Impactweets:
Methods and Techniques and for Finding Interesting Tweets

**Team members and roles**:

* Seema Puthyapurayil :: Visionary, Network Analyst
* Ignacio Pérez :: Developer, Classifiers and Content Analyst
* Corey Hyllested :: Developer, Data Analyst
* Yao Yue :: Twitter Project Mentor
* Marti Hearst :: Instructor

**Abstract:**
Today's dominant social networks and platforms are creating volumes of data that outstrip any individual's ability to consume it.  As individuals follow more persons of interest (friends, family, celebrities, etc) the amount of data s/he must sift through increases.  This preponderance of data means consumers may never be bored again, but it may be an overwhelming experience.  We will try to find the big picture of tweets for a particular user and attempt new methods of filtering interesting tweets for individual users using the characteristics like RT, Mentions, Topics, URLs and categorization of tweets.

**Overview of Project Strategy:**
This research project is about how to increase the signal to noise ratio.  A conservative filtering approach may be ineffective at reducing the noise and an aggressive approach seems fraught with issues of dropped tweets.  We believe users would prefer a report that draws attention to tweets-of-interest, or impactweets.  This could be provided periodic report providing impactweets to users.  He or she could revisit these tweets which were lost in the shuffle, or discover new ones.  We created a several groupings of tweets that may impact users: insightful, exploratory, sentimental, sharable, fun, and serendipitous.  While it's intuitive why these would have a positive impact, its not obvious how to find them.

To begin approaching such a large and open requirement, we thought it would be useful to investigate three facets: conversations, topic and link analysis, and tweet categorization.  Topic and link analysis is an attempt to find links and topics (#hashtags) of prominence in the network.  We also wanted to find important conversations in a persons network and find a way to rank those conversations.  Categorization of tweets in an individual’s timeline would provide allow users to sift through his or her categories of relevance quickly.  Finally, we wanted to see all this information come together in one report.

**Research and Methodology**
Many companies are looking at "big data" as the new oil.  The analogy is apt.  The mined extract is crude something that requires refining.  Our infrastructure extracts raw JSON from Twitter's sprinkler hose and user timelines.  After extracting non-English tweets, non-parsable Unicode, and newlines it is parsed with a Python application into more easily parsable flatfiles.

Each facet required different data and slightly different infrastructure.  Tweet categorization, for example, required training sets that were acquired through Twitter's trending topic API.  Topic, link, and conversation analysis used an individual’s ego-network (friend-of-friends) and concentrated on refining value out of the prominence to conversations, relationships, topics and links in an individual's ego-network.  We used the format below.  The format changed as requirements were added.  This placed strain on our ability to generate report on-the-fly.
**TweetID user RT hashtags links mentions in\_reply\_to date tweet.**

We describe the research, methods, and results of each facet in greater detail below.

**Naive Bayes Machine Learning.**
Each user brings his or her own interests, experiences, and events to share on Twitter.  Users may not always share information s/he is interested in consuming, but its difficult to see how that would be a long-standing pattern.  We had the idea to classify out-going tweets and recommend incoming tweets in the same categories ratios.

We wanted to provide a ML (machine-learning) tool that could categorize tweets.  Despite the naive approach, it works well.  We see this as a method, that with more signals, such as conversations, links, topics, and RT weights would provide a more holistic idea of interesting tweets.

In classifying tweets, there are three phases:

* + - 1. Retrieval:  https://github.com/iaperez/ABDTProject/tree/master/retrieval

Twitter’s Search API was used to download tweets related to certain keywords.

