Using Language Models to Quantify Gender Bias in Recent American Election Journalism

Shubha Swamy

Department of Computer Science

University of Colorado Boulder

Advisor

Dr. James H. Martin

In partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science

May, 2020

Abstract

Gender bias is an increasingly prominent topic when it comes to American elections. In this thesis, I produce quantifiable metrics using Natural Language Processing techniques to explore the variation in questions posed by journalists to female and male candidates in recent American elections. The analysis uses a bigram language model and perplexities to determine if there is statistical significance in proving that questions asked by reporters to political candidates vary by gender. The dataset includes a gender-balanced set of in media interviews for congressional, senate, and presidential candidates from 2014-2020. The results suggest that questions asked to male candidates are more elections and policy-related than those asked to female candidates.

Contents

Chapter 1

Introduction

Significant research shows that the media plays a prominent role in shaping citizen's views about political candidates [Graber, 1993]. Given the importance of the media's portrayal of political candidates, I use a language-model-based approach to analyze if the questions posed to female candidates vary significantly from those asked to male candidates.

The data set used for analysis makes use of transcripts collected from various news interviews of political candidates from YouTube. I extracted questions asked by interviewers from these transcripts, which allowed for quantitative analysis of the variance in questions asked to female and male political candidates.

This thesis follows the framework based on "Tie-breaker: Using language models to quantify gender bias in sports journalism" [Fu et al., 2016]. The paper uses a language-model based method to analyze gender bias in tennis interviews. I apply the techniques outlined in the paper to extract gender bias in political candidate interviews. This method involves using a language model to evaluate perplexities for questions asked and used statistics to determine the statistical significance of the difference.

This approach yields quantifiable results to make conclusions to the initial question proposed.

1.1 Summary

This project aims to quantify the gender bias of political candidates as portrayed by the media language. Examining the role of language bias regarding election candidates is important because the media plays a significant role in shaping people's perspectives about candidates. Using language models to perform this analysis mainly provides an insightful measure of how language plays a role explicitly. Additionally, there has also been relevant work done in the space showing gender bias when the media interacts with political members. However, their studies are often from a policy and political perspective and use regression models to conclude bias. There has been some work done in the Natural Language Processing space analyzing gender bias and language. However, the application mostly pertains to areas of sports, jobs, Tweets, culture, but not elections. Because of limited existing research in applying language models to analyze how political candidates are treated, this is an opportunity to investigate gender bias in elections using Natural Language Processing techniques.

Chapter 2

Related and Previous Work

There are different methods of detecting bias in language. In NLP, grammar and words are represented through models. One way of expressing words in NLP is by using one-hot encoding. One-hot represents a word in terms of a binary in a 1xN vector. One-hot vectors make it hard to compare words relative to each other because the context of the phrase is lost in representation. However, one-hot vectors are not helpful for this context because they do not allow us to extract bias concerning other words.

Another method of representing words is through word embeddings. In word embeddings, each word is represented by a vector, and each value in the vector is a score associated with a feature. The vector representation of word embeddings allows us to find dot products between both vectors and figure out relationships between words. Using a method like word embeddings is beneficial in extracting bias. Recent publications looked into analyzing word embeddings overtime for 100 years that gave insight into gender and ethnic stereotypes [Garg et al., 2018]. The use of word embeddings works excellent in this scenario because the research examines the trends in gender bias over time. Representing the words as vectors allows one to explore relationships concerning each other quickly. In another publication, word embeddings were created by Word2Vec used along with a K-means clustering algorithm to extract subtle gender and everyday language [McKeown and Chang, 2019]. Word embeddings seem to work great for scenarios that are being compared to each other with some relation. In order to compare bias in a binary category, it might be better to look into other more effective methods.

One recent study used language models to quantify gender bias in tennis journalism coverage. The goal of the paper was to use a language-model-based approach to measure the difference in female versus male athlete reporting [Fu et al., 2016]. To do this, the researchers used a dataset of questions asked by journalists to male and female tennis athletes. The words in the questions were processed into bigrams using the KenLM library. The probabilities were then applied to measuring the perplexity of the questions. The perplexity refers to how well a probability predicts a sample. The higher the perplexity, the less the question relates to the topic we are concerned with. Using this approach seems promising to measure and compare the binary probabilities of female versus male bias. In the study, perplexity was measured concerning how game-related the question was.

