University of Nevada, Reno

Understanding Bias Using Machine Learning

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering

by

Sara R. Davis

Dr. Emily M. Hand - Thesis Advisor

May 2020

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

Sara R. Davis

entitled

Understanding Bias Using Machine Learning

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Emily M. Hand, Ph.D.
Advisor

David Feil-Seifer, Ph.D.
Committee Member

Adam Kirn, Ph.D
Graduate School Representative

David W. Zeh, Ph.D., Dean
Graduate School

May, 2020

Abstract

Socioeconomic and gender bias are issues that have lasting social implications. We focus on mitigating socioeconomic bias by introducing an intelligent mathematical tutoring system, HWHelper. HWHelper is designed to be accessible by students, so that students who may not otherwise have help at home have access to guided instruction. To mitigate gender bias, we introduce a framework for the identification and evaluation of gender bias in political news articles. We find that HWHelper performs well on the limited dataset We find that our political gender bias detection system finds clear differences in the language used to describe male versus female political candidates.

Dedication

This thesis is dedicated to my family, for imparting in me their unique perspective, love of learning, and drive to excel and work hard;

Acknowledgments

I would like to acknowledge the following people:

- My advisor, Dr. Emily Hand, for all of her guidance and support, in the process of writing this thesis, and in all of my other endeavours;
- Dr. Feil-Seifer and Dr. Kirn for taking time to review this thesis.
- Karl Mullner for the use of his perspective and all of his encouragement;
- Cody Worsnop, Bryson Lingenfelter, Dustin Barnes, Nathan Thom, and the rest of the MPL/CVL for all of their useful advice, humor, and allowing me the use of their equipment as necessary;
- Jillian Szewczak for her superior editing skills and writing suggestions;
- My aunt, Kristi Jo, for pushing me forward when I was ready to quit;
- My grandmother, Joann, for constantly telling me to get my Ph.D.;

Without their help, this work would not exist.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Bias exists in many forms, from socioeconomic status to race, and can impact the opportunities available to individuals [1–5]. This is largely due to systemic bias. Systemic bias is the bias intrinsic to society that supports particular outcomes that favor the societal majority [6]. It is important to note that systemic bias is not always overt- it is built into codified laws that many people do not second guess, and this can lead to oppression of minority groups [7]. These two forms of systemic bias are particularly harmful because they are everywhere and difficult to detect [6]. Given the impact of socioeconomic bias and gender bias, it is particularly important to address these forms of systemic bias.

While education is often seen as a means of escaping poverty, family structure and systemic bias favoring higher socioeconomic families makes academic success difficult due to lower parental involvement [8]. Parents from lower income households are less able to be involved in their child's education [9, 10] and unable to invest in early enrichment opportunities [11]. Therefore, students born into lower socioeconomic status homes tend to under-perform academically all the way into middle and high school [12], particularly in the areas of math and reading [13], contributing to lower graduation rates [2]. Lack of a high school diploma can lead to lower employment rates, lower income earning potential, and higher rates of teen pregnancy and incarceration [2]. Students who drop out are typically of lower socioeconomic status than students who do not, and they often face more stigma than their peers when searching for jobs due to their lack of diploma [2]. This ultimately creates a cyclically biased system in which a child is born to a family with modest means, their family cannot afford enrichment or to invest extra time in their educational development, the child drops out of high school and this impacts their earning ability, and when the child has children of their own, their child is born into a family with similar modest means. This perpetuates the cycle of poverty, and embodies systemic bias [14].

Gender bias has an impact on many aspects of daily life, from medical diagnosis [15] to representations in the media [16]. Like most forms of systemic bias, the results of gender bias are seemingly innocuous. For example, during the 1996 Olympics, many news outlets chose to focus on the looks, personal grooming habits, and personal lives of female Olympic athletes, while reporting on the male training and performance [16]. [16] argues that the focus on superficial qualities of the female athletes, rather than their achievements, serves to undermine their hard work. Controlling for the number of working hours, education, and experience, the ratio of female to male earnings is 0.90 in the United States [17]. Furthermore, studies conducted since the 90's have shown that it is harder for female employees to be promoted compared to their male counterparts [3–5]. Ultimately, all of these seemingly small choices have made their way into larger areas of life, resulting in the creation of systemic gender bias that directly affects day to day life for the average woman.

It is a commonly held belief that automation will help remove systemic bias by removing human judgement from the equation. However, it has been shown that if these automated systems are not built in a conscientious manner, they can end up further perpetuating bias. Automated methods are susceptible to many different types of bias, many of which are the result of the systemic biases present in society. State-of-the-art machine learning methods result in neural networks with millions of parameters [18–20]. Such large networks require computer hardware that is often expensive and not accessible to families with low incomes. In order to provide equitable access to children from these families, and ensure that the bias that exists in other spheres does not extend to educational platforms, it is necessary to consider the structure of the machine learning models employed when creating educational applications. Natural language processing techniques have been found to harbor gender bias [21] and racial bias [22]. These studies found that the word embeddings for certain genders were more closely related in word vector space to stereotypically gendered words [21, 22]. For example, the word woman is closer in word space to the word “homemaker" than it is to the word “computer" [21]. This indicates bias in the word space, though the authors explicitly state that it is impossible to determine if the bias is due to the way various words were paired in the texts used to create the embeddings, or if it is an issue with the embedding process [21]. This implies that work can be done to exploit the inherent bias engendered by word embeddings in order to detect gender bias in text.

This thesis aims to address systemic bias, specifically with respect to socioeconomic status and gender through the use of automated methods. Thus, this work presents two major contributions to the field of bias mitigation:

1. Homework Helper (HWHelper)- a novel homework assistant that is capable of determining mathematical errors in order to provide meaningful feedback to students without solutions. This allows students from a variety of socioeconomic and educational backgrounds to easily access and practice their mathematics skills.

2. A framework for the identification and evaluation of gender bias in political news articles. The work here serves to develop a method of detecting gender bias in political media using deep learning and natural language processing techniques.

Together, these contributions serve to improve our understanding of systemic bias in the areas of socioeconomic status & education, and gender.

Chapter 2

Background

This thesis builds on works in the areas of education, computer vision, intelligent tutoring systems, document embedding, and gender bias identification in natural language processing (NLP). The relevant works are reviewed in this chapter.

2.1 Student Academic Success in Mathematics

Developing curriculum that is challenging, but still within a student's skill level, is critical to student growth in all areas of education [23]. An integral aspect of developing curriculum at an appropriate level is providing students with timely feedback [23, 24]. Research suggests that if feedback is too fast and students are not allowed to struggle with a problem, then students will not acquire new knowledge and correct misconceptions [24, 25]. Vygotsky's zone of proximal development suggests that as long as the student is working with a concept that is challenging, but still solvable with their knowledge set, their knowledge base will grow- in fact, that struggle is critical to their growth [23]. It is also important that the feedback that is provided is understood by the student [25]. Math applications like PhotoMath [26] deprive students of the opportunity to struggle with a new concept by immediately providing them with the solution to the math problem that they are working on. This ultimately removes the challenge of solving the problem, which leads to stagnation in their math skills in that area. Furthermore, these applications often do not explain why solutions are incorrect, and if they do, they do so in a way that is not necessarily understandable to students in lower grades [27]. Studies published in mathematics education journals have looked at the mathematic applications that are available in appstores. Of the 4000 applications reviewed, only 34 were considered acceptable based on the criteria of being low cost. Further narrowing the list to only allow age appropriate applications and to remove games, and any application that is not free leaves only 3 widely available applications [28]. This means that the feedback that students receive from these easily accessible applications is not often useful to their academic success.

The alternative to using applications such as Photomath for homework help in mathematics is to ask a parent for help. Studies have shown that students who have parents that help them with homework are more academically successful than their peers who do not [29–31]. It is worth noting that while academic performance can be greatly impacted by parent understanding and parenting style, studies have shown that having a parent help poorly is better than not having a parent help at all with homework [30]. Simply put, a parent being involved in their child's education is a key indicator of their child's academic success [29–31]. [30] found that parents who had high school diplomas were statistically more likely to be involved in their child's education and homework than parents who did not have high school diplomas. Other studies have shown that income level has an impact on parent involvement in homework and education [8, 10], while some argue that this socioeconomic disparity is due to the amount of time that many of these parents spend working multiple jobs [29, 30]. Ultimately, the result is the same: students who live in low income households typically have a hard time gaining access to adults with the mathematical background necessary to help them succeed academically [8, 10, 29–31].