The format of the resulting types is: (TweetID, topic, TweetText)

Reference for the classified tweets:

<https://raw.github.com/iaperez/ABDTProject/master/data/alltweets>

* + - 1. Build a base model: <https://github.com/iaperez/ABDTProject/blob/master/pigfiles/BaseModel.pig>

In the context of build a base model, and considering as input a set of classified tweets, we need to calculate for each ngram its relation with each possible topic. In Pig, we implemented this:

* + - * + Counting all tweets used as input. (ctTweet)
				+ Count every topic and tweets related to specific topic. (ctTweetTopic)
				+ Count each n-gram associated with a particular Topic. (ctNgramTopic)
				+ Count the tweets that contain a particular n-gram. (ctNgramTweet)
				+ Estimate probability an n-gram is referencing a particular topic (based on “A plan for SPAM**”** article)

      3.  Analyze new tweets using base model.

<https://github.com/iaperez/ABDTProject/blob/master/pigfiles/tagsAnalisys.pig>

We use the model proposed in the literature (“A plan for SPAM” article) to calculate the probability for each incoming tweet to be related with a certain topic: every tweet will be analyzed in terms of its n-grams, and the total probability of that tweet to be related to a particular topic is:

         

With ProbNgramTopic as the probability of that n-gram to be related with that topic (base model). When an n-gram is not founded in the base model, we are using a probability of 0.4.

Example:

<https://github.com/iaperez/ABDTProject/blob/master/data/twitterclassificationresults.txt>

       **Results:**

The initial results of applying this technique in a particular set of 200 manually classified tweets were really surprising: Using 60000 tweets classified by the twitter search, around 70 percent of the tweets were well classified. After this, we increment the base model to 80,000 tweets using the words that were present in the target dataset and around 90% of the tweets were well classified using a 90% threshold for each topic. One interesting result here is that a lot of topics were related, like business and social media. The human classification for some of that topics was specific for business, but after receiving the feedback of our classifier, part of the tweets related with social media were also using business jargon.

But the results classifying conversations is not as impressive. Conversations tend to be unfocused, so the effectiveness of the classification was much lower.  In the final dataset, we included all possible classifications for that conversation. In general, 60% percent of the conversation analyzed also included categories that were unrelated to the conversation, but the important thing was that conversation were also classified under the correct topic.

There are several ways to solve this problem:

* 1. Changing the technique: LDA has been proved as a better technique for text classification. Naive Bayes is a really simple technique used for experimental purposes, and its expected to have problems like those described.
	2. Increase the base model size.  The  base model was small.  But considering the focus of the project, we decide to focus on other ideas.

More information for this analysis here:

<https://github.com/iaperez/ABDTProject/tree/master/data>

**Conversations.**
What if Twitter was able to let you know about conversations your friends are having?

Twitter has multiple ways to 'message' other users.  A user can retweet by broadcasting the exact tweet to followers.  A user could old-school retweet, which allows for a response.  Alternately, users could @reply, which typically provides in\_response\_to metadata.  Lastly, an @reply might not include the metadata for a variety of reasons (client support, or not responding in-line).  The impact of conversations in your network can be immense.  Drawing attention to conversations in the network can provide closeness and awareness of the interactions of people of interest.

We began by trying to find the conversations that were happening within your ego-network, classifying them using the Naive Bayes Classifier and then ranking those conversations by using the KloutScore of each of the users participating in the conversation.

Why rank by influence score?
When we spoke to users, we found that people love to talk to influential people. People were most excited about the possibility of having conversations with influential people on Twitter. We wanted a way to tap the conversations that people were having with the influencers in Twitter.

Why KloutScore?
We tried to create an influence score using the user statistics in twitter, the retweet count, favorite count, the ratio of friends to followers, recursive relations with other influencers based on the number of exchanges between them (strong ties) etc. The problem was that all these statistics were available through the REST API in twitter and you can only make 350 calls per hour to the REST API. This meant that we were not able to find the influencers through our recursive function, because every call to find a user statistic, would be one call to the REST API, and further to find exchanges between just two people would require us to download the tweets buried inside their timelines (which meant 4-5 another API calls per user). We had to find the influence statistic for every friend that a user had, and so we were constantly getting rate limited. We could calculate the score by inserting waits for one hour after 350 calls but this meant one could not create the conversation score on the fly for the interactive report.

With some research we found that the Klout had a API through which you could easily get the KloutScore for every user and his influencers and influencees. This was very similar to what we wanted with the rate limit problem solved. Using the KLOUT API, the klout scores and the influence statistic for every user we came up with a way to rank the conversations. The beauty of this was that we could find the influence score for every user even if they has not registered themselves on Klout. Klout returned very accurate results for the influencers in persons network.

