Logistics

• Project proposals due February 18
• Read resources and related papers, discuss initial ideas
• Readings related to lecture---Yay or Nay
• Assignments posted this week
Project Proposal

1. Find a relevant (key) research paper for your topic

2. Write a summary of that research paper and what you took away from it as key ideas that you hope to use

3. Write what you plan to work on and how you can innovate in your final project work

4. Describe as needed
   - A project plan, relevant existing literature, the kind(s) of models you will use/explore; the data you will use (and how it is obtained), and how you will evaluate success
From Words to Word Meaning

• Words as units of text
  • BoW prominent assumption
  • Feature extraction for classification
  • Alternatives to explicit feature extraction sought

• Word representation
  • Numerical representation for words
    • Embed words in a vector space
    • Permit comparing words
Words as vectors

• Sentiment analysis:
  • Feature is a word identity
    • Feature 5: 'The previous word was "terrible"
    • requires **exact word** to be in training and test
  • With **embeddings**:
    • Feature is a word vector
    • Previous word was vector [35,22,17…]
    • In the test set we might see a similar vector [34,21,14…]
    • Generalize with **similar but unseen** words
Word-Level Models of Meaning

• Language described from 3 perspectives
  • Relations between words
  • Compositionality of how words are formed
  • Distributional properties of word co-occurrence
Word Meaning to Sentence Meaning

Starting unit: words
the, cat, cuddly, by, door

Words combine into phrases
the cuddly cat, by the door

Phrases can combine into bigger phrases
the cuddly cat by the door
Word-Level Models of Meaning

• Language described from 3 perspectives
  • Relations between words
  • Compositionality of how words are formed
  • Distributional properties of word co-occurrence
Distributional Hypothesis

**Distributional hypothesis**, stated by linguist John R. Firth (1957) as:

“You shall know a word by the company it keeps.”

≈ “words that occur in similar contexts have similar meanings”

- One way to define "usage":
  
  words are defined by their environments (the words around them)

- Zellig Harris (1954):
  
  - If A and B have almost identical environments we say that they are synonyms.
Idea 1: Defining meaning by linguistic distribution

• Let's define the meaning of a word by its distribution in language use, meaning its neighboring words or grammatical environments.
  • Distributional representation of words
Context counting

(C1) *A bottle of ______ is on the table*
(C2) *People like ______.*
(C3) *Don’t have ______ before you drive.*
(C4) *_______ is made out of corn*

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>tesgüino</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>loud</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Motor oil</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>tortillas</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>choices</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wine</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Idea 2: Meaning as a point in space (Osgood et al. 1957)

• 3 affective dimensions for a word
  • **valence**: pleasantness
  • **arousal**: intensity of emotion
  • **dominance**: the degree of control exerted

<table>
<thead>
<tr>
<th></th>
<th>Word</th>
<th>Score</th>
<th>Word</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Valence</strong></td>
<td>love</td>
<td>1.000</td>
<td>toxic</td>
<td>0.008</td>
</tr>
<tr>
<td></td>
<td>happy</td>
<td>1.000</td>
<td>nightmare</td>
<td>0.005</td>
</tr>
<tr>
<td><strong>Arousal</strong></td>
<td>elated</td>
<td>0.960</td>
<td>mellow</td>
<td>0.069</td>
</tr>
<tr>
<td></td>
<td>frenzy</td>
<td>0.965</td>
<td>napping</td>
<td>0.046</td>
</tr>
<tr>
<td><strong>Dominance</strong></td>
<td>powerful</td>
<td>0.991</td>
<td>weak</td>
<td>0.045</td>
</tr>
<tr>
<td></td>
<td>leadership</td>
<td>0.983</td>
<td>empty</td>
<td>0.081</td>
</tr>
</tbody>
</table>

• Hence the connotation of a word is a vector in 3-space

NRC VAD Lexicon (Mohammad 2018)
We'll discuss 2 kinds of embeddings

- Distributional
- Distributed
Distributional Embeddings

• **Context counting**
  • Words are represented by *counts* of nearby words (left and right context window)
  • Weighted by PPMI (positive pointwise mutual information)

• Intuition: weigh the association between two words by asking how much more the two words co-occur in our corpus than we would have a priori expected them to appear by chance
Pointwise Mutual Information

