Replicating 'Man is to Computer Programmer as Woman is to Homemakers? Debiasing Word Embeddings'

Angela Ellis, Aldo Polanco, Aishwarya Varma, Darryl York III
Computer Science Department, Carleton College, Northfield, Minnesota

1 Introduction

Machine learning focuses on creating models that learn automatically and function without needing human intervention. As positive as some of the applications for machine learning are, it is also used for tracking, surveillance, warfare, and decision-making. We need to be cognizant of equity and fairness as advancements continue to be made in this field.
Word embeddings are text data to vector representation tools that are a popular framework for many machine learning and natural language processing tasks. Because they are derived from human sources, there are implicit and explicit gender stereotypes present in the embeddings. The widespread use of the tool makes these disturbing findings a cause for concern, as they can potentially amplify these stereotypes. Bolukbasi et al. proposed a methodology to remove these gender stereotypes in their 2016 paper titled, "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings"[3]. Ultimately, they observed significantly reduced gender bias in the embeddings. We were tasked with replicating, testing, and visualizing this methodology to affirm the results reported by the original researchers and test their model's robustness.

What is a Word Embedding? A word embedding can be described as a collection of vectors, with direction and magnitude, used for text analysis and text generation through mapping words in a text corpora into individual word vectors. Each of these individual word vectors have a relationship to one another in a multi-dimensional space and In the context of implicit bias, word embeddings allow us to take a deeper look at our use of language and, in turn, highlight the relationships between words, their surrounding text, and other words found in the embedding as a whole. As stated in the original paper (OP)[3], the geometry of these vectors can reflect the gender stereotypes found in our society.

Word Embedding Applications A good example of the applications and importance of word embeddings is highlighted by the OP below,

> "Suppose [a] search query is *cmu computer science phd student* for a computer science Ph.D. student at Carnegie Mellon University. Now, the directory offers 127 nearly identical web pages for students — these pages differ only in the names of the students. A word embedding's semantic knowledge can improve relevance by identifying, for examples, that the terms *graduate research assistant* and *phd student* are related. However, word embeddings also rank terms related to computer science closer to male names than female names (e.g., the embeddings give *John:computer programmer :: Mary:homemaker*). The consequence is that, between two pages that differ only in the names Mary and John, the word embedding would influence the search engine to rank John's web page higher

than Mary. In this hypothetical example, the usage of word embedding makes it even harder for women to be recognized as computer scientists and would contribute to widening the existing gender gap in computer science."[3]

As highlighted above, some results of machine learning are not reflective of their intended purpose. While word embeddings, search engines, and machine learning models are not to blame for our stereotypic associations with words, there is work to be done to make sure that these stereotypical associations are not reinforced in our use of these technologies.
For our purposes, we use the vector mathematics proposed by the OP to expose biased gender relationships between words that go unrecognized or unproven through other forms of text analysis. This is possible because word embeddings are trained with very large amounts of data trained only on word co-occurrence, allowing more representative relationships to be exposed than a quantitative comparison of the words that show up next to each other the most. Luckily, Google's Word2Vec tool kindly created a pre-trained word embedding for our use!

2 Dataset description

The paper used the Google News Word2Vec, a tool which takes Google News articles as input and outputs each word in the text as vectors. It first constructs a vocabulary from the training text data and then learns the vector representation of words by using Mikolov et al.'s Skip-gram model. This generates context-target word pairs in each sentence, capturing words that can be found next to one another. Once this context is learned, it can be used to train the word embedding. The full embedding contains 3 million words and has 300 dimensions. These tools have been used in many machine learning and natural language processing tasks, including parsing through consumer feedback, spam detection, and information retrieval (e.g. search engines).

3 Method

We structured our work in four phases: we 1) determined the words that were appropriately gendered in the embedding, 2) used the debiasing algorithm the words, 3) generated analogies to assess the success of the algorithm, and 4) used data visualizations to compare and confirm the other phases of the work. The OP determines two methods to mitigate gender bias in a word embedding. The first, titled hard debiasing, has two steps: neutralize and equalize. Its alternative, soft debiasing, only has one, soften. Both have an 'identify gender subspace' step preceeding them. The previous experiment notes that hard debiasing 'completely removes pair bias'. As such, this project focused only on reimplementing hard debiasing.