The process was to register your application with the Klout for developers application. Once registered and confirmed, we could send HTTP requests to the Klout Api using specially constructed URLs based on the username or userid in twitter. Klout then returns a JSON response containing the users Klout ID, and then one could send another HTTP request to find the Klout Score, and the influence graph of a user. Then all that needed to be done was to desrialize the JSON responses and use these statistics to rank conversations found in a users network. This is included in the <https://github.com/iaperez/ABDTProject/blob/master/retrieval/src/ConversationDownloadSeema.java> file.

**Topic and Link Analysis.**
Topic and link analysis is an effort to harness the collective ego-network for finding and bringing prominence to #topics and links that might be of interest to the ego network’s central user.  Finding impactweets, we use the larger FoF network as opposed to the friend network.  It has two benefits; it helps increase the opportunity for discovery and serendipity (it’s a feature!) and it leverages a larger set of users to provide “mass” to the topics and links.#  The downside is a large set of non core-interest tweets now become far more prominent; this is providing high value out-of-network information and could be a valued asset.

The corpus of tweets contains intention to share and tag.   Votes are being cast as users broadcasting what is important, what web resources they’re consuming, what's got their attention.  Anything loathsome, beautiful, thought-provoking, etc.

The methodology used was making an index of each link and each topic, weighted by its occurrence count in the corpus.  These are the linkscore and topicscore, respectively.  It’s worth noting the scoring is a simple popularity, for this date and time and these users#, and includes the retweet count.  We can group all tweets containing each known link, topic, or by-user.  Pig was used to build indices of link-by-topic, link-by-user, topic-by-link, topic-by-user, user-by-topic-per-tweet, and user-by-link-per-tweet.

It’s important to take the RT into account.  The internet has vast opportunities for incoherent speech in addition to anonymous spamming, phishing, trolling and other malicious or online nuisances.  The RT, like Google’s ‘+1’ and Facebook’s ‘Like’ can be seen as a measure of coherency and non-nuisance.#  In the corpus of 1.1 million tweets, no prominent tweets looked like spam, though there were a few that were selling books or images.

**Validity**
Users go out of their way to add hashtags and distribute links they find notable or worth sharing.  Can we trust any hashtag related to links provided?  Probably not.  We’ve observed misspellings, utility (#dt, “donated tweet”), affiliation (#teamxtina, singer on a TV show) opinions (#DumbIdea), and other commentary (#MemoriesIWontForget).  A large corpus of tweets produces an aggregate of topics, links, indices of both; manual inspection of hundreds of links and topics in regard to one-another has proven highly valid -- though subjective.

Cory Doctrow’s “Metacrap” provides seven reasons why metadata is unreliable.  Based on our results we agree that state-of-the-art should not build systems that rely on semantic understanding of twitter metadata.  However, if the meta data is processed by humans, the semantic gap is much smaller and may be useful for exploration and discovery.  Again, individuals are going out of their way to share something they find notably valuable, or worth-sharing.  If we consider each tweet a possible vote for an association between topics and links -- or topics and topics, we run into a question of what the relationship is.

To answer that question, the JSON processing python application was modified to create a list of each topic to topic relationship.  An example might be topic n-grams.  For example:
*I love #pig.  It’s my favorite #ProgrammingLanguage.*
*Love this pic.  #fall #fallcolors #trees #leaves #stream #cold*
These contrived examples shows how we could try to draw associations between topics.  The script kept the order (#pig -> #programminglanaguage) but as of now, there is no reason to believe that would be necessary.  In the second example, we would divide the singular vote amongst all the topic n-grams [#fall -> #fallcolors, #fall -> #trees, #fall->#leaves ... #stream -> #cold].  A RT of the second tweet should be a multiplier of each edge between the topics.  Thus, each edge has a weight of 1/15;  (6!/(4!2!)).
This is because #tree and #leaves has a greater chance of being found related in other tweets.  We do not expect #tree and #cold will be found a such.

