A New Semantics: Merging Propositional and Distributional Information

Eduard Hovy

Information Sciences Institute
University of Southern California
hovy@isi.edu
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 are Propositions about Symbols
• 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 Model what the symbols ‘mean’
• 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
Intensional approach to concept definition

• 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
Extensio**nal** 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:
    \[
    \text{Concept } x = \{ \text{all instances of } x \text{ in the world} \}
    \]
- 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

© 2010 Eduard Hovy
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

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

---

**Table:**

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

\[ \text{tf.idf} : w_{jk} = tf_{jk} \times idf_j \]

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

(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
- \( \text{tf.idf} \) is the best for IR, among 287 methods (Salton & Buckley, 1988)
- \( m_{jk} = (\sum_j tf_{jk} \sum_k tf_{jk}) / \sum_j tf_{jk} \) : mean count for term \( j \) in text \( k \)

likelihhood ratio \( \lambda \) : 2\log \lambda = 2N \cdot I(R ; T) \]

(Lin & Hovy, 2000)

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

- \( N \) = total number terms in corpus
- \( I \) = mutual information between text relevance \( R \) and given term \( T \)
  \( = H(R) - H(R | T) \) for \( H(R) \) = entropy of terms over relevant texts \( R \)
  and \( H(R | T) \) = 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, $\cos \theta = \frac{TS_k \cdot DS_i}{\|TS_k\|\|DS_i\|}$

• Test 1 (Hovy & Lin, 1997, 1999)
  – Training: 10 topics; ~3,000 texts (TREC)
  – Contrast set (background): ~3,000 texts
  – Conclusion: tf.idf and $\chi^2$ signatures work ok but depend on signature length

• Test 2 (Lin & Hovy, 2000):
  – 4 topics; 6,194 texts; uni/bi/trigram signals
  – Evaluated using SUMMARIST: $\lambda > tf.idf$
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:
  – Stanford Topic Modeling Toolbox: [http://nlp.stanford.edu/software/tmt/tmt-0.2/](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
  - \( \Sigma : \) diagonal matrix with exactly \( \text{rank}(A) \) 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 $z$:

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

$$p(w \mid d) = \sum_z p(z \mid d) \cdot p(w \mid z)$$

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

• Assumptions:
  – You have lots of docs, but only a small number of topics
  – Each doc is a specific mixture of topics (with weight $p(z \mid d)$)
  – 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

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

using EM algorithm

© 2010 Eduard Hovy
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:
  – \( \alpha \): parameter of the uniform Dirichlet prior topic distribution per document
  – \( \theta_i \): topic distribution for document \( i \)
  – \( \beta \): 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
  - $\varphi: K \times V$ ($V$ 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:
  – 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:

\[
\text{PMI}(w_1, w_2) = \log \left[ \frac{p(w_1, w_2)}{p(w_1) \cdot p(w_2)} \right]
\]

  – Typically used for wordsense disambiguation (each sense has its characteristic vector), sense clustering, etc.

• Word mention models (e.g., (Erk 2008)):
  – Compute the vector just using the words from the single sentence instance
Some early work

• Work on word-level context vectors
  – Generally, each vector represents the ‘average meaning’ of a lemma
  – 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 paraphrase leaning methods
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

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, …

\[
mi_{ef} = \log \frac{\frac{c_{ef}}{N}}{\sum_{i=1}^{g} c_{if} \frac{1}{N} \times \sum_{j=1}^{m} c_{eg}}
\]
Why is *apple* is similar to *pear* (Pantel 02)

Blue: apple only

Green: pear only

Red: shared
Why *apple* is not similar to *toothbrush*

From Pantel (2002)
In word vectors, senses are mixed up from (Pantel 2002)
Need something like 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!

To side topic: Sense differentiation

© 2010 Eduard Hovy
...but which senses?

