Ben Packer and Byrne Hollander   CS 73   **Project**: pome [*sic*]

1. Our goal is to make an entertaining Twitter bot based off of existing tweets. We intend to combine multiple tweets to make nearly semantically coherent poems with variable structure parameters (e.g., meter and rhyme) and tweet them. We will also try to recognize words that aren't in the dictionary for rhyming purposes and use tweets that rhyme and have the closest possible semantic relationship. We will try to use the tweets that already exist (just rearranging them to make "poetry") or we might change words in tweets to make it rhyme using a thesaurus (semantic distance).

2. There are a few Twitter bots out there that use existing tweets to make "poetry." One of them is @Pentametron which "looks for poetry in everday musings." It retweets tweets that are written in perfect iambic pentameter and even posts "unintentional sonnets" on a website associated with the account. There is also an accidental haiku bot (@accidental575) that finds tweets in the 5-7-5 haiku form and tweets them formatted and with attribution. As far as existing work in the field, there is lots of work on semantic distance and the use of computational thesauruses.

3. Since both of us are very familiar with Python and there are availible Twitter APIs written in Python, the language seems like the natural choice. A description of how to create a Twitter bot using Python can be found here. Our data will be scraped from Twitter either using their standard API or their Streaming API if we decide to only limit results to live tweets. We will then combine material from many different tweets to generate different possible poems. We will devise a mechanism of assigning a score to each possible poem, based on its coherence and how neatly it conforms to desired poem structure (measured as the opposite of how much our program must change it in order for it to fit). We will manage our project using GitHub, and the final code will be run from one of our computers or a Dartmouth CS computer where we will Tweet at it to test it.

4. We have begun researching Twitter APIs and determining the best approach to create our bot. We have done research on semantic distance and rhyming as well. We have also registered a Twitter account (@PomeSic).

5. There two categories of tasks that must be done – first, the interaction with Twitter, and second, the actual poem creation. In the interest of exposing us to both categories, we will split up both. As of now, Byrne will create the basic Twitter bot system that receives and replies to Tweets, and Ben will make the the scraping program. In the poem construction category, Ben will focus on rhyming and coherence of the poem incorporating semantic distance, and Byrne will focus on meter and recognizing and processing slang.

6. Three milestones. Be realistic about the milestones, but not lazy.

   a. By Nov 5, we would like to have all interactions with Twitter programmed. This includes receiving and responding to Tweets, parsing Tweets received for parameters, and scraping Twitter.

   b. The minimum outcome will be a Twitter bot that responds to a Tweet with a short rhyming poem influenced by the subject specified in the received Tweet.

   c. Ideally, the tweets that we combine to make poems would be closely related to each other. We would also like to have many different options for the structure of the poem, including a particular rhyme scheme and an internal rhyme option. Additionally, we would ideally like a variety of meter options to be selectable and functional. Lastly, we are also considering making a companion website where longer poems would be posted.