Chapter 3

Data

The data used in this thesis is extracted from YouTube. YouTube is a platform that contains videos uploaded by users, including various media platforms that conduct news interviews. The access to a collection of various interviews makes it possible to easily extract transcripts from the interviews needed to run this analysis. YouTube provides for an outlet to collect news interviews across various media outlets since various content providers upload their videos onto YouTube.

3.1 Data Processing

The initial step included cleaning up the transcripts. Cleaning the data is necessary to reduce noise that the text may introduce when building our model. The process involved converting all strings to lowercase and removing unnecessary punctuation. Using libraries such as *nltk* and *string*, I was able to tokenize words, remove punctuation, and remove stop words. A total of 1437 transcripts across the two genders represent politics and election specific language.

3.2 Interview Transcripts

I collected news and TV-show interviews from YouTube through their API [Appendix 7.3]. Video IDs for interviews for a gender-balanced set of election candidates are collected through the YouTube API. I then use another API to get transcripts for those videos. The videos are news interviews for various candidates extracted by searching the query in the form of the candidate's name and the text: *election candidate interview*. The candidates included vary from the senate, congressional, and presidential candidates from the 2014-2020 elections. I compiled a list of candidates, and then extracted interview transcripts from YouTube for those candidates. A total of 718 interview transcripts for female candidates and 719 transcripts for male candidates were extracted.

3.2.1 Extracting News Interview Questions

A gender-balanced list of random political candidates running for presidential, senate, and house from 2014-2020 was curated. The list of candidates was fed into a Python program to get video IDs for each candidate's news interviews. The initial query was made to the YouTube API to get a list of video IDs relevant to the topic. These corresponding video IDs were then used to make another query to YouTube to get a transcript for each of those videos. To extract questions asked by interviewers, I used regex to extract phrases that followed a question's format [Appendix 7.4]. This resulted in two files compiled with all the extracted questions for females and males, respectively. This dataset consists of 4026 questions for females and 5647 questions for males. The average length of female interview transcripts is 1667.64 words, and the average length of interview transcripts for males is 1923.88 words. The discrepancy in the counts for the total number of questions extracted for each gender can be partially attributed to the average length of transcripts. The example below is an excerpt from a transcript, and the question extracted with regex.

Excerpt from a transcript:

"all right, mr. yang, let's start big.what's your view of the role government should play in our lives besides giving everyone over the age of 18 $1,000 a month?[yang chuckles]i love this question.to me, the government's responsibility is to solve the biggest problems and address the biggest needs that don't have any market incentive attached to them."

Sample question extracted:

"what's your view of the role government should play in our lives besides giving everyone over the age of 18 $1,000 a month?"

3.3 Language Model Data

The language model used in this project to model political candidate interview specific language uses the entire transcripts extracted from all the YouTube videos compiled. For the analysis, a gender-balanced set of transcripts consisting of interviews for the 2014-2020 senate, congress, and presidential elections were considered. These transcripts were cleaned up and then used to model a bigram language model using KenLM. This model evaluates how well the questions asked by reporters in our interviewers relate to elections or politics related language.

Chapter 4

Approach

I started with a preliminary analysis to determine if there was a difference in the questions reporters asked female and male candidates. This was a word-level analysis to determine the top ten distinguishing words asked in interview questions to each gender. Using the accumulated text data in transcripts, I modeled the text data as a bigram language model to represent election and politics related text. The next step was using the bigram language model to evaluate perplexities for the questions asked by reporters in interviews. The perplexities quantify the degree to which a question asked by the media is election or politics related. The last step was evaluating the statistical significance of whether there was a difference in the questions posed to female and male political candidates.

4.1 Preliminary Analysis

For the initial analysis, I extract the top words with the greatest difference in frequency for male and female candidates. To do this, I initially process the two different files which contain questions asked to female candidates and male candidates in interviews. This process includes using the *nltk* library in Python to tokenize all the words and removing stop words. The removal of stop words such as "the", "an", "a", etc. allows the analysis to extract words that are important to the context of the corpus.

For each unique word, w, in all the questions, I calculate the percentage of male and female candidates who have been asked a question containing that specific word, w. To analyze if words mentioned in questions to candidates vary by gender, I calculate the difference in frequency for each word between males and females. The words with the greatest difference in percentage represent words that are more commonly asked for one specific gender over the other [Appendix 7.1]. The top ten words that are most common are listed below by gender.