Do outcomes $x$ and $y$ co-occur more than if they were independent?

$$\text{PMI}(X, Y) = \log_2 \frac{P(x, y)}{P(x)P(y)}$$

**PMI between two words:** (Church & Hanks 1989)

Do words $x$ and $y$ co-occur more than if they were independent?

$$\text{PMI}(\text{word}_1, \text{word}_2) = \log_2 \frac{P(\text{word}_1, \text{word}_2)}{P(\text{word}_1)P(\text{word}_2)}$$
Positive Pointwise Mutual Information

- PMI ranges from $-\infty$ to $+\infty$
- But negative values are problematic
  - Things are co-occurring less than we expect by chance
  - Unreliable without enormous corpora
    - Imagine $w_1$ and $w_2$ whose probability is each $10^{-6}$
    - Hard to be sure $p(w_1, w_2)$ is significantly different than $10^{-12}$

- Positive PMI (PPMI) between word1 and word2:
  
  $$\text{PPMI}(\text{word}_1, \text{word}_2) = \max \left( \log_2 \frac{P(\text{word}_1, \text{word}_2)}{P(\text{word}_1)P(\text{word}_2)}, 0 \right)$$
Distributional Embeddings

• **Context counting**
  • Words are represented by **counts** of nearby words (left and right context window)
  • Weighted by PPMI (positive pointwise mutual information)

• **Sparse** vectors, dimensionality $|V|$
Computing PPMI on a term-context matrix

- Matrix $F$ with $W$ rows (words) and $C$ columns (contexts)
- $f_{ij}$ is # of times $w_i$ occurs in context $c_j$

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}} \quad p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}} \quad p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}} \quad ppmi_{ij} = \begin{cases} pmi_{ij} & \text{if } pmi_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

---

<table>
<thead>
<tr>
<th></th>
<th>computer</th>
<th>data</th>
<th>result</th>
<th>pie</th>
<th>sugar</th>
<th>count(w)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cherry</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>442</td>
<td>25</td>
<td>486</td>
</tr>
<tr>
<td>strawberry</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>60</td>
<td>19</td>
<td>80</td>
</tr>
<tr>
<td>digital</td>
<td>1670</td>
<td>1683</td>
<td>85</td>
<td>5</td>
<td>4</td>
<td>3447</td>
</tr>
<tr>
<td>information</td>
<td>3325</td>
<td>3982</td>
<td>378</td>
<td>5</td>
<td>13</td>
<td>7703</td>
</tr>
<tr>
<td>count(context)</td>
<td>4997</td>
<td>5673</td>
<td>473</td>
<td>512</td>
<td>61</td>
<td>11716</td>
</tr>
</tbody>
</table>
\[ p_{ij} = \frac{f_{ij}}{W \sum_{i=1}^{C} \sum_{j=1}^{C} f_{ij}} \]

\[ p(w=\text{information}, c=\text{data}) = \frac{3982}{11716} = 0.3399 \]
\[ p(w=\text{information}) = \frac{7703}{11716} = 0.6575 \]
\[ p(c=\text{data}) = \frac{5673}{11716} = 0.4842 \]

