Executive Summary

My project is designed to automate the grading process for a MBA 693R class for which I was enrolled in this semester. Currently there are weekly assignments for each student to do the following:

- Tweet 15 times
- Follow 10 new people
- Write a blog post
- Comment on 3 class members blog posts

Each week the TA must examine each of the students' social media sites to determine if they met the requirements for the assignments. With one button click, my program executes web queries using the Twitter API and html scraping to find the relevant information and makes a report sheet summarizing the fulfillment of the assignments per student.

Documentation

In order for the program to be of any use, the user must supply some information that will later be used in the queries. On the summary worksheet are two example lines of information regarding each students twitter handle and blog url. Assuming this is valid

- When the user presses the "Fetch Stats" button, the runReportButton sub procedure is called. This identifies the first row where a student's info is present and begins consuming and storing the information to local variables.
- Each piece of information is used as a parameter to some other function which in turn returns a value to be placed on the new report. Each time a the program is run, it stores these totals in a worksheet with todays date. If that worksheet already exists, it overwrites the data.
- The first method call is the checkTweets sub procedure, which takes the twitter handle as an argument. This method uses the WinHTTPRequest library to query my server twitter.tengentllc.com. The full path is constructed using the twitter handle as an argument in the url. The server call returns a integer of the total number of tweets for this user. Originally I planned on making a rest request to the twitter api natively through VBA, however, because twitter uses OAuth 1.0 authentication, managing the hashing and everything else needed to make an authenticated request proved to be too challenging and time consuming. Instead I created a simple python server that uses a library called tweepy to handle all authentication and makes the simple request. It then returns the result to the caller—VBA.
- checkTwitterFollowers is the next method called. It also take the twitter handle as the parameter and returns the total number of followers.
- checkBlogEntry is then called and takes the blogURL as well as the blogType (wordpress and blogger are the only two currently supported). Using html

scraping, these methods search for the list of blog posts on the page and identify what the latest post is called. It then returns this value as the entire link to that individual post. Before it finishes it follows this link and finds all the comments made on this post and stores the authors in a global list.

- checkBlogComments accesses the previously mentioned global list of commentAuthors and iterates through each checking to see if it matches the studentName or alias (two parameters). If so it increments a count. This count is then returned.
- All these methods return totals or current state, but for the grading to be useful it need to calculate progress week to week. Each returned value is therefore placed on it's own sheet (the name of the sheet is the date on which the report was generated) for reference. This report's values are then compared with the next most recent report to create a difference. The difference values are then placed in the summary tab. So for example, instead of total tweets per student you have number of tweets since yesterday or last week. If the blog post found is the same as before it simply says, no new blog post, otherwise it displays the name of the blog post.

Unfortunately I was not able to get the blog comments to work completely. All the code is there to run, but for some reason even though I'm accessing the right html tag/classes, the collection returns empty.

Learning

I learned many things during this project. One of which was relearning how great python could be in a jam! I also learned how to scrape HTML using vba and generally cemented my knowledge of the syntax and capabilities of the language. I also learned that there are virtually no references that you can add on the mac version of excel. Windows is much more equipped to handle networking and the like.

Assistance

Aside from Google searches about minimal tasks I did not use any major help. I did at one point consider using a library to query the twitter API from within the VBA program, but the OAuth setup proved to be too much. Instead as mentioned above I used a python library called Tweepy to handle OAuth in the creation of my wrapper api for my twitter api calls.

Write up

- When the user presses the "Fetch Stats" button, the runReportButton sub procedure is called. This identifies the first row where a student's info is present and begins consuming and storing the information to local variables.

- Each piece of information is used as a parameter to some other function which in turn returns a value to be placed on the new report. Each time a the program is run, it stores these totals in a worksheet with todays date. If that worksheet already exists, it overwrites the data.
- The first method call is the checkTweets sub procedure, which takes the twitter handle as an argument. This method uses the WinHTTPRequest library to query my server twitter.tengentllc.com. The full path is constructed using the twitter handle as an argument in the url. The server call returns a integer of the total number of tweets for this user. Originally I planned on making a rest request to the twitter api natively through VBA, however, because twitter uses OAuth 1.0 authentication, managing the hashing and everything else needed to make an authenticated request proved to be too challenging and time consuming. Instead I created a simple python server that uses a library called tweepy to handle all authentication and makes the simple request. It then returns the result to the caller—VBA.
- checkTwitterFollowers is the next method called. It also take the twitter handle as the parameter and returns the total number of followers.
- checkBlogEntry is then called and takes the blogURL as well as the blogType (wordpress and blogger are the only two currently supported). Using html scraping, these methods search for the list of blog posts on the page and identify what the latest post is called. It then returns this value as the entire link to that individual post. Before it finishes it follows this link and finds all the comments made on this post and stores the authors in a global list.
- checkBlogComments accesses the previously mentioned global list of commentAuthors and iterates through each checking to see if it matches the studentName or alias (two parameters). If so it increments a count. This count is then returned.
- All these methods return totals or current state, but for the grading to be useful it need to calculate progress week to week. Each returned value is therefore placed on it's own sheet (the name of the sheet is the date on which the report was generated) for reference. This report's values are then compared with the next most recent report to create a difference. The difference values are then placed in the summary tab. So for example, instead of total tweets per student you have number of tweets since yesterday or last week. If the blog post found is the same as before it simply says, no new blog post, otherwise it displays the name of the blog post.

Unfortunately I was not able to get the blog comments to work completely. All the code is there to run, but for some reason even though I'm accessing the right html tag/classes, the collection returns empty.