Male candidates:

president, right, democrats, house, know, congress, win, young, care, republicans

Female candidates:

question, need, believe, want, one, think, new, dont, make, going

The initial analysis suggests that the words used in questions asked to male political candidates are more election and politics related than those posed to females. Eight of the ten words for male candidates can be considered terms related to politics and election-related: president, right, democrat, house, congress, win, care, republicans. None of the top ten distinguishing words for female candidates are considered elections or politics related.

4.2 Bigram Language Model

The main goal of using a language model in this project is to quantify metrics for the verbal language used by journalists. Evaluating the perplexity for the questions asked by the journalists using bigram probabilities allows for the measurement of how election or politics related the questions asked to the candidates are. The bigram language model is trained on a gender-balanced set of interview transcripts. I train a bigram model using KenLM [Heafield et al., 2013]. The goal of creating the bigram language model is to train a system with words that are considered election or politics related to the scope of the text used. KenLM provides for a fast and scalable way to create a language model [Appendix 7.2]. KenLM uses Kneser-Ney Smoothing, which adjusts the probability distribution to make better estimates for sentence probabilities in the language model. Sample bigram probabilities produced by KenLM are displayed in the table below. Since KenLM produced log probabilities for the language model, a higher value indicates a higher probability. From the table below, *passing legislation* is more likely to occur in our corpus compared to *safety legislation*.

Bigram Log Probability	**Bigram Text**
-2.6220992	pushing legislation
-2.4817846	safety legislation
-2.4600062	writing legislation
-2.3061495	proposing legislation
-1.7251786	passing legislation
-1.7761264	blocking legislation
-2.2183554	flag legislation

Table 4.1: Bigram Text and Log Probabilities

4.3 Perplexity

I measure the individual perplexity of each question with respect to the probabilities from the bigram language model. The perplexity is a value that measures how well the test data fits the language model [Jelinek et al.., 1977]. In this analysis, the perplexity represents how closely related to politics or election the question is. A higher perplexity value indicates that the question is less politics or elections related. A question q can be represented with its perplexity as $PP(q)$ and the probability of the bigram language model as $P_{interview}$. The word sequence which makes the questions can be represented by a sequence of words as: $w_1, w_2...w_n$. The perplexity of a question, $PP(q)$, with a sequence of words $w_1, w_2...w_n$ is represented as $PP(w_1, w_2...w_n)$. And the combined bigram probability of those words can be represented as: $P_interview(w_1, w_2...w_n)$. To measure the total perplexity of the question, we define the equation as the following:

$$PP(w_1, w_2...w_n) = \sqrt[n]{\frac{1}{P(w_1, w_2...w_n)}} \tag{4.1}$$

A question's perplexity is the inverse to the language model, and therefore a lower perplexity means the better the question fits the model we trained. To analyze results, I look at the mean perplexity for each gender and compare the two means. The lower average perplexity represents questions that are more election and politics related to the specified gender.

Below are two examples of questions that represent different perplexity values, low and high. These questions are excerpted from the list of questions evaluated for their perplexities.

Perplexity	**Sample Question**	**Perplexity Value**
Low	*are we going to be trying to send troops up to the border?*	127.448
High	*how can someone be unhappy?*	1279.928

Table 4.2: Low and High Perplexity Questions

Chapter 5

Results and Discussion

5.1 Preliminary Analysis

The initial analysis of the top ten most differentiating words shows that words used in questions asked to male political candidates are more election and politics related than those posed to females.
Female:

Word	**Frequency Difference**
question	0.0016747866122532257
need	0.0016691500856798722
believe	0.0012771813379008421
want	0.0011885884281271353
one	0.0010816768069900297
think	0.0010123622945371249
new	0.000987896816328903
dont	0.0009555475831176572
make	0.0009405573753124051
going	0.0009232591444919488

Table 5.1: Top ten distinguishing words for female candidates

Male:

Word	Frequency Difference
president	0.002677132998823141
right	0.0022284103342101787
democrats	0.001994500995121601
house	0.0017741423349021688
know	0.0017605004641547198
congress	0.0016470838223651605
win	0.0015003278641141829
young	0.0014487610281676711
care	0.0012410541554455153
republicans	0.001224174973018223

