in a continuum toward different directions of nuance
3. Semantic theories show no direct connections with psycholinguistic or cognitive phenomena
   - No obvious explanations for confusions, forgetting, degrees of processing complexity, etc.

NLP today: Distributional ‘semantics’

- Topic Signature / topic model:

  { T_k, (w_{k1},s_{k1}), (w_{k2},s_{k2}), … , (w_{kn},s_{kn}) }

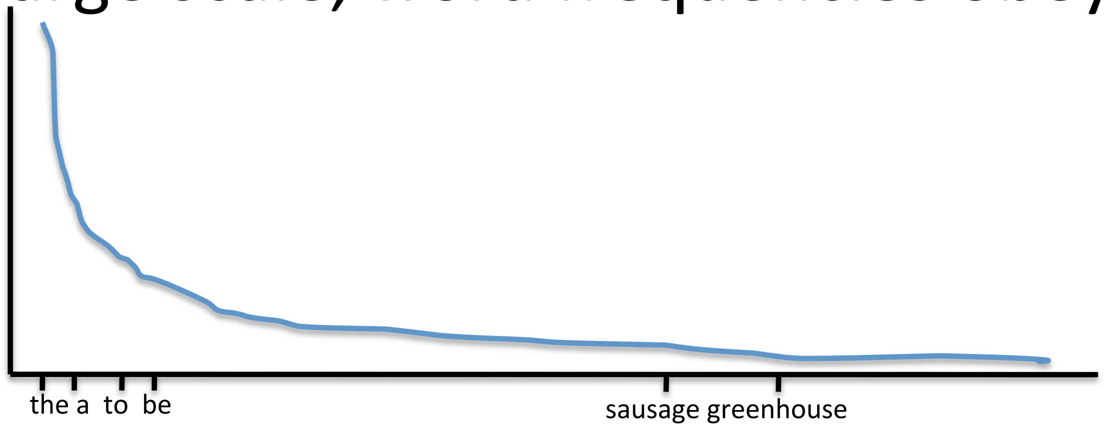
  bank$_1$ = {(*bank* 0.9), (*thrift* 0.11), (*banking* 0.4), (*loan* 0.4), (*deposit* 0.1), (*money* 0.7)…}

  bank$_2$ = {(*bank* 0.9), (*turn* 0.3), (*veer* 0.1), (*lean* 0.4)…}

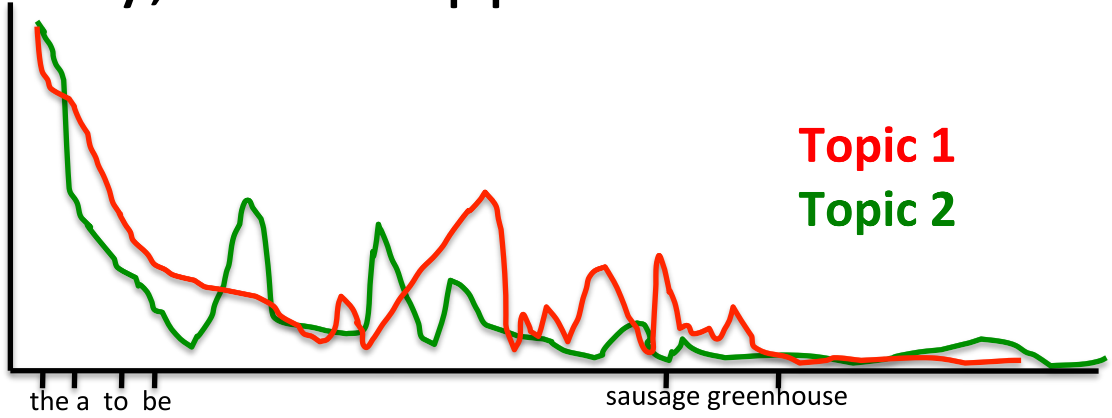
- Operates at word level
- Essentially clustering in lexical space, using tf.idf, PMI…
- Methods: LSA, pLSA, LDA, Chinese Restaurant, etc.
- Used for wordsense disambiguation, sentiment…

Theoretical basis for distributional semantics

- Over large scale, word frequencies obey Zipf's Law:



- But locally, words appear in a Poisson distribution:



Using vectors of words

- "You will know a word by the company it keeps" — Firth

- Collect co-occurring high-freq words in related texts:
  - **Topic Models**: In a **collection of texts** about various topics, topic keywords concentrate around topics; so families of related words appear in 'bursts'. To find the family, compare the word frequency distributions within each topic's texts against global background counts. Use *tf.idf*. χ^2, PMI, etc.

    bank$_1$ *= {bank, thrift, banking, loan, deposit, money…}*
  - **Word Models**: In a **set of sentences** containing the same word, the other words appearing in those sentences more often than expected form the word vector

Topic models, latent and otherwise

- Base assumption: Each document is a bag of words
  - Base model: simplest starting point
  - Zellig Harris (1954) Distributional Structure. *Word* **10** (2/3): 146–62: “And this stock of combinations of elements becomes a factor in the way later choices are made ... for language is not merely a bag of words but a tool with particular properties ….”
- Latent Semantic Analysis (LSA): Matrix operation over texts that groups the words into ‘latent’ (hidden) classes
  - Both + and – association strengths for words in topics
  - Sorted by topic ‘strength’ overall
  - (Deerwester et al., 1990)
- Latent Dirichlet Allocation (LDA): Each doc is a (weighted) set of topics; and each topic is (generates) a (weighted) set of words
  - Introduces a new layer of recombination, plus extra words
  - Automatically trained, but you have to specify how many topics
  - (Blei et al., 2003)

Word models: Contexts for building them

- Specify context from which vector words are selected:
  - Anywhere in the sentence, or left and right sides separately
  - Syntactic field (*Subj, DirectObj, AdjModifier*, etc.)
- Example from (Pantel and Lin 02): syntactic contexts
  - Used to cluster all words having similar contexts
  - http://demo.patrickpantel.com

$$mi_{ef} = \log \frac{\frac{c_{ef}}{N}}{\frac{\sum_{i=1}^{n} c_{if}}{N} \times \frac{\sum_{j=1}^{m} c_{ej}}{N}}$$

apple

-V:obj2:N 89 times:

give 30, offer 20, hand 4, feed 3, grow 3, throw 3, toss 2, name 1 …

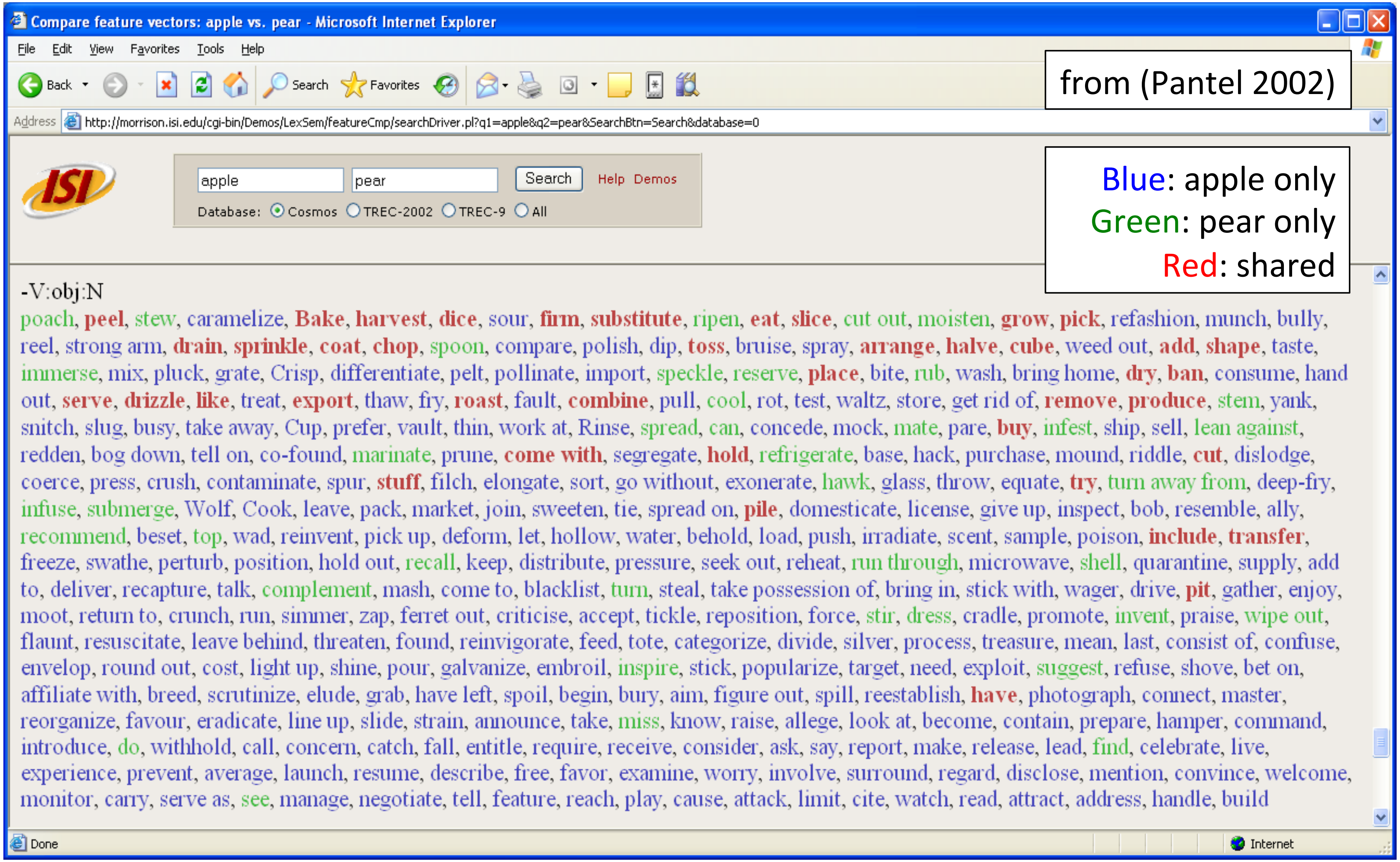
-N:nn:N 3115 times:

Tree 298, Orchard 192, computer 165, logo 64, cider 63, product 61, store 46, employee 45, cultivar 31, snail 29, variety 27, iPhone 26, iTunes 26, Farm 21, Festival 21, core 20 …

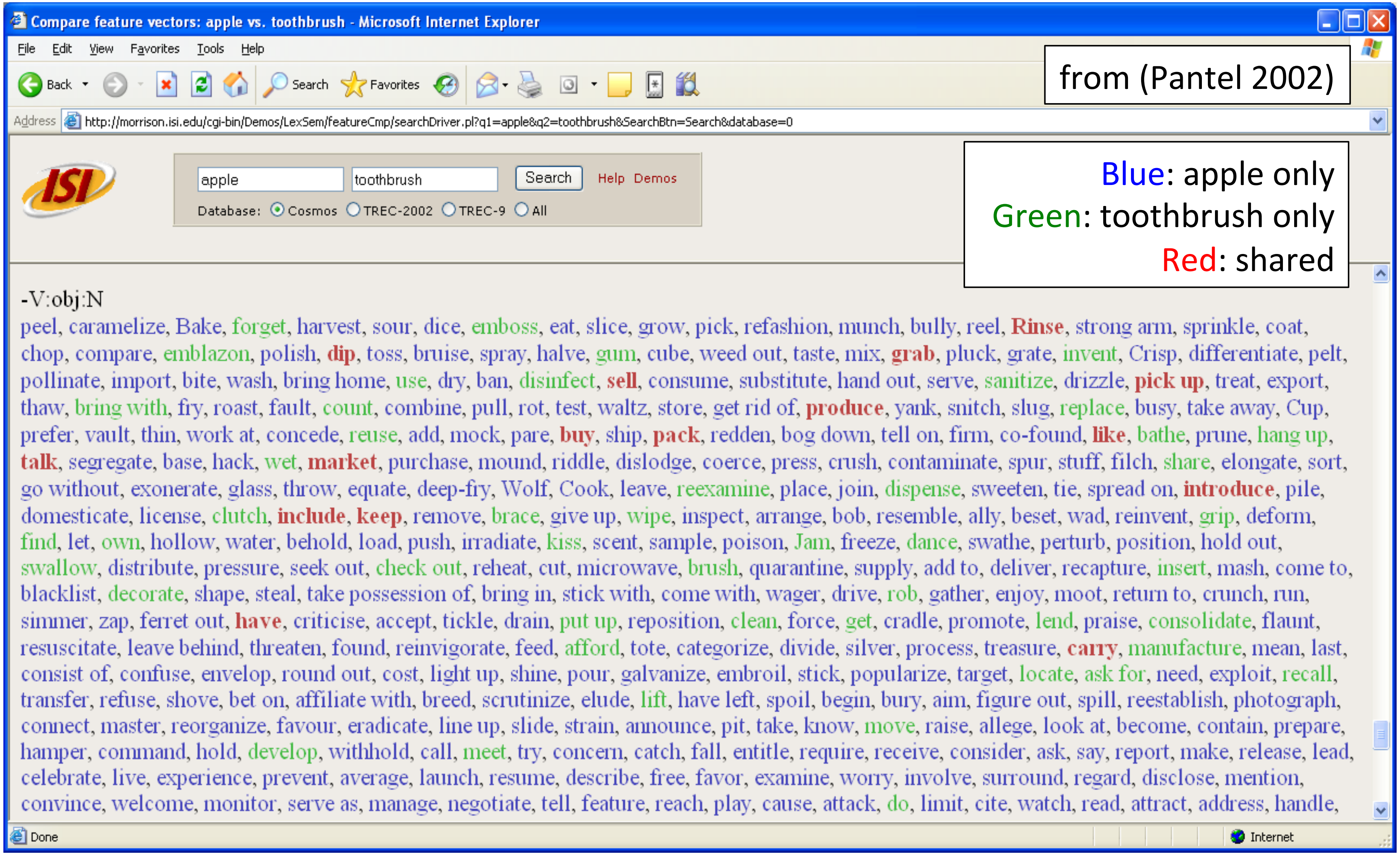
-N:nn:N 878 times:

Candy 27, one 19, fruit 16, Fuji 16, cashew 13, cider 12, silver 11, a few 10, toffee 9, apple 8, Orchard 8, cooking 7, McIntosh 7, poison 7, use 7, Gold 6, Taffy 6, album 5, company 4, crystal 4, debut 4, feature 4, red 4, Wax 4, bad 3 …

Why is *apple* is similar to *pear* (Pantel 02)



Why *apple* is not similar to *toothbrush*

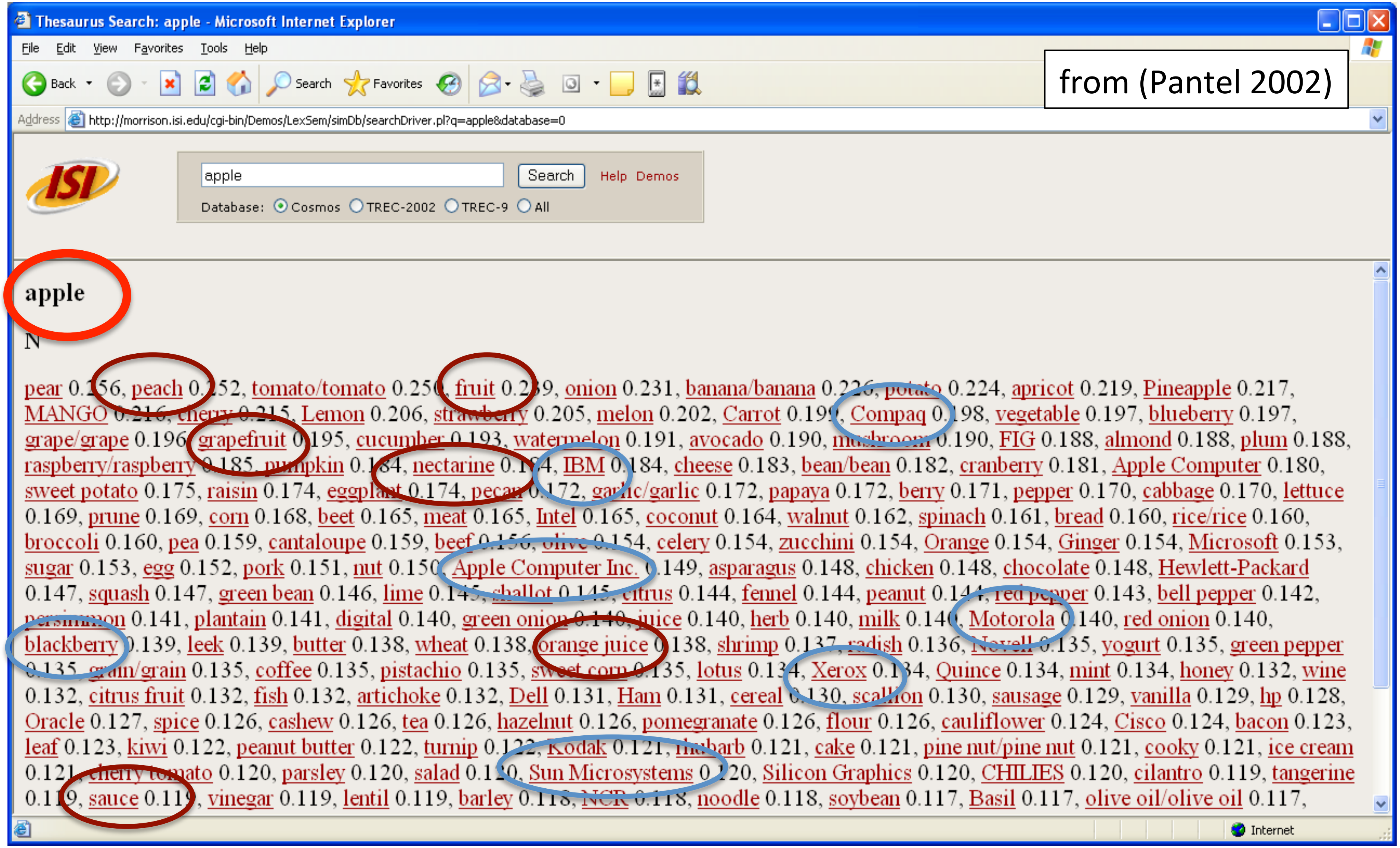


from (Pantel 2002)

Blue: apple only
Green: toothbrush only
Red: shared

In word vectors, senses are mixed up

from (Pantel 2002)



Need senses, not words

- Some words are unambiguous:
  - *Schwarzenegger; banana*
- And some are not:
  - *conclude* (to decide or to end); *party* (a festivity or a political grouping)
- Many ambiguous ones have the following property:
  - A few clearly distinct senses
  - A continuous 'field' of meaning shades, different in different 'directions', and including metaphorical uses
    - *He drove his car into the lake*
    - *His legs drove him forward despite the pain*

      ?: Physical Obj
    - *The news drove stock prices down*

      ?: Propel
    - *This computer drives me crazy*
    - *Drive the devils out of her!*

      ?: Psych state

Semi-overlapping vectors for senses

- Semantically ‘closer’ senses share more of their meaning than ‘further’ ones
- Word vectors allow near-continuous variability for shades of meaning, but can differ in different ‘directions’
  - drive-car: :patient ((car 0.4) (bus 0.2) … (PhysObj 0.05) …)
    :direction (…)) :speed (…)
    :source (…))
  - drive-legs: :patient ((legs 0.5) (fists 0.2) … (PhysObj 0.1) …)
    :direction (…)) :speed (…)
    :force (…))
  - drive-demons :pre-state ((angry 0.2) (disturbed 0.1) …)
    :post-state ((happy 0.5) (calm 0.4) …)

Distributional semantics in NLP

- Increasingly, NLP researchers are simply using the *frequency distributions of associated words* as the (de facto) 'semantics' of a word
  - Treat the word 'families' as features of the target word
  - Sometimes differentiate between left and right contexts
  - Numerous association formulas: raw frequency counts, Pointwise Mutual Information, etc.
- Many applications:
  - Word sense disambiguation, MT, sentiment recognition, entailment, paraphrases…
- Problem: No explicit theory of their semantics

Problems with Distributional models

- Not discrete enough:
  - Topic Models have no clear boundaries
  - There's no good way to evaluate LDA (etc.) output because in principle a topic is an infinitely fine-gradeable thing
- Not compositional: How to 'add' two distributions?
- No operators (negation, modality, etc.)

Propositional + Distributional = ??

- **Propositional Semantics** came to NLP from Mathematical Logic (and Philosophy) via AI
- Semantic expressions written in 'logical form' propositions using meaning symbols; composition of units
- Symbols are undefined (within the formalism): what 'meaning'?
- Operators (modals, negation, etc.) defined

- **Distributional Semantics** comes from modern-day NLP
- Statistical processing over large combinations of texts: "you shall know a word by the company it keeps"
- Combinations of words into units, but little/no composition of units
- No 'real' semantics, but wordlists capture something of contents

- Each has advantages
- Can one merge them?

For semantics: What would we like?

- Combine the properties of traditional propositional semantics and the statistical distributional approach

- From traditional logic-based KR:
  - Formal propositions consisting of symbols
  - Each symbol represents a concept or relation
  - Can compose symbols into complex representations
- From modern statistical NLP:
  - Vectors of word distributions, with weights
  - Each symbol carries its ‘content’ explicitly
  - Symbol contents are not discrete
- With links to other fields:
  - Conform with psycholinguistic and cognitive findings
  - Provide basis for Information Theory measures of info content

MERGING PROPOSITIONAL AND DISTRIBUTIONAL FORMALISMS

Defining a concept the new way

- Def: A concept C is a list of triples

$$C = \{(r_1\ w_1\ s_1)\ (r_2\ w_2\ s_2)\ \ldots\ (r_n\ w_n\ s_n))$$

where r_i ε {Relations} = R, e.g., *:subj, :agent, :color-of*
w_i ε {Words} = vocabulary, e.g., *happy, run, apple*
s_i ε [0,1]

and each w_i has been associated with C through the relation r_i, with a strength of association s_i that is computed under some measure.

In this talk, all the strength scores are simply made up and have no real meaning

Examples — this is actually an old idea

Dog = {(:type Jack Russell 0.2) (:type Retriever 0.4)
(:color brown 0.4) (:color black 0.3)
(:agent-of eat 0.4) (:patient-of chase 0.3) … }

- A **Topic Signature / Topic Model** is a very simple way of defining a topic: there's only one r_i, namely '*associated with*'

  Dog = {(brown 0.9) (bark 0.6) ("Lassie" 0.2)
  (run 0.6) (white 0.4) (chase 0.1) … }

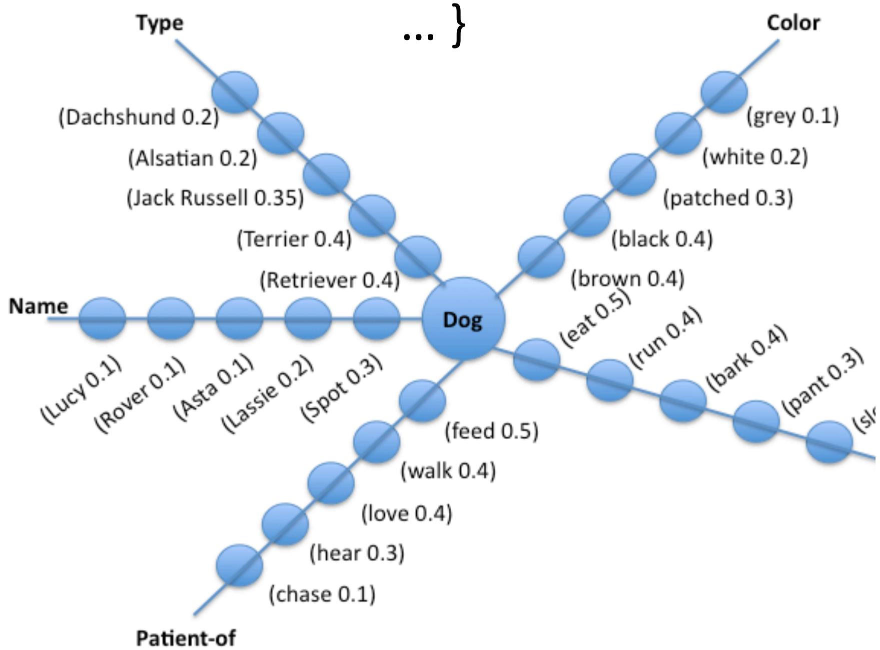
- A **Language Model** in ASR and NLP and MT is the same thing, but allows ngrams instead of words