A note on spam.  Spam is in the eye of the beholder.  It’s a weed and looks like everything else but happens to be unwelcome.  We're not trying to end spam.  Any service tracking content is susceptible to spikes in topics.  While Twitter has done a good job of finding and removing sets of bots thus far it is a continuing cat and mouse game.

**Characterization of Dataset**
[**Observed tweets counts categorized by the number of topics and links in the tweet.**](https://docs.google.com/spreadsheet/ccc?key=0AhInyBD-dPuEdFFPUGg2blFPWnhOT2N6cU5UWENranc)

Unfortunately, we were pressed for time and the qualitative analytics section is lacking polish.  Ideally, I’d perform a chi-square test for independence of retweet rate by topic and link.  Additionally we could ask if there is a quantitative difference between the groupings, and would investigate with an independent sample t-test.  Much to my chagrin, I will only be providing some rough characterizations of the data.

The average tweet in this dataset (@CoreyHyllested, captured December 1st) has an average of 0.397 topics per tweet an average of 0.586 links per tweet.   There were always more observed tweets with one link than zero links shared -- regardless of how many topics were in the set.  This did not hold true for the retweet rate.  Both tweet and RT count decreased for each grouping with more topics.

RT drops-off as more links are added to the tweet, roughly about a factor of 30x.
RT drops-off as more topics are added to the tweet, roughly a factor of 4x.
Longest topic.  139 characters.
Most topics:  19.  2 tweets.
Most Links:  6.  1 tweet.
Beyond six topics, the link was usually an image (#trees, #fall, #leaves, #fallcolors...) or advertising spam (#deal, #sale).

**Assessment.**
Much of the testing done was from my network.  My expectation was to find topics related to my currently life (UC, Berkeley, Cal), past (BoCo), and high-tech interests such as linux, opendata, opengov, and startups.

The use of topics to "describe" links was effective.  Generally the stronger the score, the more I agreed with its accuracy.  The list of topics by topicscore (may not have a link), provided a list of interesting topics that trended.  I found it would be useful when bored but not useful when I am information seeking.  The top links from the ego-network I had not seen.  I found 3/10 interesting and good.

The topic-grams were very fun to explore and perpetually provided me something to laugh about, something to investigate, sometimes both (#ws2012, #giants, #druggedout).  However, there were problems.  Tweets with a high retweet count dominate the topic-gram graph generated with graphviz.  Future work should includes investigating better methods to visualize the strength of the topics and their relationships to shake the power law affecting it.  Another avenue is looking at the length of topics for commentary.
Lastly, the clustering of topics deserves deeper investigation.  A good deal of time was spent attempting simplistic clustering methods using weights, ratios, and k-core.  None of these were useful, largely due to problems in the visualization scheme.  There was no attempt at stemming, using of outside data sources, or calculating any hierarchical cluster.  No literature review of this area was attempted -- and should be before further steps are taken.

Very Large Images.
[Example directed topic graph.](http://people.ischool.berkeley.edu/~corey.hyllested/archive/rel262.png)
[Example problem with visualization.](http://people.ischool.berkeley.edu/~corey.hyllested/archive/rel265.png)

The late addition of looking at users and their topic per tweet, link per tweet, average retweet yielded very interesting information.   Taylor Swift has ~7000 average RT rate.  Michelle Obama is also in the thousands.   By taking an individual’s average RT by their number of followers might be useful to create a generator of interesting tweets metric.  Being able to group individuals by their sharing characteristics and appeal seems useful and another opportunity for future work.  Created [userscore](https://docs.google.com/spreadsheet/ccc?key=0AhInyBD-dPuEdHFMUHVjZmIzVTJHRzBwcnpQWEUzWXc#gid=0).

Overall, this tool is great for discovery.  I often found curious associations worth exploring.  It provided a great deal of variety, far more than I would have guessed.  This was a double-edged sword as I was presented with a high-signal output of reality TV topics, quotes, and links.  In retrospect, this is obvious, I pulled in about 100,000 users’ timelines.  One possible solution is weighing direct friends higher.  Alternately, I could try normalizing this corpus against an global sample of users.  This would reduce the effect of popular culture on an individual’s tuned followers.

