Today: algorithms for WSD

- **Unsupervised**
  - Using glosses (Lesk 1986; Kilgarriff and Rosenzweig, 2000)
  - Using WN and Lexical Chains (Barzilay and Elhadad, 1997)

- **Supervised**
  - Using context words and machine learning

- **Semi-supervised**
  - Using Context and Bootstrapping (Yarowsky, 1995)
  - Using Properties of WN-Graph (Navigli and Lapata, 2010).
Word Sense Disambiguation: the task

Helps in various NLP tasks:

- Machine Translation
- Question Answering
- Information Retrieval
- Text Classification

Task-specific senses, or define task generally on basis of dictionary such as WordNet.

Organization of Wordnet

- Wordnet groups words into synsets (synonym sets).
- One synset = one sense; this constitutes the sense's definition.
- Homonyms and polysemous word forms are therefore part of multiple synsets.
- Senses are often indicated by slashes and numbers: interest/1, interest/2...
- Synsets are organized into a hierarchical structure by the use of hyponymy, e.g. a dog is-a pet, pet is-a animal
- Other relations are also recorded: metonymy, peronymy
- Play with it: http://wordnetweb.princeton.edu/perl/webwn
- Are Wordnet senses too fine grained?
WN example – “interest”

Noun

- **S (n) interest**, **involvement** (a sense of concern with and curiosity about someone or something) “an interest in music”
- **S (n) sake**, **interest** (a reason for wanting something done) “for your sake”; “died for the sake of his country”; “in the interest of safety”; “in the common interest”
- **S (n) interest, interestingness** (the power of attracting or holding one’s attention (because it is unusual or exciting etc.)) “they said nothing of great interest”; “primary colors can add interest to a room”
- **S (n) interest** (a fixed charge for borrowing money; usually a percentage of the amount borrowed) “how much interest do you pay on your mortgage?”
- **S (n) interest, stake** (law) a right or legal share of something; a financial involvement with something “they have interests all over the world”; “a stake in the company’s future”
- **S (n) interest, interest group** (usually plural) a social group whose members control some field of activity and who have common aims “the iron interests stepped up production”
- **S (n) pastime, interest, pursuit** (a diversion that occupies one’s time and thoughts (usually pleasantly)) “sailing is her favorite pastime”; “his main pastime is gambling”; “he counts reading among his interests”; “they criticized the boy for his limited pursuits”

Verb:

- **S (v) interest** (excite the curiosity of; engage the interest of)
- **S (v) concern, interest, occupy, worry** (be on the mind of) “I worry about the second Germanic consonant shift”
- **S (v) matter to, interest** (be of importance or consequence) “This matters to me!”

“interest/3” – a closer look

- **S: (n) interest** (a fixed charge for borrowing money; usually a percentage of the amount borrowed) “how much interest do you pay on your mortgage?”

  **direct hyponym** / **full hyponym**
  - **S: (n) compound interest** (interest calculated on both the principal and the accrued interest)
  - **S: (n) simple interest** (interest paid on the principal alone)

  **direct hyponym** / **inherited hypernym** / **sister term:**
  - **S: (n) fixed charge, fixed cost, fixed costs** (a periodic charge that does not vary with business volume (as insurance or rent or mortgage payments etc.))
    - **S: (n) charge** (the price charged for some article or service) “the admission charge”
    - **S: (n) cost** (the total spent for goods or services including money and time and labor)
      - **S: (n) outgo, spending, expenditure, outlay** (money paid out; an amount spent)
    - **S: (n) transferred property, transferred possession** (a possession whose ownership changes or lapses)
      - **S: (n) possession** (anything owned or possessed)
    - **S: (n) relation** (an abstraction belonging to or characteristic of two entities or parts together)
      - **S: (n) abstraction, abstract entity** (a general concept formed by extracting common features from specific examples)
    - **S: (n) entity** (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))
**Lesk’s Algorithm**

**Supervised WSD**


**Graph-based WSD**

Possible WSD algorithms

- Word itself + Words in context window + bootstrapping (Yarowsky) **Semi-supervised**
- Word itself + Words in context window + Machine Learning (Senseval; many) **Supervised**
- Word itself + Words in gloss (Lesk) **Unsupervised; Dictionary**
- Word itself + Neighbours in WN relations (Barzilay and Elhadad) **Unsupervised; Dictionary**
- Word itself + entire WN subnet per sense (Navigli and Lapata) **Unsupervised; Dictionary**
- Parallel texts in other languages (Diab, Resnik) **Unsupervised; Data**