3.1 Linear SVM

The debiasing step requires knowledge of which words are gender-specific (GS) versus gender-neutral (GN). For example, "king" and "queen" are appropriately associated with "man" and "woman," so they are GS. "Computer programmer" and "homemaker," however, should be GN.

Manually determining which words are GS and GN is infeasible: the embedding consists of 3 million words. To address this, we instead used a linear support vector classifier, a supervised learning model for data classification. If a dataset can be separated into two classes by using a single straight line, then the data is termed linearly separable data, and the classifier is a Linear Support Vector Machine (Linear SVM). Linear SVMs are Bolukbasi et al.'s chosen method of classification, and it is preferred by many because it produces accurate results with less computational power. In essence, the SVM algorithm finds a hyperplane (which can also be represented as a straight line) in an N-dimensional word embedding that distinctly classifies the data points into two classes. In our case, the data points are word vectors, and the classes are GN and GS words. There are many possible hyperplanes that could be chosen to separate the GS and GN words, so the goal is to find the plane that maximizes the margin, or the distance between the support vectors (the data point that lies closest to the plane) and the plane. The larger the margin, the more confident we can be that future data points lie in either category.

We used scikit learn's Linear SVC implementation for the classification. We trained the SVM on the first 27,000 words in the embedding, of which we knew the words that were GS and GN. These 218 GS words

were provided to us by Bolukbasi et al. in their Github repository. The words that were *not* GS were then labeled as GN, both in the training and the testing step. After training the classifier, we used it to classify all 3 million words in the word embedding.

We found 6,399 GS words, of which 88% of the words in our list were in common with the list generated by Bolukbasi et al. However, 12% of the words in our list were not in their list, and 749 words (10% of their list) were not present in our set. While these results might seem skewed, it's important to note that there were nonsensical GS words found in both lists(i.e. "Senate_Permanent_Subcommittee" in theirs and "guh" in ours). The differences in our results can probably be attributed to small differences in the initialization of the optimization; the words that are close to the hyperplane in our models are likely slightly different from one another. For the most part, both of our programs produced words that were GS, with a few exceptions. Ultimately, the GS and GN word lists were inputs to the debiasing algorithm, which treats GS versus GN words differently.

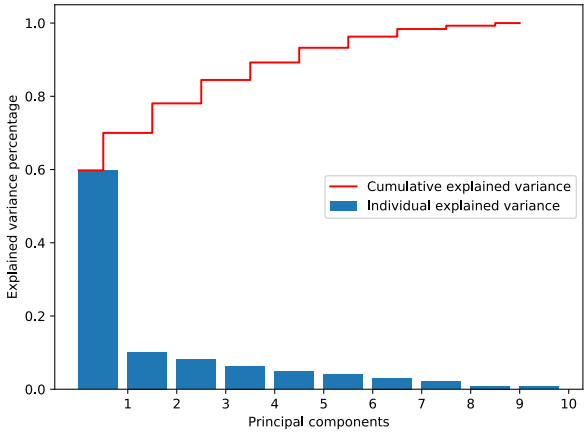

Figure 1: The percentage of variance explained in the PCA of these vector differences (each difference normalized to be a unit vector). The top component explains significantly more variance all of the others.

3.2 Identifying gender subspace/direction

If we want to remove a part of each vector that corresponds to gender, we must identify where gender is represented in a 300 dimension vector. In order to be able to determine which parts of the vector corresponds to gender, we must find the gender subspace or direction. In essence, this means finding the vector(s) that represents the expressions of gender in our word embedding.

