Fall 12-6-2017

Social Link

Samy Achour
Chapman University, achou100@mail.chapman.edu

Christina Berardi
Chapman University, berar104@mail.chapman.edu

Paul Harnack
Chapman University, harna100@mail.chapman.edu

Dylan Davis
Chapman University, davis371@mail.chapman.edu

Recommended Citation

Achour, Samy; Berardi, Christina; Harnack, Paul; and Davis, Dylan, "Social Link" (2017). *Student Research Day Abstracts and Posters*. 252.
http://digitalcommons.chapman.edu/cusrd_abstracts/252

SocialLink

Achour, S., Berardi, C., Davis, D., Harnack, P.

Central Research Question

Employers often trawl applicants' social media presences for objectionable behavior. We are creating a business facing application that streamlines the process, combining API data from LinkedIn with that of Twitter and cross referencing them. The difficulty comes down in aligning user profiles with different sets of attributes. For example, given an arbitrary applicant the LinkedIn API might give current city, university, and work history but Twitter might just have the current city and a single item in work history. Being able to match these values and pinpoint specific profiles will take some time and a finely tuned algorithm. We plan to have a solution that will cover at least 80% of the cases, given that certain users don't have a social media presence or have obfuscated their account details (another issue, applicants changing their Twitter names). This means that for 10 given searches at least 8 queries will return results containing the correct user profile. The final product will have a web-based UI that displays the search results using python to make the API calls and process the data. The search results will display the top five closest results that match the search. Relevant research includes the field of information retrieval.

Background Information

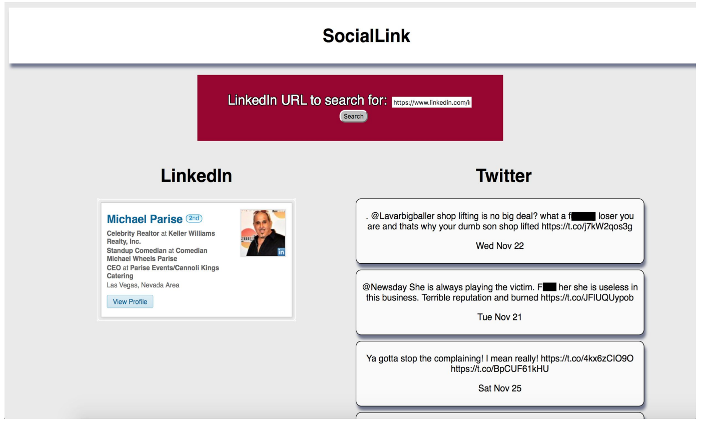
This project would not have been possible without the use of Application Programming Interfaces (API). Software applications often have APIs for developers that use function calls to retrieve and pass information between the existing software application and the new application. Through the use of APIs, a developer can connect their own code to an existing software. In this project, we attempted to use the LinkedIn and Twitter API functions to retrieve a person's Tweets and LinkedIn profile information.

Materials and Methods

Research was prepared initially on the Facebook and LinkedIn APIs. It was then decided that Twitter would be used instead of Facebook because it allowed more access to developers. Then authorization was obtained for both Twitter and LinkedIn APIs. All function calls were created using Python and the web framework was done using Django, used for creating websites.

Results

The application combines two social media platforms, LinkedIn and Twitter. The user's current LinkedIn profile is obtained, along with user's tweets. The user's tweets are scraped and sorted by objectionability. The web UI then displays the objectionable tweets along with their LinkedIn profile picture, employment history, location, and a link to their LinkedIn profile.



SocialLink

LinkedIn URL to search for: https://www.linkedin.com/i

Search

LinkedIn

Michael Parise 2nd

Celebrity Realtor at Keller Williams Realty, Inc.
Standup Comedian at Comedian Michael Wheels Parise
CEO at Parise Events/Cannoli Kings Catering
Las Vegas, Nevada Area

View Profile

Twitter

. @Lavarbigballer shop lifting is no big deal? what a f[redacted] loser you are and thats why your dumb son shop lifted https://t.co/j7kW2qos3g

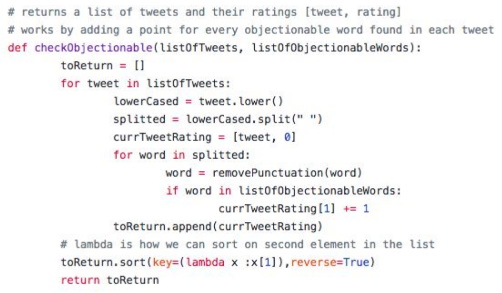
Wed Nov 22

@Newsday She is always playing the victim. F[redacted] her she is useless in this business. Terrible reputation and burned https://t.co/JFIUQUypob

Tue Nov 21

Ya gotta stop the complaining! I mean really! https://t.co/4kx6zCIO9O https://t.co/BpCUF61kHU

Sat Nov 25



```
# returns a list of tweets and their ratings [tweet, rating]
# works by adding a point for every objectionable word found in each tweet
def checkObjectionable(listOfTweets, listOfObjectionableWords):
        toReturn = []
        for tweet in listOfTweets:
                lowerCased = tweet.lower()
                splitted = lowerCased.split(" ")
                currTweetRating = [tweet, 0]
                for word in splitted:
                        word = removePunctuation(word)
                        if word in listOfObjectionableWords:
                                currTweetRating[1] += 1
                toReturn.append(currTweetRating)
        # lambda is how we can sort on second element in the list
        toReturn.sort(key=(lambda x :x[1]),reverse=True)
        return toReturn
```

Twitter Code Example From Repository

Discussion

The accessibility and usability of the Twitter API was much more 'open' in comparison to that of LinkedIn. The progress of the application was hindered due to LinkedIn's not so recent lockdown. LinkedIn is only allowing full API access through it's partnership program, and this involves applying and being hand selected by LinkedIn. Twitter's API was much more extensive in its reach into Twitter and easier to access for developers.

Future Work

There is still far more work to be done on this application. While LinkedIn and Twitter are two main social media platforms, employers often want to do a more thorough search. The addition of APIs such as Facebook and Instagram would allow employers to further vet applicants. This would provide companies a cost effective way to sift through many applicants before the interview process. Another aspect of the project that could be improved is the design of the User Interface.