Word Sense Disambiguation
LAB

Carlo Strapparava
FBK-Irst Istituto per la ricerca scientifica e tecnologica
I-38050 Povo, Trento, ITALY
strappa@fbk.eu

An Example of Sense Tagged Text

Bonnie and Clyde are two really famous criminals, I think they were **bank/1** robbers
My **bank/1** charges too much for an overdraft.
I went to the **bank/1** to deposit my check and get a new ATM card.
The University of Minnesota has an East and a West **Bank/2** campus right on the Mississippi River.
My grandfather planted his pole in the **bank/2** and got a great big catfish!
The **bank/2** is pretty muddy, I can't walk there.

Two Bags of Words (Co-occurrences in the "window of context")

FINANCIAL_BANK_**BAG**:

a an and are ATM Bonnie card charges check Clyde criminals deposit famous for get I much My new overdraft really robbers the they think to too two went were

RIVER_BANK_**BAG**:

a an and big campus cant catfish East got grandfather great has his I in is Minnesota Mississippi muddy My of on planted pole pretty right River The the there University walk West

Simple Supervised Approach

Given a sentence S containing "bank":

For each word W_i *in* S
    If W_i is in FINANCIAL_BANK_BAG *then*
        Sense_1 = Sense_1 + 1;
    If W_i is in RIVER_BANK_BAG *then*
        Sense_2 = Sense_2 + 1;

If Sense_1 > Sense_2 *then* print "Financial"
    else if Sense_2 > Sense_1 *then* print "River"
else print "Can't Decide";

General Supervised Methodology

- Create a sample of *training data* where a given *target word* is *manually annotated* with a sense from a *predetermined* set of possibilities
  - One tagged word per instance/lexical sample disambiguation
- Select a set of features with which to represent context.
  - co-occurrences, collocations, POS tags, verb-obj relations, etc...
- Convert *sense-tagged* training instances to feature vectors
- Apply a machine learning algorithm to induce a classifier
  - Form – structure or relation among features
  - Parameters – strength of feature interactions
- Convert a *held out* sample of *test data* into feature vectors
- Apply classifier to test instances to assign a sense tag

Naïve Bayes

- A premise: choosing the best sense for an input vector is choosing the most probable sense given that vector

$$\hat{s} = \arg\max_{s \in S} P(s \mid V)$$

$$\hat{s} = \arg\max_{s \in S} \frac{P(V \mid s)P(s)}{P(V)}$$

re-writing in the usual Bayesian manner

- But the data available that associates specific vectors with sense is too *sparse*
- What is largely available in the training set is information about *individual* feature-value pairs for a specific sense

Naïve Bayes (2)

- *Naïve assumption*: the features are independent

$$P(V \mid s) \approx \prod_{j=1}^{n} P(v_j \mid s)$$

$$\hat{s} = \arg\max_{s \in S} P(s) \prod_{j=1}^{n} P(v_j \mid s)$$

P(V) is the same for all possible senses ⇒it does not effect the final ranking of senses

- Training a naïve Bayes classifier consists of collecting the individual feature-value statistics wrt each sense of the target word in a sense-tagged training corpus
- In practice, considerations about *smoothing* apply

Naïve Bayesian Classifier

- Naïve Bayesian Classifier well known in Machine Learning community for good performance across a range of tasks (e.g., Domingos and Pazzani, 1997)

  …Word Sense Disambiguation is no exception
- Assumes *conditional independence* among features, given the sense of a word.
  - Parameters are estimated from training instances
- When applied to WSD, features are often "a bag of words" that come from the training data
  - Usually thousands of binary features that indicate if a word is present in the context of the target word (or not)

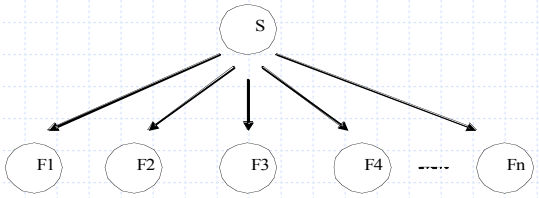
Bayesian Inference

$$p(S \mid F1, F2, F3, \ldots, Fn) = \frac{p(F1, F2, F3, \ldots, Fn \mid S) \times p(S)}{p(F1, F2, F3, \ldots, Fn)}$$

- Given observed features, what is most likely sense?
- Estimate probability of observed features given sense
- Estimate unconditional probability of sense
- Unconditional probability of features is a normalizing term, doesn't affect sense classification

Naïve Bayesian Model

Naïve assumption: the features are *independent*



$$P(F1, F2, \ldots, Fn \mid S) = p(F1 \mid S) \times p(F2 \mid S) \times \ldots \times p(Fn \mid S)$$

The Naïve Bayesian Classifier

$$sense = \underset{sense \in S}{\operatorname{argmax}}\, p(F1 \mid S) \times ... \times p(Fn \mid S) \times p(S)$$

- Given 2,000 instances of "bank", 1,500 for bank/1 (financial sense) and 500 for bank/2 (river sense)
  - P(S=1) = 1,500/2000 = .75
  - P(S=2) = 500/2,000 = .25
- Given "credit" occurs 200 times with bank/1 and 4 times with bank/2.
  - P(F1="credit") = 204/2000 = .102
  - P(F1="credit"|S=1) = 200/1,500 = .133
  - P(F1="credit"|S=2) = 4/500 = .008
- Given a test instance that has one feature "credit"
  - P(S=1|F1="credit") = .133*.75/.102 = .978
  - P(S=2|F1="credit") = .008*.25/.102 = .020

Comparative Results

- (Leacock, et. al. 1993) compared Naïve Bayes with a Neural Network and a Context Vector approach when disambiguating six senses of *line…*
- (Mooney, 1996) compared Naïve Bayes with a Neural Network, Decision Tree/List Learners, Disjunctive and Conjunctive Normal Form learners, and a perceptron when disambiguating six senses of *line*…
- (Pedersen, 1998) compared Naïve Bayes with Decision Tree, Rule Based Learner, Probabilistic Model, etc. when disambiguating *line* and 12 other words…
- … All found that Naïve Bayesian Classifier performed as well as any of the other methods!

WSD - applying Naïve Bayes

- The noun "bank" is ambiguous, as it means either *financial institution* or *sloping land (e.g. of a river)*, depending on the context where it occurs. Assume we want to disambiguate the meaning of the word "bank" using a Naïve Bayes algorithm. By analyzing a corpus of annotated examples, we obtain the following probabilities:

	Financial Institution	**Sloping Land**
P(c)	0.60	0.40
P(go | c)	0.12	0.1
P(deposit | c)	0.04	0.001
P(street | c)	0.05	0.03
P(check | c)	0.1	0.006
P(is | c)	0.13	0.1
P(money | c)	0.07	0.005

- Assuming that each open-class word in the context represents a feature, and assuming feature independence, calculate the most likely meaning for the word "bank" for the following test example.
- *I went to the bank to deposit my check and my money*

Sense annotation task

- Try to annotate with WordNet senses a general text
- On **clic3.cimec.unitn.it** under **`/mnt/data/annotate`** get the file *br-a01-untagged* (.doc or .txt)
- To access WordNet:
  - Connect via ssh to clic3.cimec.unitn.it (remember option -Y)
    - Be sure to have the path `/usr/local/WordNet-3.0/bin/` in your PATH variable
    - Call the command **wnb**
  - If you prefer a Web access try **`http://wordnet.princeton.edu/perl/webwn`**