A bayesian method of cosine-based word2vec bias

Alicja Dobrzeniecka and Rafal Urbaniak

Contents

1 Cosine-based measures of bias

Modern Natural Language Processing (NLP) models are used to complete various tasks such as providing email filters, smart assistants, search results, language translations, text analytics and so on. All of them need as an input words represented by means of numbers which is accomplished with word embeddings. It seems that in the learning process these models can learn implicit biases that reflect harmful stereotypical thinking. One of the sources of bias in NLP can be located in the way the word embeddings are made. There is a considerable amount of literature available on the topic of bias detection and mitigation in NLP models.

One of the first measures in the discussion has been developed by Bolukbasi, Chang, Zou, Saligrama, & Kalai (2016). There, the gender bias of a word w is understood as its projection on the gender direction $\vec{w} \cdot (\overrightarrow{he} - \overrightarrow{she})$ (the gender direction is the top principal compontent of ten gender pair difference vectors). The underlying idea is that no bias is present if non-explicitly gendered words are in equal distance to both elements in all explicitly gender pairs. Given the (ideally) gender neutral words N and the gender direction g the direct gender bias is defined as the average distance of the words in N from g (c is a parameter determining how strict we want to be):

$$\mathsf{directBias_c}(\mathsf{N}, \mathsf{g}) = \frac{\sum_{w \in N} |\mathsf{cos}(\vec{w}, g)|^c}{|N|} \tag{1}$$

The use of projections has been ciriticized for instance by Gonen & Goldberg (2019), who point out that while gender-direction might be an indicator of bias, it is only one possible manifestation of it, and reducing a projection of words might be insufficient. For instance, "math" and "delicate" might be in equal distance to both explicitly gendered words while being closer to quite different stereotypical attribute words. Further, the authors point out that most word pairs preserve similarity under debiasing meant to minimize projection-based bias.[1]

[1] Bolukbasi, Chang, Zou, Saligrama, & Kalai (2016) use also another method which involves analogies and their evaluations by human users on Mechanical Turk. It is discussed and criticized in (Nissim, Noord, & Goot, 2020).

To measure bias in word embeddings, Caliskan, Bryson, & Narayanan (2017) proposed the Word Embedding Association Test (WEAT). The idea is that the measure of biases between two sets of target words, X and Y, (we call them protected words) should be quantified in terms of the cosine similarity between the protected words and attribute words coming from two sets of stereotype attribute words, A and B (we'll call them attributes). For instance, X might be a set of male names, Y a set of female names, A might contain stereotypically male-related career words, and B stereotypically female-related family words. WEAT is a modification of the Implicit Association Test (IAT) (Nosek, Banaji, & Greenwald, 2002) used in psychology and uses almost the same word sets, allowing for a *prima facie* sensible comparison with bias in humans. If the person's attitude towards given pair of concept is to be interpreted as neutral, there should be no noticeable task completion time difference, and the final value from the formula should be around 0. Let f be a similarity measure (usually, cosine similarity). The association difference for a term t is:

$$s(t,A,B) = \frac{\sum_{a\in A} f(t,a)}{|A|} - \frac{\sum_{b\in B} f(t,b)}{|B|} \tag{2}$$

then, the association difference between A a B is:

$$s(X,Y,A,B) = \sum_{x\in X} s(x,A,B) - \sum_{y\in Y} s(y,A,B) \tag{3}$$

$s(X,Y,A,B)$ is the statistic used in the signifcance test, and the p-value obtained by bootstrapping: it is the frequency of $s(X_i,Y_i,A,B) > s(X,Y,A,B)$ for all equally sized partitions X_i, Y_i of $X \cup Y$. The effect size is computed by normalizing the difference in means as follows:

$$bias(A,B) = \frac{\mu(\{s(x,A,B)\}_{x\in X}) - \mu(\{s(y,A,B)\}_{y\in Y})}{\sigma(\{s(w,A,B)\}_{w\in X\cup Y})} \tag{4}$$

Caliskan, Bryson, & Narayanan (2017) show that significant biases—thus measured— similar to the ones discovered by IAT can be discovered in word embeddings. Lauscher & Glavas (2019) extended the methodology to a multilingual and cross-lingual setting, arguing that using Euclidean distance instead of similarity does not make much difference, while the bias effects vary greatly across embedding models (interestingly, with social media-text trained embeddings being less biased than those based on Wikipedia).

A similar methodology is employed by Garg, Schiebinger, Jurafsky, & Zou (2018), who employ word embeddings trained on corpora from different decades to study the shifts in various biases. For instance, to compute the occupational embeddings bias for women the authors first compute the average vector of vector emeddings of words that represent women (e.g. "she," "female"), then calculate the Euclidean distance between this mean vector and words for occupations. Then they take the mean of these distances and subtract from it the analogously obtained mean for the average vector of vector embeddings of words that represent men. Formally they take the relative norm distance between X and Y to be:

$$\text{relative norm distance} = \sum_{v_m\in M} ||v_m - v_X||_2 - |v_m - v_Y||_2 \tag{5}$$

where the norm used is Euclidean, and v_X and x_Y are average vectors for sets X and Y respectively.