---

**“interest/4” – a closer look**

S: (n) **interest**, **stake** (law) a right or legal share of something; a financial involvement with something; “they have interests all over the world”; “a stake in the company’s future”

direct hyponym/inherited hypernym/sister term:
- S: (n) **share**, **portion**, **part**, **percentage** (assets belonging to or due to or contributed by an individual person or group) “he wanted his share in cash”
- S: (n) **assets** (anything of material value or usefulness that is owned by a person or company)
  - S: (n) **possession** (anything owned or possessed)
  - S: (n) **relation** (an abstraction belonging to or characteristic of two entities or parts together)
  - S: (n) **abstraction**, **abstract entity** (a general concept formed by extracting common features from specific examples)
  - S: (n) **entity** (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))
Typically there is more than one ambiguous word in the sentence.

- Several rare ferns grow on the steep banks of the burn where it runs into the lake.

Ambiguous: rare, steep, bank, burn, run

But: humans do not perceive this sentence as ambiguous at all. Hearer selects that combination of lexical readings which leads to the most normal possible utterance-in-context. [Assumption of cooperation in communication, Grice]

Lesk Algorithms

- Chooses the sense whose gloss shares most words with target word’s neighbourhood

Kilgarriff and Rosenzweig (2000): Simplified Lesk

function SIMPLIFIED LESK(word, sentence) returns best sense of word

best-sense := most frequent sense for word

max-overlap := 0

collection := set of words in sentence

for each sense in senses of word do

signature := set of words in gloss and examples of sense

overlap := COMPUTE_OVERLAP(signature, context)

if overlap > max-overlap then

max-overlap := overlap

best-sense := sense

end

return(best-sense)

- COMPUTE_OVERLAP returns the number of words in common between two sets, ignoring function words or other words on a stop list.
Example: Disambiguation of *bank*

Context: *The bank can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.***

<table>
<thead>
<tr>
<th>bank/1</th>
<th>(a financial institution that accepts deposits and channels the money into lending activities) “he cashed a check at the bank”, “that bank holds the mortgage on my home”</th>
</tr>
</thead>
<tbody>
<tr>
<td>bank/2</td>
<td>(sloping land (especially the slope beside a body of water)) “they pulled the canoe up on the bank”, “he sat on the bank of the river and watched the currents”</td>
</tr>
</tbody>
</table>

- Sense *bank/1* has two (non-stop) words overlapping with the context (*deposits* and *mortgage*)
- Sense *bank/2* has zero, so sense *bank/1* is chosen.

Original Lesk (1986) Algorithm

- Instead of comparing a target word’s signature with the context words, the target signature is compared with the signatures of each of the context words.
- Example context: *pine cone*

<table>
<thead>
<tr>
<th>pine/1</th>
<th>kinds of evergreen tree with needle-shaped leaves</th>
</tr>
</thead>
<tbody>
<tr>
<td>pine/2</td>
<td>waste away through sorrow or illness</td>
</tr>
<tr>
<td>cone/1</td>
<td>solid body which narrows to a point</td>
</tr>
<tr>
<td>cone/2</td>
<td>something of this shape whether solid or hollow</td>
</tr>
<tr>
<td>cone/3</td>
<td>fruit of a certain evergreen tree</td>
</tr>
</tbody>
</table>

*cone/3* and *pine/1* are selected:
- overlap for entries *pine/1* and *cone/3* (*evergreen* and *tree*)
- no overlap in other entries
Lesk: Improvements

- Lesk is more complex than Simplified Lesk, but empirically found to be less successful → Simplified Lesk preferred.
- Problem with all Lesk Algorithms: dictionary entries for the target words are short → often there is no overlap with context
- Improvements:
  - Expand the list of words used in the classifier to include words related to, but not contained in their individual sense definitions.
  - Apply a weight to each overlapping word. The weight is the inverse document frequency or IDF. IDF measures how many different documents (in this case glosses and examples) a word occurs in.

Supervised Word Sense Disambiguation

- Words are labelled by their senses:
  - She pays 3% interest/INTEREST-MONEY on the loan.
  - He showed a lot of interest/INTEREST-CURIOSITY in the painting.
- Different to situation in Lesk, which is “unsupervised”, and able to disambiguate all ambiguous words in a text
- Similar to POS tagging:
  - define features that indicate one sense over another
  - learn a model that predicts the correct sense given the features
- e.g., Naive Bayes
An electric guitar and bass player stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.