Table 5.2: Top ten distinguishing words for male candidates

5.2 Graphs

5.2.1 Mean Perplexity Scores

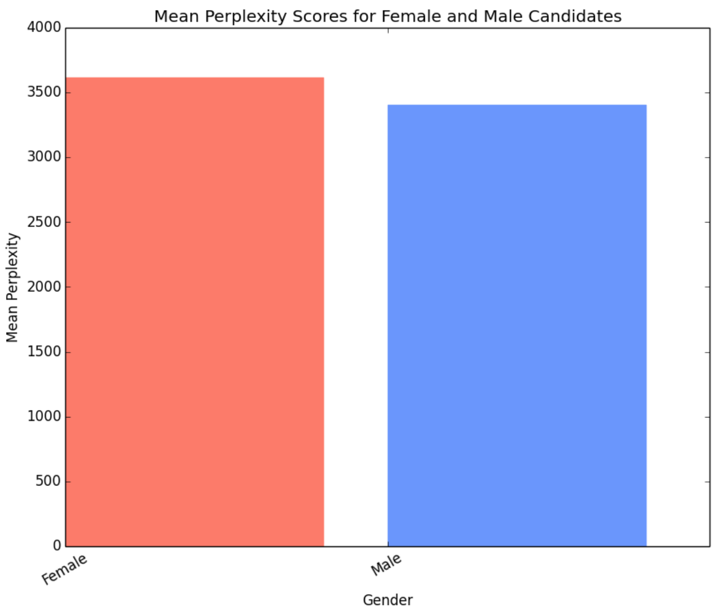

Figure 5.1 Mean perplexity values for male and female candidates

The mean perplexity score for female candidates is 3614.7074608 and the mean perplexity score for male candidates is 3402.42068263.

5.2.2 Distribution of Perplexity Scores

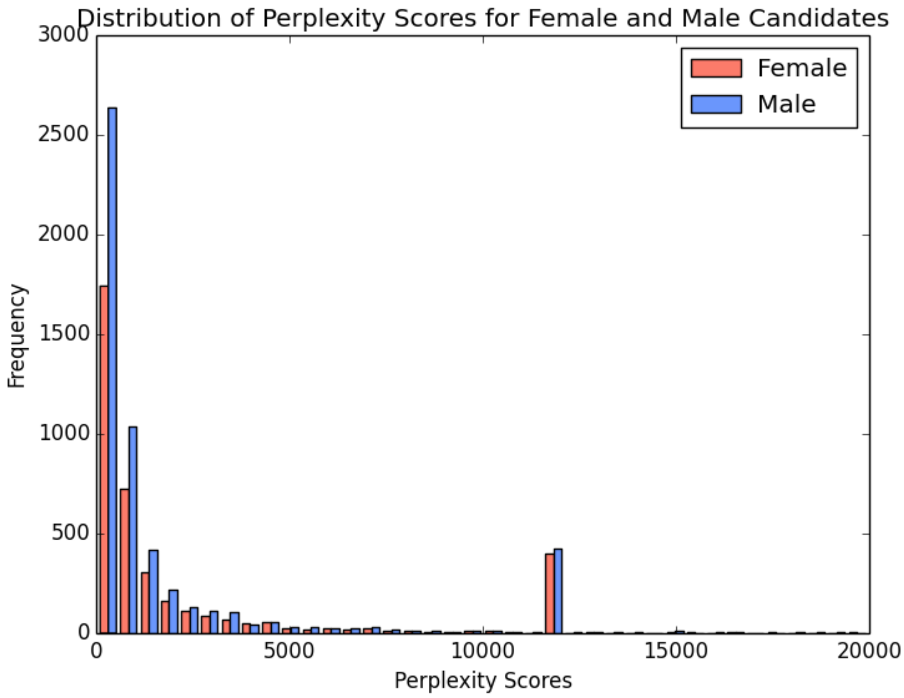

Figure 5.2 Perplexity score distribution for female and male candidates

The distribution of perplexity scores show both genders have a similar shape. There is a higher frequency of low perplexity scores for male candidates.

5.3 Statistical Significance

I use the Mann-Whitney U statistical significance test to determine if the mean difference in perplexities between the two genders is statistically significant. One benefit of using the Mann-Whitney U test over other statistical tests is that it does not assume that the two samples to be normally distributed.