(**Manzini2019blackToCriminal?**) modify WEAT to a multi-class setting, introducing Mean Average Cosine similarity as a measure of bias (in fact, in the paper they report distances rather than similarities). Let $T = \{t_1,\ldots,t_k\}$ be a class of protected word embeddings, and let each $A_j \in A$ be a set of attributes stereotypically associated with a protected word). Then:

$$S(t_i,A_j) = \frac{1}{|A_j|} \sum_{a\in A_j} \cos(t,a) \tag{6}$$

$$MAC(T,A) = \frac{1}{|T|\,|A|} \sum_{t_i\in T} \sum_{A_j\in A} S(t_i,A_j) \tag{7}$$

That is, for each protected word \$T and each attribute class, they first take the mean for this protected word and all attributes in a given attribute class, and then take the mean of thus obtained means for all the protected words. The t-tests they employ are run on average cosines used to calculate MAC.

2 Some methodological problems

3 The problem with means of means

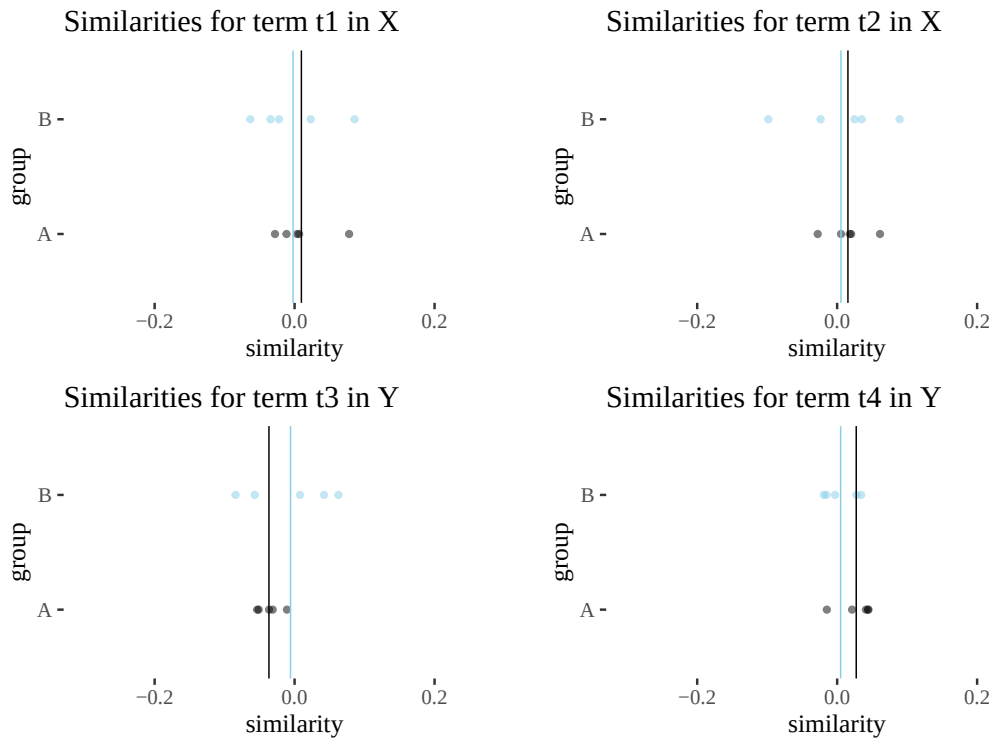
The measures described all calculate means of means and their authors run statistical tests on sets of means. This, however, is problematic for two related reasons. One, by pre-averaging data we throw away information about sample sizes. For the former point, think about proportions: 10 out of 20 and 2 out of 4 give the same mean, but you would obtain more information by making the former observation rather than by making the latter. Two, when we pre-average, we remove variation, and so pre-averaging tend to manufacture false confidence. We will have more to say about whether this happens in the case of applications of MAC, for now let's go over a simple example to make the conceptual point clear.

To illustrate let's employ the formulas used by Caliskan, Bryson, & Narayanan (2017) in a simple example. Conceptually, all such tests come up with rather short lists of protected words and rather short lists of stereotypical attributes. Clearly, these are not complete list. So let's treat them as samples from richer pools of stereotypical predicates and let's take the uncertainty and variation involved seriously.

Consider a simple situation in which there are two protected classes, $X = \{t_1, t_2\}$ and $Y = \{t_3, t_4\}$ and two five-element attribute sets A and B.