• Sense delimitation is not accurate
• OntoNotes experience (Hovy et al. 05; Pradhan et al. 07):
  – Senses created following graduated refinement procedure
  – Validated by inter-annotator agreement in sentences

• Word vectors allow near-continuous variability and shades of meaning, but differ in different ‘directions’
  – drive-car: (:patient ((car 0.4) (bus 0.2) ... (PhysObj 0.05) ...))
    (:direction (...)) (:source (...)) (:speed (...))
  – drive-legs: (:patient ((legs 0.5) (fists 0.2) ... (PhysObj 0.1) ... ))
    (:direction (...)) (:force (...)) (:speed (...))
  – 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.):

<table>
<thead>
<tr>
<th>Sense</th>
<th>Word Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waiter1</td>
<td>restaurant, waitress, dinner, bartender, dessert, dishwasher, aperitif, brasserie, . . .</td>
</tr>
<tr>
<td>Waiter2</td>
<td>hospital, station, airport, boyfriend, girlfriend, sentimentalist, adjudicator, . . .</td>
</tr>
</tbody>
</table>

• ‘Explanation’ generation (Vyas and Pantel, COLING 08):

<table>
<thead>
<tr>
<th>Word set</th>
<th>‘Explanation’ for set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Palestinian-Israeli, India-Pakistan</td>
<td>talks(NN), conflict(NN), dialogue(NN), relation(NN), peace(NN)</td>
</tr>
</tbody>
</table>
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 \( r_i \in \{\text{Relations}\} = R \), e.g., :subj, :agent, :color-of
\( w_i \in \{\text{Words}\} = \text{vocabulary} \), e.g., happy, run, apple
\( s_i \in [0,1] \)

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 = \{ (:\text{type} \text{Jack Russell} \text{0.2)} (:\text{type} \text{Retriever} \text{0.4)}
\quad (:\text{colour} \text{brown} \text{0.4)} (:\text{colour} \text{black} \text{0.3)}
\quad (:\text{agent-of} \text{eat} \text{0.4)} (:\text{patient-of} \text{chase} \text{0.3}) \ldots \} \\

- A **Topic Signature / Topic Model** is a very simple way of defining a topic: there’s only one \text{r}_i, namely ‘\textit{associated with}’

  Dog = \{ (\text{brown} \text{0.9)} (\text{bark} \text{0.6)} (\text{“Lassie”} \text{0.2)}
  \quad (\text{run} \text{0.6)} (\text{white} \text{0.4)} (\text{chase} \text{0.1}) \ldots \} \\

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

  domain = \{ (\text{“brown dog” 0.0000016)}
  \quad (\text{“the brown” 0.0000032}) \ldots \}
A useful notation variant

- It’s convenient to group together all tuples with the same $r_i$:  
  \[
  \text{Dog} = \{ (:\text{type} \ ((\text{Retriever} \ 0.4) \ (\text{Terrier} \ 0.4) \ (\text{Jack Russell} \ 0.35) \ ...)) \\
  (:\text{color} \ ((\text{brown} \ 0.9) \ (\text{black} \ 0.4) \ (\text{patched} \ 0.3) \ (\text{white} \ 0.2) \ ...)) \\
  (:\text{name} \ ((\text{“Spot”} \ 0.3) \ (\text{“Lassie”} \ 0.2) \ ...)) \\
  (:\text{agent-of} \ ((\text{eat} \ 0.5) \ (\text{run} \ 0.4) \ (\text{bark} \ 0.4) \ (\text{pant} \ 0.3) \ ...)) \\
  (:\text{patient-of} \ ((\text{feed} \ 0.5) \ (\text{walk} \ 0.4) \ (\text{love} \ 0.4) \ ...)) \\
  ... \}
  \]
Color
- grey
- white
- patched