2.2 Object Detection and Recognition

The problem of detecting a math problem, and identifying a digit can be accomplished using a convolutional neural network (CNN). The use of CNNs for object detection in general is well studied [32]. TensorFlow is one of the most robust and well documented machine learning libraries for the purposes of object detection [33]. The TensorFlow publishers and outside researchers have generated a comprehensive set of pre-trained models for use on object detection [34]. The two most common model architectures are Faster Region-CNN (Faster R-CNN) [35] and Single Shot Multibox Detector (SSD) [36] due to their ability to quickly identify areas of interest rather than searching an entire image. The three most common feature extractors – that is, trained models – are ResNet [37], Inception [38], and MobileNet [39]. Given the unique nature of applying object detection to classroom settings, it is necessary to use a CNN architecture and feature extractor that are small enough to mount on a mobile platform quickly without sacrificing too much accuracy. Of all of the combinations of architecture and feature extractors, Faster R-CNN with a ResNet feature extractor typically has the best mAP performance, but performs slowly [40]. SSD with an Inception feature extractor does not perform as well, but is faster than the Faster R-CNN network architecture [40]. The fastest, but least accurate, combination is SSD architecture with a MobileNet feature extractor [40]. In terms of ease of integration into mobile applications, the best option for use in Homework Helper, is SSD architecture with a MobileNet feature extractor for two reasons. First, the

MobileNet is specifically designed to be small enough to run efficiently on mobile devices [39]. The use of a mobile devices is critical to making the application accessible to students whose only source of computational device is their, or their parent's, smartphone. This is particularly likely in low income households [41], which is the key demographic we wish to aid with our work. Second, TensorFlow has provided an object detection API for mobile devices, known as TensorFlow Lite. TensorFlow recommends that TensorFlow Lite be used with the SSD MobileNet model [34].

2.3 Recognizing Math Problems and Digits

Studies have shown that handwritten characters are more difficult to recognize than machine printed characters [42]. Digit identification belongs to the field of optical character recognition (OCR) [43], and has been widely studied for its ability to automate many tasks, such as license plate recognition [44], postal code sorting [45], and language translation [46]. Research in the area of OCR is so robust that TensorFlow has implemented a handwritten number analysis tutorial using the MNIST dataset [34]. CNNs are used in most state-of-the-art systems involving image processing [32], with digit classification and segmentation being no exception [43]. In some cases, Bayesian classifiers outperform CNNs in digit classification [43], though this is rare, and for our purposes, a CNN is almost certainly a better choice given the uncertainty of student handwriting samples and the inability to generate models for every student that the system may interact with. Handwritten characters are more difficult to recognize than printed characters [42]. Handwritten character recognition improves when samples are collected in real time using technology such as a smartpen, likely due to the added information from pen stroke [47].

Applying character recognition to the problem of math education, past research has successfully extracted math symbols from documents for the purpose of document analysis [48]. The majority of previous work in applying character recognition to math education has utilized real-time collection methods to improve the accuracy of student sample collection [49–51]. In these systems, the pen's movement is tracked in order to determine the digit being drawn by the student [49]. In terms of freedom of expression and application, the use of a static CNN-based OCR is typically less accurate, but more flexible for student use [52]. Since the basic structure of a math problem in our dataset is consistent, the problem of recognizing multiple math problems at once can be solved using traditional segmentation methods through the application of a CNN [32].

While work in the area of intelligent tutoring systems has existed for years [53], the application of lightweight CNN-based techniques to the problem of handwritten digit and math problem recognition is less researched, and is explored in *Chapter 3*.

2.4 Math Education Software

Software in the area of math education is abundant with most programs relying on multiple choice [54], typed entry [55], stylus drawing entry [**?**], or answer selection with the mouse [54]. While this software can be useful, it either does not provide a way for students to show their work, as in the case of multiple choice/ typed entry and mouse selection, or it does not allow them to receive guided feedback [54]. Furthermore, the hardware cost involved in making stylus entry systems and state of the art computer systems available to all students at home can create a barrier to practice outside of the home, which is often an influencing factor in the software chosen by teachers [56]. Students who visualize their errors are more likely to correct themselves and retain the knowledge about how to solve that type of problem in the future [57, 58].

Despite research showing that feedback must be appropriate for student age range and background knowledge, math education software may not always provide age appropriate feedback [59]. The strategies for problem solving and recognition of how the student made the mistake differs by age group, and has a significant impact on student retention [60]. Applications such as Photomath [26] and Wolfram Alpha [27] simply provide students with a solution to a math problem, without context, which does not give students the opportunity to learn from their mistakes [60]. Although these sites may provide outside resources for additional help [27], they often require a higher reading level, so many of the linked sites are not content appropriate for young students or those who are developmentally delayed [60]. Introducing problem solving skills, such as review and reflection on errors, is imperative for learning and retention [58, 61]. [52] creates a platform for teachers to reflect on common student errors, but does not provide a system to interact with students and prompt them to correct their mistakes. HWHelper builds on prior work by utilizing well established arithmetic rules to provide students with feedback that allows them to properly solve a math problem without giving them the solution.

Math education software marketed specifically to educators has also been studied. For example, Building Blocks is a preK/Kindergarten grade math program created by McGraw Hill [62]. Studies have shown that it provides a learning boost for socioeconomically disadvantaged preK and kindergarten students [62]. Building Blocks is meant to be an end to end curriculum, and the software component utilizes manipulatives and games to help students learn [62]. It differs from our application in age range and in the manner in which it reinforces material. Rather than giving students material and having them learn from their mistakes, it repeatedly presents them with new material. Other software used in classrooms is meant to replace instruction by teachers entirely [56], or to replace portions of instruction [63]. This differs from our application in that we wish to create an in home application to supplement in class instruction.

Other works have studied automatic grading, and step by step problem solving [50, 52, 54]. [54] uses multiple choice assessment, while [50] utilizes online data gathering techniques to analyze pen stroke rather than the actual handwritten digit, making their works inconvenient for student homework use. [52] uses keyboard entry rather than handwritten work and focuses on teacher identification of student errors, rather than prompting students to correct their own errors. The work in *Chapter 3* primarily differs from prior automatic grading work in that it focuses on providing students with:

1. Age appropriate, automated feedback to correct their own mistakes **without giving them an explicit answer** [26, 27], rather than providing teachers with feedback on student learning target accomplishment [52].

2. A handwritten free response platform for students to **flexibly record their submissions and mistakes**, rather than keyboard [52], multiple choice [57], or pen stroke entry [49].

3. A portable, computationally inexpensive framework **capable of running on portable devices**, rather than on desktops only [52]

2.5 Document Embeddings

In most NLP tasks, words are represented as vector embeddings [64, 65]. Word embeddings are used in many tasks in NLP [66, 67]. Trained using a variety of sources, word embeddings provide specific context and underlying beliefs of the authors who wrote them. These underlying beliefs are what expose bias among works from many different backgrounds [21].

Past research has shown that Word2Vec– a popular word embedding model – contains gender biases in geometric relationships between word embeddings. Specifically, the word “woman" is more closely related to “homemaker" than to “computer programmer" or “engineer" in the embedding space. The gender bias exhibited in the embeddings is most likely reflective of the societal stereotypes found in public text. These findings lead to work in embedding de-biasing [21, 68].

Document embeddings represent an entire piece of writing. This can be a document, a paragraph, or even a sentence [69]. Doc2Vec is built on Word2Vec embeddings [69]. Though there has been some work on debiasing word embeddings, there has yet to be any work on debiasing document embeddings.

A common practice in visualizing embeddings is to utilize T-SNE [66, 67]. T-SNE maps high dimensional data into a space that is easy to visualize [70].

2.6 Political and Gender Bias

There has been some work in the past in identifying political bias in media. Several studies have focused on word choice, utilizing inverse document frequency of terms as well as the frequency of affective words near personally identifiable information [71,72]. The most recent approach has been to look at the labels associated with word choice, such as the use of “illegal alien" in one news source rather than “undocumented immigrant" [73]. Typically, these methods utilize classic machine learning models, such as logistic regression (LR), K-nearest neighbors (KNN), and support vector machines (SVM) to detect political bias in articles [71–73]. Other techniques for political bias identification focus on framing [74], and identification by coverage or selection bias [75].

There has been some work in gender classification, with the majority focusing on the problem of author gender identification [76]. The state-of-the-art methodology for identifying Tweet author gender is based on n-grams and skip-grams, in conjunction with a deep learning model, such as a Recurrent Neural Network (RNN) [76]. N-grams and skip-grams are also used to identify author gender in larger documents [77]. Supervised methods have been used to calculate average sentiment of code reviews in an effort to identify gender bias. The authors found that male code reviews targeted at female programmers held more negative sentiment than female code reviews targeted at female programmers [78]. A recent study found that most named entity recognition (NER) techniques do not tag female names as "PERSON" [79]. Other approaches utilize IDF and framing [21] much like political bias studies.

A simple approach, and the one taken in this work, is to utilize a bag of words model (BOW) to determine the most common words for various classes [80]. If they vary significantly across classes, the gender being the classes in this case, then word choice varies significantly across classes. This is similar to the approach taken in [21]. However, the BOW model is extremely simplified- it consists of a single input to output neuron that is trained to predict between two classes [80]. Each word in the neuron is represented as a weight, so that the most positive weights represent one class, and the most negative weights represent the other class [80]. While the model is simplistic, we find that is an especially powerful predictor of gendered words in political media, which we discuss in Chapter 4.

Studies on the effect of gendered information (pronouns, proper nouns, and personally identifiable terms) in documents are limited. Similarly there is little-to-no work on detecting the gender of the subject of an article. The complex interaction between political leaning of news affiliates and candidate gender has not been examined to date.