To do this, we first conduct principal component analysis(PCA). We create a set of 10 pairs of words that define gender including $D_1 = \{man, woman\}$ or $D_2 = \{he, she\}$ and find the average between each of the pairs defined as (in the case of set D_1):

$$\vec{d_1} := \frac{\overrightarrow{man} + \overrightarrow{woman}}{2}$$

Then, we aggregate the differences into a matrix. When we then do principal component analysis on this matrix, we find the direction(s) of most variance within this matrix. In other words, since the matrix shows vectors that represent all of the ways gender is expressed across 10 sets of word pairs, by finding the direction(s) of most variance we capture all (or most) of the ways gender is expressed in our embedding. The result of this is a vector (or vectors, meaning a subspace) representing gender, $\vec{b}$. We can find how gendered a word is by its projection onto this subspace/direction which can be defined as, where $\vec{w}$ is any vector:

$$\vec{w}_b := (\vec{w} \cdot \vec{b})\vec{b}$$

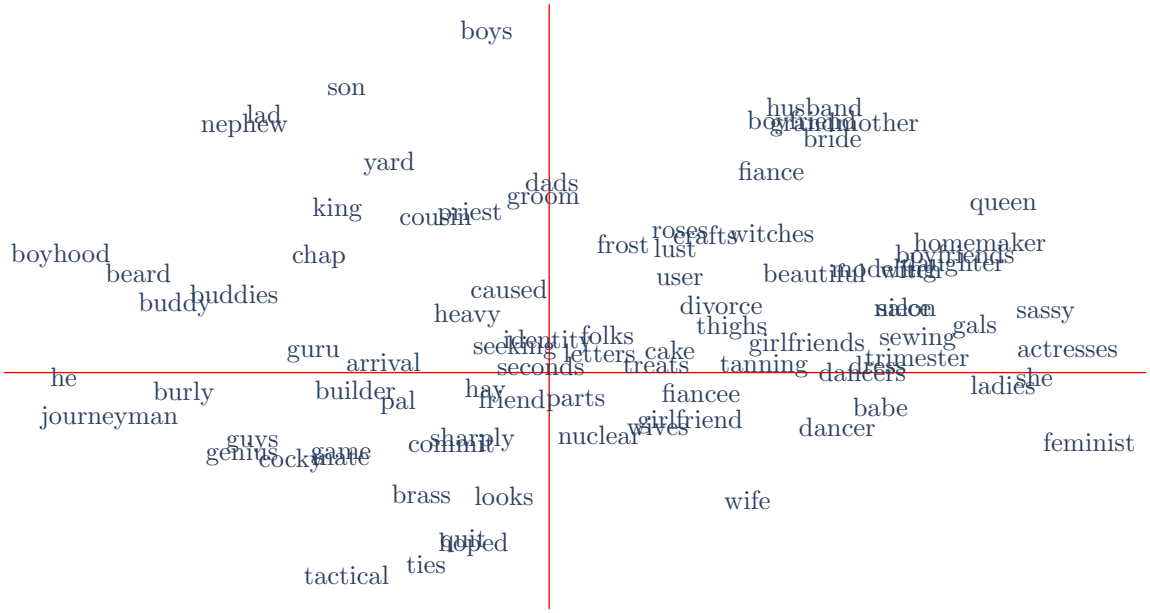

Figure 2: A scatterplot visualization of the word embedding pre-debiasing. Words are projected onto the gender direction, where words associated with 'he' are on the left of the y-axis, and words associated with 'she' are located on the right

show of this work is found in Figure 2 as even words that are gender neutral are located with bias in either the he or she directions.

3.3 Neutralize

After finding a gender direction, we can use it to neutralize the bias in words that should be gender neutral. Neutralize takes in a list of gender neutral words (i.e. the words in the embedding that were not part of the linear SVM's gender-specific word list output) that have some gender bias in the word embedding and attempts to remove it.

To do this, we simply redefine every word vector in our embedding $\vec{w}$ as:

$$\vec{w}_{neut.} := \frac{(\vec{w} - \vec{w}_b)}{||\vec{w} - \vec{w}_b||}$$

Intuitively, this removes the part of the vector that strictly represents gender, as defined by the projection $\vec{w}_b$.

