Conceptual and methodological problems with bias detection and avoidance in natural language processing

Alicja Dobrzeniecka

2021-06-16

Contents

Chapter 1

Introduction

Natural language processing (NLP) is a subfield of computer science that processes and analyzes language in text and speech with the use of modern programming methods. It has practical applications in everyday life as it concerns tasks such as email filters, smart assistants, search results, language translations, text analytics and so on. Models used to accomplish these tasks need a lot of data to learn from. This data originates from human activities and historical recordings such as texts, messages or speeches. It turns out that in the learning process these models can learn implicit biases that reflect harmful stereotypical thinking still present in modern societies. One can find methods that aim at identifying and measuring hidden biases and/or try to remove them by modifying the models. There are many different types of models in NLP depending on a task that they are supposed to solve. However, all of them need as an input words represented by means of numbers and this is accomplished with word embedding models. The models usually assign the values based on the context in which the words appear. It means that the input data can have enormous influence on the outcome. The biases seem to have their primary source in the way the words are assigned the numerical values.

There is considerable amount of literature available on the topic of bias detection and mitigation in NLP models. Bolukbasi, Chang, Zou, Saligrama, & Kalai (2016) focus on gender biases that may be observable while investigating the representation of job occupations and gender in terms of their assigned numerical values. The authors apply cosine similarity measurement to investigate the phenomenon where (the vectors corresponding to) words related to jobs that are stereotypically associated with a given gender are in fact in the model situated closer to this gender. They also use analogy tasks to evaluate if the bias is present in the word embedding model. They check analogies by comparing pairs of word vectors, for example they search for the word complementing the puzzle: man is to doctor as woman is to ...? First they subtract the word "man" from the word "woman" and then they search for the ranked list of other words pairs that have similar vector difference. They also include in the formula a threshold to ensure that the resulting pairs could not be randomly picked.

However, as Nissim, Noord, & Goot (2019) points out, there are some limitations to this approach. In practice, most of analogies implementations do not return any input words. This means that it does not make sense to expect the algorithm to return the same profession for both woman and man. Therefore, this method seems to be limited in terms of bias detection. Other problems are related for example to the choice of pairs and words that are used to detect the presence of discrimination as it is often subjective and without proper justification. Additionally the choice of the parameter used in Bolukbasi, Chang, Zou, Saligrama, & Kalai (2016) formula to ensure that word pairs are not picked by random, is also not justified and changing it drastically influences the results.

Islam, Bryson, & Narayanan (2016) touch upon the topic of biases regarding race and gender. They apply knowledge from well-known psychological studies such as Implicit Association Test to research the relation between human stereotypical thinking and model learnt biases to discover close relationship between these two. For the evaluation they use Word Embedding Association Test (WEAT) and the Word Embedding Factual Association Test (WEFAT).

(**manzini2019black?**) propose a novel way of using cosine similarity to obtain information on assumed resemblance between words. They investigate an approach that enables them to measure the bias for a class (like gender, religion, race) and express the final result with a single metric.

Islam, Bryson, & Narayanan (2016) refer to Implicit Association Test (Greenwald et al., 1998) that measures the strength of associations between concepts or stereotypes by calculating the time of reaction for special tasks. Humans naturally exhibit some biases which do not always cause social concern. One can imagine the intuitive associations between for example insects and flowers, and the feelings of pleasantness or unpleasantness. In general, people would rather associate flowers with feeling pleasant than insects, and this preference in a sense is a bias or prejudice in some direction. However, this type of preference does not cause an uproar and is a rather morally neutral case. This example is rather used to show that the methodology used in IAT makes sense. Unfortunately, the further studies discover other biases and prejudices that directly influence the quality of other people's lives and therefore they should be taken care of.

One can find various definitions trying to capture what bias and fairness actually are. With the choice of the definition, implications into the real-life applications may change as well. Mehrabi, Morstatter, Saxena, Lerman, & Galstyan (2019) mark out that there exist different types of biases such as historical bias, representation bias or measurement bias (the list is long). This indicates how complex the issue of bias is. Without the proper understanding and awareness of the problem, people are prone to unconsciously sustain the bias.

Mehrabi, Morstatter, Saxena, Lerman, & Galstyan (2019) also distinguish different types of discrimination, some of them will be briefly described. By protected attributes we mean those qualities, traits or characteristics that one should not discriminate against. Direct discrimination occurs when protected attributes of individuals explicitly result in non-favorable outcomes toward them. Such discrimination takes place for example when one uses an information about someone's religion belief to evaluate whether they are competent for the work they apply for. In contrast in indirect discrimination individuals appear to be treated equally but anyway they end up being treated unjustly due to the hidden effects of biases towards their protected attributes. An example of such situation is when in a job advert there is a requirement of 10 years experience instead of the list of a specific type of experience and knowledge. This job advert can discriminate indirectly against young people who can have equivalent skills but not yet so much work experience. Systemic discrimination takes place when policies, customs or behaviors that result from certain culture or organizational structure lead to discrimination against some groups of people. Finally, statistical discrimination consists in to using average group statistics to judge person belonging to the group.

The topic of discrimination is entangled with another concept; fairness. It is essential to grasp some concepts of fairness to take them into consideration while designing implementation of some machine learning model. In Mehrabi, Morstatter, Saxena, Lerman, & Galstyan (2019) one may notice that depending on the context and application different definitions may be applied.

The most common methods in the literature focus on comparing the similarity between words from protected groups and those that are considered to be stereotypical or harmful in some way. One can find in this group methods such as euclidean distance or cosine similarity (which is equivalent to dot product if the vectors are normalized). There are also other ways to detect the effects of biases. One can investigate the model performance on certain tasks and validate if the model uses the gender or race information to complete the task.

The currently used methods (such as cosine similarity) make the similarity values often aggregated in a way that may lead to hasty conclusions. The averaging of values and the lack of uncertainty may lead to the incomplete picture of the situation.

In the paper we indicate how current methods used to detect biases in natural language models are limited, if we investigate them from the perspective of Bayesian analysis.

Our research enhances the current way in which the bias detection is performed to make sure that it is methodologically valid.

The key hypothesis is that greater understanding of data and bias implications can be achieved when Bayesian methods are applied to issue.

Chapter 2

Cosine similarity and bias detection

2.1 Word embeddings

To understand what cosine similarity measurement is, one first needs to grasp the concept of translating words to a computer-readable form. In the field of natural language processsing there are two main types of words representation — localist and distributed. One-hot encoding is an example of a method used to achieve a localist representation of words. Here each vector contains information only about a single data point, this is achieved by first mapping categorical values (words) to integers and then to each integers a binary vector is assigned which contains only 0s except for the index of the integer, which is assigned 1. An example of a localist representation is:

word	1	2	3	4	5
woman	1	0	0	0	0
man	0	1	0	0	0
girl	0	0	1	0	0
boy	0	0	0	1	0

word	1	2	3	4	5
monarch	0	0	0	0	1

In the example above it is clear that the length of the vectors increases with the number of words in a vocabulary. It is not a very computationally efficient representation. It has other flaws as well. For example, it is unable to capture the resemblance between words appearing in similar contexts.

In contrast to the localist representation, a distributed representation returns vectors that contain continuous values instead of discrete 1s and 0s. Word embeddings are a class of various techniques that allow one to represent words as distributed vectors. Such learned representations of text have certain properties. At least prima facie, they store similar (or at least co-occurring) words close to each other in a vector space. An example of distributed representation is:

word	1	2	3	4
woman	0.456	0.267	0.675	0.131
man	0.451	0.897	0.472	0.088
girl	0.604	0.262	0.414	0.706
boy	0.279	0.172	0.475	0.010
monarch	0.565	0.678	0.463	0.975

One of the advantages of using a distributed representation is that one is able to represent an enormous number of concepts with a smaller number of units. It is also possible to better capture similarities as words of similar meanings can have similar numeric vectors.

The numbers occurring in such representations are not random. They are learned in a process that uses a very shallow neural network. There are various types of techniques used for learning the vectors representations. One of the most straightforward ones is a skip-gram model. Given a word the model tries to predict its neighboring words from the sentence. The mathematics behind the process relies on the idea that the prediction concerns the conditional probability of the adjacent words. The algorithm tries to minimize the loss function, which penalizes the system for discrepancy with actual co-occurrence frequencies in the corpus. One can choose various parameters of the model, such as the window size that determines how many surrounding words the model should predict. After preparing such a fitted modeln=, one takes only the learned weights from a neural network, and uses them as vectors in a word embeddings representation.

Word embeddings have many applications in natural language processing. They are handy in document search and information retrieval. They also play their part in improving automatic translations. Well learned word representations may also contribute to the improvement of sentiment analysis or spam detection.

2.2 Cosine similarity and distance

Cosine similarity is often used as a method of finding out whether vector representations for two words suggest that they are similar or somehow connected. Cosine similarity is the cosine of the angle between two vectors: the result of dividing their inner product (dot product usually) by the product of their magnitudes.

It is worth mentioning one more point concerning cosine similarity. After the vectors are normalized to have length equal to 1, inner product itself (often dot product) is used to measure the similarity.

$$\mathsf{cosineSimilarity}(A,B) = \frac{A \cdot B}{||A||\,||B||} \tag{Sim}$$

Cosine similarity is considered a proper tool for this operation as its result has a clear connection to geometry and at least for a low number of dimensions may be easily interpreted. Using this scale, one can compare vector similarities in a fairly clear manner. When the vectors are aligned perpendicularly to each other, their similarity equals 0 (which is the same as the cosine of 90 degrees). This tells us that the similarity between the vectors is small. As the angle between vectors decreases, cosine similarity approaches one, which stands for the greatest similarity.

One of the limitations of this measure is that it informs us only about similarities between vectors in terms of their orientation. However, it is often argued that in comparing words in terms of this metric, the magnitude of vectors may be treated as irrelevant, as the most important information pertains to direction.

In what follows, it is important to distinguish between cosine similarity and cosine distance, defined as:

$$\begin{aligned}\mathsf{cosineDistance}(A,B) &= 1 - \mathsf{cosineSimilarity}(A,B) \qquad \text{(Sim)}\\ &= 1 - \frac{A \cdot B}{||A||\,||B||}\end{aligned}$$

The greater the similarity between two vectors, the smaller the distance between them. The cosine distance ranges between 0 and 2. If the vectors are in an opposite direction to each other, the cosine distance is 2. And if the vectors are extremely similar then the cosine distance is very close to 0.

2.3 Cosine distance in a one-class bias detection

Bolukbasi, Chang, Zou, Saligrama, & Kalai (2016) define similarity between words as the outcome inner product of their normalized vectors. They focus on examining what the geometry of word embedding is in regard to “he” and “she” words. In other words, whether the similarity between those concepts and other words reflects expected gender stereotypes. They test this hypothesis by investigating whether there is a connection between word embeddings representing certain professions and words referring to gender. They also evaluate whether automatically produced analogies between words reflect the stereotypes as well.

Here 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 gender 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 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):

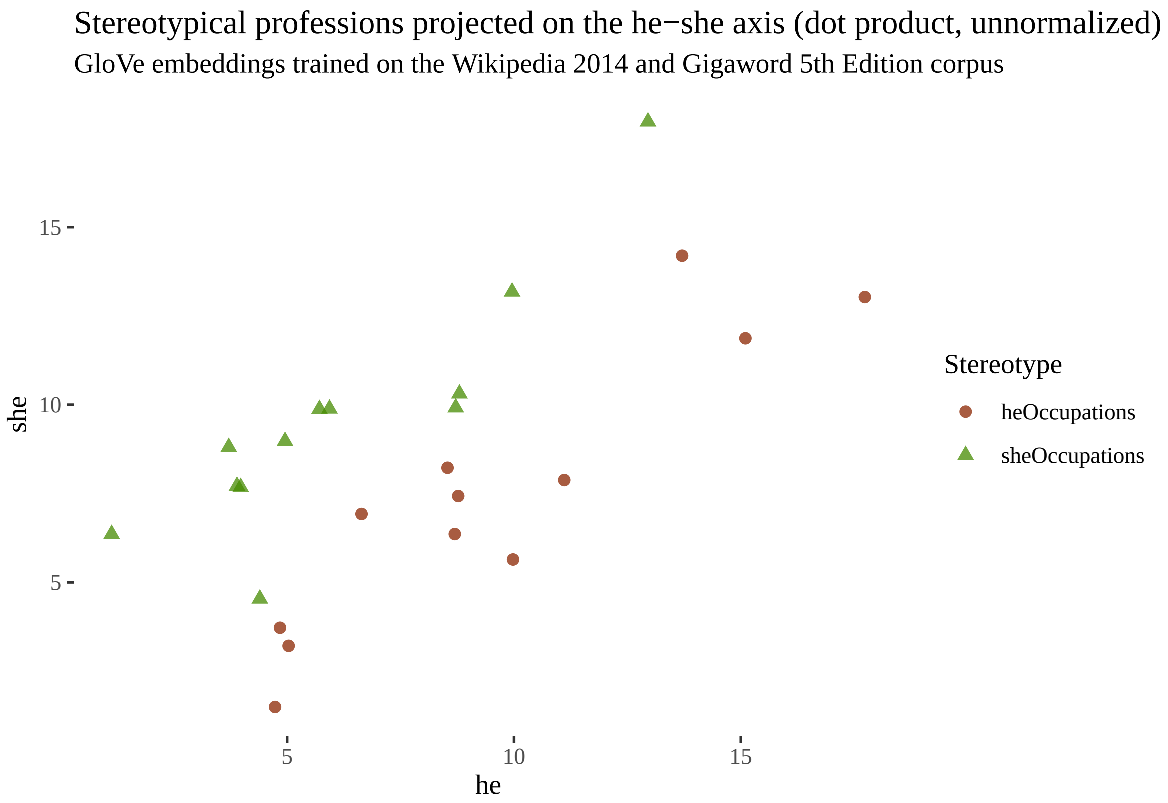
$$\mathsf{directBias_c(N, g)} = \frac{\sum_{w \in N} |\mathsf{cos}(\vec{w}, g)|^c}{|N|} \tag{2.1}$$

A very vivid way to follow their method of arguing that bias in word embeddings is real is to plot the values of inner product of chosen words. The plot below does not originate from the original paper (it is from https://www.kaggle.com/rtatman/gender-bias-in-word-embeddings) but similar visualization may be found there. Data used to create our plot is as follows.