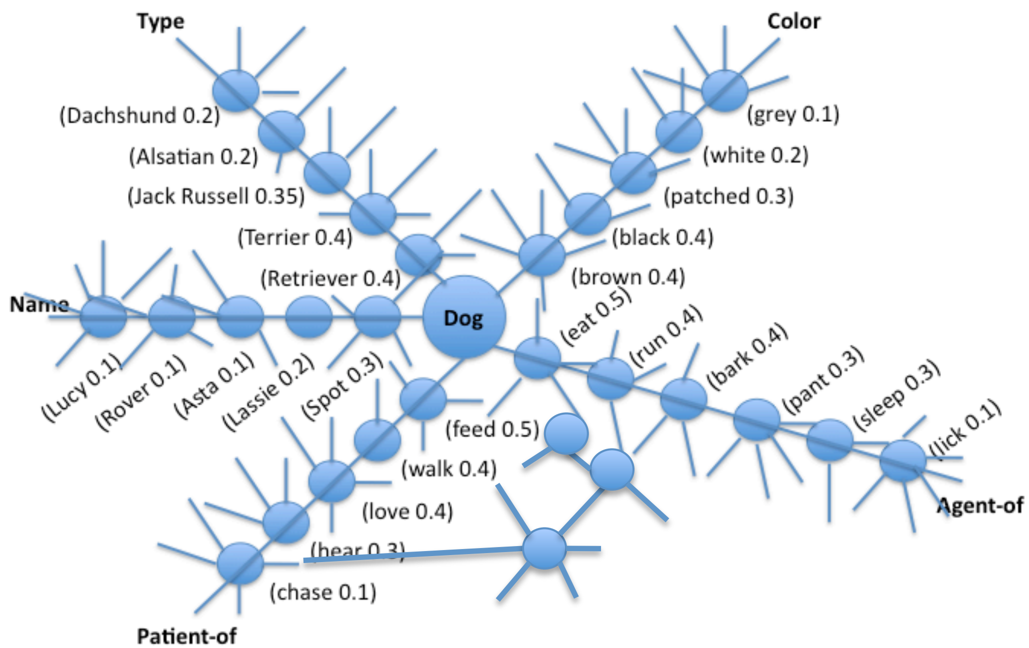
  domain = {("brown dog" 0.0000016)
  ("the brown" 0.0000032) … }

Tensors: A useful notation variant

- It's convenient to group together all tuples with the same r_i:

  Dog = {(:type ((Retriever 0.4) (Terrier 0.4) (Jack Russell 0.35) …))
  (:color ((brown 0.9) (black 0.4) (patched 0.3) (white 0.2) …))
  (:name ((“Spot” 0.3) (“Lassie” 0.2) …))
  (:agent-of ((eat 0.5) (run 0.4) (bark 0.4) (pant 0.3) …))
  (:patient-of ((feed 0.5) (walk 0.4) (love 0.4) …))
  … }

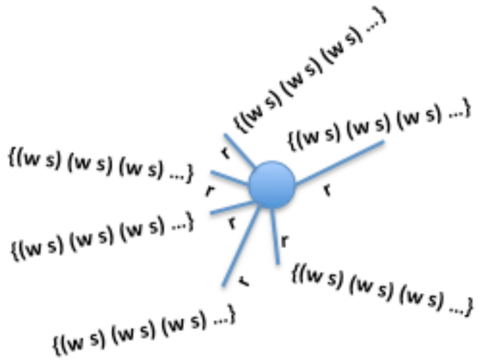




Slightly more formally

- The semantic knowledge base ('lexicon') consists of:
  - $\mathcal{R}$: the list of all relations
  - C : the list of all concepts C_i
  - S : a real number in [0,1]
  - $\mathcal{D}$: the domain (a collection of texts)
  - $\mathcal{M}$: the matrix $\mathcal{R}$ X C containing everything, initialized to zero
  - $\mathcal{KB}$: the knowledge base: a set of all tensors $\mathcal{T}_{Ci}$ for all C_i
- Each generic concept (word) C_i is a tensor as follows:
  - $\mathcal{ID}$: the identifier ('name') of C_i (a string)
  - $\mathcal{T}_{Ci}$: the portion of $\mathcal{M}$ that contains nonzero values of S, computed as appropriate from $\mathcal{D}$ (a tensor)
  - In practice, we store also the source info for the values of $\mathcal{T}_{Ci}$
- Synonymy: C_i approximates C_i insofar as *syn*(C_i,C_j) –> 1
  - *syn(A,B)* must be defined as a continuous-valued function, transitive, but not necessarily obeying the triangle inequality

Scale invariance of the notation



Object:

Apple = {(:isa ((fruit 0.9) (:symbol 0.4)))
(:color ((green 0.5) (red 0.6))) …}

Instance:

Beethoven's 9th Symphony = {(:composed-by (Beethoven 1.0))
(:has-part (("Ode to Joy" 1.0) (movements 1.0) …)) …}

Event:

"John saw the World Cup" = {e0 (:type see) (:agent John)
(:theme World Cup) (:instr ((eyes 1.0) (binoculars 0.2) …)) …}

Topic:

NLP = {(:subareas ((WSD 0.9) (MT 0.9) (Info Extraction 0.9) …))
(:conferences ((ACL 1.0) (COLING 1.0) (HLT 1.0) …)) …}

DS concept 'lexicon'

play$_1$ = {e (:agent ((player 0.1) (contestant 0.1) (Tiger_Woods 0.002) …))
(:theme ((soccer 0.4) (tennis 0.2) (bridge 0.1) (pinochle 0.01)…)) (:loc
((soccer-field 0.4) (tennis-court 0.2) …)) (:time (…)) …}

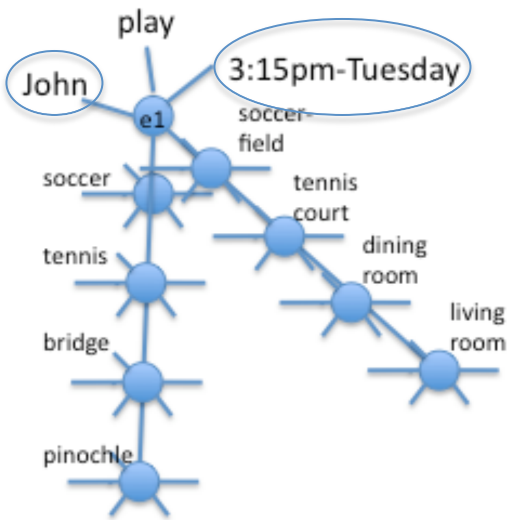
play$_2$ = {e (:agent ()) (:instr ((piano 0.1) (cello 0.1)) (:loc (…)) …}

soccer = {(:isa game) (:loc soccer-field) (:instr soccer-ball) …}

…

"John played at 3:15 pm on Tuesday" =

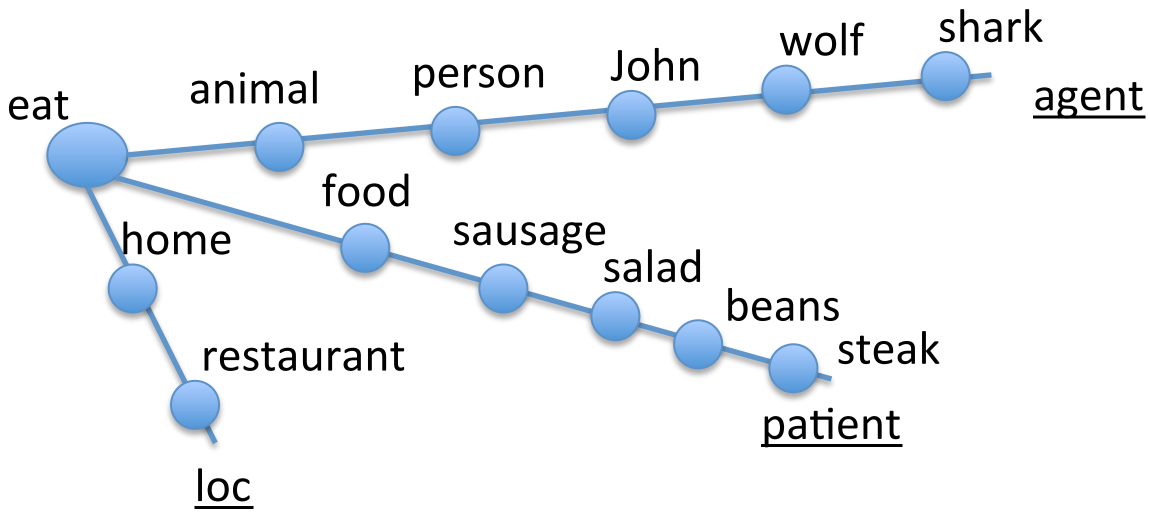
{e0 (:type play$_1$)
(:agent John)
(:theme ((soccer 0.4) (tennis 0.2)
(bridge 0.1) (pinochle 0.01) …))
(:time 3:15pm-Tuesday)
(:location ((soccer-field 0.4)
(tennis-court 0.2)
(dining-room 0.05) …))
…}



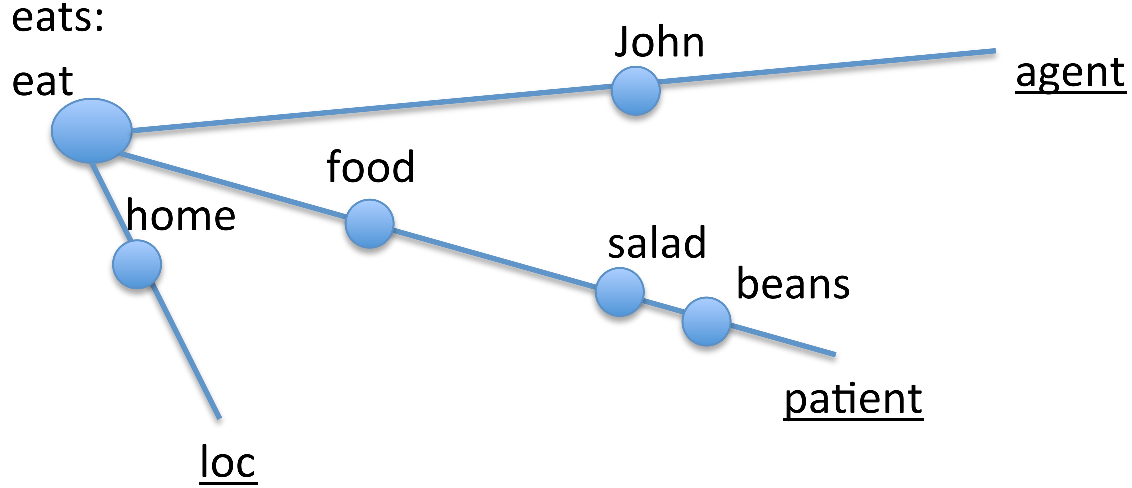
Record just what is given,
use from DS lexicon what is not.
Note underlying dependencies!

Dependency: Compositionality problem

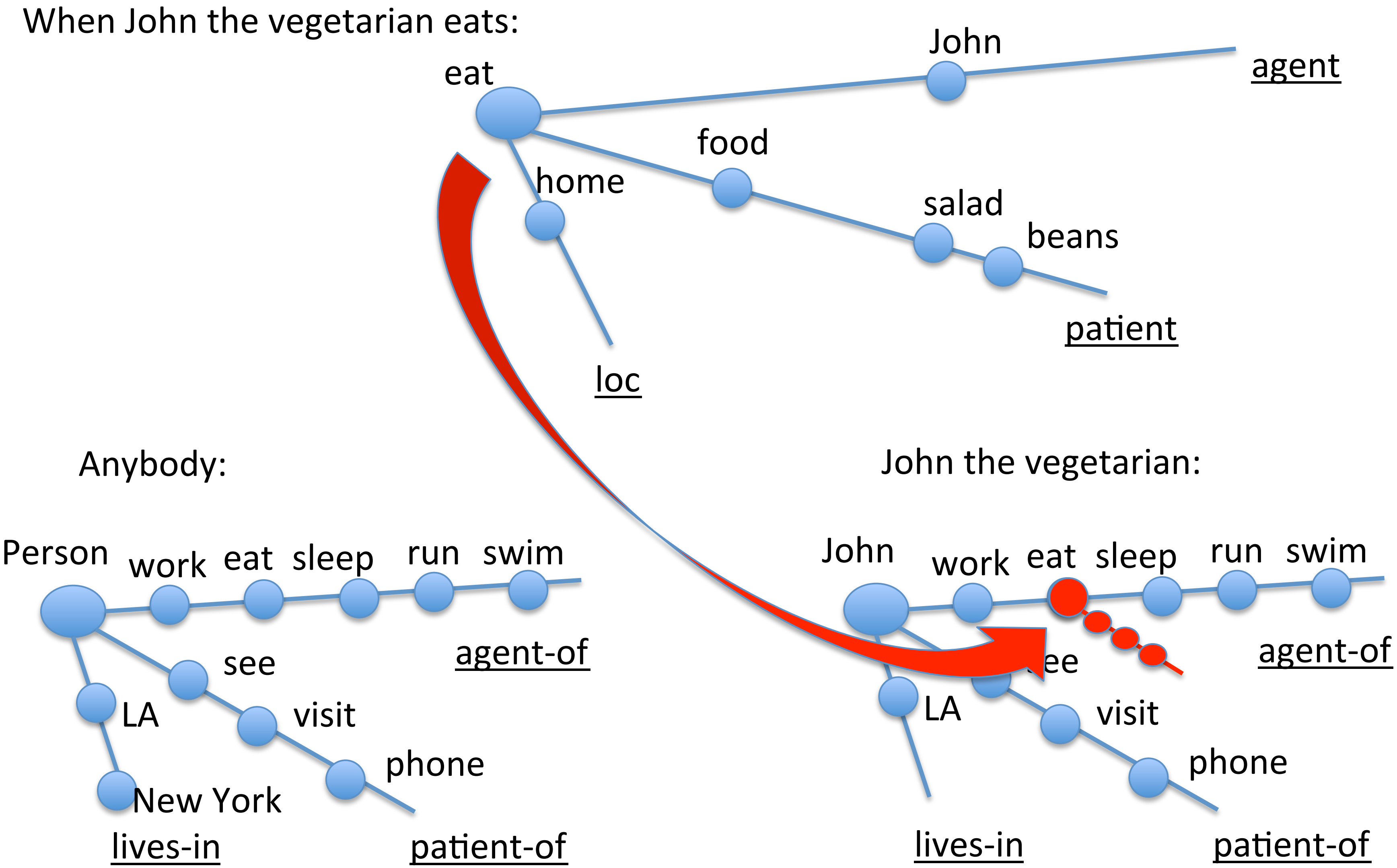
Eat in general:



When John the vegetarian eats:



More Compositionality



Work to date on composition

- Problem: You need to compute the expectation model for each slot in each surrounding context:

  “*Eat*” ≠ “*John eats*” ≠ “*John eats in Paris*” ≠ “*John eats in Beijing*”

- Early work:
  - Wilks 75
  - Baroni and Lenci 10
  - Turney, Turney&Pantel 10
  - Clarke 07
- Compositionality:
  - Mohammad and Hirst 06
  - Lapata&Mitchell, Erk&Pado, etc.
  - (Novacek and others)
  - Grefenstette et al. 10
- Interesting model
  - Socher 11

Computing raw tensor scores

- How to compute it? Definitions:
  - Most people use **co-occurrence probability**
  - Pantel and Lin (2002) use **PMI**
  - Novacek (PhD thesis, 2010) uses **certainty**
    - Real number in [-1,+1]
    - Negative range expresses certainty that NOT(x)
- Problems arise in comparison (synonymy) and compositionality:
  - Tensor for “John is not sad” must look very much like tensor for “John is happy”
  - Tensor for “John doesn’t *like* skiing, he *loves* it!” must not have negative value in *like* cell(s)
- So far, no-one has provided a proper account

WHAT CAN YOU DO WITH THIS?
RECENT WORK AT ISI

Context: Machine Reading

- DARPA-funded program (2009–2014):
  - ERUDITE (BBN, CMU, U Washington, U Oregon, USC/ISI, CYC)
  - FAUST (SRI, Stanford, U Washington, UIUC, etc.)
  - RACR (IBM, USC/ISI, U Texas, U Utah, CMU)
- Challenge: Build system that can extend its own knowledge by reading domain text
  - Target: Single text, not large-scale text harvesting or IE
  - Involves NLP (semantic analysis, QA) and KR (inference, knowledge accretion)
  - Evaluation: Questions on the text just read
- Domains:
  - US football; terrorism actions; medical informatics; …

Our work in RACR

- We address the ‘knowledge gap’ problem: Language is full of omissions and leaps and type coercions
  - Assumption that reader knows the world and can use inference
  - Machines need the same knowledge in order to even start the machine reading bootstrapping process
- We are building a general knowledge support service
- Uses: Bridge various kinds of knowledge gaps:
  - Unknown words/phrases — specialist domain language problem
  - Unclear reference — coref problem
  - Missing fillers — assumed-knowledge problem
  - Missing inter-proposition relations — term connection problem

Tensor Tables: A Proposition Store

- Construct propositions consisting of multiple triples in useful combinations (sentence patterns)
  - NV (noun-verb), AN (adj-noun), NVNPN (NVN-prep-N), etc.
- Obtain counts for each proposition combination:

```
bash-3.2$ grep 'person#n#1:eat:food#n#2:with'
eat.with.trp.dobj
person#n#1:eat:food#n#2:with  family      6
person#n#1:eat:food#n#2:with  chopstick   2
person#n#1:eat:food#n#2:with  spoon       2
person#n#1:eat:food#n#2:with  and         1
person#n#1:eat:food#n#2:with  glass       1
person#n#1:eat:food#n#2:with  variety     1
person#n#1:eat:food#n#2:with  husband     1
person#n#1:eat:food#n#2:with  hand    1
person#n#1:eat:food#n#2:with  president   1
person#n#1:eat:food#n#2:with  child       1
person#n#1:eat:food#n#2:with  Ginsburg    1
person#n#1:eat:food#n#2:with  dressing    1
person#n#1:eat:food#n#2:with  fork    1
person#n#1:eat:food#n#2:with  globalizat  1
person#n#1:eat:food#n#2:with  parent      1
```

```
person#n#1:eat:food#n#2:with  cornichon 1
person#n#1:eat:food#n#2:with  Stanley   1
person#n#1:eat:food#n#2:with  meat 1
person#n#1:eat:food#n#2:with  opponent  1
person#n#1:eat:food#n#2:with  gusto     1
person#n#1:eat:food#n#2:with  Cleopatra 1
person#n#1:eat:food#n#2:with  blood     1
person#n#1:eat:food#n#2:with  fruit     1
person#n#1:eat:food#n#2:with  mother    1
person#n#1:eat:food#n#2:with  mustard   1
person#n#1:eat:food#n#2:with  money     1
person#n#1:eat:food#n#2:with  Newhouse  1
person#n#1:eat:food#n#2:with  group     1
person#n#1:eat:food#n#2:with  kid       1
person#n#1:eat:food#n#2:with  mid-after 1
person#n#1:eat:food#n#2:with  student   1
person#n#1:eat:food#n#2:with  friend    1
```

PropStore construction

1. Take a lot of domain text
2. Parse every sentence (dependency parse)
3. (Convert the syntactic and prep relations to semantic ones)
4. Cut up the dependency tree into [*Head-Rel-Mod*] triples
5. (If needed, combine triples into longer propositions)
6. Save every triple/prop in a large Store:

   [*rel head mod 1 doc-id sent-id*]
7. When done, add together all identical triples/props:

   [*rel head mod total* ((doc-id$_1$ sent-id$_1$) (doc-id$_2$ sent-id$_2$) …)]
8. Regroup as needed (e.g., sort under the heads):

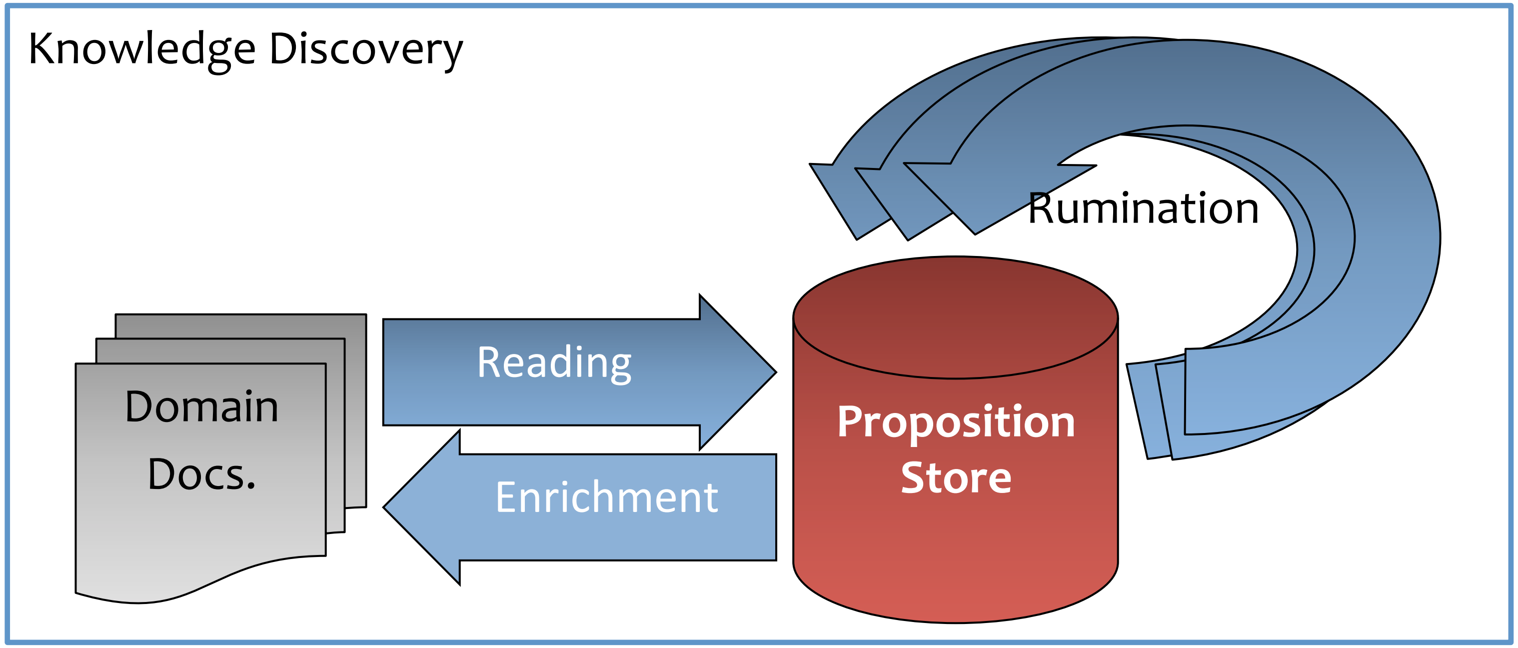
   [*head (rel (mod$_1$ total$_1$) (mod$_2$ total$_2$) …)* ((doc-id$_1$ sent-id$_1$)…)]

   (rel (mod$_1$ total$_1$) (mod$_2$ total$_2$) …) ((doc-id$_1$ sent-id$_1$)…)]

Current Proposition Stores at ISI

- Various Machine Reading project domains:
  - NFL: 30,000 docs (1,000,000 sentences)
  - IC: 200,000 docs (~6,500,000 sentences)
  - BIO: 75,000,000 sentences (all PubMed abstracts)
  - General: 220 million triples (6.3GB compressed to 517.7MB)
    - Triple types: 50,840,754
    - Triple count sum: 461,941,244
    - About 30 relations (all syntactic): DOBJ, etc.
    - Source corpus: 50,000,000+ sentences from New York Times
- Various formats:
  - Raw parse tree triples
  - Nested role fillers (modifiers) for each head
- Machinery to rapidly build new ones
- Large central Store and access machinery being built at CMU
- IBM's PRISMATIC (from 30 gb text: over 1b propositions)

The MR knowledge enrichment cycle



Cycle:

1. Read text from collection
2. Ruminate in PropStore: generalize, etc.
3. Enrich text representation and store
4. Repeat

Knowledge enrichment pattern definition notation

Patterns over dependency trees in Proposition Store

- Pattern definition language (implemented in Prolog):

  prop(Type, Form : DependencyConstrains : NodeConstrains).

- Examples:

  prop('NV', [N,V] : **[V:N:nsubj, not(V:_:'dobj')]** : [verb(V)]).

  prop('NVNPN', [N1,V,N2,P,N3]:**[V:N2:'dobj', V:N3:Prep, subj(V,N1)]**:[prep(Prep,P)]).

  prop('N-has-value-C', [N,Val]:[N:Val:_]:**[nn(N), cd(Val), not(lemma(Val,'one'))]**).

