# Twitter Product Referral Bot

**Team members:**

- Geobio Boo
- Leslie Chang
- Lauren Fratamico

All the members of our group will contribute equally throughout the project. Each member is expected to contribute to coding, research, etc.

**Mentor:**

- Marcus Phillips

**Project Goal:**

We built a referral bot (@sirosquishy) that first finds keywords in tweets where someone is expressing a desire for a product and then sends a reply to the tweeter with a message such as “Here is the top selling [product they were looking for]!”. This is a subscription only bot. It will only tweet to followers if they express a desire for a product in their tweet. The product results will be queried from Amazon. Since this bot provides automated responses, we need to be sure that the tweets are valuable for our subscribers so we don’t get marked as spam. The grade we earn will be determined as follows:

**Grading Criteria:**

A: We finish the bot and get at least one non-spam follower (not including us) or have people interacting with our bot.
B: We finish the bot but nobody follows/uses it.
C: We do not finish the bot in time.

**Project timeline with milestones, strategy:**

| Deadline | Task | Completed? |
|---|---|---|
| 11/5 | Do background research in the areas of Twitter spam detection, Amazon referral links, and Twitter bots in general. | ✔ |
| 11/13 | Bot is set up and is able to read tweets and reply with a recommendation. | ✔ |
| 11/26 | Perform any refactoring necessary to have a more successful referral bot. | ✔ |

| | | |
|---|---|---|
| 12/10 | Bot is finished and we have successfully referred a product. | ✔ |

**Literature Review:**

Most of our research was done on spam detection, referral efficiency over social networks, and tweet text processing. Our research on spam detection was needed to help our bot avoid being classified as spam by Twitter and avoid replying to spam tweets we receive. Our research on referral efficiency will increase the probability a user clicks and buys the recommended product. Our research on word processing helps us correctly extract the keyword from a tweet and give a relevant product recommendation.

The next steps for literature review is to look into successful product referrals and algorithms for detecting what people really want. Additionally, extra research will go into detecting what the user meant, since the top search result on Amazon often does not match what the user intended.

Spam Detection Research

- Search engines rank tweets based not only on the content of the tweet, but also on how influential the user who posted the tweet is and the more followers the user has, the more influential they are. (1)
- Spammers increase their influence by following popular users. (1)
- The more popular a user is, the more likely they are to reciprocate to spammers' follower requests. (1)
- Many spammers use popular hashtags in their tweets along with a link unrelated to the trending topic. (2)
- Spammers post a higher number of tweets with URLs versus non-spammers. (2)
- Spammers post a higher number of hashtags per tweet versus non-spammers. (2)
- Most spammers have very new accounts. (2)
- Spammers mention many users in their tweets in order to increase their visibility. (3)
- Spammers tend to follow more users than have users following them. (3)
- Spammers who tweet out links to the same site are part of a spam campaign. If the link is not blacklisted, then the spammers may be undetected. (7)
- Tweets that contain a high number of frequently used spam words versus the total number of words are more likely to be spam. (7)

Referral Efficiency

- Tweets with longer lengths are more likely to be clicked. (4)
- Tweets that had linked about 25% of the way through had a higher click rate. (4)
- Tweets with more adverbs and verbs than nouns and adjectives had a higher click rate. (4)
- Weekends and nighttime give you the highest visibility. (4)
- The higher the time gap between tweets, the more likely your link will be clicked. (4)
- Share valuable content in your own voice - personalize the tweets (8)
- use keywords - they are the backbone of content (8)

- connect with followers - in our case, give them more personalized recommendations to better ensure that they will use our service again (8)
- engage your audience - don't just set the twitter account on autopilot (8)
- use hashtags to curate conversation around our bot (8)
- to increase twitter followers (9):
    - encourage retweeting - gets your @username more seen
    - fill in your bio
    - include pictures in tweet
    - get involved with hash tag memes and trending topics
    - track results
        - TwitterCounter (http://twittercounter.com/) shows how many new users you're adding per day
        - Qwitter (http://useqwitter.com/) will email you when someone unfollows you after a tweet