First, we play around with a scenario in which all the protected terms are on average equally unsimilar to both sets ($\mu = 0$) with standard deviation of .05. Let's randomly draw similarity scores and plot the results with group means plotted as vertical lines.

```
set.seed(123)
t1 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,0,0.05))
t2 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,0,0.05))
t3 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,0,0.05))
t4 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,0,0.05))
```



When you look at the datapoints, do you have the impression of a strong bias here? We wouldn't think so. But now let's run the calculations from (Caliskan, Bryson, & Narayanan, 2017).

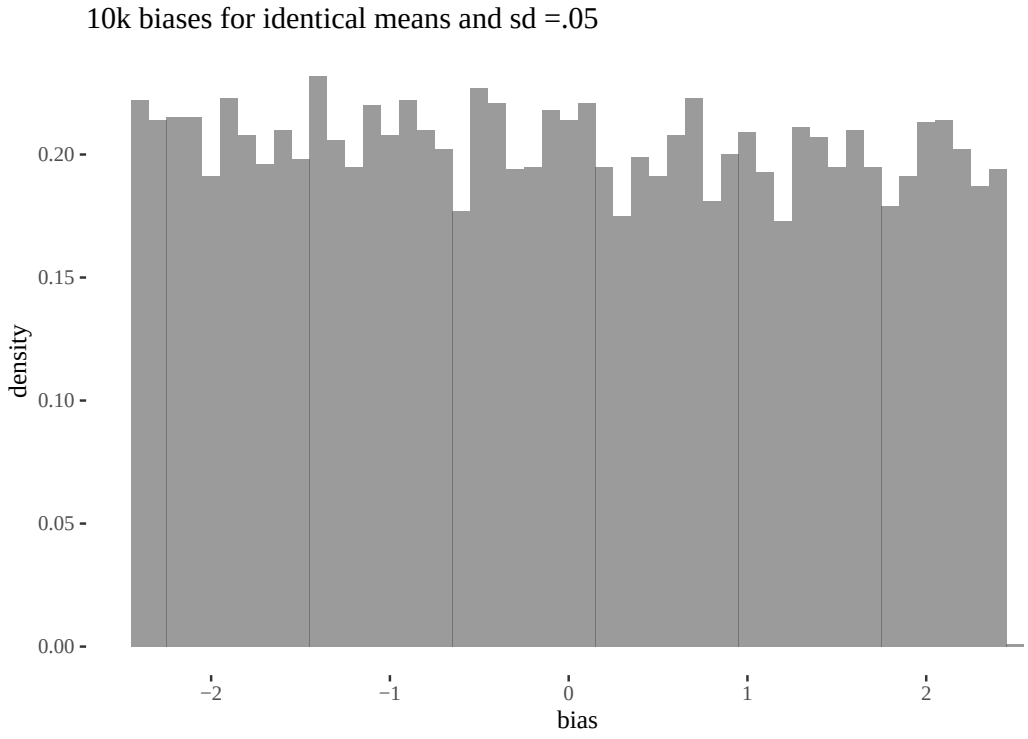
```
s <- function (table){ mean(table$A) - mean(table$B)}
factor <- sd(c(s(t1),s(t2),s(t3),s(t4)))
```

```
numerator <-  mean(s(t1),s(t2)) - mean(s(t3),s(t4))
print(list(factor = factor,numerator = numerator, bias = numerator / factor))
```

```
## $factor
## [1] 0.02342637
##
## $numerator
## [1] 0.04275325
##
## $bias
## [1] 1.825005
```