<table>
<thead>
<tr>
<th></th>
<th>computer</th>
<th>data</th>
<th>result</th>
<th>pie</th>
<th>sugar</th>
<th>count(w)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cherry</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>442</td>
<td>25</td>
<td>486</td>
</tr>
<tr>
<td>strawberry</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>60</td>
<td>19</td>
<td>80</td>
</tr>
<tr>
<td>digital</td>
<td>1670</td>
<td>1683</td>
<td>85</td>
<td>5</td>
<td>4</td>
<td>3447</td>
</tr>
<tr>
<td>information</td>
<td>3325</td>
<td>3982</td>
<td>378</td>
<td>5</td>
<td>13</td>
<td>7703</td>
</tr>
<tr>
<td>count(context)</td>
<td>4997</td>
<td>5673</td>
<td>473</td>
<td>512</td>
<td>61</td>
<td>11716</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{PPMI} &= \frac{\sum_{i=1}^{C} f_{ij}}{W} \\
p(w_i) &= \frac{1}{N} \sum_{j=1}^{C} f_{ij} \\
p(c_j) &= \frac{1}{N} \sum_{i=1}^{W} f_{ij}
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>computer</th>
<th>data</th>
<th>result</th>
<th>pie</th>
<th>sugar</th>
<th>p(w)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cherry</td>
<td>0.0002</td>
<td>0.0007</td>
<td>0.0008</td>
<td>0.0377</td>
<td>0.0021</td>
<td>0.0415</td>
</tr>
<tr>
<td>strawberry</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.0051</td>
<td>0.0016</td>
<td>0.0068</td>
</tr>
<tr>
<td>digital</td>
<td>0.1425</td>
<td>0.1436</td>
<td>0.0073</td>
<td>0.0004</td>
<td>0.0003</td>
<td>0.2942</td>
</tr>
<tr>
<td>information</td>
<td>0.2838</td>
<td>0.3399</td>
<td>0.0323</td>
<td>0.0004</td>
<td>0.0011</td>
<td>0.6575</td>
</tr>
<tr>
<td>p(context)</td>
<td>0.4265</td>
<td>0.4842</td>
<td>0.0404</td>
<td>0.0437</td>
<td>0.0052</td>
<td></td>
</tr>
</tbody>
</table>
The centroid is the multidimensional version of the mean; the centroid of a set of vectors is sometimes referred to as the cosine of their tf-idf or PPMI vectors; high cosine, high similarity. This entire model vectors are sparse (since most values are zero).

In summary, the vector semantics model we've described so far represents a target by tf-idf (for term-document matrices) or PPMI (for term-term matrices), and the values are discounted. (discounting) all the non-zero values. The larger the constant (0.02 0.09 0.28 0.0437 0.0052)

\[
\text{ppmi}_{ij} = \log_2 \frac{p_{ij}}{p_i \cdot p_j}
\]

<table>
<thead>
<tr>
<th></th>
<th>computer</th>
<th>data</th>
<th>result</th>
<th>pie</th>
<th>sugar</th>
<th>p(w)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cherry</td>
<td>0.0002</td>
<td>0.0007</td>
<td>0.0008</td>
<td>0.0377</td>
<td>0.0021</td>
<td>0.0415</td>
</tr>
<tr>
<td>strawberry</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.0051</td>
<td>0.0016</td>
<td>0.0068</td>
</tr>
<tr>
<td>digital</td>
<td>0.1425</td>
<td>0.1436</td>
<td>0.0073</td>
<td>0.0004</td>
<td>0.0003</td>
<td>0.2942</td>
</tr>
<tr>
<td>information</td>
<td>0.2838</td>
<td>0.3399</td>
<td>0.0323</td>
<td>0.0004</td>
<td>0.0011</td>
<td>0.6575</td>
</tr>
</tbody>
</table>

| p(context)     | 0.4265   | 0.4842 | 0.0404 | 0.0437 | 0.0052 |

\[
pmi(\text{information}, \text{data}) = \frac{0.3399}{0.6575 \times 0.4842} = 0.0944
\]

Resulting PPMI matrix (negatives replaced by 0)
Weighting PMI

• PMI is biased toward infrequent events
  • Very rare words have very high PMI values
• Two solutions:
  • Give rare words slightly higher probabilities
  • Use add-one smoothing (which has a similar effect)
Traditional Approach

• Context counting
  • Count left and right context in a window
  • Reweight with Pointwise Mutual Information
  • Reduce dimensionality with SVD or NNMF
  • Why?
  • Latent Semantic Analysis of documents [Deerwester et al. 1988]
Spare to Dense Vectors