The fulfillment of the required assumptions also makes it possible to conclude results with this test. The assumptions are as follows: 1) The dependent variable, the perplexity scores, is measured as a continuous variable. 2) The independent variable, gender, consists of two categorical, independent groups– female and male candidates. 3) Independence of observation between female and male candidates.

4) The distribution of perplexity scores for female and male candidates have a similar shape.
The Mann-Whitney U test results indicate that the mean perplexity of questions posed to male candidates is significantly smaller ($p\text{-value} < 0.000$) than that of questions posed to female candidates. This suggests that the questions male candidates receive are more politics and elections-related than questions posed to their female counterparts.

5.4 Perplexity Values

The mean perplexity score for female candidates is 3614.707, and the mean perplexity score for male candidates is 3402.421. A lower perplexity score for a string of words indicates that the text was a better fit for the model. The corpus used for the language model is a collection of all the interview transcripts. The entirety of the corpus represents election and political language, which is represented in the language model. A lower mean score for male candidates indicates that reporters' questions to male candidates fit the language model better. Since there is statistical significance, questions posed to male candidates better fit the language model than the questions posed to their female counterparts.

Chapter 6

Conclusions

6.1 Summary

This thesis further extends the work of "Tie-breaker: Using language models to quantify gender bias in sports journalism" [Fu et al., 2016] to analyze gender bias in the coverage of political candidates running for national office in recent American elections. The language-model based approach makes it easy to quantify the bias that the media pose to female vs. male candidates running for office. There is statistical significance in showing that questions asked to male candidates are more election and politics related than those asked to their female counterparts.

This thesis provides evidence in proving gender bias in questions asked by the media to election candidates exists. Questions posed to male candidates are more politics and elections-related.

6.2 Future Work

This experiment has some limitations, and has further scope for improvement. There is some noise in the data collected, despite controlling for some of it. The process for collecting the transcripts of news interviews was automated via the YouTube API. I sampled random videos to control for the quality of the news interviews. However, this standard might not hold for all the video transcripts collected. Using regex to extract questions from transcripts provided a guarantee that interviews involved reporters asking questions to candidates; however, the quality was not controlled.

In this work, the scope of analyzing bias is limited to the questions reporters ask election candidates. For future work, the scope can be extended to account for the differences in various reporters and control for the media outlet conducting the interview. Additionally, analyzing the candidates' responses to the questions could also help provide meaningful insight. Looking at whether the same trends exist by breaking down the questions into categories, or analyzing the bias by office position can also help provide helpful insight.

6.3 Acknowledgments

I thank Dr. James H martin for being my advisor and guiding me on the right path for this project. And many thanks to the committee members, Chenhao Tan and Alan Paradise, for reviewing and evaluating my work.

Chapter 7

Appendix

7.1 Extracting top 10 distinguishing words for each gender and frequency differences

```
import string
import heapq
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize


def remove_stop_words(cleaned_str):
    # clean stop words from corpus
    word_tokens = word_tokenize(cleaned_str)
    stop_words = set(stopwords.words('english'))

    filtered_words = []
    for w in word_tokens:
        if w not in stop_words:
            filtered_words.append(w)

    return filtered_words

def unique_file(input_filename):
    #get all unique words in the corpus
    input_file = open(input_filename, 'r')
    file_contents = input_file.read()
    input_file.close()

    translator = str.maketrans('', '', string.punctuation)
    cleaned_string = file_contents.translate(translator) #remove
                                punctuation
    filtered_words = remove_stop_words(cleaned_string)

    #hold unique words only
    unique_words = [] #array to hold all unique words
    for word in filtered_words:
        if word not in unique_words:
            unique_words.append(word)
    return unique_words

def get_all_words(input_filename):
    #get all the words
```

```
    input_file = open(input_filename, 'r')
    file_contents = input_file.read()
    input_file.close()

    all_words = []

    translator = str.maketrans('', '', string.punctuation)
    cleaned_string = file_contents.translate(translator) #remove
                                 punctuation

    all_words = remove_stop_words(cleaned_string)

    return all_words

def check_files():
    #main: get top ten distinguishing words and frequency
                                 difference
    file_in = "female_presidential_questions.txt"
    unique_female_words = unique_file(file_in)
    all_female_words = get_all_words(file_in)

    file_in2 = "male_presidential_questions.txt"
    unique_male_words = unique_file(file_in2)
    all_male_words = get_all_words(file_in2)


    percent_male = {}
    percent_female = {}

    total_female_words = len(all_female_words)
    total_male_words = len(all_male_words)

    for word in unique_female_words:
        occurence = all_female_words.count(word)
        percent_female[word] = occurence/total_female_words

    for word in unique_male_words:
        occurence = all_male_words.count(word)
        percent_male[word] = occurence/total_male_words

    diff_percents_female ={}

    for k, v in percent_female.items():
        if k in percent_male.keys():
            diff_percents_female[k] = v-(percent_male[k])

    diff_percents_male ={}

    for k, v in percent_male.items():
        if k in percent_female.keys():
            diff_percents_male[k] = v-(percent_female[k])

    top_n_diffs_female_desc = sorted(diff_percents_female, key=
                                 diff_percents_female.get,
                                 reverse=True)[:10]
    top_n_diffs_male_desc = sorted(diff_percents_male, key=
                                 diff_percents_male.get,
                                 reverse=True)[:10]
```