Occupations associated with feminine: **”homemaker”**, **”nurse”**, **”receptionist”**, **”librarian”**, **”socialite”**, **”hairdresser”**, **”nanny”**, **”bookkeeper”**, **”stylist”**,

”housekeeper”, ”interior designer”, ”guidance counselor”

Occupations associated with masculine: **”maestro”, ”skipper”, ”protege”, ”philosopher”, ”captain”, ”architect”, ”financier”, ”warrior”, ”broadcaster”, ”magician”, ”fighter pilot”, ”boss”**


Stereotypical professions projected on the he−she axis (dot product, unnormalized)

GloVe embeddings trained on the Wikipedia 2014 and Gigaword 5th Edition corpus

The points in the plot above result from the calculation of the inner product of a chosen vector for a profession word and a vector for a gender word (she or he). Inner product of two vectors expresses similarity between words. This assumption originates from the geometry and properties of a vector space.

2.4 Cosine distance in a multi-class bias detection

Manzini, Lim, Tsvetkov, & Black (2019) present a different approach towards finding similarities between classes of words. The authors claim that texts available online are full of direct or indirect human stereotypes. As a result, word embeddings are prone to learn and maybe amplify those biases and propagate them further into AI models that are used for various applications. Cosine distance is used in the article as a measure to first argue for the existence of multi-class bias and then to show how through bias mitigation techniques the bias may be decreased.

They 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 $T = \{t_1, \dots, t_k\}$ 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:

$$S(t_i, A_j) = \frac{1}{|A_j|} \sum_{a \in A_j} \mathsf{cos}(t, a) \tag{2.2}$$

$$MAC(T, A) = \frac{1}{|T|\,|A|} \sum_{t_i \in T} \sum_{A_j \in A} S(t_i, A_j) \tag{2.3}$$

As the code has been provided by the authors we were able to reconstruct their results. The main steps in the procedure are as follows. Let us go through an example that refers to the process of hard debiasing on religious attributes.

First we load word embeddings from reddit.US.txt.tok.clean.cleanedforw2v.w2v dataset. The word embeddings have only 50 dimensions and the number of individual words from the dataset is 44895. The authors assume that the protected group should ideally not have high cosine similarity to stereotypical words. The word embeddings geometry should not place this group close to harmful stereotypes, if it is to be bias-free. For instance, let's look at the religion-related words.

Protected words by religion type:

”jew” : [”judaism”, ”jew”, ”synagogue”, ”torah”, ”rabbi”]

”christian” : [”christianity”, ”christian”, ”church”, ”bible”, ”priest”]

”muslim” : [”islam”, ”muslim”, ”mosque”, ”quran”, ”imam”]

Stereotypical words by religion type:

”jew” : [”greedy”, ”cheap”, ”hairy”, ”liberal”]

”christian” : [”judgemental”, ”conservative”, ”familial”]

”muslim” : [”violent”, ”terrorist”, ”dirty”, ”uneducated”]

We have prepared a table presenting the values of cosine distance for each protected word with each attribute (stereotype). The part of the results is shown below.

```
religion <- read.csv("../datasets/religionReddit.csv")[-1]
colnames(religion) <- c("protectedWord","wordToCompare","wordClass",
                        "cosineDistance","cosineSimilarity","connection")
religion$wordClass <- as.factor(religion$wordClass)
levels(religion$wordClass) <- c("christian","human","jewish","muslim","neutral")
head(religion)  %>%  kable(format = "latex",booktabs=T,
                      linesep = "",  escape = FALSE,
                      caption = "Head of the religion dataset.") %>%
                      kable_styling(latex_options=c("scale_down"))
```

Table 2.3: Head of the religion dataset.

protectedWord	wordToCompare	wordClass	cosineDistance	cosineSimilarity	connection
judaism	violent	muslim	0.7141939	0.2858061	different
judaism	terrorist	muslim	0.7461333	0.2538667	different
judaism	dirty	muslim	1.2002599	-0.2002599	different
judaism	uneducated	muslim	0.7885469	0.2114531	different
judaism	greedy	jewish	1.0026172	-0.0026172	associated
judaism	cheap	jewish	1.2323229	-0.2323229	associated

In the article there was no analysis of individual distances but the general look at the data through the usage of mean. The authors introduced a metric that tries to generalize the presence of bias through the classification of multi-class bias in groups of words connected with gender, religion or race. In the process they first take the mean of cosine distances between a given protected word and attributes assigned to each stereotype. They do not differentiate between stereotypes associated with a word and stereotypes associated with different words (in case of religion, stereotypes characteristic for Christianity has also cosine distance measured with for instance, Judaism or Islam). Then, after collecting the list of mean cosine distances, they average the list to obtain one final value representing the whole group, in this example religion, for which the final mean of all mean distances is equal to 0.859.

In the article the authors also try to remove previously defined biases from word embedding. First they identify the bias subspace using Principal Component Analysis (PCA) which is a technique for dimensionality reduction. It is applied here to choose the subspace that contains the greatest amount of information. There can be many subspaces found in a given group, for example in terms of religion one can identify at least a few sets that are to grasp the concept of religion in general:

[”judaism”, ”christianity”, ”islam”]

[”jew”, ”christian”, ”muslim”]

[”synagogue”, ”church”, ”mosque”]

The idea is to find a set that provides enough information to create from it a vector representing the concept of religion among words. This strategy is based on the idea that different dimensions of vectors contain different types of information and in some words in vector layers (subspaces) the information about religiousness is implicitly conveyed. In some cases this knowledge is useful but in the case of harmful stereotypes one does not want to have religion concept in stereotypical words.

After finding bias subspace, they use it to modify the vector values individually so that their cosine distances towards certain words are changed. In the case of stereotypes the aim is to make the cosine distances larger so that the association between protected word and harmful stereotype is smaller.

In the final step there is an evaluation of the results. The cosine distances are calculated again but this time using debiased vocabulary. After taking the mean of all distances one final value is obtained and then it is compared with the average value from the beginning. If the cosine distances are on average greater than before then it leads the authors to the conclusion that improvement was made. As the cosine distance increases it is assumed that the association between protected and stereotypical words decreases.

2.5 Limitations of the approach

1. Selection of attributes

The attributes are taken from different sources, there is no principled justification for their choice. From our analysis it will become clear that the list is rather uneven.

There is no mention of methodology for deciding on the number of attributes necessary to decide a hypothesis on the given size of dataset. There are however some ways to estimate how many samples we need to make sure that the result is significant. Our research will show that the numbers used are rather insufficient.

2. No control while taking the mean of cosines

The authors use the mean average cosine similarity to check on multi-class similarity between protected word and harmful stereotypes. They average the results until they obtain one final value to represent the mean cosine distance between protected word from a given class and the attributes of that class. As there is no control of the individual values of cosine distances, one can observe values that have cosine distances greater than 1 which would suggest that these compare words do not have significant similarity. However, as there is a lack of control, even such words are taken into the account for calculations.

3. The lack of uncertainty

A mean hides this issue and as there are pairs having negative and small similarities and there are those that have similarity equal to 0.5, the resulting calculation seems to be in norm. Additionally in such method the uncertainty is also not included which makes it even more difficult to give reasonable interpretations of the results. We propose the use of Bayesian method to obtain some understanding of the influence the uncertainty has on the interpretation of final results.

4. No word class distinction

In the original paper words from all three religions were checked with all of the stereotypes which means that there was no distinction on whether the stereotype is associated with given religion or with the other. Not all of the stereotypical words should be considered as harmful for all of the religions. In our analysis we distinguished between associated stereotypes, not associated (stereotype is valid but for different religion than currently checked) and none (meaning neutral words).

5. Interpreting the results

Assuming for a moment that the value of multi-class cosine distance is correct, one may question the results' interpretation. In Manzini, Lim, Tsvetkov, & Black (2019) Table 2, there are summarized the averages of cosine distance per group (gender, race, religion). I would like to focus now on analyzing the values relating to religious biases. Here is the fragment of table that refers to that:

Religion Debiasing	MAC
Biased	0.859
Hard Debiased	0.934
Soft Debiased ($\lambda = 0.2$)	0.894

MAC stand for mean average cosine similarity although in reality the values of cosine distance are stored there. What may attract attention is the fact that the value of cosine distance in "Biased" category is already quite high even before "debiasing." High cosine distance indicates low cosine similarity between values. One could think that average cosine similarity equal to approximately 0.141 is not significant enough to consider it as biased. However the authors aim to mitigate "biases" in vectors with such great distance to make it even larger. Methodologically there is a question on what basis this small similarity is still considered as a proof of bias presence.

This is in general the problem of scale and the lack of universal intervals that could be applied to know whether the cosine similarity is high enough to considered given words more similar that if we chose them on random. It seems like similar practice could be beneficial for metrics aimed at measuring similarity between words.

6. The curse of dimensionality

Curse of dimensionality may take place when there is an increase in volume of data that results in adding extra dimensions to the Euclidean space. According to the article “https://analyticsindiamag.com/curse-of-dimensionality-and-what-beginners-should-do-to-overcome-it/” as the number of features increases, it may be harder and harder to obtain useful information from the data with the usage of available algorithms. One may notice that more data should contribute to greater amount of information but more information also means greater risk of noise and distractions in data. At the same time, many times modern solutions are adapted to smaller dimensions and their results in higher ones are not intuitive or may be prone to be mistaken.

Using cosine similarity in high dimensions in word embeddings may also be prone to the curse of dimensionality. According to this article “https://www.researchgate.net/publication/327498046_The_Curse_of_Dimensionality_Inside_Out” there are reasons to consider this phenomenon when searching for word similarities in higher dimensions.

In the article an experiment is conducted that aims at showing how the similarity values and variation change as the number of dimensions increases. The hypothesis made in the paper states that two things will happen as the number of dimensions increase, the first one is that effort required to measure cosine similarity will be greater and the second one is that the similarity between data will blur out and have less variation. In details, the authors generate random points with increasing number of dimensions where each dimension of a data point is given a value between 0 and 1. Then they pick one vector at random from each dimension class and calculate cosine similarity between the chosen vector and the rest of the data. Then they check how the variation of values changes as the number of dimensions increases. It seems like the more dimensions there are, the smaller the variance and therefore it is less obvious how to interpret the resulting cosine similarities. Maybe the scale should be adjusted to the number of dimensions and variance so that it still gives us sensible information about data. According to some researches the cosine similarity in high dimensions is not reliable enough to trust it as it may be the case that choosing words on random may result in getting similar values as when picking them consciously.

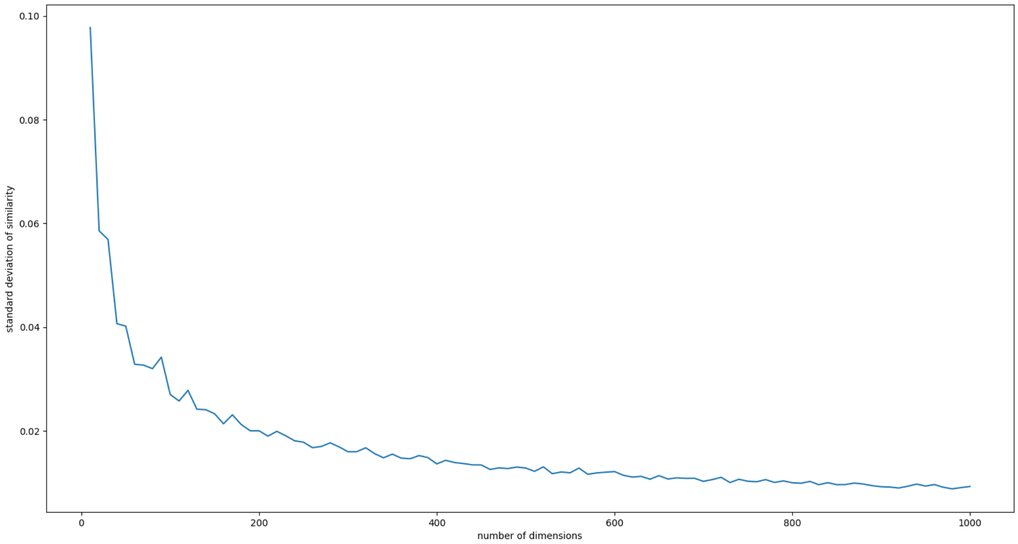


Figure 2.1: curse of dimensionality, number of dimensions on the x axis, standard deviation of similarity on the y axis

Chapter 3

Walkthrough with the religion dataset

3.1 Loading and understanding the dataset

We start with loading the libraries needed for the analysis.

```
library(ggplot2)
library(ggthemes)
library(rethinking)
library(tidyverse)
library(ggpubr)
library(kableExtra)
library(dplyr)
library(ggExtra)
library(cowplot)
```

We will use the choice of protected words and stereotypical predicates used in REF. This is a decent point of departure, not only we want to compare our method to that of REF, but also because this data format is fairly general (as contrasted, say, with a set up for binary stereotypes). Note also that the method we develop here can fairly easily be run for different stereotypization patterns. Let's start with explaining the method and its deployment using a dataset obtained for the religion-related protected words.

