To identify gender bias in science, technology, engineering and mathematics (STEM) in language data and prototype a method to identify the source of this bias, in order to address it. Rationale There is a gender imbalance in STEM careers (Robnett, 2016). In 2016, women were particularly underrepresented in IT and engineering, comprising 28% and 12.4% of the workforce, respectively (Australian Government, 2019). Because gender disparities are rooted in cultural norms, values and discourse (Robnett, 2016), it is expected that this imbalance should appear as bias in language data (Sun et al., 2019; Zhao et al., 2018). This bias may even be amplified in any predictions made based on language data (Sun et al. 2019). Natural Language Processing (NLP) is the application of data science and analytics to human language (Chowdhury, 2003). Language comprises some of the richest and most complex data available, and can provide nuanced insights to virtually any topic, in terms of human thoughts and reactions to different phenomena (Chowdhury, 2003; Hirschberg and Manning, 2015). Unsurprisingly, the rise of social media has provided a bounty of data for NLP applications. In 2020, 500 million Tweets were published daily on social media site Twitter, providing an extensive range of language data on a multitude of different topics (Sayce, 2020). Social media platforms, such as Twitter, behave as a largely uncensored, global-scale historical record from the perspective of everyday members of society (Neethu, 2013), not just the powerful and privileged, as has been the case in the past (although, importantly, and in line with the topic of this report, Tweets written by people of colour are more likely to be flagged as offensive and removed by machine learning algorithms (Wei, 2020)). Because this investigation seeks to research implicit gender bias in STEM-related topics, and because gender bias is a social issue, it is sensible to look for data where people produce a high volume of text. Because Twitter can be treated as a repository of people's opinions and attitudes, and therefore their personal biases, it naturally harbours the biases of society in general, making Twitter an ideal source of data for sentiment analysis (Neethu, 2013). Scraping data from Twitter, using specific search terms, allows targeted collection of information and ensures that the data collected are relevant to this investigation. For these reasons, Twitter is the sole website from which data were extracted. In [3]: # Import packages import pandas as pd import numpy as np from getpass import getpass from time import sleep from selenium import webdriver from selenium.webdriver.common.keys import Keys from selenium.common.exceptions import NoSuchElementException from selenium.common.exceptions import StaleElementReferenceException from webdriver_manager.chrome import ChromeDriverManager import csv import re import nltk import string from nltk.stem import WordNetLemmatizer import matplotlib import matplotlib.pyplot as plt from matplotlib.pyplot import figure from matplotlib.pyplot import xticks from matplotlib.pyplot import yticks import matplotlib.ticker as mtick from matplotlib.ticker import StrMethodFormatter Methodology This report will analyse language data bias in STEM, with respect to gender, using **Python version 3.9.0**. The methodology is as follows: Scrape tweets about STEM, relating to people. Clean the data so it can be used in NLP tasks. Investigate the number of STEM-related tweets containing female pronouns vs male pronouns. Report findings from a preliminary exploration of these data. Identify the source of bias using a combination of supervised classification and sentiment analysis. Although Twitter has an API which can be used for academic purposes, in line with the task requirements Selenium was used to simulate a search within a browser and data were collected using **xpath()**. The search term was (science OR scientist OR technology OR technologist OR engineering OR engineer OR math OR mathematics OR mathematician) (she OR her OR he OR him) -filter: retweets. Webscraper The data from Twitter were complicated to extract because the site is complex and the content is continuously updating. Due to the perpetual scrolling nature of the site, it was decided that the best way to scrape Tweet