Word Processing

- Research on detecting questions that are desired to be answered by friends in tweets - related as we are detecting desires and may branch out to more than just the basic "I want X" construction (5)
    - people trust their friends to answer questions that are difficult to answer using search engines. they trust their friends to provide tailored, contextual responses. social network like Q&A is becoming more popular
    - hard to extract questions from twitter due to the short, informal, nature of tweets
    - created a NLP parser for tweets, as sometimes even things that are formatted like questions, are not ones that are meant to be answered - we may run into this problem too
    - problems with tweets that make them hard to parse
        - people less concerned with correctness when composing tweets, errors are common
        - garden variety spelling errors can be handled with spelling correction algorithms
        - tweets often have repeated letters "hrmmmmmmm", "hahahahahhaha" - probably won't have to deal with this in our case as we will be looking for the nouns
        - homophones - shorter syllables generally substituted for longer ones "b4", "2morrow", "sumthing"
        - punctuation - can be random
        - emoticons
    - developed a parser that took the above into account - if we run into trouble parsing our tweets, could follow up with them for more direction
- Research on creating a part of speech tagger that is more appropriate for use with tweets (6)
    - poor ability of standard NLP tools on tweets. research focused on re-building the NLP pipeline starting with part-of-speech tagging
    - difficult because 140 character limit - lacks sufficient content to determine what a word is referring to (could be a band, movie, store, company, product)

- by training their parser on large amounts of unlabeled data, with dictionaries of words in Freebase, and using topic models, they were able to develop a parser that achieves a 25% increase in F1 score over a previous approach
- tools available at http://github.com/aritter/twitter_nlp with directions on how to run there

## Mid Project Checkpoint Accomplishments:

We have a fully working prototype which currently streams tweets, filters for "I want ____", and then does natural language processing to get the adjectives/noun that the user is looking for, and then does a search on Amazon, and tweets to the user that they should look at the product we found from Amazon. Currently we use streaming because we have not yet built up our subscribers, and are just prototyping (all current tweets are manually deleted within a few minutes of posting).

Currently, we use the twitter4j library for streaming and posting, OpenNLP for natural language processing, Amazon's API for product lookups, and bitlyj for shortening URLs.

Additional work will go into improving the matching algorithm, and including more info as to why we selected this item to recommend to the subscriber.

## Final Accomplishments:

We wanted to improve our bot in three ways so that it could be more useful to people: respond to more tweets by improving our matching algorithm, cater the tweet response based on the desired product, and recommend the right product. We attempted each of these in numerous ways, outlined below:

**Respond to More Tweets:** Initially, we were just detecting the phrase "i want" in a tweet, and processing those. We improved this in a couple of ways. First, we added more key phrases to look for. Our bot now responds to tweets with any of the following phrases: "want", "wish", "hope", "desire", "would like", "need", "require", "must have", "crave", "craving", "gotta have", "gotta get". This vastly expanded the tweets we were able to process, however we still felt that we could detect more. For example, we wanted to be able to detect people saying things like "Laptop died. Need a new one." and be able to give them laptop recommendations. We tried two solutions to solve this problem. Since Google already has NLP built into its search, we decided to use this existing service instead of building our own from the ground up. We passed in the entire tweet to both the normal Google search and Google shopping. With passing it into the normal Google search, we looked at the top 25 results to see if they came from websites like Amazon, ebay, Target, Walmart, Newegg, ToysRus, etc. We hypothesized that if some number of the top 25 were from websites like those, then the tweet must be one that is expressing a desire and warrants a response from our bot. Figure 1 shows a screenshot of searching google for "Laptop died. Need a new one." As you can see, all the hits are from forum websites. Though we labeled this tweet as one that we would like to respond to, Google search is not helping us label it that way. The results from searching the same tweet in Google Shopping are shown in Figure 2. Similarly, it does not appear that Google Shopping will help solve our problem. Google Shopping returns products like laptop covers with dead images on them, which is not what we want to be recommending. To be sure that using Google Search would not help up, we collected ~5000 US tweets and passed them to the Bing Search API. 5,000 is the amount of queries you could pass to the Bing API monthly, and this was higher than the amount Google allows. We looked at the top 25 results and if the ith result was from one of the websites mentioned earlier, then that position would get a score of 1. We would end up with a 25 digit binary number with 1s in the places where one of the above

websites was the site the hit was coming from. We then sorted the results. An excerpt of it is shown below (result string where most significant bit is the top website hit, the original tweet) and the entirety of the list is in the file sortedTweetsWithBing.csv:
111111001000000000000100011, I'm at Target (Paramus, NJ) http://t.co/qjYI1GBO
110010000000000000000000, I have a lego iphone case
110000001000000000000000, My Mockingjay book came in! :)
110000000000000001000000000, Gotta get all new stuff for my dorm, wardrobe, laptop, supplies, food, sigh*.....
110000000000000000000000, in years since the song released.. riding dirty is one of the most garbage songs chamillionare has released.
110000000000000000000000, Eyy they brought back the box logo pullover
101101100011010000000000, I'm at Target (Columbus, OH) w/ 2 others http://t.co/7vGyfHKE
101000000000000000000000, UNTIL TODAY
101000000000000000000000, Burger kind has buy a whopper get one 55 cents

The majority of these results are not ones that we would want to be detecting. The 1st and 7th ones are checkins at Target. The 2nd and 3rd are by people who already have the object. The 4th is one that we would like our bot to respond to (however, it already would because that tweet has the phrase “gotta get” which is one that we look for). The last tweet is one that may warrant a response, but not one that we would want to look up on Amazon. At this point, we decided to abandon this approach and try other ways to improve our bot.

