Debiasing Word Embeddings

Dalia Shanshal, Anika Tabassum
Data Science and Analytics, Ryerson University
dshansha@ryerson.ca, anika.tabassum@ryerson.ca

Abstract

Background. Word embeddings are vastly used in NLP tasks in various domains. However, the embeddings carry direct and indirect biases, and these biases are translated into real world applications. **Aim.** Our aim is to use a debiasing algorithm to mitigate this issue. **Methodology.** The method used consists of neutralizing and equalizing gender word pairs in such a way that any non-gendered/neutral word is at equal distance to gender word pairs such as *she-he*. **Results.** After plotting the extreme s*he-he* occupations, we find that all occupations are at equal distance from the *she* and *he* axis. We also find that gender specific words have moved closer to their respective gender axis (corresponding s*he* or *he* axis). **Conclusions.** The application of the suggested debiasing algorithm demonstrated promising results in terms of debiasing occupational stereotypes.

I – Introduction

Word embeddings are vector representations of words used in different machine learning applications such as sentiment analysis, semantic similarity, word clustering, document clustering and others [1]. There are various methods for training word embeddings. Some methods result in long and sparse vector representations such as tf-idf (term frequency-inverse document frequency) or ppmi (positive pointwise mutual information); other common methods result in short and dense vectors [1]. The latter include the word2vec family of models, such as skip-gram and CBOW, and GloVe embedding algorithms [1]. They usually have pre-trained embeddings available online and trained on very large amount of data such as Google News articles, Wikipedia articles, web common crawl or twitter. The problem with the embeddings is that they capture the semantic relationships as presented in the training text. As such, ethnic, racial or gender biases present in the training text are reflected in the embeddings and therefore, their use in real-world applications can introduce bias in everyday practices. In fact, the author in [2] raises concerns over the racial and gender bias inherent in different algorithms, such as hiring algorithms that are at risk of being biased towards a particular gender or race for a given position.

As such, it is important to tackle this issue strategically. Since we cannot change the training text, we need to change the algorithm that is learning the word embeddings in such a way that we remove any stereotypes from the output.

In this project, we aim to perform hard gender debiasing on pre-trained GloVe [3] embeddings. For this project, we have chosen the 50-dimensional version of GloVe, which is based on Wikipedia 2014 and Gigaword5 and has 400,000 words.

II - Related work and preliminary

Our related work and preliminary review is concerned with: a) proving biases in word embeddings and b) the methodology and algorithm for debiasing word embeddings.

Proving Bias in Word Embeddings

To prove bias in word embeddings, researchers have identified the words that are most closely associated with a particular ethnic word or gender. For example, the authors in [4] use the Google News word2vec embedding, and list the top 10 adjectives associated with women from the embeddings dated from 1910, 1950 and 1990. In their paper, they find that not only do word embedding representations carry biases within them, but that the biases change over time, reflecting societal changes within them. They also prove the existence of ethnic biases in occupations present in word embeddings by listing the top 10 occupations most closely associated with each ethnic group: Hispanic, Asian and White.

A similar method is used to prove gender occupation biases in Bolukbasi's [5]. A list of 320 occupations is projected onto the *she-he* axis where the Google News word2vec embedding is used. The 10 extreme occupations for *she* and *he* are then listed, and these were shown to correlate with crowd judgment of stereotypical occupations.

Methodology and Algorithm for Debiasing Word Embeddings

In his paper [5], Bolukbasi follows by debiasing the Google News word2vec using a particular technique described in section III. The method relies on changing the embeddings of gender neutral words by removing their gender associations. The results of this method were promising. Parts of this paper was replicated in the following project [6]. We aim to apply the debiasing algorithm on GloVe word embeddings trained on Wikipedia articles, assess the result and compare with Bolukbasi's findings.

Other methods are proposed in debiasing algorithms such as [7] and [8], however, these are more computationally expensive.

