OBITUARIES

FREDERICK JELINEK, 1932 - 2010

Pioneer of speech recognition systems

MICHAEL DRESSER

Frederick Jelinek, an electrical engineering professor who was a pioneer in creating the technology that allows computers to interpret human speech and translate languages, died Sept. 14 of a heart attack in his office at Johns Hopkins University in Baltimore. He was 77.

In more than 40 years at IBM Research and Johns Hopkins, Jelinek led the way in developing the statistical theory behind modern voice-recognition systems. Essentially, he helped turn a nascent science that merely transcribed human speech into a sophisticated one that could interpret meaning and anticipate what the speaker would say next.

"He envisioned applying the mathematics of probability to the problem of processing speech and language," said Sanjeev Khudanpur, associate professor of electrical engineering at Johns Hopkins. "This revolutionized the field. Fifty years ago no one thought that was possible. Today, it's the dominant paradigm."

Born Nov. 18, 1932, to a Jewish father being ousted from their home by Nazi occupiers, said Jelinek's son William. Jelinek's father died of disease in the concentration camp at Terezin shortly after the Allied liberation.

In 1949, the family moved to the United States. He earned a bachelor's degree at the Massachusetts Institute of Technology in 1956. He stayed on at MIT to earn a master's degree in 1958 and a doctorate in 1962.

William Jelinek said his father traveled in 1957 to a professional conference in what was then Czechoslovakia, where he met and fell in love with Milena Tobolova, a filmmaker and dissident against the Communist government.

For years after that, Tobolova was barred from leaving the country, her son said. But during a visit by Soviet leader Nikita Khrushchev to the United States, Jelinek's academic adviser Jerome Wiesner, who was also a science aide to then-Sen. John F. Kennedy, asked Khrushchev to intervene with Czech authorities. Soon after Kennedy was elected president, Tobolova was allowed to emigrate.

"As an inaugural gift to Kennedy, the Czechs released nine dissidents and one of them was my

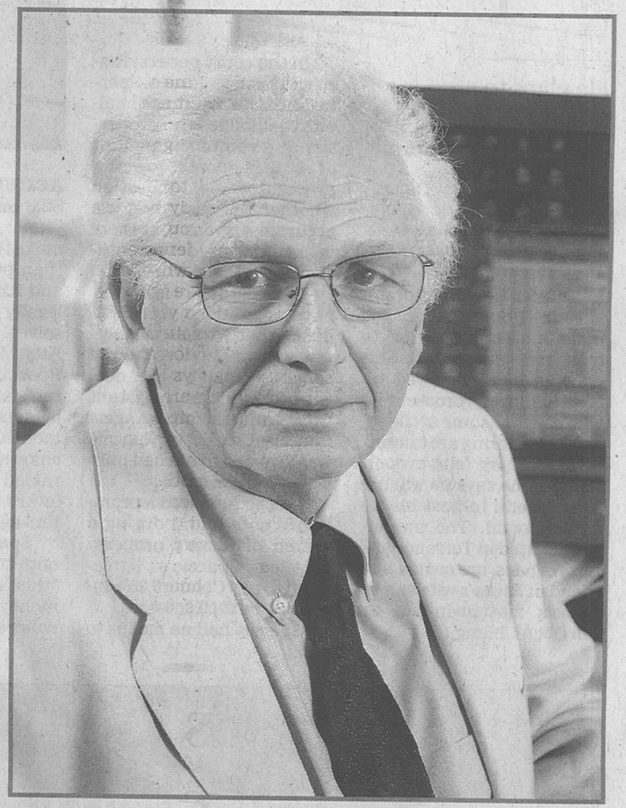
versity.

In 1972, Jelinek accepted a summer position at IBM Research, which was just beginning to work on speech recognition. Eventually, he said, he was forced to decide between Cornell and his expanding role in IBM Research. He chose IBM, where he worked for 21 years and headed a team that sought to apply the power of supercomputers to the challenges of transcribing and translating the spoken word.

Khudanpur said that previous efforts at voice recognition and translation focused on codifying rules and applying them — an approach that was frustrated by the complexity and subtlety of language. Jelinek's approach was to assemble a huge database of text and let the computer calculate the probabilities of words appearing in relation to other words — deriving meaning from context rather than rules.

The strategy was widely questioned at the time, but when the Defense Advanced Research Projects Agency sponsored a competition in the field in 1980, Khudanpur said, Jelinek's approach prevailed.

"By the '90s, everybody was on



Baltimore Sun

ENGINEERING PROFESSOR
Frederick Jelinek's approach — a huge database of text from which a computer could calculate the probabilities of words appearing in relation to other words — was a breakthrough.

employ artificial intelligence, including stock market prediction

in 2006.

A New Semantics: Merging Propositional and Distributional Approaches

Eduard Hovy

Information Sciences Institute
University of Southern California
hovy@isi.edu

USC INFORMATION SCIENCES INSTITUTE

To side topic:
intension and extension

Two styles of representing semantics

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 was strong

Logic: (have-property e0 x0)
(table x0) (green x0) (strong x0)

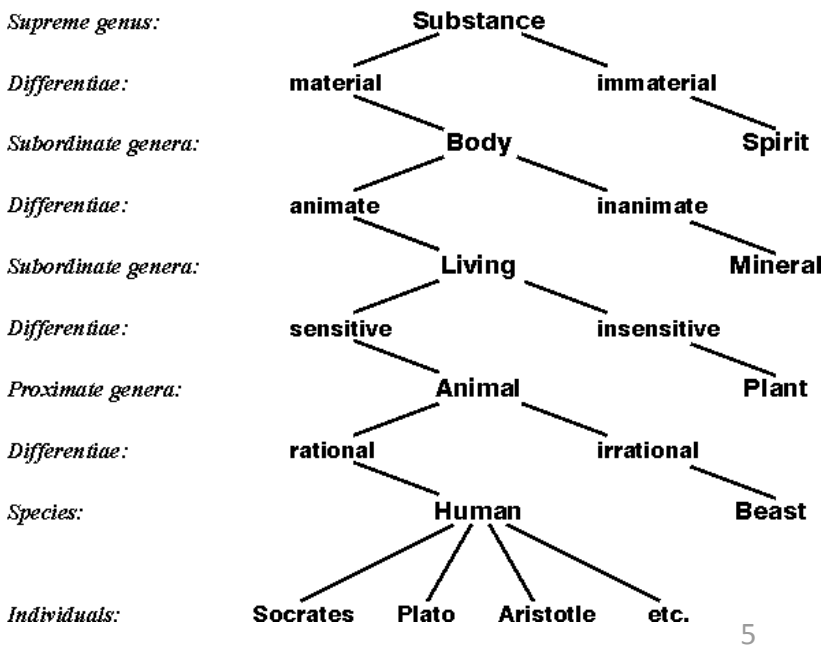
Frame: (x0 (:type table) (:colour green) (:strength +5))

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 the **content**: representing explicitly **what the propositions are about**
  - The propositions provide relationships among the symbols, but leave to the Denotational Model what the symbols 'mean'

Concept definition: Intensional approach

- Back to Aristotle:
  - A concept is described by a collection of features
  - Starting from the most general concept, you add increasingly specific differentiae, to eventually assemble all definitional features of a particular concept
  - Example from (Sowa, 2000):



- Leibnitz used this approach: Terms in the logic stand for (collections of) properties or concepts, rather than for the things having these properties

Extensional approach

- Intensional approach sounds nice, but…

  Have *you* ever tried to define a table? Anything else?

  Have you ever seen anyone's definition using this method?

- In contrast, the **extensional** approach:
  - 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 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 (in AI for example)
  - 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 today's theories

- 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
- Symbols are discrete
  - Yet meanings are shaded, spread in a continuum toward different directions of nuance
- Semantic theories show no direct connections with psycholinguistic or cognitive phenomena
  - No obvious explanations for confusions, forgetting, degrees of processing complexity, etc.

INTRO: TRADITIONAL SEMANTICS

DISTRIBUTIONAL SEMANTICS

1. TOPIC MODELS

2. WORD MODELS

A NEW MODEL OF SEMANTICS

COMPOSITIONALITY 1: VECTORS

BUILDING A SEMANTIC LEXICON

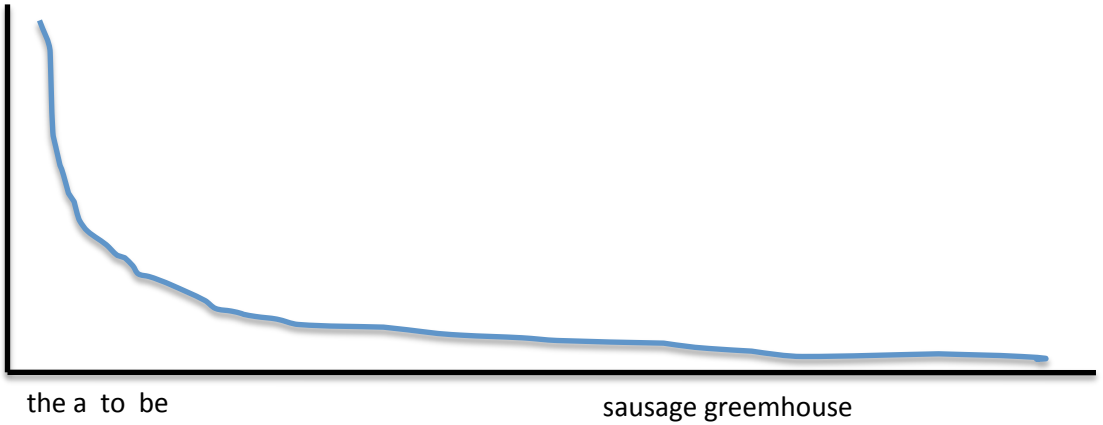
RECENT EXPERIMENTS AT ISI

COMPOSITIONALITY 2: OPERATORS

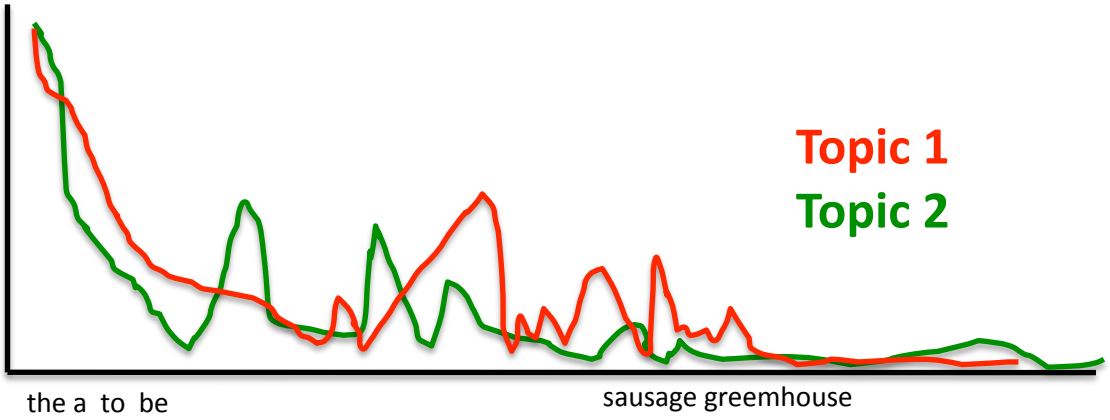
CONCLUSION

Theoretical basis for distributional semantics

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



- But locally, words appear in a Poisson distribution:



Using word vectors

- "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
  - **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

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 and paraphrases…
- Problem: No explicit theory of how this works

1. TOPIC MODELS

Def: Topic Signature (Lin and Hovy, COLING-00)

- Definition: A **Topic Signature** T is a head word plus a set of related words w, each with a strength s:

  $\{ T_k, (w_{k1},s_{k1}), (w_{k2},s_{k2}), \ldots , (w_{kn},s_{kn}) \}$

- Approximate relatedness by simple term co-occurrence...

- Example study (Lin and Hovy 1997):
  - Corpus
    - Training set WSJ 1987:
      - 16,137 texts (32 topics)
    - Test set WSJ 1988:
      - 12,906 texts (31 topics)
    - Texts indexed into categories by WSJ
  - Signature data
    - 300 terms each, using *tf.idf*
    - Variations: single words, demorphed words, multi-word phrases
    - Created topic hierarchy

RANK	ARO	BNK	ENV	TEL
1	contract	bank	epa	at&t
2	air_force	thrift	waste	network
3	aircraft	banking	environmental	fcc
4	navy	loan	water	cbs
5	army	mr.	ozone	cable
6	space	deposit	state	bell
7	missile	board	incinerator	long-distance
8	equipment	fslic	agency	telephone
9	mcdonnell	fed	clean	telecomm.
10	northrop	institution	landfill	mci
11	nasa	federal	hazardous	mr.
12	pentagon	fdic	acid_rain	doctrine
13	defense	volcker	standard	service
14	receive	henkel	federal	news

Calculating weights for Topic Signatures

Approximate relatedness using various formulas

$tf.idf$: $w_{jk} = tf_{jk} * idf_j$

χ^2 : $w_{jk} = \begin{cases} (tf_{jk} - m_{jk})^2 / m_{jk} & \text{if } tf_{jk} > m_{jk} \\ 0 & \text{otherwise} \end{cases}$

(Hovy & Lin, 1997)

- tf_{jk} : count of term j in text k ("waiter" often only in some texts)
- $idf_j = log(N/n_j)$: within-collection frequency ("the" often in all texts)
  n_j = number of docs with term j, N = total number of documents
- *tf.idf* is the best for IR, among 287 methods (Salton & Buckley, 1988)
- $m_{jk} = (\Sigma_j tf_{jk} \Sigma_k tf_{jk}) / \Sigma_{jk} tf_{jk}$: mean count for term j in text k

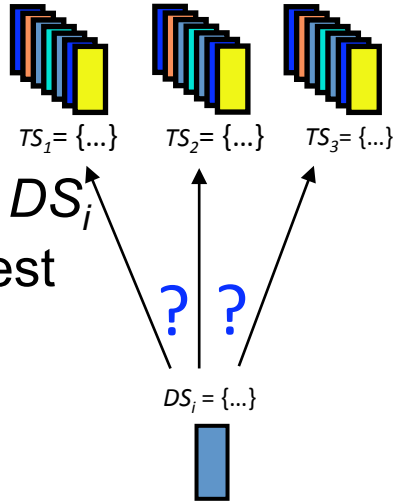
likelihood ratio λ : $2log\ \lambda = 2N . I(R ; T)$ (Lin & Hovy, 2000)