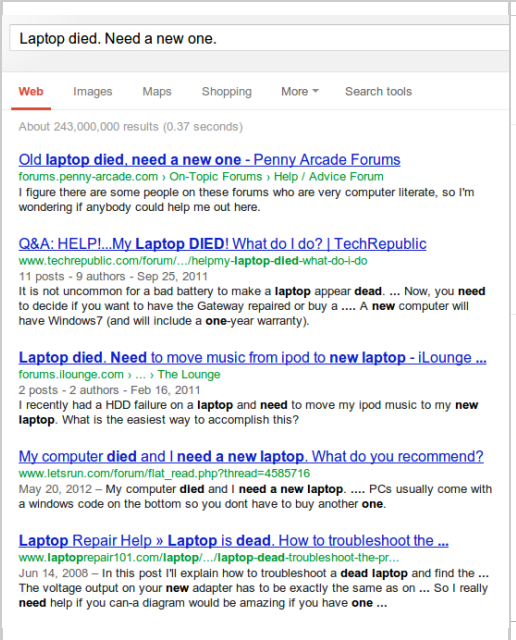


Figure 1: Google search for “Laptop died. Need a new one.”

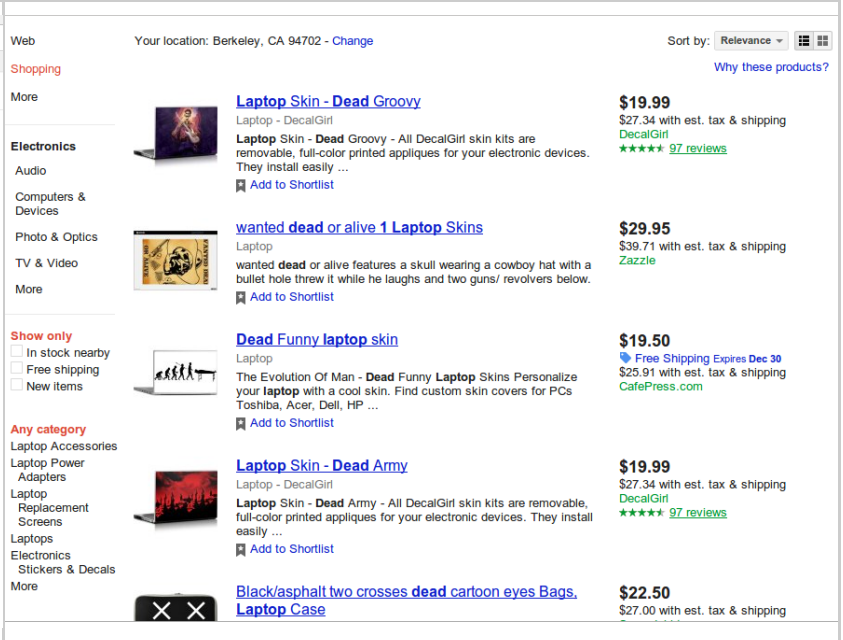


Figure 2: Google Shopping search for “Laptop died. Need a new one.”

**Cater Tweet Response:** We solved this using the Amazon API. When a query is passed to the API, in addition to the top hit for that query, it also returns the department that the product is in. When we detect a tweet in categories like groceries, books, or toys, we change the format that we tweet back in. For example, when someone tweets “I want cereal!” and we pass “cereal” to the Amazon API, Amazon returns the top hit along with the department that it is in (in this case groceries). Since it is in the groceries category, we change the response tweet to the form “[product] delivered straight to your

doorstep! [link]". So in this case, we would tweet back "Cereal delivered straight to your doorstep! amzn.to/11MRipA". In the case of books, we would respond with a tweet of the form "here's the book you were looking for! [link]". We believe that this will improve the retention of followers and users as people will be more interested when they get a varied response instead of the same cookie cutter "Buy this! [link]" tweet.