Chapter 3

Homework Helper: Providing Valuable Feedback on Math Mistakes

Timely feedback is critical to student achievement [24]. However, class sizes are increasing in most U.S. public schools [81] which makes it difficult for students to receive timely, individualized feedback. Compounding this problem, more than 60% of parents are not able to help their children with homework, either due to a lack of understanding or due to time constraints [82]. Students who do not receive feedback at home or in the classroom may turn to the use of applications like Photomath [26] to quickly complete assignments as it is portable and easy to use. These type of applications do not provide students with any feedback on their mistakes, opting to simply solve the problem for them [26].

In this work, we introduce the HWHelper application, capable of analyzing student handwritten math problems for correctness, and providing relevant feedback to the student so that they can correct their work, never giving them the solution. Ultimately, learning from their own mistakes and generating their own procedure to fix

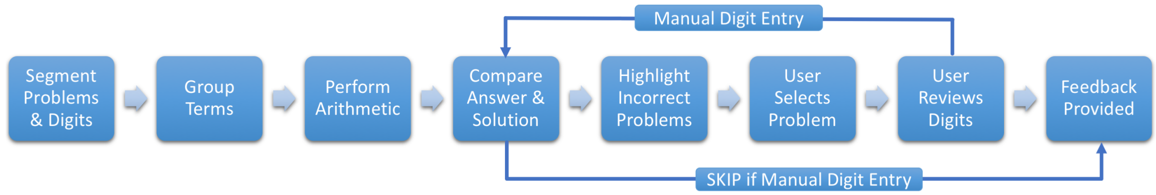

Figure 3.1: The HWHelper system work flow. Problems and digits are first segmented. Digits are then grouped into terms and the addition is performed. The student answer is compared to the solution. All incorrect problems are highlighted for the student, and the student can select problems for review. If HWHelper misidentifies any digit, the student can manually enter digits to correct the system. Error messages are then displayed, providing feedback for improvement.

their mistakes in a guided manner should lead to better content retention [83]. The HWHelper pipeline is shown in Figure 3.1.

The contributions of this work are as follows:

1. 2D-Add, a two-digit addition dataset consisting of 2010 worked problems, hand-labeled with problem and digit bounding boxes, digit values, solutions and errors.

2. HWHelper, an android application to provide automated feedback on freeform two-digit addition problems.

3.1 HWHelper Application

In this section, we detail the HWHelper application, including the data collection, math problem and digit segmentation, term grouping, arithmetic and logic for error identification and reporting.

3.1.1 Educational Standards

The HWHelper application's addition portion would be suitable for grade levels one and two under Common Core Math Standards [84]. Students need to be able to identify the ones and tens places within a number in order to be able to use the error flagging provided by the system.This falls under standard CCSS.MATH.CONTENT.1.NBT.B.2, a first grade standard. Since this would be required background knowledge, the use of HWHelper would acceptable once students are proficient in 1.NBT.B.2. The goal of HWHelper at this point is to allow students to learn to add to values up to 100. This falls under CCSS.MATH.CONTENT.1.NBT.C.4, which requires students to use concrete models, strategies, and written methods to explain what they have done. It also requires students to understand that adding two-digit numbers is accomplished by adding tens and tens, and ones and ones. Addition of two digit numbers also falls under the second grade standard CCSS.MATH.CONTENT.2.NBT.B.5, which is focused on the fluency of adding numbers up to 100 using place value and various other properties.

3.1.2 Data Collection

Two digit addition problems are collated so that there are six math problems on each page: two columns with three problems per column. Figure 3.2 shows an example of a sheet in 2D-Add. There are ten sheets of problems with fewer than six problems per page, where whitespace is randomly added; the random whitespace is added to ensure that the system does not simply learn to draw boxes in the same six spots when identifying problems.

The terms in each problem range from 1 to 49, so that no solution is greater than

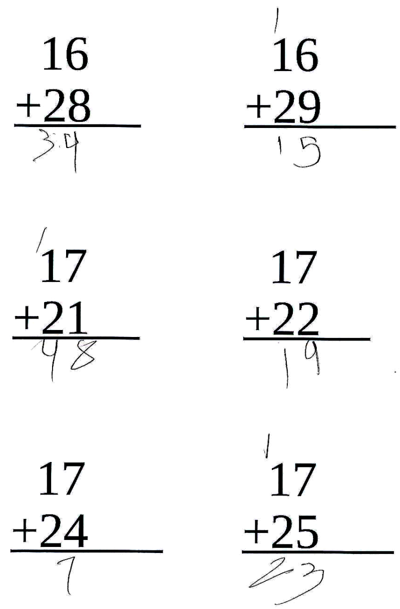

Figure 3.2: An example of a worked math sheet in 2D-Add.

two digits in length, for the sake of simplicity. Problems are not repeated, but flipping of terms is allowed. That means that $1 + 12$ is in the set, as is $12 + 1$, but neither of those patterns are repeated.

The sheets were randomly divided among five labelers, aged 20-28. Before filling out the sheets, the participants were instructed to "write like a first grader" to simulate the challenges that HWHelper might encounter in a classroom setting. They were also told to make at least 30% of their answers incorrect. This exercise generated 340 sheets of math problems, and 2010 individual problems, which make up the 2D-Add dataset. Seven possible mistakes are identified from the data collected.

There are seven consistent addition errors found in the 2D-Add dataset. They can be seen in Figure 3.3, and are listed below:

1. Drop down first and second term answer

2. Ones place added incorrectly

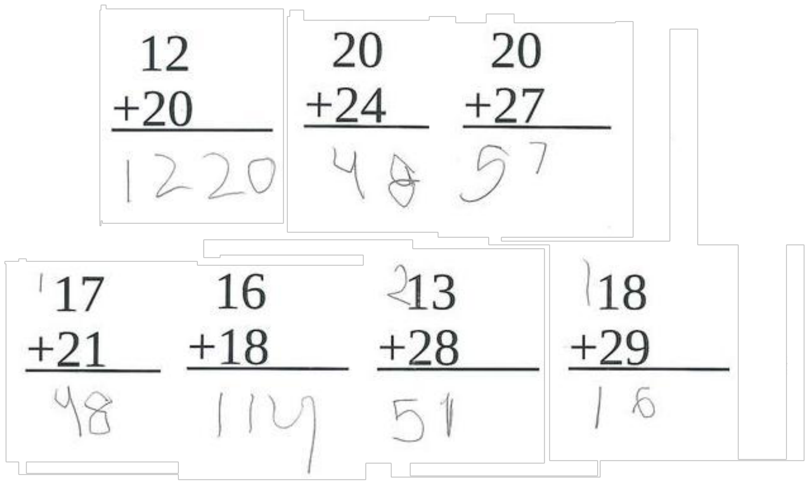

Figure 3.3: Examples of each of the seven mistakes. Errors 1, 2, 3, 4, 5, 6 and 7 are shown from left to right, top to bottom.

3. Tens place added incorrectly

4. Carried when a carry was not necessary

5. Carried incorrectly

6. Did not carry when a carry was necessary

7. Other- Typically non-numeric

3.1.3 Problem and Digit Identification

We utilize a deep convolutional neural network (CNN) as our model for problem and digit segmentation. HWHelper is a mobile application and needs to provide real-time feedback to students. With this in mind, we utilize a light-weight model capable of segmenting problems and terms in real time. While 2D-Add is composed of consistently spaced math problems, we do not want to rely on this format in the future, and so we use a CNN to segment problems rather than simply dividing the sheets into segments. This allows HWHelper to be more versatile.

In contrast to problem detection, where there is only a single class, digit identification requires that all digits (0 to 9) be both localized and recognized. Both of these steps are essential, as the precise location and value of each digit is needed in order to perform the arithmetic properly, and for HWHelper to reason about the mistakes that are made. The digit detector must be capable of identifying typed digits as well as hand-written digits in a variety of locations within the problem. Young student handwriting can be very messy, so the digit recognition must be robust to different types of handwriting.

3.1.4 Term Grouping

Once all digits have been localized and recognized, they can be grouped together to form terms in the math problem. First, all duplicate digit detections are removed, which results in the final set of digits for a math problem. This is accomplished by taking the digits with the top ten confidence scores from the model. All confidence scores are values between 0 and 1. The x and y midpoints of each pair of boxes are compared, and if their difference is greater than a threshold, they are added to the final list of digits in the problem. This removes multiple detections of the same digit. If multiple digits in the same position have a difference in midpoints less than the threshold, then the box with the higher confidence score is maintained. Since no more than seven digits should be in a single problem from the 2D-Add dataset - one carried value, and two in the first and second term and answer - the seven highest scores and their associated boxes and labels are saved as the final digits in the problem.

There are four possible terms in each problem: 1^{st} term, 2^{nd} term, *carry* and *answer*. There may not always be a carry or an answer, but the 1^{st} and 2^{nd} terms will always be in each problem. A rule-based method utilizing digit position constructs each term. Since all addition problems in 2D-Add consist of terms stacked on top of each other, digits with similar y_{min} coordinates and dissimilar x_{min} coordinates must belong to the same term. In cases where a single digit has multiple neighboring digits within a threshold, the closest digit is selected as belonging to the same term. Digits with similar y_{min} and dissimilar x_{min} values are grouped into a term with two digits. Digits without a similar y_{min} and x_{min} are kept as single digit terms.