- **Collocational feature**: (directly neighbouring words in specific positions)
  \[w_{i-2}, \text{POS}, w_{i-1}, \text{POS}, w_{i+1}, \text{POS}, w_{i+2}, \text{POS}\]
  [guitar, NN, and, CC, player, NN, stand, VB]

- **Bag of Words feature**: (any content words in a 50 word window)
  12 most frequent content words from bass collection: [fishing, big, sound, player, fly, rod, pound, double, runs, playing, guitar, band]
  \[\rightarrow [0,0,0,1,0,0,0,0,0,0,1,0]\]

Goal: choose the best sense \(\hat{s}\) out of the set of possible senses \(S\) for an input vector \(\vec{F}\):

\[\hat{s} = \arg\max_{s \in S} P(s|\vec{F})\]

It is difficult to collect statistics for this equation directly.

Rewrite it using Bayes' rule:

\[\hat{s} = \arg\max_{s \in S} \frac{P(\vec{F}|s)P(s)}{P(\vec{F})}\]

Assumption that \(F_i\) are independent gives us:

\[P(\vec{F}|s) \approx \prod_{i=1}^{j} P(F_i|s)\]
Naive Bayesian Classifier

- Naive Bayes Classifier:
  \[ \hat{s} = \arg\max_{s \in S} P(s) \prod_{n} P(F_i|s) \]

- Parameter Estimation (Max. likelihood):
  - How likely is sense \( s_i \) for word form \( w_j \)?
  \[ P(s_i) = \frac{\text{count}(s_i, w_j)}{\text{count}(w_j)} \]
  - How likely is feature \( f_j \) given sense \( s_i \)?
  \[ P(f_j|s_i) = \frac{\text{count}(s_i, f_j)}{\text{count}(s_i)} \]

Intrinsic Evaluation

- Sense accuracy: percentage of words tagged identical with hand-tagged in test set
- How can we get annotated material cheaply?
  - Pseudo-words
    - create artificial corpus by conflating unrelated words
    - example: replace all occurrences of banana and door with banana-door
  - Multi-lingual parallel corpora
    - translated texts aligned at the sentence level
    - translation indicates sense
- SENSEVAL competition
  - bi-annual competition on WSD
  - provides annotated corpora in many languages
  - “Lexical Sample” Task for supervised WSD
  - “All-word” Task for unsupervised WSD
Baselines for supervised WSD

- First (most frequent) sense
- LeskCorpus (Simplified, weighted Lesk, with all the words in the labeled SEMEVAL corpus sentences for a word sense added to the signature for that sense).
- LeskCorpus is the best-performing of all the Lesk variants (Kilgarriff and Rosenzweig, 2000; Vasilescu et al., 2004)

Semi-supervised WSD by Bootstrapping

Yarowsky’s (1995) algorithm uses two powerful heuristics for WSD:

- **One sense per collocation**: nearby words provide clues to the sense of the target word, conditional on distance, order, syntactic relationship.
- **One sense per discourse**: the sense of a target word is consistent within a given document.

The Yarowsky algorithm is a **bootstrapping** algorithm, i.e., it requires a small amount of annotated data:

- It starts with a small seed set, trains a classifier on it, and then applies it to the whole data set (bootstrapping);
- Reliable examples are kept, and the classifier is re-trained.

Figures and tables in this section from Yarowsky (1995).
Step 1: Extract all instances of a polysemous or homonymous word.

Step 2: Generate a seed set of labeled examples:
- either by manually labeling them;
- or by using a reliable heuristic.

Example: target word plant: As seed set take all instances of
- plant life (sense A) and
- manufacturing plant (sense B).
Classification

**Step 3a:** Train classifier on the seed set.

**Step 3b:** Apply classifier to the entire sample set. Add those examples that are classified reliably (probability above a threshold) to the seed set.

Yarowsky uses a **decision list** classifier:

- rules of the form: collocation → sense
- rules are ordered by log-likelihood:
  \[
  \log \frac{P(sense_A|collocation_i)}{P(sense_B|collocation_i)}
  \]
- Classification is based on the first rule that applies.