Type
- Dachshund
- Alsatian
- Jack Russell
- Retriever
- Terrier
- Alsatian

Name
- Lucy
- Rover
- Asta
- Lassie
- Spot
- walk
- love
- feed
- hear
- chase

Patient-of

Agent-of

Slightly more formally

• The semantic knowledge base (‘lexicon’) consists of:
  – \( R \) : the list of all relations
  – \( C \) : the list of all concepts \( C_i \)
  – \( S \) : a real number in [0,1]
  – \( D \) : the domain (a collection of texts)
  – \( M \) : the matrix \( R \times C \) containing everything zero
  – \( KB \) : the knowledge base: a set of all tensors \( T_{C_i} \) for all \( C_i \)

• Each generic concept (word) \( C_i \) is a tensor as follows:
  – \( ID \) : the identifier (‘name’) of \( C_i \) (a string)
  – \( T_{C_i} \) : the part of \( M \) that contains nonzero values of \( S \), computed as appropriate from \( D \) (a tensor)
  – In practice, we store also the source info for the values of \( T_{C_i} \)

• 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, 1972**

<table>
<thead>
<tr>
<th></th>
<th>Lassie</th>
<th>W.A. Mozart</th>
<th>…</th>
<th>music</th>
<th>dog</th>
<th>…</th>
<th>WW I</th>
<th>Lunch today</th>
<th>…</th>
<th>lunch</th>
<th>composing</th>
<th>…</th>
</tr>
</thead>
<tbody>
<tr>
<td>:type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>:name</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:age</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:shape</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:color</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:agent</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:theme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:loc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:cause</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>John sees soccer</th>
<th>John</th>
<th>S. Afr.</th>
<th>Switz</th>
<th>...</th>
<th>human</th>
<th>eye</th>
<th>...</th>
<th>Lunch today</th>
<th>...</th>
<th>seeing</th>
<th>soccer game</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>:type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:age</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:nation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:loc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:agent-of</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:agent</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:theme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.001</td>
</tr>
<tr>
<td>:instr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td>:loc</td>
<td>0.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2010 Eduard Hovy
A Swiss John seeing soccer

<table>
<thead>
<tr>
<th>John2 sees soccer</th>
<th>John</th>
<th>S. Afr.</th>
<th>Switz</th>
<th>...</th>
<th>human</th>
<th>eye</th>
<th>...</th>
<th>Lunch today</th>
<th>...</th>
<th>seeing</th>
<th>soccer game</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>:type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:name</td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:age</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:nation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:loc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>:agent-of</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.7</td>
<td>0.99</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:agent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:theme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:instr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:loc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2010 Eduard Hovy
When Mozart was young he lived in Salzburg; when he was an adult he lived in Vienna.

<table>
<thead>
<tr>
<th>W.A. Mozart</th>
<th>Salzburg</th>
<th>Vienna</th>
<th>...</th>
<th>music</th>
<th>youth</th>
<th>adult</th>
<th>...</th>
<th>living</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>:type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:age</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:shape</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:color</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:agent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:theme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:loc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:cause</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2010 Eduard Hovy
Which relations?

- Minimum: relation *associated-with* (in topic signature)
- Better: syntactic relations (*subj, dobj, iobj, preps...*)
- Even better: semantic relations
  - **Events: Case roles**
    - Agent family
      - Agent
      - Experiencer
    - Patient family
      - Patient
      - Theme
    - Spatio-Temporal family
      - Instr
      - Loc
      - Source
      - Dest
      - Time
      - Benef
  - **Objects: Property relations**
    - Structure family
      - Morphology
      - Material
    - Function family
      - Use
      - Operation
    - Provenance family
      - Source
      - Reason

© 2010 Eduard Hovy
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) ...)) ...\}

© 2010 Eduard Hovy
John saw the final match of the World Cup. He had bought a ticket in 2009.

