**Automating Privacy Settings For Social Networks:**

# Content Classification of Public Tweets

Progress Report - Spring 2011

Muhammad Ali Akbar (UNI: maa2206)
Supervision: Maritza Johnson & Dr. Steven Bellovin
maa2206@columbia.edu

Computer Science Department,
Fu Foundation School of Engineering & Applied Sciences,
Columbia University, NY

## 1 Introduction

The social networks have taken over the Internet world. According to Wikipedia, there are at least 13 online virtual communities with more than 100 million users [1]. This is a staggering number, given that the phenomenon of social networking is still relatively young. The primary reason for the success of the social networks is that they provide users with the ability to share their thoughts, profiles, photos and videos in a very effective way. This user-content driven model has enticed users to open up their life to anyone who can see the content they share. With more and more personal data on the web, the question of privacy has become very important: Is the content shared by a user actually visible only to the 'friends' for whom it was intended?

Large social networks like Facebook[1] provide users with options to specify access control for the content they share. Although this is a very important aspect of sharing data online, the effectiveness of these access control systems is questionable. A recent study done by Maritza and Steven measured the effectiveness of these access control systems through an empirical experiment. They asked a sample of Facebook users questions related to the privacy of content they have shared. The study found that every user had at least one piece of content with access control settings different from the intention of the user.

[1] http://www.facebook.com/

In this scenario, the aim of this project is to come up with an automated system for setting access control lists (aka privacy settings) for the user content. The project currently focuses on the text content shared by the users. For text based content, the text status updates known as 'tweets' (on the social network Twitter[2]) are a fairly accurate representation of the text content shared by users on most networks. In fact, platforms like Facebook allow a user to automatically set their tweets as their status updates. We formulate the problem statement as:
***Given a 'tweet' shared by a user, determine the appropriate class of access control settings for this content.***

## 2 Dataset

The first step towards the project is data collection. We are using public data[3] for our experiments. The steps used to generate the dataset for the experiments are described below.

### 2.1 Collection

We use Twitter API [2] for gathering public tweets from the Twitter platform. The Twitter API are quite comprehensive and provide a comprehensive amount of metadata related to the tweets. We use the JTwitter library [3] for accessing the Twitter API through Java. A custom Java daemon is written that collected public tweets from twitter created within last 15 minutes, and containing words from a large pool of keywords selected to cover 12 different categories of content topics. The daemon is deployed on a Linux server, scheduled to run once in every 15 minute period using `crontab`.

The following information is collected about each tweet.

1. Unique user identifier aka username
2. Text of Tweet
3. Time of Tweet

[2] `http://www.twitter.com/`
[3] We define public data from twitter as 'tweets that are set to be publicly visible by the user.'

### 2.2 Anonymization

Every collected tweet is anonymized before storage. The anonymization is done to ensure privacy of user by stripping off identifiers that connect the tweet to the specific user. However, we replace these identifiers with an encrypted hash of the actual value so that it remains possible to identify tweets from/about same user. We perform anonymization on both the username and the tags of usernames found in the text of the tweet.

### 2.3 Storage

The collected tweets are stored in a `mysql` database. The database is stored on a protected system with user level access to the database only from the local machine.

### 2.4 Labeling

We have created a `PHP+Apache` based web interface that shows random uncategorized tweets from the database to an operator and allow him/her to label it with one of the categories. The anonymized tags in the tweet are replaced with gender-neutral first names. The web interface also allows the administrator to see the statistics of the collected tweets in the database.

The actual labeling of the data has not been performed yet, and is a task for future work on the project.

## 3 Content Classification Methodology

### 3.1 Training phase

In the training phase, we perform the following steps for the tweets collected in each category:

1. for each *category* do
   (a) Initialize the n-gram data structure
   (b) Retrieve tweets for that category
   (c) for each *tweet* $\epsilon$ *category* do
      i. Preprocessing of tweet