Let's load, clean a bit and inspect the head of the religion dataset we prepared. In order to obtain this dataset, we calculated the cosine distance between each protected word and each word from both the bias-related attribute groups, which were used in the original study, and to neutral and human control attributes which we added as control groups. For instance, for religion, the bias-related predicates (coming from the original study in REF) include muslim bias attributes, jew bias attributes, christian bias attributes (see a list in the APPENDIX).

We decided to add control groups in the form of two classes — neutral words and human-related words. Without a proper control group it is quite hard to compare the resulting cosine distances and decide on their significance in bias detection. We prepared approximately 230 neutral words to double-check the prima-facie neutral hypothesis that their cosine similarity to the protected words will oscillate around 0 (that is, the distances will be around 1). This provides us with a more reliable point of reference. Moreover, we added human attributes that are associated with people in general to investigate whether the smaller cosine distance between protected words and stereotypes can result simply from the fact that the stereotype predicates are associated with humans. For two control groups, we have randomly drawn 230 words that do not express any property usually attributed to humans, and human related attributes.

```r
religion <- read.csv("../datasets/religionReddit.csv")[-1]
colnames(religion) <- c("protectedWord","wordToCompare","wordClass",
                        "cosineDistance","cosineSimilarity","connection")
#head(religion)
#library(plyr)
religion$wordClass <- as.factor(religion$wordClass)
```

Table 3.1: Head of the religion dataset.

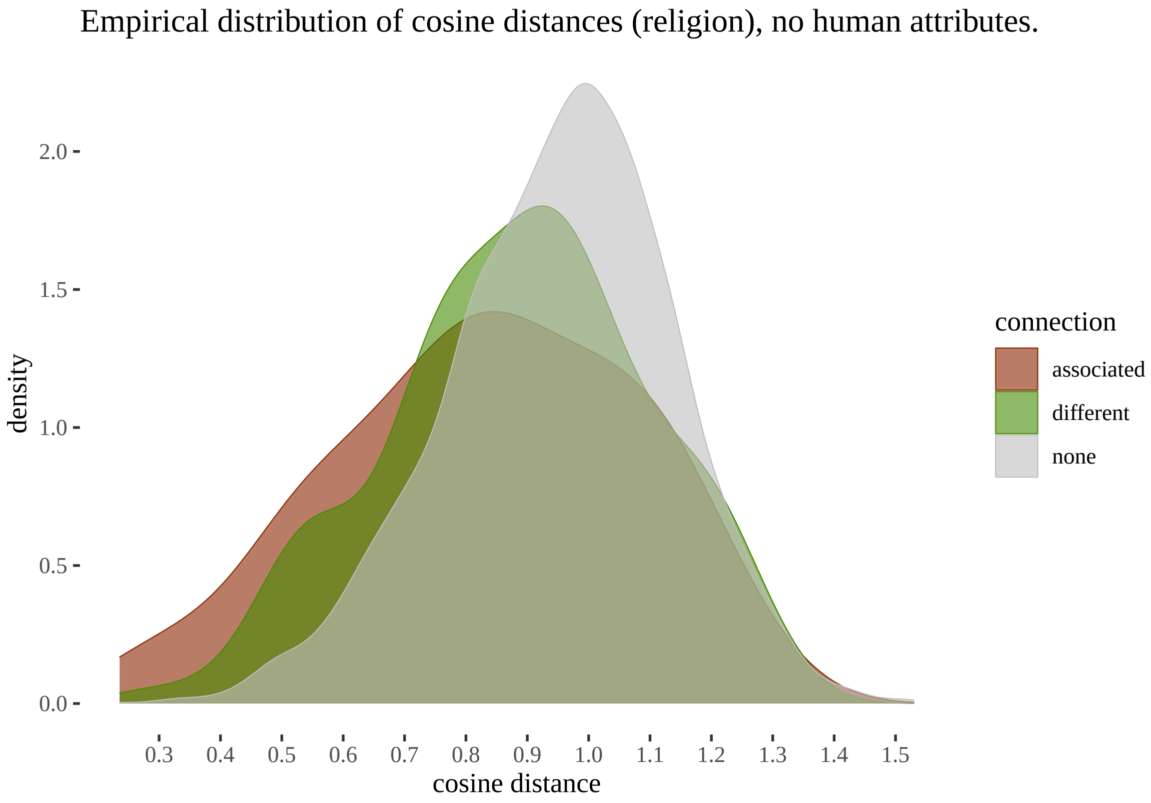
protectedWord	wordToCompare	wordClass	cosineDistance	cosineSimilarity	connection
judaism	violent	muslim	0.7141939	0.2858061	different
judaism	terrorist	muslim	0.7461333	0.2538667	different
judaism	dirty	muslim	1.2002599	-0.2002599	different
judaism	uneducated	muslim	0.7885469	0.2114531	different
judaism	greedy	jewish	1.0026172	-0.0026172	associated
judaism	cheap	jewish	1.2323229	-0.2323229	associated

```r
#levels(religion$wordClass)
#religion$wordClass <- mapvalues(religion$wordClass,
#                          from = levels(religion$wordClass),
#                          to = c("christian","human","jewish","muslim","neutral"))
levels(religion$wordClass) <- c("christian","human","jewish","muslim","neutral")
head(religion)  %>%  kable(format = "latex",booktabs=T,
                      linesep = "",  escape = FALSE,
                      caption = "Head of the religion dataset.") %>%      kable_styling(latex_opti
```

The `protectedWord` column contains words from a protected class that (in a perfect world according to the assumptions of the orignal study) should not be associated with harmful stereotypes. `wordToCompare` contains attributes, including stereotypes and control group words. For each row we compute the cosine distances between a given protected word and a given attribute word. `wordClass` tells us which class an attribute is supposed to be stereotypically associated with, that is, whether the word from `wordToCompare` is associated stereotypically with jews, christians or muslims, or whether it belongs to a control group. `cosineDistance` is simply a calculation of the cosine distance between protected word and atrribute. `cosineSimilarity` contains the result of substracting cosine distance from 1. `connection` contains information about the relation type between a protected word and an attribute. If the attribute is e.g. a harmful jewish stereotype and the protected word is also from the judaism group, the connection has value `associated`. If the attribute is still stereotypically jewish, but the protected word comes from another religion, the connection is labelled as `different`. If the attribute belongs to a neutral group then the connection is labelled as `none` and if an attribute belongs to the `human` class, then the connection is labelled as `human`.

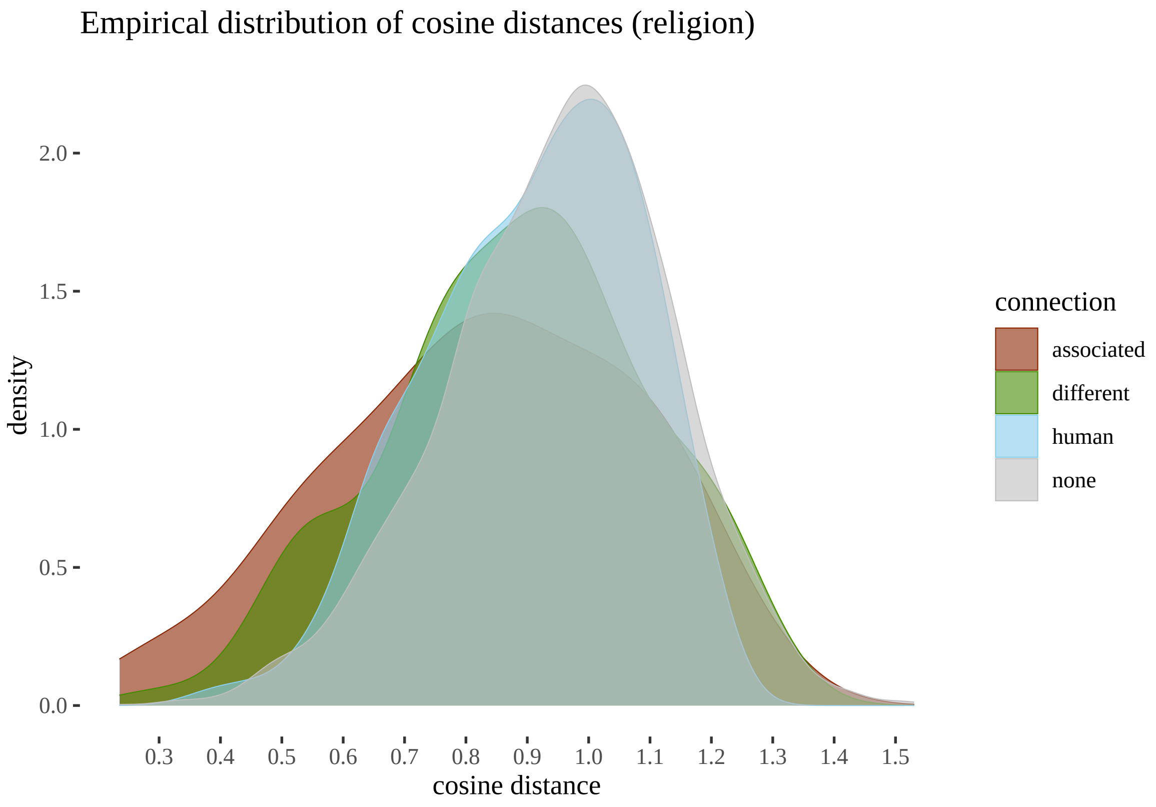
3.2 First look at the empirical distributions

First let's take a look at the empirical distribution of distances by the connection type, initially ignoring the human control class for now.



The first impression is that while there is a shift for associated words towards smaller cosine distances as compared to the neutral words, slightly surprisingly a slightly weaker shift in the same direction is visible for attributes associated with different stereotypes.

Moreover, the empirical distributions overlap to alarge extent and the means grouped by connection type do not seem too far from each other. In fact, as there is a lot of variety in the cosine distances (as we will soon see), we need to gauge the uncertainty involved, and to look more carefully at individual protected words to get a better idea of how the cosine distance distribution changes for different attribute groups and different protected classes. Now, let's add the human attributes to the picture:



Notice that the distribution for `human` (even though we did our best not to include in it any stereotype-related atributes) is left-skewed, with much overlap with `associated` and `different`, which illustrates the need to take being associated with humans as an important predictor.

Our focus lies in `connection` as a predictor. Morever, later on we'll be interested in looking at the protected words separately, and at protected words split by connection.

For technical reasons it is useful to represent these factors as integer vectors.

```
religion$con <- as.integer(religion$connection)
```

```
## Warning: NAs introduced by coercion
```

```
religion$pw <- as.integer(religion$protectedWord)
```

```
## Warning: NAs introduced by coercion
```

```
religion$pwFactor <- factor(paste0(religion$protectedWord, religion$connection))
religion$pwIndex <- as.integer(religion$pwFactor)
```

A short script, `cleanDataset` to make this faster, so equivalently:

```
source("../functions/cleanDataset.R")
religion <- read.csv("../datasets/religionReddit.csv")[-1]
religion <- cleanDataset(religion,c("christian","human","jewish","muslim","neutral"))
```

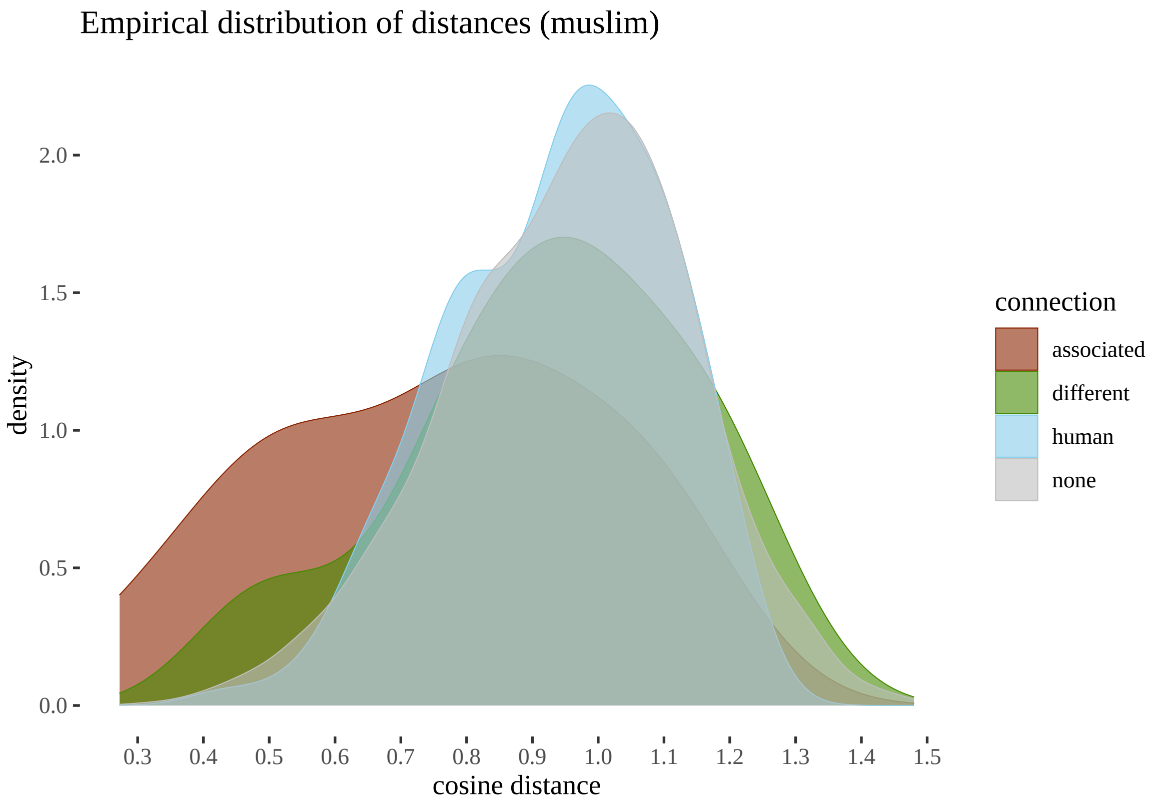
```
## Warning in cleanDataset(religion, c("christian", "human", "jewish", "muslim", :
## NAs introduced by coercion

## Warning in cleanDataset(religion, c("christian", "human", "jewish", "muslim", :
## NAs introduced by coercion
```