Ex 1: Filling knowledge gaps

Peñas et al., HLT workshop 2010

Example: *San Francisco's Eric Davis intercepted a Steve Walsh pass on the next series to set up a seven-yard Young touchdown pass to Brent Jones.*

Implicit	(More) explicit
San Francisco’s Eric Davis	Eric Davis plays for San Francisco
Eric Davis intercepted pass	—
Steve Walsh pass	Steve Walsh threw pass Steve Walsh threw interception
Young touchdown pass	Young completed pass for touchdown
touchdown pass to Brent Jones	Brent Jones caught pass for touchdown

These are inferences on the language side

Queries to US Football Proposition Store

?> NPN 'pass':X:'touchdown‘

NPN 712 'pass':'for':'touchdown'

NPN 24 'pass':'include':'touchdown’

…

?> NVN 'quarterback':X:'pass'

NVN 98 'quarterback':'throw':'pass'

NVN 27 'quarterback':'complete':'pass‘

…

?> NVNPN 'NNP':X:'pass':Y:'touchdown'

NVNPN 189 'NNP':'catch':'pass':'for':'touchdown'

NVNPN 26 'NNP':'complete':'pass':'for':'touchdown‘

…

?> NVN 'end':X:'pass‘

NVN 28 'end':'catch':'pass'

NVN 6 'end':'drop':'pass‘

…

?> NN NNP:’pass’

NN 24 'Marino’:'pass‘

NN 17 'Kelly':'pass'

NN 15 'Elway’:'pass’

…

?>X:has-instance:’Marino’

20 'quarterback':has-instance:'Marino'

6 'passer':has-instance:'Marino'

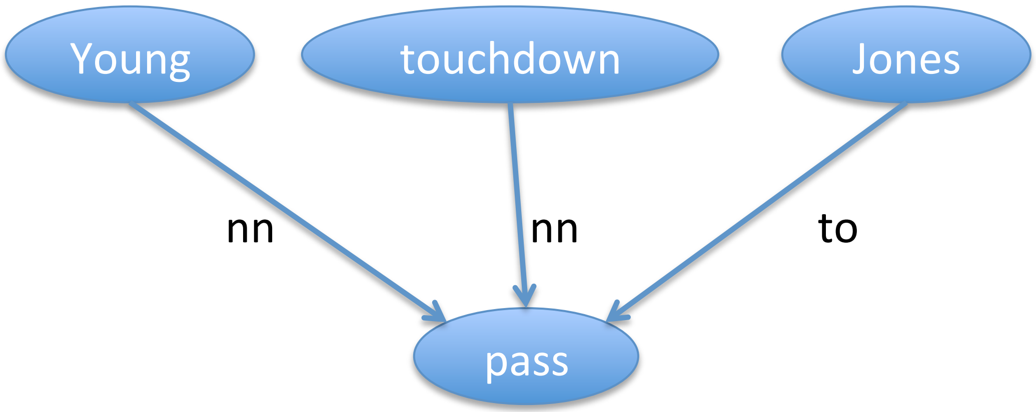
4 'leader':has-instance:'Marino'

3 'veteran':has-instance:'Marino'

2 'player':has-instance:'Marino'

Enrichment example: 1

…to set up a 7-yard Young touchdown pass to Brent Jones



Young pass

?> X:has-instance:Young

X=quarterback

?> NVN:quarterback:X:pass

X=throw

X=complete

Pass to Jones

?> X:has-instance:Jones

X=end

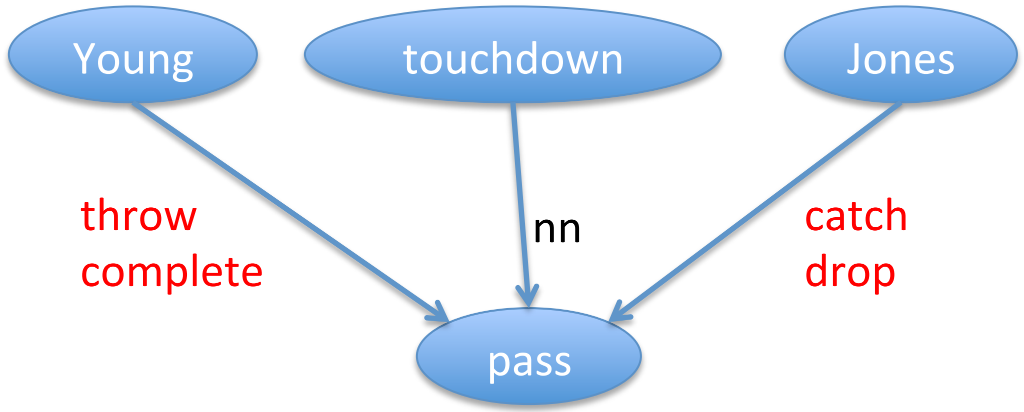
?> NVN:end:X:pass

X=catch

X=drop

Enrichment 2

...to set up a 7-yard Young touchdown pass to Brent Jones



touchdown pass

?> NVN touchdown:X:pass

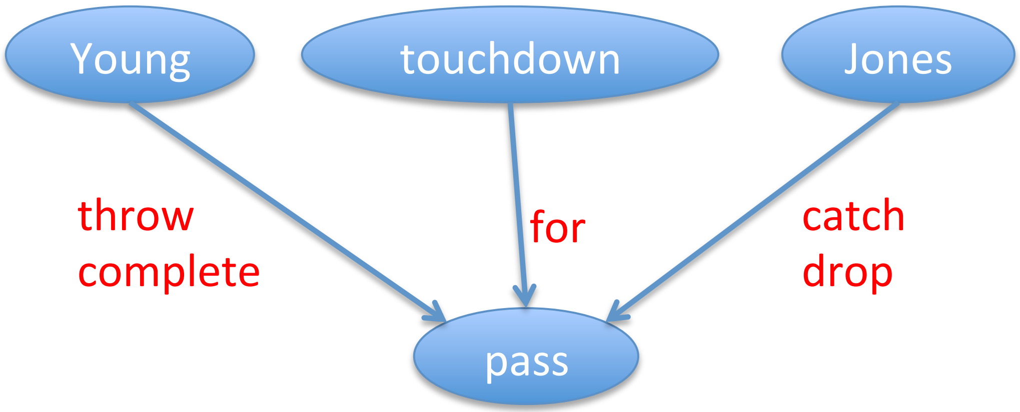
False

?> NPN pass:X:touchdown

X=for

Enrichment 3

…to set up a 7-yard Young touchdown pass to Brent Jones



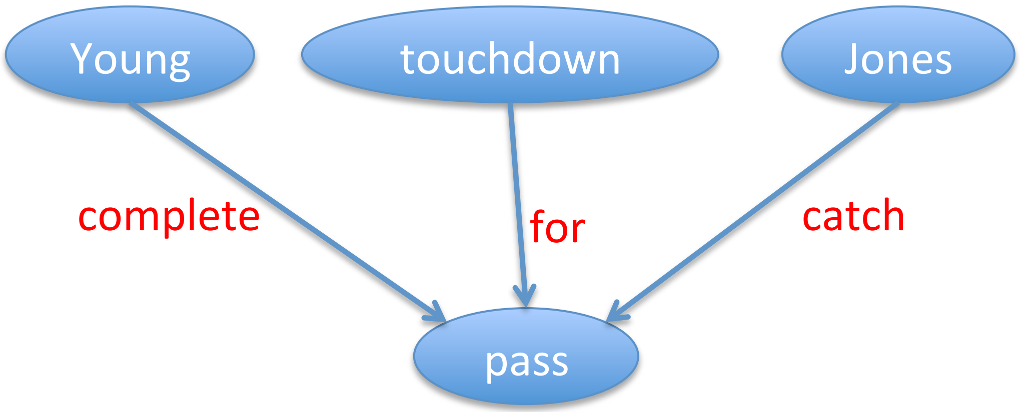
?> NVNPN NAME:X:pass:for:touchdown

X=complete

X=catch

Enrichment 4

…to set up a 7-yard Young touchdown pass to Brent Jones



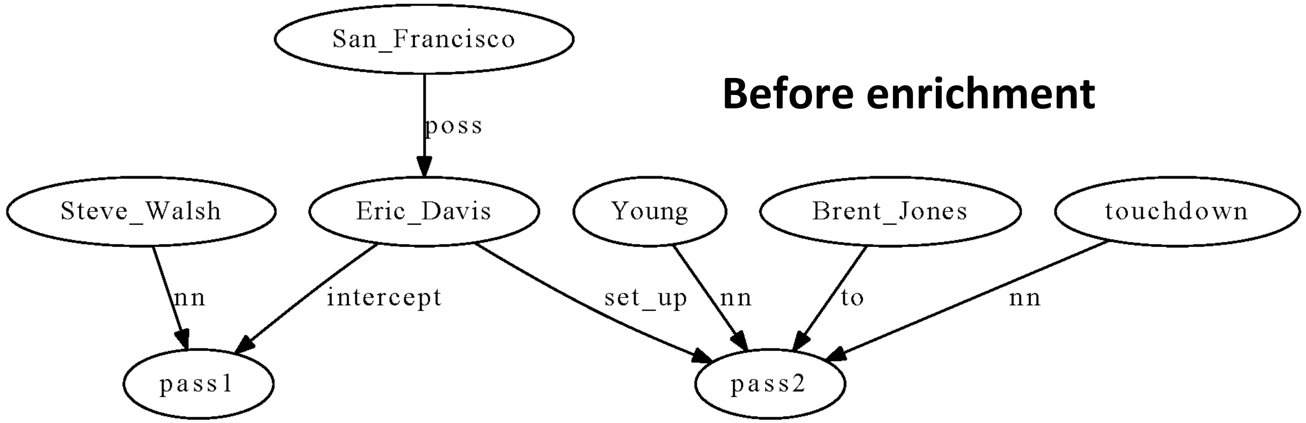
⇒ Young complete pass for touchdown

⇒ Jones catch pass for touchdown

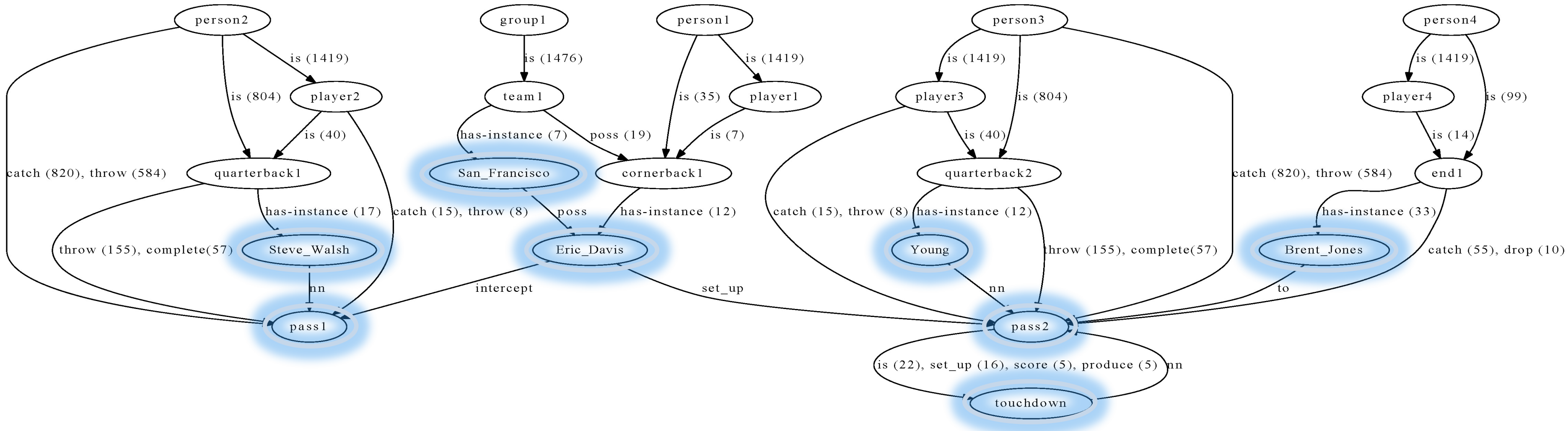
Example result

San Francisco's Eric Davis intercepted a Steve Walsh pass on the next series to set up a seven-yard Young touchdown pass to Brent Jones.

Before enrichment



After enrichment



Uses of Proposition Store 1

Building domain instance knowledge

- 334:has_instance:[quarterback:n, ('Kerry':'Collins'):name].
- 306:has_instance:[end:n, ('Michael':'Strahan'):name].
- 192:has_instance:[team:n, 'Giants':name].
- 178:has_instance:[owner:n, ('Jerry':'Jones'):name].
- 151:has_instance:[linebacker:n, ('Jessie':'Armstead'):name].
- 145:has_instance:[coach:n, ('Bill':'Parcells'):name].
- 139:has_instance:[receiver:n, ('Amani':'Toomer'):name].

- 20 'quarterback':has-instance:'Marino'
- 6 'passer':has-instance:'Marino'
- 4 'leader':has-instance:'Marino'
- 3 'veteran':has-instance:'Marino'
- 2 'player':has-instance:'Marino'

Discovering what people do

- nvn(('NNP':'player'):'catch':'pass'):83.
- nvn(('NNP':'player'):'miss':'game'):66.
- nvn(('NNP':'player'):'have':'yard'):59.
- nvn(('NNP':'player'):'gain':'yard'):49.
- nvn(('NNP':'player'):'throw':'pass'):43.
- nvn(('NNP':'team'):'beat':('NNP':'team')):1151.
- nvn(('NNP':'quarterback'):'throw':'pass'):1093.
- nvn(('NNP':'team'):'win':'game'):1032.
- nvn(('NNP':'team'):'play':('NNP':'team')):798.
- nvn(('NNP':'receiver'):'catch':'pass'):628.

- NVN 26 'Marino':'throw':'pass'
- NVN 15 'Marino':'complete':'pass'
- NVN 9 'Marino':'miss':'game'
- NVN 8 'Marino':'throw':'interception'
- NVN 5 'Marino':'toss':'pass'
- NVN 5 'Marino':'throw':'touchdown'

Uses of Proposition Store 2

Discovering ‘causes’ within ‘to’ sentences

- 109 present:v, evidence:n -> answer:v, question:n
- 107 present:v, evidence:n -> answer:v, (clinical:question):n
- 64 reduce:v, (detrimental:custom):n -> affect:v, (perinatal:community:morbidity):n
- 64 modulate:v, (electron:therapy):n -> achieve:v, (conformal:dose:distribution):n
- 64 use:v, (electrophoresis:device):n -> fractionate:v, (complex:protein:mixture):n
- 64 have:v, (incisional:infection:rate):n -> undergo:v, (abdominal:exploration):n

Enrichment

- e.g., quarterback & receiver
  - nvn:('NNP':'quarterback'):'hit':('NNP':'receiver'),177).
  - nvnpn:('NNP':'quarterback'):'throw':'pass':'to':('NNP':'receiver'),143).
  - nvnpn:('NNP':'quarterback'):'complete':'pass':'to':('NNP':'receiver'),79).
  - nvn:('NNP':'quarterback'):'find':('NNP':'receiver'),69).
  - nvnpn:('NNP':'receiver'):'catch':'pass':'from':('NNP':'quarterback'),43).