<table>
<thead>
<tr>
<th>LogL</th>
<th>Collocation</th>
<th>Sense</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.10</td>
<td><em>plant</em> life</td>
<td>→ A</td>
</tr>
<tr>
<td>7.58</td>
<td>manufacturing <em>plant</em></td>
<td>→ B</td>
</tr>
<tr>
<td>7.39</td>
<td>life (within +-2-10 words)</td>
<td>→ A</td>
</tr>
<tr>
<td>7.20</td>
<td>manufacturing (in +- 2-10 words)</td>
<td>→ B</td>
</tr>
<tr>
<td>6.27</td>
<td>animal (within +-2-10 words)</td>
<td>→ A</td>
</tr>
<tr>
<td>4.70</td>
<td>equipment (within +-2-10 words)</td>
<td>→ B</td>
</tr>
<tr>
<td>4.39</td>
<td>employee (within +-2-10 words)</td>
<td>→ B</td>
</tr>
<tr>
<td>4.30</td>
<td>assembly <em>plant</em></td>
<td>→ B</td>
</tr>
<tr>
<td>4.10</td>
<td><em>plant</em> closure</td>
<td>→ B</td>
</tr>
<tr>
<td>3.52</td>
<td><em>plant</em> species</td>
<td>→ A</td>
</tr>
<tr>
<td>3.48</td>
<td>automate (within +-10 words)</td>
<td>→ B</td>
</tr>
<tr>
<td>3.45</td>
<td>microscopic <em>plant</em></td>
<td>→ A</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
**Step 3c:** Use one-sense-per-discourse constraint to filter newly classified examples:

- If several examples have already been annotated as sense A, then extend this to all examples of the word in the discourse.
- This can form a bridge to new collocations, and correct erroneously labeled examples.

**Step 3d:** repeat Steps 3a–d.
Step 4: Algorithm converges on a stable residual set (remaining unlabeled instances):

- most training examples will now exhibit multiple collocations indicative of the same sense;
- decision list procedure uses only the most reliable rule, not a combination of rules.

Step 5: The final classifier can now be applied to unseen data.

Discussion

Strengths:
- simple algorithm that uses only minimal features (words in the context of the target word);
- minimal effort required to create seed set;
- does not rely on dictionary or other external knowledge.

Weaknesses:
- uses very simple classifier (but could replace it with a more state-of-the-art one);
- not fully unsupervised: requires seed data;
- does not make use of the structure of the sense inventory.

Alternative: Exploit the structure of the sense inventory for WSD:
- Lexical Chains (Barzilay and Elhadad)
- Graph-based (Navigli and Lapata)
Lexical Chain-based Disambiguation

- Idea: think of lexical chains as “topics” in text, related areas, which consist of senses (not word forms)
- Polysemous word forms could thus belong to several lexical chains;
- the word sense disambiguation consists in choosing membership of senses to lexical chain (globally, only one sense can survive)
- Consider several WN lexical relations (with different weights): identity, synonymy, hypo/hypernymy, siblings
- Treat WSD as an optimization problem – optimal groupings contain most senses which are related (strong chains)

Barzilay and Elhadad’s algorithm

- Build combinations of all ambiguous word forms occurring in the text, if you can find a WN connection between them.
- Score different relations as follows:
  - reiteration and synonym: 10
  - antonym: 7,
  - hyperonym and holonym: 4
- After the entire text has been processed, start from strongest chain and claim all ambiguous word forms for it, i.e., delete them from all other chains.
- This produces the correct lexical chains at the same time as the correct word senses.
Example: Lexical Chain construction

Mr. Kenny is the person that invented an anaesthetic machine which uses microcomputers to control the rate at which anaesthetic is pumped into the blood. Such machines are nothing new. But his device uses two microcomputers to achieve much closer monitoring of the pump feeding the anaesthetic to the patient.

Interpretations:

1. [Mr]
2. [Mr, person/1] [Mr, [person/3]
3. [Mr, person/1] [machine/1] [Mr, machine/2] [person/3]
4. [Mr, person/1, machine/2] [Mr] [person/3] [machine/1]

After adding “pump”, “microcomputer”, “device”, the following interpretations are strongest:

1. [Mr, machine/2, person/1] [pump/1, microcomputer, device/1]
2. [Mr, person/1] [machine/1, pump/1, microcomputer, device/1]

The second interpretation wins, because it contains the strongest lexical chain overall.

This means that machine/1 is now correctly disambiguated.

This algorithm is exponential, but a polynomial algorithm exists (Silber and McCoy, 2002).
The internal structure of sense inventories can be exploited even further.

Represent Wordnet as a graph whose nodes are synsets and whose edges are relations between synsets.

The edges are not labeled, i.e., the type of relation between the nodes is ignored.

Figures and tables in this section from Navigli and Lapata (2010).

