Classifying Metatweets

F. Yo-Shang Cheng
i256: Applied Natural Language Processing
Final Project
yoshang@gmail.com

Abstract

Spam classification has been a very useful tool as people's reliance on email has increased. The same sort of analysis could be applied toward micro-blogging services such as Twitter. A more subtle type of spam, metatweets, exist that don't provide much, if any, content at all. The aim of this project was to filter these metatweets with a Naive Classifier that relied on both structural and lexical features of tweets.

1 Introduction

Twitter is a rapidly growing, popular social networking and micro-blogging service that allows its users to send short, 140 character messages, or "tweets" to their friends and the internet-community at large. Recently it has been leveraged as an effective medium for political campaigns and public relations, and as an outlet for immediate breaking news.

Through a undocumented and somewhat mysterious process, Twitter keeps tracks of and displays "trending topics" on its service. These are the most tweeted terms at the moment, often covering news, sports, television and jokes. A hashtag (#) is often appended to these terms in order to make them searchable and to specifically identify a term as a topic of interest. On its rules page, Twitter lists a number of behaviors that can result in a tweet being classified as spam, which may then lead to the suspension of an account. These include *"Tweeting about each trending topic in turn in order to drive traffic to your profile, especially when mixed with advertising."* and *"Repeatedly tweeting the same topic/hashtag without adding value to the conversation in an attempt to get the topic trending/trending higher."*

Despite Twitter's warnings, many tweets can be found that bend or violate these rules about tweeting actual content. This is particularly true of topics that have been trending for a while. As documented somewhat jokingly by blogger Meg Pickard, given enough time, the public discourse on a particular trend or meme shifts from actually discussing the topic, toward discussing the fact that other people are discussing the topic. People entering the discussion late can only contribute a *"What is this $\#something$ that everyone is talking about?"*, people who have been in on the discussion from the beginning may pipe in with a *"Yay! $\#something$ is a trending topic!"* On the other hand, people who have been observing the trend and have moved on may interject in with *"Why are people still talking about $\#something$?"*. Ironically, these metatweets about other tweets keep feeding and strengthening the popularity of a trending topic despite not actually furthering the discourse. As Twitter has explained on its site, *"The most important thing is to make sure your tweets are genuine thoughts or impressions and not just attempts to get attention by inserting yourself into a trend. When you click on the trending topics, we would like you to see real people's ideas and links to further relevant information, not spam and people begging for follows."*

The goal of this project was to build a classifier to distinguish between contentful tweets and metatweets which don't contribute to the public con-

versation.

2 Related Work

The classification and filtering of spam is a well-researched area of NLP. In particular, Bayesian filtering models have proved to be very effective in distinguishing meaningless spam from legitimate emails. Spam filtering was a track at the Text REtrieval Conference from 2005 to 2007, with corpora of spam messages being published for training and evaluation purposes. Noted computer scientist Paul Graham advocated the use and efficacy of Bayesian spam filtering in his essays *A Plan for Spam* (2002) and *Better Bayesian Filtering* (2003). Twitter itself has done work on removing spam and blocking spammers, but these mainly focus on isolating tweets with links redirecting users to marketing websites.

As Twitter has only recently gained popularity, not much work has been done using actual Twitter data. As of yet, no one has amassed and published a corpus of tweets. However, as new Twitter data is constantly being produced and the API is relatively easy to use, many experimenters choose to simply collect their own data as they see fit. *Twitter Sentiment*[1] is an ongoing project that emerged from a class project at Stanford by Go, Huang and Bhayani (2009). Their goal was to build a model to classify the sentiment of Twitter messages into three categories: positive, negative or neutral. To collect their training data, the team searched for tweets with the emoticons ':)' and ':(' to use as positive and negative examples, respectively. This extraction allowed them to collect a very large amount of automatically labeled data. Their training set contained hundreds of thousands of tweets, and used a few hundred thousand features. In their experiments, a Maximum Entropy model outperformed the classification by both a Naive Bayes and Support Vector Machine model.

3 Data and Features

A distinguishing characteristic of Twitter is how short its messages are. Because tweets are limited to 140 characters (including punctuation and spacing), it is difficult to do any deep processing on them. Furthermore, to work around this limitation, many users rely on nonstandard spellings and abbreviations in order to fit as much text into a single tweet as possible. As mentioned above, the hashtag is often used to indicate that a specific term in a tweet should be identified as a topic. Other conventions include the "@" symbol, which indicates a username and the abbreviation "RT", which stands for re-tweet. Retweets are used to share content that someone else has produced, often with the user's own commentary added.

Here is a sampling of three tweets with the hashtag $\#Smallville$:

1. Loved seeing some exterior locations like the farm again in $\#Smallville$ tonight.

2. $\#Smallville$ is moving up in the trending topics! Push it all the way!