(more approp. for sparse data; $-2log\lambda$ asymptotic to χ^2)

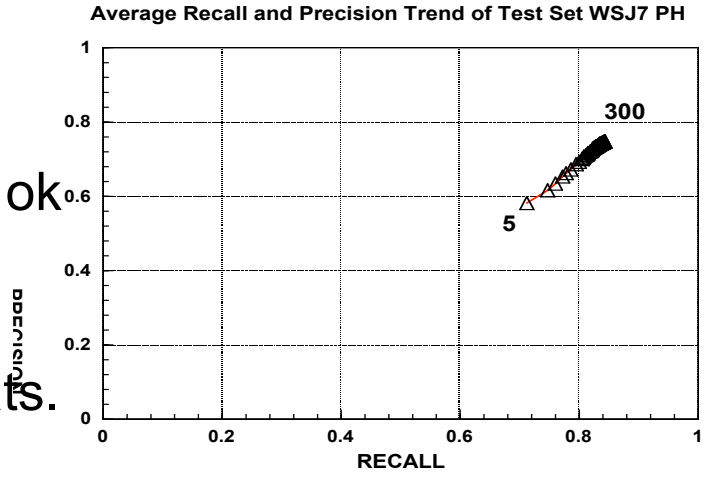
- N = total number terms in corpus
- I = mutual information between text relevance R and given term T
  = $H(R) - H(R \mid T)$ for $H(R)$ = entropy of terms over relevant texts R
  and $H(R \mid T)$ = entropy of term T over rel and nonrel texts

Evaluating Topic Signatures

- **Test**: Perform text categorization task:
  - create N sets of texts, one per topic
  - create N topic signatures TS_k
  - for each new document, create document signature DS_i
  - compare DS_i against all TS_k; assign document to best
- **Matching function**: vector space similarity measure:
  - Cosine similarity, $cos\ \theta = TS_k \cdot DS_i\ /\ |TS_k||DS_i|$



- **Test 1** (Hovy & Lin, 1997, 1999)
  - Training: 10 topics; ~3,000 texts (TREC)
  - Contrast set (background): ~3,000 texts
  - Conclusion: *tf.idf* and χ^2 signatures work ok but depend on signature length
- **Test 2** (Lin & Hovy, 2000):
  - 4 topics; 6,194 texts; uni/bi/trigram signats.
  - Evaluated using SUMMARIST: λ > *tf.idf*



To side topic:
LSA, LDA, etc.

Topic Models

- A *Topic* consists of a cluster of words that frequently occur together. Using contextual clues, topic models can connect words with similar meanings and distinguish between uses of words with multiple meanings
- General introduction **Probabilistic Topic Models** by Steyvers and Griffiths (2007)
- **Latent Semantic Analysis** (LSA): Matrix operation over texts that groups the words into 'latent' (hidden) classes (Deerwester et al., 1990)
- **Latent Dirichlet Allocation** (LDA) is a graphical model for topic discovery (Blei, Ng, and Jordan, 2002)
- Many packages:
  - UMass MALLET: http://mallet.cs.umass.edu/topics.php
  - Stanford Topic Modeling Toolbox: http://nlp.stanford.edu/software/tmt/tmt-0.2/

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 which have been fashioned in the course of its use."
- 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)

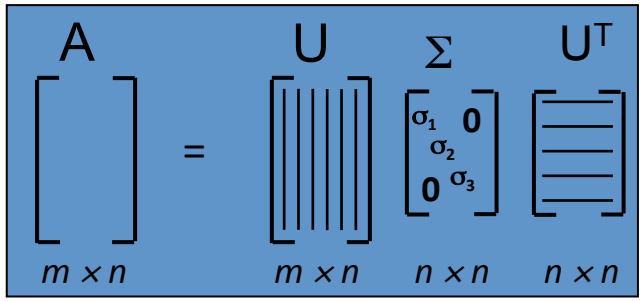
Latent Semantic Analysis

- Also called Singular Value Decomposition (SVD) or Principal Components Analysis (PCA)
  - Used by engineers to determine essential elements in complex data problems
  - Used by psychologists to determine basic cognitive conceptual primitives (Deerwester et al., 1990; Landauer et al., 1998)
  - In text processing, used for text categorization, lexical priming, language learning…
- LSA automatically creates collections of items that are correlated or anti-correlated, with strengths:

  ice cream, drowning, sandals ⇨ summer
- Each such collection is a 'semantic primitive' in terms of which objects in the world are understood
- Can use LSA to find most reliable signatures in a collection—reduce number of signatures in contrast set

LSA for signatures

- Create matrix A, one signature per column (words × topics).
- Apply SVDPAC to compute U so that $A = U \Sigma U^T$:
  - U : $m \times n$ orthonormal matrix of left singular vectors that span space
  - U^T : $n \times n$ orthonormal matrix of right singular vectors
  - Σ : diagonal matrix with exactly *rank(A)* nonzero singular values; $\sigma_1 > \sigma_2 > \ldots > \sigma_n$



- Use only the first *k* of the new concepts: $\Sigma' = \{\sigma_1, \sigma_2 \ldots \sigma_k\}$.
- Create matrix A′ out of these *k* vectors: $A' = U \Sigma' U^T \approx A$.

A′ is a new (words × topics) matrix, with different weights and new 'topics'. Each column is a purified signature.

Probabilistic LSA (Hofmann, SIGIR-99)

- Pick a doc *d*; pick a (latent) topic *z* in that doc; pick a word *w* from the topic; then you get a pair (*d,w*). Do this many times over, and discard *z*:

$$p(d,w) = p(d) \cdot p(w \mid d)$$
$$p(w \mid d) = \Sigma_z\, p(z \mid d) \cdot p(w \mid z)$$

$$\Big\} \; p(d,w) = p(d) \cdot \Sigma_z\, p(z \mid d) \cdot p(w \mid z)$$

- Assumptions:
  - You have lots of docs, but only a small number of topics
  - Each doc is a specific mixture of topics (with weight $p(z|d)$)
  - Conditional independence: words are generated from topics *regardless of docs*
  - The (weighted) combination of topics constituting a doc generates the actual words in the doc — bag-of-words model

- Obtain parameters by maximizing

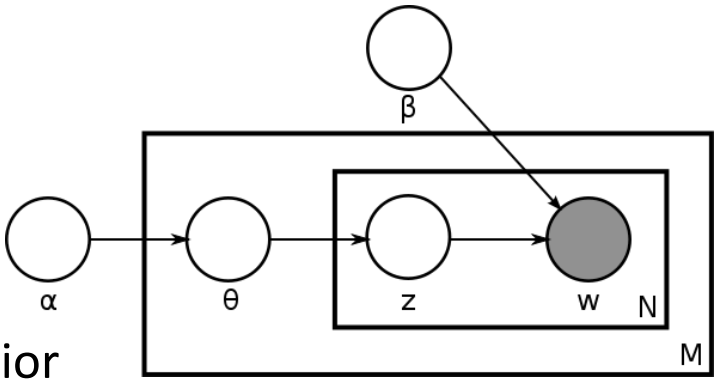
$$L = \Sigma_d \Sigma_w\, n(d,w) \cdot \log p(d,w) \quad \text{where } n(d,w) = \text{freq of } w \text{ in } d$$

using EM algorithm

Latent Dirichlet Allocation (Blei et al., 2003)

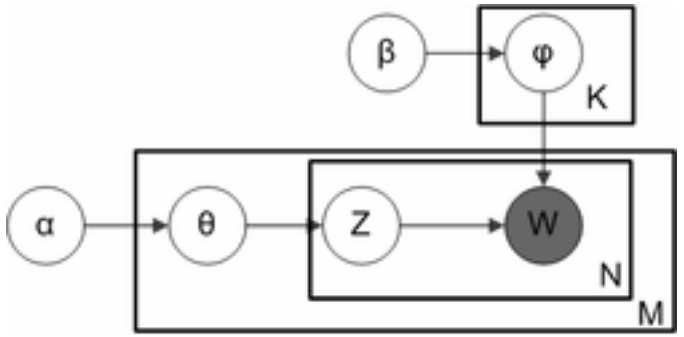
- Current hot topic in NLP (also see Wikipedia)
- LDA is a generative model that allows sets of observations to be explained by hidden (unobserved) groups that recombine observations to explain why some parts of the data are similar
- Example: Observe **documents** as **sets of words**. LDA sees **each document as a mixture of a small number of hidden topics**; where each topic generates a set of words. The topics and words are scored for best fit to documents. LDA returns the document's topics as sets of its words, with 'strength' scores

LDA, intuitively



- Parameters:
  - α : parameter of the uniform Dirichlet prior topic distribution per document
  - θ_i : topic distribution for document *i*
  - β : parameter of the uniform Dirichlet prior word distribution per topic
  - z_{ij} is the topic for the *j*th word in document *i*
  - w_{ij} is a specific word. The w_{ij} are the only observable variables; all the other variables are latent
- Training:
  - User provides the number of topics desired/expected
  - Algorithm starts with a random distribution of topic strengths
  - Cyclic approximation phase ('burn-in'): Each topic generates words; each topic 'belongs to' documents; the words combine to form the documents … rearrange the topic and word strength distributions to maximally fit the observed documents
  - Selection phase, after the burn-in: User selects a dozen or so answer sets (one every 100 or so iterations) and picks the one that seems best

Extensions to LDA



- Usually use smoothed version for better results:
  - *K* : number of topics considered in the model
  - *φ* : *K*V* (*V* is the dimension of vocabulary) Markov matrix, each line giving the word distribution of a topic
- To nudge learning algorithm in right direction, can sample data and provide better initial parameter distributions (Gibbs sampling, etc.)
- LDA is similar to PLSA (LDA model is essentially the Bayesian version of PLSA)

Summary: Topic Models

- A word[sense] is just a very small topic
  - Its content is represented the same way a topic's is: a vector of words with 'strengths'
- A document is a (weighted) collection of topics, but they are hidden
- A topic is also a collection of weighted words

- Is this horribly recursive, or what?

- Questions, and research to do:
  - How many topics *should* there be?

2. WORD MODELS

Building word models

- Typically, each word is modeled by its context vector:
  - Each vector represents the 'average meaning' of a word
  - Collect many sentences containing the target word
  - Use some association formula to collect the words that co-occur with it more than they 'should' on average
  - Pointwise mutual information (PMI) is popular:

$$PMI(w_1,w_2) = \log [\, p(w_1,w_2) \;/\; (p(w_1) \;.\; p(w_2)) \,]$$

  - Typically used for wordsense disambiguation (each sense has its characteristic vector), sense clustering, etc.
- Word mention models (e.g., (Erk 2008)):
  - Compute vector using just words from current sentence

Some early work

- Work on word-level context vectors
  - Schütze 1998: 'first-order' vector of co-occurrence words over corpus; then 'second-order' vector for a word in context ('single-use meaning')
  - For lexicons: Navigli PhD thesis; McCarthy and Navigli 2007
  - In Cognitive Science: Landauer and Dumais 1997; McDonald and Ramscar 2001
- Using them: example
  - Pantel and Lin 2002; Pantel et al. 2006 (and much subsequent work): Given one or two anchor words, find all associated phrases in the corpus; compute vectors from them for the anchored region; find other words that can replace the anchors
  - This is now one of the standard methods to learn paraphrases

Contexts for learning models

- 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

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

Lincoln

-V:obj:N 1869 times:

- {V1662 offer, provide, make} 156, have 108, {V1650 go, take, fly} 51, sell 45, {V1754 become, remain, seem} 34, … give 24, {V1647 oppose, reject, support} 24, buy 21, {V1653 allocate, earmark, owe} 21, win 20 …

-N:conj:N 536 times:

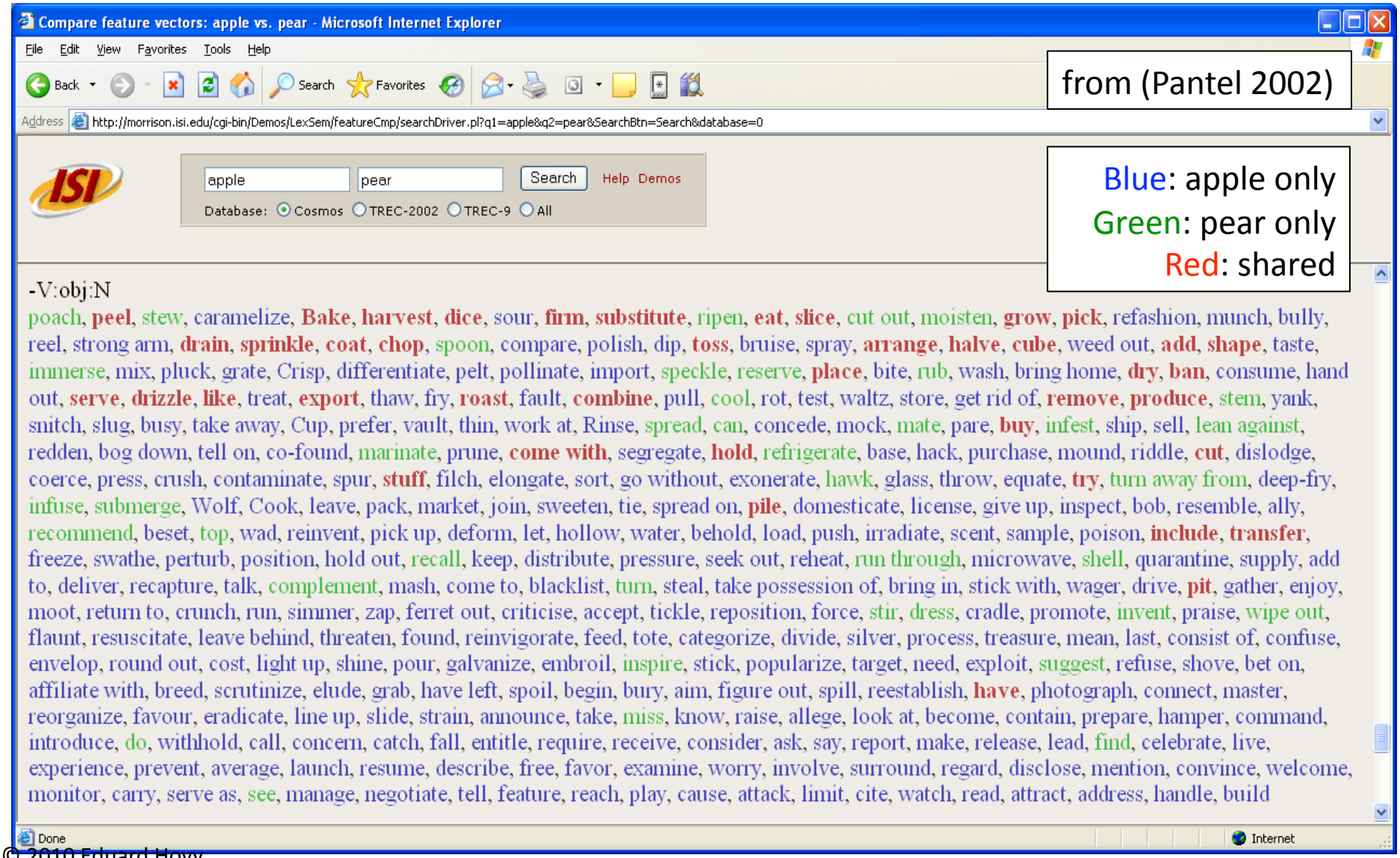
- {N719 Toyota, Nissan, BMW} 65, {N257 Cadillac, Buick, Lexus} 59, {N549 Philadelphia, Seattle, Chicago} 41, American Continental 20, Cadillacs 11, …

