Tshrdlu Expanded

Jim Evans
Department of Linguistics
University of Texas at Austin
j.s.evans@utexas.edu

Jason Mielens
Department of Linguistics
University of Texas at Austin
jmielens@utexas.edu

Abstract

We have created MOODY, an automated Twitter user that is based on the TSHRDLU bot. When another user tweets at MOODY, it resorts to either searching Twitter or searching an offline corpus of tweets to find a tweet to use as a response to that user. Several of MOODY's characteristics make it appear more humanlike in its interactions, including its mood system, its ability to acquire opinions, and its geographical awareness.

1 Introduction

In this paper we introduce the automated Twitter user (henceforth 'bot') MOODY, which is based on the TSHRDLU bot. TSHRDLU's standard behavior is to search Twitter for tweets to use as responses when another user tweets to it; it randomly chooses key words from the incoming tweet to query Twitter and find a tweet to use as a response. In order to make MOODY seem more human, it has been designed to have emotions, to be sensitive to emotion in other users, and to have its emotions be affected by 'local' weather (the bot 'lives' in Austin, Texas). To make its tweets more relevant, we have made bot so that it is sensitive to the geolocation of tweets in certain situations.

Our primary motivation for adding these enhancements to TSHRDLU was the observation that many Twitter bot implementations either simply do one specific thing (potentially very well), or are of the more hodge-podge variety as the original TSHRDLU was. The original TSHRDLU had a set of behaviors that could be selected in replying to a particular tweet directed at the bot, but it lacked any sort of overarching features or cohesive quality. This is in direct contrast to humans, who may be tweeting about baseball one minute, and a news story the next, but they are still the same human and tend to display a pattern of similar qualities; a 'style'. We felt a simple way to begin to give a style (and potentially bring some unification of behaviors) to TSHRDLU was to add features that would ground it in the real-world of human events and emotions.

2 Methods

MOODY's enhancements over TSHRDLU are primarily in its mood system, which is the biggest single addition. As described by Bollen et al. (2011), tweets do contain emotional content that can be correlated with real-world events. This means that a system that intends to model a human twitter user must somehow be grounded in the world. Our system, then, has a number of sensitivities to real-world information – for instance it is sensitive to weather, as well as to geographic relevance.

Another major development is the addition of a large previously collected corpus that can be searched for relevant tweets along with actually querying the current, live version of Twitter. The mood variable is such that $m \in \{happy, angry, sad\}$, and the mood that the bot is in can be affected the weather as well as the mood expressed in content that users tweet at the bot.

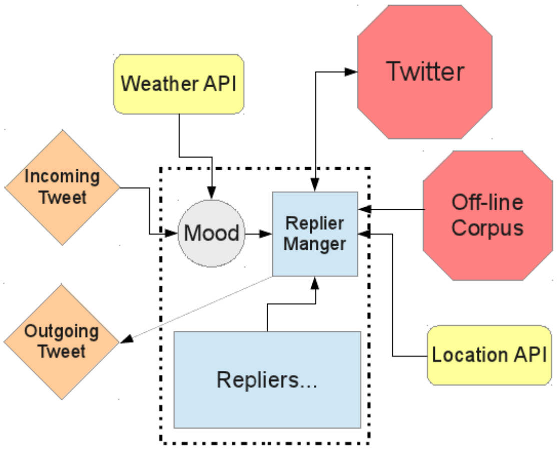
The following sections describe the major components of the bot and how they interact, a diagram of which is shown in Figure 1.

2.1 Mood / Sentiment

As previously mentioned, the single biggest development of this project was the addition of a variable describing the 'mood' of the bot. This impacts the overall quality of the tweets sent by the bot by doing a better job of emulating human behaviors in the form of emotions currently held by the bot.

