L114 Lexical Semantics

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

2014/2015

Last time: the theory behind word senses

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

Today:

- 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 (part-of), paronymy (same stem, morphological variation)
- 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!"*

Multilingual aspect of word sense ambiguity

Example: *interest* translated into German

- Zins: financial charge paid for load
- Anteilnahme: curiousness
- Anteil: stake in a company
- Hobby: hobby
- Interesse: all other senses

Word Senses: Example *interest*

- *She pays 3% interest on the loan.*
- *He showed a lot of interest in the painting.*
- *Microsoft purchased a controlling interest in Google.*
- *Playing chess is one of my interests.*
- *He said nothing of great interest.*
- *It is in the national interest to invade the Bahamas.*
- *I only have your best interest in mind.*
- *Business interests lobbied for the legislation.*
- *Primary colours can add interest to a room.*

Zins; Anteilnahme; Anteil; Hobby; Interesse

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 (e.g., Decadt et al. 04)

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

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]

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

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)
  $[w_{i-2}, POS_{i-2}, w_{i-1}, POS_{i-1}, w_{i+1}, POS_{i+1}, w_{i+2}, POS_{i+2}]$
  [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 $\widehat{s}$ out of the set of possible senses S for an input vector $\overrightarrow{F}$:

$$\widehat{s} = argmax_{s \in S} P(s|\overrightarrow{F})$$

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

$$\widehat{s} = argmax_{s \in S} = \frac{P(\overrightarrow{F}|s)P(s)}{P(\overrightarrow{F})}$$

- Drop $P(\overrightarrow{F})$ – it is a constant factor in *argmax*
- Assume that F_i are independent:

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

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.

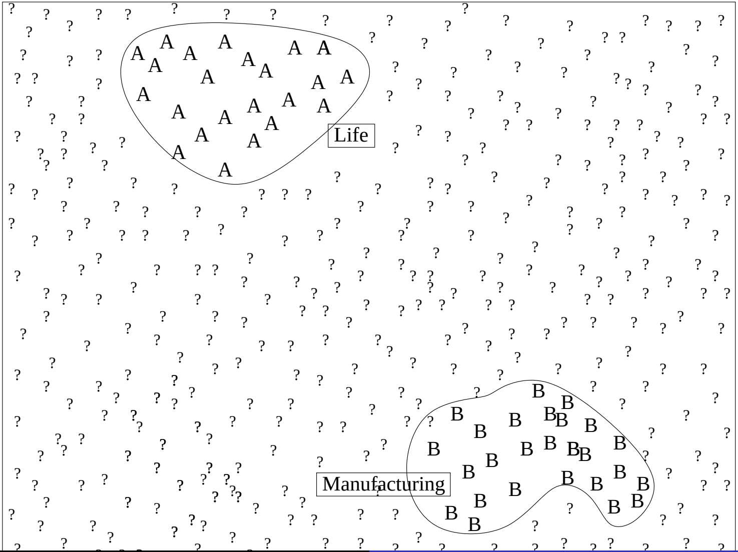
Step 2: Generate a seed set of labeled examples:

- 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 → 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
	...	

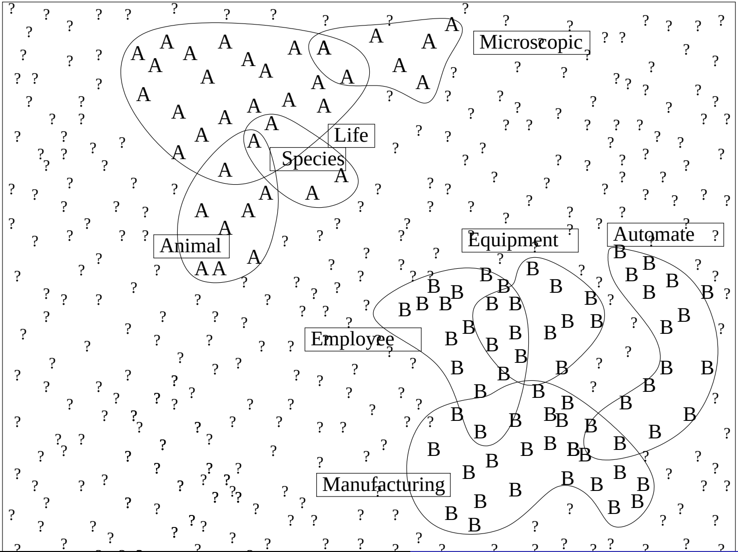
Classification

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:

- Graph-based (Navigli and Lapata) – NEXT TIME

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.

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.