Singular value decomposition (SVD) of PPMI weighted co-occurrence matrix

\[
\begin{bmatrix}
X \\
|V| \times |V|
\end{bmatrix} =
\begin{bmatrix}
W \\
|V| \times |V|
\end{bmatrix}
\begin{bmatrix}
\sigma_1 & 0 & 0 & \ldots & 0 \\
0 & \sigma_2 & 0 & \ldots & 0 \\
0 & 0 & \sigma_3 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & \sigma_v \\
|V| \times |V| & |V| \times |V| & |V| \times |V|
\end{bmatrix}
\begin{bmatrix}
C \\
|V| \times |V|
\end{bmatrix}
\]

\[
\begin{bmatrix}
X \\
|V| \times |V|
\end{bmatrix} =
\begin{bmatrix}
W \\
|V| \times k
\end{bmatrix}
\begin{bmatrix}
\sigma_1 & 0 & 0 & \ldots & 0 \\
0 & \sigma_2 & 0 & \ldots & 0 \\
0 & 0 & \sigma_3 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & \sigma_k \\
|V| \times k & k \times k
\end{bmatrix}
\begin{bmatrix}
C \\
k \times |V|
\end{bmatrix}
\]

Only keep the top k (e.g., 100) singular values!
Traditional Approach

• Context counting

Other methods include Brown clustering and hierarchical clustering based on bigram mutual information.
Word Embeddings

• Context counting

• Prediction-based
  • Vector space representations learned on unlabeled linear context (i.e., left/right words)
  • Dense vectors

SENN A [Collobert and Weston, 2008; Collobert et al., 2011]: Multi-layer DNN w/ ranking-loss objective; BoW and sentence-level feature layers, followed by std. NN layers. Similar to [Bengio et al., 2003].
Distributed Embeddings

• Context counting

• Prediction-based
  • Vector space representations learned on unlabeled linear context (i.e., left/right words)
  • Representation created by training a classifier to predict whether a word is likely to appear nearby
  • Breakthrough idea word2vec [Mikolov et al 2013]
    • Continuous Bag of Words idea (using context words to predict target)
    • Skip-gram (predict surrounding context words given current word)
    • Demo: https://code.google.com/p/word2vec
Distributed Embeddings

• Context counting

• Prediction-based
  • Big idea: self-supervision:
  • A word $c$ that occurs near $w$ in the corpus acts as the gold "correct answer" for supervised learning
  • No need for human labels

[Bengio et al. (2003); Collobert et al. (2011)]
Distributed Embeddings
Other Approaches

• Canonical Correlation Analysis
  Word-context correlation [Dhillon et al., 2011, 2012]
  Multilingual correlation [Faruqui and Dyer, 2014; Lu et al., 2015]

• Multi-sense embeddings [Reisinger and Mooney, 2010; Neelakantan et al., 2014]

• Task-tailored embeddings to capture specific types of similarity/semantics
  Lexicon evidence (PPDB, WordNet, FrameNet) [Xu et al., 2014; Yu and Dredze, 2014; Faruqui et al., 2014; Wieting et al., 2015]
  Combining advantages of global matrix factorization and local context window methods [GloVe; Pennington et al., 2014]
Multi-view Embeddings via CCA

Before CCA
- foul
- ugly
- awful
- charming
- magnificent
- splendid
- elegant
- cute
- gorgeous
- marvelous
- hideous
- beastly
- grotesque
- horrid

After CCA
- elegant'
- charming'
- cute'
- gorgeous'
- magnificent'
- splendid'
- marvelous'
- ugly'
- pretty'
- awful'
- hideous'
- beastly'
- grotesque'
- horrid'

[Faruqui and Dyer 2014]
Evaluation

• Extrinsic method
  • Use embeddings for a task and see if performance improves
  • Can be expensive (time) but still most important evaluation metric

• Analogy: solve problems of the form $a:b :: a^* :b^*$, given $a$, $b$, and $a^*$, find $b^*$
Evaluation

• Measure of similarity
  • Cosine of angle between vectors --- length ignored
  • Vectors are normalized to unit length before they are used for similarity calculation, making cosine similarity and dot-product equivalent. [Levy et al., 2015]
  • Most applications of word embeddings explore not the word vectors themselves, but relations between them to solve, for example, similarity and word relation tasks. For these tasks, it was found that using normalised word vectors improves performance. [Wilson and Schakel, 2015]
Evaluation

• Importance of word length [Schakel and Wilson, 2015]
  • A word that is consistently used in a similar context will be represented by a longer vector than a word of the same frequency that is used in different contexts.

• Not only the direction, but also the length of word vectors carries important information.