3.3 Looking at the islam-related words

For now, let's focus on five protected words related to islam ("imam," "islam," "mosque," "muslim," and "quran"). The word list associates with islam four stereotypical attributes ("violent," "terrorist," "uneducated" and "dirty"). First, we select and plot the empirical distributions for these protected words.

```
library(tidyverse)
muslimWords <- c("imam","islam","mosque","muslim","quran")
muslim <- religion %>% filter(protectedWord %in% muslimWords)
ggplot(muslim, aes(x =  cosineDistance, fill = connection, color = connection))+
  geom_density(alpha=0.6,size = .2)+
  scale_fill_manual(values = c("orangered4","chartreuse4", "skyblue", "gray"))+
  scale_x_continuous(breaks = seq(0.3,1.5, by = 0.1))+xlab("cosine distance")+
  scale_color_manual(values = c("orangered4","chartreuse4","skyblue","gray"))+
  theme_tufte()+ggtitle("Empirical distribution of distances (muslim)")
```



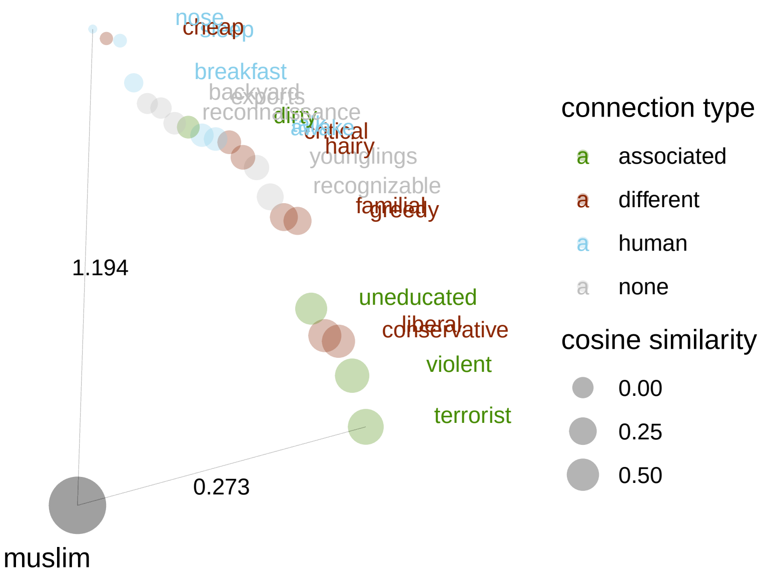
Once we focus on words related to islam, the associated bias seems to be stronger than in the whole dataset. This is a step towards illustrating that the distribution of bias is uneven.

Now, say we want to look at a single protected word. Since the dataset also contains comparison multiple control neutral and human attributes, we randomly select only 5 from `none` and 5 from `human` control groups of those for the visualisation purposes.

```
library(tidyverse)
muslimClass <- muslim %>% filter(protectedWord == "muslim")
neutralSample <- sample_n(filter(muslimClass,connection == "none"), 5)
humanSample <- sample_n(filter(muslimClass,connection == "human"), 5)
muslimVis <- muslimClass %>% filter(connection != "none" & connection !="human")
muslimVis <- rbind(muslimVis,neutralSample,humanSample)
```

```
## Warning in `[<-.factor`(`*tmp*`, ri, value = structure(c("muslim_stereotype", :
## invalid factor level, NA generated
```

```
#we plug in our visualisation script
source("../functions/visualisationTools.R")
#two arguments: dataset and protected word
visualiseProtected(muslimVis,"muslim")
```

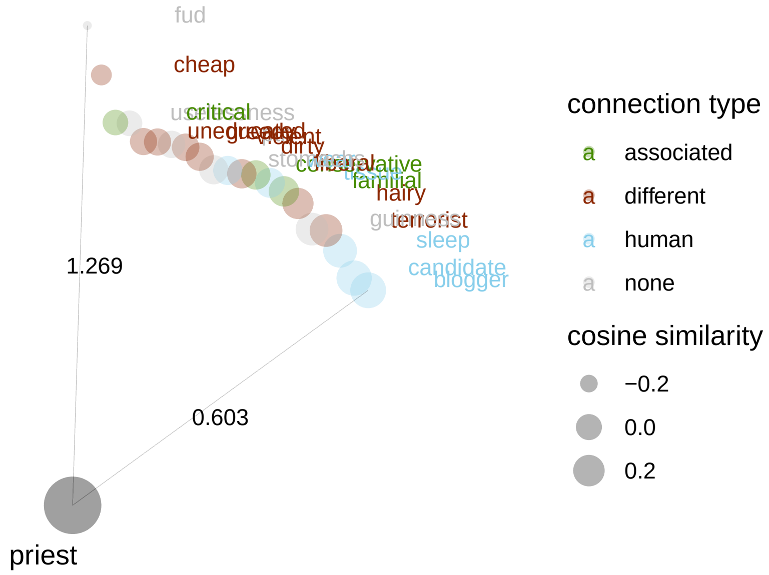


Note that the distance between the grey point and the other points is proportional to cosine distance, the non-grey point size is proportional to cosine similarity to the protected word, and color groups by the connection type. So for `muslim` it seems that the stereotypes coming from the word list are fairly well visible. To give you some taste of how uneven the dataset is, compare this to what happens with `priest`.

```
library(tidyverse)
priestClass <- religion %>% filter(protectedWord == "priest")
neutralSample <- sample_n(filter(priestClass,connection == "none"), 5)
humanSample <- sample_n(filter(priestClass,connection == "human"), 5)
priestVis <- priestClass %>% filter(connection != "none" & connection !="human")
priestVis <- rbind(priestVis,neutralSample,humanSample)
```

```
## Warning in `[<-.factor`(`*tmp*`, ri, value = structure(c("muslim_stereotype", :
## invalid factor level, NA generated
```

```
#we plug in our visualisation script
source("../functions/visualisationTools.R")
#two arguments: dataset and protected word
visualiseProtected(priestVis,"priest")
```



Here you can see that some human attributes are closer than stereotype attributes, and that there is no clear reason to claim that `associated` attributes are closer than `different` or `human` attributes. This, again, illustrates the need of case-by-case analysis with control groups.

The general idea now is that the word lists provided in different pieces of research are just samples of attributes associates with various stereotypes and should be treated as such: the uncertainty involved and the sample sizes should have clear impact on our estimates.

3.4 Bayesian model structure and assumptions

We will now think of cosine distance as the output variable, and will build a few Bayesian models to compare. First, we just build a baseline model which estimates cosine distance to the attributes separately for each protected word. The underlying idea is that different protected words might in general have different relations to all the attributes and these relations should be our point of departure.

Here is the intuition behind the mathematical Bayesian model involved. Our outcome variable is `cosine difference`, which we take to me normally distributed around the predicted mean for a given protected word (that is, we assume the residuals are normally distributed). The simplest model specification is:

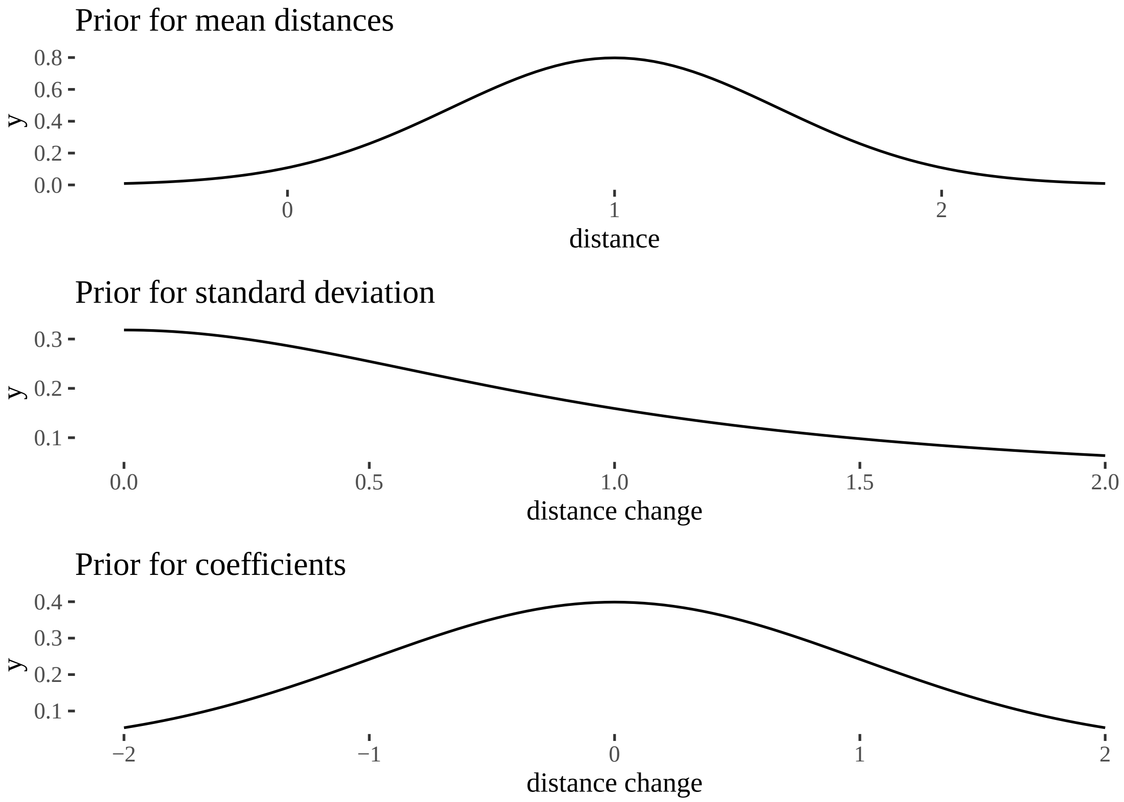
$$cosineDistance_i \sim dnorm(\mu_i, \sigma) \tag{3.1}$$

$$\mu_i = m_{pw} \tag{3.2}$$

$$m_{pw}\ dnorm(1, .5) \tag{3.3}$$

$$\sigma \sim dcauchy(0, 1) \tag{3.4}$$

That is, we assume the estimated means might be different for diferent protected words and our prior for the mean and the overal standard deviation are normal with mean 1 and sd=.5 and half-cauchy with parameters `0,1`. Further on we'll also estimate additional impact the connection type may have. For this impact we take a slightly skeptical prior centered around 0 distributed normally with sd = 1. These are fairly weak and slightly skeptical regularizing priors, which can be illustrated as follows:



3.5 Choosing predictors

Now we can define and compile the baseline model. Its parameters will have a posterior distribution obtained using either Hamiltonian Monte Carlo methods (STAN) available through the `rethinking` package.

```
library(rethinking)
options(buildtools.check = function(action) TRUE )
religionBaseline <- ulam(
  alist(
    cosineDistance ~ dnorm(mu,sigma),
    mu <- m[pw],
    m[pw] ~ dnorm(1,.5),
    sigma ~ dcauchy(0,1)
  ),
  data = religion,
  chains=2 , iter=4000 , warmup=1000,
  start= list(mu = 1, co = 0, sigma= .3),
  log_lik = TRUE, cores=4
)
#saving
#saveRDS(religionBaseline,
#file = "cosineAnalysis/models/religionBaseline.rds")
```

The only reason we need it is the evaluation of connection as a predictor. Does including it in o the model improve the situation? To investigate this, let's now build a model according to the following specification:

$$cosineDistance_i \sim dnorm(\mu_i, \sigma) \tag{3.5}$$

$$\mu_i = m_{pw} + co_{con} \tag{3.6}$$

$$m_{pw}\ dnorm(1, .5) \tag{3.7}$$

$$co_{con}\ dnorm(0, 1) \tag{3.8}$$

$$\sigma \sim dcauchy(0, 1) \tag{3.9}$$

The idea now is that each connection type comes with its own coefficient co that has impact on mean distances for protected words taken separately.

```
library(rethinking)
options(buildtools.check = function(action) TRUE )
religionCoefs <- ulam(
  alist(
    cosineDistance ~ dnorm(mu,sigma),
    mu <- m[pw] + co[con],
    m[pw] ~ dnorm(1,.5),
    co[con] ~dnorm(0,.5),
    sigma ~ dcauchy(0,1)
  ),
  data = religion,
  chains=2 , iter=8000 , warmup=1000,
  log_lik = TRUE
)
```