An example of neutralize with the gender direction as the x-axis with the word nurse. Note that lady is not affected by neutralize as it is a gender-specific word:

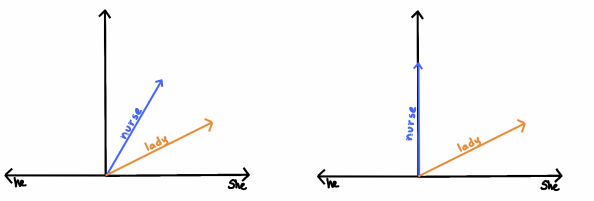

Figure 3: Rough sketch of pre-neutralize and post-neutralize 'nurse'.

3.4 Equalize

Equalize takes in a list of pairs of words that are equivalent to each other, similar to the definition sets used when finding the gender direction. Given sets of words $E = \{word_{male}, word_{female}\}$ we can then reassign each of the vectors in the set with the following formulas:

$$\mu := \sum_{\vec{w} \in E} \frac{\vec{w}}{2}$$

$$\nu := \mu - \mu_b$$

$$\vec{w}_{equal.} := \nu + \sqrt{1 - ||v||^2} \frac{\vec{w}_b - \vec{\mu}_b}{||\vec{w}_b - \vec{\mu}_b||}$$

We do this in order to make each part of the pair equidistant to the respective gender they represent, as well as every gender neutral word (that has just been neutralized). ν represents the non-gendered component of the pair of words. For equality pair $E = \{king, queen\}$, ν can be understood to represent a gender-neutral royal person. $\vec{w}_b - \vec{\mu}_b$ represents the gender component of each word minus it's average with its equivalent. This centers the gender components of the two words in the equality pair, such that the gender component for 'king' is just as similar to 'male' than the gender component in 'queen' to 'female'. The square root term scales the new gender component in order to maintain unit vector length. An illustration of this can be seen below:

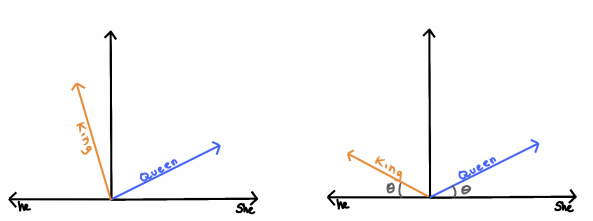

Figure 4: Rough sketch of equalize with pair 'king', 'queen'.

3.5 Generating Analogies

We generated analogies to identify gender stereotypes within the word embedding and to detect a change in the gender bias after running neutralize and equalize on the embedding. We created analogies of the form *she* is to x as *he* is to y. This format then produces pairs of words x,y that the word embedding believes are parallel to *she*,*he*, for example [queen, king] and [nurse, doctor].

The first step to find the x,y pairs is to identify the seed direction between a pair of gendered words (e.g., she/he, woman/man, etc). The two words in the gender pair are called seed words. The seed direction is the difference vector between the seed words. We will call this difference vector $\vec{v}$. Because the only difference between the words in the seed pairs is the gender to which they correspond, their difference vector represents gender. (Note, that we normalize all the words in the embedding, so each vector has a length of 1.)

Then, we calculate the difference vectors between all the words in the embedding. More specifically, we take one word in the embedding x and subtract every word in the embedding from it. We keep track of the difference vector between x and the other word y that is most similar to the difference vector between *she* and *he*. We will call this vector $\vec{u}$. We calculate the similarity by finding which $\vec{u}$ is most parallel with $\vec{v}$. We do this for every word in the embedding.The equation below summarizes this process.

$$S_{(a,b)}(x, y) = \begin{cases} cos(\vec{a} - \vec{b}, \vec{x} - \vec{y}) & \text{if } ||\vec{x} - \vec{y}|| \leq \delta \\ 0 & \text{otherwise} \end{cases}$$

By generating analogies, we rely on the word embedding to create relationships between words. Thus,

with an analogy like *she* is to *queen* as *he* is to *king*, we know that the word embedding sees the relationship between *queen* and *king* to be the same (or close to the same as) *she* and *he*. When the word embedding produces *she* is to *nurse* as *he* is to *physician*, we can identify a bias within the word embedding. Inherently, *nurse* is not more female than male, and *physician* is not more male than female. However, according to the word embedding, *nurse* is more female and *physician* is more male. Thus, the word embedding contains a gender bias.