-V:by:N 50 times:

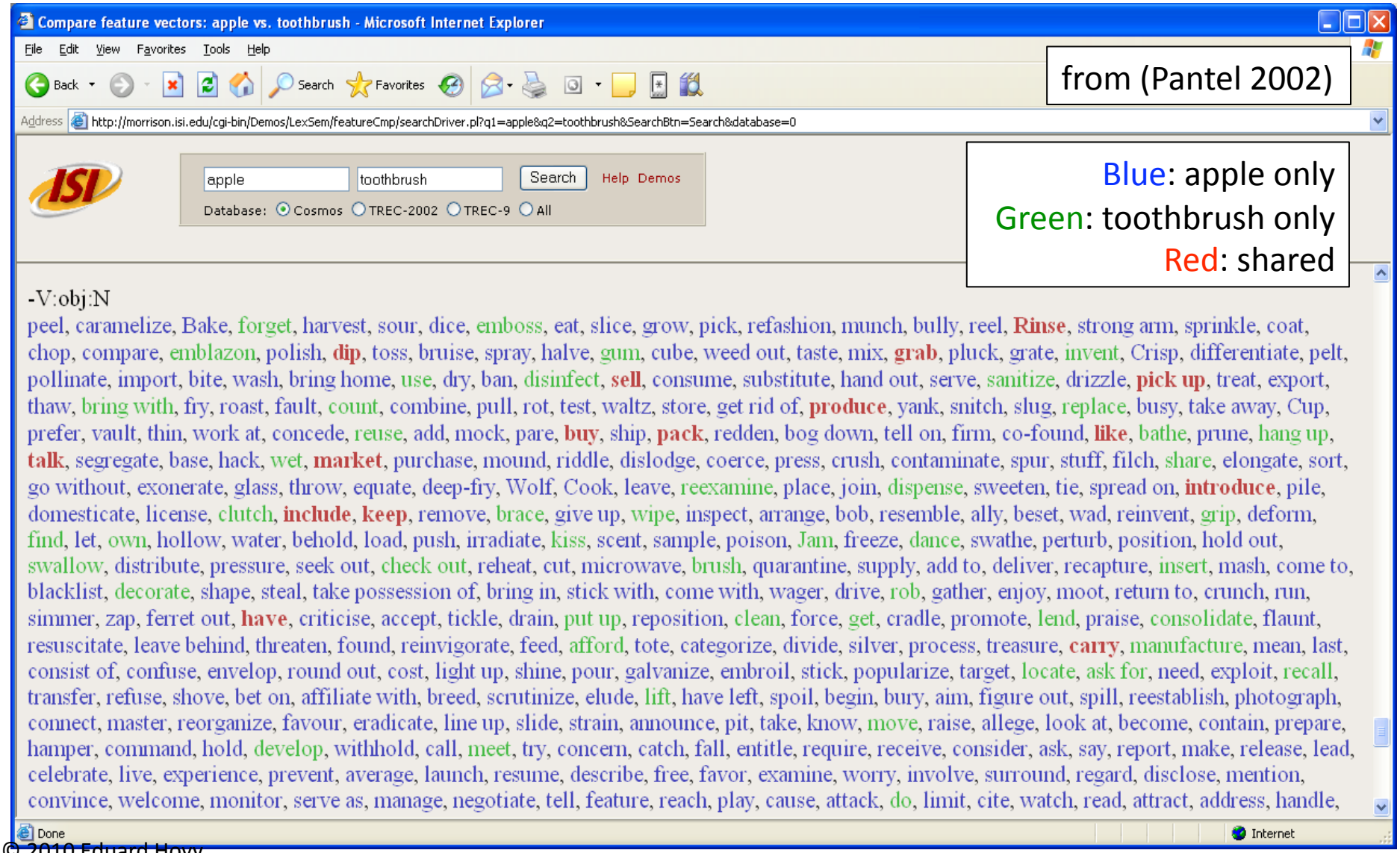
- {V1662 offer, provide, make} 12, own 5, hire 4, target 4, write 3, buy 2, …

Why is *apple* is similar to *pear*

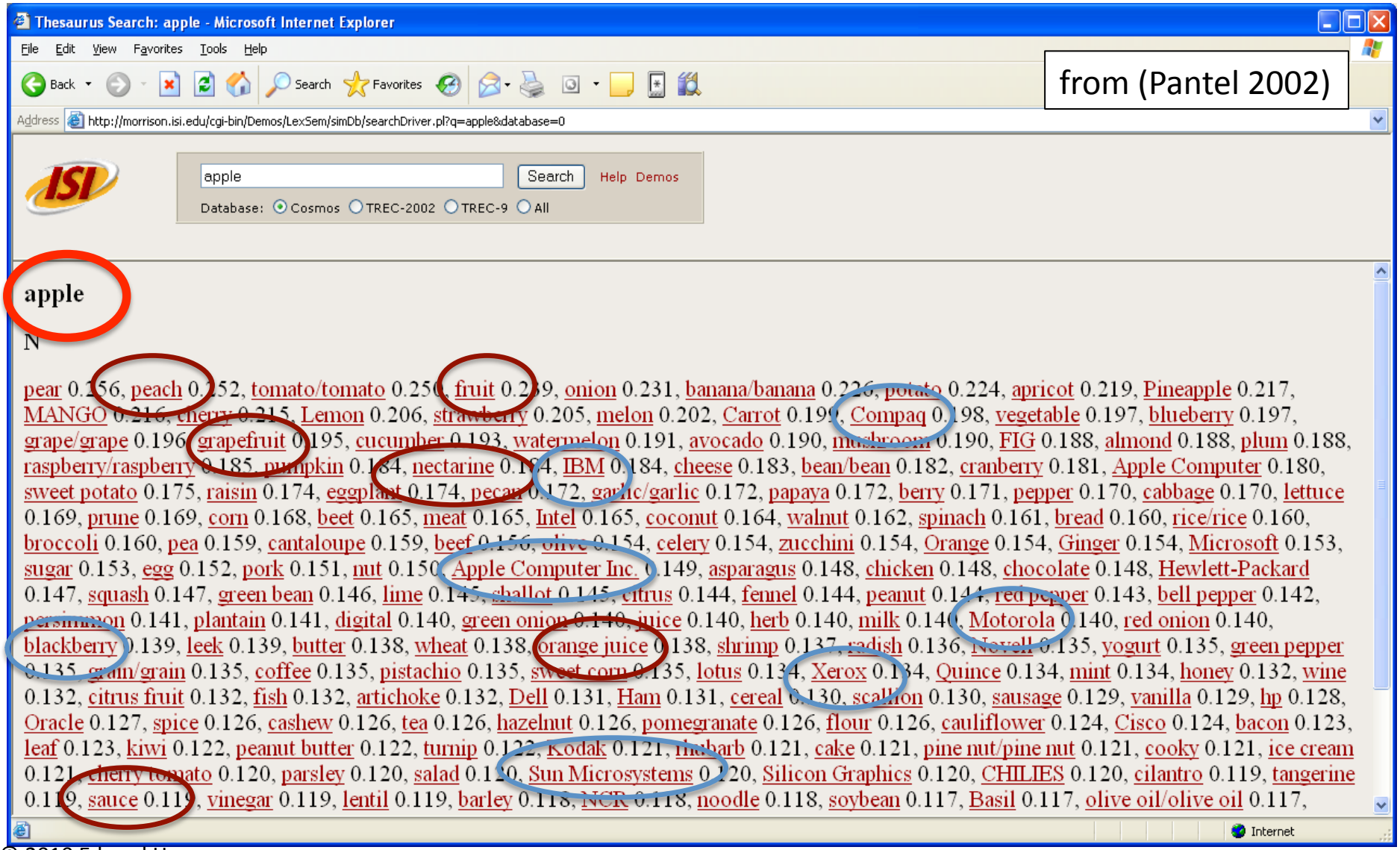
(Pantel 02)



Why *apple* is not similar to *toothbrush*



In word vectors, senses are mixed up

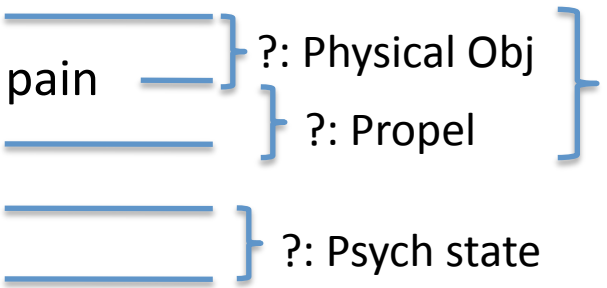


from (Pantel 2002)

To side topic:
Sense differentiation

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
    - The news drove stock prices down
    - This computer drives me crazy
    - Drive the devils out of her!



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) …)

Example applications of word models

- Word sense disambiguation (Agirre et al.):

Sense	Word Signature
Waiter1	*restaurant, waitress, dinner, bartender, dessert, dishwasher, aperitif, brasserie, . . .*
Waiter2	*hospital, station, airport, boyfriend, girlfriend, sentimentalist, adjudicator, . . .*

- 'Explanation' generation (Vyas and Pantel, COLING 08):

Word set	'Explanation' for set
Palestinian-Israeli, India-Pakistan	talks(NN), conflict(NN), dialogue(NN), relation(NN), peace(NN)

Summary: Word Models

- Word-level concepts can be defined using structured vectors. Already used for WSD and other tasks
- Questions, and research to do:
  - Word sense delimitation
  - Word facet/aspect determination (see later)
  - The strength/probability questions: Cognitive and psycholinguistic evidence for various aspects of the word/ concept definitions, including strengths of associations, etc.
  - Construction of word-level 'concept lexicons': corpora, speed, etc.
  - Handling incomplete corpora — unseen cases
  - Multilinguality: cross-language 'concept' definitions

INTRO: TRADITIONAL SEMANTICS

DISTRIBUTIONAL SEMANTICS

1. TOPIC MODELS
2. WORD MODELS

A NEW MODEL OF SEMANTICS

COMPOSITIONALITY 1: VECTORS

BUILDING A SEMANTIC LEXICON

RECENT EXPERIMENTS AT ISI

COMPOSITIONALITY 2: OPERATORS

CONCLUSION

For semantics: What would we like?

- Combine the properties of the traditional semantics and the statistical word family 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

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

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) … }

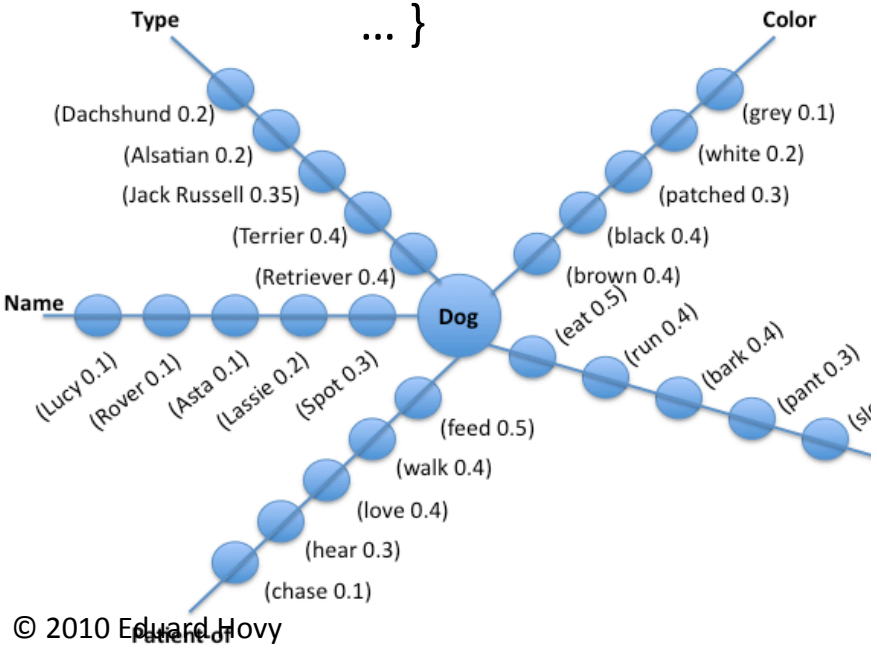
- A **Language Model** in ASR and NLP and MT is the same thing, but allows ngrams instead of words