First, let's see if this model is really better in terms of the Widely Acceptable Information Criterion (WAIC):

```
##                    WAIC SE dWAIC dSE pWAIC weight
## religionCoefs    -2328 93     0  NA    20      1
## religionBaseline -2283 95    45  17    16      0
```

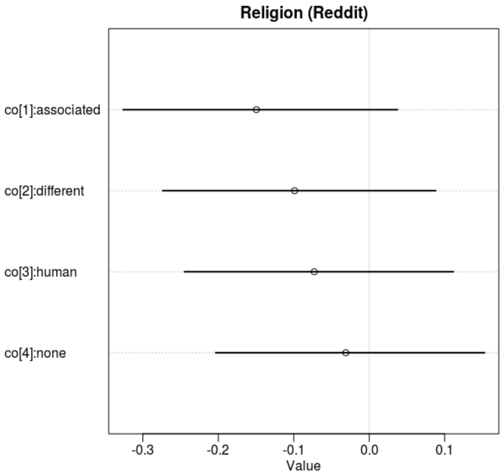
Clearly, it should be given weight 1 as compared to the baseline model. So far, we've learned that the connection type actually has predictive value. Let's take a look at the coefficient estimates:

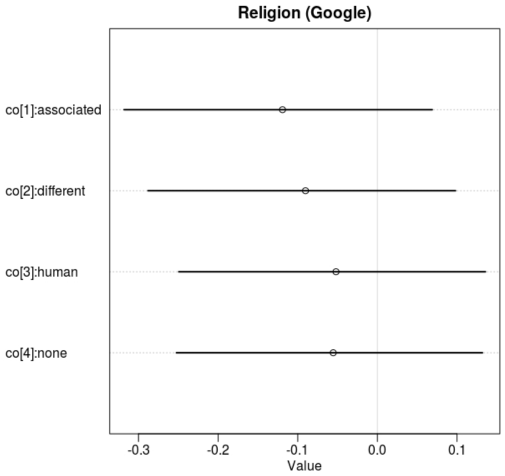
```
##              mean        sd       5.5%      94.5%    n_eff    Rhat4
## co[1] -0.14956420 0.1151675 -0.3261650 0.03741930 294.2033 1.001449
## co[2] -0.09880543 0.1145024 -0.2736985 0.08813271 291.5044 1.001564
## co[3] -0.07282752 0.1133894 -0.2447778 0.11158986 287.7820 1.001627
## co[4] -0.03103179 0.1131420 -0.2034442 0.15268770 286.8283 1.001606
```

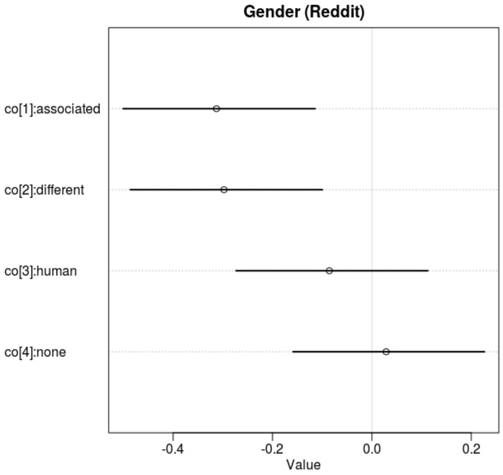
3.6 Dataset-level coefficients

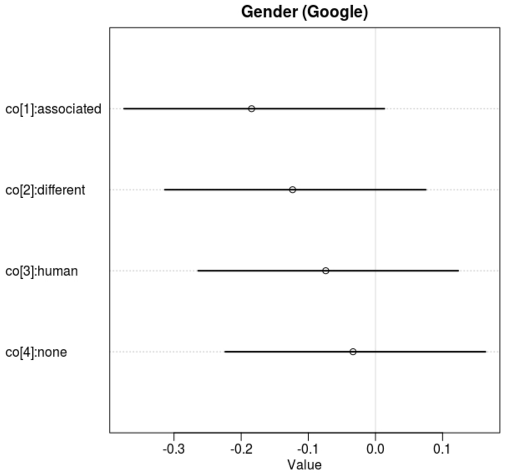
Let's plot them together with their highest posterior density invervals, for the three topic groups.

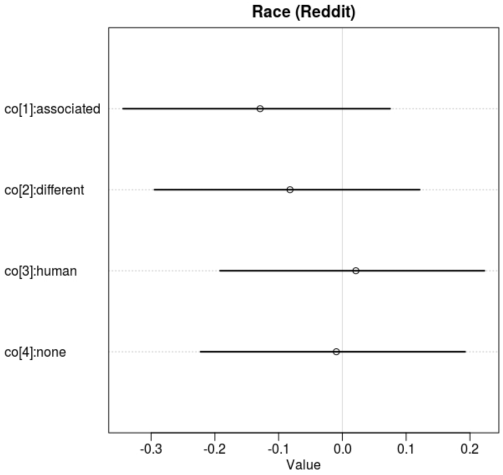
What should strike us is that while the mean estimates of the coefficients indeed do differ a bit, usually the highest posterior density invervals all include zero, and so we

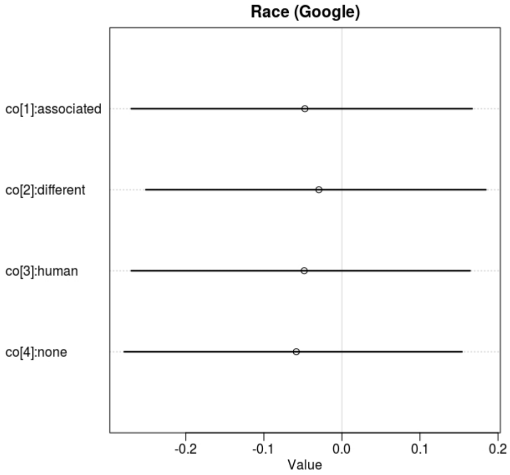

Religion (Reddit)
co[1]:associated
co[2]:different
co[3]:human
co[4]:none
-0.3
-0.2
-0.1
0.0
0.1
Value


Religion (Google)
co[1]:associated
co[2]:different
co[3]:human
co[4]:none
-0.3
-0.2
-0.1
0.0
0.1
Value


Gender (Reddit)
co[1]:associated
co[2]:different
co[3]:human
co[4]:none
-0.4
-0.2
0.0
0.2
Value


Gender (Google)
co[1]:associated
co[2]:different
co[3]:human
co[4]:none
-0.3
-0.2
-0.1
0.0
0.1
Value


Race (Reddit)
co[1]:associated
co[2]:different
co[3]:human
co[4]:none
-0.3
-0.2
-0.1
0.0
0.1
0.2
Value


Race (Google)
co[1]:associated
co[2]:different
co[3]:human
co[4]:none
-0.2
-0.1
0.0
0.1
0.2
Value

do not have strong reasons to say that, say, as far as the whole religion dataset is involved, being associated indeed is connected with lower cosine distance. A second striking observation is that the estimated impact for associated stereotypes is quite often not too different from the estimated impact of attributes associated with different stereotypes, and both are sometimes not too far from the estimated impact for simply human attributes. In general, once the uncertainty involved is taken seriously by using control groups and statistical uncertainty estimation that does not dispose of pointwise data, the picture which focuses only on differences between means of means is too simplistic.

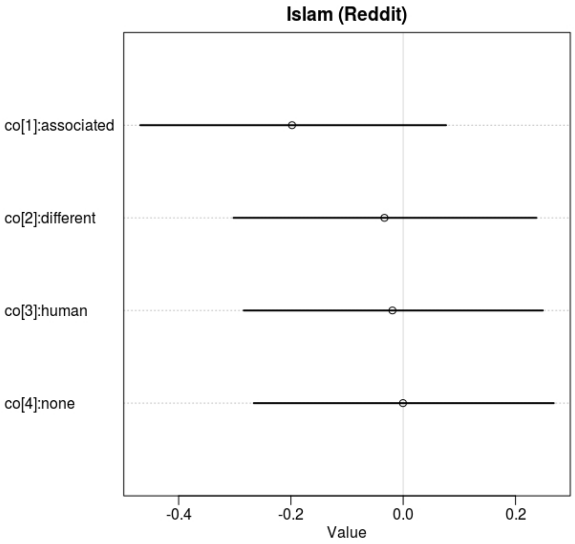
But this doesn't mean important differences for some protected words are not there. For one thing, if you start with a word list that is very uneven, the actually not so bad status of some of the protected words might mask a pretty bad situation in which some other protected words are. For comparison, let's see what a model focused on words related to islam tells us.

```r
#this is how we build the model
religion <- read.csv("cosineAnalysis/datasets/religionReddit.csv")[-1]
colnames(religion) <- c("protectedWord","wordToCompare","wordClass",
                        "cosineDistance","cosineSimilarity","connection")
levels(religion$wordClass) <- c("christian","human","jewish","muslim","neutral")
muslimWords <- c("imam","islam","mosque","muslim","quran")
muslim <- religion %>% filter(protectedWord %in% muslimWords)
muslim$protectedWord <- droplevels(muslim$protectedWord)
muslim$pw <- as.integer(muslim$protectedWord)
muslim$con <- as.integer(muslim$connection)
muslim$pwFactor <- factor(paste0(muslim$protectedWord, muslim$connection))
muslim$pwIndex <- as.integer(muslim$pwFactor)

islamCoefs <- ulam(
  alist(
    cosineDistance ~ dnorm(mu,sigma),
    mu <- m[pw] + co[con],
    m[pw] ~ dnorm(1,.5),
    co[con] ~dnorm(0,.5),
    sigma ~ dcauchy(0,1)
  ),
  data = muslim,
  chains=2 , iter=10000 , warmup=1000, cores = 4,
  log_lik = TRUE
)
```

Let's take a look at the coefficients:

```
##                  mean        sd       5.5%      94.5%    n_eff    Rhat4
## co[1] -0.1979930035 0.1708785 -0.4682696 0.07634977 1789.894 1.003445
## co[2] -0.0334215769 0.1687587 -0.3021954 0.23720938 1738.720 1.003575
## co[3] -0.0192492753 0.1675754 -0.2840596 0.24860974 1732.907 1.003755
## co[4] -0.0003911363 0.1670610 -0.2661047 0.26815172 1723.758 1.003837
```



While muslim words were unusual in the sense that the disparity between associated attributes and others is stronger, the evidence is still not conclusive. This is because the variation even within islam-related words is large enough (and sample sizes sufficiently small) for all the highest posterior density invervals to still include zeros.

So, it seems, taking a closer look does seem to make a difference. The question is, what happens if we do take a close look at the level of protected words?

Chapter 4

Protected-word level analysis

4.1 Model structure and assumptions

Let's turn then to data analysis that takes a look at protected words separately. This time for each combination of a protected word and a connection status we will have a separate mean cosine distance estimate, each coming with its own highest posterior density interval. This means we will use indices that are result from all such combinations (and then we will split them up in the model precis to build visualisation, feel free to look at the `visualiseStats.R` script for details).

```r
options(buildtools.check = function(action) TRUE ) #removes install pop-up request
religion$pwFactor <- factor(paste0(religion$protectedWord, "-", religion$connection))
religion$pwIndex <- as.integer(religion$pwFactor)

religionSeparate <- ulam(
  alist(
    cosineDistance ~ dnorm(mu,sigma),
    mu <- c[pwIndex],
    c[pwIndex] ~ dnorm(1,.5),
    sigma ~ dcauchy(0,1)
  ),
  data = religion,
  chains=2 , iter=10000 , warmup=1000,
  start=list(no = 1, a = 0, d = 0, sigma= .3), log_lik =  TRUE
)
```