data (other than using the API) would be using the **Selenium** package. Selenium works by automating web browser activity, replicating functions such as opening a browser, clicking on links and scrolling pages (Silman, 2019). The package requires installation of a driver to interface with a browser (Chrome was used in this case). Selenium was used to open, log in, navigate and enter a search term to scrape Tweet data while scrolling downwards, allowing new Tweets to appear. The xpath() function was used to select specific data from each Tweet. Username, handle, Tweet and the number of replies, retweets, likes and the post date were retrieved. Figure 1 shows the webscraper in progress. y (20) (science OR technology OR ∈ x + ← → C 🔒 twitter.com/search?q=(science%20OR%20technology%20OR%20engineering%20OR%20math)%20(she%20OR%20he)... 😉 🛠 😩 🚼 Chrome is being controlled by automated test software (science OR technology OR engineering OR math) (she OF Trending in Australia #letter Latest People 0 Trending in Australia A Georgia congressman says he fears additional Race 1 residents could cause Guam to topple. Later, he sa... 56.9K Tweets Bustle . April 22, 2021 20+ 44 17 **How Frankie Jonas Found His Voice On** TikTok wandy your meow meow @laughingobamaa5 · 14m TODAY MY **MATH** TEACHER COMPLIMENTED MY HAIR EVEN THOUGH THE DYE IS FADING AWAY OML AND HE RUNS THE PRIDE CLUB AT MY SCHOOL \Box ₹ GUYS DOES HE KNOW IM FRUITY Who to follow \blacksquare ALIENS ARE GAY 👽 @wild_forms · 16m Martha has tried to take him to listens as a kid but he wants to reading Department ... 8 @DAFQId math books and sc-fi instead Imaooo Promoted 17 \bigcirc 6 \odot NASA Earth Show this thread Follow @NASAEarth NASA Mars 🤣 soo I asked the guy, what year was he born bc I assume he's a year younger Follow for some reason then he RESPONDED with a math equation . Terms of Service Privacy Policy Cookie Policy Ads info More ... Figure 1: The webscraper in action After identifying issues with the running of the webscraper, modifications were made to the code to catch two types of exception: **NoSuchElementException**, which can occur when the scraper attempts to access an element before the page has fully loaded, and **StaleElementReferenceException**, which occurs when some element is temporarily inaccessible. Elements were identified and extracted using the HTML inspect navigation panel corresponding to the page in the Chrome browser (*Figure 2*): Elements Console Sources Network Performance Memory >> 13 🛕 1 F KUIV CIASS= CSS-IUDJC4H > ... K/UIV> flex ▼<div class="css-1dbjc4n"> flex ▼<div class="css-1dbjc4n"> flex ▼<div lang="en" dir="auto" class="css-901oao r-18jsvk2 r-1qd 0xha r-a023e6 r-16dba41 r-rjixqe r-bcqeeo r-bnwqim r-qvutc0"> <span class="css-9010a0 css-16</pre> my406 r-poiln3 r-bcqeeo r-qvut c0">Today I witnessed a 7th grade == \$0 <span class="css-9010a0 css-16</pre> my406 r-poiln3 r-b88u0q r-bcqe eo r-qvutc0">math ▼<span class="css-9010a0 css-16 my406 r-poiln3 r-bcqeeo r-qvutc " teacher engage her students in ways that had me in awe. She had her students engaged from the moment they walked in until they left at the end. " ▶ ... ▶ ... </div> </div> <div class="css-1dbjc4n">...</div> ▶<div aria-label="1 Retweet, 6 lik es" <mark>role</mark>="group" <mark>class</mark>="css-1dbjc4n ▼ Figure 2: Tweet withn Inspect panel In [2]: # Function to scrape Twitter using Selenium def get_tweet_data(card): #Extract tweet data #username username = card.find_element_by_xpath('.//span').text except NoSuchElementException: except StaleElementReferenceException: return #twitter handle handle = card.find_element_by_xpath('.//span[contains(text(), "@")]').text except NoSuchElementException: except StaleElementReferenceException: return #tweet text try: comment = card.find_element_by_xpath('.//div[2]/div[2]/div[1]').text responding = card.find_element_by_xpath('.//div[2]/div[2]/div[2]').text text = comment+responding except NoSuchElementException: return except StaleElementReferenceException: return #reply count comment = card.find_element_by_xpath('//div[@data-testid="reply"]').text except NoSuchElementException: except StaleElementReferenceException: return #retweet count try: retweet = card.find_element_by_xpath('//div[@data-testid="retweet"]').text except NoSuchElementException: return except StaleElementReferenceException: return #likes like = card.find_element_by_xpath('//div[@data-testid="like"]').text except NoSuchElementException: except StaleElementReferenceException: return try: #post date date = card.find_element_by_xpath('.