Twitter Dedupe

- Sravan Bhamidipati (sb3400), PID8

Proposal

Brief Re-introduction
Twitter Dedupe is a low-latency clusterer that automatically divides any stream of tweets into different clusters based on their content.

Data
I used AllMyTweets to collect tweets of large news feeds like NYTimes, CNN, IMDB, ESPN, etc. Their HTML outputs are in turn parsed by my script to get the text of the tweets, which become the raw data of my clusterer. I hand-picked the feeds so that their categories can be well-defined, which would be helpful to verify the accuracy of clustering. I have collected about 14000 tweets in this way, and they are sufficient for my initial development and testing.

I also wrote my own tweet-collection script using the Twython API. The script can crawl all tweets in an authenticated user's timeline (friends' tweets), or crawl all tweets posted by a specific user. This is experimental, and intended to evolve into the backend of my AppEngine app.

Data Point
Each data point is a tweet, which is a short collection of words with one or more sentences (<= 140 characters). I normalize a tweet in a trivial manner: discard non-English tweets, discard non-ascii characters (tweets are in unicode) and punctuation, convert to lowercase, split around white-space, ignore a common list of words, and save the tweet as a set of words, with URL as additional metadata.

Also available is the tweet metadata like timestamp, location, retweet and favorite counts, replies, and the source (user) metadata. I believe some of this metadata can be used as additional "features" for an advanced clusterer. e.g. Given that several of the past tweets from a user belonged to a particular cluster C1, a new tweet may be grouped into C1. Given that several tweets from a geographic location belonged to a particular cluster C2, a new tweets from the same geographic location may be grouped into C2.

Assumptions
I have decided to ignore tweets from non-English users, to maintain a list of ignorable words instead of POS tagging for greater speed, to show at most 100 latest tweets when a user logs in. (When a user first logs in, the Twitter website shows the latest 20 tweets, and their TweetDeck app loads the latest 50 tweets.) Because of this, the process of creating a dictionary of unique words, word counts, word counts in each tweet is fast.

So far, using the word frequencies to cluster tweets has shown decent accuracy, but is resulting in a very small number of tweets being clustered. (The rest of the tweets, belonging to no cluster in particular, are kept as one large unclassified cluster.) For this reason, I am exploring the option of classification instead of clustering, to see whether news feeds of different news categories can be used to train a classifier which can classify tweets into one of a fixed number of categories: politics, sports, entertainment, others.

Application

I am implementing the AppEngine app in Python. I have the basic skeletal and have figured out OAuth as well as the Twitter Streaming API. (The app still runs into exceptions, but those are bugs which will be hardened later.) I am using a single user for my internal experiments, but I plan to extend the app to support multiple users logging in.

Repository Snapshot
I uploaded a recent snapshot of the project repo to cs699804.cs.columbia.edu in path “/home/sb3400/twitter-dedupe” with read permissions for everybody.