The mood of a tweet (and thus, indirectly, the bot) is calculated by finding the relative percent-

Figure 1: MOODY System Overview



ages of all of the sentiment words that belong to a particular emotion. For instance, if a tweet contains eight sentiment words, two of which are angry, we describe the mood of that tweet as 25% angry. Empirically, we have found this relative formulation superior to competing techniques such as simply counting the sentiment words. While doing counts is useful in the sense that a tweet with 10 happy words is 'more happy' than a tweet with two happy words, we found the relative formulation to produce more obviously sentimental tweets (more tweets are less ambiguous in their emotional content) and it makes to bot do a better job of finding tweets that match the sentiment (because we don't filter out all of the candidates nearly as much).

As a source of sentiment words grouped into emotional categories, we have chosen to use the Linguistic Inquiry and Word Count (LIWC)[1] dictionary of Pennebaker and King (1999), which contains a finely tuned set of emotional words. We have chosen the LIWC lexicon because the words in the categories of the lexicon has been validated and checked by actual psychologists as being predictive of emotional state (Tausczik and Pennebaker, 2010), rather than being a set of words someone put together because they just intuitively seem negative or positive in some undefined way.

Having a bot that has moods or emotions raises the question of which moods or emotions it should display. Currently, the bot maintains one of three moods: Happy, Sad, and Angry. These categories are from the LIWC dictionary (they correspond to LIWC's 'Positive', 'Sad,' and 'Anger' word lists, respectively). There are many more LIWC word lists could potentially be considered.

'Positive' is a large LIWC category, while 'Sad' has only around one hundred terms. We can't simply count the number of words of each category that a tweet has, and assign the emotion based on these numbers. We must account for the fact that Positive just has more words in it. In developing our system, we found that classifications (per tweet, but also as part of the task of opinion formation, which is discussed in its own section seemed best to us when the raw count of Sad and Anger words were unchanged, but the Positive raw count was divided by 5.0. This gave 'calibrated' counts.

To implement this non-binary sentiment, we keep track of the relative percentages of emotions that the bot should be exhibiting, and then locate tweets that are dominated by whichever emotion is currently greatest in the system. For instance if the bot is 20% angry, 15% sad, and 65% happy, we would look for happy tweets. Looking for tweets that contain the exact mix that the bot is in would likely lead to even greater sparsity issues, so we have adopted this winner-take-all model of emotion rather than searching for more fine-grained emotional mixes.

2.2 Location

Grounding the bot in location and time was also an important development we made. To do this, we have chosen to have the bot 'live' in Austin, TX. This choice means that the bot will prefer to tweet about the things that the real people of Austin tweet about. This is achieved by an additional filtering step that proceeds similarly to the filtering based on mood described earlier; the filtered tweets that match the bots mood are subjected to an additional constraint of sorting by their distance from Austin. Once this sorting has been done, the closest top third of the tweets passes through the filter to become the final tweets returned to the replier.

By doing this sorting based on distance to Austin, we are restricting ourselves to geolocated tweets. This may initially seem to present a problem, as only a very small percentage of tweets are geolocated. However, the offline corpus that we are using consists entirely of geolocated tweets, which provides us a large pool to draw from to find matching geolocated tweets.

[1] James W. Pennebaker, R. J. Booth, and M. E. Francis. 2007. Linguistic Inquiry and Word Count: LIWC [Computer software]. Austin, TX: LIWC.net.

2.3 Per-User Behavior

The mood of the bot is not determined on a global level. That is to say, there is a not a single mood that the bot is 'in' at any one given time. This is the way previous versions of the bot worked. Rather, the mood with which the bot responds to a particular user is chosen based on an individual mood variable that is unique to that particular user.

By breaking up the mood into a per-user behavior, we hope to more accurately model the way in which humans interact with other humans – people may be happy while talking with one person and angry while talking with another, perhaps based on previous encounters. Of course in the real world, interactions with one person may impact the mood with which someone interacts with another person, but this 'bleed-over' is not modeled by our bot currently.

Each individual mood variable is initialized based on the current Austin weather. When individual users tweet at the bot, they are only impacting their own personal mood variable. The mood is updated on a continuous basis – if someone consistently tweets angry things at the bot, the bot will respond in an angry manner, even if many other people are tweeting only positive things at it.