In [7], the authors construct a word-to-word co-occurrence matrix, and apply a modified version of the GloVe embedding algorithm resulting in a Gender-Neutral variant of GloVe: GN-GloVe. They add two terms to the overall objective function which are J_D and J_E. The J_D objective function minimizes the distance between male-female words (words that are gendered by definition). The J_E objective function encourages the neutral words to be retained in the null space of the gender direction, where the gender direction is estimated by taking the mean of the differences between female words and their male equivalent in a predefined set.

In [8], authors use adversarial learning techniques to mitigate gender bias in prediction tasks. The task of the algorithm is to predict a word analogous to the input word, in the following format *man : woman :: he : ?*. They add an "adversarial discriminator network" so that the discriminator cannot perfectly learn the gender direction *g* and would therefore avoid outputting a biased analogous word. They train machine learning models with and without applying their method, and find that their method results in unbiased training of the model, as opposed to the original one.

III- Methodology

Proving Bias

We select 320 occupations and project them onto the *she* and *he* vectors. The occupations' list are read from the **professions.json** file [9]. First, all words are normalized in such a way that ||*w*||=1. We apply a scalar projection following the equation below [10], where b_1 is a scalar projection of b onto a, θ is the angle between a and b, and $\bar{a}$ is the unit vector in the direction of a:

$$b_1 = |b|cos\theta = \frac{b\,.\,a}{|a|} = b.\bar{a}$$

All our vectors are normalized to a unit vector, and thus our scalar projection is simply the dot product between the two word vectors. We list the top 15 most extreme *she* and *he* occupations in Table 1.

Gender Debiasing Algorithm

The gender debiasing algorithm takes four input parameters described below, and the output is the debiased embedding vectors. The four **input parameters** are**:**

1. The embedding vectors that we aim to debias. In our case these are the 50-dimensional Glove embedding vectors.
2. A list of gender specific words which should not be debiased. These are words like *man, woman, boy, girl* etc. In our case, we read the list from the **gender_specific_seed.json** file which contains 218 gender specific words.
3. A list of gender definitional word pairs (tuples) which is used to calculate the gender subspace. In other words, these are the words that will define the gender direction in the embedding space. Examples of these word pairs include *(woman, man), (girl, boy)* etc. In our case, we read these pairs from the **definitional_pairs.json** file, which has **10** such word pairs.
4. A list of equalize word pairs (tuples). The goal is that after debiasing, any word that is not gender specific should be at equal distance from both words in each of these pairs. Examples of these word pairs include *(woman, man), (girl, boy)* etc. In our case, we read these pairs from the **equalize_pairs.json** file, which has 52 such word pairs.

It is important to note the distinction among parameters 2, 3 and 4 above. The words in all 3 parameters are essentially gender specific. However, parameter 2 is a discrete list of words not to be debiased. Parameter 3 is list of *pair* of words for defining the gender subspace and parameter 4 is also a list of *pair* of words for equalizing purpose. Theoretically, parameters 2 and 3 could well be the same list. However, in our case parameter 3 is a superset of parameter 2.

Now that we know what the inputs and output are, let's dig into the algorithm itself. The algorithm at a high level has two major steps:

1. Identifying the gender subspace
2. Neutralizing and Equalizing the words

We should point out that we perform the algorithm on a normalized version of the embedding, so that all vectors in the embedding are unit vectors. We are primarily interested in the bias direction and the cosine similarities between vectors, and not the exact magnitudes. So normalization does not hurt. Let's look at the details of how the steps of the algorithm work.