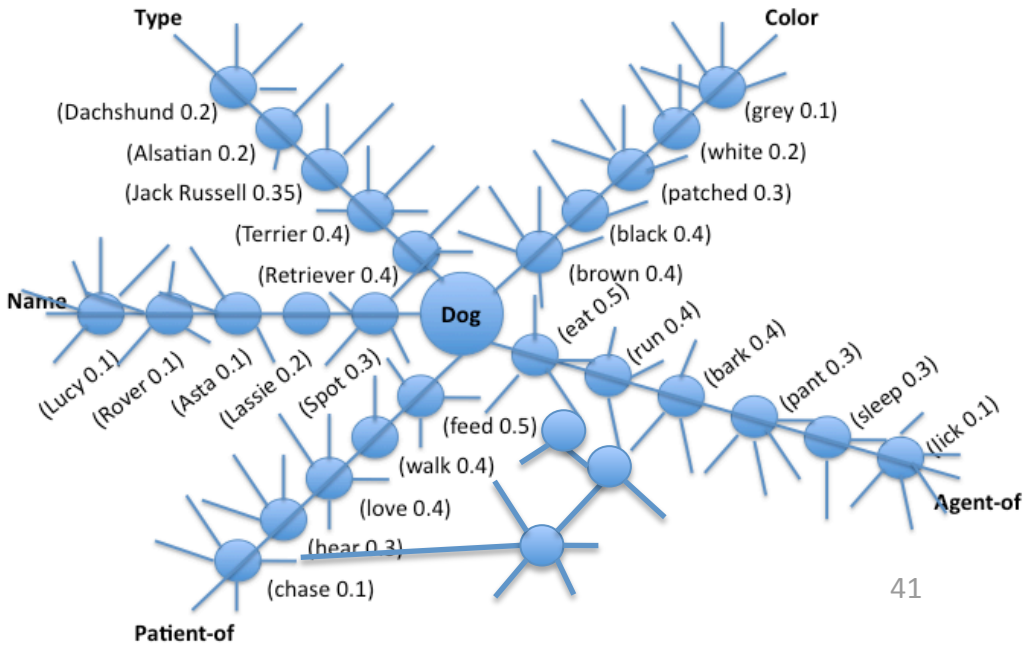
  domain = {("brown dog" 0.0000016)
  ("the brown" 0.0000032) … }

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 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 part 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 $syn(C_i,C_j) \rightarrow 1$
  - *syn(A,B)* must be defined as a continuous-valued function, transitive, but not necessarily obeying the triangle inequality

The knowledge base

Lassie the dog

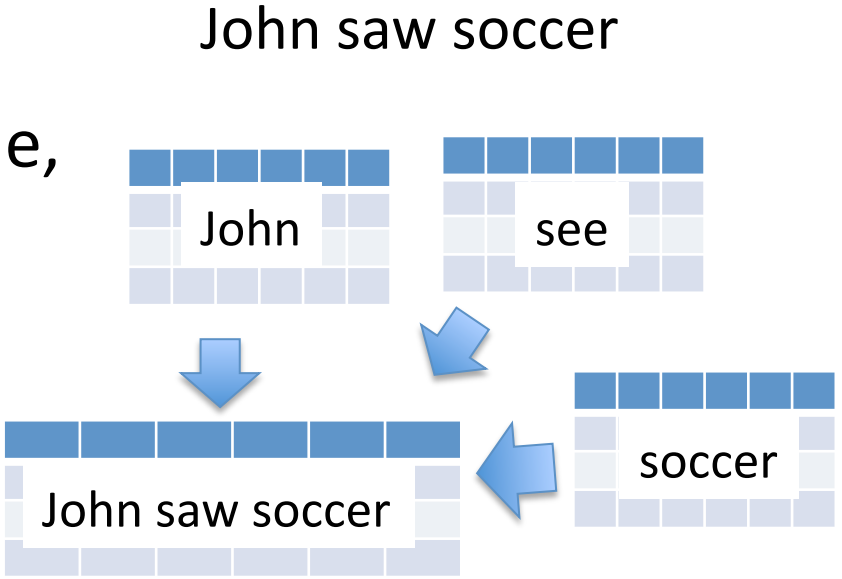
Eating lunch today

Mozart composed on Aug 18, 1772

	Lassie	W.A. Mozart	…	music	dog	…	WW I	Lunch today	…	lunch	composing	…
:type					1.0			1.0		1.0	1.0	
:name	1.0											
:age												
…												
:shape												
:color												
…												
:agent		1.0										
:theme				0.9								
:loc												
…												
:cause												

Instances

- When propositions are made, their representations are composed from their components' tensors, 'overlaid'



- Questions:
  - How does composition affect the tensor scores?
  - How are multiple instances of the same entity or event kept apart?

Compositionality problem

Dependency problem

Composition changes scores

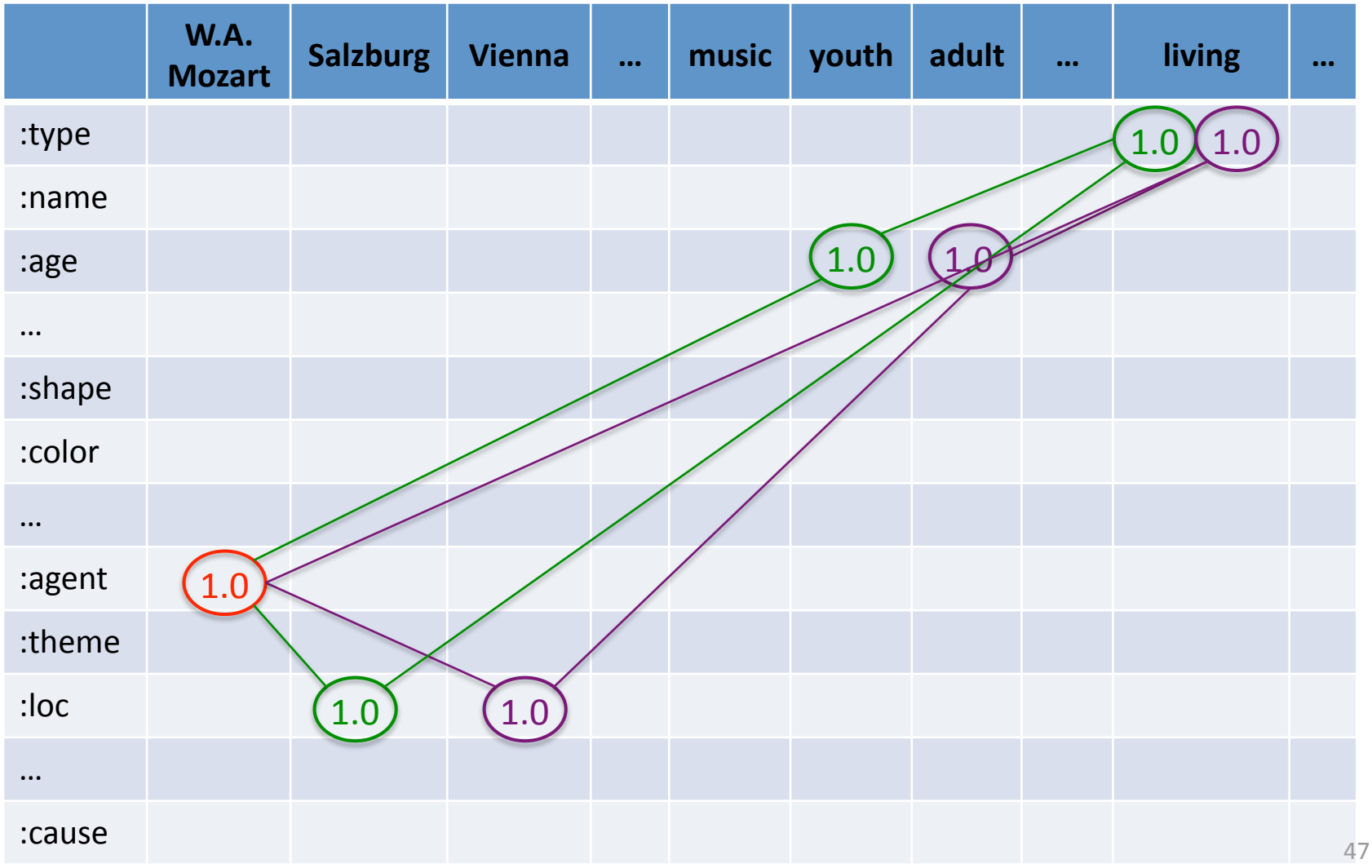
John sees soccer	John	S. Afr.	Switz	…	human	eye	…	Lunch today	…	seeing	soccer game	…
:type					1.0							
:name	1.0											
:age												
:nation		1.0										
:loc												
:agent-of								0.7		0.99		
…												
:agent	1.0				1.0					1.0		
:theme											0.001	
:instr						1.0						
:loc		0.7										
…												

A Swiss John seeing soccer

John2 sees soccer	ohn	S. Afr.	Switz	…	human	eye	…	Lunch today	…	seeing	soccer game	…
:type					1.0							
:name	1.0											
:age												
:nation			1.0									
:loc					1.0					1.0		
:agent-of								0.7		0.99	1.0	
…						1.0						
:agent	1.0											
:theme												
:instr												
:loc			0.7									
…												

Dependencies

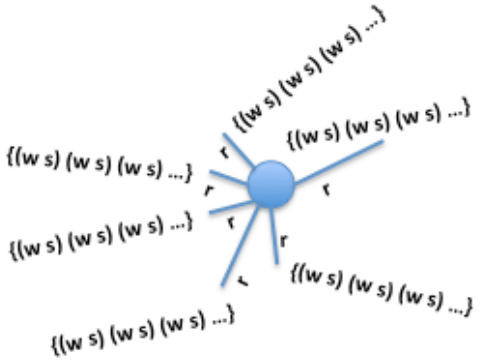
When Mozart was young he lived in Salzburg; when he was an adult he lived in Vienna



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) …)) …}

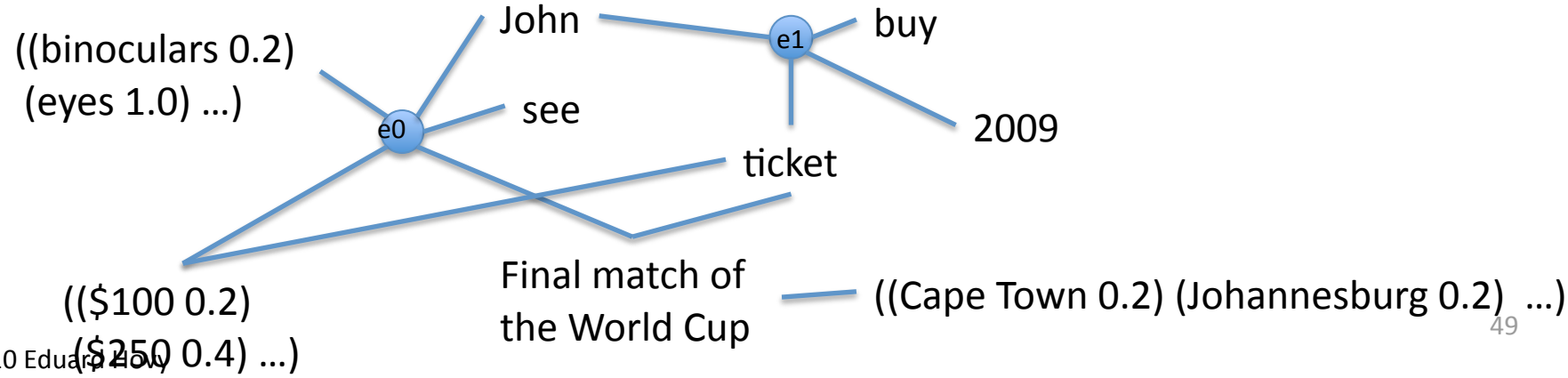
Topic:

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

Linking across sentences

John saw the final match of the World Cup. He had bought a ticket in 2009.

e0 = {(:type see) (:agent John) (:theme World Cup)
(:instr ((eyes 1.0) (binoculars 0.2) …))) …}
e1 = {(:type buy) (:agent John) (:patient ticket) (:time 2009)
(:amount (($100 0.2) ($250 0.4) …)) …}
ticket = {(:venue ((concert 0.1) (game 0.1) (opera 0.1) …)
(:price (($20 0.3) ($100 0.2) ($250 0.1) …)) …}



To side topic:
Probability, certainty, etc.

Computing 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

Summary: Core model

- One can perhaps define the semantics of statements in a way that combines the propositional and the distributional

- Questions, and research to be done:
  - What is the proper/best formulation?
  - What types/facets to use?
  - How to compute the score?
  - How to integrate scores and terms for synonymy?
  - How to compose the individual propositions? And what then happens with the scores?
  - How to manage dependencies?
  - …and many more

INTRO: TRADITIONAL SEMANTICS

DISTRIBUTIONAL SEMANTICS

1. TOPIC MODELS
2. WORD MODELS

A NEW MODEL OF SEMANTICS

CONCEPT FACETS

ONTOLOGIES

COMPOSITIONALITY 1: VECTORS

BUILDING A SEMANTIC LEXICON

RECENT EXPERIMENTS AT ISI

COMPOSITIONALITY 2: OPERATORS

CONCLUSION

CONCEPT FACETS OR DIMENSIONS

The problem of facets

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

Syntactic or semantic relations?