//time').get_attribute('datetime') except NoSuchElementException: return except StaleElementReferenceException: return tweet = (username, handle, text, comment, retweet, like, date) return tweet # create instance of webdriver driver = webdriver.Chrome() driver.get('https://www.twitter.com/login') In [3]: #navigate to twitter and login username = driver.find_element_by_xpath('//input[@name="session[username_or_email]"]') username.send_keys('laura.vodden@outlook.com') mypassword = getpass() password = driver.find_element_by_xpath('//input[@name="session[password]"]') password.send_keys(mypassword) password.send_keys(Keys.RETURN) In [5]: #find search input and search for term #navigte to 'explore' tab driver.find_element_by_xpath('//a[@data-testid="AppTabBar_Explore_Link"]').click() search_input = driver.find_element_by_xpath('//input[@aria-label="Search guery"]') search_input.send_keys('(science OR technology OR engineering OR math OR mathematics OR scie ntist OR technologist OR engineer OR mathematician) (she OR her OR he OR him) -filter:retwee ts') search_input.send_keys(Keys.RETURN) #navigte to 'latest' tab driver.find_element_by_link_text('Latest').click() # get all tweets on the page data = []tweet_ids = set() last_position = driver.execute_script("return window.pageYOffset;") scrolling = **True** while scrolling: page_cards = driver.find_elements_by_xpath('//div[@data-testid="tweet"]') for card in page_cards[-15:]: tweet = get_tweet_data(card) if tweet: tweet_id = ''.join(tweet) if tweet_id not in tweet_ids: tweet_ids.add(tweet_id) data.append(tweet) $scroll_attempt = 0$ while True: driver.execute_script('window.scrollTo(0, document.body.scrollHeight);') curr_position = driver.execute_script("return window.pageYOffset;") if last_position == curr_position: scroll_attempt += 1 #end of scroll region if scroll_attempt >= 3: scrolling = **False** break else: sleep(2) # attempt to scroll again else: last_position = curr_position #save tweet data with open('stem_tweets_2.csv', 'w', newline='', encoding='utf-8') as f: header = ['Username', 'Handle', 'Text', 'Comments', 'Retweets', 'Likes', 'Date'] writer = csv.writer(f) writer.writerow(header) writer.writerows(data) Viewing the scraped data Once the scraper had collected all the data, the information was saved to a .csv file. *Table 1* shows the resulting dataset comprised over 34,000 records of STEM-related tweets from April 14, 2021 (the day of collection). In [3]: # Ipmort data and view dataframe head tweets_df = pd.read_csv("stem_tweets.csv", dtype=object) tweets_df = tweets_df.fillna(0) tweets_df.head(10) Out[3]: Username Handle Text Comments Retweets Likes Date Blackwell is great. As a 2021-04-Angi @AngiMaryssa 0 lawyer, you have to b math dumb af, if y=mx+b then 2021-04-Señor Luis (1LUV) 1 @thiccbb69 14T20:32:24.000Z y=tf don't she lo... Wow, he was a hell of an @JaredEHalpern Jared Halpern 14T20:32:23.000Z engineer --Grant Imah... Matt Kelley is so bad he thinks 2021-04-**Bob Brigham** @BobBrigham 0 0 14T20:32:10.000Z coronavirus is... Replying to 2021-04-AlaskanTzar @AlaskanTzar \n@science_bradyBut he 14T20:31:58.000Z 2021-04that math test i just took was word vomit (erika) @vomit_bestie 0 0 14T20:31:55.000Z my absolute bit... Thomas Howard Riley MAD Replying to 2021-04-SCIENTIST of EPIC @ornithopteryx \n@R_M_AndrewsOMG that 14T20:31:43.000Z **FANTASY** is so funn... 2021-04-Replying to \n@Mhal2005\n 7 m d ryan @hawkdynasty526 0 14T20:31:40.000Z and \n@Kr8ckenyou're... 2021-04-Replying to \n@716babe\n Stephen @Steve_ForChange 14T20:31:33.000Z @StephieKonicki\n and... Replying to \n@shaTIRED5th 2021-04catlady @suchmeerkat 0 14T20:31:18.000Z grade he had everyo... **Data cleaning** There were several necessary steps involved in preparing the Tweet data for analysis using NLP methods. First, user handles were removed using a lambda function targeting the @ symbol before user handles. After this, the remaining punctuation, special characters were removed. Furthermore, a new column was generated with values 'M' or 'F', corresponding to male or female, depending on which pronouns were