Foundations of Natural Language Processing
Lecture 20c
Word Sense Disambiguation

Alex Lascarides

School of informatics

So far

- NL and its use relies on commonsense inference and hence on lexical semantics
- Words are sense ambiguous
- Some word senses are the result of productive rules that apply to whole word classes
- NLP influenced by inference involving word senses

Now: Word sense disambiguation

Word sense disambiguation (WSD)

- For many applications, we would like to disambiguate senses
  - we may be only interested in one sense
  - searching for chemical plant on the web, we do not want to know about chemicals in bananas
- Task: Given a sense ambiguous word, find the sense in a given *context*
- Popular topic, data driven methods perform well

WSD as classification

- Given a word token in context, which sense (class) does it belong to?

- We can train a supervised classifier, assuming sense-labeled training data:

  - She pays 3% **interest/INTEREST-MONEY** on the loan.
  - He showed a lot of **interest/INTEREST-CURIOSITY** in the painting.
  - Playing chess is one of my **interests/INTEREST-HOBBY**.

- **SensEval** and later **SemEval** competitions provide such data

  - held every 1-3 years since 1998
  - provide annotated corpora in many languages for WSD and other semantic tasks

What kind of classifier?

Lots of options available:

- Naïve Bayes, MaxEnt (see Lecture 7)
- Decision lists (see J&M, 20.2.2)
- Decision trees (see any ML textbook)
- Neural approaches. . . (next year in NLU+)

Naïve Bayes for WSD

$$\begin{aligned}
\hat{s} = &\quad \arg\max_{s\in S} P(s|\vec{f}) & \\
= &\quad \arg\max_{s\in S} \frac{P(\vec{f}|s)P(s)}{P(\vec{f})} & \text{Bayes} \\
= &\quad \arg\max_{s\in S} P(\vec{f}|s)P(s) & P(\vec{f}) \text{ is fixed} \\
\approx &\quad \arg\max_{s\in S} P(s) \prod_{j=1}^{n} P(f_j|s) & \text{cond. independence}
\end{aligned}$$

- Naïve Bayes requires estimates of:
  - The prior probability of each class (sense)
  - The probability of each feature given each class

- These can be estimated from the training data.

- But what features to use? (Same question for other classifiers!)

Simple features

- Directly neighboring words (and/or their lemmas)
  - **interest** paid
  - rising **interest**
  - lifelong **interest**
  - **interest** rate
  - **interest** piqued
- Any content words in a 50 word window
  - pastime
  - financial
  - lobbied
  - pursued

More features

- Syntactically related words
- Syntactic role in sense
- Topic of the text
- Part-of-speech tag, surrounding part-of-speech tags

Of course, with NB we have the usual problem with correlated features. MaxEnt doesn't assume they are independent.

Evaluation

- Extrinsic: test as part of IR, QA, or MT system
- Intrinsic: evaluate classification accuracy or precision/recall against gold-standard senses
- Baseline: choose the most frequent sense (sometimes hard to beat)

Issues with WSD

- Not always clear how fine-grained the gold-standard should be
- Difficult/expensive to annotate corpora with fine-grained senses
- Classifiers must be trained separately for each word
  - Hard to learn anything for infrequent or unseen words
  - Requires new annotations for each new word
  - Motivates unsupervised and semi-supervised methods (see J&M 20.5, 20.10)

Semantic Classes

- Other approaches, such as **named entity recognition** and **supersense tagging**, define coarse-grained semantic categories like PERSON, LOCATION, ARTIFACT.

- Like senses, can disambiguate: APPLE as ORGANIZATION vs. FOOD.

- Unlike senses, which are *refinements* of particular words, classes are typically larger groupings.

- Unlike senses, classes can be applied to words/names not listed in a lexicon.

Named Entity Recognition

- Recognizing and classifying **proper names** in text is important for many applications. A kind of **information extraction**.
- Different datasets/named entity recognizers use different inventories of classes.
  - Smaller: PERSON, ORGANIZATION, LOCATION, MISCELLANEOUS
  - Larger: sometimes also PRODUCT, WORK_OF_ART, HISTORICAL_EVENT, etc., as well as numeric value types (TIME, MONEY, etc.)
- NER systems typically use some form of feature-based sequence tagging, with features like capitalization being important.
- Lists of known names called **gazetteers** are also important.

Supersenses in WordNet

N:TOPS	N:OBJECT	V:COGNITION
N:ACT	N:PERSON	V:COMMUNICATION
N:ANIMAL	N:PHENOMENON	V:COMPETITION
N:ARTIFACT	N:PLANT	V:CONSUMPTION
N:ATTRIBUTE	N:POSSESSION	V:CONTACT
N:BODY	N:PROCESS	V:CREATION
N:COGNITION	N:QUANTITY	V:EMOTION
N:COMMUNICATION	N:RELATION	V:MOTION
N:EVENT	N:SHAPE	V:PERCEPTION
N:FEELING	N:STATE	V:POSSESSION
N:FOOD	N:SUBSTANCE	V:SOCIAL
N:GROUP	N:TIME	V:STATIVE
N:LOCATION	V:BODY	V:WEATHER
N:MOTIVE	V:CHANGE	

- The **supersense tagging** goes beyond NER to cover all nouns and verbs.

Summary (1)

- In order to support technologies like question answering, we need ways to reason computationally about **meaning**. **Lexical semantics** addresses meaning at the word level.
  - Words can be ambiguous, sometimes with related meanings (**regular polysemy**), and other times with unrelated meanings (**homonymy**).
  - Different words can mean the same thing (**synonymy**).

- Computational lexical databases, notably WordNet, organize words in terms of their meanings.
  - **Synsets** and relations between them such as hypernymy and meronymy.

Summary (2)

- **Word sense disambiguation** is the task of choosing the right sense for the context.
  - Classification with contextual features
  - Relying on dictionary senses has limitations in granularity and coverage

- **Semantic classes**, as in NER and supersense tagging, are a coarser-grained representation for semantic disambiguation and generalization.

Next Lecture: Distributional lexical semantics

- What can we learn about a word's meaning from "the company it keeps"?
- What do we do if our thesaurus is incomplete?
- Distributional lexical semantics is about learning word meaning from the contexts in which words appear