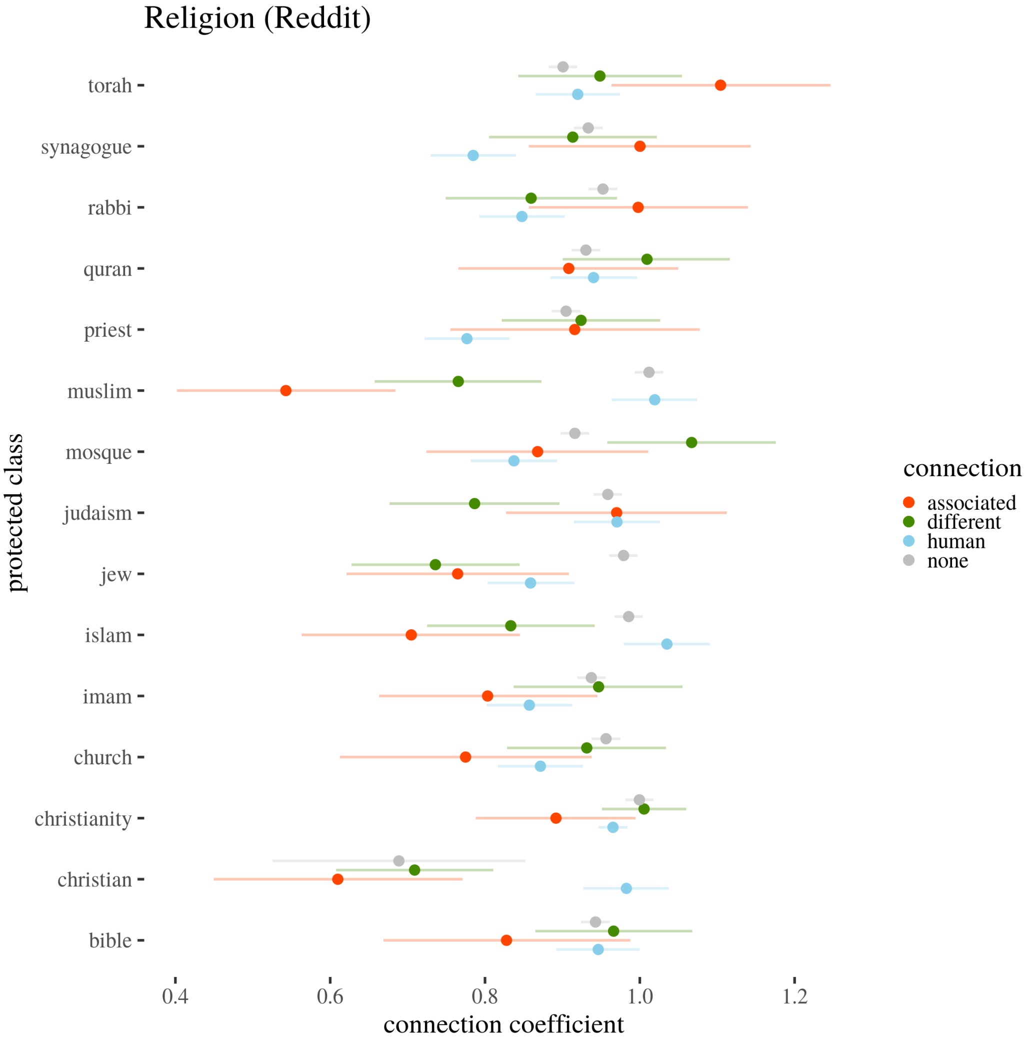
Let's see if the individualized model does better than the previous models in light of WAIC which does add penalty for the number of parameters.

```
compareBaselineCoefsSeparate<- readRDS("../datasets/compareBaselineCoefsSeparate.rds")
compareBaselineCoefsSeparate
```

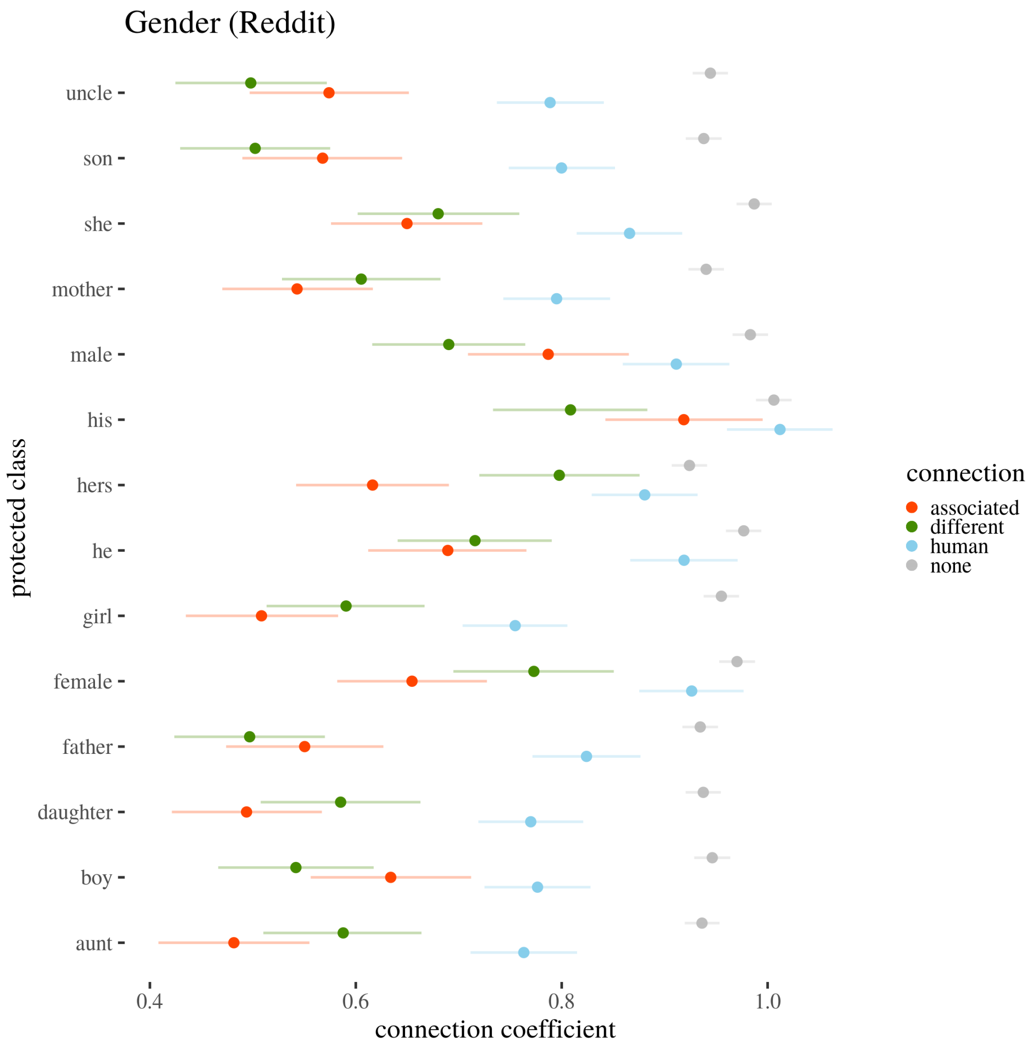
```
##                    WAIC SE dWAIC dSE pWAIC weight
## religionSeparate -2400 93     0  NA    60      1
## religionCoefs    -2328 93    72  29    20      0
## religionBaseline -2283 95   117  37    16      0
```

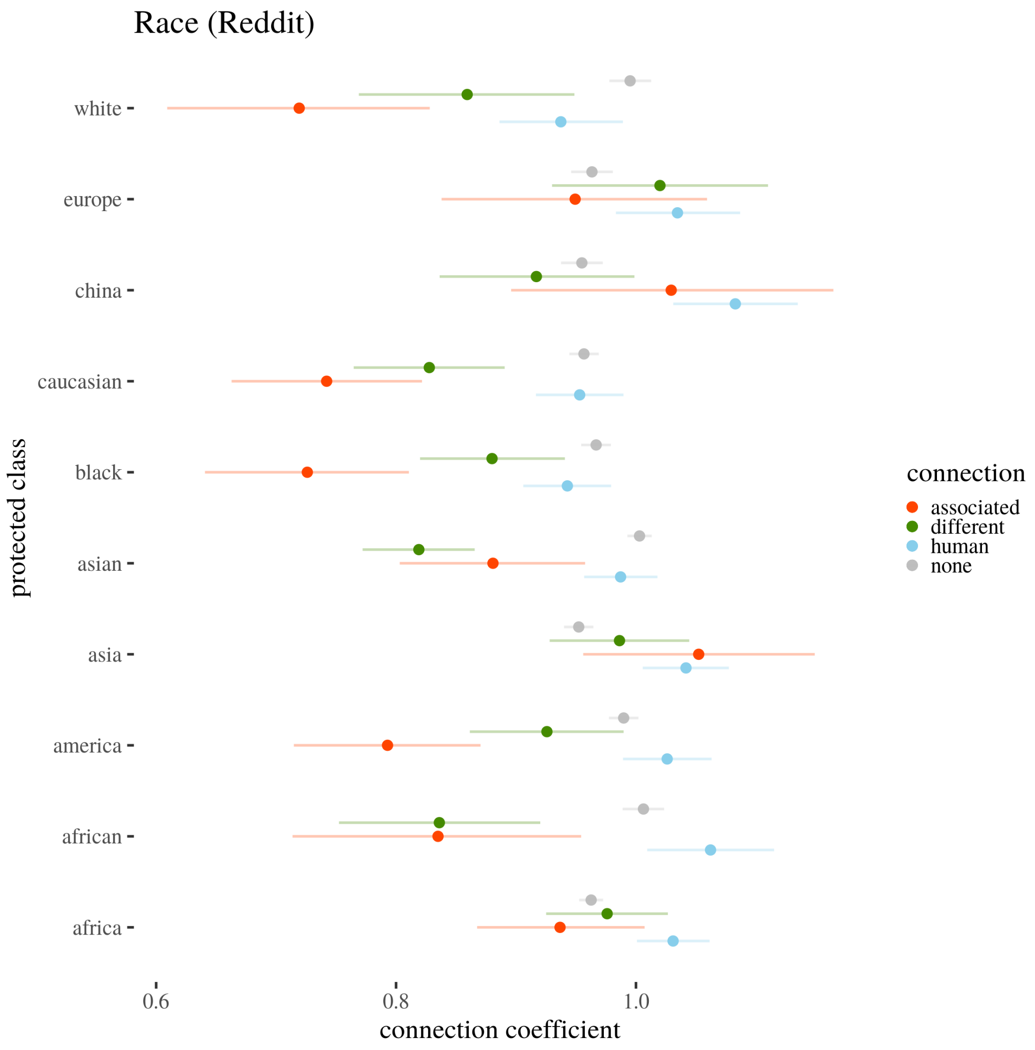
4.2 Protected classes in Reddit and Google embeddings

It seems that we do want to prefer this model, despite its relative complication. Now, what does it tell us about the protected words? Let's visualise the predicted means together with 89% highest posterior density intervals.