present in the original Tweet, and a final 'Topic' column containing the relevant STEM field. The resulting dataframe contained 25,365 rows, with four columns containing the User Handle, the text of each Tweet, and the subject gender (*Table 2*). The dataframe was saved as a .csv file for later use. In [10]: # Remove handles tweets_df['Text'] = tweets_df['Text'].apply(lambda x: re.sub('@[\w]+','', str(x))) #Remove punctuation # remove 's, '11, 'd tweets_df['Text'] = tweets_df['Text'].str.replace("'s","") tweets_df['Text'] = tweets_df['Text'].str.replace("'ll","") tweets_df['Text'] = tweets_df['Text'].str.replace("'d","") tweets_df['Text'] = tweets_df['Text'].str.replace("Replying to ","") tweets_df['Text'] = tweets_df['Text'].str.replace("\n","") def remove_punctuation(text): no_punct=[words for words in text if words not in string.punctuation] words_wo_punct=''.join(no_punct) return words_wo_punct tweets_df['Text']=tweets_df['Text'].apply(lambda x: remove_punctuation(x)) # Assign a gender to each tweet based on pronouns present tweets_df.loc[tweets_df['Text'].str.contains(' he '), 'Gender'] = 'M' tweets_df.loc[tweets_df['Text'].str.contains('He '), 'Gender'] = 'M' tweets_df.loc[tweets_df['Text'].str.contains(' him '), 'Gender'] = 'M' tweets_df.loc[tweets_df['Text'].str.contains(' she '), 'Gender'] = 'F' tweets_df.loc[tweets_df['Text'].str.contains('She'), 'Gender'] = 'F' tweets_df.loc[tweets_df['Text'].str.contains(' her '), 'Gender'] = 'F' # Assign a topic to each tweet based on topic in text tweets_df.loc[tweets_df['Text'].str.contains('science'), 'Topic'] = 'science' tweets_df.loc[tweets_df['Text'].str.contains('scientist'), 'Topic'] = 'science' tweets_df.loc[tweets_df['Text'].str.contains('technology '), 'Topic'] = 'technology' tweets_df.loc[tweets_df['Text'].str.contains('technogist'), 'Topic'] = 'technology' tweets_df.loc[tweets_df['Text'].str.contains('engineering'), 'Topic'] = 'engineering' tweets_df.loc[tweets_df['Text'].str.contains('engineer'), 'Topic'] = 'engineering' tweets_df.loc[tweets_df['Text'].str.contains('math'), 'Topic'] = 'mathematics' tweets_df.loc[tweets_df['Text'].str.contains('mathematics'), 'Topic'] = 'mathematics' tweets_df.loc[tweets_df['Text'].str.contains('mathematician'), 'Topic'] = 'mathematics' tweets_df = tweets_df[["Handle", "Text", "Likes", "Gender", "Topic"]] # Drop any rows containing blanks tweets_df = tweets_df.replace(r'^\s*\$', np.nan, regex=True) tweets_df = tweets_df.dropna() # Shuffle data tweets_df = tweets_df.sample(frac=1).reset_index(drop=True) tweets_df.head(10) Out[10]: Text Likes Gender Handle Topic 0 @KellyLovesBirds he loves science so much why did he kill his ... science @Annie447 18 year old daughter got her vaccine today w... science @LXV and Musk isnt an engineer He a marketer Maa... engineering 3 31 @wvu999 core math Oh wait she too old for common core F mathematics @Pathfinder4545 and Yeah dont ask a physicist to explain th... science @empathspeaks If the existence of a painting is enough proof... science 2nd engineer visit due today only to be told ... @scribblerdc engineering @IAmLordKPS Why was so confused seeing Bunnies He knows i... mathematics @NDSUSMINOR Reagan Mulqueen 20 partnered with Frank Fransi... science 9 @CatholicUalumni "I saw for myself how applications of science ... science In [73]: # Save to csv tweets_df.to_csv ('tweets_df.csv', index = None, header=True) **Exploratory Data Analysis** Preliminary analysis of the scraped data show, as expected, a strong bias in gender. In general, there are twice as many Tweets about men in relation to STEM as there are about women (66.5% and 33.5%, respectively) (Figure 3). In terms of gender, the dataset is unbalanced, which has implications for NLP tasks like classification. This trend is reflected within each of the STEM topics individually (*Figure 4*). The greatest discrepancy in men and women is shown in engineering, where female pronouns accompany less than 30% of all engineering Tweets. The smallest discrepancy is in mathematics, with female pronouns appearing in 39% of all Tweets about mathematics. Figure 5 shows that the vast majority of Tweets are on the topic of science in general, comprising almost 9,000 of the total 17,000 Tweets. These preliminary results reflect the broader issue of female representation in STEM-related fields.