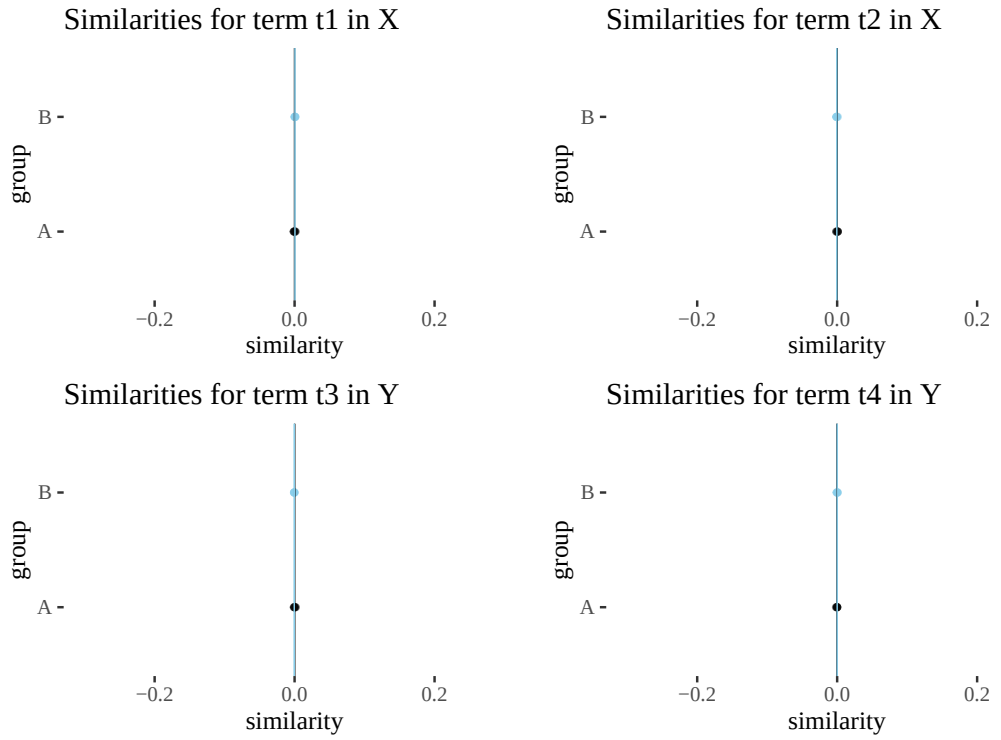
This should make us pause. We know these were points randomly drawn from distributions where there is no difference in means. Yet, the calculated effect size is 1.82, whereas the largest effect size reported by Caliskan, Bryson, & Narayanan (2017) is 1.81!
Interestingly, if we repeat the drawing 10000 times, each time calculating the bias, it turns out that with this variance and sample size, pretty much anything can happen.

```
biasesNull <- numeric(10000)
for(i in 1:10000){
t1 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,0,0.05))
t2 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,0,0.05))
t3 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,0,0.05))
t4 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,0,0.05))

factor <- sd(c(s(t1),s(t2),s(t3),s(t4)))
numerator <-  mean(s(t1),s(t2)) - mean(s(t3),s(t4))
biasesNull[i]  <- numerator / factor
}
ggplot()+geom_histogram(aes(x=biasesNull, y = ..density..), alpha = 0.6, bins=50)+
  theme_tufte()+labs(title="10k biases for identical means and sd =.05")+ xlab("bias")
```



Now, let's simulate a situation where the means are identical but the standard deviation is much smaller, .001.

```
set.seed(124)
t1v <- data.frame(A  = rnorm(5,0,0.001), B = rnorm(5,0,0.001))
t2v <- data.frame(A  = rnorm(5,0,0.001), B = rnorm(5,0,0.001))
t3v <- data.frame(A  = rnorm(5,0,0.001), B = rnorm(5,0,0.001))
t4v <- data.frame(A  = rnorm(5,0,0.001), B = rnorm(5,0,0.001))
```



When you look at the datapoints, do you have the impression of a strong bias here? We wouldn't think so. But now let's run the calculations from (Caliskan, Bryson, & Narayanan, 2017).

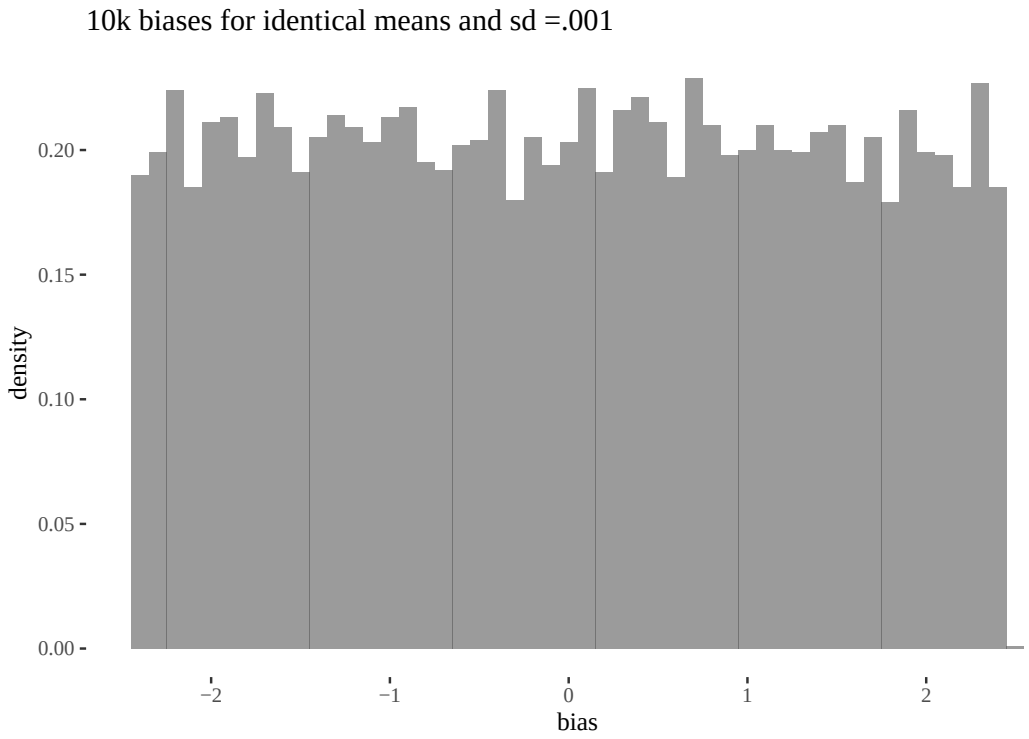
```
factorV <- sd(c(s(t1v),s(t2v),s(t3v),s(t4v)))
numeratorV <-  mean(s(t1v),s(t2v)) - mean(s(t3v),s(t4v))
print(list(factor = factorV,numerator = numeratorV, bias = numeratorV / factorV))
```

```
## $factor
## [1] 0.0006402666
##
## $numerator
## [1] -0.001237637
##
## $bias
## [1] -1.933003
```

