L113 Word Meaning and Discourse Understanding

Session 2: Word Sense Disambiguation Algorithms

Simone Teufel

MPhil in Advanced Computer Science
Computer Laboratory Natural Language and Information Processing (NLIP) Group
UNIVERSITY OF CAMBRIDGE
Simone.Teufel@cl.cam.ac.uk

2012/2013

Last time: the theory behind word senses

- Homonymy and polysemy
- Tests for ambiguity
- Request to take a look at data: shower

Today:

- shower – your impressions
- WordNet
- Algorithms for Word Sense Disambiguation (WSD)

Organization of Wordnet

- Wordnet groups words into synsets (synonym sets).
- One synset = one sense; this constitutes the senses's definition.
- Homonyms and polysemous word forms are therefore associated with multiple (different) synsets.
- Senses are 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 around with it: http://wordnetweb.princeton.edu/perl/webwn

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))

"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))

Word Sense Disambiguation: the task

- Helps in various NLP tasks:
  - Machine Translation
  - Question Answering
  - Information Retrieval
  - Text Classification
- What counts as "one sense"?
  - Task-specific senses
  - dictionary-defined senses.
- Sense-tagged corpora exist, e.g., SemCor
  - 186 texts with all open class words WN synset tagged (192,639)
  - 166 texts with all verbs WN synset tagged (41,497)

Types of Algorithms for WSD

- Supervised
- Unsupervised
- Semi-supervised

Supervised: We know the answers for many examples and can use them to learn from their (automatically determinable) characteristics. We then apply the learned model to a comparable set of examples (not the same ones!)

- lexical items occurring near bank/1 and bank/2 (Schütze)

Unsupervised WSD

In unsupervised WSD, we start with no known answers. Instead, we use only unannotated texts to infer underlying relationships using, for instance:

- dictionary glosses (Lesk)
- mutual sense constraints (Barzilay and Elhadad)
- properties of WN-Graph (Navigli and Lapata).

Semi-supervised WSD

In Semi-supervised WSD, we know the answers for some examples, and can gain more examples from the data by finding similar cases and inferring the answers they should have.

- Bootstrapping of context words (Yarowsky)
- Active Learning

What is used as input data?

- 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
- Word itself + Words in context window + Machine Learning (Senseval; many) Supervised; Data+Annotation
- Word itself + Words in context window + bootstrapping (Yarowsky) Semi-supervised: Data+Examples
- Parallel texts in other languages (Diab, Resnik) Unsupervised; Data

Simplified Lesk (Kilgarriff and Rosenzweig; 2000)

```
function SIMPLIFIED LESK(word, sentence) returns best sense of word
  best-sense := most frequent sense for word
  max-overlap := 0
  context := 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)
```

- Algorithm chooses the sense of target word whose gloss shares most words with sentence
- 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.*

bank/1	(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"*
bank/2	(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"*

- 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*

pine/1	kinds of evergreen tree with needle-shaped leaves
pine/2	waste away through sorrow or illness

cone/1	solid body which narrows to a point
cone/2	something of this shape whether solid or hollow
cone/3	fruit of a certain evergreen tree

cone/3 and *pine/1* are selected:

- overlap for entries *pine/1* and *cone/3* (*evergreen* and *tree*)
- no overlap in other entries

Idea behind Original Lesk: Mutual Disambiguation

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: Improvements

- Lesk is more complex than Simplified Lesk, but empirically found to be less successful
- Problem with all Lesk Algorithms: dictionary entries for the target words are short $\rightarrow$ often no overlap with context at all
- Possible improvements:
  - Expand the list of words used 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 with their senses:
  - She pays 3% interest/INTEREST-MONEY on the loan.
  - He showed a lot of interest/INTEREST-CURIOSITY in the painting.
- Define features that (you hope) will indicate one sense over another
- Train a statistical model that predicts the correct sense given the features, e.g., Naive Bayes
- Classifier is trained for each target word separately
- Unlike situation in Lesk, which is unsupervised, and able to disambiguate all ambiguous words in a text

Features for Supervised WSD

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}, POS, w_{i-1}, POS, w_{i+1}, POS, w_{i+2}, 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]

Naive Bayes

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

$$\hat{s} = argmax_{s \in S} P(s|\vec{F})$$

- It is difficult to collect statistics for this equation directly.
- Rewrite it using Bayes' rule:

$$\hat{s} = argmax_{s \in S} = \frac{P(\vec{F}|s)P(s)}{P(\vec{F})}$$

- Assumption that F_i are independent gives us:

$$P(\vec{F}|s) \approx \prod_{n}^{j=1} P(F_i|s)$$

Naive Bayesian Classifier

- Naive Bayes Classifier:

$$\hat{s} = argmax_{s \in S} P(s) \prod_{n}^{j=1} P(F_i|s)$$

- Parameter Estimation (Max. likelihood):
  - How likely is sense s_i for word form w_j?

$$P(s_i) = \frac{count(s_i, w_j)}{count(w_j)}$$

  - How likely is feature f_j given sense s_i?

$$P(f_j|s_i) = \frac{count(s_i, f_j)}{count(s_i)}$$

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 (SemCor corpus)

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 words 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).