The terms are then ordered into *carry*, 1^{st}, 2^{nd} and *answer* by comparing each term's y_{min} values. If only two terms are detected, then the *carry* and *answer* are left empty. If four terms are detected then all terms are present. If only three terms are detected, we must determine whether the *carry* or the *answer* is present. If there is no term detected in the bottom $\frac{1}{4}$ of the image, then we assume there is a *carry* and no *answer*, and vice-versa.

3.1.5 Answer and Solution Comparison

The seven errors present in 2D-Add are detailed in section 3.1.2. The algorithm below demonstrates how each error is identified.

HWHelper performs the arithmetic, adding the 1^{st} and 2^{nd} terms to find the solution to the problem. The solution is compared to the student's answer. If the student's answer does not match the solution, the specific error is determined and then feedback is provided to the student. Figure 3.3 shows an example of each of the

7 errors in the 2D-Add dataset.

Algorithm 1: HWHelper Error Generation

```
Result: Error
if answer == concat(1^st, 2^nd) then
    report Error 1
else if answer[0] != solution[0] then
    // First digit of answer is not correct.
    report Error 2
else if 1^st[0] + 2^nd[0] > 9 then
    expected_carry = 1^st[0] + 2^nd[0] - 9
    // The solution requires a carry.
    if ∃ carry & expected_carry != carry then
        // Expected carry and carry do not match.
        report Error 5
    else if ¬∃ carry then
        // No carry when there should have been.
        report Error 6
else if ∃ carry then
    // Carry when there should not have been one.
    report Error 4
else if answer[0] == solution[0] then
    if answer[1] != solution[1] then
        // Carry is correct, but second digit of answer is
           incorrect.
        report Error 3
else
    // Catchall for any other errors.
    report Error 7
```

3.2 Experiments

In this section, we detail the experiments conducted to evaluate the efficacy of the HWHelper application. For both problem detection and digit recognition experiments, 80% of the 2D-Add dataset is reserved for training, and the other 20% is set aside for validation and testing. The TensorFlow object detection API – specifically the SSD Mobilenet pre-trained on COCO – is used to segment individual problems as well as digits [39]. Our experiments are separated into problem and digit segmentation, and error reporting and prompting.

3.2.1 Problem and Digit Segmentation

For problem detection, we utilize the SSD Mobilenet pre-trained on COCO. For training, we use a learning rate of 0.01, batch size of 15, with images of size 600x600. Problem segmentation creates bounding boxes around each problem (see Figure 3.4). HWHelper achieves 100% accuracy on problem detection on 2D-Add. This indicates that given a specific template for problem location, the HWHelper system is capable of identifying the locations of each individual problem in a practice sheet. The segmented problems are then fed to a second model for digit segmentation.

For digit segmentation, we again utilize an SSD Mobilenet model pre-trained on the COCO dataset. We modify the configuration for 10-class classification (digits 0-9). In training, we again use a learning rate of 0.01 and a batch size of 15. Our model achieves an average accuracy of 95.3% accuracy on digit segmentation and recognition. The majority of misclassifications ware made on handwritten digits, with 110 of the 118 misclassified digits being handwritten. The 8 misclassified typed digits are covered by handwritten digits or scribbles, resulting in their misclassification. We implemented a user check within HWHelper to determine if a digit had been identified

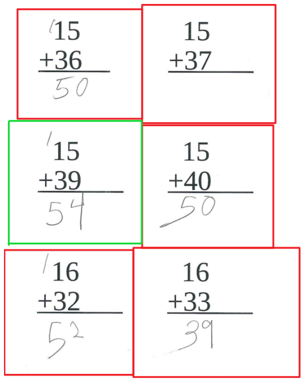

Figure 3.4: A correctly segmented arithmetic calculation sheet with color coding, where all problems except one in the sample have been incorrectly solved.

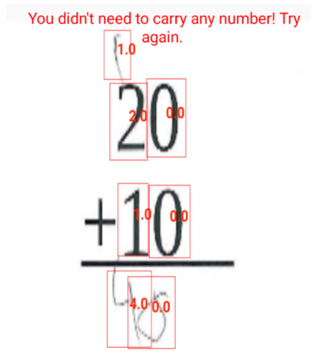

Figure 3.5: An incorrectly solved arithmetic calculation, and its associated error message.

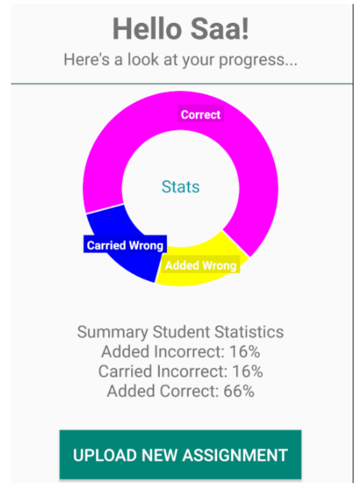

Figure 3.6: The statistics screen presented to students after uploading assignments.

correctly, as outlined in section 3.2.2.

Both the problem and digit segmentation models are converted to TensorFlow Lite for integration into HWHelper. TensorFlow Lite offers the fastest method of detection. Since HWHelper is a mobile application, it is important to utilize models with as few parameters as possible to avoid slow response times due to low computational power.

3.2.2 User Interface

The generated codes are compared to a list of hand-labeled errors for each problem. The accuracy for error reporting is 63.2% on the test set. To determine if the mistakes made by the error message production are due to the error messaging system or the detected digits, the same experiment was repeated using the ground truth values for first term, second term, carry, and answer. This resulted in a accuracy of 100%. This indicates that the 63.2% accuracy using detected digits is due to inaccurate digit identification, which can be prevented with manual digit entry.

HWHelper is implemented as an Android application. Students can take a photo of their homework using their android tablet or phone. HWHelper then performs all of the problem and digit segmentation, term grouping, and mathematical computations in its backend. Correct problems are shown in a green box while incorrect problems are shown in a red box, as shown in Figure 3.4. The user may then select a problem that has been identified as incorrectly solved. The application outputs which of the seven math errors outlined in section 3.1.2 has been found by the application, and prompts the user with an age appropriate message to try again. This process is shown in Figure 3.5.

In order to use the application, teachers and students need to create and register for an account. Once registered, students can upload sheets of two-digit addition problems for evaluation. After the student has evaluated a sheet of problems, the application presents them with a cumulative pie chart for all problems ever completed by the student. This pie chart is found on the home page and describes the relative percentages of different mistakes and the percent of problems completed correctly. An example is shown in Figure 3.6. This allows the student to reflect on their previous mistakes and assess the improvement of their skills over time.

If multiple answers are detected in a problem, HWHelper prompts the user to enter the values found on the sheet (see Figure 3.8) to ensure that the application can correctly assess the math problem. Even when the application does not detect multiple responses, it is possible that model incorrectly identified an individual digit. To ensure that the application is as user friendly as possible, the application always outputs the message “Does something seem wrong?". This is shown in Figure 3.9, where the output is particularly useful since a number has been misidentified by the system. Selecting “click here" brings the user to a screen where they can manually input the problem information (Figure 3.7), after which the application will display

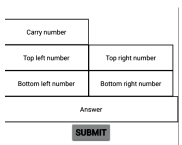

Figure 3.7: The screen that the application pulls up when the user selects “click here" after being prompted with “Does something seem wrong?"

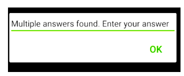

Figure 3.8: The prompt output by the Android application when multiple answers are detected in a problem.

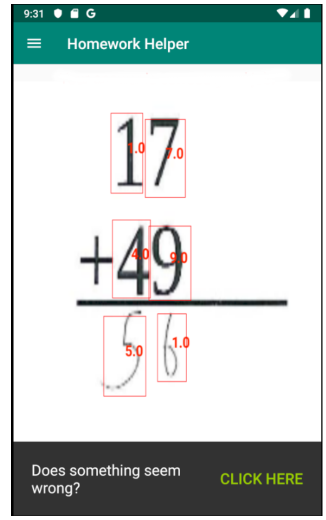

Figure 3.9: The screen shown to the user after all digits have been identified; in this particular example, not all digits have been correctly identified and the user should select “click here" to correct the misidentification.

an error message with feedback if necessary. The user may also opt to simply retake the picture if the image appears blurry to them.

3.3 Summary

In this chapter, we introduce the HWHelper application, providing automated feedback to students. With 100% accuracy on problem segmentation, HWHelper performs very well when provided with sheets of math problems that conform to a standard format like those in 2D-Add and most homework assignments. HWHelper is also capable of identifying individual digits with high accuracy – 95.3% on average. The proposed algorithms to group digits into terms are robust enough to handle a variety of inputs. HWHelper is capable of overcoming errors in digit and term detection as it allows for user input to fix such mistakes.

HWHelper shows that it is possible to use technology to provide automatic feedback to free response student work. Along with the HWHelper application, we introduce 2D-Add, a dataset of worked two-digit addition problems, which we will make available for future research in this direction.

Chapter 4

Gender Bias Recognition in Political News Articles

Several decades of research have shown that the media portrays the actions of male and female political representatives differently [85–87]. A study published in 2003 found that across 2000 gubernatorial mixed-gender races, gender stereotypes were commonly portrayed across media outlets [88]. Gender bias is present in both word choice used to present male and female subjects, and in the selection of topics covered when discussing male and female subjects.