4 Results

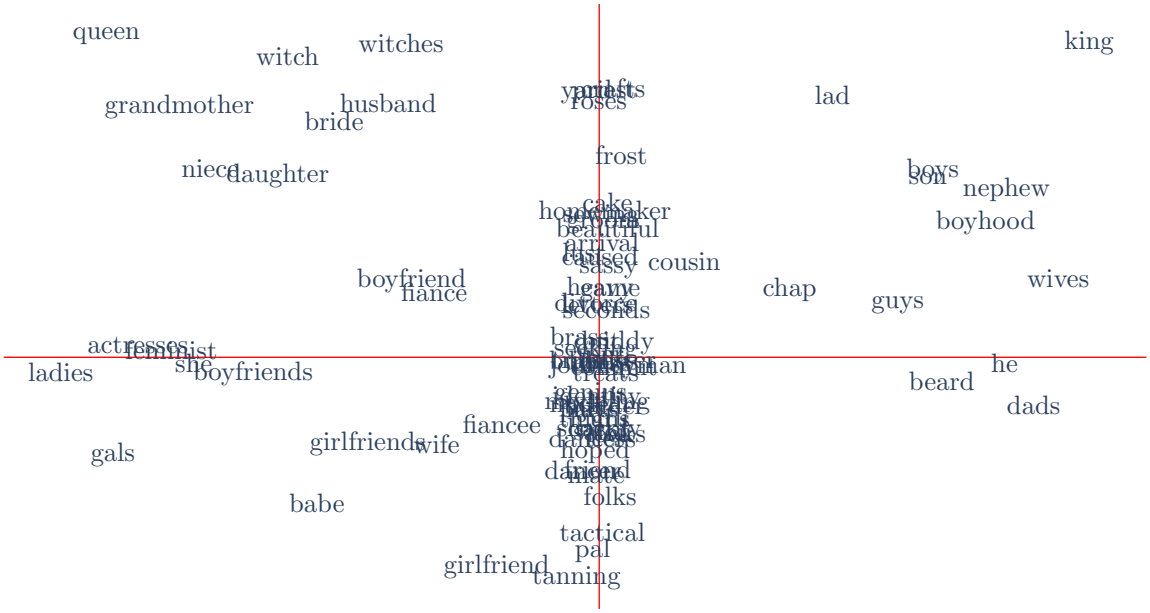

Figure 5: A debiased visualization of the word embedding. Recognize that gender neutral words are 'neutralized' near the middle of the plot, and gender specific words are equalized, being the same distance from the y axis.

Currently, our scatterplots are projected onto the gender subspace, found through performing PCA. This is different from OP's Figure 7, as Bolukbasi et al. represent gender neutrality by using their Linear SVM's threshold output to decide gender specificity or gender neutrality of each word. Visually, this difference would place those words identified as gender specific below the x axis, while gender neutral words would be placed above the x axis.

A visualization of our debiaising model can be found in Figure 5 as a debiased scatterplot representation of the word embedding. Here we can highlight those expected outcomes found in the methods section of our paper. Post debiasing, those gender neutral words with bias have been neutralized and concentrated to the center of the gender subspace, removing their association with male or female. Those gender specific words are also left with their gendered associations, but they are also equidistant to its gendered pair, making words like groom no more masculine than bride is feminine. (Man:groom :: Woman:bride)

To determine the efficacy of our debiasing model, we generated analogies on the word embedding before and after running the model. The analogies generated from the pre-debiased word embedding matched those of the paper. Thus, using the same results from the crowd-sourced evaluations, we know that 19% of the 150 highest scoring analogies demonstrated a gender stereotype.