• Word vector length furnishes, in combination with term frequency, a useful measure of word significance.
Evaluation

• Intrinsic method
  • Fast to compute, but not clear if it really helps downstream tasks

  • Similarity: compute correlation between an algorithm’s word similarity scores and word similarity ratings assigned by humans.
  • WordSim-353 (Finkelstein et al., 2002) is a commonly used set of ratings from 0 to 10 for 353 noun pairs
    • (plane, car) had an average score of 5.77.

  • Analogy: solve problems of the form \( a:b :: a^* : b^* \), given \( a, b, \) and \( a^* \), find \( b^* \)
Limitation

• Variability
  • randomness in the initialization and sampling
  • word2vec may produce different results even from the same dataset, and individual documents in a collection impact the resulting embeddings [Tian et al. 2016, Hellrich and Hahn 2016, Antoniak and Mimno 2018]

• Best practice to train multiple embeddings with bootstrap sampling over documents and average the results [Antoniak and Mimno, 2018]
Sentence Structure for Sentence Meaning

Humans communicate complex ideas by composing words together into bigger units to convey complex meanings.

Listeners need to work out what modifies [attaches to] what.

A model needs to understand sentence structure in order to be able to interpret language correctly.
Models of composition

• Initial approaches
  • Point-wise sum, tensor product [Mitchell and Lapata, 2010; Smolensky 1990]
    • Worked well for adjective-noun and noun-noun phrases
    • Fail to capture structural differences
      • Lice on dogs; lice and fleas
      • Fails on recursion
        • nice toilette-trained spayed short-haired Siamese cat
Models of composition

• Initial approaches
  • Matrix-vector compositionality [Baroni and Zamparelli, 2010; Zanzotto et al., 2010; Grefenstette and Sadrzadeh, 2011; Socher et al., 2011; Yessenalina and Cardie, 2011]
    • content words (such as nouns) are vectors
    • functional words (such as determiners) are functions mapping from expressions of one type onto composite expressions of the same or other types.
Sentence Structure for Sentence Meaning

Humans communicate complex ideas by composing words together into bigger units to convey complex meanings.

Listeners need to work out what modifies [attaches to] what.

A model needs to understand sentence structure in order to be able to interpret language correctly.
Two Views of Linguistic Structure

• Sentence interpreted via Constituency structure
  • Sets of rules of how words are grouped to form phrases

• Sentence represented as a Dependency structure
  • shows which words depend on (modify, attach to, or are arguments of) which other words
Constituency Structure

• A sentence as a set of constituents

• Sentence interpreted via Constituency Grammars
  • Sets of rules of how words are grouped to form phrases
  • Context-Free Grammars (CFG)
  • Popularized by Noam Chomsky
Constituency Parsing

- A sentence as a set of constituents
- **Constituency** parsing: task of recognizing a sentence and assigning a **constituency** structure to it

- NP -> det N
- VP -> V NP
- PP -> prep NN
- VP -> VP PP
- S -> NP VP
Constituency Parsing
Why Constituency-based structures?

• Typically useful for fixed word-order languages
  • Grammar checking: If a sentence can’t be parsed, it may have grammatical errors (or at least hard to read)
• Intermediate representations
  • Syntax-based understanding
  • MT (pre-NMT) & Low-resource MT
  • Information Extraction

• Several parsers (late 1980- early 2000s)
  • CKY parser
  • Earley parser
  • Now NN-based parsers
Performance

Detailed English Results (New)
Treebanks

- Corpus of sentences with parse trees
- Penn Treebank

\[
\begin{align*}
\text{(a)} & \quad (\text{NP-SBJ} \quad \text{(DT That)}) \\
& \quad (\text{JJ cold}) \quad (\text{, ,}) \\
& \quad (\text{JJ empty}) \quad (\text{NN sky}) \\
& \quad (\text{VP} \quad \text{(VBD was)}) \\
& \quad (\text{ADJP-PRD} \quad (\text{JJ full}) \\
& \quad (\text{PP} \quad (\text{IN of}) \\
& \quad (\text{NP} \quad (\text{NN fire}) \\
& \quad (\text{CC and}) \\
& \quad (\text{NN light}) ))))
\end{align*}
\]

\[
\begin{align*}
\text{(b)} & \quad (\text{NP-SBJ} \quad \text{The/DT flight/NN}) \\
& \quad (\text{VP} \quad \text{should/MD}) \\
& \quad (\text{VP} \quad \text{arrive/VB}) \\
& \quad (\text{PP-TMP} \quad \text{at/IN}) \\
& \quad (\text{NP} \quad \text{eleven/CD a.m./RB}) \\
& \quad (\text{NP-TMP} \quad \text{tomorrow/NN}) ))))
\end{align*}
\]

