EECS 582 Team 12 – Final Design Report

Matt Bailey, David Cuellar, Jacob Long, Manan Shah

2/4/2018

Team Name: DeepHate.py

Team Members and email addresses:

David Cuellar david.cuellar@ku.edu,
Matt Bailey m229b487@ku.edu,
Manan Shah m406s889@ku.edu,
Jacob Long j930l636@ku.edu

Contact: Matt Bailey m229b487@ku.edu

Project Description (150-250 words)

Predictive Behavioral Modelling software that will integrate deep learning neural networks (MIT's DeepMoji and IIT-Hyderabad's Deep Learning for Hate speech identification network) with analysis of given user's relationship networks from twitter to predict likelihood that user may engage in discriminatory behaviors and risk adverse outcomes to their employer. That likelihood will be represented by a "behavioral profile" returned to the customer, who provides input.

The end result is a web-based application intended for use by HR departments and hiring managers (herein referred to as the customer) to screen the social media presences of prospective employees for indicators of hate speech/latent hateful ideology that may arise in unbecoming circumstances both within the company or in public view. By providing an automated, quantitative means for analyzing the social media of prospective hires, the customer would insulate themselves from legal risks of discovering protected information via social media screening, as would occur if such screenings were performed manually by the customer during pre-employment background checks.

Project Milestones

· 3-5 specific and measurable objectives per semester for first & second semester

Fall Semester.

1. Complete Dynamic Data-set generation scripts -- November 2nd
2. Finalize mathematical model used to make prediction from raw input data (Sets of tweets, user's relationship graphs, facebook object graph) -- November 15th.
3. Initial web page diagramming -- December 3rd
4. Code review of DeepMoji and Deep Learning for Hate Speech ID LSTM implementations -- December 10th
5. Initial Backend diagram/design completion -- ~December 15th

Spring Semester:

1. Complete initial deployment of back-end -- February 21st
2. Complete front-end design -- January February 21st
3. Connect Front and Back ends -- March 28th
4. Initial testing and model validation via front-end -- April 4th
5. Complete final form front end GUI.

· Both implementation and documentation milestones

The Gantt Chart is shown in Figure 1 (1.0 and 1.1), with the complete chart available in its entirety at the following URL:
https://www.tomsplanner.com/public/eecs581-team12

The Gantt Chart is password protected – see proposal submission for password.