3. New $\#Smallville$ tonight!! New $\#Smallville$ tonight!! $\#TGIF$ New $\#Smallville$ tonight!! New $\#Smallville$ tonight!! New $\#Smallville$ tonight!!

Example (1) falls squarely in the contentful category, it expresses the thoughts of its author about what she just saw on the TV show *Smallville*. Thankfully the majority of tweets on Twitter look like this. On the other hand, (2) is a metatweet, it merely comments on the fact that trend is growing in popularity, with nothing to do about *Smallville* itself. The author's purpose is simply to keep the trend alive, and to push the term *Smallville* up the trending topics list. Finally, (3) is a metatweet as well, but lacks the grammatical structure and thought of (2). It instead just repeats the hashtag multiple times.

Data for this project was collected from two sources. The first was a corpus of 46,246 hashtag-marked tweets, collected by classmates Abe Coffman, Karen Nomorosa and Nate Gandomi for a project in another class. Some processing work had to be done with these as a significant number of the tweets were written in languages other than English. Simply filtering out any tweet with non-ASCII characters left the corpus with 41,653 mostly English tweets. Next a metatweet corpus was assembled by extracting any tweet with the term "trend" (and any of its derivatives) in it, the remaining tweets constituted the contentful corpus.

[1] `http://twittersentiment.appspot.com/`

The second source of data for metatweets was assembled by using the Twitter API to collect tweets with the term "trending" in it. This was done over the course of a few days to balance the topics of the tweets (each call to the API resulted in a batch of 100 tweets, most of which were about whatever was trending at the time). These were quickly looked over by hand to remove anything not in English, and resulted in an additional 8,828 metatweets.

After some testing on a toy data set (around 600 tweets), it became clear that the classifier was missing metatweets like example (3), where a term is repeated multiple times in order to boost it on the trending topics list, but the word "trending" does not actually appear. The metatweet corpus was then further augmented by moving any tweet with more than two hashtags in it from the contentful corpus into the metatweet corpus. The final corpus was composed of 38,953 content tweets and 11,522 meta tweets. This corpus was further divided into training, devtest and test sets, split 81%/9%/10%, with the same proportion of content and metatweets as the overall corpus. Specifically, 40,883 tweets in the training set (31,235 contentful, 9,648 meta), 4,543 tweets in the devtest set (3,584 contentful, 959 meta) and 5,050 tweets in the test set (4,134 contentful, 915 meta)

The initial set of features extracted from the tweets included a mixture of structural and lexical features. The more structural features consisted of the length of the tweet, number of words (tokenized on whitespace), lexical diversity, and number of repeated words (a raw count version of lexical diversity). The following characters and sequences were counted and used as features: hashtags, URLs (anything starting with the sequence *http://*), @ symbols, exclamation points, question marks, emoticons, *RT*, *ha(ha)+*, *lol(ol)** (both *haha* and *lol* occasionally show up in longer forms like *hahahah* and *lololol*, so a regular expression was used to capture these variants), words spelled entirely in capital letters (greater than three characters long, as *RT* shouldn't be double counted), and the use of any other nonalphanumeric characters.

Specific words were also used as features. A list of the most common words were extracted from the metatweet corpus and a list of the top 30 were used as features. This list was chosen by hand, in addition to stopwords that should be ignored, a lot of terms appeared at the top of the list because of the content of the tweets. For example, the words *new*, *moon*, and *smallville* occurred very frequently, but this simply an artifact of the data, which featured many tweets about the television show *Smallville* and the movie *New Moon*. These words aren't particular to the metatweet corpus, they occur very frequently in the contentful tweet corpus as well. Because of the way the data was automatically labeled, *trending* and *topic* were by far the most frequent words. Other words used as features included *twitter*, *why*, *start*, *still*, *top*, *help*, *love* and *sucks*.

Experimenting with the devtest set led to the refinement of the features being used by the classifier. Neither length nor the presence of URLs is a good discriminative feature, as the length of both types vary and they both consistently feature links. Both of these features were removed in later experiments. Interestingly, the number of words was a mildly useful feature, tweets with very few words or many words tended to be more contentful. A possible explanation for this is that shorter messages have less room to repeat the trending topic term. Contentful messages also use short stop words (*the*, *a*, etc.) not present in many metatweets, these increase the number of words without increasing the overall length of the tweet by very much.

An important decision to make was whether to use raw counts as a feature or to bucket the values and turn them into binary features. Lexical diversity was a very difficult feature in this regard. Very high values (greater than 2.0) were clear indicators for metatweets. It was less clear with lower values, and it seemed that overfitting was occurring. A lexical diversity of 1.217 was a good indicator for a metatweet, while a lexical diversity of 1.231 was the reverse, being an indicator for content. This type of behavior suggested that lexical diversity should be turned into a binary feature, as it definitely had some value, but was being somewhat abused in certain value ranges. In the end, the cutoff was set at the feature being true if the lexical diversity were greater than 1.1. Although not perfect, this value seemed to maximize the accuracy empirically. The same sort of question was posed by the number of exclamation points, emoticons, etc. but the situation was a bit cleaner since these involved integer values.