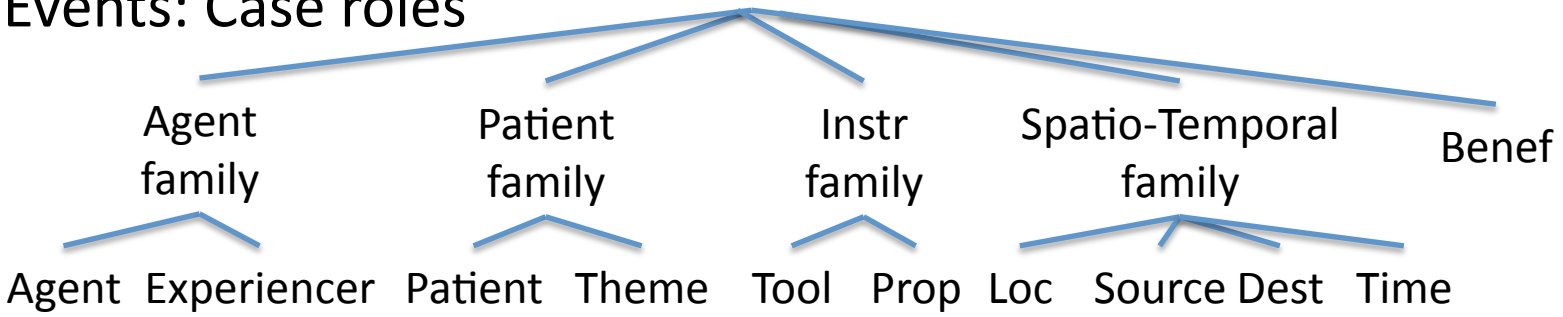
Parse tree gives merely syntactic relations

Nice, if you can get them:

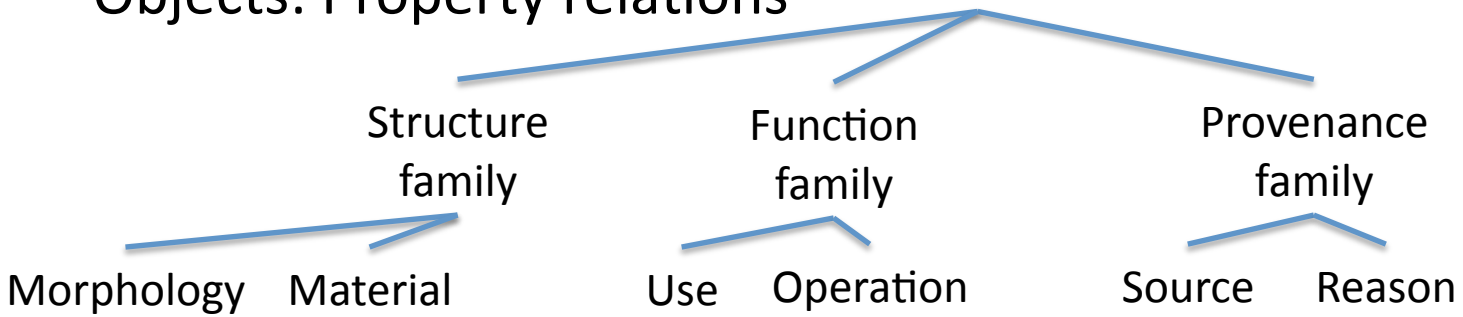
- Verb relations:
  - Case roles: from Framenet or PropBank
  - Prepositions: Prep sense disambiguation
- Noun relations:
  - Noun-noun compounds: NN relation classification
  - Noun-adjective modifiers: relation classification
- Multi-clause relations:
  - Verb-verb relation classification

Reminder of 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



Relating noun compounds

Tratz & Hovy ACL10

- Automated disambiguation of N-N relations (pairwise so far):

passenger complaints – Communicator of Communication
embassy spokeswoman – Employer of Employee
leukemia patient – Experience of Experiencer
food fight – Instrument of Use
cancer surgery – X + Mitigate/Oppose/Destroy
plastic bag – Substance/Material/Ingredient of Whole

morning flight – Time of X
navy destroyer – Owner of Owned
wine writer – Topic of Communication
aircraft fuel – Consumer of Consumed
highway accident – Location of X
maple leaf – Whole + Part/Member of

- Taxonomy of 42 relations created (+ OTHER)
  - Correlated with existing literature
- Validation
  - 17.5K NN pairs annotated by one person
  - MechTurk annotation underway since September; Kappa scores vary greatly with different annotators
- Automated classifier results
  - MaxEnt classifier: 64% agreement
  - General domain from NYT: ~58% agreement

NN rels and freqs

CATEGORY GROUP	CATEGORY	% of Total
TIME		
	Time+X	2.35
	X+Time	0.51
LOCATION/ PART_OF		
	Location+Located	5.07
	Whole+Part/Member Of	1.68
SUBSTANCE/PART/ MEMBER/ CONTAINEE		
	Substance/Material/Ingredient+Whole	2.30
	X+Collection/Configuration/Series	1.85
	X+Container/Location	1.40
TOPIC		
	Topic of Communication/Depiction	9.32
	Topic of Plans/Rules	3.96
	Topic of Observation/Examination	1.75
	Topic of Experience/Emotion	0.57
	Topic/Thing<->Attribute	3.38
	Topic/Thing+Attribute Value Characteristic Of	0.31
	Topic of Event/Process	1.10
	Topic of State	1.67
EQUATIVE/ SUBTYPE/ MEASURE		
	Coreferential	4.27
	Coreferential (Partial Attribute Transfer →)	0.70
	Measure/Dimension	4.37
OTHER		
	Fixed Pair / Opaque / Lexicalized	0.61
	Other	1.55

CATEGORY GROUP	CATEGORY	% of Total
CAUSE		
	Communicator+Communication	0.77
	Performer+Performed	2.08
	Cause/Creator/Provider	1.19
	Source/Cause of Money/Cost	1.26
PURPOSE (does/tries to/is used to)		
	Action/Activty+Perform(er)	13.34
	Created/Provided+Provide(r)	8.93
	Obtained/Achieved+Obtain(er)	1.53
	Managed+Manage(r)	4.78
	Domain+Position/Person of Prestige	0.91
	Propelled + Propel(lor)	0.14
	Moved + Transport/Transact/Transfer(er)	1.85
	Modified + Modify(er)	1.50
	Conserved + Conserve/Protect(er)	0.24
	Destination + Visit/Traverse(r)	0.10
	Opposed + Mitigate/Oppose/Destroy(er)	2.31
USE(R)		
	Owner+Owned	2.08
	Experiencer+Experience/Mental_Object	0.44
	Employer+Employee	2.74
	Consumer+Consumed	0.09
	User+Used	1.13
	…more…	

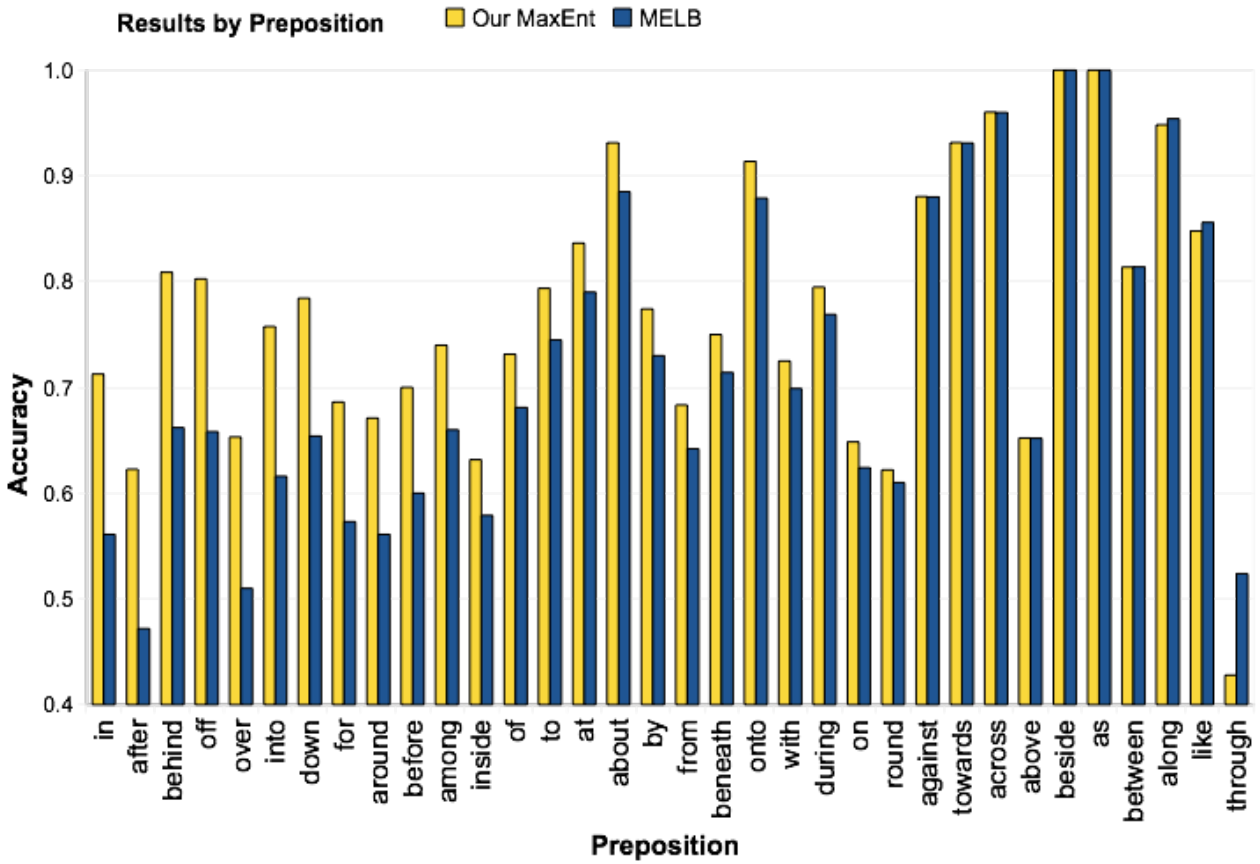
Preposition disambiguation

Hovy et al. COLING10

- Goal: Determine sense of each preposition
- Corpus
  - SemEval 2007 relation set and corpus
  - 34 preps; approx. 250 relations underlying preps
- Baseline
  - Implemented MaxEnt classifier using parse tree heads
  - Outperforms top scorer of SemEval 2007
- Current approach
  - Determine senses of prep and args simultaneously
  - Cast task as tagging problem: sequence <Arg1, Prep, Arg2>
  - Implemented lattice-based EM:
    - Constrain emission probabilities based on POS tag (possible senses for word class)
    - Constrain transition probabilities based on triple structure (Arg2 has to be noun, etc.)
  - Compare POS-based and parse-based arg identification:

POS-based arg error (max 230):	
48 arg1 errors	0.791
18 arg2 errors	0.922
Combined avg	0.856
Precision	0.726
Recall	0.746
Parse-based arg error (max 227):	
55 arg1 errors	0.758
25 arg2 errors	0.890
Combined avg	0.823
Precision	0.683
Recall	0.710

Baseline: Simple classifier



- Classifier uses parse tree attachments; trained on SemEval data
- Yellow: our system; black: top SemEval system

Summary: Concept facets

- The bag of words concept vectors are generally too weak to go far. But introducing structure requires relations, and these are hard to obtain for the general case.
- Questions, and research to do:
  - Sets of relations, and their definitions in the abstract: can they be defined in DS format as well?
  - Definitions of relations operationalized for computational analysis
  - Differences in performance between syntactic and semantic relations: can we get away with a reduced set?
  - Multilinguality: do various languages need specialized relations?
  - Non-language phenomena: can one handle pictorial info as well?

INTRO: TRADITIONAL SEMANTICS
DISTRIBUTIONAL SEMANTICS
1. TOPIC MODELS
2. WORD MODELS
A NEW MODEL OF SEMANTICS
CONCEPT FACETS
ONTOLOGIES
COMPOSITIONALITY 1: VECTORS
BUILDING A SEMANTIC LEXICON
RECENT EXPERIMENTS AT ISI
COMPOSITIONALITY 2: OPERATORS
CONCLUSION

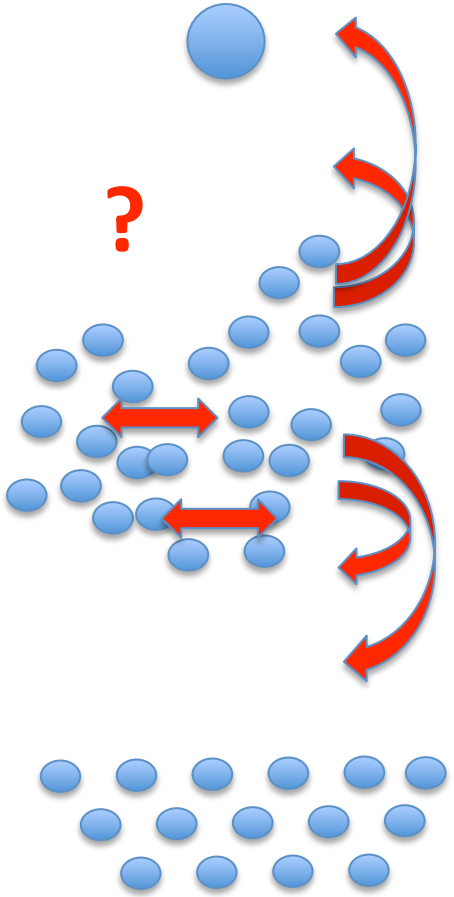
ONTOLOGIES

Organizing word meanings into ontologies

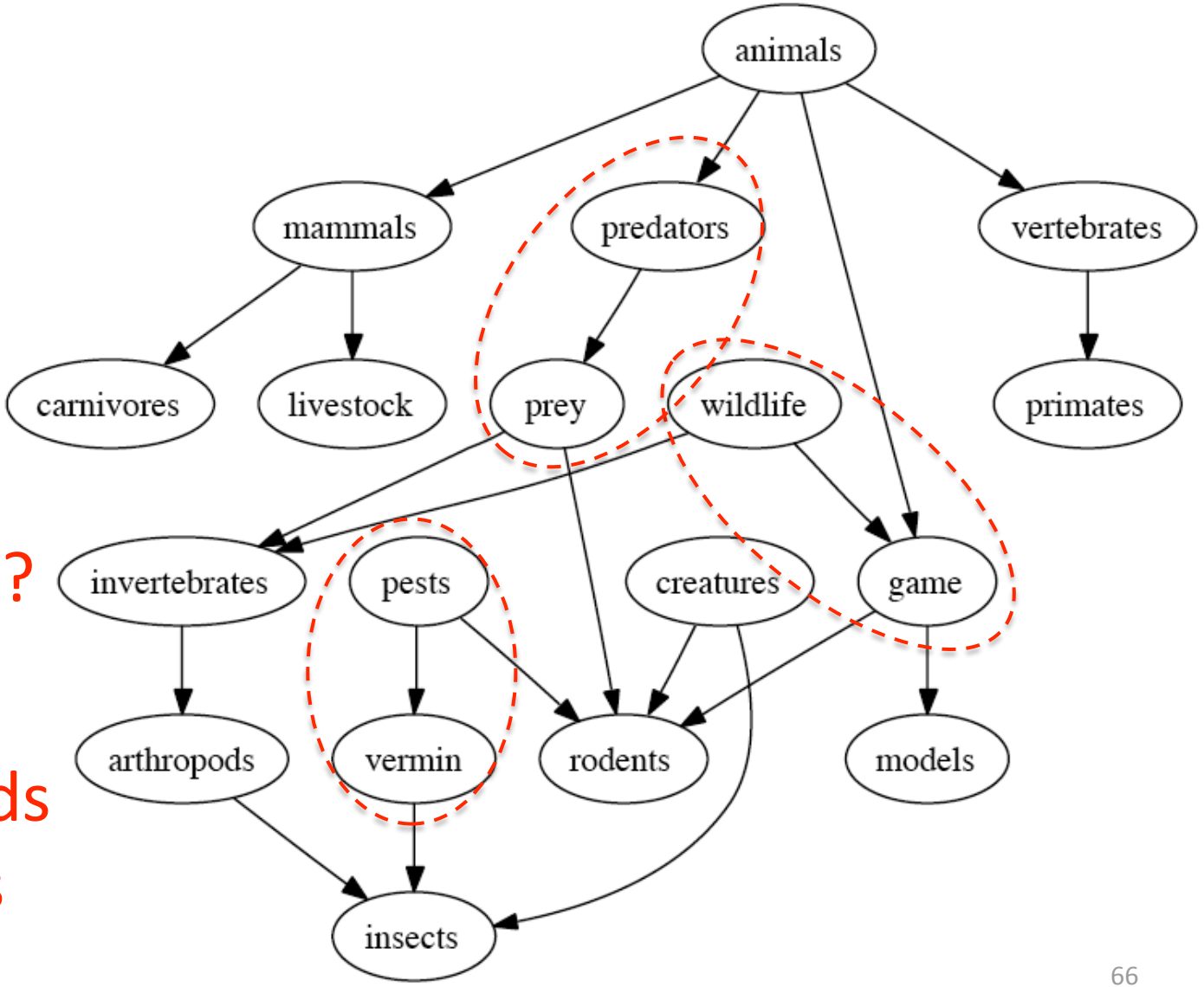
- Panel and Turney
- Current WN work
- Kozareva et al. work

Challenge: Taxonomizing concepts

- **Start**: animals
- **NP_0**: amphibians apes … felines fish fishes food fowl game game_animals grazers grazing_animals grazing_mammals herbivores herd_animals household_pests household_pets house_pets humans hunters insectivores insects invertebrates laboratory_animals … monogastrics non-ruminants pets pollinators poultry predators prey … vertebrates water_animals wetlands zoo_animals
- **NP_2**: … alligators ants bears bees camels cats cheetahs chickens crocodiles dachshunds dogs eagles lions llamas … peacocks rats snails snakes spaniels sparrows spiders tigers turkeys varmints wasps wolves worms …



Still…results are a bit of a mess



The problem?

