BYEas Final Report

Due: May 5, 2020
Senior Design
Chloe, Jen and Monica

Table of Contents

Section 1: Work Distribution

Monica

- Web Scraping
    - Web scraper works using the BeautifulSoup python library
    - Parses the html of the text and gets the article by finding the text within <p> tags
    - Checks if words are valid by making a call to the database
    - Returns a list of words in the article
- API calls
    - Swagger used to route the calls
    - Created a custom API using swagger.yml file
    - Use a POST request to run the python algorithm through our dashboard website
    - Then the data that the python returns is a json file that contains the bias data
    - We parse the json file using javascript and display it on the website
-

Jen

- BYEas Website/ Dashboard
    - HTML, CSS, and JavaScript for the front end of the Website
    - Apache server setup to host the website
    - Copy and design of website
- BYEas webplugin
    - HTML, CSS for the front end of the webplugin
    - JavaScript for handling communication and data
    - Communication with the website being examined, in order to query the dom
    - Communication between website and web plugin
    - Setup of listeners for communication

Chloe

- Algorithm for bias scores
    - Python for the algorithm and SQL to query the database
    - Used word embeddings from the glove algorithm to calculate the bias scores
- Database
    - Python to populate the database and SQL to create and update the database.
    - Stored the word embeddings and cosine similarity values

Section 2: Overview

Diversity begins with the material that we are exposed to! Studies have shown that people who are exposed to biased content may be subconsciously internalizing these biases. Whether it is a woman not applying to a job in engineering or a man feeling like the woman belongs in the home, these biases are holding us back from making progress with reducing inequality in our society. In the current day, we have the opportunity to harness the power of machine learning to better ourselves.

BYEas is a lightweight, easy to use web-plugin that sheds light on potential biases in the sourcers we read everyday. It does this by presenting a bias score for an article on the internet and highlighting the sentence in that article that has the highest score. In order to use BYEas, the user downloads the web extension, and browses the web for a relevant article. After the article has been selected, the user has the option to open the BYEas web extension and click "Compute Bias Score". After the score is calculated, this will bring the user to a dashboard with explanations of the bias scores and a visualization of the data gathered.

The bias detection algorithm is built from the GloVe word embeddings developed by Stanford University. GloVe is an unsupervised learning algorithm that obtains vector representations for each word based on a common crawl of the internet. These word embeddings are especially interesting because the distances between the words relate to their similarities, and these word embeddings even allow us to do computations with words. Given this ability, we are able to create an algorithm that compares the text in an article to words that are related to the biases that we are testing. These words come from the Implicit Association Test. We use cosine similarity to determine how similar words are and effect length to determine the magnitude of the similarity. The scores with higher values out of one hundred are a more significant difference. The scores with the lower values out of one hundred have a less significant difference. For example, if the score for math and female is 90% this means that they are very different with respect to the article in question. If examining the same pair of biases and the score of 5%, this means that they are fairly similar with respect to the article being examined. Therefore, a lower score indicates a higher bias.

Section 3: Project Website

http://wright.seas.gwu.edu/public_html/sd-20-hutchins-kavathekar-wright/BiasDetectionWebExtension/templates/about-us.html

Note: server must be running to access the site (ssh jen@wright.seas.gwu.edu, password jen_seas)

Section 4: Libraries and APIs

jQuery

w3schools color library

GloVe

BeautifulSoup

Chrome Developer API

Flask

Swagger

Connexion

Ajax

Section 5: Technical Overview of Project

Our project is coded in python, javascript, and HTML. We use various libraries to run both the front end and back end. The user first interacts with the web plugin, which is made in javascript. The button press in the web plugin triggers a custom API that we created using swagger and ajax. Ajax allows for us to make a send and request data from our python program. Swagger is used to route our POST call to send the url from the webpage into the python file. Once the python program receives the URL, it uses BeautifulSoup to parse the contents of the webpage.

BeautifulSoup returns a list of all the words used in the article. Then, the python program queries our SQLite database that contains the GloVe word embeddings. It checks that each word from the article is a valid word in the database, therefore eliminating meaningless strings that may have been contained in the html. The python program uses these word embeddings and effect lengths to calculate a bias score for each of the biases we are testing. Then, these numbers are sent to a json file. Ajax and the swagger API accesses the json file, and stores the bias numbers as javascript variables. Finally, the javascript variables are displayed in an HTML table on the dashboard page.

Section 6: Weaknesses and Limitations

- Section 6: One "if I had to do this again" paragraph from each team member that contains *technical* lessons learned, and what would you have done differently if you know what you know now (but with exactly the same project). Examples could include using a different API, some other type of restructuring etc.

Monica

I think that we all went into this project with very little knowledge of servers, and the way different parts of a web plugin need to talk to each other in order for data to seamlessly pass between them. We all had background knowledge of machine learning, which is why we started by creating our algorithm in python. Had we known more about web plugin communication, we might have rethought how to create the backend of our project. The python to javascript to web extension communication stalled us for many months, and we never really discovered a clear solution. Our solution in the final product is to avoid displaying biased data on the plugin and instead to open a website. However, I know that there must be a way to have machine learning in the backend of a web extension, and our team might have made a more cohesive product if we had the time to restructure our backend.

Jen

One of the most challenging parts of our project was communication between the different modules we created. One reason why this was difficult was because the root of our project was a chrome web extension which had some security requirements that made communicating with servers difficult.

Another thing that would be helpful would be if we had taken more time to ensure that the libraries and tools we were using fit together well. For example using a python server made our project slightly more difficult than if it had been written in pure javascript. Lots of these issues we just learned because it was our first time planning out something like this.

Chloe

One of the most important things that I learned from this project was just how important it is to really plan how you are going to structure your project when you first start it. Although we did do this, there were some parts of the project that we did not have as much knowledge about, such as how to create our own web plugin, which meant that at the beginning we had very little knowledge of what all it would require for us to achieve. Therefore, we did not sufficiently plan for how this would be structured in our project at the beginning. This led to problems when we eventually transferred our project onto the seas server. For me specifically, I found that the database I had initially chosen to use may not have been the best for use on the server. Even more so, we had split up how we were going to complete the project and as each person worked on their own part, I think we could have communicated more about how exactly we were doing it and how our parts related to all the other parts of the project.

Section 7: Next Steps

There are several steps involved inorder to get our project running. First, clone the code from our github. Next, getting the server running. You can ssh in to our server by using the following commands, ssh jen@wright.seas.gwu.edu, and when prompted, the password is jen_seas. From there you can navigate to our repository, the path is /var/www/html/public_html/sd-20-hutchins-kavathekar-wright/BiasDetectionWebExtension and the commands to start our python server are, source env/bin/activate to kick off the virtual environment, and python3 server.py to actually start the server. From there, we can set up the chrome extension. If you go to chrome://extensions/, you are able to 'load unpacked' which will allow you to choose the file from the repo that is titled BiasDetectionWebExtension. Then you should see a B pop up where the chrome extensions do and when clicked, a dialog box that asks about running the algorithm should appear. Pip install any libraries that are missing and we used python3.

The really tricky part of our project was communicating with so many different modules, consolidating these would be extremely helpful. In terms of ideas for future projects, instead of spending so much time working on the communication and web plugin aspect, we think it could be cool to make a website where a user submits a URL and focus most of the brain power on developing an efficient and accurate algorithm that has more scale. Also, perhaps having more data processing and analyzing articles in terms of others could be a cool route to take. Tackling and identifying more biases is also another potential way to expand the project.