Difference in word choice presents obstacles to women in politics. Many of the phrases that dominate the descriptions of political leaders are gendered in nature [89]. The media often portrays political candidates as being “tough" or “dominant," characteristics that often have a negative connotation with respect to female candidates [89]. The words “aggressive” and “bossy” are used in place of “tough” and “dominant”, respectively, when referring to female candidates [89]. Female political candidates are subject to additional gender constraints due to media portrayal, such as choice in dress and their own word choice [89]. According to U.S. census projections conducted in 2015, 50.8% of all Americans are women [90], yet less than 25% of congress members are women, and only 10% of all Republican representatives are women [91]. Much of the media coverage of political candidates is influenced by gender stereotypes [89].

Textual gender bias has been studied in multiple contexts using Natural Language Processing (NLP), ranging from workplace reviews [78] to word embedding analysis [92]. Despite the social implications, media gender bias remains relatively untouched by the NLP community.

The main contributions are as follows:

1. We introduce a new political dataset (News-Bias) with over 4,000 articles from Breitbart, Fox News, USA Today, HuffPost, and New York Times consisting of ten political candidates: five female, five male.

2. We evaluate several supervised methods for gender classification on the News-Bias dataset both prior to and after scrubbing the articles of gendered information.

3. We perform bag of word analysis on the document embeddings to determine the words most closely associated with each gender.

The goal of this work is not just to determine *if* gender bias in political articles can be detected, but more importantly, *how gender bias is encoded in document embeddings and what words are most strongly associated with the gender embodied by these embeddings.*

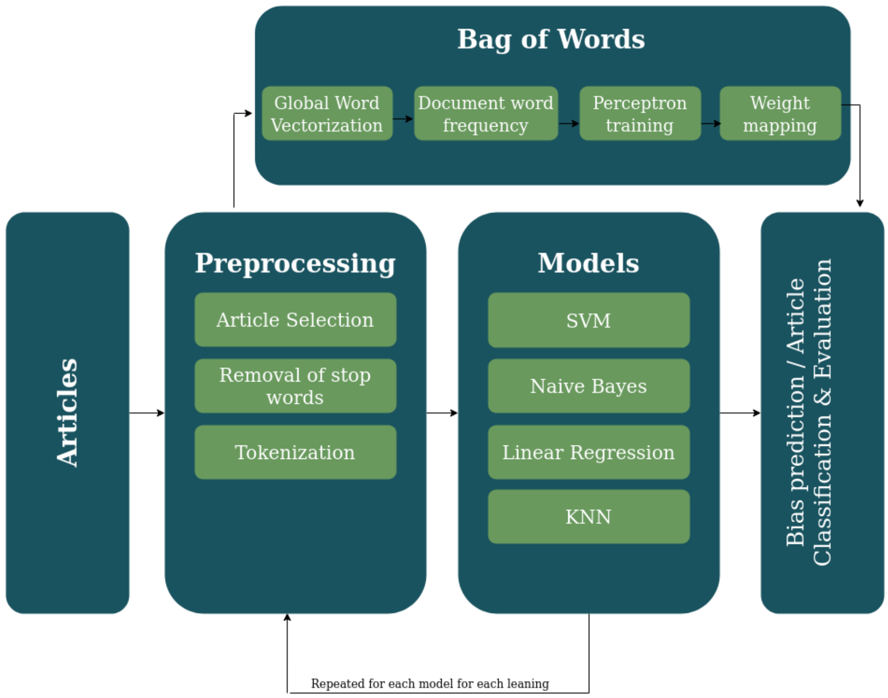

Figure 4.1: The flow of our algorithm. In the preprocessing step, articles are tokenized and stop words and gendered terms are removed. Article embeddings are generated using Doc2Vec then fed into one of four supervised classification models. A bag of words model is trained on the cleaned articles and used to analyze the most common gendered words.

4.1 Proposed Method

In this section, we detail the proposed method for identifying gender bias in political news articles, as well as methods for analyzing word and document embeddings. In the following subsections, we discuss data collection, preprocessing methods, document embeddings and gender classification, and bag of words analysis. The overall flow of the proposed work is shown in figure 4.1.

4.1.1 Dataset Collection

Over 4,000 articles are scraped from five popular online media sources: Huffpost, New York Times, USA Today, Fox News, and Breitbart given their respective leanings of far left, left, neutral, right, and far right [93].These sources were chosen because previous studies have established that these sources represent a variety of political viewpoints [93,94]. Each of these leanings are associated with the news outlet overall, but do not necessarily describe the leaning of a particular article. Therefore, all political labels associated with an article should be considered “weakly labelled." Ten political targets are chosen for this work: Hillary Clinton, Elizabeth Warren, Sarah Palin, Betsy Devos, and Alexandria Ocasio-Cortez representing the women, and Bernie Sanders, Barack Obama, Joe Biden, Mitch McConnell, and Donald Trump representing the men. The political targets were selected in an effort to represent the spectrum of political leanings across both genders. Information regarding author, article title, article subtitle, and date is also collected. Since each source has five male political candidate articles and five female political candidate articles, the dataset is equally balanced.

4.1.2 Preprocessing Methods

Preprocessing consists of several steps that are run before the data is fed to the model. First, the removal of gendered terms is done using named entity recognition and regular expressions. Woman, man, girl, boy, and any other singular gendered words, as well as proper names, are replaced with the word “person". Specific offices, such as “president of the united states", are replaced with “person" as well. Women, men, girls, boys, and any other plural gendered words are replaced with the word “people". Singular pronouns, such as he and she, are replaced with “they". Male and female are replaced by “human", while mother and father and their variants are replaced with “parent". These substitutions allow gender bias to be analyzed without the presence of gendered words. Countries, cities, and states are replaced with “gpe", and all locations that are not part of the GPE category are replaced with

Before Cleaning	**After Cleaning**
White House counselor *Kellyanne Conway* said the *Trump* administration was preparing a robust defense of *President Donald Trump* during a possible Senate impeachment trial.	White House counselor *person* said the *person* administration was preparing a robust defense of *person* during a possible Senate impeachment trial
Biden still holds a lead in *South Carolina*, where *he* is relying on strong support from *African-American* voters.	*person* still holds a lead in *gpe*, where *they* is relying on strong support from *norp* voters
Hillary Clinton said Wednesday *she* hoped for a return to "boring, normal times" after the 2020 election, voicing skepticism of *her* party's populist wing and predicting that Senator *Elizabeth Warren's* proposal for single-payer health care would never get enacted.	*person* said Wednesday they hoped for a return to "boring, normal times" after the election, voicing skepticism of *their* party' populist wing and predicting that Senator *person'* proposal for single-payer health care would never get enacted.

Figure 4.2: Sample sentences before cleaning, and after cleaning. Gendered text changes are are in italics.

"loc". Nationalities as well as religious and political groups are replaced with "norp". Finally, personally identifiable organizations, such as "department of education" are removed all together. Location and group replacement minimizes the possibility that the model learns to associate location or political group with a specific gender or political leaning. Examples of replacement can be found in figure 4.2. Preprocessing allows word choices in news articles to be analyzed independently of the possible effects of having male or female names, gender specific words such as man or woman, or inadvertent personally identifiable information in the articles.

4.1.3 Document Embeddings and Gender Classification

Doc2Vec is used embed each article after preprocessing. Any unknown words are discarded. We utilize several supervised methods to perform gender classification on the article embeddings. We detail the process of gender classification below.

Experiments are conducted using Gaussian Naive Bayes (NB), Support Vector

Machine (SVM), Linear Regression (LR), K-Nearest Neighbor (KNN), and Neural Network (NN) classifiers. We use 5-fold cross validation for each of the five classifiers.

Each classifier (NB, SVM, LR, KNN, NN) is trained to distinguish gender on each leaning (far left, left, neutral, right, far right), producing 25 models for each split. The goal of each model is to determine the gender of the subject of the article *without the use of gendered words or the subject's name*, since they have been removed in preprocessing. If a classifier is capable of determining the gender of a subject with better than chance precision, recall, and f1, then it can be assumed that the classifier– and by extension, the document embedding– has expressed some form of gender bias. That model is then further evaluated to determine the way that the embedding and the model are encoding gender bias.

	Breitbart	**Fox News**	**USA Today**	**HuffPost**	**NYT**
F1	0.89	0.95	0.86	0.90	0.87

Table 4.1: The best average F1 scores across all five folds when classifying gender using a NN trained on Doc2Vec embeddings from the cleaned articles.

4.1.4 Bag of Words

A bag of words evaluation is performed on the cleaned articles to determine the top words associated with each gender. All of the words in all articles excluding stop words, and any cleaned information is vectorized and counted so that there is a count word vector that describes the entire document word space. Next, a count word vector is created for the male and female articles separately. Then, a simple input to output neural network (perceptron) is trained on the labeled count word vectors, and F1 scores and accuracy are recorded. Finally, the largest ten weights corresponding to male and the smallest ten weights corresponding to female are mapped back to their respective words in the original count word vector to determine the words most closely associated with their respective gender.

4.2 Experiments

In this section we outline the experiments conducted to evaluate the bias within each article for each political leaning. A summary of the average F1 results across all five folds can be found in table 4.1. Each model is trained and evaluated on a five-fold cross validation scheme where each fold is randomized with six political figures being set aside for training, two for validation, and two more for testing. The candidate names are then used to filter the given articles for each fold.