Too many different kinds of categories

Solution: Group classes into small sets

- Goal: Create smaller sets, then taxonomize
- Need to find groups / families of classes

  [*predators prey*]

  [*carnivores herbivores omnivores*]

  [*pets wild_animals lab_animals …*]

  [*water_animals land_animals …*]

- Approach: Consult online dictionaries, encyclopedias:
  - Some classes are defined by behaviors (such as eating), some by body structure, some by function …
  - Try to define search patterns that capture salient aspects:

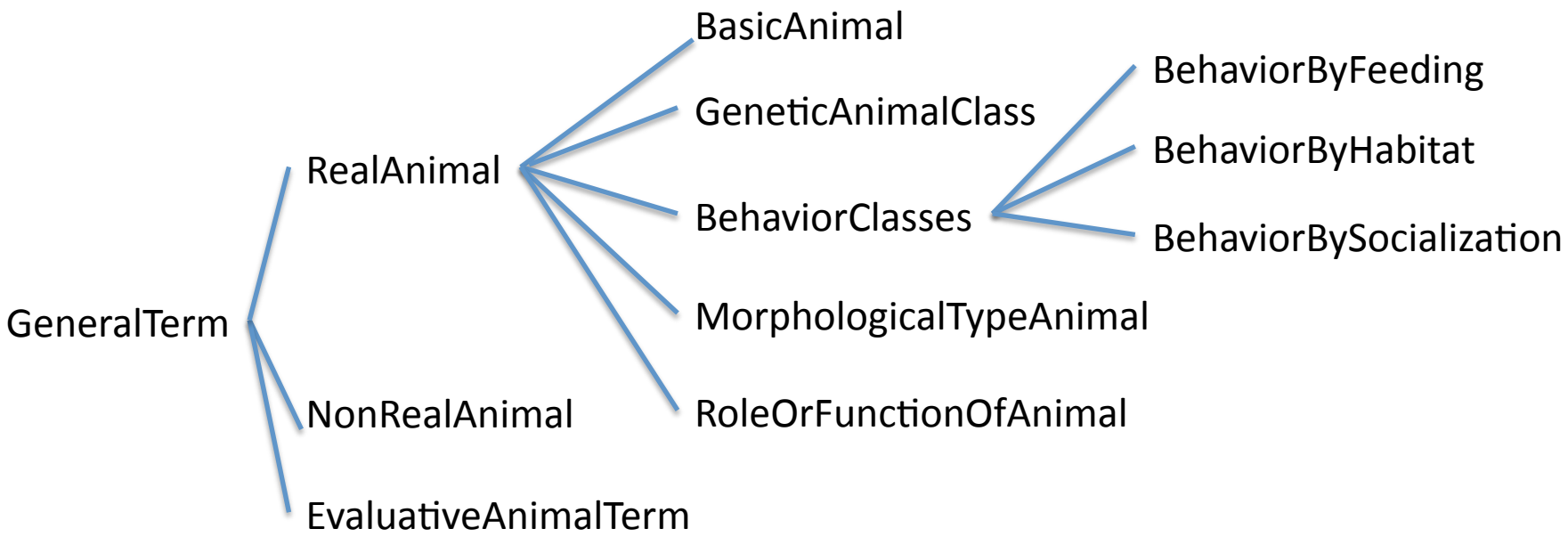
    "[*carnivores|herbivores|omnivores] are animals that **eat**…*"

    "[*water_animals|land_animals] are animals that **live**…*"

    "[*pets|lab_animals|zoo_animals] are animals that **?** *"

Evaluating sets (Kozareva et al. AAAI Spring Symp 09)

- First, created a small Upper Model manually:



- Then, had 4 independent annotators choose appropriate Upper Model class(es) for several hundred harvested classes
- Kappa agreement for some classes ok, for others not so good
  - Sometimes quite difficult to determine what an animal term means

1. BasicAnimal

The **basic individual** animal. Can be visualized mentally. Examples: Dog, Snake, Hummingbird.

2. GeneticAnimalClass

A **group** of basic animals, defined by **genetic similarity**. Cannot be visualized as a specific type. Examples: Reptile, Mammal. Note that sometimes a genetic class is also characterized by distinctive behavior, and so should be coded twice, as in Sea-mammal being both GeneticAnimalClass and BehavioralByHabitat. (Since genetic identity is so often expressed as body structure—it's a rare case that two genetically distant things look the same structurally—it will be easy to confuse this class with MorphologicalTypeAnimal. If the term refers to just a portion of the animal, it's probably a MorphologicalTypeAnimal. If you really see the meaning of the term as both genetic and structural, please code both.)

3. NonRealAnimal

Imaginary animals. Examples: Dragon, Unicorn. (Does not include 'normal' animals in literature or films.)

4. BehavioralByFeeding

A type of animal whose essential defining characteristic relates to a **feeding pattern** (either feeding itself, as for Predator or Grazer, or of another feeding on it, as for Prey). Cannot be visualized as an individual animal. Note that since a term like Hunter can refer to a human as well as an animal, it should not be classified as GeneralTerm.

5. BehavioralByHabitat

A type of animal whose essential defining characteristic relates to its habitual or otherwise noteworthy **spatial location**. Cannot be visualized as an individual animal. (When a basic type also is characterized by its spatial home, as in South African gazelle, treat it just as a type of gazelle, i.e., a BasicAnimal. But a class, like South African mammals, belongs here.) Examples: Saltwater mammal, Desert animal. And since a creature's structure is sometimes determined by its habitat, animals can appear as both; for example, South African ruminant is both a BehavioralByHabitat and a MorphologicalTypeAnimal.

6. BehavioralBySocializationIndividual

A type of animal whose essential defining characteristic relates to its patterns of **interaction with other animals**, of the same or a different kind. Excludes patterns of feeding. May be visualized as an individual animal. Examples: Herding animal, Lone wolf. (Note that most animals have some characteristic behavior pattern. So use this category only if the term explicitly focuses on behavior.)

7. BehavioralBySocializationGroup

A natural **group of basic** animals, defined by **interaction with other animals**. Cannot be visualized as an individual animal. Examples: Herd, Pack.

8. MorphologicalTypeAnimal

A type of animal whose essential defining characteristic relates to its internal or external **physical structure** or appearance. Cannot be visualized as an individual animal. (When a basic type also is characterized by its structure, as in Duck-billed platypus, treat it just as a type of platypus, i.e., a BasicAnimal. But a class, like Armored dinosaurs, belongs here.) Examples: Cloven-hoofed animal, Short-hair breed. And since a creature's structure is sometimes determined by its habitat, animals can appear as both; for example, South African ruminant is both a MorphologicalTypeAnimal and a BehavioralByHabitat. Finally, since genetic identity is so often expressed as structure—it's a rare case that two genetically distant things look the same structurally—it will be easy to confuse this class with MorphologicalTypeAnimal. If the term refers to just a portion of the animal, it's probably a MorphologicalTypeAnimal. But if you really see both meanings, please code both.

9. RoleOrFunctionOfAnimal

A type of animal whose essential defining characteristic relates to the **role or function** it plays with respect to others, typically humans. Cannot be visualized as an individual animal. Examples: Zoo animal, Pet, Parasite, Host.

G. GeneralTerm

A term that includes animals (or humans) but refers *also* to things that are neither animal nor human. Typically either a very general word such as Individual or Living being, or a general role or function such as Model or Catalyst. Note that in rare cases a term that refers mostly to animals also includes something else, such as the Venus Fly Trap plant, which is a carnivore. Please ignore such exceptional cases. But when a large proportion of the instances of a class are non-animal, then code it as GeneralTerm.

E. EvaluativeAnimalTerm

A term for an animal that carries an opinion judgment, such as "varmint". Sometimes a term has two senses, one of which is just the animal, and the other is a human plus a connotation. For example, "snake" or "weasel" is either the animal proper or a human who is sneaky; "lamb" the animal proper or a person who is gentle, etc. Since the term can potentially carry a judgment connotation, please code it here as well as where it belongs.

A. OtherAnimal

Almost certainly an animal or human, but none of the above applies, or: "I simply don't know enough about it".

Human category judgments

Animals

Code	An1	An2	An3	An4	Ex.M	Par.M	Kappa
BasicAnimal	29	24	13	4	2	12	0.51
BehavioralByFeeding	48	33	45	49	27	17	0.68
BehavioralByHabitat	85	58	56	54	36	36	0.66
BehavioralBySocializationGroup	1	2	6	7	0	3	0.47
BehavioralBySocializationIndividual	5	4	1	0	0	2	0.46
EvaluativeTerm	41	14	10	29	6	19	0.51
GarbageTerm	21	12	15	16	12	3	0.74
GeneralTerm	83	72	64	79	19	72	0.52
GeneticAnimalClass	95	113	81	73	42	65	0.61
MorphologicalTypeAnimal	29	33	42	39	13	26	0.58
NonRealAnimal	0	1	0	0	0	0	0.50
NotAnimal	81	97	82	85	53	40	0.68
OtherAnimal	34	41	20	6	1	24	0.47
RoleOrFunctionOfAnimal	89	74	76	47	28	56	0.58
Totals	**641**	**578**	**511**	**488**	**239**	**375**	**0.57**

People

Code	An1	An2	An3	An4	Ex.M	Par.M	Kappa
BasicPerson	5	6	1	3	1	3	0.55
FamilyRelation	7	6	7	6	5	2	0.86
GeneralTerm	38	12	21	12	4	18	0.50
GeneticPersonClass	1	2	1	0	0	1	0.44
ImaginaryPeople	14	16	5	2	1	10	0.47
NationOrTribe	2	3	3	2	2	1	0.78
NonTransientEventParticipant	29	63	41	32	16	33	0.57
NotPerson	31	31	28	38	24	9	0.80
OtherHuman	4	5	0	2	0	0	0.50
PersonState	23	1	25	1	0	8	0.47
RealPeople	1	7	1	0	0	1	0.50
ReligiousAffiliation	10	16	12	15	5	11	0.61
SocialRole	62	61	39	44	25	36	0.61
TransientEventParticipant	30	27	13	7	2	17	0.48
Totals	**257**	**256**	**197**	**164**	**85**	**150**	**0.58**

INTRO: TRADITIONAL SEMANTICS

DISTRIBUTIONAL SEMANTICS

1. TOPIC MODELS
2. WORD MODELS

A NEW MODEL OF SEMANTICS

COMPOSITIONALITY 1: VECTORS

BUILDING A SEMANTIC LEXICON

RECENT EXPERIMENTS AT ISI

COMPOSITIONALITY 2: OPERATORS

CONCLUSION

Combining concepts: A new ‘algebra’?

- If you define each concept individually, how can you compose concepts?
  - Composition of propositions using logical operators is a core part of traditional logic-based semantics: extensively studied
  - But how to combine distributed/statistically defined concepts?
  - And how to combine concepts in the new model?

concept ‘Red’ ✜ concept ‘Apple’ => ?

concept ‘John’ ✜ concept ‘attend’
✜ concept ‘soccer game’ => ?

- You need to take into account the relationship(s) between the concepts

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

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

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* : $v = w \oplus C$. The context *C* is just another tensor. But what is $\oplus$?