Uses of Proposition Store 3

- Overcoming problems in parsing
  - Improve POS tagging (especially for long noun phrases):
    - NVN 46 'Giants':'coach':'Jim_Fassel'
    - nvn(('NNP':'team'):'coach':('NNP':'coach')):538.
  - Learn domain terminology: (running:back)
  - Make correct PP attachments
  - Handle conjunctions (especially of clauses)
  - Discover hidden prepositions:
    - John ran 3 yards -> NVN:John:run:yard
    - Should be NVPN:John:run:PREP:yard
      - 163:nvpn:[person:n, run:v, for:in, yard:n].
      - 48:nvpn:[player:class, run:v, for:in, yard:n].

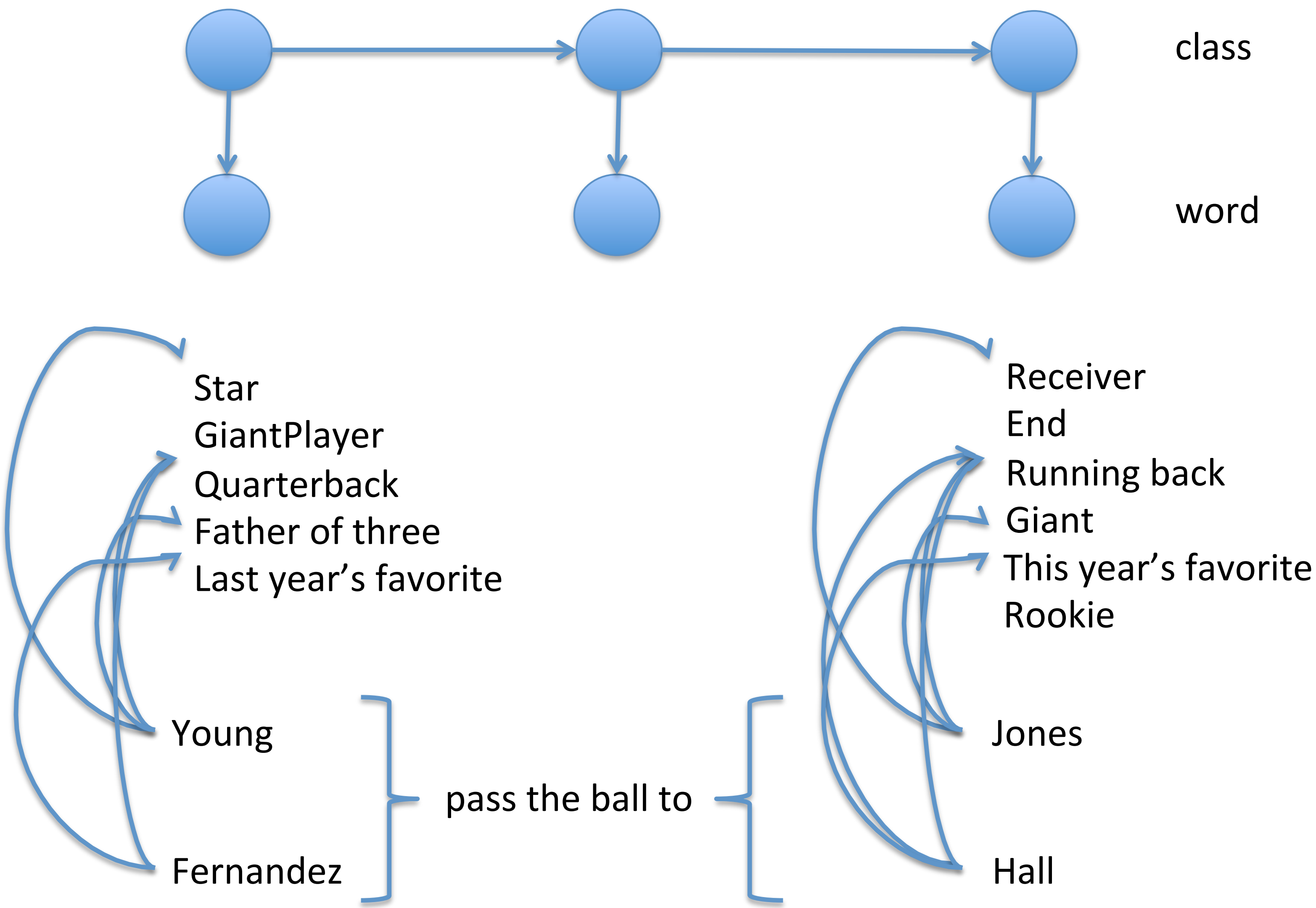
Ex 2: Generalization

Hovy et al., ACL poster 2011

- Goal: Infer implicit domain generalizations from text
- Automatically extract the classes to allow for more fine-grained labeling than NE tags

Marino throws to Fernandez	Quarterback throws to receiver	Human throws to human

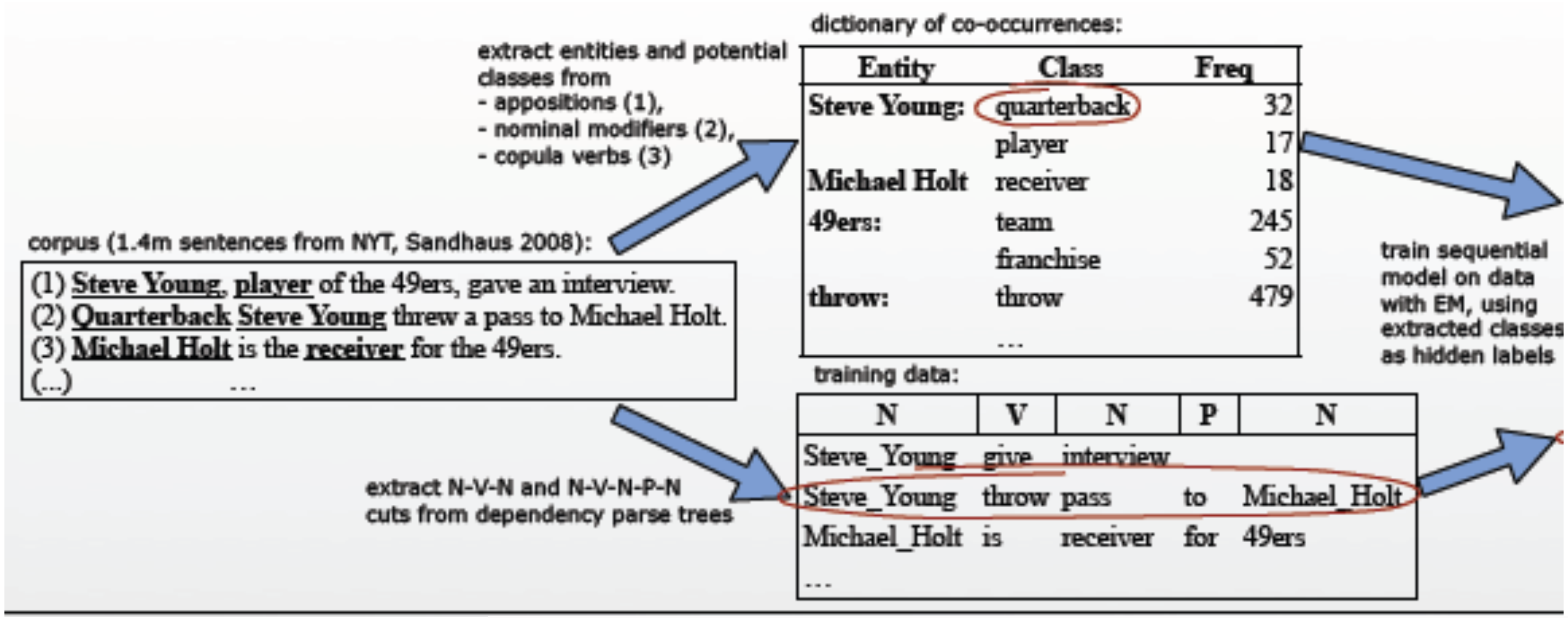
  - Domain-specific classes can be inferred from simple lexico-syntactic patterns
  - Formulate as unsupervised labeling problem

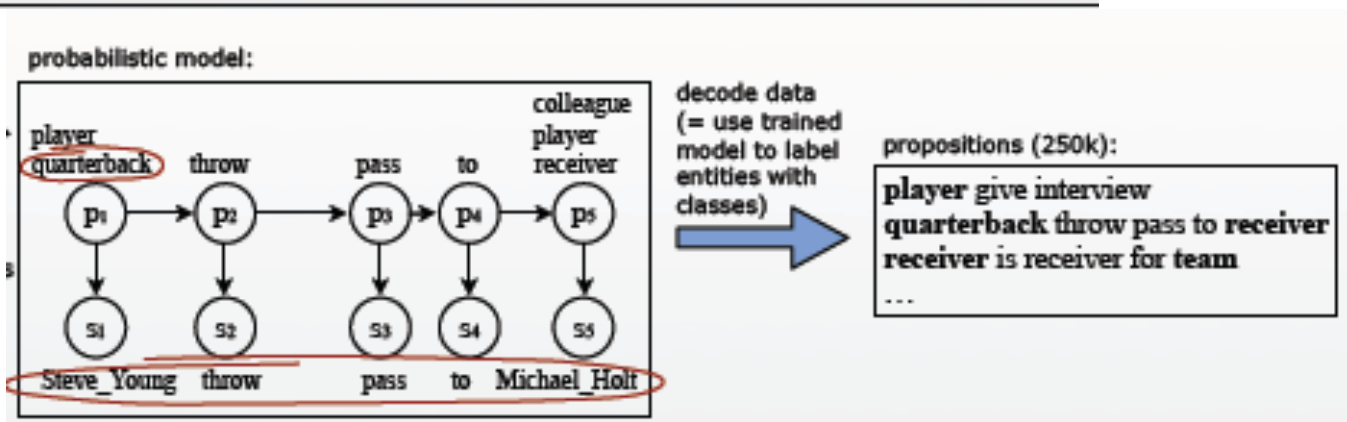


$$\prod_i P\,(class_i \mid class_{i-1})\ .\ P\,(word_i \mid class_i)$$

Model

Define as sequential labeling task
Apply Viterbi decoding
Learn ~250k generalized propositions





$$P(\mathbf{s}, \mathbf{p}) = \prod_{i=1}^{n} \left(P(p_i | p_{i-1}) \cdot P(s_i | p_i) \right)$$

Evaluation 1: 3 measures

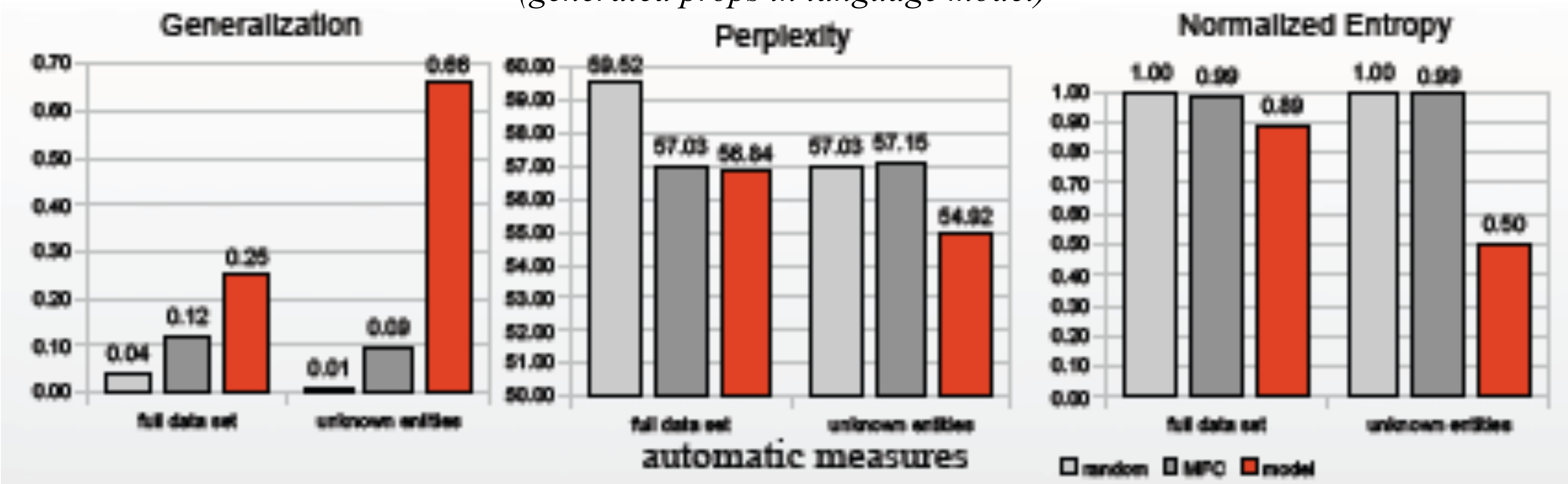
- Unclear how to assess model's explanatory power: *what relation between prop generalization and its sentences?*
  - Better to generalize high? — then only top class
  - Or low? — then word itself
  - Where in the middle? — best entropy reduction?

$$g = 1 - \frac{|propositions|}{|sentences|}$$

$$perplexity(data) = 2^{\frac{-\log P(data)}{n}}$$

(generated props in language model)

$$H_N = \frac{-\sum_{i=0}^{n} P_i \cdot \log P_i}{\log n}$$



Evaluation 2: Annotation

- Question: Is the proposition sensible?

