Determining Gender through Information Clustering Techniques with the Enron Email Corpus

Patrick Trinkle
tri1@umbc.edu

Mike Corbin
corbin2@umbc.edu

CS691A
Final Project
5/12/2009

1. Summary

The goal of our project was to see if we could determine the gender of an author based on the frequency of the words they use. Successful studies have been done using word frequency to determine authorship on corpus ranging from the Bible to The Wizard of Oz books. We applied these techniques, in parallel, to the Enron email corpus and graphed the results.

2. Code

We first used python to help us get a better understanding of the Enron corpus. In python we counted the term frequencies and recorded the documents. We then copied the emails of the top 100 users with the most emails to bluegrit. On bluegrit we processed all the users with mpi using a client server model. We required a least 3 process for this model, 1 for the server, 1 for the print server, and the request as clients. The server would read though the directory of users and pass a path to each client as it requested work. Each client would then process all the emails in the path it was sent, pass its results to the print server, and then request more work. Once the server was out of folders it would terminate each client the next time it requested for more work. Once all the clients were terminated it would inform the print server that it could process all the data it received. The print server would then process the data and write the results to a file.

The code is broken up into 3 files.

tokenizer.h - header file which describes the structures, and stop words.
tokenizer.c - main file which contains the server and all the clients.
pca.c - print server which performs the principal component analysis and outputs a file

The pca.c code was originally written by F. Murtagh. We modified the code to use MPI to accept the matrix instead of reading it in from a file. We also redirected its output to a file instead of printing it to the screen.

3. Results

Our results did not define the gender of the users as clearly as we had hoped mainly because we did not have enough emails for most users. We were able to parallelize the code in a way that would accept any number of users and emails. Our code was also able to speed up the process for around 80 seconds for 3 processors to around 11 seconds for around 20 processors.

4. Stop Words

Below are the stop words we used.

the, to, and, of, a, in, for, on, that, this,
from, with, it, by, as, at, or, if, not, please,
an, any, original, all, power, but, thanks, energy, know, said,
mail, up, gas, new, about, time, what, so, there, which,
out, no, company, need, get, one, let, should, also, more