Word Sense Disambiguation

CS-585

Natural Language Processing

Derrick Higgins

Word Sense Disambiguation

- Many words have multiple meanings
  - E.g, river *bank*, financial *bank*
- **Problem:** Assign proper sense to each ambiguous word in text
- **Applications:**
  - Machine translation
  - Information retrieval
  - Semantic interpretation of text

Sense Tagging

- Idea: Treat sense disambiguation like POS tagging, just with “semantic tags”

Distributional Similarity

- The problems differ:
  - POS tags depend on specific structural cues (mostly neighboring tags)
  - Senses depend on semantic context – less structured, longer distance dependency

Approaches

- Dictionary-Based Learning
  - Learn to distinguish senses from dictionary entries
- Supervised learning:
  - Learn from a pretagged corpus
- Unsupervised Learning
  - Automatically cluster word occurrences into different senses

Evaluation

- Train and test on pretagged texts
  - Difficult to come by
- Artificial data: ‘merge’ two words to form an ‘ambiguous’ word with two ‘senses’
  - E.g, replace all occurrences of door and of window with doorwindow and see if the system figures out which is which

Performance Bounds

- How good is (say) 83.2%??
- Evaluate performance relative to lower and upper bounds:
  - Baseline performance: how well does the simplest “reasonable” algorithm do?
  - Human performance: what percentage of the time do people agree on classification?

Inter-rater Reliability (Agreement)

Measure how often humans agree on annotations

– If they don't often agree, then the task is ill-defined

- Agreement probability - P(agree)

  Number of times raters agree / Number of ratings

  – But if 90% of things are annotated as X, then agreement could be high by chance

- Cohen's Kappa

$$\frac{P_{agree} - P_{chance}}{1 - P_{chance}}$$

Inter-rater Reliability (Agreement)

- Cohen's Kappa

$$\frac{P_{agree} - P_{chance}}{1 - P_{chance}}$$

- P_{agree}: Observed agreement rate between annotators (or annotator/system)
- P_{chance}: Expected agreement rate between two annotators assigning labels randomly, but using the true class distribution

Inter-rater Reliability (Agreement)

- For a binary classification task with equiprobable outcomes, P_{chance} is 0.5. We'd expect raters using the two classes with equal frequency to agree half the time.
- So in this case, if $P_{agree} = 0.7$, then

$$
\begin{aligned}
\kappa &= \frac{P_{agree} - P_{chance}}{1 - P_{chance}} \\
&= \frac{0.7 - 0.5}{1 - 0.5} \\
&= 0.4
\end{aligned}
$$

Inter-rater Reliability (Agreement)

- For a distribution with N classes, $P_{chance} = \sum_{i=1}^{N} P_i^2$
- For example, for labels distributed according to $\langle 0.1, 0.3, 0.4, 0.2 \rangle$:

	A (0.1)	B (0.3)	C (0.4)	D (0.2)
A (0.1)	0.01	0.03	0.04	0.02
B (0.3)	0.03	0.09	0.12	0.06
C (0.4)	0.04	0.12	0.16	0.08
D (0.2)	0.02	0.06	0.08	0.04

$$P_{chance} = 0.01 + 0.09 + 0.16 + 0.04 = 0.3$$

Inter-rater Reliability (Agreement)

- For labels distributed according to $\langle 0.1, 0.3, 0.4, 0.2 \rangle$, $P_{chance} = 0.3$
- So if $P_{agree} = 0.7$,

$$
\begin{aligned}
\kappa &= \frac{P_{agree} - P_{chance}}{1 - P_{chance}} \\
&= \frac{0.7 - 0.3}{1 - 0.3} \\
&\approx 0.57
\end{aligned}
$$

DICTIONARY-BASED LEARNING

Dictionary-Based Disambiguation

- Idea: Choose between senses of a word given in a dictionary based on the words in the definitions

`cone`:

- A mass of ovule-bearing or pollen-bearing scales in trees of the pine family or in cycads that are arranged usually on a somewhat elongated axis
- Something that resembles a cone in shape: as a crisp cone-shaped wafer for holding ice cream

Algorithm (Lesk 1986)

- Define $D_i(w)$ as the bag of words in the ith definition for w
- Define $E(w)$ as $\bigcup_i D_i(w)$
- For all senses s_k of w, do:

$$Score(s_k) = similarity\left(D_k(w), \left[\bigcup_{v \in c} E(v)\right]\right)$$