Example

Wordnet Synsets (senses) of **drink**:

- \{drink^1_n, imibe^3_v\} (take in liquids)
- \{drink^2_v, booze^1_v, fuddle^2_v\} (consume alcohol)
- \{toast^2_v, drink^3_v, pledge^2_v, salute^1_v, wassail^2_v\} (propose a toast)
- \{drink in^1_v, drink^5_v\} (be fascinated, pay close attention)
- \{drink^5_v, tope^1_v\} (be an alcoholic)

Wordnet Synsets (senses) of **milk**:

- \{milk^1_n\} (a white nutritious liquid secreted by mammals and used as food by human beings)
- \{milk^2_n\} (produced by mammary glands of female mammals for feeding their young)
- \{Milk^3_n, Milk River^1_n\} (a river that rises in the Rockies in northwestern Montana and flows eastward to become a tributary of the Missouri River)
- \{milk^4_n\} (any of several nutritive milklike liquids)
Disambiguation algorithm:

1. Use the Wordnet graph to construct a graph that incorporates each content word in the sentence to be disambiguated;
2. Rank each node in the sentence graph according to its importance using graph connectivity measures;
   - **Local measures**: give a connectivity score to an individual node in the graph; use this directly to pick a sense;
   - **Global measures**: assign a connectivity score the to the graph as a whole; apply the measure to each interpretation and select the highest scoring one.
Given a word sequence \( \sigma = (w_1, w_2, \ldots, w_n) \), find all possible word senses of all words; call this set \( V \).

Perform a depth-first search of the Wordnet graph: every time we encounter a node \( v' \in V_\sigma (v' \neq v) \) along a path \( v \rightarrow v_1 \rightarrow \cdots \rightarrow v_k \rightarrow v' \) of length \( L \), we add all intermediate nodes and edges on the path from \( v \) to \( v' \) to the graph \( G \).

For tractability, we set the maximum path length to 6.

Example: graph for *drink milk*.
Example: graph for \textit{drink milk}.

\begin{center}
\begin{tikzpicture}
    
    \node [circle, draw] (drink1) at (0,0) {drink$_1$};
    \node [circle, draw] (drink2) at (1,0) {drink$_2$};
    \node [circle, draw] (drink3) at (2,0) {drink$_3$};
    \node [circle, draw] (drink4) at (3,0) {drink$_4$};
    \node [circle, draw] (drink5) at (4,0) {drink$_5$};
    
    \node [circle, draw] (beverage1) at (1,1) {beverage$_1$};
    \node [circle, draw] (food1) at (2,1) {food$_1$};
    \node [circle, draw] (nutriment1) at (3,1) {nutriment$_1$};
    
    \node [circle, draw] (milk1) at (1,2) {milk$_1$};
    \node [circle, draw] (milk2) at (2,2) {milk$_2$};
    \node [circle, draw] (milk3) at (3,2) {milk$_3$};
    \node [circle, draw] (milk4) at (4,2) {milk$_4$};
    
    \draw (drink1) -- (beverage1); 
    \draw (beverage1) -- (food1); 
    \draw (food1) -- (nutriment1); 
    \draw (nutriment1) -- (milk1); 
    \draw (milk1) -- (milk2); 
    \draw (milk2) -- (milk3); 
    \draw (milk3) -- (milk4); 
    \end{tikzpicture}
\end{center}
Graph Construction

Example: graph for *drink milk*.

![Graph for *drink milk*](image)
Graph Construction

Example: graph for *drink milk*.

![Graph for drink milk](image_url)
Lesk’s Algorithm
Supervised WSD
Graph-based WSD

Graph Construction

Example: graph for *drink milk*.

We get $3 \cdot 2 = 6$ interpretations, i.e., subgraphs obtained when only considering one connected sense of *drink* and *milk*.

A Local Measure: Degree Centrality

Assume a graph with nodes $V$ and edges $E$. Then the **degree** of $v \in V$ is the number of edges terminating in it:

$$deg(v) = |\{\{u, v\} \in E : u \in V\}|$$

**Degree centrality** is the degree of a node normalized by the maximum degree:

$$C_D(v) = \frac{deg(v)}{|V| - 1}$$

For the previous example, $C_D(drink_1) = \frac{3}{14}$, $C_D(drink_2) = C_D(drink_5) = \frac{2}{14}$, and $C_D(milk_1) = C_D(milk_2) = \frac{1}{14}$. So we pick *drink*$_1$, while *milk*$_n$ is tied.
A Global Measure: Edge Density

**Edge density** of a graph is the number of edges compared to a complete graph with $|V|$ nodes (given by $\binom{|V|}{2}$):

$$ED(G) = \frac{|E(G)|}{\binom{|V|}{2}}$$ (3)

The first interpretation of *drink milk* has $ED(G) = \frac{6}{\binom{5}{2}} = \frac{6}{10} = 0.60$, the second one $ED(G) = \frac{5}{\binom{2}{2}} = \frac{5}{10} = 0.50$.

