Naive Bayes Classifier Approach to Word Sense Disambiguation

Daniel Jurafsky and James H. Martin

Chapter 20
Computational Lexical Semantics
Sections 1 to 2

Seminar in Methodology and Statistics 3/June/2009

Outline

What is WSD?

- WSD is the task of automatically assigning appropriate meaning to a polysemous word within a given contex
- Polysemy is:
  - the ambiguity of an individual word or phrase that can be used (in different contexts) to express two or more different meanings
- Here WSD is discussed in relation to computational lexical semantics

Example of polysemous word

In 1834, Sumner was admitted to the [[**bar (law)**|**bar**]] at the age of twenty-three, and entered private practice in Boston.
It is danced in 3/4 time (like most waltzes), with the couple turning approx. 180 degrees every [[**bar (music)**|**bar**]].
Vehicles of this type may contain expensive audio players, televisions, video players, and [[**bar (counter)**|**bar**]]s, often with refrigerators.
Jenga is a popular beer in the [[**bar (establishment)**|**bar**]]s of Thailand.
This is a disturbance on the water surface of a river or estuary, often cause by the presence of a [[**bar (landform)**|**bar**]] or dune on the riverbed.

Figure: Example sentences of the polysemous word bar

Variants of generic WSD

- Many WSD algorithms rely on contextual similarity to help choose the proper sense of a word in context
- Two variants of WSD include:
  - All words approach and
  - Suppervised or lexical sample approach

Unsupervised WSD approach

All words WSD approach

A system is given entire texts and a lexicon with an inventory of senses for each entry and the system is required to disambiguate every context word in the text, disadvantages:

1. Training data for each word in the test set may not be available
2. The approach of training one classifier per term is not practical

Supervised WSD approach

Supervised WSD approach or lexical sample WSD approach

- Takes as input a word in context along with a fixed inventory of potential word senses and outputs the correct word sense for that use
- The input data is hand-labled with correct word senses
- Unlabeled target words in context can then be labeled using such a trained classifier

Collecting features for Supervised WSD

- Input for Supervised WSD are collected in feature vectors
- A feature vector consits of numeric or nominal values to encode linguistic information as input to most ML algorithms
- Two classes of feature vectors extracted from neighbouring context are:
  1. Bag-of-word feature vectors and
  2. Collocational feature vectors

Classes of feature vectors

Bag-of-word feature vectors

- These are unordered set of words with their exact position ignored

Classes of feature vectors

Collocation feature vectors

- A collocation is a word or phrase in a position of specific relationship to a target word
- Thus a collocation encodes information about specific positions located to the left or right of the target word e.g. take **bass** as target
  An electric guitar and **bass** player stand off to one side, ...
- Collocation feature vector, extracted from a window of two words to the right and left of the target word, made up of the words themselves and their respective POS, that is:
  $[w_{i-2}, POS_{i-2}, w_{i-1}, POS_{i-1}, w_{i+1}, POS_{i+1}, w_{i+2}, POS_{i+2}]$
- Would yield the following vector:
  [guitar, NN, and, CC, player, NN, stand, VB]

Naive Bayes Classifier

Because of the feature vector annotations we can use a Naive Bayes Classifier approach to WSD

This approach is based on the premise that:

Choosing the best sense $\hat{s}$ out of the set of possible senses S for a feature vector $\vec{f}$ amounts to choosing the most probable sense given that vector.

This is to say:

$$\hat{s} = \arg\max_{s \in S} P(s|\vec{f}) \tag{1}$$

Statistics difficulty

- Collecting reasonable statistics for above equation is difficult.

For example:

Consider that a binary bag of words vector defined over a vocabulary of 20 words would have

$$2^{20} = 1,048,576 \tag{2}$$

possible feature vectors.

To get around the problem

Equation 1 is Reformulated into the usual Bayesian manner:

$$\hat{s} = \arg\max_{s \in S} \frac{P(\vec{f}|s)P(s)}{P(\vec{f})} \tag{3}$$

- Data that associates specific $\vec{f}$ with each sense is sparse
- But information about individual feature-value pairs in the context of specific senses is available in a tagged training set

Assumption

- We **naively** assume that features are independed of one another and that features are **conditionally independent given the word sense**
- Yielding the following approximation for $P(\vec{f}|s)$:

$$P(\vec{f}|s) \approx \prod_{j=1}^{n} P(f_j|s) \tag{4}$$

- Probability of an entire vector given a sense can be estimated by the product of the probability of its individual features given that sense

Naive Bayes Classifier for WSD

- Since $P(\vec{f})$ is the same for all possible senses it does not affect the final ranking of senses
- Leaving us with the following formulation when we subtitute for $P(\vec{f}|s)$ in equation 3 above

$$\widehat{s} = \arg\max_{s \in S} P(s) \prod_{j=1}^{n} P(f_j|s) \tag{5}$$

Training a Naive Bayes Classifier

We can estimate each of the probabilities in equation 5 as shown below:

Prior probability of each sense $P(s)$

This probability is the sum of the instances of each sense of the word, i.e.:

$$P(s_i) = \frac{count(s_i, w_j)}{count(w_j)} \tag{6}$$

Individual feature probabilities $P(f_j|s)$

$$P(f_j|s) = \frac{count(f_j, s)}{count(s)} \tag{7}$$

Intuition of Naive Bayes Classifier for WSD

- Take a target word in context
- Extract the specified features e.g. neighbouring words, POS, position
- Compute $P(s)\prod_{j=1}^{n} P(f_j|s)$ for each sense
- Return the sense associated with the highest scores.

Conclusion

- We discussed the Naive Baye's classifier for WSD based on Baye's theorem and shown that it is possible to disambiguate word Senses in context
- But we have not discussed:
  - Evaluation of such systems, and
  - Disambiguation of phrases
  - To find out, come to my TabuDag presentation