CMPSCI 585
Programming Assignment 2

Out: February 17, 2004

Due: February 26, 2004

Word Sense Disambiguation using EM

Goal

The goal of this assignment is to implement the Expectation-Maximization (EM) algorithm to classify the sense of a word given contextual information. For example, the word "slight" can denote the adjective "insubstantial" or the verb "to snub." You are provided with instances of usage for each word sense, and your task is to build a model that will predict the word sense based on the context in which the word appears.

EM

If you are given labeled training data, then this task can be treated simply as a document classification problem (the *document* corresponds to the context in which the word appears, and the *label* is the word sense). However, because labeled data is hard to come by, and because we'd like to leverage as much data as possible, we would like to use algorithms that do not need labeled training data. EM is one such *unsupervised* learning algorithm that incrementally improves its probability estimates by interleaving expectation calculations (E-step) and likelihood maximization (M-step). See `http://citeseer.nj.nec.com/pedersen98knowledge.html` for an example of EM applied to word sense disambiguation.

Data

The data was collected from `http://www.itri.brighton.ac.uk/events/senseval/ARCHIVE/train.tar.gz`. For this assignment, use the slightly modified version found at `http://canberra.cs.umass.edu/~culotta/cs585/ass2-data.tgz` (1.2M). The data consists of 12 words to be disambiguated. See the README for more information. Each file is a text excerpt containing the word to be disambiguated. Note that the word to be disambiguated is marked with SGML

tags containing the sense-id. **Be sure you do not use the SGML tag as a feature.** This would be cheating, since these tags are known only at test time.

Tasks

- Since there are 12 words to be disambiguated, this requires 12 independent models. For each word, read in all the context files. Note that you do not have to split the data in to training and testing, since you are not using *any* label information.

- For each word, run your EM implementation to cluster word mentions into clusters with identical senses.

Additional Experiments

You should also compare your EM results with the supervised Naive Bayes classifier you implemented for the previous assignment. Use two training splits, one with .5 for training and the other with .75 for training.
Also, perform **1 of the following 2** experiments:

- **Effect of initial estimates**: EM is sensitive to the initial probability estimates (the parameter estimates before the first E-step). Using the data for only one word, choose 10 random initial estimates and report the mean and variance of (1) accuracy and (2) number of iterations until convergence.
- **Effect of context window:** It is intuitive that some context words are more important than others in predicting the correct sense of a target word. Your previous experiments included the entire context as a "bag-of-words," but words that are far away from the target word may be introducing noise into your model. To determine the effects of this, vary the size of context window of words used to create features in your model. For example, a window size of 3 only includes as features the target word plus the words that occur directly to the left and right of the target word. Implement this "windowing" technique using window sizes of 3, 11, and 31. Compare these results with the "bag-of-words" approach.

What to turn in

Code: Print out all source code written for the project.
Report: Write a 2 page report that includes the following:

1. Implementation experience - What questions and issues arose during your implementation. If there were difficulties, how did you resolve them?

2. Experimental results - Report the per-word accuracy, as well as the overall accuracy for all words. Report results for Naive Bayes, EM, and any variations you tried. Also include the confusion matrix for one word and provide some error analysis, indicating why you think the system made the errors it did.

3. Discussion - Discuss your experiments and explain your results. What additional experiments do you think would improve performance?

Suggested Time-line

- Tuesday, February 17. Assignment given.
- Friday, February 20. Implementation finished, baseline experiments in progress.
- Monday, February 23. 1 or more additional experiments in progress.
- Thursday, February 26. Report written, code printed, hand it in.