### Evaluation on SemCor

<table>
<thead>
<tr>
<th>Measure</th>
<th>WordNet All</th>
<th>WordNet Poly</th>
<th>EnWordNet All</th>
<th>EnWordNet Poly</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtLesk</td>
<td>47.85</td>
<td>34.05</td>
<td>48.75</td>
<td>35.25</td>
</tr>
<tr>
<td><strong>Degree</strong></td>
<td><strong>50.01</strong></td>
<td><strong>37.80</strong></td>
<td><strong>56.62</strong></td>
<td><strong>46.03</strong></td>
</tr>
<tr>
<td>PageRank</td>
<td>49.76</td>
<td>37.49</td>
<td>56.46</td>
<td>45.83</td>
</tr>
<tr>
<td>HITS</td>
<td>44.29</td>
<td>30.69</td>
<td>52.40</td>
<td>40.78</td>
</tr>
<tr>
<td>KPP</td>
<td>47.89</td>
<td>35.16</td>
<td>55.65</td>
<td>44.82</td>
</tr>
<tr>
<td>Betweenness</td>
<td>48.72</td>
<td>36.20</td>
<td>56.48</td>
<td>45.85</td>
</tr>
<tr>
<td><strong>Compactness</strong></td>
<td>43.53</td>
<td>29.74</td>
<td>48.31</td>
<td>35.68</td>
</tr>
<tr>
<td><strong>Graph Entropy</strong></td>
<td>42.98</td>
<td>29.06</td>
<td>43.06</td>
<td>29.16</td>
</tr>
<tr>
<td><strong>Edge Density</strong></td>
<td>43.54</td>
<td>29.76</td>
<td>52.16</td>
<td>40.48</td>
</tr>
<tr>
<td><strong>First Sense</strong></td>
<td><strong>74.17</strong></td>
<td><strong>68.80</strong></td>
<td><strong>74.17</strong></td>
<td><strong>68.80</strong></td>
</tr>
</tbody>
</table>
Lesk’s Algorithm
Supervised WSD
Graph-based WSD

Evaluation on Semeval All-words Data

<table>
<thead>
<tr>
<th>System</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Unsupervised (Sussex)</td>
<td>45.8</td>
</tr>
<tr>
<td>ExtLesk</td>
<td>43.1</td>
</tr>
<tr>
<td><strong>Degree Unsupervised</strong></td>
<td>52.9</td>
</tr>
<tr>
<td>Best Semi-supervised (IRST-DDD)</td>
<td>56.7</td>
</tr>
<tr>
<td><strong>Degree Semi-Unsupervised</strong></td>
<td>60.7</td>
</tr>
<tr>
<td>First Sense</td>
<td>62.4</td>
</tr>
<tr>
<td>Best Supervised (GAMBL)</td>
<td>65.2</td>
</tr>
</tbody>
</table>

Simone Teufel
L113 Word Meaning and Discourse Understanding
43

Discussion

**Strengths:**
- exploits the structure of the sense inventory/dictionary;
- conceptually simple, doesn’t require any training data, not even a seed set;
- achieves good performance for unsupervised system.

**Weaknesses:**
- performance not good enough for real applications (F-score of 53 on Semeval);
- sense inventories take a lot of effort to create (Wordnet has been under development for more than 15 years).
Lesk’s Algorithm
Supervised WSD
Graph-based WSD

Summary

- The **Lesk** algorithm uses overlap between context and glosses.
- **Supervised WSD** uses context and bag-of-words features and machine learning.
- The **Yarowsky** algorithm uses bootstrapping and two key heuristics:
  - one sense per collocation;
  - one sense per discourse;
- WSD and **Lexical Chain** construction use mutual constraints to pick the best senses.
- **Unsupervised graph-based WSD** creates a graph that represents all possible interpretations of a sentence.
- The nodes with the highest connectivity are picked as correct senses; simple degree is best connectivity measure.

References

**Lesk** (1986): Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In SIGDOC ’86, ACM.