2.4 Opinions

Aside from the ability to filter tweets based on mood, the other major feature of MOODY is to form what we call 'opinions' about things. Essentially, these are items that are treated like elements of the sentiment lexicon despite not being traditionally thought of as emotional words. For instance, the bot may have an opinion that 'cheese' is a happy word, and will accordingly feel more positive about tweets that contain this word. While the LIWC sentiment lexicons provide words that are somehow indicative of an emotional state, the words we form opinions about have the slightly different interpretation that they make the bot 'feel' a certain way; in any case, it seems natural to combine the sentiment and the opinions, rather than maintain them separately.

Currently, the bot gets opinions by receiving a tweet with the command 'getOpinion: X' where 'X' is the thing to form an opinion about. To actually form this opinion, the bot searches Twitter for tweets containing the opinion item and averages the sentiment of the words found in those tweets.

Empirically, we have found a good correlation between the opinions the bot discovers using this method and what we assume should be the 'correct' opinion. For instance, 'cookies' was around 80% positive, 'death' was around 6% positive, and 'politicians' was around 30% positive; More details can be found in the Results section.

In the future we would like to improve the way opinions are formed, potentially using follower information to privilege the opinions of certain users (people the bot follows, for instance); see the Future Work section for additional details.

2.5 Offline Corpora

One of the primary issues that we encounter when dealing with Twitter, despite its popularity and near ubiquitous presence in the micro-blogging world, is that of data sparsity. In many situations, when a replier object wishes to search for a given query, there simply aren't any tweets returned from Twitter that express the same mood that the bot is currently attempting to express. For instance, there may be many tweets about 'baseball', but if the bot is currently in a sad mood it would need to find sad tweets about baseball, which may be much more sparse. This problem could be addressed in a number of ways, and we considered multiple possibilities. It is possible to simply redo the search for any query that that fails to provide any tweets of the correct mood. However, this strategy is taken then the bot would likely be forced to limit itself to a small number of redos per search, as this repeated searching is likely to result in the bot being rate-limited by Twitter.

An alternative method that we are currently using with some success is the use of a large offline corpus of collected tweets. This corpus essentially becomes our own 'private Twitter' and we are obviously able search it for as many tweets as we like without concern for rate-limits. Additionally, the ability to index this corpus and the fact that it is local means queries are much faster than dealing with Twitter. The availability of this offline resource provides several benefits, but perhaps the largest benefit is that we are now more likely to find tweets that match both the content and the sentiment that we are searching for, because we now have an additional set of millions of tweets. Additional methods of dealing with sparsity in Twitter sentiment include the use of sentiment topics to augment the sentiment features of tweets (Saif et al., 2012), or the inclusion of additional tweets by the user as context (Jiang et al., 2011) – such approaches could be beneficial for the bot, but have not currently been implemented.

This offline corpus is implemented as a Lucene index that was created from a large set of geolocated tweets.

One potential downside of this technique is the lack of temporal relevance. By using a stored corpus, we obviously do not have access to current tweets that may reference current events or trending topics, and thus be more interesting or relevant to users. For instance, the majority of current corpus being used was collected over a year ago. However, there are certainly a wide variety of topics that are not at all temporally linked (or cyclical, in the case of holidays and such), so losing this relevance isn't a primary concern. We mitigate this potential downside by first searching the live version of Twitter, and if no matching tweets are found, backing off to searching the offline corpus rather than simply re-searching Twitter and incurring an additional API call. This way, we get the most temporally relevant if at all possible.

3 Results

3.1 Successful behavior

The following interaction occurred while the bot was in a happy mood (NB in the following interactions, we use '@bot' when in actuality the tweet is '@evans_anlp'.):

- @bot movies or television?
- @jj_anlp saw it! It was badass! Loved it. Can't wait for more super hero movies this summer

This one occurred while the bot was sad:

- @bot movies or television?
- @jj_anlp I hate love movies can't believe I'm crying :'(<3

Here is an example of a tweet when the bot was angry.