"Quarterbacks can throw passes to receivers" vs. "Coaches can intercept teams"

  - Baselines: 100 sampled from most-frequent class (293,028 props) + 100 random from data
  - Model: 200 sampled from 250,169 props

- Percentages labeled 'sensible':

		100 most frequent		random		combined	
Data set	System	agg	maj	agg	maj	agg	maj
full	baseline	90.16	92.13	69.35	**70.57**	88.84	90.37
	model	**94.28**	**96.55**	**70.93**	70.45	**93.06**	**95.16**
unknown	baseline	51.92	51.51	32.39	28.21	50.39	49.66
	model	**66.00**	**69.57**	**48.14**	**41.74**	**64.83**	**67.76**

Rating the model

- Model has good explanatory power and generalizes well
- Evaluation:
  - Human subjects judge up 95.2% of resulting propositions sensible, 67.8% for the ones with unseen entities
  - Inter-annotator agreement reasonably high (raw agreement = 0.82, G = 0.58, κ = 0.48)

What's the general plan?

- Use PropStore as a very simple Knowledge Base of background world knowledge

  (similar to language model, but with structure)

- Develop methods to
  - Create it
  - Use it to cover knowledge gaps and check semantic interpretations
  - Produce expectations for machine reading

CONCEPT FACETS OR DIMENSIONS

The problem of facets

- Differentiate the tensor into facets using relations
- Problems:
  - Which facets for events?
  - Which facets for objects?
  - What is the representation of a relation?
  - Interaction with compositionality

Syntactic or semantic relations?

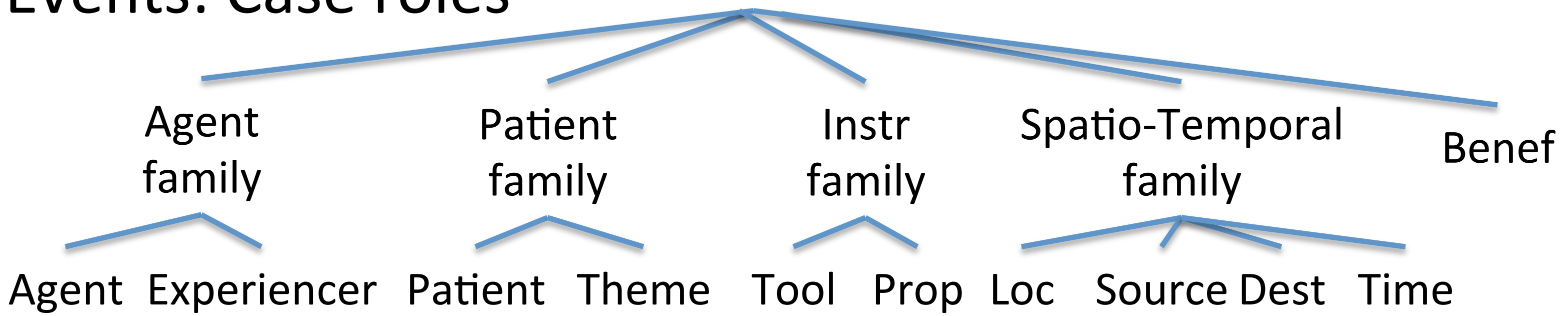
Parse tree gives merely syntactic relations

Nice, if you can get them:

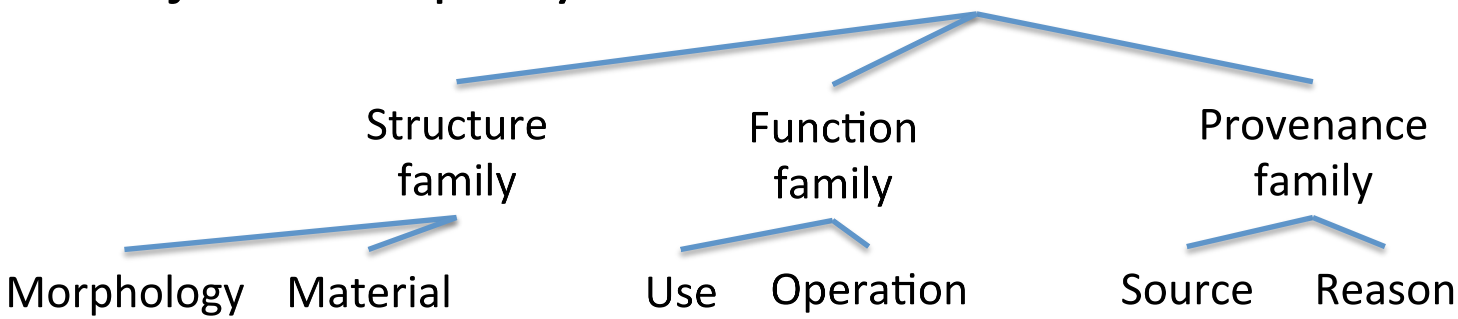
- Verb relations:
  - Case roles: from Framenet or PropBank
  - Prepositions: Prep sense disambiguation
- Noun relations:
  - Noun-noun compounds: NN relation classification
  - Noun-adjective modifiers: relation classification
- Multi-clause relations:
  - Verb-verb relation classification

Reminder: Case Relations

- Minimum: relation *associated-with* (in topic signature)
- Better: syntactic relations (*subj, dobj, iobj, preps*…)
- Even better: semantic relations
  - Events: Case roles



  - Objects: Property relations



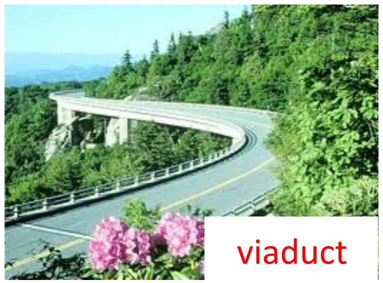
CONCEPT (SENSE) GRANULARITY

Semantic ‘fields’ are continuous

- Many ambiguous words have the following:
  - First, a few clearly distinct senses:

Bridge:

  - For the rest, a continuous ‘field’ of shades of meaning, different in different ‘directions’ … even including metaphorical uses



What do we need from a semantic representation?

Must be granular yet continuous

- **Granular**: Ability to name specific and different concepts
  – bridge_1 = card game; bridge_2 = structure/path over a gap
- **Continuous**: Ability to represent ‘shades’ of meaning, almost continuously variable in different ‘directions’
  – bridge_{2a} — but narrow and in/along buildings
  – bridge_{2b} — but made of rope, for single person

Semi-overlapping vectors for senses

- Word vectors allow near-continuous variability for shades of meaning, *and* can differ in different 'directions'

No overlap: discrete senses

	card	play	trump	cross	high	nose	concrete	...
$bridge_1$	0.7	0.5	0.35					...
$bridge_2$				0.55	0.31		0.12	...

Some overlap: continuum

	cross	high	concrete	steel	rope	suspension	narrow	...
$bridge_{2a}$	0.55	0.31	0.15	0.11		0.09		...
$bridge_{2b}$	0.49	0.27			0.22		0.07	...

COMPOSITIONALITY

Combining vectors/tensors

- Question: How to compose word/concept tensors into new meanings?

  The meaning of word *w* in context *C* is a new tensor *v* that is a function of *w* and *C* : $v = w ✤ C$. The context *C* is just another tensor. But what is ✤ ?

- Centroid of tensor's vectors? What would this look like?
- Bag of words? Kintsch, 2001; Mitchell and Lapata, 2008: simply use the words associated with the composed phrase in context
  - But then cannot formally distinguish between "he sees a peach" and "a peach sees him"; and "John sees a peach" is different even if he = John

Semantic distance between concepts

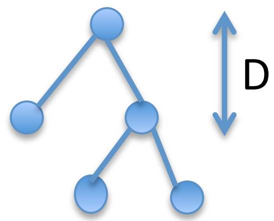
- Overview of semantic distance measures: (Budanitsky and Hirst, *Computational Linguistics* 2006)

1. **In a semantic structure** like WordNet or a thesaurus:

- Various metrics counting number of inter-concept links and depth

  e.g. (Leacock and Chodorow 1998)

  $\text{Dist}(c_1,c_2) = -\log len(c_1,c_2) \,/\, 2D$



2. **In a word distribution**:

- Distributionally close and semantically related:
  - two target words have many common strongly co-occurring words
  - (*doctor–surgeon* and *doctor–scalpel*)
- Distributionally close and semantically similar:
  - two target words have many common strongly co-occurring words that each have the <u>same syntactic relation</u> with the two targets
  - (*doctor–surgeon*, but **not** *doctor–scalpel*)

Common distance measures

- All distributional measures have two parts:
  1. Method to create distributional profiles (DPs)
  2. Method to calculate distance between two DPs

1. head = {(w_1 s_1) (w_2 s_2) …}

Here, need a measure of the strength of association s_i between the head and the profile elements

2. Dist (*head*$_1$ *head*$_2$)

3. Combinations

Measures of DP distance
α-skew divergence (ASD)
cosine (Cos)
Dice coefficient (Dice)
Euclidean distance (L_2 norm)
Hindle's measure (Hin)
Kullback-Leibler divergence (KLD)
Manhattan distance (L_1 norm)
Jensen–Shannon divergence (JSD)
Lin's measure (Lin)

Measures of strength of association
ϕ coefficient (Phi)
conditional probability (CP)
cosine (Cos)
Dice coefficient (Dice)
odds ratio (Odds)
pointwise mutual information (PMI)
Yule's coefficient (Yule)

Standard combinations
α-skew divergence with ϕ coefficient (ASD–CP)
cosine with conditional probability (Cos–CP)
Dice coefficient with conditional probability (Dice–CP)
Euclidean distance with conditional probability (L_2 norm–CP)
Hindle's measure with pointwise mutual information (Hin–PMI)
Kullback-Leibler divergence with conditional probability (KLD–CP)
Manhattan distance with conditional probability (L_1 norm–CP)
Jensen–Shannon divergence with conditional probability (JSD–CP)
Lin's measure with pointwise mutual information (Lin–PMI)

(Mohammad and Hearst, 2012)

Combining concept vectors: ex.