While the numerator and the factors changed a lot, the bias actually increased. One reason bias increases is that once the standard deviation goes down, so does the factor used in the calculation of bias.
Again, to see whether this metric would provide us with meaningful information, let's simulate 10000 drawings.

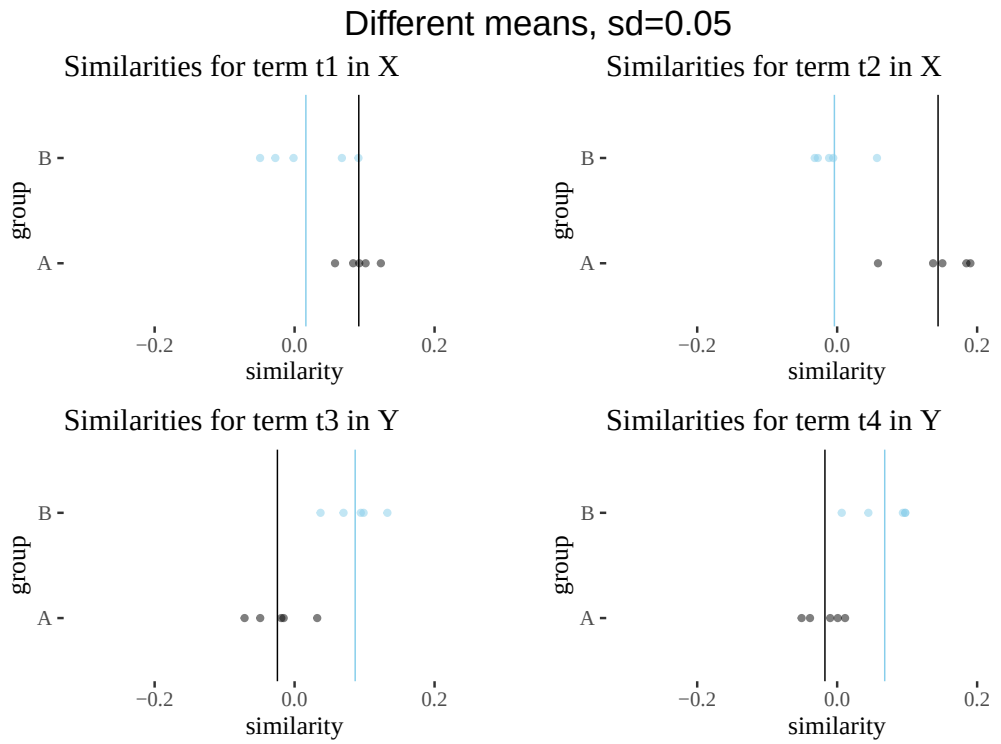
```
set.seed(124)

biasesLowVariance <- numeric(10000)
for(i in 1:10000){
t1v <- data.frame(A  = rnorm(5,0,0.001), B = rnorm(5,0,0.001))
t2v <- data.frame(A  = rnorm(5,0,0.001), B = rnorm(5,0,0.001))
t3v <- data.frame(A  = rnorm(5,0,0.001), B = rnorm(5,0,0.001))
t4v <- data.frame(A  = rnorm(5,0,0.001), B = rnorm(5,0,0.001))
```

```
factorV <- sd(c(s(t1v),s(t2v),s(t3v),s(t4v)))

numeratorV <-  mean(s(t1v),s(t2v)) - mean(s(t3v),s(t4v))

biasesLowVariance[i] <- numeratorV / factorV
}
ggplot()+geom_histogram(aes(x=biasesLowVariance, y = ..density..), alpha = 0.6, bins=50)+
  theme_tufte()+labs(title="10k biases for identical means and sd =.001")+ xlab("bias")
```



Again, not so informative. Now, let's try sampling from distributions where there in fact is a difference in means. Terms from X are on average .1 similar to A (and still 0 to B), while terms from Y are .1 similar to B (and 0 to A). The standard deviation is 0.05 in all the cases. There is a clear difference between X and Y and quick visual inspection should tell us so.

```
set.seed(766)
t1d2 <- data.frame(A  = rnorm(5,.1,0.05), B = rnorm(5,0,0.05))
t2d2 <- data.frame(A  = rnorm(5,.1,0.05), B = rnorm(5,0,0.05))
t3d2 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,.1,0.05))
t4d2 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,.1,0.05))
```



Is this clear difference mirrored in the calculations?