Document 2. Twitter Scraper using Selenium

Aim

Name: Gender, dtype: int64 Total Tweets: 17313 Percentage of STEM Tweets by gender M 0.665049 F 0.334951 Name: Gender, dtype: float64

Count of STEM Tweets by gender

In [4]: # Import data

print(counts)

print(prop)
print()

plt.subplot(1, 2, 1)

plt.xlabel('Gender')
plt.ylabel('Count')
xticks(rotation=0)

plt.subplot(1, 2, 2)
prop.plot(kind = 'bar')

plt.xlabel('Gender')

xticks(rotation=0)

11514 5799

12000

10000

8000

6000

0.6

0.5

Proportion 0.4

0.2

0.1

engineering

mathematics

Topic

Figure 5: Tweet counts by gender across fields.

Website/data copyright considerations

plt.show()

counts.plot(kind = 'bar')

print()

tweets_df = pd.read_csv("tweets_df.csv", dtype=object)

prop = tweets_df['Gender'].value_counts(normalize=True)

Count tweets about women vs men

figure(figsize=(10, 5), dpi=100)

print("Count of STEM Tweets by gender")
counts = tweets_df['Gender'].value_counts()

print("Total Tweets:", len(tweets_df))

plt.title('STEM Tweet Counts by Gender')

plt.title('Percentage of STEM Tweets by Gender')

plt.gca().yaxis.set_major_formatter(mtick.PercentFormatter(1.0))

print("Figure 3: Count and percentage of Tweets by gender")

STEM Tweet Counts by Gender

print("Percentage of STEM Tweets by gender")

In [5]: print("Count of STEM Tweets by gender")
#counts = tweets_df['Gender'].
print(counts)
print()

gender_counts = tweets_df.groupby(['Topic', 'Gender']).Gender.count().unstack()
gender_prop = tweets_df.groupby(['Topic', 'Gender']).size().groupby('Topic').transform(lambd a x: x/x.sum()).unstack()

60%

50%

40%

30%

20%

Percentage of STEM Tweets by Gender

gender_counts.plot(kind='bar') plt.title('Tweet counts by Topic and Gender') plt.xlabel('Topic') plt.ylabel('Tweet count') plt.xticks(rotation = 0) print("Figure 4: Gender proportions by topic, showing gender imbalance across fields.") gender_prop.plot(kind = 'bar') plt.title('Gender proportions by Topic') plt.ylabel('Proportion %') xticks(rotation=0) plt.rcParams["figure.figsize"] = (15,5) plt.show() print("Figure 5: Tweet counts by gender across fields.") Count of STEM Tweets by gender 11514 5799 Name: Gender, dtype: int64 Figure 4: Gender proportions by topic, showing gender imbalance across fields. Tweet counts by Topic and Gender 6000 Gender F 5000 M 4000 3000 2000 1000 engineering mathematics science technology Topic Gender proportions by Topic 0.7 Gender

such, scraping Tweet data from Twitter does not violate copyright laws.

Even though Tweets are protected under copyright, this is an academic exercise and, as