Figure 1 -- Gantt Chart

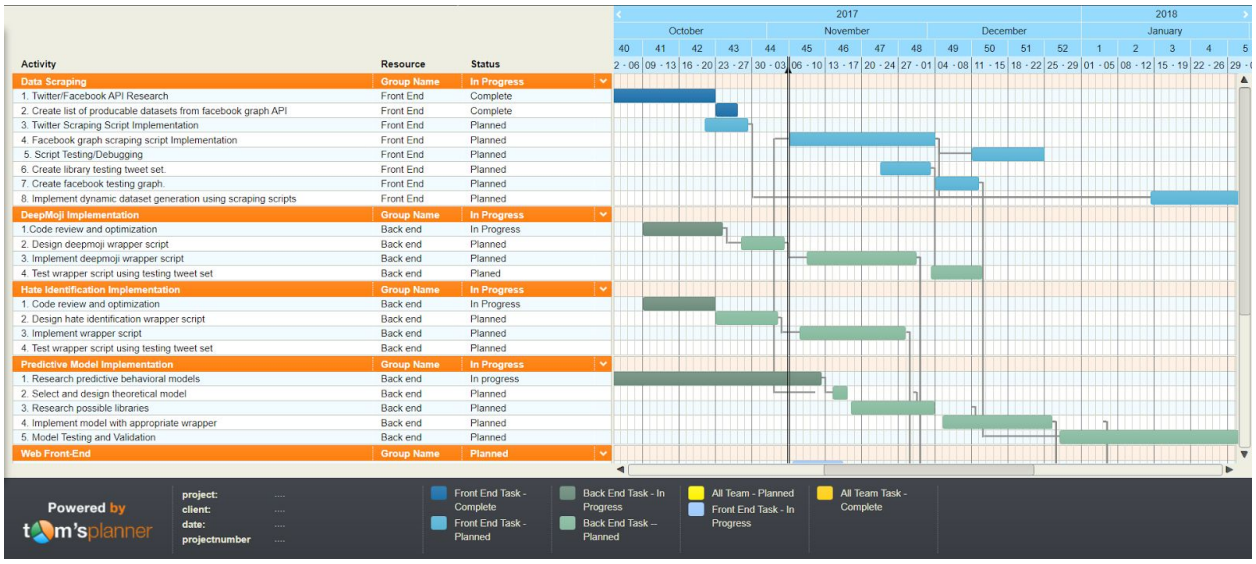


Figure 1.1 -- Gantt Chart continued:

Final Project Design

UnMask behavioral analysis tool is best understood when broken down in terms of software. The complexities of human behavior and natural language necessitate the use of resource intensive techniques from artificial intelligence and data mining. This complexity stems from two sources – the very large number of variables that could potentially affect human decision-making both in general and in the interpersonal domain, and the difficulty disambiguating sentiment from language due to things like sarcasm, slang, or other forms of obscured semantic embedding that eludes accurate description and classification in many natural language processing approaches. To deal with this complexity, DeepHate aims to create a “behavioral profile” of a prospective job candidate by gathering all publicly available tweets by the candidate and analyzing them using MIT’s DeepMoji pre-trained neural network, and IIT Hyderabad’s Hate Speech Identification neural network; the output of each of these algorithms

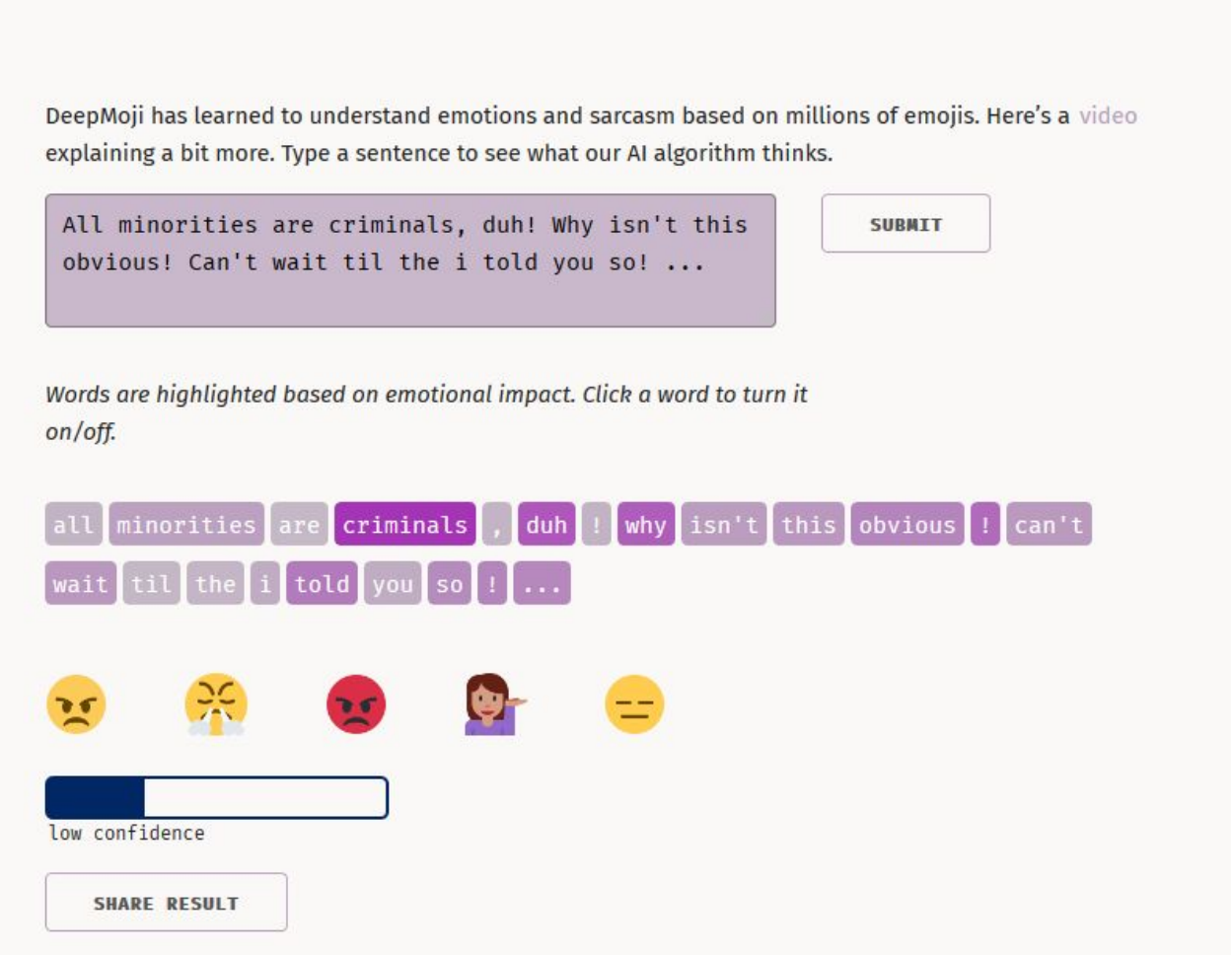
will be fed into the predictive model to establish a sense of how the user views or responds to racially inflammatory events or other flashpoints of social division.