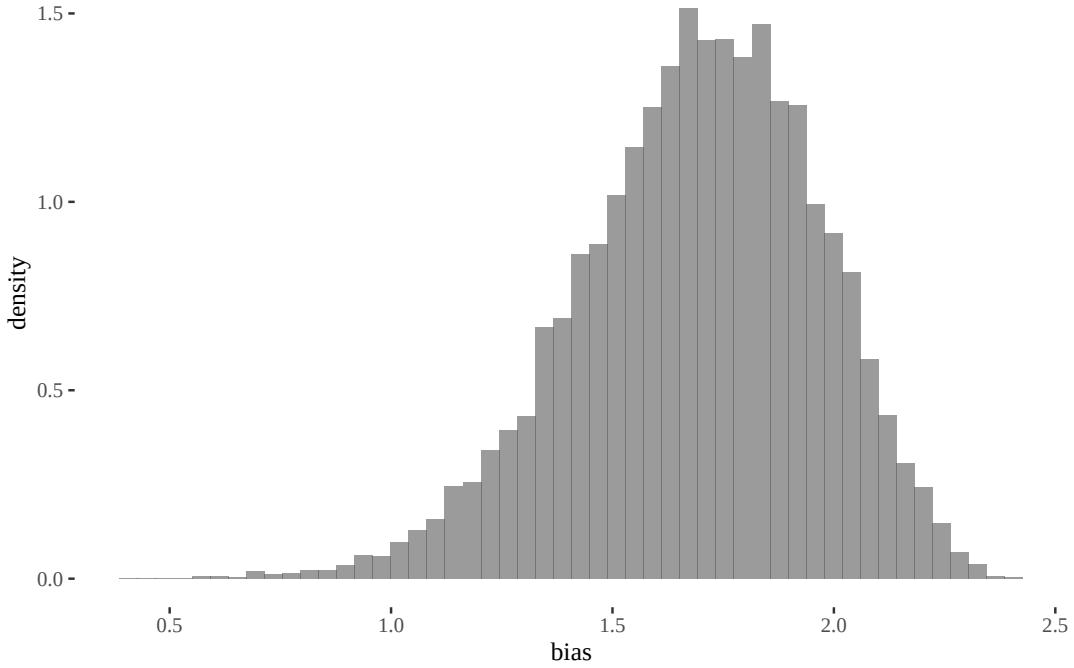
```
factorD2 <- sd(c(s(t1d2),s(t2d2),s(t3d2),s(t4d2)))
numeratorD2 <-  mean(s(t1d2),s(t2d2)) - mean(s(t3d2),s(t4d2))
biasD2 <- numeratorD2 / factorD2
biasD2
```

```
## [1] 1.490014
```

The absolute value of the effect size is smaller than in the null case with the same standard deviation. Let's simulate 10000 drawings:

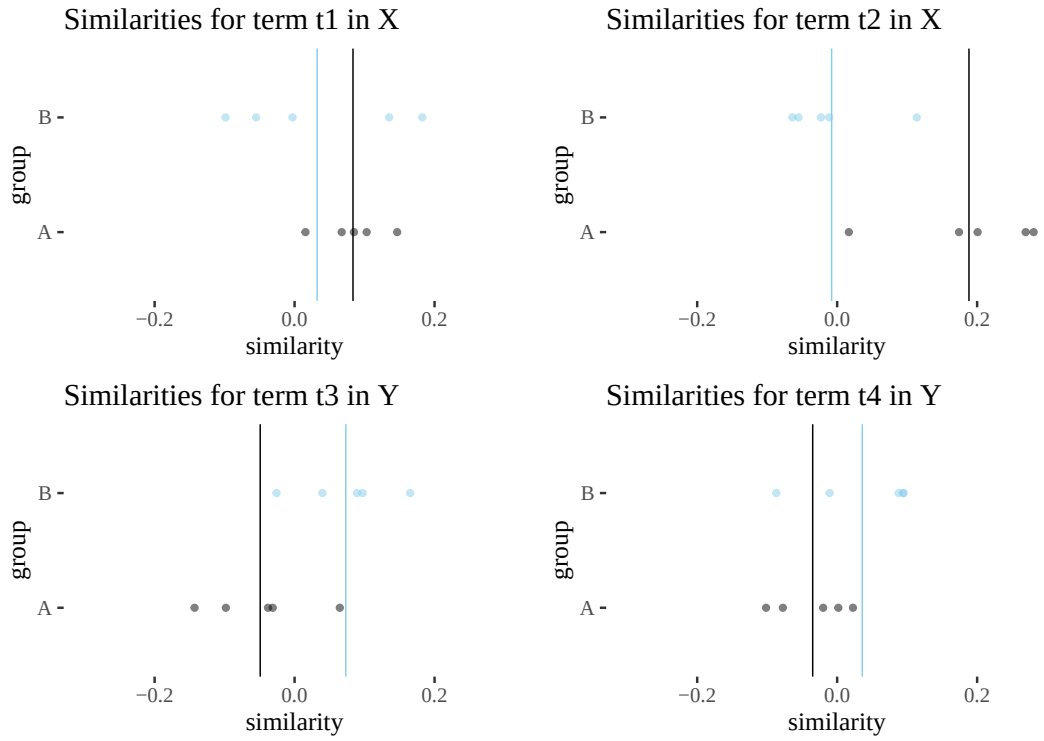
```
biasesD2 <- numeric(10000)
for(i in 1:10000){
t1d2 <- data.frame(A  = rnorm(5,.1,0.05), B = rnorm(5,0,0.05))
t2d2 <- data.frame(A  = rnorm(5,.1,0.05), B = rnorm(5,0,0.05))
t3d2 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,.1,0.05))
t4d2 <- data.frame(A  = rnorm(5,0,0.05), B = rnorm(5,.1,0.05))

factorD2 <- sd(c(s(t1d2),s(t2d2),s(t3d2),s(t4d2)))
numeratorD2 <-  mean(s(t1d2),s(t2d2)) - mean(s(t3d2),s(t4d2))
biasesD2[i] <- numeratorD2/factorD2
}

ggplot()+geom_histogram(aes(x=biasesD2, y = ..density..), alpha = 0.6, bins=50)+
  theme_tufte()+labs(title="10k biases for different means and sd =.05")+ xlab("bias")
```