\[
e_0 = \{(:type \text{ see}) (:agent \text{ John}) (:theme \text{ World Cup})
    (:instr ((\text{eyes } 1.0) (\text{binoculars } 0.2) ...))) ...\}\n\]
\[
e_1 = \{(:type \text{ buy}) (:agent \text{ John}) (:patient \text{ ticket}) (:time \text{ 2009})
    (:amount ((\$100 0.2) (\$250 0.4) ...)) ...\}\n\]
\[
\text{ticket} = \{(:\text{venue} ((\text{concert } 0.1) (\text{game } 0.1) (\text{opera } 0.1) ...)
    (:\text{price} ((\$20 0.3) (\$100 0.2) (\$250 0.1) ...)) ...\}\n\]
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 positive value in like cell(s)

• So far, no-one has provided a proper account

© 2010 Eduard Hovy
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

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

• 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: $p_i = \sum_j u_i \times v_{i,j}$ (Kisch)
  - etc.

- Examples:
  - horse = {animal 0) (stable 6) (gallop 2) (village10) (jockey 4)}
  - run = {animal 1) (stable 8) (gallop 4) (village 4) (jockey 0)}
  - horse $\bowtie$ run = {animal 1) (stable 14) (gallop 6) (village 14) (jockey 4)}
  - horse $\boxtimes$ 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) \ldots\}
• face = \{(pretty x) (beam x) (glow x) (happy x) (friendly x) (smile x) \ldots\}
• glow = \{(shine x) (red x) (warm x) (friendly x) (happy x) \ldots\}
• burn = \{(hot x) (red x) (energy x) (shine x) (glow x) (warm x) \ldots\}
• beam = \{(shine x) (light x) (dazzle x) (happy x) (smile x) \ldots\}

• fire \& glow = \{(warm x) (red x) (friendly x) \ldots\}
• fire \& burn = \{(warm x) (red x) (glow x) \ldots\}
• fire \& beam = \{(light x)\}
• face \& glow = \{(warm x) (friendly x) (x) \ldots\}
• face \& burn = \{(glow x)\}
• face \& beam = \{(shine x) (happy x) (smile x) \ldots\}
Evaluation results

<table>
<thead>
<tr>
<th>Noun</th>
<th>Reference</th>
<th>High</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>The fire</td>
<td>glowed</td>
<td>burned</td>
<td>beamed</td>
</tr>
<tr>
<td>The face</td>
<td>glowed</td>
<td>beamed</td>
<td>burned</td>
</tr>
<tr>
<td>The child</td>
<td>strayed</td>
<td>roamed</td>
<td>digressed</td>
</tr>
<tr>
<td>The discussion</td>
<td>strayed</td>
<td>digressed</td>
<td>roamed</td>
</tr>
<tr>
<td>The sales</td>
<td>slumped</td>
<td>declined</td>
<td>slouched</td>
</tr>
<tr>
<td>The shoulders</td>
<td>slumped</td>
<td>slouched</td>
<td>declined</td>
</tr>
</tbody>
</table>

- 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: multiplicative and combined

<table>
<thead>
<tr>
<th>Model</th>
<th>High</th>
<th>Low</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>NonComp</td>
<td>0.27</td>
<td>0.26</td>
<td>0.08</td>
</tr>
<tr>
<td>Add</td>
<td>0.59</td>
<td>0.59</td>
<td>0.04</td>
</tr>
<tr>
<td>Weighted Add</td>
<td>0.35</td>
<td>0.34</td>
<td>0.09</td>
</tr>
<tr>
<td>Cross-word effect</td>
<td>0.47</td>
<td>0.45</td>
<td>0.09</td>
</tr>
<tr>
<td>Multiply</td>
<td>0.42</td>
<td>0.28</td>
<td>0.17</td>
</tr>
<tr>
<td>Combined</td>
<td>0.38</td>
<td>0.28</td>
<td>0.19</td>
</tr>
<tr>
<td>UpperBound</td>
<td>4.94</td>
<td>3.25</td>
<td>0.40</td>
</tr>
</tbody>
</table>