From the top-level description of UnMask, it seems as though the involved algorithms could become very resource intensive. Although there initially was a plan to acquire custom hardware in order to create a custom server to host UnMask, the plans have been altered and the tech stack will no longer include graph analysis with the Facebook API. Therefore team 12 has elected to use Amazon Web Services to host the UnMask web application.

In terms of software, DeepHate consists of seven primary components: web application interface, backend handler, data scraping module, DeepMoji neural network module, hate speech ID neural network module, and the predictive model module. The web application interface is the front-end through which the end user passes a job candidates information (Name, Location, Phone Number, e-mail address, twitter handle, and education) and receives the behavioral profile generated by DeepHate. The backend handler deals with the generation and pre-processing of candidate Twitter datasets by passing the candidate's information to the data scraping module. With these datasets generated, the tweet vector of length n is passed to DeepMoji and Hate Speech ID. DeepMoji analyzes individual tweets and encodes the sentiment of the tweet in the form of a vector of 5 emojis using a long short-term memory (LSTM) neural network, a type of recurrent neural network architecture; the DeepMoji module would take the tweet vector as input and return an n x 5 matrix of integers, with each integer being mapped to a corresponding emoji. Hate Speech ID takes a tweet and classifies it as racist or sexist, thus the output of the Hate Speech ID module would be a 2 x n matrix of integers, with the first entry being 1 if the tweet is racist, 0 otherwise and the second entry being 1 if the tweet is sexist, 0 otherwise.

The prototype's predictive profile generator's design is still in progress; however, the overarching idea can best be characterized by an analogy. Imagine you see a tweet as follows: "All minorities are criminals, duh! Why isn't the danger obvious to y'all?! Cant wait to say I told you so! … " From verbal inspection, this could be either a genuine racist sentiment, or a sarcastic statement from a minority, mocking how they are represented at a particular cultural moment or in a specific situation. The resulting emoji output vector from DeepMoji is shown in figure 2.

