Web Extraction & Information Retrieval
Project Update

Kai Hayashi, Cary Lee, & Daniel Myers

Introduction:

Twitter contains a vast recorded collection of human interaction. With its easy data scraping and open nature, it represents a treasure trove of information for the well-equipped scientist. Recently, Twitter added a feature allowing users to keep track of their location via their phone's GPS coordinates. We are interested in the following question: "Can someone determine a tweeter's location based off of the content of the tweet alone?"

This question has been a hot topic in the news as of late, with reported robberies and many concerns over the public data available in tweets. The GPS information combined with Twitter's large user base, may provide enough data to build a distribution of user locations given the tweet text.

In this report we present our method of gathering Twitter data and provide some preliminary results of a very basic attempt at solving this problem. We utilize WEKA and the Twitter API to collect and process tweets. We represent the tweets as a word vector and run Naive Bayes with this representation.

Data:

We collected that data by scraping twitter once an hour for a week. We wrote a short scraping script that collects tweets from the public timeline, collects the friends of those users, and then grabs 200 most recent tweets from each user in this collection.

Data Statistics:

After scraping every hour for a week we have a database with just under 2 million tweets. While this may seem like a lot, we are actually only interested in the tweets containing the "place id" field which contains the location in which the tweet was

tweeted. Currently about one percent of all tweets in the database have this information thus we have about 20,000 tweets in which we can use with WEKA.

Preliminary Results:

The results given here were obtained via a Naive Bayes algorithm run on a small sub-set of the data we collected. This subset consisted of 315 tweets originating from 3 different cities/provinces (Kentucky, Gennep (Netherlands), Caieiras (Brazil)). We utilize the "bag-o-words" representation. This small number of classes and the fact that each has a different language produces excellent results.

Based off of a 10-fold cross-validation scheme we obtain a 93.2 % precision accuracy and a 93.4 % recall accuracy.

We were unable to successfully run the Naive Bayes classification on all our data due to the extreme memory requirements that that entails (we tried on machines with up to 5GB of RAM while running java with an extended Heap). We predict that our scores will be much less impressive.

Next Steps:

Our top priority right now is to figure out how to either optimize our representation so that it will run on a machine, or to find a machine that will handle building the model. We estimate that we will need a machine with somewhere in between 12-18 GB of memory.

After that task is done we would like to add geography somehow into the algorithm. Currently, the place names just represent classes and as such have no other information what that name means. WE would like to rather use the coordinates of the tweets to build a 2-d regression.

In addition to this previous idea, we would like to slim down our word vector representation by only utilizing words that will indicate something about location. This could be obtained, or approximated, through a number of different methods. The first of which could be to exclude common words (e.g. the, a, an, him, her) from the word vector. We could also use one of the many techniques discussed in the papers we've read for class to classify the words in the vector, and then choose only the ones relevant to location.