Identifying the gender subspace

Mathematically, the gender subspace ***B*** is defined as a set of ***k*** orthogonal unit vectors $\{b_1, b_2,...,b_k\}$ where ***k*** is the number of components we want in the subspace. In our case, we take ***k***=10. Also, since we are using the 50-dimensional embedding vectors, the individual unit vectors in the subspace, namely $b_1, b_2,...,b_k$ will all be 50-dimensional vectors as well. To obtain the gender subspace, we follow the following steps:

1. Define an empty list **L**.
2. For each pair vector (***x, y***) in the gender definitional pairs, compute the center vector ***c*** as ***c = (x + y) / 2***. Calculate difference vectors $\boldsymbol{diff_x = c - x}$ and $\boldsymbol{diff_y = c - y}$. Append $\boldsymbol{diff_x}$ and $\boldsymbol{diff_y}$ to the list **L**.
3. Apply Principal Component Analysis (PCA) on the list **L** with 10 components. The set of 10 components $\{b_1, b_2,...,b_{10}\}$ is the gender subspace.

Neutralizing and Equalizing the words

Once we have the gender subspace, we are left with the task of neutralizing and equalizing words. This is done in a two-fold approach.

1. For each word, we calculate the projection of that word onto the gender subspace. If the word is not gender specific, we subtract the projection from that word, so that the resulting vector is effectively the projection on the orthogonal subspace of the gender subspace. The projection of a word vector ***w*** onto the gender subspace $\{b_1, b_2,...,b_{10}\}$ would be:

$$w_B = \sum_{j=1}^{10} (w.b_j) b_j$$

   If ***w*** is not gender specific, we update vector ***w*** as***:***

$$w = \frac{w - w_B}{||w - w_B||}$$

Note that a normalization is performed after the subtraction. For implementation convenience and performance, we first calculate the subtraction result and perform a normalization over all vectors of the embedding.

2. The final task is to make sure that any word that is not gender specific is at equal distance from both words in each of the equalizing pairs. For doing this, we perform the following steps for each pair vector ($\boldsymbol{x, y}$) in the equalizing pairs:

   We calculate the mean vector of the pair:

$$\mu = \frac{x + y}{2}$$

   We calculate the projection of this mean vector onto the gender subspace using the same subspace formula as before:

$$\mu_B = \sum_{j=1}^{10} (\mu . b_j) b_j$$

   We subtract the projection from the mean vector and calculate a vector $\boldsymbol{v}$:

$$v = \mu - \mu_B$$

   For both words $\boldsymbol{x}$ and $\boldsymbol{y}$, we update their corresponding embedding vector as follows:

$$x = v + \sqrt{1 - v^2} \frac{x - x_B}{||x - x_B||}$$
$$y = v + \sqrt{1 - v^2} \frac{y - y_B}{||y - y_B||}$$

Note that $\boldsymbol{x_B}$ and $\boldsymbol{y_B}$ have already been calculated in step 1 above. Now that the vectors have been updated one more time, we perform one final normalization of all the vectors.

VI- Results

After projecting occupations onto the *she-he* axis, using the debiased GloVe embedding, we find the following extreme occupations (Table 1):

Table 1 Extreme She-He Occupations

Extreme She Occupations	***Extreme He Occupations***
Actress	Coach
Ballerina	Caretaker
Stylist	Captain
Socialite	Manager
Waitress	Colonel
Maid	Marshal
Narrator	Midfielder
Housewife	Skipper
Homemaker	Commander
Receptionist	Archbishop
Housekeeper	Bishop
Hairdresser	Footballer
Nurse	Substitute
Businesswoman	Lieutenant
Dancer	Superintendent

We compare the s*he-he* extreme occupations found in the GloVe embeddings with the ones found in the word2vec [5] and find the following similarities:

- Extreme *She* occupations: homemaker, nurse, socialite, hairdresser, stylist, housekeeper
- Extreme *He* occupations: skipper, captain

It is important to note that in Bolukbasi's findings, the gender specific occupations, such as businesswoman, were omitted from the list.
From the extreme *she* occupations, we find 6 occupations that are gender specific. These are: actress, ballerina, waitress, maid, housewife and businesswoman. In other words, these occupations are by definition gendered, and we want to make sure that, after applying the debiasing algorithm, these words are not equidistant to gender pair words, such as *she-he*, *woman-man* etc. in order to maintain their gendered definitional property.