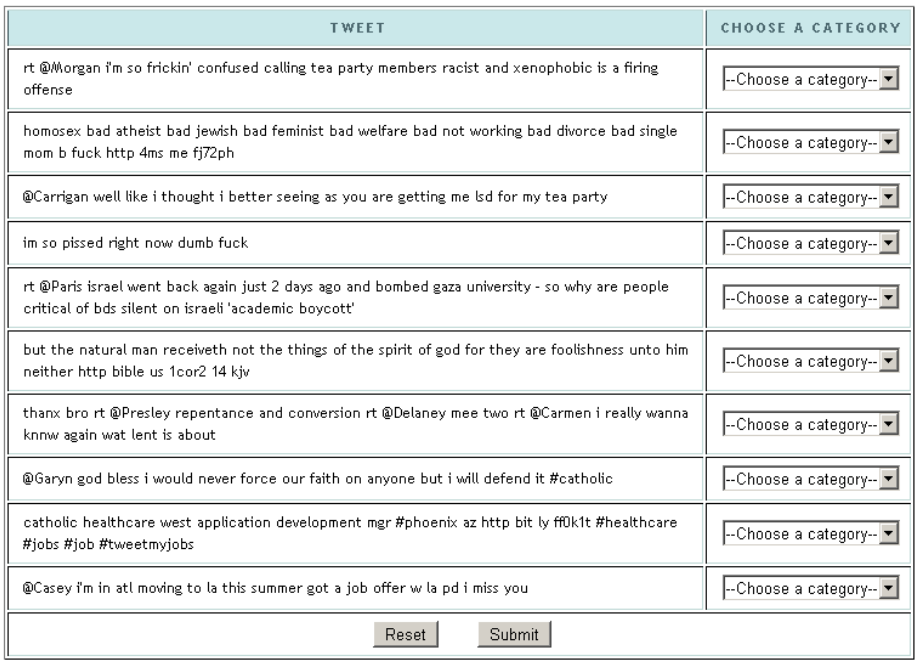


**Fig. 1.** Web Interface for Labeling of Tweets

ii. Keyword extraction
iii. Insertion of Keywords in the n-gram data structure

(d) Extracting high frequency keywords from each category and insert in to the model for that category

**Pre-processing & Keywords extraction from Tweets** The pre-processing of the tweets consists of following steps:

1. **Removal of punctuation and whitespace:** We remove the punctuation and whitespace and replace them with a single space character.
2. **Tokenization:** We tokenize the words (separated by space character) in the string.
3. **Expansion of abbreviations and Twitter acronyms:** We expand the common abbreviations and acronyms used in twitter to words.

4. **Stemming:** We try to find the stem of all words. Stemming is a process that reduces a word to a prefix from which it is derived. For example, the stem of *laughing* is *laugh*. We use Porter's algorithm for suffix stripping as a stemming algorithm [4].
5. **Removal of Stop words**: We remove the most common words. We use a comprehensive list of the common words which are typically stripped off by search engines. These common words have high frequency and have no relation to the topic of tweet, so removing them helps to keep the final model relevant.

In the end, the words left in the tweet are extracted as keywords, and fed in to the ngram datastructure as grams.

### 3.2 Testing phase

1. for each *tweet* do
   (a) Preprocessing of the tweet
   (b) Keyword extraction
   (c) Classification based on keywords and the model generated in training phase

The preprocessing of the tweets and keyword extraction involves the same steps as in training phase. The tweets are then classified by computing a bayesian score for the keywords in the tweet and the high frequency keywords in the models for each category. The tweet is classified as belonging to the category for which it has maximum score. This step hasn't been implemented yet, as we need labeled data first. It is part of the future work for the project.

## 4 Current Progress

Currently we have more than 1.2 million distinct (unique) anonymized tweets in our database, collected over a period of three weeks. We have tested the algorithm on these tweets to successfully extract convincing high frequency keywords for each category. However, we need labeled data for designing and testing the classification algorithm properly.

## 5 Future Work

The following tasks are the next steps for this project.

1. Select 10,000 tweets.
2. Get labeling of tweets done through Amazon's Mechanical Turk[4].
3. Use labeled tweets to compute the training model for each category.
4. Design and optimize the classification strategy, incorporating the NLP algorithms and tweaking them for good accuracy at a very small sized input corpus (maximum 140 characters tweet).
5. Test Unions & Intersections of keywords from different categories and corresponding Unions or Intersections of access control policies for better accuracy when a tweet may map to multiple categories.
6. Test the effectiveness by translating this approach to Facebook statuses. Specifically, convert the final algorithm to a Facebook application that a user can install and then set statuses through it. Empirically evaluate the accuracy of the approach in this case by getting user's feedback.

## References


1. Wikipedia: List of virtual communities with more than 100 million users `http://en.wikipedia.org/wiki/List_of_virtual_communities_with_more_than_100_million_users`.
2. Twitter: Twitter API Wiki `http://apiwiki.twitter.com/`.
3. WinterWell Associates: JTwitter: The Java Library for Twitter API `http://www.winterwell.com/software/jtwitter.php`.
4. Porter, M.: The porter stemming algorithm. (2009)


[4] `http://aws.amazon.com/mturk/`