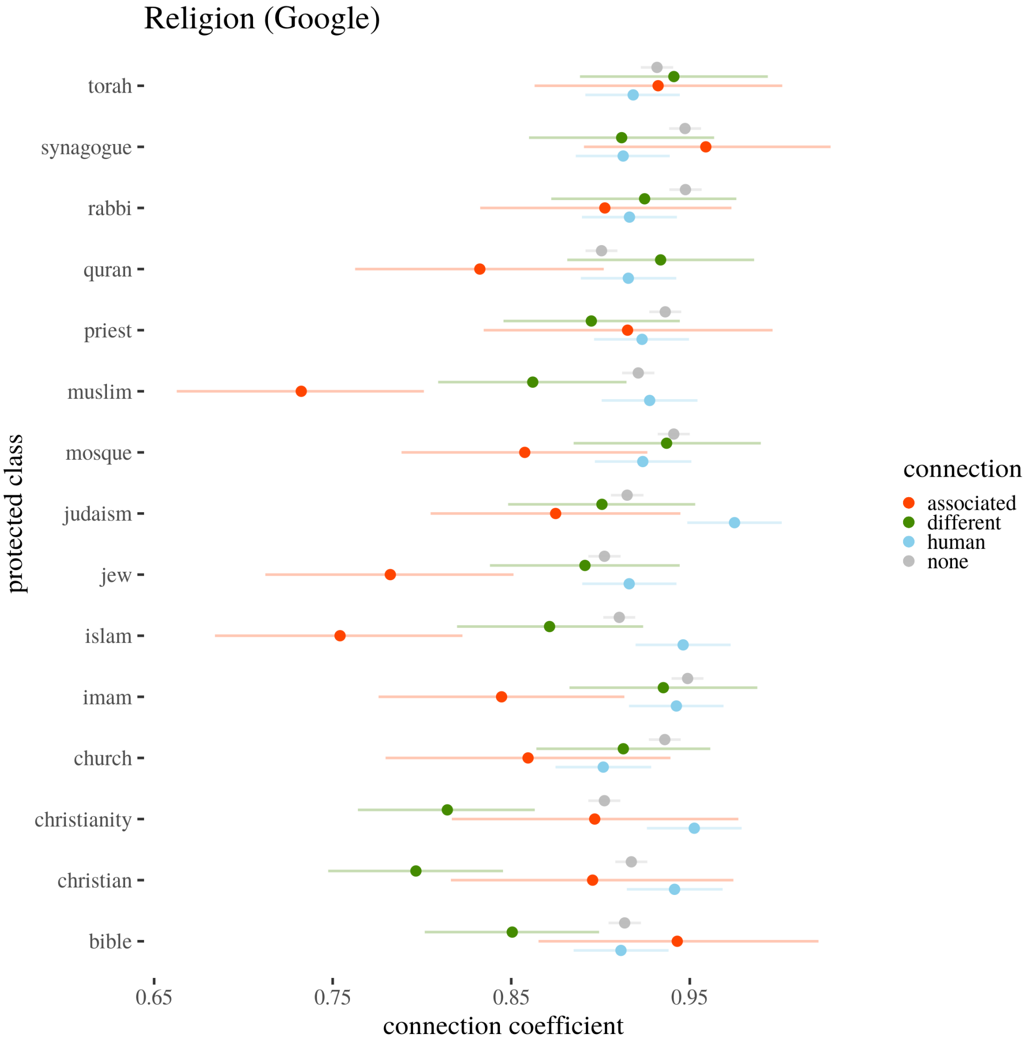


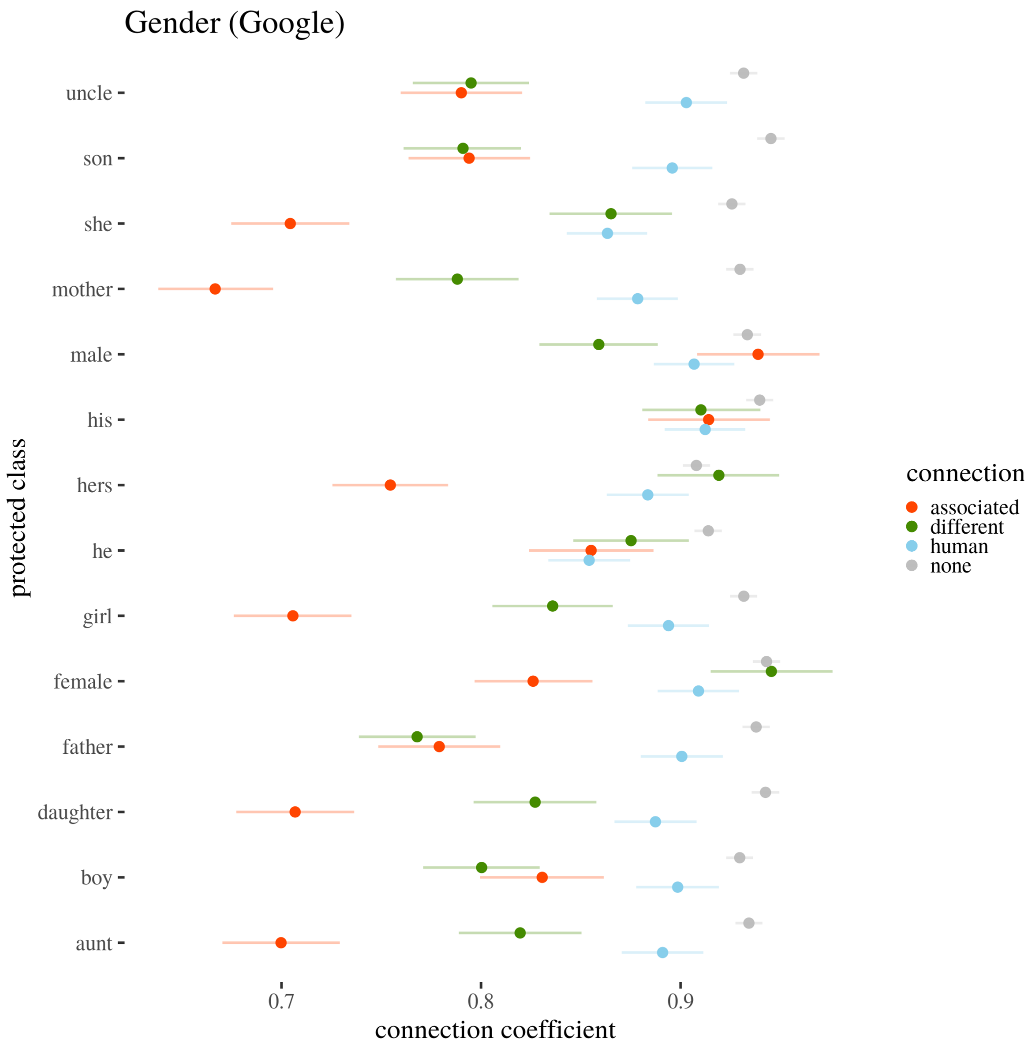
Before we move on, let's perform analogous analyses for the remaining types of supposed bias: gender and race (the model building is analogous).

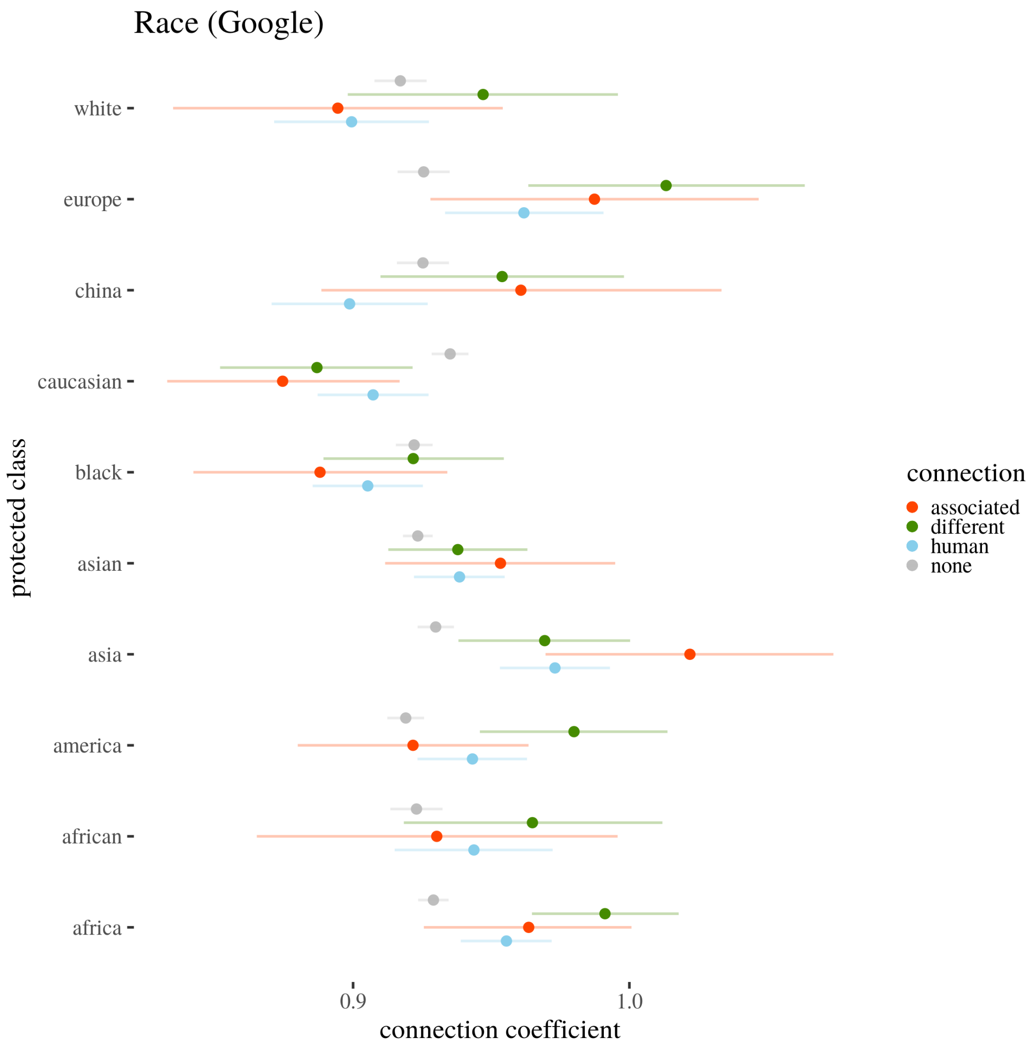

Gender (Reddit)
uncle
son
she
mother
male
his
hers
he
girl
female
father
daughter
boy
aunt
protected class
0.4
0.6
0.8
1.0
connection coefficient
connection
associated
different
human
none



We first encountered the GoogleNews word embedding model in Bolukbasi, Chang, Zou, Saligrama, & Kalai (2016). They use it for their calculations as written in Tolga Code. We decided to compare the results obtained with the use of Reddit L2 model and the ones that we got by GoogleNews model. The details of the model can be found here Google News model. One of the main differences between the models is that Reddit word embeggings have 50 dimensions and GoogleNews word embeddings have 300 dimensions. As the dimension is increasing, the vector can capture much more information but this is not always the case. According to some researchers the amount of dimensions should be chosen by taking into account the type of corpus and its features. In our case both Reddit and GoogleNews models are already trained and ready to use so we do not analyse further the choice of their hyperparameters.




Gender (Google)
protected class
uncle
son
she
mother
male
his
hers
he
girl
female
father
daughter
boy
aunt
0.7
0.8
0.9
connection coefficient
connection
associated
different
human
none


Race (Google)
protected class
white
europe
china
caucasian
black
asian
asia
america
african
africa
0.9
1.0
connection coefficient
connection
● associated
● different
● human
● none

Chapter 5

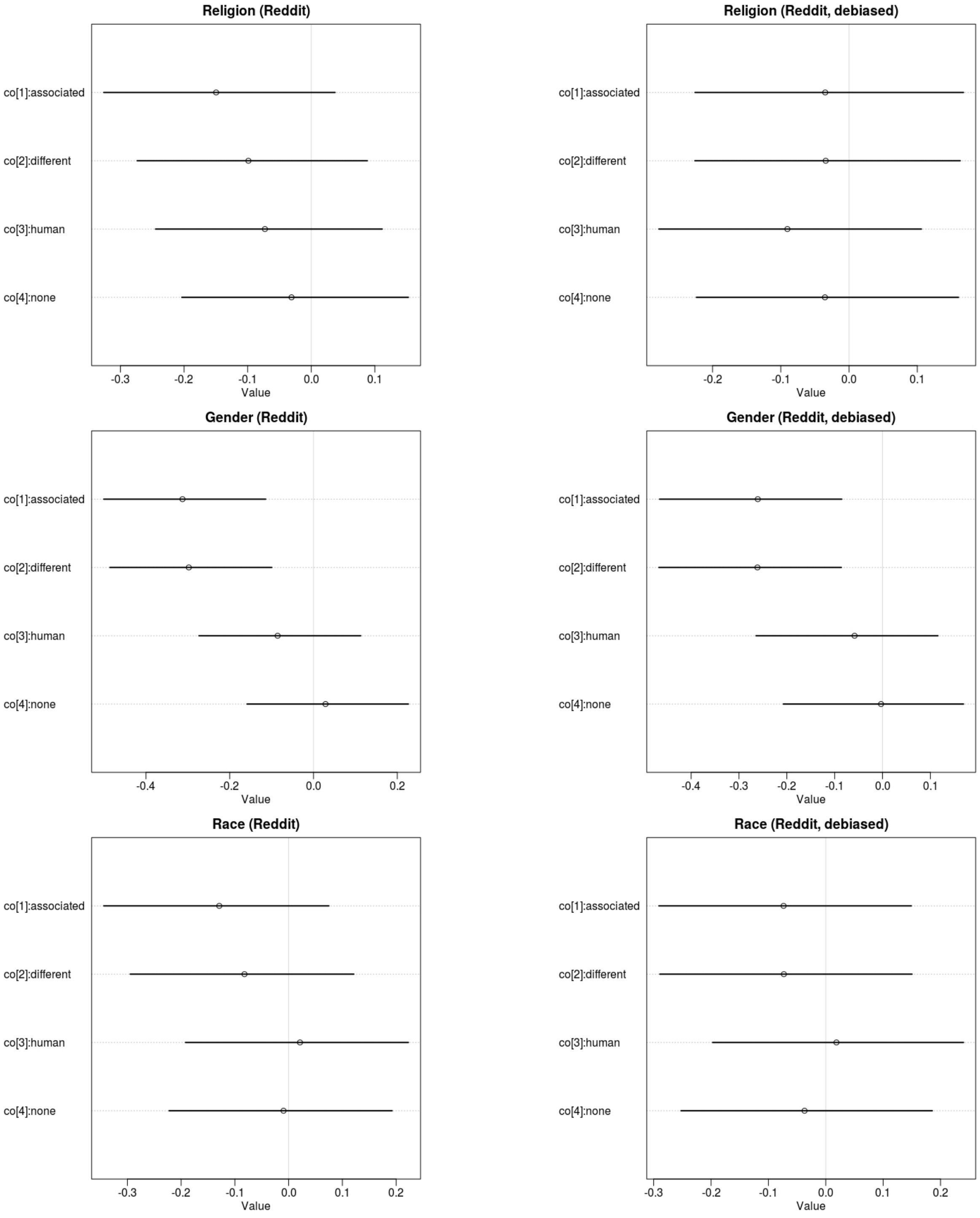
The role of debiasing

We also applied to the word embeddings hard debiasing method found in Manzini, Lim, Tsvetkov, & Black (2019). As it was mentioned before, debiasing consists of two components: identifying the bias subspace and then removing this vector subspace from the chosen embeddings. The aim is to increase the cosine distance between protected words and the set of attributes. After debiasing the Reddit word embeddings we calculated the cosine distance again to see if there are any significant changes in terms of the similarity between protected words and the attributes.