4 Models and Results

Both a Naive Bayes and Maximum Entropy classifier were tested (using NLTK's implementation), with the Naive Bayes model being used in the end. The Maximum Entropy model didn't improve upon the accuracy of the Naive Bayes and it took longer to run. This MaxEnt classifier's speed issues mainly had to do with the hill-climbing algorithm used in NLTK, GIS, which runs very slowly. Unfortunately, there were some package conflicts which prevented the use of another, faster hill-climbing algorithm.

The classifier using the full set of features achieved a 98% accuracy on the test set, and 95% without the use of the *trending* lexical feature. There was a lot of hesitancy in including the *trending* feature, as the majority of the training data was assembled around using that term to automatically label and categorize the data. Indeed, the baseline accuracy rate from *only* using the *trending* feature was around 94%. It would be good to go back and better hand construct the test set. It is difficult to think of another way to automatically extract a large set of metatweets for training purposes without introducing significant artifacts in the data. In addition to this issue, metatweet data can be quite sparse. Although it seems to be true that metatweets emerge in long-living trends, many topics die out before they reach "trending" status. Being able to predict which trends will last long enough for metatweets to emerge is a difficult, unpredictable and entirely separate task, one that would no doubt be very lucrative. The most resilient trending topics seem to be sensational news stories, such as celebrity deaths and the Balloon Boy hoax. Clearly there's no way to know when these will enter the Twitter community's discussion.

Regardless of some of the faults with the data, the model still revealed some interesting results. Among the few errors that the model made, there were three categories of tweets that the classifier would misclassify as being metatweets when they (arguably) possess content:

4. Saw about 20 people holding up Jerry Denham 4 $\#Congress$ signs in $\#Livermore$ - $\#cd10$ $\#ca10$ $\#SpecialElection$ $\#yal$ $\#c41$ $\#tlot$

5. I love love LOVE Patrick Dempsey!!! And I feel SO bad for Karev! I swear! :(:(:($\#GreysAnatomy$

6. If trending topics on here are indicative of the state of our world, I might need to find a more intelligent one to inhabit.

Some people just really enjoy using hashtags, perhaps for their own personal organizational scheme, or perhaps as attention-seeking behavior. Tweet (4) is a good example of this, as it seems that the author actually wants to share some interesting election information, but has attached eight different hashtags to his tweet. The hashtag feature could be revised to distinguish between distinct hashtag terms, but there already is another feature counting the number of repeated words.

Another type of tweet that gets misclassified are those typified by teenage girls, like in (5). These simply possess many features that typically show up in metatweets: many repeated words, exclamation points, emoticons and words in all capital letters. Some people might argue that there really isn't much content to this, but as the categories have been established, this should be classified as being contentful.

Finally there's the question of (6), or meta-metatweets. Here the tweet is commenting on the concept of trending topics in general, not a particular one. This clearly has more to say than tweets like (2) or (3). However, the strength of the *trending* and *topics* features places this in the meta category. Maybe these should constitute their own, third category, but in any case they are quite different from the majority of other metatweets. It should be noted that the categories typified by examples (5)-(7) do not occur that often, relative to the majority of the data encountered. However, this is not a reason to simply ignore them.

5 Conclusions

As Twitter grows, the number of people trying to take advantage of the service (sometimes violated the terms of use) will no doubt grow as well. Spam is already an issue being investigated by the Twitter team, but increasingly there must be a way to filter out the noise from the actual content. This more subtle type of spam filtering may prove to be very useful. Again, despite some issues with the training data, this was a very useful and enlightening project.

Looking beyond your own personal network and examining how others are using the exact same service revealed many interesting user behaviors, conventions and styles. Twitter is an interesting avenue for NLP and machine learning research, as the data is plentiful, easy to collect and constantly being produced. Well established models should do well on Twitter data, but attention must be paid to their particular features.

References

Paul Graham. 2002. *A Plan for Spam*. http://www.paulgraham.com/spam.html

Paul Graham. 2003. *Better Bayesian Filtering*. http://www.paulgraham.com/better.html

Alec Go, Lei Huang, and Richa Bhayani. 2009. *Twitter Sentiment Analysis* Final project report for Stanford CS224N, http://nlp.stanford.edu/courses/cs224n/2009/fp/3.pdf.

Meg Pickard. 2009. *Twitter Trending analysis.* http://meish.org/2009/05/17/twitter-trending-analysis/.