- 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

Combination method 1 (Mitchell and Lapata 2008)

- Vectors contain:
  - Words (not word senses)
  - No (explicit) relation: implicitly all *associated-with*
- General form: assume composed vector is in same space as individual vectors, $p = f(u,v)$. Make some sensible assumptions regarding vector components, then define:
  - Additive model: $p_i = u_i + v_i$ for each component *i*
  - Multiplicative model: $p_i = u_i \times v_i$
  - Allow components to affect one another: $p_i = \sum_j u_i \times v_{i-j}$ (Kinsch)
  - etc.
- Examples:
  - horse = {(animal 0) (stable 6) (gallop 2) (village10) (jockey 4)}
  - run = {(animal 1) (stable 8) (gallop 4) (village 4) (jockey 0)}
  - horse $\boxplus_+$ run = {(animal 1) (stable 14) (gallop 6) (village 14) (jockey 4)}
  - horse $\boxplus_\times$ run = {(animal 0) (stable 48) (gallop 8) (village 40) (jockey 0)}

Evaluating composition (Mitchell and Lapata 2008)

- M&L show subjects pairs of sentences, given a context; they must judge (semantic) similarity. How well do different formulas for wordsense vector combination predict their judgments?
  - fire = {(warm x) (glow x) (burn x) (red x) (match x) (friendly x) (light x) …}
  - face = {(pretty x) (beam x) (glow x) (happy x) (friendly x) (smile x) …}
  - glow = {(shine x) (red x) (warm x) (friendly x) (happy x) …}
  - burn = {(hot x) (red x) (energy x) (shine x) (glow x) (warm x) …}
  - beam = {(shine x) (light x) (dazzle x) (happy x) (smile x) …}

  - fire ✣ glow = {(warm x) (red x) (friendly x) …}
  - fire ✣ burn = {(warm x) (red x) (glow x) …}
  - fire ✣ beam = {(light x)}
  - face ✣ glow = {(warm x) (friendly x) (x) …}
  - face ✣ burn = {(glow x)}
  - face ✣ beam = {(shine x) (happy x) (smile x) …}

(The fire glowed)
The fire burned
The fire beamed

(The face glowed)
The face burned
The face beamed

Evaluation results

Noun	Reference	High	Low
The fire	glowed	burned	beamed
The face	glowed	beamed	burned
The child	strayed	roamed	digressed
The discussion	strayed	digressed	roamed
The sales	slumped	declined	slouched
The shoulders	slumped	slouched	declined

- Parameters:
  - Cosine similarity, 2000 words per vector, from window of ± 5 words
  - Baseline: similarity of verb and target word
  - In combined model, weights: verb = 0.95, noun = 0.0, comb = 0.05
  - Upper bound: human ratings
- Findings:
  - Humans: Spearman rank correlation = 0.4
  - **All models correlate significantly with human ratings**
  - **Best models: multiplic. and combined**

Model	High	Low	r
NonComp	0.27	0.26	0.08
Add	0.59	0.59	0.04
Weighted Add	0.35	0.34	0.09
Cross-word effect	0.47	0.45	0.09
Multiply	0.42	0.28	0.17
Combined	0.38	0.28	0.19
UpperBound	4.94	3.25	0.40

Combination method 2 (Erk and Padó 2008)

- Structure the vectors: add under relations
- Erk et al. 2006–: pointwise models of words in contexts
  - Like Schütze's second-order model
- Erk and Padó, 2008: introduce 'structured vector space':
  - Meaning of lemma ('concept for a') = (a, R, R^{-1})

    where a is the word vector, and R maps a's relations to its selectional preferences (other lemmas) in each position
  - Try two spaces: bag-of-words (BOW: all co-occurring words) and syntactic relations (SYN: built with parses by Minipar)
  - Variations: Selpref (basic), Selpref-cut (keep only filler words with freq over threshold), Selpref-pow (raise each component by nth power to strengthen common ones)

Combination rule

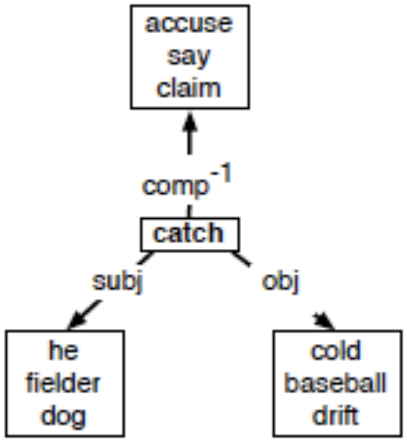
(Erk and Padó 2008)

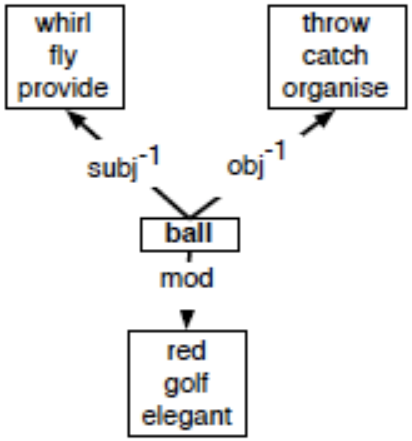
- For vectors a and b, capture each one in the context of the other (and don't include their 'irrelevant' parts in the joint context) — asymmetrical over the vectors
- For relation r that links concept a to concept b:

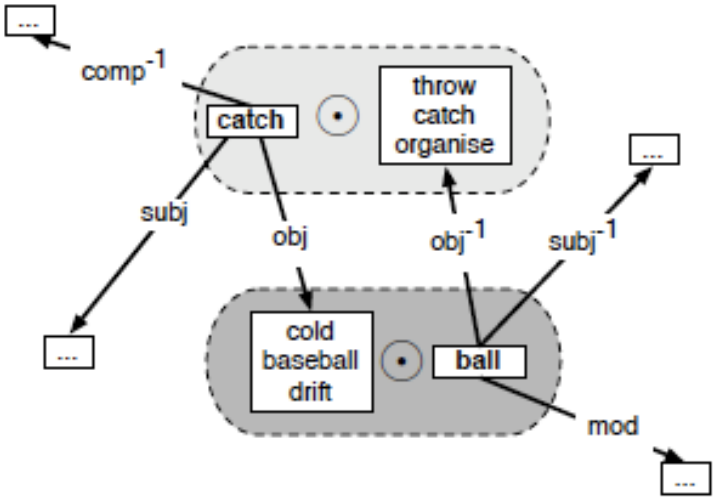
$$a' = (a \odot R_b^{-1}(r), R_a\text{-}\{r\}, R_a^{-1}) \text{ is meaning of a in context of b}$$

add/multiply a's elements with b's preferences

remove relation r from vector: it's now filled







Evaluation

- Expt 1: Same word-sense preference task as Mitchell & Lapata
  - Using same words and sentences, redo M&L and try own models
  - Results: **this method of combination works, but is not much better**
- Expt 2: Word substitution in paraphrases
  - **All methods provide more or less same performance**

Model	High	Low	ρ
Bag of words space (unstructured: no relations)			
Target only	0.32	0.32	0.00
Sel. Pref. only	0.46	0.40	0.06
Mitchell & Lapata	0.25	0.15	0.20
Selpref	0.32	0.26	0.12
Selpref-cut	0.31	0.24	0.11
Selpref-pow	0.11	0.03	**0.27**
Upper Bound	–	–	0.40
Syn space (structured with syntactic relations)			
Target only	0.20	0.20	0.08
Sel. Pref. only	0.27	0.21	0.16
Mitchell & Lapata	0.13	0.06	**0.24**
Selpref	0.22	0.16	0.13
Selpref-cut	0.20	0.13	0.13
Selpref-pow	0.08	0.04	0.22
Upper Bound	–	–	0.40

Combination method 3: A new way

- Approach: Given some concepts to combine,
  - assemble a single frame-like proposition using relations to link them,
  - collect from corpus the remaining relevant vector elements

fork = {(:isa cutlery)
(:material ((metal 0.7) (:plastic 0.3) (:plastic 0.2) …))
(:instr-of ((eat 0.8) (cook 0.2)) (pin-down 0.00004) …)) …}

eat = {(:isa ingest-action)
(:agent ((person 0.4) (animal 0.4) (John 0.0001) …))
(:patient ((food 0.8) (porridge 0.00004) (apple 0.001) …))
(:instr ((spoon 0.2) (fork 0.19) …)) …}

Get new numbers from corpus!

eat ✣$_{instr}$ **fork** = {(:isa ingest-action) (:instr fork)
(:agent ((person 0.9) (animal 0.00001) (:John 0.004) …))
(:patient ((food 0.9) (pasta 0.4) (veg 0.24) …))
(:loc ((dining-room 0.4) (restaurant 0.4) …)) …}

Increasing the specificity of content

John saw the game

{e0 (:**type** see) (:**agent** John)
(:**theme** ((:type (football 0.1) (soccer 0.1) (tennis 0.1)) …))
(:**instr** ((eyes 1.0) (binoculars 0.3) …))
(:**loc** ((Brazil 0.15) (UK 0.2) (South Africa 0.1) …))
…}

John saw the World Cup

{e0 (:**type** see) (:**agent** John)
(:**theme** ((:**instance** soccer) (:**name** "World Cup")…))
(:**instr** ((eyes 1.0) (binoculars 0.2) …))
(:**loc** ((Brazil 0.15) (UK 0.2) (South Africa 0.1) …))
…}

John saw the 2010 World Cup in South Africa

{e0 (:**type** see) (:**agent** John)
(:**theme** World-Cup-2010)
(:**instr** ((eyes 1.0) (binoculars 0.2) …))
(:**loc** South-Africa)
…}

Representations can be made arbitrarily specific; each time, remainder is learned from corpus

Why this method?

- Does not start with predefined vectors for each word and then 'bend' them together; rather recomputes distributions for the composed concept itself
- Basic tenet: The 'composed' concepts are valid concepts in their own right, and define their unique associational environments
  - Implies fundamentally different view of compositionality: not accurate to simply combine concepts learned independently of one another
  - Respects concept 'boundaries': even a relatively 'stable' concept can assume subtle differences in the context of other concepts
  - The Erk and Padó 2008 Structured Vector Space model is a way to approximate this

Problems with this method

- **Data sparsity problem**: The longer the composed proposition, the smaller the number of corpus exemplars to build tensor vectors from
  - See later in the talk
- **'De-compositionality' problem**: What is the relationship between the pieces of the composed proposition and the individual separately defined vectors?
  - Compare this method to Erk and Padó, for example

Summary: Compositionality

- It is possible to define various ways of combining the contents of word/concept vectors. The 'best' one is unclear, and how to measure the results is also unclear

- Questions, and research to do:
  - Additional methods to compose individual concept tensors/ vectors
  - The data sparsity problem
  - Interactions between tensors and logical operators
  - Is an algebra over the composition methods possible? Could one derive theorems and prove statements about complex concepts in the abstract, without the actual vectors' values? Wow!!

LINK WITH INFORMATION THEORY

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 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 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 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
  - I = p . log p

- Info Theory shortcoming: 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 DS, the reader's knowledge can be explicitly encoded
  - Represented in individual lexical entries' score contents

INTRO: TRADITIONAL SEMANTICS

DISTRIBUTIONAL SEMANTICS

1. TOPIC MODELS
2. WORD MODELS

A NEW MODEL OF SEMANTICS

COMPOSITIONALITY 1: VECTORS

BUILDING A SEMANTIC LEXICON

RECENT EXPERIMENTS AT ISI

COMPOSITIONALITY 2: OPERATORS

CONCLUSION

To side topic:
Construction

Construction procedure

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 Propositions)
6. Save every triple/prop in a large Store:

   [rel head mod 1 doc-id sent-id]
7. When done, add together all the 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$_1$)…)]