5.1 Dataset-level coefficients after debiasing

First, let's look at coefficient estimated for the whole datasets, as compared to their estimation prior to debiasing:

One may observe very slight changes in the estimated coefficients. It seems that in religion dataset the changes are the most significant as the mean moves towards zero



Religion (Reddit)

Religion (Reddit, debiased)

Gender (Reddit)

Gender (Reddit, debiased)

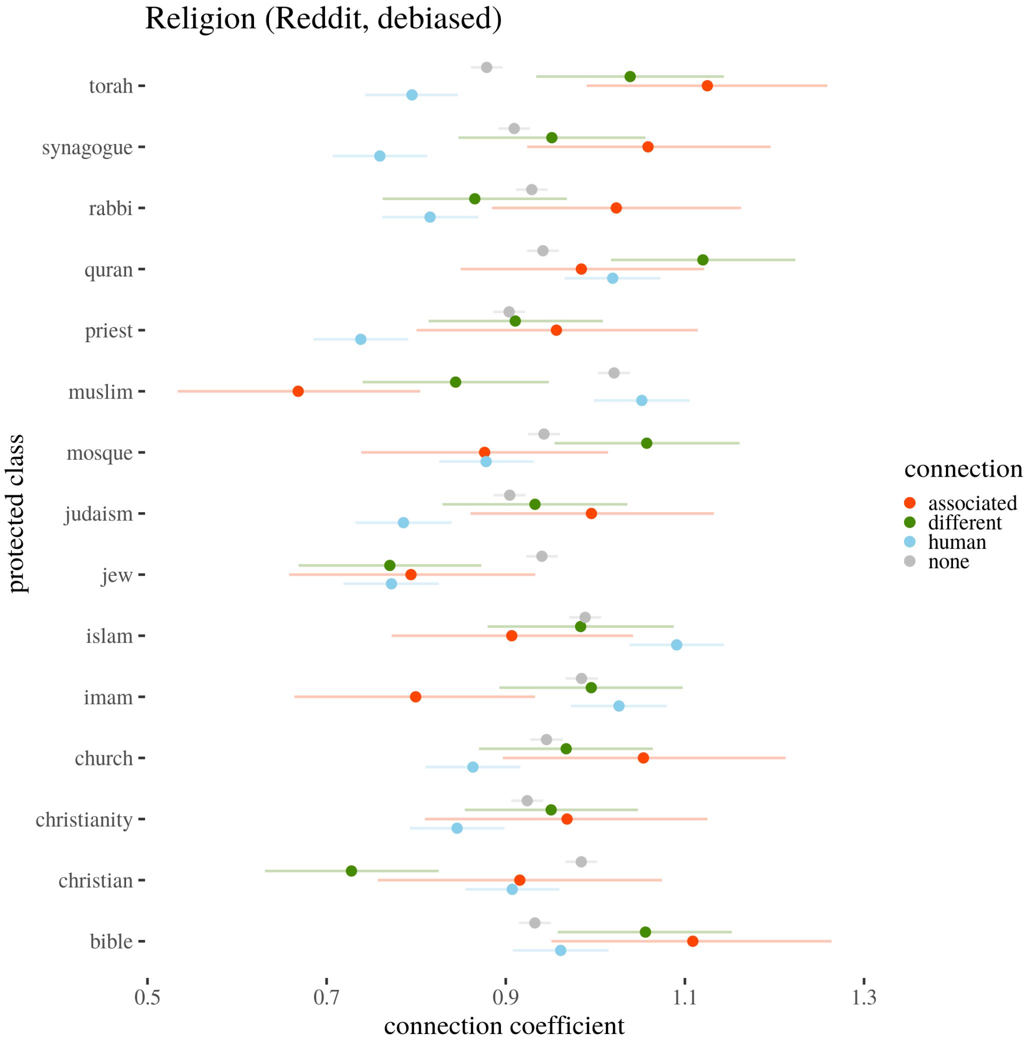
Race (Reddit)

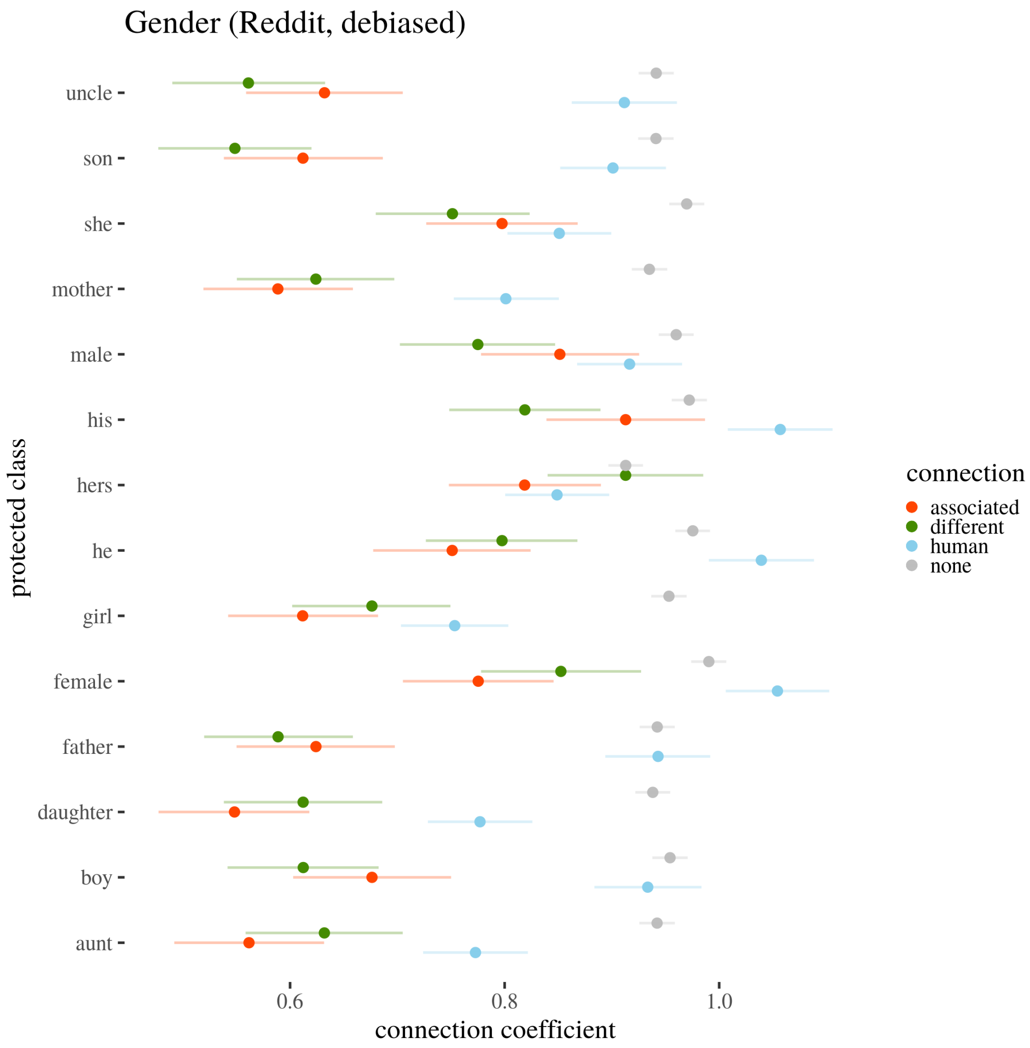
Race (Reddit, debiased)

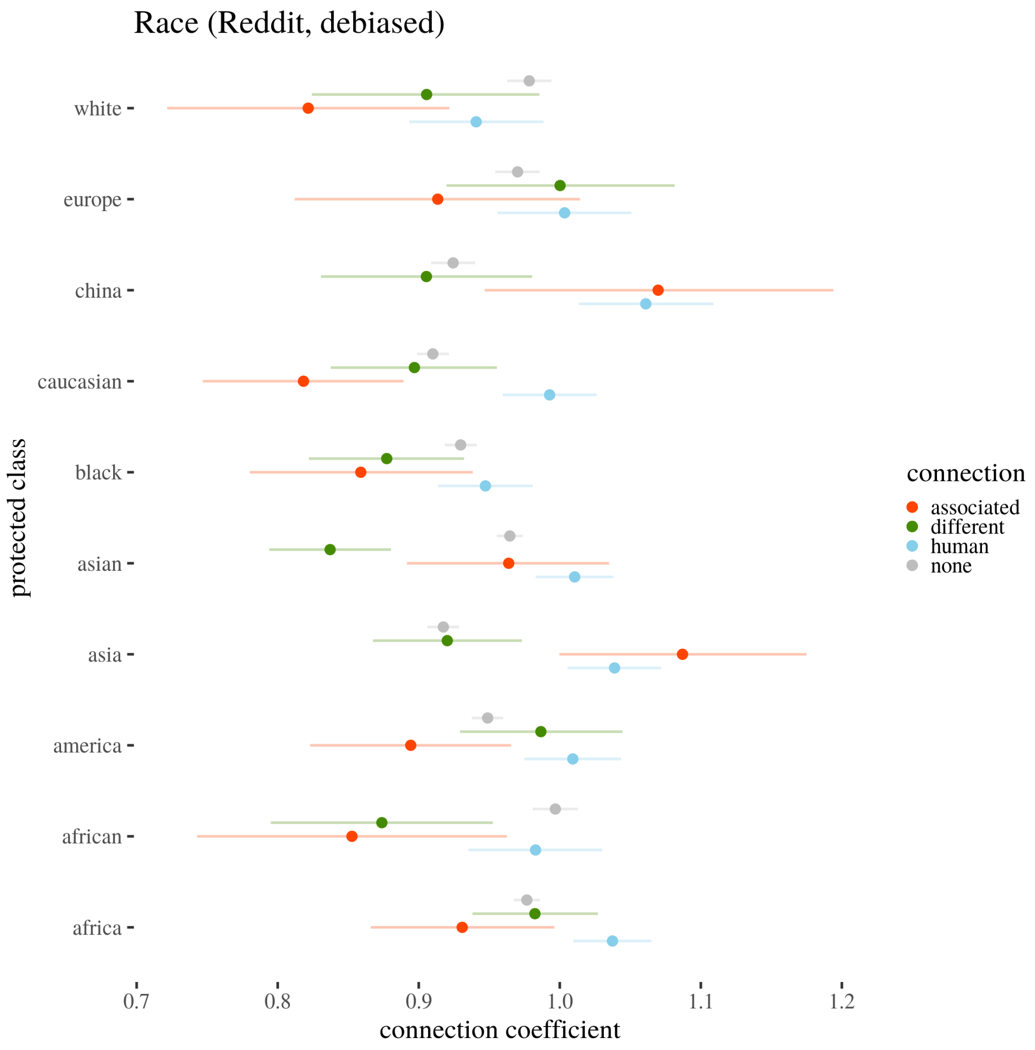
which stands for no similarity between words.

5.2 Protected classes after debiasing

Now, perhaps, the effects of debiasing will be better appreciated if we look at the level of protected words. After all, the hope is, the situation of extremely ill-positioned protected words have improved?




Gender (Reddit, debiased)
protected class
uncle
son
she
mother
male
his
hers
he
girl
female
father
daughter
boy
aunt
0.6
0.8
1.0
connection coefficient
connection
associated
different
human
none



One may notice that in a religion data set most of the distances for `associated` words slightly increased. Some of the distances from `different` class are still smaller than for the `associated` class. The distances for all classes seem in most cases to cluster together even after debiasing. The reason may be that uncertainty is quite high in `associated` and `different` class because the number of samples is so small. Additionally in some cases distances from `human` class are smaller than for the `assosiated` class which also makes it less clear how to interpret the results, if the distances are really able to catch the bias presence.

It is worth paying attention to the gender debiased dataset. The change is really minor, still zero out of HPDI range. The `different` attributes still have high similarity value. It seems that the distances for gender protected words are still clustered together which suggests that this method is not able to catch the complex bias presence. One could conclude that co-occurrence of certain terms does not always mean direct connection in terms of terms associations. The fact that some of stereotypically associated with female attributes still have high similarity with male protected words suggest that other techniques should be applied to detect and remove the bias.

At the same time, there is a small improvement in race dataset. In some cases the distances just cluster together. In general distances for `assosiated` class are still higher than those for the rest of classes. One may assume that there reason for so small change is that either the metric does not indicate properly biases or that the issue is more complex and subtracting subspaces is not enough.

In Manzini, Lim, Tsvetkov, & Black (2019) it is mentioned that debiasing with this method that is based on bias component removal may be insufficient. The reason is that words containing similar biases may still cluster together even after debiasing.

Chapter 6

Discussion

It is worth diving deeper into some observations that arise while analyzing the results.

- the distributions of cosine distances of whole population and individual words can vary enormously
- including uncertainty shows how in some cases it is not clear why a word is according to this method classified as anomalous
- debiasing with the method provided by Manzini, Lim, Tsvetkov, & Black (2019) does not solve the bias issues

One should remember that there is no baseline for the interpretation of the averaged cosine distances in (**Manzini2019blackToCriminalToCriminal?**). One may assume that if the cosine distance is close to 1 then it is a desired outcome as it means, according to cosine similarity assumptions, that there is almost no similarity between the words. However what does it mean to be close to 0? If the averaged cosine similarity is equal to 80, then should we still debiase it? It is unclear what the criteria are. On one hand, it seems to be beneficial when the outcome is simplified as it is easier to compare results with one value per set. On the other hand, it is prone to misunderstanding of how to interpret the results and what threshold to assume.

We propose the use of Bayesian method to introduce the uncertainty measure in bias detection.

Chapter 7

Summary

The aim of this paper was to show the analysis of bias detection methods used in modern natural language processing projects. In the further research it is worth to dive deeper into some issues or to examine new ideas that occurred during our work.

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

Islam, A. C., Bryson, J. J., & Narayanan, A. (2016). Semantics derived automatically from language corpora necessarily contain human biases. *CoRR*, *abs/1608.07187*. Retrieved from http://arxiv.org/abs/1608.07187

Manzini, T., Lim, Y. C., Tsvetkov, Y., & Black, A. W. (2019). Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. Retrieved from http://arxiv.org/abs/1904.04047

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. *CoRR*, *abs/1908.09635*. Retrieved from http://arxiv.org/abs/1908.09635

Nissim, M., Noord, R. van, & Goot, R. van der. (2019). Fair is better than sensational: Man is to doctor as woman is to doctor. *CoRR*, *abs/1905.09866*. Retrieved from http://arxiv.org/abs/1905.09866