4.2.1 Document Embeddings and Visualization

For each dataset, document embeddings are generated with Doc2Vec from the gensim library [95] using the cleaned data outlined in section 4.1.2. The model itself is an implementation of the Distributed Bag of Words version of Paragraph Vector (PV-DBOW) [69] trained against the cleaned articles and is generated with a document vector size of 50, trained over 100 epochs, with the default learning rate set to 0.001. Before training, the document order is randomized.

The document embeddings for each leaning are plotted using t-SNE [70] from Scikit-Learn 0.21.3 [96]. This shows any document trends by gender and correlations that are present in the different political leanings. This is done for each political leaning so as to show the gender relationships in the data present for only one leaning. Figure 4.3 shows that the document embeddings for Breitbart, HuffPost, Fox News, New York Times, and USA Today are generally linearly separable after t-SNE visualization. We emphasize that this is remarkable because all gendered information has been removed from the articles prior to generating document embeddings.

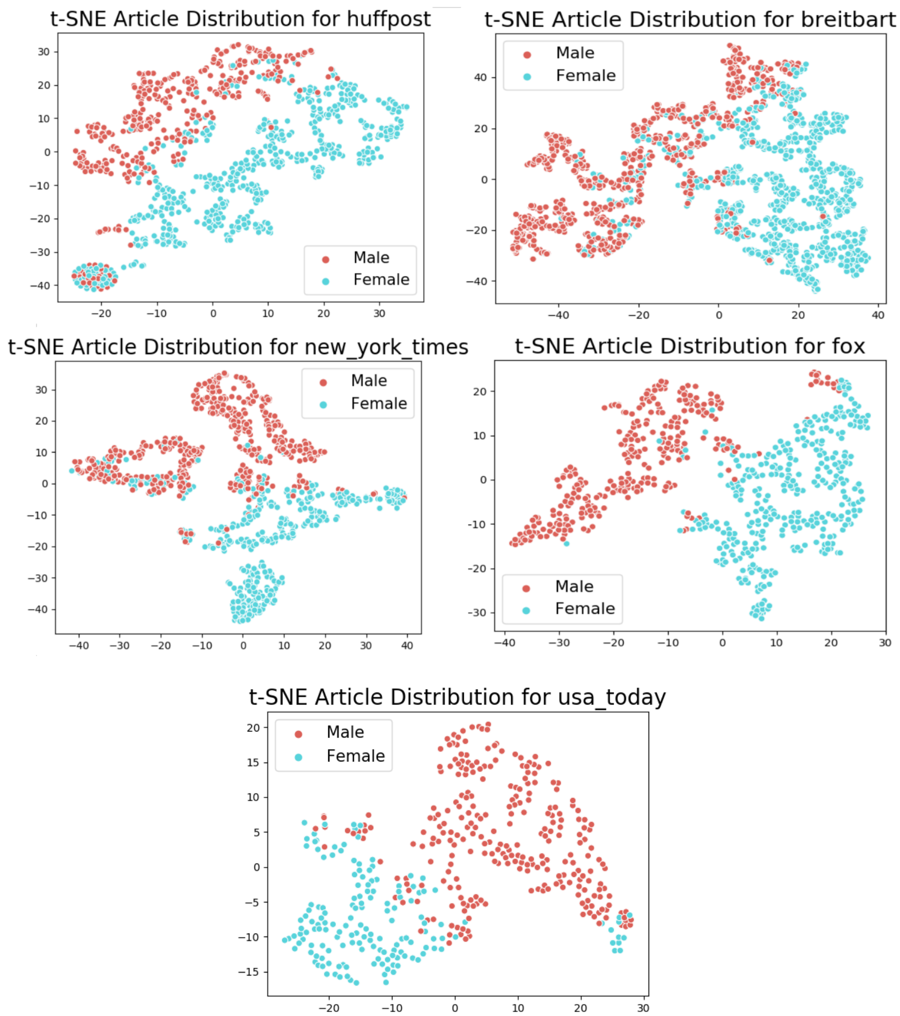

Figure 4.3: The t-SNE mapped cleaned document embeddings for all articles from Breitbart, Huffpost, Fox News, New York Times, and USA Today. We can see a clear delineation between male and female targets, even after all gendered information is removed from the articles.

Even without gendered information, the Doc2Vec embeddings show a clear gender distinction.

4.2.2 Models

Each of the models - SVM, KNN, NB, LR, and NN - are implemented in Python 3.6.7 using Scikit-Learn 0.21.3 [96]. Typical metrics such as average precision, average recall, and average f-measure are evaluated first, as these provide insight into the ability of a classifier to accurately distinguish between the document embeddings used to describe male and female subjects. The higher these scores, the better performing the classifier, and the better able the classifier is to distinguish between male and female political candidate subject matter. A higher degree of gender bias is represented by a larger recall, precision, and f-measure, since these scores mean that the classifier is better able to distinguish between when an article is about a male political candidate and when an article is about a female candidate. The basic flow we created for bias detection, including preprocessing, document embedding process, models, and visualizations is shown in figure 4.1.

4.2.3 Bag of Words

Male F1-Score = 70%	**Female F1-Score = 69%**
fans	gov
elections	believe
award	responded
hopeful	lawmaker
hear	freshman
netflix	put
stated	find
majority	formal
pointed	strange
watch	power

Figure 4.4: The top 10 words associated with each gender using a bag of words model on the cleaned document embeddings.

The results for bag of words can be seen in Figure 4.4. The overall accuracy of the perceptron used to find the top ten words is 69.38%. The f1-score for male classification is 70%, and 69% for female classification. Given the simplicity of the direct input to output perceptron used as our bag of words classifier, these scores are quite impressive and indicate that the top ten words are quite strongly aligned with the two genders.

4.3 Summary

In this work we present a method to detect gender bias in political writing, along with a dataset (News-Bias) of over 4,000 political news articles from five sources ranging from far left to far right. Using gender classification, we show that gender bias is present in most political writings, and this bias holds even after all gendered information is removed from the articles (figure 4.1). Visualizing the cleaned document embeddings shows a clear delineation between male and female articles, indicating that removal of gendered pronouns, proper nouns, and personally identifiable information has little effect on the ability of Doc2Vec to distinguish between genders.

Chapter 5

Conclusion and Future Work

Though in its initial stages, the research outlined in HWHelper and Gender Bias in Political News Articles serves as a solid body of work to be built upon in future work in understanding bias using machine learning. While the work itself is still very new and should be expanded upon, the results are promising. With outreach to local schools for testing, HWHelper's performance could easily be improved, and its mathematical capabilities expanded. The work done on detecting Gender Bias in Political News Articles can easily be extended to other embedding methods, and other news outlets and target candidates. While extendable, the work presented here serves to mitigate real world bias in the areas of socioeconomic status, & education, and gender.

We provide an easily accessible homework system that is capable of providing meaningful feedback to students in a way that is meant to help them learn optimally, unlike existing systems. The system allows students from a variety of socioeconomic situations to easily gain the feedback necessary to grow in the area of mathematics. Normally, students from poverty stricken families are less able to gain access to adults to help them with their homework for a variety of reasons [?, 29, 31]. This means an autonomous system can become an invaluable tool for students in situations where they need to practice math skills, but are struggling with the material and do not have a knowledgeable adult available. A tool like HWHelper mitigates socioeconomic bias against students from a low socioeconomic background by providing them with access to the help they need, and allowing them to practice their math skills. The ultimate goal of HWHelper is to help ease the systemic negative educational impacts that exist for students who come from families of modest means, including poor performance in middle and high school [12], and more importantly to our research, the negative impact found in the area of mathematics [13] and low graduation rates [2]. Improving graduation rates for impoverished students would improve their socioeconomic status, destabilizing the cycle of poverty [14], serving to help undermine the systemic bias that exists in education as a result of socioeconomic status.

Future research will explore variations in problem layout, such as side by side addition, to accommodate a wider classroom audience. Additionally, the framework will be expanded to include the capability to perform different arithmetic calculations, such as subtraction and multiplication, which could be accomplished by detecting arithmetic symbols in addition to digits. The modularity of the current HWHelper will make the addition of other operators (subtraction and multiplication) very reasonable. We will collect additional data from students in the classroom working with local K-12 schools. This additional data will improve digit recognition and the HWHelper system overall. We will also perform usability testing within the target population, to identify areas for improvement in the user interface and feedback portions of HWHelper.

The work in Gender Bias in Political News Articles serves to help understand systemic gender bias by creating a method of detecting gender bias in political media. Gender bias detection in political news is a relatively unexplored area in NLP, despite the societal implications. Gender bias not only impacts the way that female athletes are portrayed in the media [16], but also has an impact on the way that female political candidates dress, the word choice associated with female candidates [89], and the overall rates of female candidates in congress in the U.S. [90, 91]. This indicates that the study of gender bias in political news articles is of significant importance. Eventually, it could be used to create early gender bias detection systems for news sites, further mitigating gender bias

Our work in identifying gender bias in political news articles is a first step in the direction of understanding systemic gender bias in political news. Future work should explore debiasing document embeddings, much like previous work done to word embeddings, to determine if gender separability in vector space can be removed. We also plan to add more data to the News-Bias dataset allowing for broader conclusions to be drawn. Furthermore, annotating News-Bias with sentiment labels would allow us to train a sentiment classifier directly from news data, allowing us to analyze difference in sentiment between male and female political candidates. Visualizing the activations of the NN gender classifier may provide deeper insight into the specific embedding dimensions that have the greatest effect on gender classification. Finally, looking at the effect of the gradual removal of gendered pronouns, proper nouns, and personally identifiable information from documents prior to running neural network model analysis would allow a unique insight into the effect of specific gendered words on the ability of the neural network to discriminate between genders.