10k biases for different means and sd =.05



Now suppose we keep the means the same but increase the standard deviation to .1.

```
set.seed(766)
t1d1 <- data.frame(A  = rnorm(5,.1,0.1), B = rnorm(5,0,0.1))
t2d1 <- data.frame(A  = rnorm(5,.1,0.1), B = rnorm(5,0,0.1))
t3d1 <- data.frame(A  = rnorm(5,0,0.1), B = rnorm(5,.1,0.1))
t4d1 <- data.frame(A  = rnorm(5,0,0.1), B = rnorm(5,.1,0.1))
```



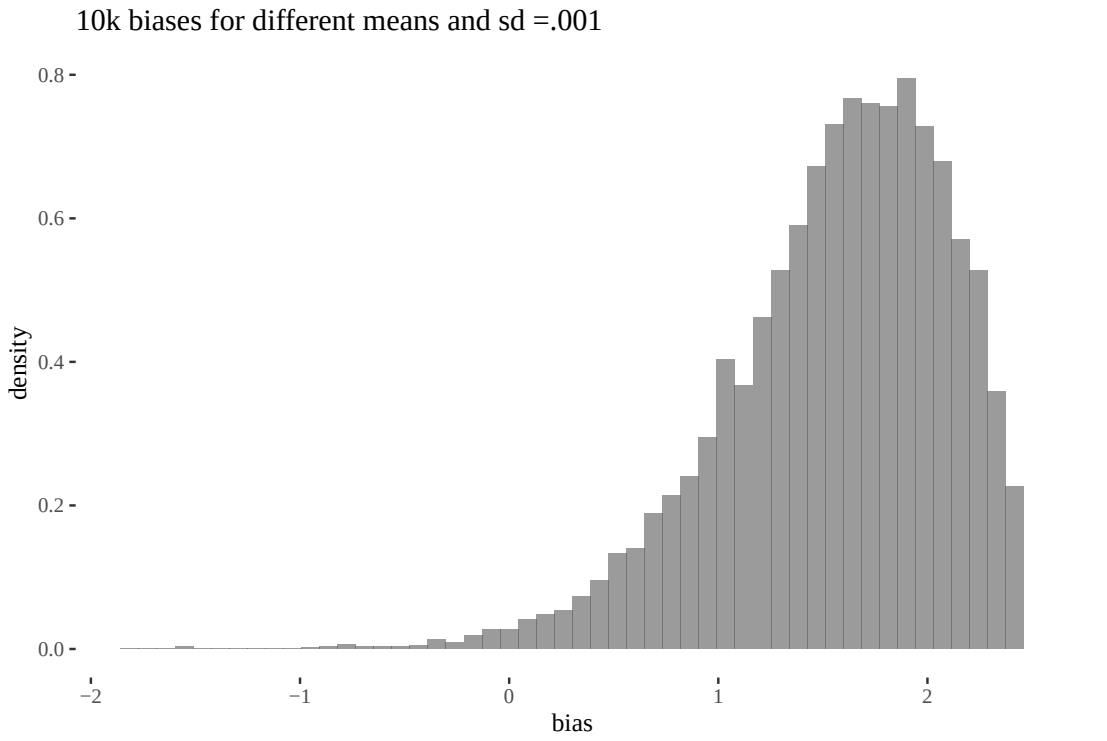
Is this clear difference mirrored in the calculations?

```
factorD1 <- sd(c(s(t1d1),s(t2d1),s(t3d1),s(t4d1)))
numeratorD1 <-  mean(s(t1d1),s(t2d1)) - mean(s(t3d1),s(t4d1))
biasD1 <- numeratorD1 / factorD1
biasD1
```

```
## [1] 1.223023
```