Mohammad and Hirst, 2006: For concept C = {(w_1,s_1) (w_2,s_2) …} and C(c) = {w_1,w_2…}, they adapt Cosine distance as

$$Cos_{cp}(c_1, c_2) = \frac{\sum_{w \in C(c_1) \cup C(c_2)} \left(P(w|c_1) \times P(w|c_2)\right)}{\sqrt{\sum_{w \in C(c_1)} P(w|c_1)^2} \times \sqrt{\sum_{w \in C(c_2)} P(w|c_2)^2}}$$

Evaluation:

- Macquarie Thesaurus, with 812 coarse-grained ‘concepts’ to form word distributions
- Two tests:
  1. Rank word pairs in order of their semantic distance
  2. Correct real-word spelling errors
- On both tasks, distributional concept distance measures much better than distributional word-distance measures

Recent work on semantic composition

Lots of recent work on combining word vectors (though not in the context of propositional frames):

- Mitchell and Lapata 08; 10
- Erk and Padó 08; 10
- Baroni and Lenci 10
- Thater et al. 10
- Grefenstette et al. 10; 11
- Wu 11
- Clarke 07, 11
- Socher and Manning 11
- Mohammad and Hirst 06; 12

RANDOM INTERESTING THINGS

A note on informativeness

- "John saw the 2010 World Cup in South Africa"
  {e0 (:**type** see) (:**agent** John)
  (:**theme** World-Cup-2010)
  (:**instr** ((eyes 0.99) (binoculars 0.2) …))
  (:**loc** South-Africa)
  …}

- "John saw the 2010 World Cup in SA with binoculars"
  {e0 (:**type** see) (:**agent** John)
  (:**theme** World-Cup-2010)
  (:**instr** (binoculars 1.0))
  (:**loc** South-Africa)
  …}

- "John saw the 2010 World Cup in SA with his eyes"
  {e0 (:**type** see) (:**agent** John)
  (:**theme** World-Cup-2010)
  (:**instr** (eyes 1.0))
  (:**loc** South-Africa)
  …}

- "John saw the 2010 World Cup in SA through a telescope"
  {e0 (:**type** see) (:**agent** John)
  (:**theme** World-Cup-2010)
  (:**instr** (telescope 1.0))
  (:**loc** South-Africa)
  …}

This is not news **…but this is!**

Relation to Information Theory

- Shannon's approach:
  - Information content is a function of the novelty (to the reader) in the message
  - Methodology: Count the number of guesses, compute probability of items and of message
  - *Info* = $\sum_{\mathrm{i}} p(x_i) \, . \log p(x_i)$
- Info Theory has no explicit record of the reader's knowledge
  - In all work, informativeness is computed relative to a (large) background knowledge store that is assumed to give default knowledge
- In Extended Semantics, the reader's knowledge can be explicitly encoded
  - Represented in individual lexical entries' score contents

Some semantic NL phenomena

Bracketing (scope) of predications

Quantifier phrases and numerical expressions

Direct quotations, reported speech

Polarity/negation

Modalities (epistemic modals, evidentials)

Comparatives

Pragmatics/speech acts

Information structure (theme/rheme)

Focus

Temporal relations (incl. discourse and aspect)

Manner relations

Spatial relations

Word sense selection (incl. copula)

Concepts: ontology definition

NP structure: genitives, modifiers…

Identification of events

Concept structure (incl. frames and thematic roles)

Pronoun classification (referential, bound, event, generic, other)

Coreference (entities and events)

Coordination

Discourse structure

Presuppositions

Opinions and subjectivity

Metaphors

Red: propositional

Blue: distributional

Two ‘modes’ of semantics?

- We need to handle **two classes** of semantic phenomena
- **Logical operations**: **Propositional**
  - Phenomena not anchored in individual open-class word meanings, but in closed-class words, and apply in general to the whole proposition
  - Examples: negation, modality, quantifier phrases, pragmatics…
  - Representation: a new proposition clause containing specific (closed-class) keywords, bracketing, etc.
  - NLP task and approach: tagging and delimiting, using CRFs for example
- **Concept content**: **Distributional**
  - Phenomena anchored in open-class word meanings
  - Examples: word senses, NP structure, coreference…
  - Representation: within a propositional clause, a selected specific term representing some element of the sentence
  - NLP task and approach: selection or tagging, using context vectors

Negation: Soccer on the moon

New semantics: **John attended the World Cup**:

(e0 (:type attend) (:agent John) (:theme WC) (:loc ((Germany 0.1)
(Italy 0.1) (Netherlands 0.1) (SA 0.1) (Argentina 0.1) …)) (:year
((2010 0.1) (2006 0.1) …)) (:accomp ((wife 0.2) (friends 0.3) …)) …)

Old: **John didn't attend the Word Cup on the moon**:

Negating *attend*:

(attend e0 x0 x1 x2) (John x0) (WC x1) (moon x2) (not e0)
(e0 (:type attend) (:agent John) (:theme WC) (:loc moon) (:polarity neg))

Negating *moon*:

(attend e0 x0 x1 x2) (John x0) (WC x1) (moon x2) (not x2)
(e0 (:type attend) (:agent John) (:theme WC) (:loc x2))
((x2 (:type moon) (:polarity neg))

Same, in new semantics:

No change! The moon's 'probability' was already zero

(e0 (:type attend) (:agent John) (:theme WC) (:loc ((Germany 0.1)
(Italy 0.1) (Netherlands 0.1) (SA 0.1) (Argentina 0.1) …)) (:year
((2010 0.1) (2006 0.1) …)) (:accomp ((wife 0.2) (friends 0.3) …)) …)

Negation with Mozart

Mozart composed a melody

Old 1: (compose e0 x0 x1) (Mozart x0) (melody x1)
(have-difficulty e1 x2 x3 x4) (= x2 x0) (= x3 e0) (= x4 0)

Old 2: (e0 (:type compose) (:agent Mozart) (:patient melody))
(e1 (:type have-difficulty) (:experiencer Mozart) (:activity e0) (:degree 0))

New: (e0 (:type compose) (:agent Mozart) (:patient melody) (:instr ((piano 0.8)
(pen 0.5) (violin 0.3) …))) (:difficulty ((0 0.6) (1 0.2) (2 0.1) … (5 0.001)))
(:loc ((Vienna 0.4) (Prague 0.1) (Paris 0.2) …)) (:time ((1762 0.5) …)) …)

It was easy for Mozart to compose a melody

(e0 (:type compose) (:agent Mozart) (:patient melody) (:instr ((piano 0.8)
(pen 0.5) (violin 0.3) …))) (:difficulty 0) (:loc ((Vienna 0.4) (Prague 0.1)
(Paris 0.2) …)) (:time ((1762 0.5) …)) …)

Negation in DS: Mozart 2

It was not difficult for Mozart to compose a melody

Old 1:
(compose e0 x0 x1) (Mozart x0) (melody x1)
(have-difficulty e1 x2 x3 x4) (= x2 x0) (= x3 e0) (val x4 (< +4))

Old 2:
(e0 (:type compose) (:agent Mozart) (:patient melody))
(e1 (:type have-difficulty) (:experiencer Mozart) (:activity e0) (:degree (< +4)))

New form "easy":
(e0 (:type compose) (:agent Mozart) (:patient melody) (:instr ((piano 0.8) (pen 0.5) (violin 0.3) …))) (:difficulty 0) (:loc ((Vienna 0.4) (Prague 0.1) (Paris 0.2) …)) (:time ((1762 0.5) …)) …)

New "not difficult":
(e0 (:type compose) (:agent Mozart) (:patient melody) (:instr ((piano 0.8) (pen 0.5) (violin 0.3) …))) (:difficulty ((0 0.5) (1 0.3) (2 0.2) (3 0.1))) (:loc ((Vienna 0.4) (Prague 0.1) (Paris 0.2) …)) (:time ((1762 0.5) …)) …)

General schema for operators

- In traditional semantics, operators within propositions apply over terms and clauses:
  - NOT(x), AND(x, y), etc.
  - Their specific action is manifest in the eventual result of composition
- In new semantics, operators probably (?) apply to the distributional scores
  - NOT(sad) –> happy
  - For each operator, we somehow need to determine *which* scores change, and how

Inverses: Some adjs

rel:adm	too	very	so	unusually	as	bitterly	really	how	not	relatively	running	extremely	pretty	unseasonably
COLD	694	643	517	230	218	203	165	131	106			95	74	138
HOT	1370	1351	1166	101	571		415	285	166			158	164	36
LONG	5268	3557	1610	260	975		229	117	930	112		132	161	
SHORT	1285	1363	516		270		99	74	66	838	161	101	57	2

rel:anm	season	month	time	year	week	day	night	period
COLD	3	3	3		2		4	
HOT	9	8	4	19	7	27	4	
LONG		3	3	8	5	7	4	5
SHORT	7		18	12	4	6		3

rel:nam	time	water	weather	air	summer	day	term	period	rain	beer	sauce	notice	career	distance
COLD		2744	2345	1341		394			320	316				
HOT			900		962						596			
LONG	32603					1895	6977	3619					1470	
SHORT	785						5273	2225				810		792

rel:avc	be	have	get	make	go	do	swim	cook	wait	spend	freeze	handle	become
COLD	23	27					6				19		
HOT	92	29	13	12	11	6		10				74	
LONG	154	48	96	39	18	24			41				23
SHORT	55	25	10	6	10	10				5			

New York Times corpus

In many regards, these [physical] adjs are all the same…

rel:adm	too	very	so	unusually	as	bitterly	really	how	not	relatively	running	extremely	pretty	unseasonably
COLD	0.216	0.200	0.161	0.072	0.068	0.063	0.051	0.041	0.033	0.000	0.000	0.030	0.023	0.043
HOT	0.237	0.234	0.202	0.017	0.099	0.000	0.072	0.049	0.029	0.000	0.000	0.027	0.028	0.006
LONG	0.395	0.266	0.121	0.019	0.073	0.000	0.017	0.009	0.070	0.008	0.000	0.010	0.012	0.000
SHORT	0.266	0.282	0.107	0.000	0.056	0.000	0.020	0.015	0.014	0.173	0.033	0.021	0.012	0.000

rel:anm	season	month	time	year	week	day	night	period
COLD	0.200	0.200	0.200	0.000	0.133	0.000	0.267	0.000
HOT	0.115	0.103	0.051	0.244	0.090	0.346	0.051	0.000
LONG	0.000	0.086	0.086	0.229	0.143	0.200	0.114	0.143
SHORT	0.140	0.000	0.360	0.240	0.080	0.120	0.000	0.060

…especially here

But sometimes opposites pair together…

rel:nam	time	water	weather	air	summer	day	term	period	rain	beer	sauce	notice	career	distance
COLD	0.000	0.368	0.314	0.180	0.000	0.053	0.000	0.000	0.043	0.042	0.000	0.000	0.000	0.000
HOT	0.000	0.000	0.366	0.000	0.391	0.000	0.000	0.000	0.000	0.000	0.242	0.000	0.000	0.000
LONG	0.700	0.000	0.000	0.000	0.000	0.041	0.150	0.078	0.000	0.000	0.000	0.000	0.032	0.000
SHORT	0.079	0.000	0.000	0.000	0.000	0.000	0.533	0.225	0.000	0.000	0.000	0.082	0.000	0.080

rel:avc	be	have	get	make	go	do	swim	cook	wait	spend	freeze	handle	become
COLD	0.307	0.360	0.000	0.000	0.000	0.000	0.080	0.000	0.000	0.000	0.253	0.000	0.000
HOT	0.372	0.117	0.053	0.049	0.045	0.024	0.000	0.040	0.000	0.000	0.000	0.300	0.000
LONG	0.348	0.108	0.217	0.088	0.041	0.054	0.000	0.000	0.093	0.000	0.000	0.000	0.052
SHORT	0.455	0.207	0.083	0.050	0.083	0.083	0.000	0.000	0.000	0.041	0.000	0.000	0.000

…though in some aspects they remain unique

CONCLUSION

Summary

- Combine older logic-style and newer word distribution-style representations into single form
- Treat this as a new semantics
- Scale-independent notation
- Compositionality using large Proposition Stores
- Use their contents to assist with various NLP tasks
- Negation and modality seem to be feasible in new semantics

Where next?

- Careful and formal definition of semantics:
  - Theoretical connections to Formal Semantics
  - Proper treatment of synonymy and composition
  - Algebra-like machinery for concept manipulation (composition, negation, etc.)
  - Generalize Topic Models and Topic Signatures
- Empirical usage in various NLP and KR applications:
  - Tasks: Parsing, (co)reference, WSD, etc.
  - Applications: QA, Machine Reading, IR, etc.
  - Reasoning and inference in KR
  - Semantic Web research
- Other fields:
  - Connection to Information Theory
  - Predictions and confirmation with Cognitive Science, Psycholinguistics, etc.

Readings

- Formal models
  - Preference Semantics: Wilks, 1975
  - Turney: several papers since 2005
  - Novacek, PhD 2010
  - D. Clarke, CL 2011; PhD thesis 2007
- Topic modeling
  - LSA: Deerwester et al., 1990
  - LSA; Landauer et al., 1998
  - Signatures Lin and Hovy, COLING 2000
  - LDA: Blei et al., 2003
  - Many others
- Word meaning vector models
  - Lin, 1998; and Pantel, 2003
  - Navigli, PhD 2008
  - Turney, several papers
  - Erk, ACL 2010 and earlier
  - Budanitsky and Hirst, *CompLing* 2006
- Triple Stores and PropStores
  - Lots of background on triples
  - P. Clark, K-CAP 2009
- Organizing vectors into hierarchies and finding default values
  - Turney and Pantel, 2010
  - Tan and Hovy, in prep
- Compositionality: Combining vectors
  - Mitchell and Lapata, Cognitive Science 2010; Lapata et al.. HLT 2009
  - Erk and Padó; Pinkal et al., on vector comb
  - Ritter et al., ACL 2010
  - Coecke, et al. 2010
  - Baroni and Lenci, CL 2010
  - Grefenstette et al., 2010
  - Socher, Manning, et al., 2011
  - Mohammad and Hirst, 2012
- Word/concept facets
  - Fillmore, Case for Case 1967
  - Guarino, Identity Criteria 2001
  - Pustejovsky, Generative Lexicon 1995
  - Fillmore et al., FrameNet
  - Recasens and Hovy, Near-Identity 2010
- Using DS for NLP tasks
  - Parsing: Klein, ACL 2010
  - WSD: Agirre et al.
  - Paraphrase learning: Pantel and Pennacchoitt, 2008
  - Text enrichment: Peñas and Hovy, COLING 2010
  - Coref: many people

THANK YOU