Seed Set

Step 1: Extract all instances of a polysemous or homonymous word.

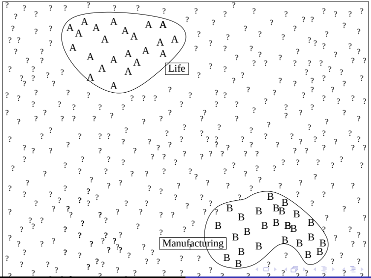
Step 2: Generate a seed set of labeled examples:

- 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).

Seed Set



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 $\rightarrow$ 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.

Classification

LogL	Collocation	Sense
8.10	*plant* life	$\rightarrow$ A
7.58	manufacturing *plant*	$\rightarrow$ B
7.39	life (within +-2-10 words)	$\rightarrow$ A
7.20	manufacturing (in +- 2-10 words)	$\rightarrow$ B
6.27	animal (within +-2-10 words)	$\rightarrow$ A
4.70	equipment (within +-2-10 words)	$\rightarrow$ B
4.39	employee (within +-2-10 words)	$\rightarrow$ B
4.30	assembly *plant*	$\rightarrow$ B
4.10	*plant* closure	$\rightarrow$ B
3.52	*plant* species	$\rightarrow$ A
3.48	automate (within +-2-10 words)	$\rightarrow$ B
3.45	microscopic *plant*	$\rightarrow$ A
	...	

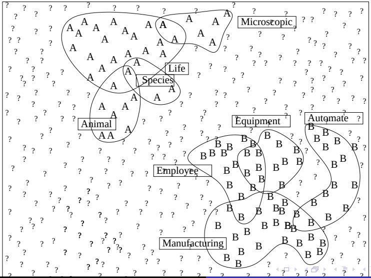
Classification

Step 3c: Use one-sense-per-discourse constraint to filter newly classified examples:

- If several examples in one document have already been annotated as sense A, then extend this to all examples of the word in the rest of the document.
- This can bring in new collocations, and even correct erroneously labeled examples.

Step 3d: repeat Steps 3a–d.

Classification



Generalization

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 a possibly existing dictionary (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 (Okumura and Honda 1994).
- Important: lexical chains consist of senses (not word forms).
- Claim: Lexical chains provide the context for the resolution of ambiguous terms
- Each noun occurrence must belong to only one lexical chain; this determines its sense.
- Algorithm proceeds by assigning possible senses of each word form to lexical chains.
- Assignment allows for various WN relations, with different weights: reiteration (10), synonymy (10), antonymy (7), hyperonymy(4), holonymy (4)

WSD and Lexical Chains: an optimization problem

- While the algorithm is running, polysemous word forms belong to several lexical chains.
- After the algorithm finishes, optimise for the overall grouping of words to terms
- Assumption: optimal groupings (strong chains) are those that have attracted the highest number of related senses
  - Start from strongest chain and claim all ambiguous word forms for it, i.e., delete them from all other chains.
- Mutual disambiguation should achieve 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 controll 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.

WN has the following senses:

- person/1 – normal, but person/3 – (a grammatical category) "stop talking about yourself in the third person"
- machine/1 – normal, but machine/2: (an efficient person) "the boxer was a magnificent fighting machine"

The first three steps

- process "Mr", only one sense: [Mr]
- process "person", 2 senses:
  [[Mr, person/1]] [[Mr], [person/3]]
  ($\rightarrow$ resulting in two interpretations)
- process "machine", 2 senses:
  [[Mr, person/1] [machine/1]] [[Mr, machine/2] [person/3]]
  [[Mr, person/1, machine/2]] [[Mr] [person/3] [machine/1]]

  ($\rightarrow$ resulting in four interpretations)

Example: Lexical Chain construction

- After adding "pump", "microcomputer", "device", the following interpretations contain the strongest chains:
  [[Mr, machine/2, person/1] [pump/1, microcomputer, device/1]]

  [[Mr, person/1] [machine/1, pump/1, microcomputer, device/1]]
- The second interpretation 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).

Lexical Chains: Other Applications

- Summarisation (Barzilay, Elhadad 97)
- IR (Stairmand, 96)
- Detection of malapropisms (St Onge 98)
- Topic Segmentation (Hearst 97, Kazman 96)
- Hypertext link generation (Green 97, 99)
- Word Prediction (Fazli and Hirst 2002)

Graph-Based WSD (Navigli and Lapata (2010)

- 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

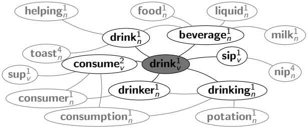
Wordnet Synsets (senses) of drink:

- {**drink**^1_v, *imbibe*^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**^4_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)

Graph for first sense of *drink*



Graph Construction

Disambiguation algorithm:

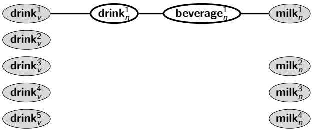
1. Use the Wordnet graph to construct a graph that incorporates each content word in the sentence to be disambiguated;
2. Rank each node in the sentence graph according to its importance using **graph connectivity measures;**
   - **Local measures:** give a connectivity score to an individual node in the graph; use this directly to select 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.