- @You seem really agitated.
- @jm_anlp Irritated and don't feel good I wish I can move TF AWAY out the states and I hate my Older ugly goat one eye lookin Azz sister

These are examples of the influence of our bot's mood. More examples are available at the following two web pages: `https://twitter.com/evans_anlp`, `https://twitter.com/jm_anlp`.

Here are a few examples of 'correct' opinions. The 'winning' opinion is given, followed by the tuple of word counts as '(positiveCount, angryCount, sadCount)'[2]:

- Hitler: angry (28.6,125.0,97.0)
- crime: angry (7.2,26.0,12.0)
- taxes: angry (8.4,19.0,16.0)
- sunshine: happy (13.8,7.0,6.0)
- pie: happy (12.2,9.0,2.0)
- injury: sad (7.2,9.0,23.0)
- death: sad (8.8,32.0,13.0)

3.2 Unsuccessful behavior

Most of the bot's serious failures came out of attempts to have the bot develop reasonable opinions. We hoped that classifying the mood/sentiment of tweets that contain a target word would allow us to find out if the word is something that people makes people happy, sad, or angry. Most of the bot's opinions were reasonable, but some were not. For example, here are the results for 'victory,' 'massage' and 'comedy.'

- victory: angry (10.2,16.0,3.0)
- massage: sad (7.6,9.0,9.0)
- comedy: angry (10.4,13.0,3.0)

3.3 Evaluation

Since our goal has been to make the bot respond appropriately to human-generated tweets, we will use feedback from human users that have interactions with the bot to evaluate it. There are two kinds of evaluation: single interaction (i.e. a tweet to the bot and its response) and full conversation. In the single interaction phase, subjects were asked to tweet something at the bot, independent of anything they might have previously tweeted at

[2] counts here are not raw counts, but rather the 'calibrated counts' mentioned in the Methodology section.

the bot or heard from the bot. With each interaction, the evaluator rated the bot's response for relevance, humanness, entertainment value and overall quality on a scale of 1 to 5. After rating ten interactions, the subject were asked to carry on a conversation consisting of four tweets to the bot, and the bot's responses. The evaluator rated the conversation with the same criteria as before (relevance, etc.). Each evaluator had five conversations with the bot and scored them. The single interactions are our main concern, since our impression is that people tend to tweet in a single-interaction style. The evaluation tables are provided below. Recall that scores are from 1 to 5 in each category.

	relevant	human	entertain	quality
Evaluator 1	3.3	3.9	3.9	3.8
Evaluator 2	2.2	2.6	2.9	2.3
Evaluator 3	2.0	2.3	4	2.6
Average	2.5	2.9	3.6	2.9

Table 1: Single interaction evaluation

	relevant	human	entertain	quality
Evaluator 1	2.0	3.0	2.6	2.6
Evaluator 2	1.8	1.6	2.0	1.8
Evaluator 3	1.8	1.8	2.2	1.8
Average	1.9	2.1	2.3	2.1

Table 2: Conversation evaluation

The evaluation scores are discussed in the next section.

4 Discussion

4.1 Evaluations

Since the purpose of the bot is to be an interesting, humanlike Twitter user, we decided to evaluate it based on interactions with volunteer evaluators. It is unlikely that these evaluators were able to notice the complexity of the bot's mood system, because the bot initialized the per-user moods based on the weather, which was good on the day of evaluations. None of the evaluators seem to have changed the bot's mood during the course of evaluation. So the bot probably just came across as mostly positive in all of its tweets.

It is clear from the tables that the bot is better at single interactions than conversations, but this is to be expected, since presumably all four of the bot's replies would have to be completely relevant for the conversation overall to have a relevance of 5.0. It appears that in both conversations and single interactions the bot performs best as a source of entertainment. High entertainment value does not indicate that our bot is doing what we wanted it to do. We had hoped to see better numbers for relevance and humanness.