**Interface design:**
We decided to generate a web report for users showing information found inside his or her network.  The report contains interesting tweets from the users network based on conversations and classification of each tweet, in addition to a summary view of hashtags, links from the FoF network.  The tweets are colored by the categorization type, stacked to show the number of total tweets,  organized by day of the week:


Java was used to analyze all information generated by PIG.  The output files were converted from XML to HTML documents.  Those documents are published in the same folder that our interface. To access those files, you can use the interface:
http://people.ischool.berkeley.edu/~iaperez/twitter/impactweets/indexStack.htm
<https://github.com/iaperez/ABDTProject/blob/master/retrieval/src/xmlGenerator.java>
<https://github.com/iaperez/ABDTProject/tree/master/data>

This report can be generated for CoreyHyllested but it can be generated for any user, by running the
1)https://github.com/iaperez/ABDTProject/blob/master/retrieval/src/TimelineDownload.java
2)<https://github.com/iaperez/ABDTProject/blob/master/retrieval/src/ConversationsDownload.java> using the username of the user, followed by running
3)<https://github.com/iaperez/ABDTProject/blob/master/retrieval/src/xmlGenerator.java>
4) Then finally loading the report https://github.com/iaperez/ABDTProject/blob/master/html/indexStack.html

**Problems and Further Research:**
We have also decided to explore the area of images posted on twitter, to find the interesting images posted in a users network.

We found interesting information during the classification of tweets. With a small amount of tweets classified to create a base model, (60000 tweets) the classifier is already discarding topics that are not related with a tweet with a 90% of precision, using an example dataset of 200 tweets of one of our users.  We are not filtering the tweets, nor removing stopwords.

We are planning to use external libraries to classify and find the topics of a Tweet. The benefit of using those libraries is that we will use a preexisting set of categories to classify tweets.

We didn't have a base model for topic and links.  The first sample was dominated by hurricane Sandy.  Politics were the other major topic.  The other large topics were travel, photograph tags, and politics.  They should be normalized to find the “trending” topics.  This should be approached in two ways.  The first is within the ego-network.  The second should be in the totality of twitter.  Following 100 people creates an FoF reach of 100,000, implying we’re being affected by a very, very large set of users.  Another approach is to weight those directly followed heavier than those merely in FoF network.

Another issue is time.  Data collected by timeline may reach as far into the past.  We found many tweets nine-months old.  Recent events will be more prevalence because more will fall under the “recent” tweets.  We saw this with the US Election; but a month past Sandy hurricane Sandy left a long shadow.  It would be useful to have a complete corpus that could be split up by time boundaries.   As it currently stands, results can be affected by the past.

A stumbling block for a true service offering this feature is providing medium and short term topics of interest and related topics.  For example.  Near the end of the calendar year, Christmas, will begin to “trend” up.  This is expected and should be notable in the short-term trends, but probably not in the medium-term trending topics.  However, Hurricane Sandy shouldn’t be part of a long-term base model.  It’s trending in December however feels quite notable.  As if it has passed into a medium term.  Events related to Hurricane Sandy or Election 2012, perhaps in January, would be able to recall related terms of interest.

A user’s profile understandably changes what she or he finds valuable.  Some are looking for news and using twitter as an aggregator.  Others use twitter to engage with friends people of interest, i.e. celebrities and the ambient intimacy twitter provides.  Lastly, some are staving off boredom by exploring and hoping for a serendipity.

**Project timeline:** (Green : Completed)

**Date:** Oct 27th

* Finalizing data formats, programming language of choice:  Python, Pig
* Identifying potential risks and planning mitigation measures:
	+ Issues recognized were each of us working on different sets of data.
	+ Was Naive Bayes categorization was achievable? Yes
	+ Training set for Naive Bayes?  Filter by keyword and topic.
	+ Rate limiting with REST API.

**Date:** Nov 2nd

* List of users whose networks we will analyze:

CoreyHyllested, Coldspire, aditi\_deshmukh, myy\_precious, imedha, akthiel
 Scraped.  We focused on building systems instead of collecting ego-networks.

* Code for extraction of data in the format decided.