Figure 12.7 Parsed sentences from the LDC Treebank3 version of the (a) Brown and (b) ATIS corpora.
Treebanks

- Robust grammars
  - Context-free grammar rules

```
VP → VBD PP
VP → VBD PP PP
VP → VBD PP PP PP
VP → VBD PP PP PP PP
VP → VB ADVP PP
VP → VB PP ADVP
VP → ADVP VB PP
```

~4500 rules for VP
Two Views of Linguistic Structure

• Sentence interpreted via **Constituency structure**
  • Sets of rules of how words are grouped to form phrases

• Sentence represented as a **Dependency structure**
  • shows which words depend on (modify, attach to, or are arguments of) which other words
Two Views of Structure
Dependency Structure

I prefer the morning flight through Denver
Dependency Structure

• Relations between words
  • binary, asymmetric
  • subject, prepositional object, apposition

• Denoted via arrows with labels
  • Arrow connects a head with a dependent

• Dependencies form a connected, acyclic, single-root graph
Dependency Parsing

- Syntactic parsing: task of recognizing a sentence and assigning a structure to it.
- **Dependency** parsing: the task of recognizing a sentence and assigning a **dependency** structure to it.
Pāṇini’s grammar (c. 5th century BCE)

Gallery: [http://wellcomeimages.org/indexplus/image/L0032691.html](http://wellcomeimages.org/indexplus/image/L0032691.html)

[CC BY 4.0](http://creativecommons.org/licenses/by/4.0) File:Birch bark MS from Kashmir of the Rupavatra Wellcome L0032691.jpg
Dependency Grammars

- Characteristics
  - Lexical/syntactic dependencies between words
  - The top-level predicate of a sentence is the root
  - Simpler to parse than context-free grammars
  - Particularly useful for free word order languages
Techniques (1)

- Dynamic programming
  - CKY – similar to lexicalized PCFG, cubic complexity (Eisner 96)

\[
\text{nsubj} \quad \text{dobj} \\
\downarrow \quad \downarrow \\
\text{Mary likes cats} \\
\quad \rightarrow \\
\text{likes} \\
\text{nsubj} \quad \text{dobj} \\
\uparrow \quad \uparrow \\
\text{Mary likes cats}
\]
Techniques (2)

- Constraint-based methods
  - Maruyama 1990, Karlsson 1990
  - Example
    - word(pos(x)) = DET \implies \text{label}(X) = \text{NMOD}, \text{word(mod(x))} = \text{NN}, \text{pos(x)} < \text{mod(x)}
    - A determiner (DET) modifies a noun (NN) on the right with the label NMOD.
  - NP complete problem; heuristics needed

- Constraint graph
  - For initial constraint graph using a core grammar: nodes, domains, constraints
  - Find an assignment that doesn’t contradict any constraints. If more than one assignment exists, add more constraints.
Techniques (3)

- Deterministic parsing
  - Covington 2001
  - MaltParser by Nivre
    - shift/reduce as in a shift/reduce parser
    - reduce creates dependencies with the head on either the left or the right

- Graph-based methods
  - Maximum spanning trees (MST)
    - MST Parser by McDonald et al.
MaltParser (Nivre 2008)

• Very similar to shift–reduce parsing.
• It includes the following components
  – A stack
  – A buffer
  – Set of dependencies (arcs)
• The reduce operations combine an element from the stack and one from the buffer
• Arc–eager parser
  – The actions are shift, reduce, left–arc, right–arc
Universal Dependencies

Universal Dependencies (UD) is a framework for consistent annotation of grammar (parts of speech, morphological features, and syntactic dependencies) across different human languages. UD is an open community effort with over 300 contributors producing nearly 200 treebanks in over 100 languages. If you're new to UD, you should start by reading the first part of the Short Introduction and then browsing the annotation guidelines.