- Choose

$$s = \underset{s_k}{\operatorname{argmax}}\, Score(s_k)$$

Similarity Metrics

$$similarity(X,Y) = \begin{cases} \text{Matching coefficient } |X \cap Y| \\ \text{Dice coefficient } \frac{2|X \cap Y|}{|X|+|Y|} \\ \text{Jaccard coefficient } \frac{|X \cap Y|}{|X \cup Y|} \\ \text{Overlap coefficient } \frac{|X \cap Y|}{\min(|X|,|Y|)} \end{cases}$$

Simple Example

ash:

s_1: a tree of the olive family

s_2: the solid residue left when combustible material is burned

The **fire** had left behind nothing but a pile of ash_2

The ash_1 can be recognized by its serrated **leaves**

After being struck by **lightning** the **maple** was reduced to $ash_?$

Simple Example

ash:

s_1: a tree of the olive family

s_2: the solid residue left when combustible material is burned

The **fire** had left behind nothing but a pile of ash$_2$

The ash can be recognized by its serrated **leaves**

After being struck by **lightning** the **maple** was reduced to ash$_2$

fire:

1. **combustion** or **burning**, in which substances combine chemically with oxygen from the air
2. the shooting of projectiles from weapons

Simple Example

ash:

s_1: a tree of the olive family

s_2: the solid residue left when combustible material is burned

The **fire** had left behind nothing but a pile of ash$_2$

The ash$_1$ can be recognized by its serrated **leaves**

After being struck by **lightning** the **maple** was reduced

to ash$_?$

leaf:

1. a flattened structure of a higher plant or **tree**, typically green and blade-like
2. a thing that resembles a leaf in being flat and thin

Simple Example

ash:

- s_1: a tree of the olive family
- s_2: the solid residue left when combustible material is burned

lightning:

1. the occurrence of a natural electrical discharge of between a cloud and the ground, often causing **combustion**
2. very fast

The **fire** had left behind nothing but a pile of ash$_2$

The ash$_1$ can be recognized by its serrated **leaves**

After being struck by **lightning** the **maple** was reduced to ash$_?$

Simple Example

ash:

s_1: a tree of the olive family

s_2: the solid residue left when combustible material is burned

maple:

1. a **tree** or shrub with lobed leaves, winged fruits, and colorful autumn foliage
2. maple syrup or maple sugar

The **fire** had left behind nothing but a pile of ash$_2$

The ash$_1$ can be recognized by its serrated **leaves**

After being struck by **lightning** the **maple** was reduced to ash$_?$

Some Improvements

- Lesk obtained results of 50-70% accuracy
- Possible improvements:
  - Run iteratively, each time only using definitions of "appropriate" senses for context words
  - Expand each word to a set of synonyms, using a thesaurus

Thesaurus-Based Disambiguation

- Thesaurus assigns subject codes to different words, assigning multiple codes to ambiguous words
- $t(s_k)$= subject code of sense s_k for word w in the thesaurus
- $\delta(t, v) = 1$ iff t is a subject code for word v

“mean”

mean (adj.)
- average 29
- small 32
- middle 68
- contemptible 643
- stingy 819
- shabby 874
- ignoble 876
- sneaking 886
- base 940
- selfish 943

....

<u>29 Mean</u>

N mean, medium, average, balance, mediocrity, generality, golden mean, mid-course, middle, compromise, middle course, state neutrality

V split the difference, take the average, reduce to a mean, strike a balance, pair off

Adj mean, intermediate, middle, average, neutral, mediocre, middle class, commonplace, unimportant

<u>643 Unimportance</u>

N unimportance insignificance nothingness immateriality....

Adj unimportant, of little/small/no account/importance, immaterial, un/non-essential, indifferent, subordinate, inferior, mediocre, average, passable, fair, respectable, tolerable, commonplace, uneventful,...

... pitiful, contemptible, contempt, sorry, mean, meager, shabby, miserable, wretched, vile, scrubby,...

Simple Algorithm

Count up number of context words with same subject code:

for each sense s_k of w_i, do:

$$Score(s_k) = \sum_{v \in c} \delta(t(s_k), v)$$

$$s(w_i) = \operatorname{argmax}_{s_k} Score(s_k)$$

SUPERVISED LEARNING

Supervised Learning

- Each ambiguous word token w_i in the training is tagged with a sense from $\text{Senses}(w_i) = s_1, \ldots, s_k$
- Each word token occurs in a context c_i
  - (usually defined as a window around the word occurrence – up to ~100 words long)
- Each context contains a set of words used as features v_{ij}

Bayesian Classification

- Bayes decision rule:
  - Classify $s(w_i) = \text{argmax}_s P(s \mid c_i)$
- Minimizes probability of error
- How to compute? Use Bayes' Theorem:

$$P(s_k|c) = \frac{P(c|s_k)P(s_k)}{P(c)}$$

Bayes' Classifier (cont.)