She	He	Score	Rank	Debiased He	Debiased Score	Debiased Rank
her	his	0.6815	1	his	0.70465237	9
herself	himself	0.6502	2	himself	0.6557	13
heroine	hero	0.5419	3	villain	0.33965504	127
businesswoman	businessman	0.4941	4	businessman	0.7056	8
petite	lanky	0.4782	5	soft spoken	0.018593162	13198
queens	kings	0.3673	47	kings	0.7323	4
registered nurse	physician	0.3210	79	nursing	0.0283	7531
sewing	carpentry	0.2985	100	yarn	0.0352	4450
pediatrician	orthopedic surgeon	0.2679	136	doctors	0.0754	528
midwife	doctor	0.2588	154	physician	0.0214	11461
homemaker	schoolteacher	0.1616	774	schoolteacher	0.0147	15577

Table 1: The analogies generated from the word embedding. Score is the dot product of the difference vectors of $\vec{she} - \vec{he}$ and the *x,y* words of the analogy (e.g., $\vec{homemaker} - \vec{schoolteacher}$). Rank shows how parallel each analogy is to $\vec{she} - \vec{he}$ compared to the other 26,423 analogies. The last three columns show the results after debiasing the word embedding.

After generating the analogies from the debiased word embedding, we make three observations. First, the gender-specific words maintained their relationship after the debias. This is demonstrated in the [businesswoman, businessman] pair. In fact, the relationships between gendered pairs are strengthened after the debias. This is shown in the increased score between [businesswoman, businessman] and [queens, kings] increases after the debias. Secondly, analogies that demonstrate a gender stereotypes have a lower score after the debias. The gender neutral pair [homemaker, schoolteacher] rank drops from 774 to 15577. This lower score means that the angle between 'homemaker' and 'schoolteacher' become more different from the from the angle between [she, he]. Thus, while 'homemaker' was the highest scoring word to complete the analogy she is to homemaker as he is to schoolteacher, it was not the ideal choice. Thirdly, when there existed a better fitting, less stereotypical word in the embedding, the debias word embedding chose that word to complete the analogy. This is demonstrated in [sewing, carpentry] and [registered nurse, physician] before debias to [sewing, yarn] and [registered nurse, nursing] after debias.

5 Discussion

In replicating the experiments done in "Man is to Computer Programmer as Woman is to Homemaker?", we aimed to remove the gender bias present within word embeddings. We used analogies to evaluate the success of our methods. The analogies produced before debiasing contain gender stereotypes. For example, the word *physician* is gender-neutral. However, in the word embedding, *physician* is more male than female. This is demonstrated by the analogy *she* is to *registered nurse* as *he* is to *physician*. After the debias, the analogy becomes *she* is to *registered nurse* as he is to *nursing*. Thus, we can conclude that the debias algorithm neutralized the gender direction of gender-neutral words. Analogies of gender-specific words remained the same after debias because we did not neutralize these words. A contestation to one of the methods that was used to quantify bias would be the OP's use of Amazon Mechanical Turks to judge stereotypical and appropriate analogies found in of the paper. They were limited to only 10 user inputs and analogies were stereotypical if more than half of the surveyors deemed them. Classifying some analogies as containing a gender bias requires nuance, and future work could include being explicit about what it means for an analogy to be stereotypical. Additionally, a larger word embedding could allow for more appropriate and unbiased analogies. A subset of about 26,000 words limits the the options to complete the analogy *she* is to x as *he* is to y. Potentially, having more options for y could allow less stereotypical analogies.

While having word embeddings that accurately reflect society may be useful in some applications, the default condition of word embeddings should be debiased. By having to intentionally opt in to biased embeddings, we avoid inadvertently amplifying gender stereotypes. However, gender bias is not the only bias that exists in word embeddings. Next steps could include removing religious or racial bias within embeddings.

Acknowledgements

Thank you to Dr. Anna Rafferty for all her support this term. Also, we appreciate our classmates for providing thoughtful feedback throughout the term. Financial support provided by the Carleton College Computer Science Department.

References

Řehůřek, R., Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora [Conference paper]. 45–50. http://is.muni.cz/publication/884893/en

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

T. Bolukbasi, K. Chang, J. Zou, V. Saligrama, and A. Kalai, "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings," arXiv, vol. 1, no. 1607.06520, July, 2016.

T. Mikolov. (2013). Word2vec. https://github.com/tmikolov/word2vec (accessed Sept. 20, 2022).