- Universal dependencies defined
- Same annotation standard
- Applicable to wider set of languages, including free word-order
- Efficient parsing algorithms
- Useful in applications including information extraction
Semantic Role Labeling

Who
---
did what to whom
---
at where?

The police officer detained the suspect at the scene of the crime

Agent  Predicate  Theme  Location
# Semantic Roles

<table>
<thead>
<tr>
<th>Thematic Role</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT</td>
<td>The volitional causer of an event</td>
</tr>
<tr>
<td>EXPERIENCER</td>
<td>The experiencer of an event</td>
</tr>
<tr>
<td>FORCE</td>
<td>The non-volitional causer of the event</td>
</tr>
<tr>
<td>THEME</td>
<td>The participant most directly affected by an event</td>
</tr>
<tr>
<td>RESULT</td>
<td>The end product of an event</td>
</tr>
<tr>
<td>CONTENT</td>
<td>The proposition or content of a propositional event</td>
</tr>
<tr>
<td>INSTRUMENT</td>
<td>An instrument used in an event</td>
</tr>
<tr>
<td>BENEFICIARY</td>
<td>The beneficiary of an event</td>
</tr>
<tr>
<td>SOURCE</td>
<td>The origin of the object of a transfer event</td>
</tr>
<tr>
<td>GOAL</td>
<td>The destination of an object of a transfer event</td>
</tr>
</tbody>
</table>

**Figure 19.1** Some commonly used thematic roles with their definitions.
Use of Semantic Roles

- Infer meaning components even when structurally dissimilar
  - John broke the window.
  - John broke the window with a rock.
  - The rock broke the window.
  - The window broke.
  - The window was broken by John.
Semantic Roles

- Defined with respect to the predicates and nouns
  - Semantically related verbs/nouns
  - Available in manually created resources
    - FrameNet and PropBank
FrameNet (2004)

- Project at UC Berkeley led by Chuck Fillmore for developing a database of frames, general semantic concepts with an associated set of roles.
- Roles are specific to frames, which are “invoked” by the predicate, which can be a verb, noun, adjective, adverb
  - JUDGEMENT frame
    - Invoked by: V: blame, praise, admire; N: fault, admiration
    - Roles: JUDGE, EVALUEE, and REASON
- Specific frames chosen, and then sentences that employed these frames selected from the British National Corpus and annotated by linguists for semantic roles.
- Initial version: 67 frames, 49,013 sentences, 99,232 role fillers
PropBank := proposition bank (2005)

- Project at Colorado led by Martha Palmer to add semantic roles to the Penn treebank.
- Proposition := verb + a set of roles
- Annotated over 1M words of Wall Street Journal text with existing gold-standard parse trees.
- Statistics:
  - 43,594 sentences    99,265 propositions
  - 3,324 unique verbs  262,281 role assignments
Semantic Role Labeling

- POS tagging
- Parsing
- Feature extraction
Semantic Role Labeling

- POS tagging
- Parsing
- Feature extraction

Figure 19.5  Parse tree for a PropBank sentence, showing the PropBank argument labels. The dotted line shows the path feature NP↑S↓VP↓VBD for ARG0, the NP-SBJ constituent The San Francisco Examiner.
Features for Semantic Role Labeling

• Common features:
  • Governing predicate
  • Constituent type
  • Head word of the constituent
  • Part of speech of the head word
  • Path in the parse tree from the constituent to the predicate
  • Whether the voice of the surrounding clause is active or passive
  • Whether the constituent appears before or after the predicate
  • Set of expected arguments for the verb phrase
  • Named entity type of the constituent
  • First and last word(s) of the constituent
Global Optimization

• Semantic roles are not independent of one another!
• Many approaches perform a second pass to address **global consistency**
  • Viterbi decoding
  • Reranking
  • Integer linear programming
Evaluation of Semantic Role Labeling

- **True positives:** Argument labels assigned to the correct word sequence or parse constituents
- Then, we can compute our standard NLP metrics:
  - Precision
  - Recall
  - F-measure
Accuracy for such feature-based SRL models then highly depends on accuracy of underlying parse tree!

- So quite high SRL results when using ground-truth parses
- Much lower results with automatically-predicted parses!