7.2 Producing the Bigram language model with KenLM

```
import kenlm


def ken_lm_model():
    model = kenlm.Model("political_bigram.arpa")


    female_question_perplexity = []
    male_question_perplexity = []


    filename = 'female_presidential_questions.txt'
    with open(filename, 'r') as f:
      for line in f:
        perplexity = model.perplexity(line)
        female_question_perplexity.append(perplexity)

    filename = 'male_presidential_questions.txt'
    with open(filename, 'r') as f:
      for line in f:
        perplexity = model.perplexity(line)
        print("S: " + line + " P: " + str(perplexity))
        male_question_perplexity.append(perplexity)


    female_avg = sum(female_question_perplexity) / len(
                                        female_question_perplexity)
    male_avg = sum(male_question_perplexity) / len(
                                        male_question_perplexity)


    print('FEMALE perplexity mean')
    print(female_avg)

    print('MALE perplexity mean')
    print(male_avg)

    return (female_question_perplexity, male_question_perplexity)

ken_lm_model()
```

7.3 YouTube API Query

```
def get_api_request():
    # list of videos matching search query
    video_dict_list = {}
    video_id_list = []
    query_list = search_query()

    for query in query_list:
        QUERY = query
```

```
        #change max results or edit query
        curl_url = 'https://www.googleapis.com/youtube/v3/search?
                                                part=snippet&maxResults=7
                                                &type=video&videoCaption=
                                                closedCaption&q='+ QUERY
                                                +'&key=[INSERT_KEY]'
        api_request = requests.get(curl_url)
        api_output = api_request.json()

        for video in api_output['items']:
            video_id = video['id']['videoId']
            title = video['snippet']['title']
            published = video['snippet']['publishedAt']

            video_id_list.append(video_id)
            video_dict_list[video_id] = [title, published]

    filename = "all_videos_read.txt"
    f = open(filename, "a")
    for key, value in video_dict_list.items():
        write_str = key + "     " + value[0] + "     " + value[1] +
                                                "\n"
        f.write(write_str)
    f.close()
    return video_id_list
```

7.4 Regex to extract questions from text

```
import re
import os


all_files = os.listdir("Folder/")

for filename in all_files:
    textfile = open('textfile/'+filename, 'r')
    filetext = textfile.read()
    textfile.close()

    # (regex to read the question
    regex = r'(?<=[.?!])s*[A-Za-z,;\'"\s]+\?'
    matches = re.findall(regex, filetext)


    filename2 = 'textfile_questions.txt'

    f = open(filename2, "a")
    for question in matches:
        f.write(question)
        f.write('\n')
    f.close()
```

Bibliography

[Bender and Friedman, 2018] Bender, E. M., Friedman, B. (2018). Data Statements for Natural Language Processing: Toward ... Retrieved from https://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00041

[Cameron, 1996] Cameron, D. (1996, April 1). Style policy and style politics: a neglected aspect of the language of the news - Deborah Cameron, 1996. Retrieved from https://journals.sagepub.com/doi/10.1177/016344396018002008

[Denton and Kuypers, 1996] Denton, R. E., Kuypers, J. A. (2008). Politics and communication in America: campaigns, media, and governing in the 21st century. Long Grove, IL: Waveland Press.