**Recommend the Right Product:** We developed an algorithm for this where we would use machine learning to ensure that our recommendations are always getting better. Each time we make a recommendation, we add the item recommended to a list along with its category. This allows us to track the most popular items and categories. In addition, we keep track of the quality of the recommendation. Each time someone responds to our recommendation, we analyze the sentiment of the tweet. We rank these based on how positive/negative the sentiment is. We also keep track of when a person buys a product on amazon that we linked them to. When a purchase is made, that product is ranked higher. We keep track of the query that was made and the positive product that was recommended (or the new product to recommend if a person clarifies in their tweet response), and use this knowledge as a factor of which product to recommend next time.

## Software:

Repository: https://github.com/geobio/info290bot/ (private, contains my API keys)

Run the Java code. Main class is TweetBot.java and doesn't take any args.

To interact with the bot: Follow @sirosquishy, and then start tweeting!

For example, "gotta get me a cake"

## Next Steps:

The next steps for our project would be to get more users. With this, we could fine tune the recommendations and eventually converge on the best recommendations for all people and products. We would also like to be able to handle products that could not be found on Amazon. For example, if someone tweets at 2:00 AM, "really hungry for pizza", we could link them to an open pizza place. We also want to expand to other e-commerce websites that may be more relevant to the product the follower wants. For example, if the user expresses the desire for a new Apple Macbook Air, our bot can return results from Newegg, BestBuy, etc.

## Goals we met:

We were able to build a bot that listens to its followers tweets and filters for tweets that express a desire for a product. Using natural language processing, the bot identifies the product and queries Amazon for the best product match. The bot then sends a reply with a link to the product on amazon. The challenges of our project were: filtering for the correct tweets, giving more relevant product recommendations, and catering our reply tweet based on the desired product. Originally, we wanted to do machine learning but from the advice given by our project mentor, Marcus, he recommended us to use other services that have already implemented them, like Google Search.

Our goal for an A on this project was "We finish the bot and get at least one non-spam follower (not including us) or have people interacting with our bot." We accomplished this. Thanks to a classmate for following us with no coercion from us! Ideally we would have liked to have more followers,

but we ended up focussing more on making a good recommendation service versus spending more time on marketing.

**Work Percentage:**

| | Geobio | Lauren | Leslie |
|---|---|---|---|
| **Code** | 33.3% | 33.3% | 33.3% |
| **Research** | 33.3% | 33.3% | 33.3% |
| **Report** | 33.3% | 33.3% | 33.3% |

**Bibliography:**


1. Ghosh Saptarshi, Viswanath Bimal, et al. “Understanding and Combating Link Farming in the Twitter Social Network.” *21st International World Wide Web Conference.* Lyon, France. April 2012. http://www.mpi-sws.org/~farshad/TwitterLinkfarming.pdf

2. Benevenuto Fabricio, Magno Gabriel, et al. “Detecting Spammers on Twitter.” *Seventh Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference.* Redmond, Washington. 2010. http://ceas.cc/2010/papers/Paper%2021.pdf

3. M. McCord, M. Chuah. “Spam Detection on Twitter using Traditional Classifiers.” *Autonomic and Trusted Computing Conference.* September 2011. http://www.cse.lehigh.edu/~chuah/publications/atc11_spam_camera.pdf

4. Dan Zarrella. “How to Get More Clicks on Twitter.” *The Social Media Scientist.* http://danzarrella.com/infographic-how-to-get-more-clicks-on-twitter.html#

5. Kyle Dent, Sharoda Paul. “Through the Twitter Glass: Detecting Questions i Micro-Text.” *AAAI*. Palo Alto Research Center. May 2011. www.parc.com/content/attachments/through-twitter-glass.pdf

6. Alan Ritter et al. “Named Entity Recognition in Tweets: An Experimental Study”. *EMNLP*. University of Washington. 2011. turing.cs.washington.edu/ritter-emnlp2011-twitter_ner.pdf

7. Zi Chu, Indra Widjaja, et al. “Detecting Social Spam Campaigns on Twitter.” *ACNS 2012.* Singapore. June 2012. http://www.cs.wm.edu/~hnw/paper/acns.pdf

8. Cindy King. "17 Twitter Marketing Tips From the Pros". *The Social Media Examiner.* October 2011. www.socialmediaexaminer.com/17-twitter-marketing-tips-from-the-pros

9. Kevin Rose. “10 Ways to Increase Your Twitter Followers”. *Techcrunch.* January 2009. http://techcrunch.com/2009/01/25/kevin-rose-10-ways-to-increase-your-twitter-followers/