Figure 2 – Deep Moji Tweet Sample

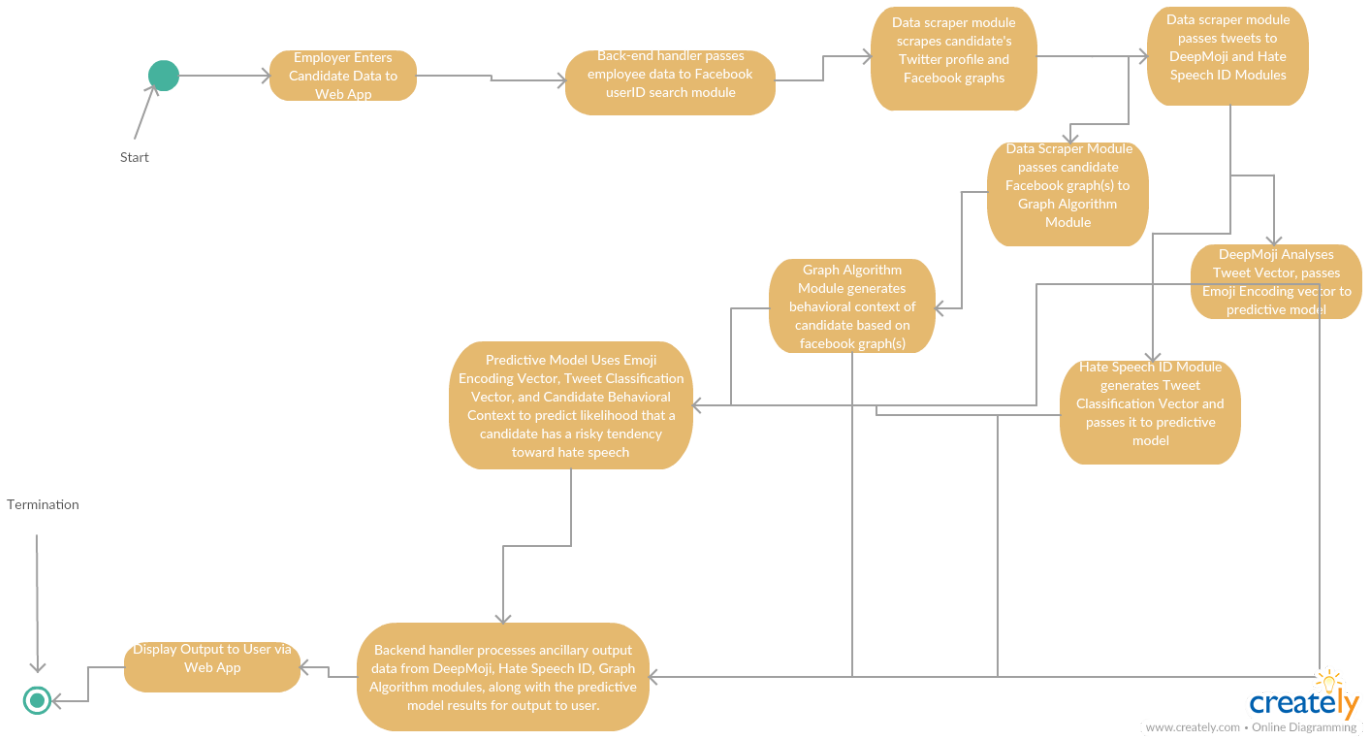


The emoji vector and confidence level given here do not accurately explain which of the two possible cases informs the tweet author's sentiment. The first three emojis clearly correspond with the sort of frustration felt by either a person that feels dismissed due to their views on race (the racist sentiment) or a person that is demoralized by discriminatory social elements to the point of sarcasm (the sarcastic minority sentiment). The final two emojis correspond to the sarcastic subtext of both possible sentiments. This exemplifies an instance of noisy data that DeepMoji's creators describe as difficult to accurately classify and use. Through combining the DeepMoji data with IIT Hyderabad's data, a clearer profile can be produced.

The output of the behavioral model would be combined, by the backend handler, with ancillary output data from DeepMoji and Hate Speech ID, such as the number of racist/sexist tweets, the number of racist/sexist tweets paired with a positive/approving emoji encoding vector, and so on, to generate the behavioral profile to be given to the end user via the web portal.

The relationships between such components and their functions are outlined in the activity diagram in Figure 4.

Figure 3 – Activity Diagram



Note: Due to changes in design, activities shown to involve Facebook / Graph Analysis may not be present in prototype.

The initial graphical user interface for UnMask has been designed as follows:

Figure 4 - GUI mockup

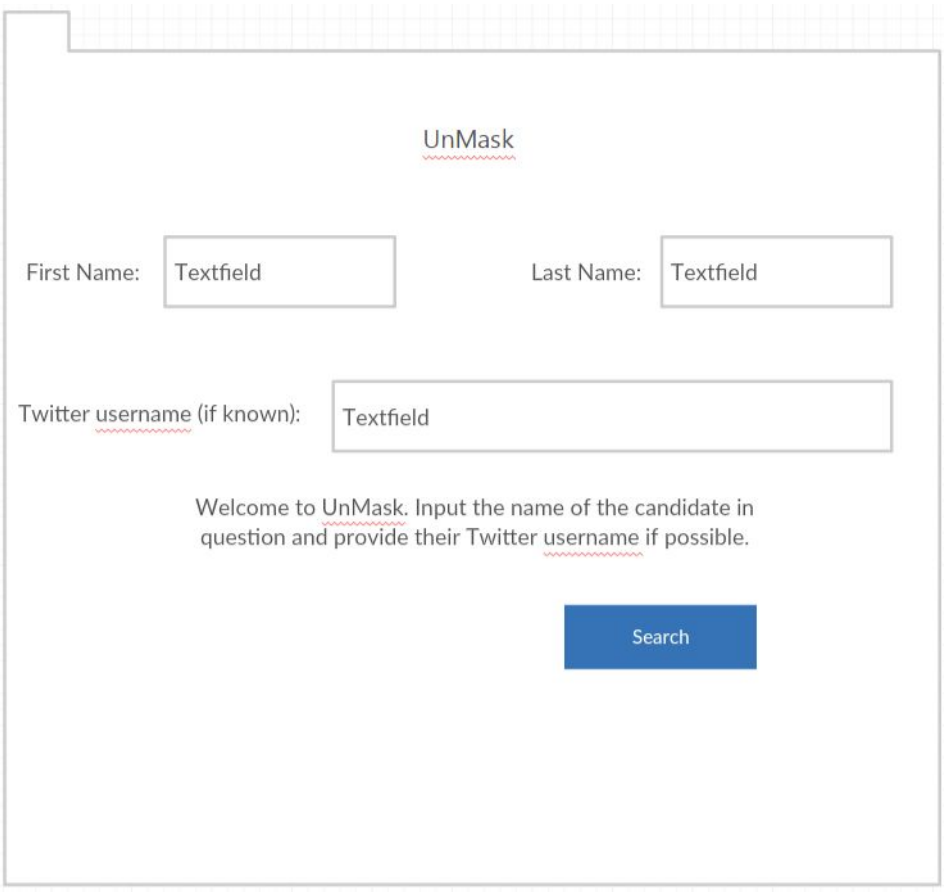
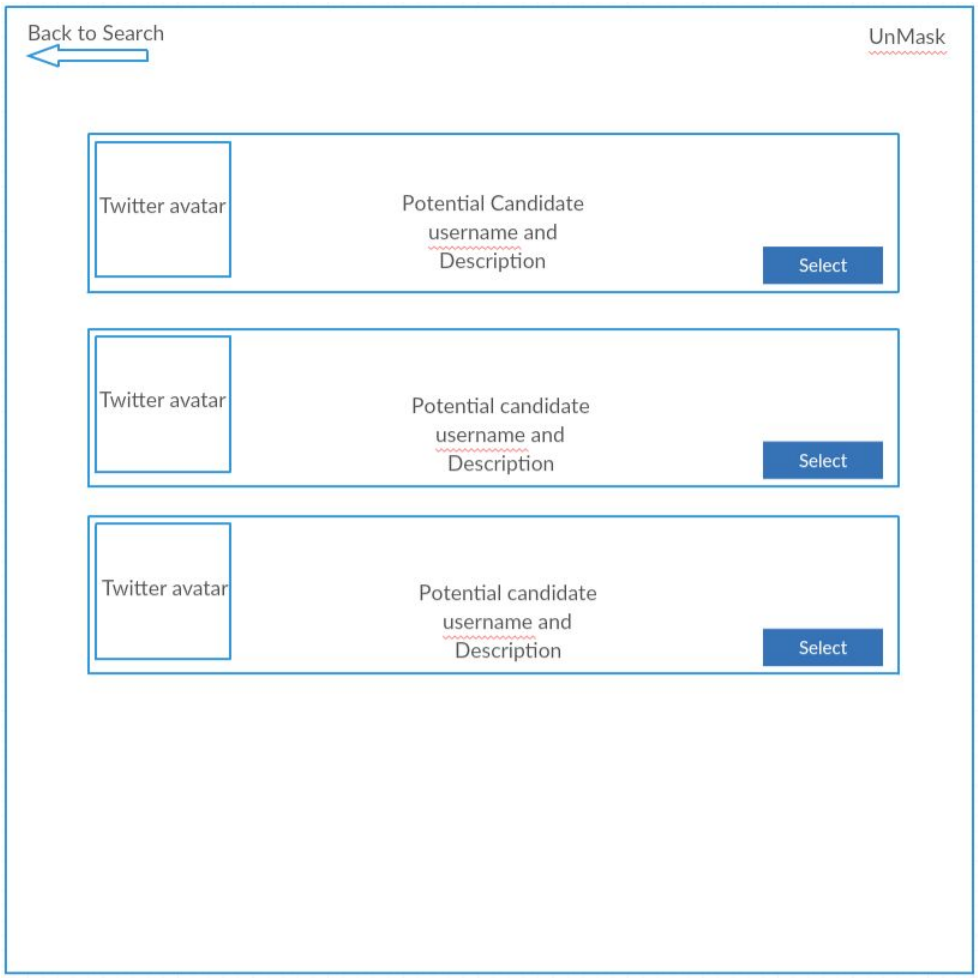