Bibliography

[1] J. A. Ritter and L. J. Taylor, "Racial disparity in unemployment," *The Review of Economics and Statistics*, vol. 93, no. 1, pp. 30–42, 2011.

[2] S. Sahin, Z. Arseven, and A. Kiliç, "Causes of student absenteeism and school dropouts." *International Journal of Instruction*, vol. 9, no. 1, pp. 195–210, 2016.

[3] T. H. Shore, "Subtle gender bias in the assessment of managerial potential," *Sex Roles*, vol. 27, no. 9-10, pp. 499–515, 1992.

[4] M. Thomas, "The impact of mandated maternity benefits on the gender differential in promotions: Examining the role of adverse selection," 2016.

[5] M. Yap and A. M. Konrad, "Gender and racial differentials in promotions: Is there a sticky floor, a mid-level bottleneck, or a glass ceiling?" *Relations Industrielles/Industrial Relations*, vol. 64, no. 4, pp. 593–619, 2009.

[6] M. Schwarz and M. Thompson, *Divided we stand: Redefining politics, technology, and social choice.* University of Pennsylvania Press, 1990.

[7] C. Cheney, J. LaFrance, and T. Quinteros, "Institutionalized oppression definitions," *Tri-County Domestic & Sexual Violence Intervention Network Anti-Oppression Training for Trainers*, 2006.

[8] W. Fan and C. M. Williams, "The effects of parental involvement on students' academic self-efficacy, engagement and intrinsic motivation," *Educational psychology*, vol. 30, no. 1, pp. 53–74, 2010.

[9] C. E. Cooper, "Family poverty, school-based parental involvement, and policy-focused protective factors in kindergarten," *Early Childhood Research Quarterly*, vol. 25, no. 4, pp. 480–492, 2010.

[10] M.-A. Suizzo and L. M. Stapleton, "Home-based parental involvement in young children's education: Examining the effects of maternal education across us ethnic groups," *Educational Psychology*, vol. 27, no. 4, pp. 533–556, 2007.

[11] D. Cloney, G. Cleveland, J. Hattie, and C. Tayler, "Variations in the availability and quality of early childhood education and care by socioeconomic status of neighborhoods," *Early Education and Development*, vol. 27, no. 3, pp. 384–401, 2016.

[12] D. M. Fergusson, L. J. Horwood, and J. M. Boden, "The transmission of social inequality: Examination of the linkages between family socioeconomic status in childhood and educational achievement in young adulthood," *Research in Social Stratification and Mobility*, vol. 26, no. 3, pp. 277–295, 2008.

[13] S. J. Ritchie and T. C. Bates, "Enduring links from childhood mathematics and reading achievement to adult socioeconomic status," *Psychological science*, vol. 24, no. 7, pp. 1301–1308, 2013.

[14] M. Metzler, M. T. Merrick, J. Klevens, K. A. Ports, and D. C. Ford, "Adverse childhood experiences and life opportunities: shifting the narrative," *Children and youth services review*, vol. 72, pp. 141–149, 2017.

[15] J. D. Katz, G. Mamyrova, O. Guzhva, and L. Furmark, “Gender bias in diagnosing fibromyalgia,” *Gender medicine*, vol. 7, no. 1, pp. 19–27, 2010.

[16] C. T. Higgs, K. H. Weiller, and S. B. Martin, “Gender bias in the 1996 olympic games: A comparative analysis,” *Journal of Sport and Social Issues*, vol. 27, no. 1, pp. 52–64, 2003. [Online]. Available: https://doi.org/10.1177/0193732502239585

[17] F. D. Blau and L. M. Kahn, “The gender pay gap,” *The Economists' Voice*, vol. 4, no. 4, 2007.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.

[19] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” 2014.

[21] T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai, “Man is to computer programmer as woman is to homemaker? debiasing word embeddings,” 2016.

[22] T. Manzini, Y. C. Lim, Y. Tsvetkov, and A. W. Black, “Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings,” *arXiv preprint arXiv:1904.04047*, 2019.

[23] D. Sanders and D. S. Welk, “Strategies to scaffold student learning: Applying vygotsky's zone of proximal development,” *Nurse educator*, vol. 30, no. 5, pp. 203–207, 2005.

[24] R. Marzano, D. Robert J. Marzano, D. Pickering, J. Pollock, A. for Supervision, and C. Development, *Classroom Instruction that Works: Research-based Strategies for Increasing Student Achievement*, ser. Gale virtual reference library. Association for Supervision and Curriculum Development, 2001. [Online]. Available: https://books.google.com/books?id=c25kDO0adxwC

[25] J. Underwood, "Effective feedback: Guidelines for improving performance," 2008.

[26] Photomath, Inc, "Photomath," 2019. [Online]. Available: https://photomath.net/

[27] Wolfram Research, Inc., "Mathematica, Version 12.0," 2019, champaign, IL, 2019.

[28] K. Highfield, K. Goodwin *et al.*, "Apps for mathematics learning: a review of'educational'apps from the itunes app store," 2016.

[29] K. V. Hoover-Dempsey, A. C. Battiato, J. M. Walker, R. P. Reed, J. M. DeJong, and K. P. Jones, "Parental involvement in homework," *Educational psychologist*, vol. 36, no. 3, pp. 195–209, 2001.

[30] E. A. Patall, H. Cooper, and J. C. Robinson, "Parent involvement in homework: A research synthesis," *Review of educational research*, vol. 78, no. 4, pp. 1039–1101, 2008.

[31] J. M. Walker, K. V. Hoover-Dempsey, D. R. Whetsel, and C. L. Green, "Parental involvement in homework," *A Review of Current Research and Its Implications for teachers, After School Program Staff, and Parent Leaders. Harvard Family Research Project*, 2004.

[32] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," in *2017 International Conference on Communication and Signal Processing (ICCSP)*, April 2017, pp. 0588–0592.

[33] I. Stančin and A. Jović, "An overview and comparison of free python libraries for data mining and big data analysis," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2019, pp. 977–982.

[34] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016. [Online]. Available: http://arxiv.org/abs/1603.04467

[35] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1440–1448.

[36] Chengcheng Ning, Huajun Zhou, Yan Song, and Jinhui Tang, "Inception single shot multibox detector for object detection," in *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2017, pp. 549–554.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[38] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1–9.

[39] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *ArXiv*, vol. abs/1704.04861, 2017.

[40] A. Gupta, R. Puri, M. Verma, S. Gunjyal, and A. Kumar, "Performance comparison of object detection algorithms with different feature extractors," in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, March 2019, pp. 472–477.

[41] C. Ryan and J. M. Lewis, "Computer and internet use in the united states: 2015."

[42] P. D. Gader, J. M. Keller, R. Krishnapuram, Jung-Hsien Chiang, and M. A. Mohamed, "Neural and fuzzy methods in handwriting recognition," *Computer*, vol. 30, no. 2, pp. 79–86, Feb 1997.

[43] B. Kir, C. Oz, and A. Gulbag, "The application of optical character recognition for mobile device via artificial neural networks with negative correlation learning algorithm," in *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, Nov 2013, pp. 220–223.

[44] C. Chuang, L. Tsai, M. Deng, J. Hsieh, and K. Fan, "Vehicle licence plate recognition using super-resolution technique," in *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug 2014, pp. 411–416.

[45] D. X. Ni, "Application of neural networks to character recognition," *Proceedings of students/faculty research day, CSIS, Pace University, May 4th*, 2007.

[46] M. Koga, R. Mine, T. Kameyama, T. Takahashi, M. Yamazaki, and T. Yamaguchi, "Camera-based kanji ocr for mobile-phones: Practical issues," in *Eighth*

International Conference on Document Analysis and Recognition (ICDAR'05). IEEE, 2005, pp. 635–639.

[47] K. B. M. R. Batuwita and G. E. M. D. C. Bandara, "An improved segmentation algorithm for individual offline handwritten character segmentation," in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 2, Nov 2005, pp. 982–988.

[48] R. Zanibbi and L. Yu, "Math spotting: Retrieving math in technical documents using handwritten query images," in *2011 International Conference on Document Analysis and Recognition*, Sep. 2011, pp. 446–451.

[49] L. Anthony, J. Yang, and K. R. Koedinger, "A paradigm for handwriting-based intelligent tutors," *International Journal of Human-Computer Studies*, vol. 70, no. 11, pp. 866–887, 2012.