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
```

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)

Summary: Building Prop Stores

- It is possible to build large Proposition Stores quite easily. These contain much information needed for a DS lexicon

- Questions and research to do:
  - What is the optimal format and content of a Prop Store?
  - What is the best method for rapid construction?
  - What is the best computational architecture for access, updating, etc.?
  - How can one handle sparse data — what smoothing can one do for unseen examples?

INTRO: TRADITIONAL SEMANTICS

DISTRIBUTIONAL SEMANTICS

1. TOPIC MODELS
2. WORD MODELS

A NEW MODEL OF SEMANTICS

COMPOSITIONALITY 1: VECTORS

BUILDING A SEMANTIC LEXICON

RECENT EXPERIMENTS AT ISI

COMPOSITIONALITY 2: OPERATORS

CONCLUSION

Context: Machine Reading

- Challenge: Build systems that can extend their 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; …
- 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)

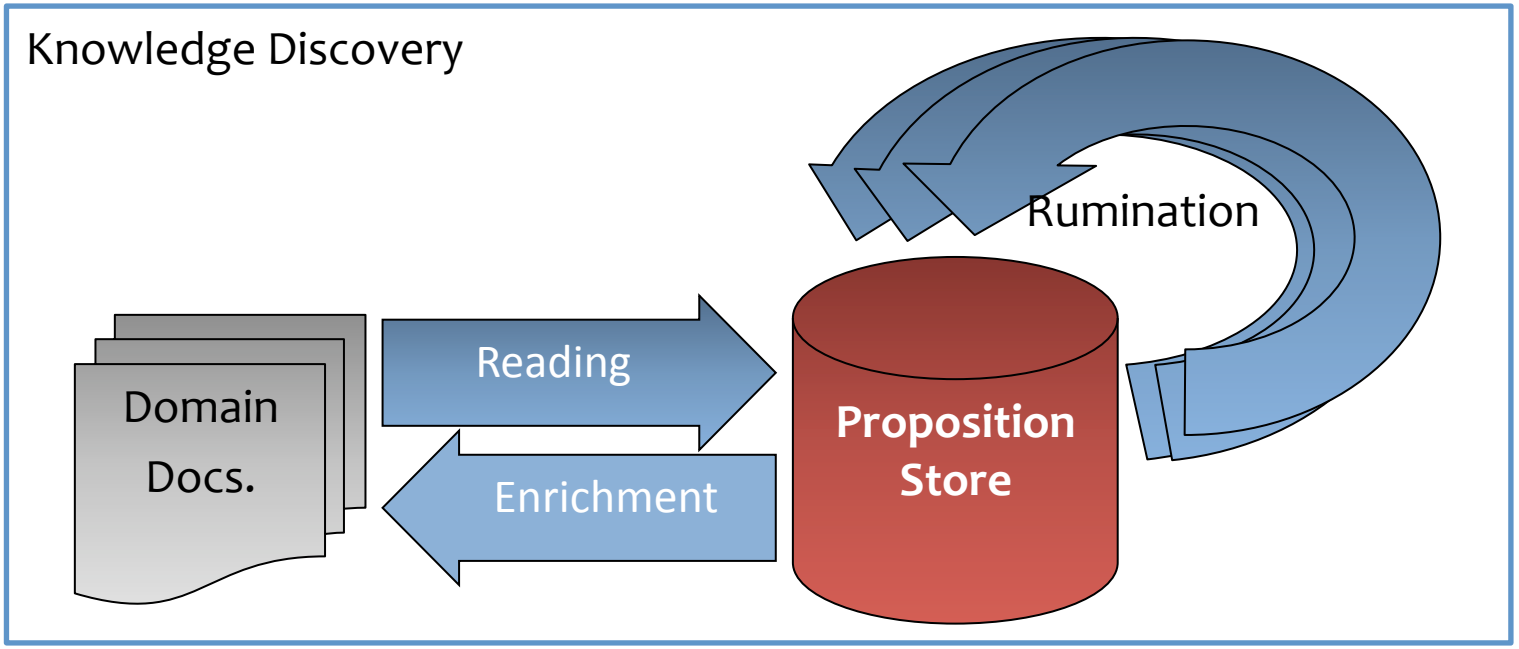
Our work in RACR

- 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

ISI's knowledge support service

- We are building a general 'knowledge support service'
- **Proposition Store**: A large general world model, and/or specialized domain models:
  - Lexical and semantic 'connotation knowledge' for content words
  - Model can be tailored to each new domain for rapid (though averaged) semantic predictions
- 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
- Methods:
  - Providing semantic preferences for parsing, interpretation, inference
  - 'Funneling' expressive variations into preferred terms
  - Reranking inference preferences to improve performance speed

The MR knowledge enrichment cycle



Cycle:

1. Read text from collection
2. Ruminate in BKB
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'))]).

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'

Using the knowledge service

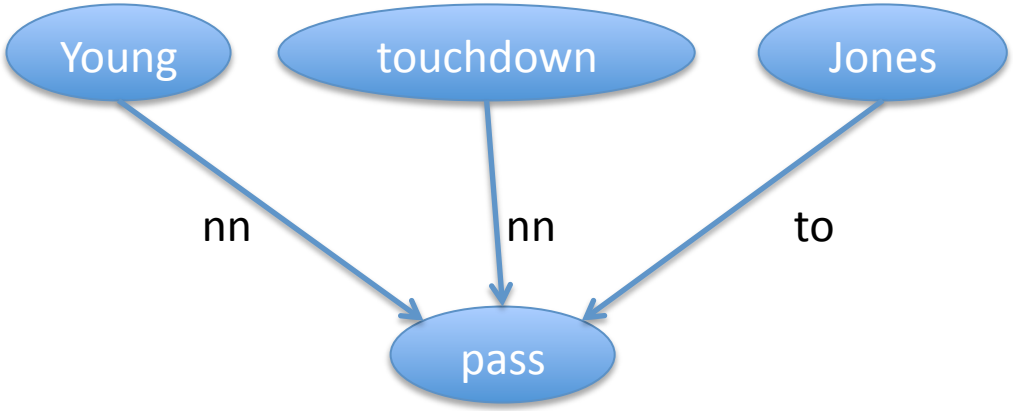
Example: *San Francisco's Eric Davis intercepted a Steve Walsh pass on the next series to set up a seven-yard Young touchdown pass to Brent Jones.*

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

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

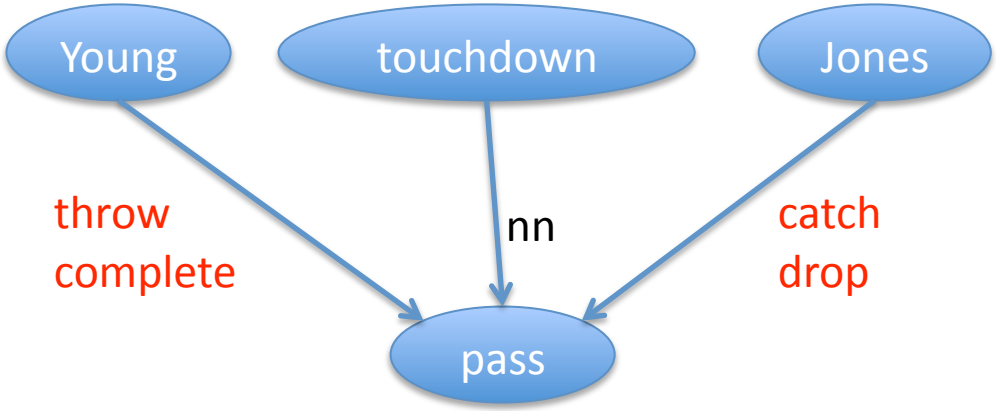
?> NVN:end:X:pass

X=catch

X=drop

Enrichment 2

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



touchdown pass

?> NVN touchdown:X:pass

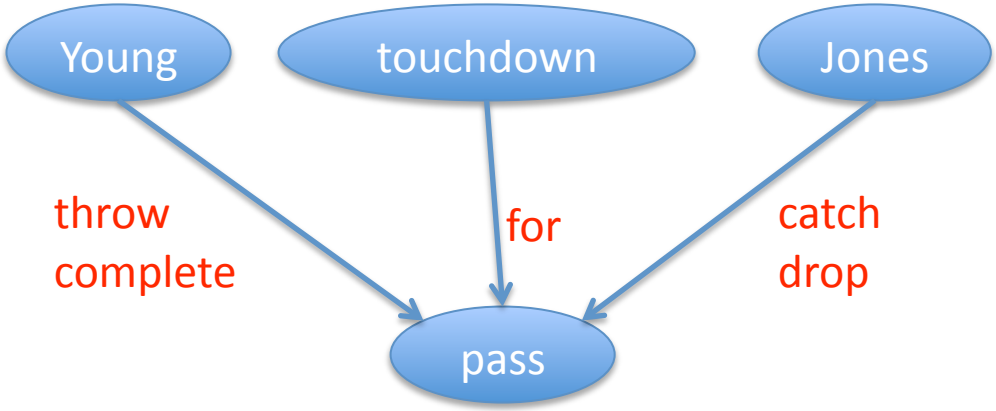
False

?> NPN pass:X:touchdown

X=for

Enrichment 3

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



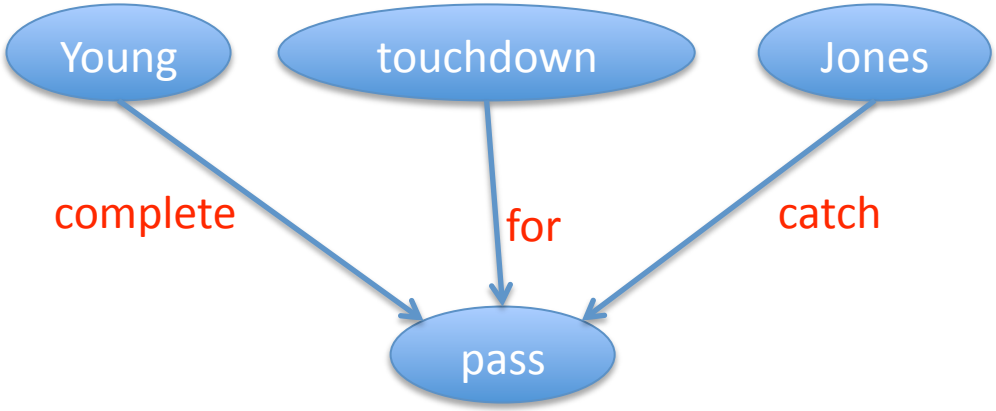
?> NVNPN NAME:X:pass:for:touchdown

X=complete

X=catch

Enrichment 4

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



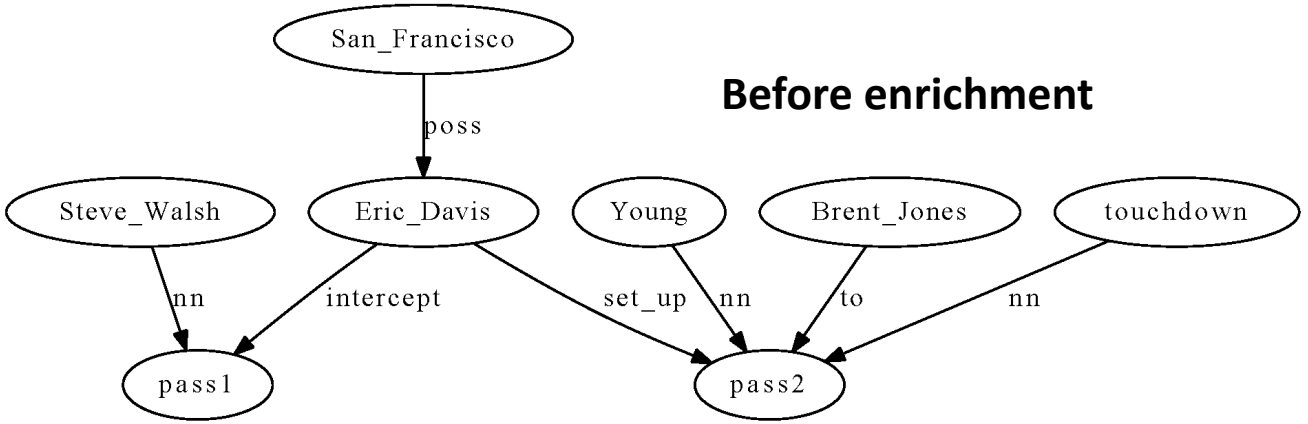
⇒ Young complete pass for touchdown

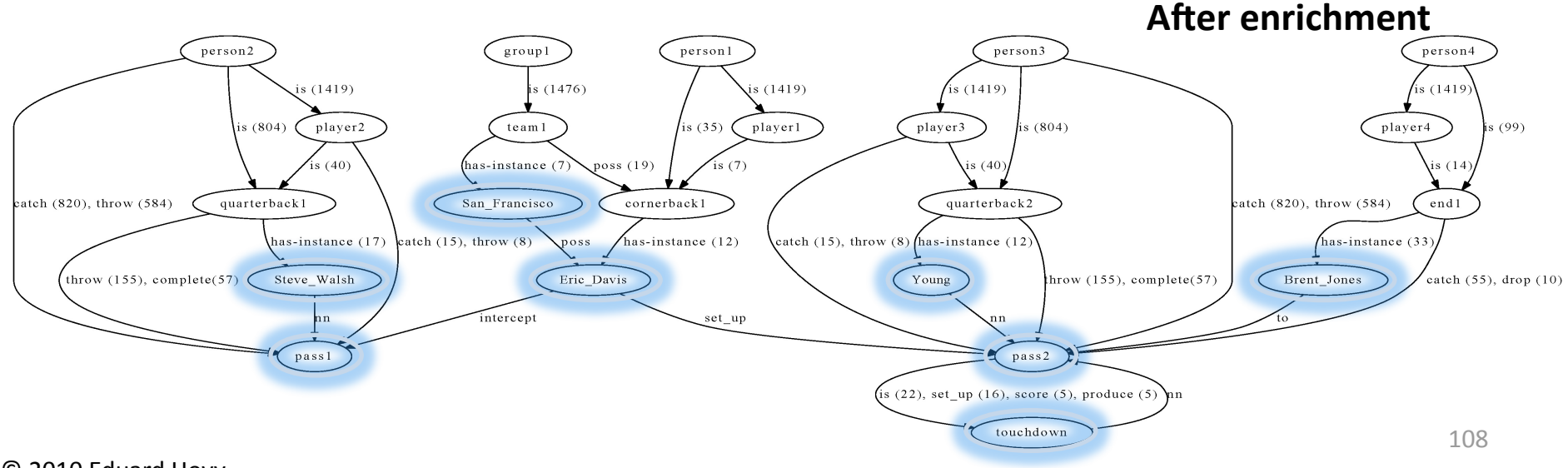
⇒ Jones catch pass for touchdown

To side topic:
Text Enrichment

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.





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].

INTRO: TRADITIONAL SEMANTICS

DISTRIBUTIONAL SEMANTICS

1. TOPIC MODELS
2. WORD MODELS

A NEW MODEL OF SEMANTICS

COMPOSITIONALITY 1: VECTORS

BUILDING A SEMANTIC LEXICON

RECENT EXPERIMENTS AT ISI

COMPOSITIONALITY 2: OPERATORS

CONCLUSION

Composition of propositions and operators

- Composition of propositions using logical operators is a core part of traditional logic-based semantics: extensively studied

Definition (Dorr et al., Modality Study 2009)

- Def.: "Modality (derivative of "mood") is, roughly speaking, an attitude on the part of the speaker toward an *action* (such as "go to work", "move to Mexico", "put in jail",) or *state* ("be at home", "be in Mexico", or "be jailed"). Modality is expressed with bound morphemes or free standing words or phrases. Modality interacts in complex ways with other grammatical units such as tense and negation."
- Terminology:
  - **Trigger (M):** a word or words that expresses modality
  - **Target (R):** an annotatable unit—an action or state over which the modality is expressed
  - **Holder (H):** holder of modality

Negation/modalities in this semantics

- Apply operation to appropriate aspects of concept/ proposition:
  - Negate/modify just the value(s) in question
  - Adjust remaining values' scores as appropriate

Soccer on the moon in new semantics

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**:

Old neg v1:
(attend e0 x0 x1 x2) (John x0) (WC x1) (moon x2) (not e0)
(e0 (:type attend) (:agent John) (:theme WC) (:loc moon) (:polarity neg))

Old neg v2:
(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))

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

Same, in new semantics:

(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 in DS: Mozart again

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
  - We somehow need to determine *which* [aspects'] scores change, and which do not, for each operator

To side topic:
Modality in this model

Summary: Compositionality 2

- Since the formalism remains basically the same as the traditional logic-based or frame-based formalisms, the traditional methods of compositionality using logical operators and relations carries over

- Questions and research to do:
  - Interactions between logical operators and content vectors
  - Representational treatment of various propositional phenomena

INTRO: TRADITIONAL SEMANTICS

DISTRIBUTIONAL SEMANTICS

1. TOPIC MODELS
2. WORD MODELS

A NEW MODEL OF SEMANTICS

COMPOSITIONALITY 1: VECTORS

BUILDING A SEMANTIC LEXICON

RECENT EXPERIMENTS AT ISI

COMPOSITIONALITY 2: OPERATORS

CONCLUSION

Using DS for NLP

- Preference semantics
  - Wilks 75 etc.
- WSD
  - Agirre et al.
  - Everyone
- Learning paraphrases
  - DIRT (Pantel and Lin 02)
  - Later
- Parsing and PP attachment
  - Klein et al. ACL10

Why does IR work?

- Document is represented in a vector space as a vector of words: 'document signature'
- In DS terms:

  DS(doc) = {(r w_i s_i)} for i different open-class words

  where r = 'word-inside-doc' except for stop words

  w_i = word and s_i = word's count

- This is simply and directly a use of DS
- Two docs are similar when they have a similar (normalized) DS document signature

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.

Where should we be going?

- Handle many more of the individual semantic phenomena
- Create the intensional terminology 'lexicon':
  - Features (event features; object features; etc.)
    - Framenet, PropBank, etc.
  - Ontology of feature combinations
    - WordNet, CYC, etc.
  - Instance bases of knowledge (all instantial facts)
    - YAGO, Text mining, etc.
- Integrate with the extensional Distributional Semantics we are now building
- Start building multisentence semantic representations
  - Not just Discourse Structure, but dense semantic networks
  - Example: DARPA's Machine Reading Project

THANK YOU

Readings

- Formal models
  - Preference Semantics: Wilks, 1975
  - Turney: several papers since 2005
  - Novacek, PhD 2010
- 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
  - Navigli, PhD 2008
  - Turney, several papers
  - Erk, ACL 2010 and earlier
- 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
- 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
- Organizing vectors into hierarchies and finding default values
  - Turney and Pantel, 2010
  - O'Sean…, ACL 2010
  - Tan and Hovy, in prep
- 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