4.2 Sentiment

Our sentiment classification is often inaccurate. To some extent, this is because our simple lexicon approach to sentiment analysis is inherently problematic due to oversimplification. For instance, 'care' is listed in the positive emotion word list of LIWC. Therefore, a common problem for us is 'don't care,' which, from our impressions, more often expresses some kind of negative emotion rather than positive emotion (though not always). This means that our bot would think 'I don't care about you' is a 100% happy tweet. Another issue we have is that 'like' is in the happy lexicon, but its most common use is not as the verb, but as a preposition or a filler word. Despite occasional errors, our mood system generally works quite well.

4.3 Opinions

When we use sentiment classification to form opinions, 'errors' are much less frequent. Of course, one's feelings about something, which is what we mean by 'opinion,' cannot be accurate or inaccurate per se. So if the bot has a few uncommon opinions, that might not be a problem. However, there are some opinion that really should be avoided if the goal is having the bot seem human, such as having a positive opinion about 'pain' or 'disease' (which were opinions that earlier versions of MOODY developed, before its current sentiment system was designed). The most problematic opinion that MOODY has developed since it was finalized is its angry opinion of 'victory.' It was attempted again on a different day, and that time 'happy' had the highest count, but barely. It seems that tweets containing the word victory have surprisingly few of our sentiment words.

5 Future Work

5.1 Non-Binary Sentiment

Despite being the core element of this project, we remain slightly underwhelmed with the ability of the bot to find clear examples of emotional tweets. We believe this is due to the size of the lexicons we are using, which in the case of some of the smaller LIWC lexicons is no more than 100 words. This small corpus means that many angry tweets, for instance, are simply missed because the words that make them angry are not included in the lexicon.

To fix this, we recommend collecting a larger corpus, perhaps using Twitter hashtags as labels (#sad, #happy, for instance). Despite the occasionally poor performance, we are confident that word-level sentiment lexicons can be effectively exploited to perform sentence-level classification, as shown by Mohammad (2012b). Furthermore, it is not immediately apparent that additional linguistic processing such as Part-of-Speech (POS) tagging would be needed or even useful, as Kouloumpis et al. (2011) noted no particular improvement in sentiment classification accuracy when using POS tags.

5.2 Opinion Seeding

Currently, the opinions formed by the bot are taken from essentially the average opinion of Twitter. While this is certainly a viable option, it means that the bot doesn't really have many 'interesting' opinions ever – it never has things it inexplicably hates or loves, like most humans. Such randomness could be encoded by either randomly initializing some set of opinions, or manually seeding them. For instance, it would be a relatively simple task to give the bot opinions about sports teams or music genres by randomly initializing a list of these items.

Another alternative, mentioned earlier, is that opinions could be created from the average opinion of Twitter users the bot is following, or some sort of 'influential people' list. This would have the downside of potentially not having enough data from such a limited set of users to adequately form opinions about arbitrary items; for example, perhaps no one the bot follows has tweeted using the word 'cheese', which would mean the bot couldn't form an opinion about cheese.

6 References

Johan Bollen, Alberto Pepe, and Huina Mao. 2011. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proceedings of the AAAI Conference on Weblogs and Social Media.*

L Jiang, M Yu, M Zhou, X Liu, T Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of ACL: HLT.*

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg. In *Proceedings of the AAAI Conference on Weblogs and Social Media.*

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, June 2013, Atlanta, USA.

Saif Mohammad. #Emotional Tweets. 2012. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (∗Sem)*, Montreal, Canada.

Saif Mohammad. 2012b. Portable Features for Classifying Emotional Text. In *Proceedings of NAACL.*

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval* 2.1-2: 1-135.

James W. Pennebaker and Kaura A. King. 1999. Linguistic Styles: Language Use as an Individiual Difference. *Journal of Personality and Social Psychology*, 77(6).

H. Saif, Y. He, and H. Alani. 2012. Alleviating data sparsity for Twitter sentiment analysis. In *Proceedings of the #MSM2012 Workshop*, CEUR, volume 838

Yla R. Tausczik and James W. Pennebaker. 2010. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Jounal of Language and Social Psychology* 29(1) 24-54.