Man is to Computer Programmer as Woman is to Homemaker?

Debiasing Word Embeddings

Angela Ellis, Aishwarya Varma, Aldo Polanco, & Darryl York III
Carleton College Computer Science Comps Fall 2022

Introduction

In this comps, we replicated the experiments done in "Man is to Computer Programmer as Woman is to Homemaker" by Bolukbasi et al. The goal of our project was to remove gender bias from word embeddings and compare our results with those of the paper to test robustness.

Word Embeddings

Word embeddings are vector representations of a set of words. In this paper, the words came from a collection of Google News articles. The word embedding we used came from Google's Word2vec and contains 3 million words each with 300 dimensions. The meaning of a word is given by its relation to other words in the text. A word embedding can be imagined as a space, where the words with similar meanings exist closer to each other. Some applications of word embeddings include parsing through consumer feedback, spam detection, and information retrieval (e.g., search engines).

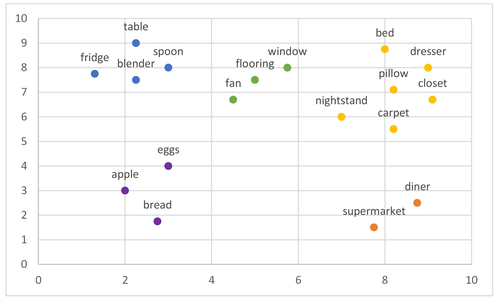


Figure 1. An example word embedding in 2D. Like words are grouped together.

Bias in Word Embeddings

Because word embeddings are built from texts written by humans, the bias present in the real world is also present in the embeddings. Thus, when the embeddings are used, for example, to return the most relevant pages after a query, the embedding determines which pages should be returned and in what order. If the embedding contains the bias that women are more likely to be homemakers, then a search for computer programmers is likely to return fewer women, all things equal.

Methods

Removing Bias

To remove the bias from the embedding, we first identified the gender subspace. Then, we equalized and neutralized the vectors. The equalize step ensures that gender-neutral words are equally close to each word in a gender-specific pair (e.g., *babysit* is as close to *grandmother* as it is to *grandfather*). The neutralize step adjusts gender-neutral words such that they are zero in the gender subspace (e.g., *nurse* is not more in the *she* direction than in the *he* direction). We classified each word in the embedding as gender-specific or gender-neutral using a support linear classifier.

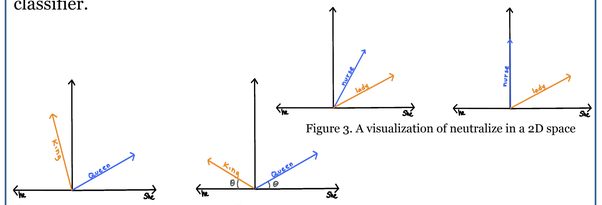


Figure 2. A visualization of equalize in a 2D space

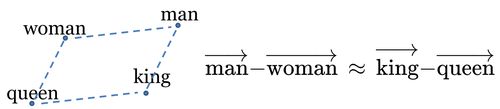
Figure 3. A visualization of neutralize in a 2D space

Detecting Bias

To 1) demonstrate that a bias exists and 2) to confirm that our debias method is effective, we created analogies using our word embedding. To identify gender stereotypes, we can generate analogies with the format *she* is to *x* as *he* is to *y*. We provide the algorithm with three words, for example, *she, he, nurse*, and using Equation 1, we return the *y* that maximizes *S*.

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

Equation 1. To find the *y* of the analogy, we first subtract the gendered pair of vectors, *a* and *b* (e.g., *she/he, man/woman*). Then, we subtract the vector *x* from every other vector in the embedding, y_0, y_1, ..., y_n. We chose the *y* whose difference with *x* is most close to the difference of *a* and *b*.



$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{king}} - \overrightarrow{\text{queen}}$$

Figure 4. In this 2D word embedding, the relationship between *woman* and *man* is the same as the relationship between *queen* and *king*. Thus, when *a* = *woman*, *b* = *man*, and *x* = *queen*, the algorithm will return *y* = *king*.

Results

Analogies Before Debias

'she' is to	as 'he' is to
nurse	surgeon
sewing	carpentry
queen	king
registered nurse	physician
housewife	shopkeeper
actress	actor
midwife	doctor
interior designer	architect
cosmetics	pharmaceuticals

Analogies After Debias

'she' is to	as 'he' is to
nurse	doctors
sewing	yarn
queen	king
registered nurse	nursing
housewife	schoolteacher
actress	actor
midwife	physician
interior designer	architectural
cosmetics	pharmaceuticals

Acknowledgements

Thank you to my group mates, Darryl, Aishwarya, and Aldo. We really appreciate all the support from our comps advisor, Dr. Anna Rafferty. Also, thank you to our classmates for providing thoughtful feedback throughout the term. Financial support provided by the Carleton College Computer Science Department.

Conclusion

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.

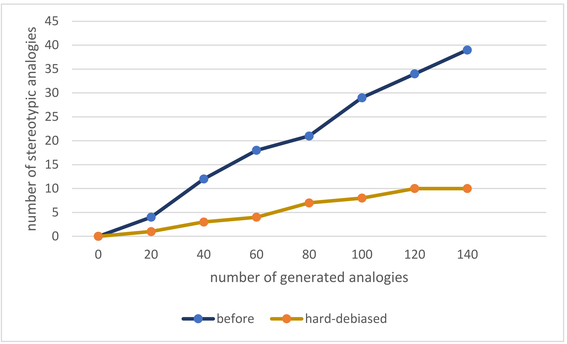


Figure 5. The number of stereotypical analogies after debiasing is less than before debiasing. Analogies were categorized as stereotypical via crowdsourcing data gathered from the Bolukbasi et al. paper.

Discussion

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.

References

1. 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.
2. Řehůřek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora [Conference paper]. 45–50. http://is.muni.cz/publication/884893/en
3. T. Mikolov. (2013). Word2vec. https://github.com/tmikolov/word2vec (accessed Sept. 20, 2022).