© 2010 Eduard Hovy
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 $b$:

$$a' = (a \odot R_b^{-1}(r), R_a^{-\{r\}}, R_a^{-1})$$

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

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

© 2010 Eduard Hovy
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

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

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

\[
\text{eat} \overset{\text{instr}}{=} \{(:\text{isa ingest-action}) (:\text{instr fork})\\
\text{(:agent ((}\text{person 0.9) (}\text{animal 0.00001) (:John 0.004) ...})\\
\text{(:patient ((}\text{food 0.9) (}\text{pasta 0.4) (}\text{veg 0.24) ...}))\\
\text{(:loc ((}\text{dining-room 0.4) (}\text{restaurant 0.4) ...})) ...}\}\}
\]
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 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
  – \[ I = p \cdot \log p \]

• 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

© 2010 Eduard Hovy
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:

<table>
<thead>
<tr>
<th>Proposition</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>person#n1:eat:food#n2:with family</td>
<td>6</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with chopstick</td>
<td>2</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with spoon</td>
<td>2</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with and</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with glass</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with variety</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with husband</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with hand</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with president</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with child</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with Ginsburg</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with dressing</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with fork</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with globalizat</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with parent</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with cornichon</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with Stanley</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with meat</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with opponent</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with gusto</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with Cleopatra</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with blood</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with fruit</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with mother</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with mustard</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with money</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with Newhouse</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with group</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with kid</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with mid-after</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with student</td>
<td>1</td>
</tr>
<tr>
<td>person#n1:eat:food#n2:with friend</td>
<td>1</td>
</tr>
</tbody>
</table>
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:
   
   \[
   [\text{rel} \ \text{head} \ \text{mod} \ 1 \ \text{doc-id} \ \text{sent-id}]
   \]

7. When done, add together all the identical triples/props:
   
   \[
   [\text{rel} \ \text{head} \ \text{mod} \ \text{total} \ ((\text{doc-id}_1 \ \text{sent-id}_1) \ (\text{doc-id}_2 \ \text{sent-id}_2) \ ...)]
   \]

8. Regroup as needed (e.g., sort under the heads):
   
   \[
   [\text{head} \ (\text{rel} \ (\text{mod}_1 \ \text{total}_1) \ (\text{mod}_2 \ \text{total}_2) \ ...) \ ((\text{doc-id}_1 \ \text{sent-id}_1) \ ...)]
   \]

   \[
   (\text{rel} \ (\text{mod}_1 \ \text{total}_1) \ (\text{mod}_2 \ \text{total}_2) \ ...) \ ((\text{doc-id}_1 \ \text{sent-id}_1) \ ...)]
   \]

© 2010 Eduard Hovy
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)
Sparsity: Smoothing methods

• Data Sparsity problem: Background knowledge is only as complete as data in corpus; long propositions provide few examples

• Approach: Collect what you can, place scores in WN, ‘smooth’ probabilities to nearby (similar?) terms

• Example: for a verb, can we discover all fillers?
  – First populate in WordNet all examples found in corpus
  – Approach 1: Discover verbs’ selectional preferences
    • Method: compute ‘density’ to find most-specific superclass with max. chosen children
  – Approach 2: Discover probability of any term to fill given verb role
    • Method: use various smoothing methods (incl. EM) to discover probability of every synset/term
    • Kneser-Ney Smoothing, our own formula, etc.
Evaluation