The absolute value of the effect size is smaller than in the previous case!

```
biasesD1 <- numeric(10000)

for(i in 1:10000){
t1d1 <- data.frame(A  = rnorm(5,.1,0.1), B = rnorm(5,0,0.1))
t2d1 <- data.frame(A  = rnorm(5,.1,0.1), B = rnorm(5,0,0.1))
t3d1 <- data.frame(A  = rnorm(5,0,0.1), B = rnorm(5,.1,0.1))
t4d1 <- data.frame(A  = rnorm(5,0,0.1), B = rnorm(5,.1,0.1))

factorD1 <- sd(c(s(t1d1),s(t2d1),s(t3d1),s(t4d1)))
numeratorD1 <-  mean(s(t1d1),s(t2d1)) - mean(s(t3d1),s(t4d1))
biasesD1[i] <- numeratorD1/factorD1
}

ggplot()+geom_histogram(aes(x=biasesD1, y = ..density..), alpha = 0.6, bins=50)+
  theme_tufte()+labs(title="10k biases for different means and sd =.001")+ xlab("bias")
```



This is is a bit better, but still quite some uncertainty is involved, far from what systematically low mean-based p-values reported in the papers might suggest. Of course, this is a bit of a caricature, as our word lists were short (four protected words and 10 attributes). But the word lists used in the actually papers are not much longer. In such a set-up the key observations are as follows:

- Seemingly high bias measures might arise even if the underlying processes actually have the same mean.
- Even if the mean remains the same, non-negligible changes can result from a shift in the standard deviation of the original process, and the change might go in the opposite direction than a visualisation of datapoints might suggest, because with the decrease of standard deviation, the factor decreases and the bias increases.

- Even if the underlying means are the same, but the variation is different, the bias metric in the long run could tend toward a different value:
- The lack of control group in the paper and our analysis indicates that without neutral baseline it is difficult to interpret the effectiveness of the metric.

```
mean(biasesD1); mean(biasesD2)
```

```
## [1] 1.563319
```

```
## [1] 1.690664
```

```
median(biasesD1); median(biasesD2)
```

```
## [1] 1.638136
```

```
## [1] 1.707511
```

and so the point estimations of bias are sensitive to factors other than the underlying process means.

- Even if there is a difference in means, the bias metric can be lower, and the uncertainty about it needs to be gauged.
- The uncertainty resulting from including the raw datapoint variance into considerations is more extensive than the one suggested by the low p-values obtained from taking means as datapoints.

```
quantile(biasesD2, probs = c(0.275,0.975))
```

```
##    27.5%    97.5%
## 1.539149 2.166488
```

```
quantile(biasesD1, probs = c(0.275,0.975))
```

```
##    27.5%    97.5%
## 1.300648 2.356815
```

4 Bayesian estimation

- reddit i google, ogolnie i z podzialem na protected words
- dokladniej gender, reszta na koniec w zalaczniku

5 Effects of debiasing

6 Discussion

Appendix
Word lists, including human and neutral predicates

References

Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., & Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. Retrieved from http://arxiv.org/abs/1607.06520

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, *356*(6334), 183–186. American Association for the Advancement of Science (AAAS). Retrieved from https://doi.org/10.1126/science.aal4230

Garg, N., Schiebinger, L., Jurafsky, D., & Zou, J. (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, *115*(16), E3635–E3644. Proceedings of the National Academy of Sciences. Retrieved from https://doi.org/10.1073/pnas.1720347115

Gonen, H., & Goldberg, Y. (2019). Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 609–614). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/N19-1061

Lauscher, A., & Glavas, G. (2019). Are we consistently biased? Multidimensional analysis of biases in distributional word vectors. *CoRR*, *abs/1904.11783*. Retrieved from http://arxiv.org/abs/1904.11783

Nissim, M., Noord, R. van, & Goot, R. van der. (2020). Fair is better than sensational: Man is to doctor as woman is to doctor. *Computational Linguistics*, *46*(2), 487–497. MIT Press - Journals. Retrieved from https://doi.org/10.1162/coli_a_00379

Nosek, B. A., Banaji, M. R., & Greenwald, A. G. (2002). Harvesting implicit group attitudes and beliefs from a demonstration web site. *Group Dynamics: Theory, Research, and Practice*, *6*(1), 101–115. American Psychological Association (APA). Retrieved from https://doi.org/10.1037/1089-2699.6.1.101