Python and java binaries.  Categorization uses different acquisition methods.

* Visualize format of the analysis output.

Generate a text report for users showing information.

* + Interesting tweets based on conversations
	+ hashtags, links, retweets
	+ A classification for each tweet.
* Logical framework to perform the analysis should be ready
* Each part is a separate entity, requiring different inputs.
	+ Naive Bayes classification is hand-rolled, written in PIG and Java.
	+ Link and topic analysis is hand-rolled, written in PIG and Python.
	+ Using NetworkX to analyze network and find relations between users.

**Date:** Nov 9th

* Run the analysis code on the first set of data.

Ordering tweets by retweets, done.

Aggregation of most popular links and hashtags is working.

Finding conversations needs work, doesn’t scale well.

Classification is working based on a dataset of 60,000 classified tweets.

* Identify problems in execution and possibly correct them before the iteration

Data gathering process.  The Twitter API is delaying our timeline.  Hence we plan to run the data collection overnight for different users.  We have decided to test out all the analysis with one user’s data and test it, find issues

**Dates:** Nov 9th to 25th

* Testing recommendations with users.  (removed activity)
* Iteratively refine analysis code based on user-input.(removed activity)
* Develop an interactive report (new activity, completed)

**Work Percentage:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Corey** | **Seema** | **Ignacio** |
| **Data Collection For Links/Hashtags** | 100 | 0 | 0 |
| **Data Collection For Conversations** | 0 | 80 | 20 |
| **Data Collection For Categorization** | 0 | 0 | 100 |
| **Klout Integration** | 0 | 0 | 100 |
| **Literature Review** | 33 | 33 | 33 |
| **Analysis (Network, Topics & Data Gathering)** | 33 | 33 | 33 |
| **Meetings** | 33 | 33 | 33 |
| **Coding** | 33 | 33 | 33 |
| **Project Report** | 33 | 33 | 33 |
| **Interactive Report (Web)** | 0 | 15 | 85 |

**Coding and data references**:

* <https://github.com/iaperez/ABDTProject/>
* <https://github.com/CoreyHyllested/impactweets>
* <https://github.com/seemahari/impactweets.git>

**Appendix: References**

* + [A plan for SPAM](http://www.paulgraham.com/spam.html) - (MIT Spam Conference 2003)
	+ [Better Bayesian Filtering](http://www.paulgraham.com/better.html) - (MIT Spam Conference 2003)
	+ [Twitter Sentiment Classiﬁcation using Distant Supervision](http://cs.wmich.edu/~tllake/fileshare/TwitterDistantSupervision09.pdf)
	+ [Follow Me: Spam Detection in Twitter](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5741690&tag=1)  - SECRYPT 2010
	+ [Why Do People Retweet? Anti-Homophily Wins the Day!](http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewFile/2790/3291) - Sofus A. Macskassy and Matthew Michelson (ICWSM 2011.)
	+ [Unsupervised Modeling of Twitter Conversations](http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=16885300): Ritter, Alan; Cherry, Colin; Dolan, Bill (HLT '10)
	+ [Beyond Microblogging: Conversation and Collaboration via Twitter](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4755499&tag=1): Courtenay Honeycutt, Susan C. Herring (System Sciences, 2009)
	+ [Networks, Crowds, and Markets: Reasoning About a Highly Connected World: David Eakley and John Kleinberg](http://www.cs.cornell.edu/home/kleinber/networks-book/) (Cambridge University Press, 2010)
	+ Social networks that matter: Twitter under the microscope. - Huberman, Romero, and Wu
	+ Twitter Power: Tweets as Electronic Word of Mouth - Jansen and Zhang
	+ Everyone’s an Influencer: Quantifying Influence on Twitter Bakshy, Hofman, Mason and DJ Watts.
	+ I tweet honestly, I tweet passionately: Twitter users, context collapse, and the imagined audience - Marwick and boyd.
	+ [OMG, I Have to Tweet That! A Study of Factors that Influence Tweet Rates](http://www.aaai.org/ocs/index.php/ICWSM/ICWSM12/paper/view/4659/4980) - Emre Kiciman