• Choose proposition (“people eat food with ?”)
  – 6 verbs, each separately for Subj, Dobj, and one Prep per verb
• Calculate probability scores for all WordNet Object terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNSET{SID-04330163-N : Words[W-04330163-N-1-tableware]}</td>
<td>3.69231</td>
</tr>
<tr>
<td>SYNSET{SID-05163219-N : Words[W-05163219-N-1-external_body_part]}</td>
<td>3.51487</td>
</tr>
<tr>
<td>Lemma: family #Count: 7.0</td>
<td>3.24826</td>
</tr>
<tr>
<td>SYNSET{SID-04232784-N : Words[W-04232784-N-1-spoon]}</td>
<td>1.78571</td>
</tr>
<tr>
<td>SYNSET{SID-13427890-N : Words[W-13427890-N-1-linear_unit]}</td>
<td>1.68539</td>
</tr>
<tr>
<td>SYNSET{SID-13575001-N : Words[W-13575001-N-1-containerful]}</td>
<td>1.47820</td>
</tr>
<tr>
<td>Lemma: eye #Count: 3.0</td>
<td>1.40515</td>
</tr>
<tr>
<td>SYNSET{SID-03305870-N : Words[W-03305870-N-1-finger]}</td>
<td>1.40515</td>
</tr>
<tr>
<td>Lemma: utensil #Count: 3.0</td>
<td>1.40515</td>
</tr>
</tbody>
</table>

• Take most-likely 20 and least-likely 20
• Formulate and pose question on Mechanical Turk
  – Choices: Always Reasonable / Sometimes / Never / I don’t know
• Obtain answers, discard bad annotators

© 2010 Eduard Hovy
Evaluation: MTurk statistics

- Poor results if paid less than 4c per 10 choices
- Generally discarded at least 3 people per run

© 2010 Eduard Hovy
Evaluation results

- System vs each annotator (Cohen Kappa):
  - A1: 0.65
  - A2: 0.7
  - A3: 0.5
  - A4: 0.65
  - Average system vs annotator: 0.625

- All annotators to one another pairwise (Cohen kappa):
  - A1-A2: 0.64467
  - A1-A3: 0.63350
  - A1-A4: 0.69309
  - A2-A3: 0.38144
  - A2-A4: 0.64467
  - A3-A4: 0.63350
  - Average of all pairwise annotators: 0.60512

- Precision: 0.7
- Recall: 0.933
- F-score-B1: 0.8
- System vs majority vote of all annotators (Cohen kappa): 0.65
- Krippendorff’s alpha-reliability: 0.4144
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

© 2010 Eduard Hovy
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

© 2010 Eduard Hovy
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

© 2010 Eduard Hovy
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):

  \[
  \text{prop( Type, Form : DependencyConstrains : NodeConstrains ).}
  \]

- 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'

- `?NPN 'pass':X:'touchdown'`
  - NPN 712 'pass':for:'touchdown'
  - NPN 24 'pass':include:'touchdown'

- `?NVP 'pass':Y:'touchdown'`
  - NVP 189 'pass':catch:'touchdown'
  - NVP 26 'pass':complete:'touchdown'

- `?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.*

<table>
<thead>
<tr>
<th>Implicit</th>
<th>(More) explicit</th>
</tr>
</thead>
<tbody>
<tr>
<td>San Francisco’s Eric Davis</td>
<td>Eric Davis plays for San Francisco</td>
</tr>
<tr>
<td>Eric Davis intercepted pass</td>
<td>—</td>
</tr>
<tr>
<td>Steve Walsh pass</td>
<td>Steve Walsh threw pass</td>
</tr>
<tr>
<td></td>
<td>Steve Walsh threw interception</td>
</tr>
<tr>
<td>Young touchdown pass</td>
<td>Young completed pass for touchdown</td>
</tr>
<tr>
<td>touchdown pass to Brent Jones</td>
<td>Brent Jones caught pass for touchdown</td>
</tr>
</tbody>
</table>

These are inferences on the language side

© 2010 Eduard Hovy
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

![Diagram](diagram.png)

?> 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
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':'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'

© 2010 Eduard Hovy
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

Enrichment

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

© 2010 Eduard Hovy
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:
  \[ DS(\text{doc}) = \{(r_{w_i}s_i)\} \text{ for } i \text{ different open-class words} \]
  where \( r = \text{‘word-inside-doc’} \) except for stop words
  \( w_i = \text{word and } s_i = \text{word’s count} \)

• 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 instan*al 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 Pennacchioitt, 2008
  – Text enrichment: Peñas and Hovy, COLING 2010
  – Coref: many people