[Druckman and Parkin, 2005] Druckman, J. N., Parkin, M. (2005, January 5). The Impact of Media Bias: How Editorial Slant Affects Voters. Retrieved from https://www.journals.uchicago.edu/doi/abs/10.1111/j.1468-2508.2005.00349.x

[Entman, 2007] Entman, R. M. (2007, February 28). Framing Bias: Media in the Distribution of Power. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1460-2466.2006.00336.x

[Fu et al., 2016] Fu, L., Danescu-Niculescu-Mizil , C., Lee, L. (2016, July 13). Tie-breaker: Using language models to quantify gender bias

... Retrieved from http://www.cs.cornell.edu/~cristian/papers/gender-tennis.pdf

[Garg et al., 2017] Garg, N., Schiebinger, L., Jurafsky, D., Zou, J. (2017, November 22). Word Embeddings Quantify 100 Years of Gender and Ethnic ... Retrieved from https://arxiv.org/pdf/1711.08412.pdf

[Heafield et al., 2013] Heafield, K., Pouzyrevsky, I., Clark, J. H., Koehn, P. (2013, August). Scalable Modified Kneser-Ney Language Model Estimation. Retrieved from https://kheafield.com/papers/edinburgh/estimate$_$paper.pdf

[Hooghe et al., 2015] Hooghe, M., Jacobs, L., Claes, E. (2015, July 23). Enduring Gender Bias in Reporting on Political Elite ... Retrieved from https://www.researchgate.net/publication/280922359_Enduring_Gender_Bias_in_Reporting_on_Political_Elite_Positions_Media_Coverage_of_Female_MPs_in_Belgian_News_Broadcasts_2003-2011

[Kinnick, 2010] Kinnick, K. N. (2010, November 11). Gender Bias in Newspaper Profiles of 1996 Olympic Athletes ... Retrieved from https://www.researchgate.net/publication/271994353_Gender_Bias_in_Newspaper_Profiles_of_1996_Olympic_Athletes_A_Content_Analysis_of_Five_Major_Dailies

[Kulshreshtha et al., 2017] Kulshrestha , J., Eslami, M. E. profilM., Messias, J., Zafar, M. B., Ghosh, S., Gummadi, K., Karahalios, K. (2017, April 5). Quantifying Search Bias: Investigating Sources of Bias for ... Retrieved from https://arxiv.org/pdf/1704.01347.pdf

[Kuypers, 2002] Kuypers, J. A. (2002). Press bias and politics: how the media frame controversial issues. Westport, CT: Praeger.

[Lawless, 2009] Lawless, J. L. (2009, March 9). Sexism and Gender Bias in Election 2008: A More Complex ... Retrieved from https://www.cambridge.org/core/journals/politics-and-gender/article/sexism-and-gender-bias-in-election-2008-a-more-complex-path-for-women-in-pol 6CB30AAFFE94AA55118E1D369775084D

[McKeown and Chang, 2019] McKeown, K., Chang, S. (2019, August 30). Automatically Inferring Gender Associations from Language. Retrieved from https://www.aclweb.org/anthology/D19-1579.pdf

[Milyo and Schosberg, 1998] Milyo, J., Schosberg, S. (1998, January 1). Gender Bias and Selection Bias in House Elections, by Jeffrey Milyo; Samanth Schosberg. Retrieved from https://ideas.repec.org/p/tuf/tuftec/9809.html

[Schmidt, 2017] Schmidt, H. C. (2017, December 8). Forgotten Athletes and Token Reporters: Analyzing the Gender Bias in Sports Journalism. Retrieved from https://www.tandfonline.com/doi/abs/10.1080/15456870.2018.1398014

[Gaut et al., 2019] Tony, Gaut, Andrew, Tang, Huang, Yuxin, ... Diba. (2019, June 21). Mitigating Gender Bias in Natural Language Processing: Literature Review. Retrieved from https://arxiv.org/abs/1906.08976

[Voigt et al., 2017] Voigt, R., Camp, N. P., Prabhakaran, V., Hamilton, W. L., Hetey, R. C., Griffiths, C. M., ... Eberhardt, J. L.

(2017, June 20). Language from police body camera footage shows racial disparities in officer respect. Retrieved from https://www.pnas.org/content/114/25/6521