[50] G.-J. Hwang and H.-F. Chang, "A formative assessment-based mobile learning approach to improving the learning attitudes and achievements of students," *Comput. Educ.*, vol. 56, no. 4, pp. 1023–1031, May 2011. [Online]. Available: http://dx.doi.org/10.1016/j.compedu.2010.12.002

[51] X. Huang, S. D. Craig, J. Xie, A. Graesser, and X. Hu, "Intelligent tutoring systems work as a math gap reducer in 6th grade after-school program," *Learning and Individual Differences*, vol. 47, pp. 258–265, 2016.

[52] M. Q. Feldman, J. Y. Cho, M. Ong, S. Gulwani, Z. Popović, and E. Andersen, "Automatic diagnosis of students' misconceptions in k-8 mathematics," in *Pro-*

ceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 2018, pp. 1–12.

[53] K. VanLEHN, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," *Educational Psychologist*, vol. 46, no. 4, pp. 197–221, 2011. [Online]. Available: https://doi.org/10.1080/00461520.2011.611369

[54] H.-Y. Sung and G.-J. Hwang, "A collaborative game-based learning approach to improving students' learning performance in science courses," *Computers & Education*, vol. 63, p. 43–51, 04 2013.

[55] G. A. Cayton-Hodges, G. Feng, and X. Pan, "Tablet-based math assessment: What can we learn from math apps?" *Journal of Educational Technology Society*, vol. 18, no. 2, pp. 3–20, 2015. [Online]. Available: http://www.jstor.org/stable/jeductechsoci.18.2.3

[56] S. Okumuş, L. Lewis, E. Wiebe, and K. Hollebrands, "Utility and usability as factors influencing teacher decisions about software integration," *Educational Technology Research and Development*, vol. 64, no. 6, pp. 1227–1249, 2016. [Online]. Available: http://www.jstor.org/stable/26746593

[57] R. Borasi, "Capitalizing on errors as "springboards for inquiry": A teaching experiment," *Journal for Research in Mathematics Education*, vol. 25, no. 2, pp. 166–208, 1994. [Online]. Available: http://www.jstor.org/stable/749507

[58] S. Isotani, D. Adams, R. E. Mayer, K. Durkin, B. Rittle-Johnson, and B. M. McLaren, "Can erroneous examples help middle-school students learn decimals?" in *Towards Ubiquitous Learning*, C. D. Kloos, D. Gillet, R. M. Crespo García,

F. Wild, and M. Wolpers, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 181–195.

[59] B. Rittle-Johnson, M. Schneider, and J. R. Star, “Not a one-way street: Bidirectional relations between procedural and conceptual knowledge of mathematics,” *Educational Psychology Review*, vol. 27, no. 4, pp. 587–597, 12 2015, copyright - Springer Science+Business Media New York 2015; Document feature.

[60] M. Kapur, “Productive failure in learning math,” *Cognitive science*, vol. 38 5, pp. 1008–22, 2014.

[61] A. C. Butler, J. D. Karpicke, and H. Roediger, “Correcting a metacognitive error: feedback increases retention of low-confidence correct responses.” *Journal of experimental psychology. Learning, memory, and cognition*, vol. 34 4, pp. 918–28, 2008.

[62] M. E. Foster, J. L. Anthony, D. H. Clements, J. Sarama, and J. M. Williams, “Improving mathematics learning of kindergarten students through computer-assisted instruction,” *Journal for Research in Mathematics Education*, vol. 47, no. 3, pp. 206–232, 2016. [Online]. Available: http://www.jstor.org/stable/10.5951/jresematheduc.47.3.0206

[63] Y. Zengin and E. Tatar, “Integrating dynamic mathematics software into cooperative learning environments in mathematics,” *Journal of Educational Technology Society*, vol. 20, no. 2, pp. 74–88, 2017. [Online]. Available: http://www.jstor.org/stable/90002165

[64] T. Mikolov, G. Corrado, K. Chen, and J. Dean, “Efficient estimation of word representations in vector space,” 01 2013, pp. 1–12.

[65] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[66] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, "Sentiment embeddings with applications to sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 496–509, Feb 2016.

[67] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings for sentiment analysis," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.* Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 534–539.

[68] M.-E. Brunet, C. Alkalay-Houlihan, A. Anderson, and R. Zemel, "Understanding the origins of bias in word embeddings," 2018.

[69] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014. [Online]. Available: http://arxiv.org/abs/1405.4053

[70] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html

[71] G. Grefenstette, Y. Qu, J. Shanahan, and D. Evans, "Coupling niche browsers and affect analysis for an opinion mining application." 01 2004, pp. 186–194.

[72] S. Lim, A. Jatowt, and M. Yoshik, "Towards bias inducing word detection by linguistic cue analysis in news articles," 2018. [Online]. Available: https://db-event.jpn.org/deim2018/data/papers/275.pdf

[73] F. Hamborg, A. Zhukova, and B. Gipp, "Automated identification of media bias by word choice and labeling in news articles," in *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, June 2019, pp. 196–205.

[74] L. H. Cheeks and A. Gaffar, "A social influence model for exploring double subjectivity through news frames in online news," in *2017 Intelligent Systems Conference (IntelliSys)*, Sep. 2017, pp. 23–33.

[75] K. Lazaridou, R. Krestel, and F. Naumann, "Identifying media bias by analyzing reported speech," in *2017 IEEE International Conference on Data Mining (ICDM)*, Nov 2017, pp. 943–948.

[76] S. Krüger and B. Hermann, "Can an online service predict gender? on the state-of-the-art in gender identification from texts," in *2019 IEEE/ACM 2nd International Workshop on Gender Equality in Software Engineering (GE)*, May 2019, pp. 13–16.

[77] J. Doyle and V. Keselj, "Automatic categorization of author gender via n-gram analysis," 01 2005.

[78] R. Paul, A. Bosu, and K. Z. Sultana, "Expressions of sentiments during code reviews: Male vs. female," in *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Feb 2019, pp. 26–37.

[79] N. Mehrabi, T. Gowda, F. Morstatter, N. Peng, and A. Galstyan, "Man is to person as woman is to location: Measuring gender bias in named entity recognition," 2019.

[80] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 43–52, 2010.

[81] "National center for education statistics." [Online]. Available: https://nces.ed.gov/fastfacts/display.asp?id=28

[82] U.S. Department of Education, "Parent and family involvement in education: Results from the national household education surveys program of 2016," 2016. [Online]. Available: https://nces.ed.gov/pubs2017/2017102.pdf

[83] R. Borasi, "Capitalizing on errors as "springboards for inquiry": A teaching experiment," *Journal for Research in Mathematics Education*, vol. 25, no. 2, pp. 166–208, 1994. [Online]. Available: http://www.jstor.org/stable/749507

[84] N. G. Association *et al.*, "Common core state standards," *Washington, DC*, 2010.

[85] J. L. Lawless, "Sexism and gender bias in election 2008: A more complex path for women in politics," *Politics and Gender*, vol. 5, no. 1, p. 70–80, 2009.

[86] D. Niven, "A fair test of media bias: Party, race, and gender in coverage of the 1992 house banking scandal," *Polity*, vol. 36, no. 4, pp. 637–649, 2004. [Online]. Available: https://doi.org/10.1086/POLv36n4ms3235406

[87] K. Wilz, "Bernie bros and woman cards: Rhetorics of sexism, misogyny, and constructed masculinity in the 2016 election," *Women's Studies in Communication*, vol. 39, no. 4, pp. 357–360, 2016. [Online]. Available: https://doi.org/10.1080/07491409.2016.1227178

[88] M. C. Banwart, D. G. Bystrom, and T. Robertson, "From the primary to the general election: A comparative analysis of candidate media coverage in mixed-gender 2000 races for governor and u.s. senate: Prod," *The American Behavioral Scientist*, vol. 46, no. 5, pp. 658–676, 01 2003, name - Senate; Copyright - Copyright SAGE PUBLICATIONS, INC. Jan 2003; Last updated - 2019-11-23; CODEN - ABHSAU; SubjectsTermNotLitGenreText

- States-US. [Online]. Available: http://unr.idm.oclc.org/login?url=https://search.proquest.com/docview/214767391?accountid=452

[89] M. C. Kittilson, "Gender and elections: Shaping the future of american politics: Susan j. carroll and richard l. fox (eds)," *Party Politics*, vol. 18, no. 2, pp. 287–288, 2012. [Online]. Available: https://doi.org/10.1177/1354068811433320

[90] S. L. Colby and J. M. Ortman, "Projections of the size and composition of the u.s. population: 2014 to 2060," 2015. [Online]. Available: https://www.census.gov/content/dam/Census/library/publications/2015/demo/p25-1143.pdf

[91] D. Kurtzleben, "The new congress has a record number of women - but very few republican women," Jan 2019. [Online]. Available: https://www.npr.org/2019/01/03/681027434/the-new-congress-has-a-record-number-of-women-but-very-few-republican-women

[92] T. Sun, A. Gaut, S. Tang, Y. Huang, M. ElSherief, J. Zhao, D. Mirza, E. Belding, K.-W. Chang, and W. Y. Wang, "Mitigating gender bias in natural language processing: Literature review," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1630–1640.

[93] K. Baraniak and M. Sydow, "News articles similarity for automatic media bias detection in polish news portals," in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sep. 2018, pp. 21–24.

[94] Y. Woldemariam, "Sentiment analysis in a cross-media analysis framework," in *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, March 2016, pp. 1–5.

[95] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[96] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.