- Note that *P(c)* is constant for all senses, therefore:

$$
\begin{aligned}
s(w_i) &= \operatorname{argmax}_s P(s|c) \\
&= \operatorname{argmax}_s \frac{P(c|s)}{P(c)} P(s) \\
&= \operatorname{argmax}_s P(c|s)P(s)
\end{aligned}
$$

$$s(w_i) = \operatorname{argmax}_s (\log P(c \mid s) + \log P(s))$$

Naïve Bayes

- Assume:
  - Features are conditionally independent, given the example class
  - Feature order doesn't matter
  - (bag of words model – repetition counts)

$$
\begin{aligned}
P(c|s) &= P(\{v_j : v_j \in c\}|s) \\
&= \prod_{v_j \in c} P(v_j|s) \\
\log P(c|s) &= \sum_{v_j \in c} \log P(v_j|s)
\end{aligned}
$$

Naïve Bayes Training

- For all senses s_k of w, do:
  - For all words v_j in the vocabulary, do:

$$P\left(v_j | s_k\right) = \frac{Count(v_j, s_k)}{Count(s_k)}$$

- For all senses s_k of w, do:

$$P(s_k) = \frac{Count(s_k)}{Count(w)}$$

Naïve Bayes Classification

- For all senses s_k of w_i, do:

$$Score(s_k) = \log P(s_k)$$

- For all words v_j in the context window c_i, do:

$$Score(s_k) += \log P(v_i|s_k)$$

- Choose

$$s(w_i) = \underset{s_k}{\operatorname{argmax}} Score(s_k)$$

Significant Features

Senses of "drug" (Gale et al. 1992):

'medication' prices, prescription, patent, increase, consumer, pharmaceutical

'illegal substance' abuse, paraphernalia, illicit, alcohol, cocaine, traffickers

UNSUPERVISED LEARNING

Some Issues

- Domain-dependence: In computer manuals, “mouse” will not be evidence for topic “mammal”
- Coverage: “Michael Jordan” will not likely be in a thesaurus, but is an excellent indicator for topic “sports”

Tuning for a Specific Corpus

- Use a naïve-Bayes formulation:

$$P(t \mid c) = \frac{P(t)\prod_{v \in c} P(v|t)}{\prod_{v \in c} P(v)}$$

- Initialize probabilities as uniform
- Re-estimate $P(t)$ and $P(v_j \mid t)$ for each topic t and each word v_j by evaluating all contexts in the corpus, assuming the context has topic t if $P(t \mid c) > \theta$ (where θ is a predefined threshold)
- Disambiguate by choosing the highest probability topic

Training:

for all contexts *c* and topics *t*, **do:**

$$\text{Score}(c,t) = \frac{P(t)\prod_{v\in c}P(v|t)}{P(c)}$$

for all contexts *c*, **let**

$$t(c) = \{t|Score(c,t) > \theta\}$$

for all topics t_l, **let**

$$T_l = \{c|t_l \in t(c)\}$$

for all words v_j , **let**

$$V_j = \{c|v_j \in c\}$$

for all words v_j , topics t_l, **do:**

$$P(v_j|t_l) = \frac{|V_i \cap T_l|}{\cup_j |V_j \cap T_l|}$$

for all topics t_l, **do:**

$$P(t_l) = \frac{\cup_j |V_j \cap T_l|}{\cup_m \cup_j |V_j \cap T_m|}$$

LEVERAGING BILINGUAL DATA

Using a Bilingual Corpus

Use correlations between phrases in two languages to disambiguate

E.g, interest =	'legal share'	(acquire an interest)
	'attention'	(show interest)
In German	Beteiligung erwerben	
	Interesse zeigen	

Depending on where the translations of related words occur, determine which sense applies

Scoring

- Given a context c in which a syntactic relation $R(w, v)$ holds between w and a context word v:
  - Score of sense s_k is the number of contexts c' in the second language such that $R(w', v') \in c'$ where w' is a translation of s_k and v' is a translation of v.
  - Choose highest-scoring sense