We plot the *she-he* extreme occupations (Figure 1) where we can see that the extreme *he* occupations are geometrically closer to the *he* axis than the *she* axis, and vice versa. The idea is to change the occupations word vectors in such a way that any non-gender specific occupation is equally close to the *he* axis as it is to the *she* axis.

After applying the debiasing algorithm, we plot the extreme occupations again (Figure 2) and find that the non-gender specific occupations have equalized and are now equidistant to the *he* and *she* axis. The gender specific occupations, on the other hand, have moved closer to the corresponding gender axis. For example, we see that the words *businesswoman, actress, housewife, maid, waitress and ballerina* moved closer to the *she* axis and further away from the *he* axis.

Figure 1: Occupation visualization pre-debiasing

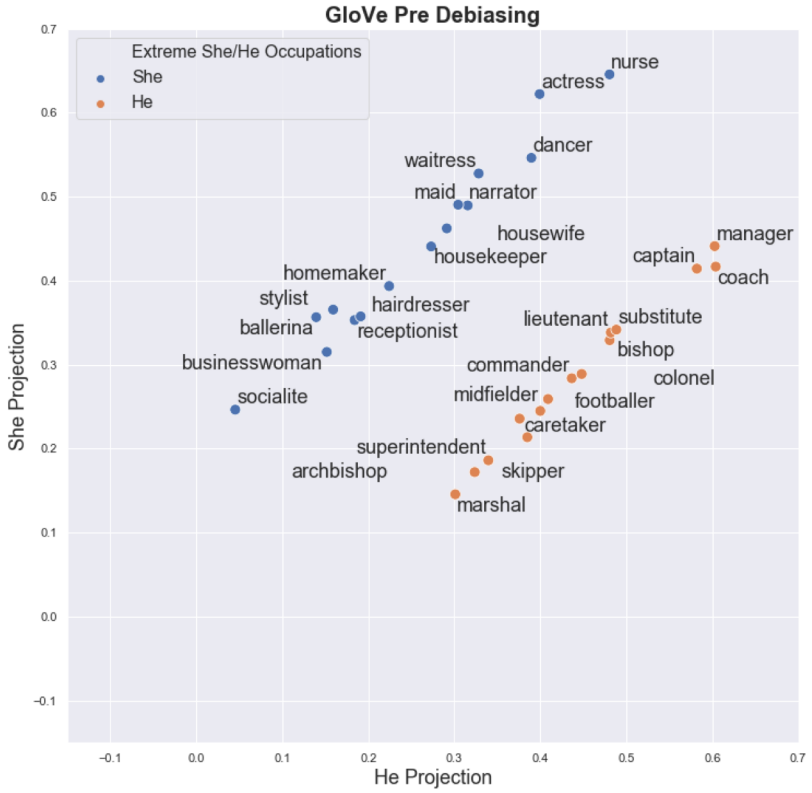
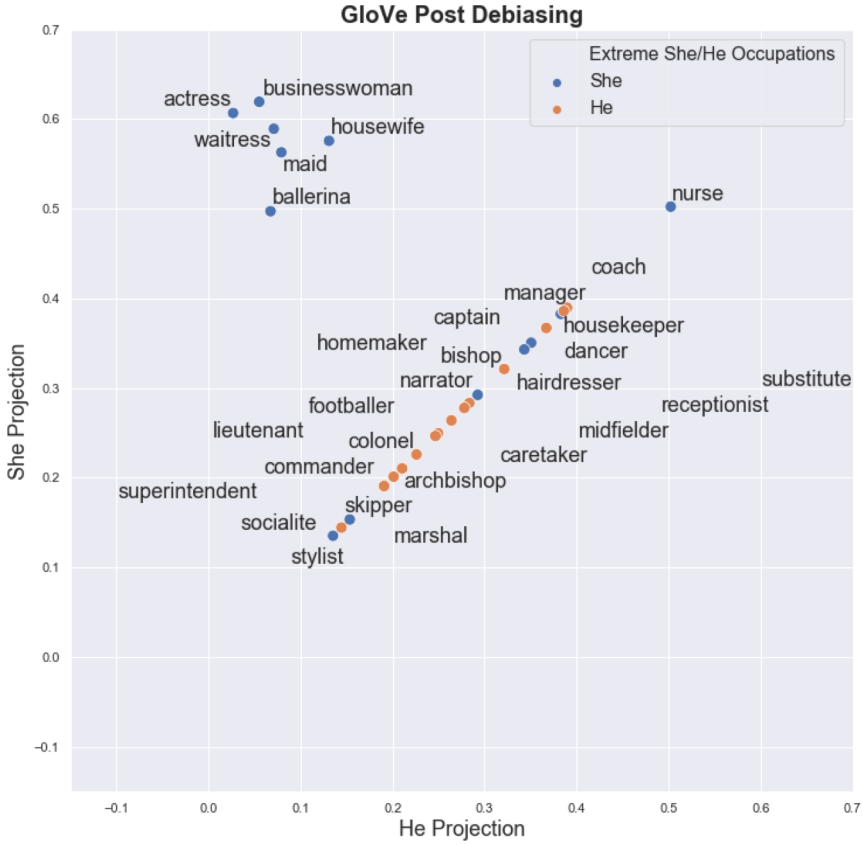