Graph Construction

- Given a word sequence $\sigma = (w_1, w_2, \dots, w_n)$, 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 $v \to v_1 \to \cdots \to v_k \to v'$ 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.

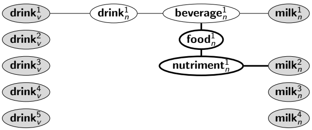
Graph Construction

Example: graph for *drink milk*.



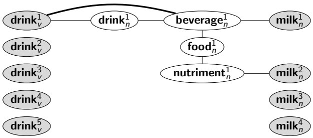
Graph Construction

Example: graph for *drink milk*.



Graph Construction

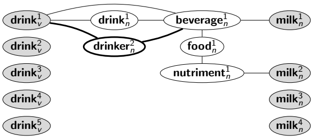
Example: graph for *drink milk*.



WordNet
WSD Algorithms
Lesk Algorithms
Supervised WSD
Yarowsky
Graph-based WSD

Graph Construction

Example: graph for *drink milk*.

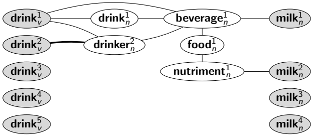


Simone Teufel L113 Word Meaning and Discourse Understanding 43

WordNet
WSD Algorithms
Lesk Algorithms
Supervised WSD
Yarowsky
Graph-based WSD

Graph Construction

Example: graph for *drink milk*.

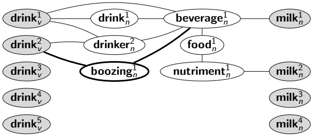


Simone Teufel L113 Word Meaning and Discourse Understanding 43

WordNet
WSD Algorithms
Lesk Algorithms
Supervised WSD
Yarowsky
Graph-based WSD

Graph Construction

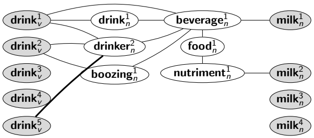
Example: graph for *drink milk*.



Simone Teufel L113 Word Meaning and Discourse Understanding 43

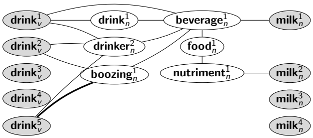
Graph Construction

Example: graph for *drink milk*.



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:

$$deg(v) = |\{\{u, v\} \in E : u \in V\}| \tag{1}$$

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

$$C_D(v) = \frac{deg(v)}{|V| - 1} \tag{2}$$

For the previous example, $C_D(drink_v^1) = \frac{3}{14}$, $C_D(drink_v^2) = C_D(drink_v^5) = \frac{2}{14}$, and $C_D(milk_n^1) = C_D(milk_n^2) = \frac{1}{14}$. So we pick $drink_v^1$, 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 $\binom{|V|}{2}$):

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

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

Evaluation on SemCor

	Measure	WordNet All	WordNet Poly	EnWordNet All	EnWordNet Poly
	Random	39.13	23.42	39.13	23.42
	ExtLesk	47.85	34.05	48.75	35.25
Local	**Degree**	**50.01**	**37.80**	**56.62**	**46.03**
Local	PageRank	49.76	37.49	56.46	45.83
Local	HITS	44.29	30.69	52.40	40.78
Local	KPP	47.89	35.16	55.65	44.82
Local	Betweenness	48.72	36.20	56.48	45.85
Global	Compactness	43.53	29.74	48.31	35.68
Global	Graph Entropy	42.98	29.06	43.06	29.16
Global	Edge Density	43.54	29.76	52.16	40.48
	First Sense	74.17	68.80	74.17	68.80

Evaluation on Semeval All-words Data

System	F
Best Unsupervised (Sussex)	45.8
ExtLesk	43.1
Degree Unsupervised	52.9
Best Semi-supervised (IRST-DDD)	56.7
First Sense	62.4
Best Supervised (GAMBL)	65.2

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 0.53 on Semeval);
- sense inventories take a lot of effort to create (Wordnet has been under development for more than 15 years).

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** finds the most connected nodes (senses) in a graph that represents all possible interpretations of a sentence.

Essential Reading

- Jurasfky and Martin, chapter 20.1-20.4.
- Barzilay and Elhadad (1997)
- Navigli and Lapata (2010)

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.

Yarowsky (1995): Unsupervised Word Sense Disambiguation rivaling Supervised Methods. Proceedings of the ACL.

Barzilay and Elhadad (1997): Using lexical chains for summarization, ACL workshop on Summarisation, ACL-1997.

Navigli and Lapata (2010): An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 32(4), IEEE Press, 2010, pp. 678-692.