Figure 4 is the template with which team 12 is creating the web interface. This particular mockup is the first web page that the end user will see upon entering the web application. The system will be designed to receive a person's first and last name and/or receive the Twitter username of the person in question if the user happens to be privy to that information. The search button will utilize an on click listener that runs a function to retrieve Twitter usernames that are potential matches to the desired candidate. The following page is a prototype mockup of what will be returned upon clicking the search button:

Figure 4.1



A list of names that match the customer's search, formatted like the above template is returned on the new page. On the left hand side of each of the candidate boxes, the customer should be able to see the avatar of the respective potential candidate. The rest of the box will be populated with the candidate's username and description if that information can be obtained. Note: the above designs are mere mockups, and the prototype will of course be more aesthetically pleasing.

Upon selection of a candidate, their information is passed to the back end, where the aforementioned neural networks will analyze strings of text from their tweeting history. Each of the neural nets will return their respective outputs to the wrapper function, which will then build the behavior profile to be returned to the customer.

Once the analysis is complete, a third page will load. This page will contain the entire behavioral profile, which will feature a numeric coefficient ranging from 1 to 100. Higher coefficients will indicate greater proclivity to submit insensitive posts to social media. The profile generator is still a work in progress, and as such there is no accurate mockup to present at this

time. If there is time, a “confidence coefficient” similar to the confidence level section of DeepMoji may be implemented as well.

Ethical and Intellectual Property Issues

The intellectual property issues of DeepHate remain to be seen; as of writing, both DeepMoji and Hate Speech ID neural network libraries are licensed under licenses allowing for unrestricted public use in other products or capacities. This may change depending on what is used to implement both the graph analysis algorithm as well as the predictive behavioral model.

The ethical implications of such a software tool are a bit more complex. Some could argue that it persecutes thought crimes on behalf of prospective job candidates. However, the mathematical rigor of DeepHate and other behavioral models mitigates claims of persecution. Further, such claims ignore the fact that all material gathered and used by DeepHate results from an active, conscious decision by the job candidate to associate themselves with a particular type of web content; absent a compromised social media presence, an individual reacts to tweets, crafts tweets, and responds to Facebook objects in ways that align with how they wish to represent themselves. Thought crimes in the traditional sense are not associated with behaviors that consciously curated like the social media presence of modern professionals.

Project Budget

1. Monthly amazon web service -- in the event that server construction is too cost-prohibitive, AWS may be necessary, assuming that AWS instances exist that have compatible versions of tensorflow -- more research is needed to project monthly compute costs.

Meeting time with TA (Amir):

Monday lab - 2:15 pm

Work Plan

Front End/Web App

Jacob, Manan

Neural Nets

David, Matt.

Github link:
https://github.com/dsca1729/581_team12_17-18

Change log

Changes made to initial plan on 2/2/2018 --

Project description changed. Includes GUI details.

Project milestones changed. Dates shifted.

Changes made to initial plan on 2/4/2018 --

Changes from **preliminary project design** to **final project design** --

As of the creation of this document, the team is no longer planning on implementing the graph analysis portion of the preliminary project design. Items involving this have been edited / removed. Details on the GUI have been added to the Final Project Design section.