Figure 2: Occupation Visualization post-debiasing



An extended result of the debiasing algorithm is present in the accompanying notebook. We have demonstrated that after the equalizing step is done, in the resulting debiased embedding, each non-gender specific word in the 'profession' set is at equal distance from both words in each of the equalizing pairs, which is not the case in the original embedding.

V- Conclusion

Applying debiasing techniques is an important step to consider prior to using embeddings for real-world machine learning applications. The debiasing algorithm applied in this paper proved to work efficiently when it comes to removing occupational biases and stereotypes in the GloVe embedding trained on Wikipedia articles. We also found more common occupational biases corresponding to female (*she*) occupations between the GloVe embedding used in this paper and the word2vec trained on Google News. It would be interesting to take this study further and apply it on other embeddings, such as GloVe trained on twitter data or web common crawl. A comparison between the inherent biases within different embeddings, and the result of their debiasing would be another step that could be done for future research.

References

[1] Jurafsky, Dan, and James H. Martin. *Speech and language processing*. Vol. 3. London: Pearson, 2014.

[2] Garcia, Megan. "Racist in the machine: The disturbing implications of algorithmic bias." *World Policy Journal* 33.4 (2016): 111-117.

[3] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

[4] Garg, Nikhil, et al. "Word embeddings quantify 100 years of gender and ethnic stereotypes." *Proceedings of the National Academy of Sciences* 115.16 (2018): E3635-E3644.

[5] Bolukbasi, Tolga, et al. "Man is to computer programmer as woman is to homemaker? debiasing word embeddings." *Advances in neural information processing systems*. 2016.

[6] Vera, Ma Francesca Luisa C. "Exploring and Mitigating Gender Bias in GloVe Word Embeddings." 2018.

[7] Zhao, Jieyu, et al. "Learning gender-neutral word embeddings." *arXiv preprint arXiv:1809.01496* (2018).

[8] Zhang, Brian Hu, Blake Lemoine, and Margaret Mitchell. "Mitigating unwanted biases with adversarial learning." *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 2018.

[9] Bolukbasi, Tolga, Debiaswe: try to make word embeddings less sexist, (2018), GitHub repository, https://github.com/tolga-b/debiaswe

[10] Dot Products and Projections. https://math.oregonstate.edu/home/programs/undergrad/CalculusQuestStudyGuides/vcalc/dotprod/dotprod.html. [Date Accessed March 12, 2019]