Statistical Language Models

Week 10 wish

Debiasing Word Embeddings: Still more to do

- Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics
-

Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings But do not Remove Them

Hila Gonen[1] and **Yoav Goldberg**[1,2]
[1]Department of Computer Science, Bar-Ilan University
[2]Allen Institute for Artificial Intelligence
{hilagnn,yoav.goldberg}@gmail.com

Abstract

Word embeddings are widely used in NLP for a vast range of tasks. It was shown that word embeddings derived from text corpora reflect gender biases in society. This phenomenon is pervasive and consistent across different word embedding models, causing serious concern. Several recent works tackle this problem, and propose methods for significantly reducing this gender bias in word embeddings, demonstrating convincing results. However, we argue that this removal is superficial. While the bias is indeed substantially reduced according to the provided bias definition, the actual effect is mostly hiding the bias, not removing it. The gender bias infor-

swer the analogy "man is to computer programmer as woman is to x" with "x = homemaker". Caliskan et al. (2017) further demonstrate association between female/male names and groups of words stereotypically assigned to females/males (e.g. arts vs. science). In addition, they demonstrate that word embeddings reflect actual gender gaps in reality by showing the correlation between the gender association of occupation words and labor-force participation data.

Recently, some work has been done to reduce the gender bias in word embeddings, both as a post-processing step (Bolukbasi et al., 2016b) and as part of the training procedure (Zhao et al., 2018). Both works substantially reduce the bias

Normal Distribution

- Normal pdf, but writing the precision κ=1/σ^2
- $f_p(x,\mu,\kappa) = \left(\sqrt{\frac{\kappa}{(2\pi)}}\right)^p exp\left(-\kappa\frac{(x-\mu)^2}{2}\right)$ = $C_1 exp\left(-\kappa\frac{(x-\mu)^2}{2}\right)$
- If X values are on the unit sphere
- Then: $(x-\mu)^2 = x^2 + \mu^2 - 2\mu^T x = 2 - 2\mu^T x$
- $f_p(x,\mu,\kappa) = C_2 exp\left(\kappa x^T \mu\right)$

Von Mises - Fisher Distribution

- Von Mises Distribution: PDF on the circle
- Von Mises - Fisher Distribution: generalizes to the (p-1) sphere in $\mathbb{R}^p$
- Generally written as $f_p(x, \mu, \kappa) = C_p(\kappa) exp(\kappa \mu^T x)$
- $f_p(x, \mu, \kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{p/2-1}(\kappa)}(\kappa) exp(\kappa \mu^T x)$
- Where $I_{p/2-1}$ is the modified Bessel function of the first kind at order (p/2-1)
- Parameters: μ = mean direction and κ = concentration around the mean

MLE of von Mises-Fisher

- Log likelihood:
- $$\ell(X \mid \mu, \kappa) = log \left[\frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{p/2-1}(\kappa)} \right] + \sum_{i=1}^{N} \kappa \mu x_i$$
- where: $|\,|\,\mu\,|\,| = 1, \kappa \geq 0$
- Can’t make μ infinite, so maximize by emphasizing the most important directions.

MLEs of μ κ

- $\hat{\mu} = \dfrac{\sum_{i=1}^{N} x_i}{||\sum_{i=1}^{N} x_i||}$ = vector sum of all x ÷ length of that sum vector
- Define the average vector: $\bar{r} = \dfrac{||\sum_i x_i||}{n}$
- Then $\hat{\kappa} = A_p^{-1}(\bar{r})$ where $A_p(\kappa) = \dfrac{I_{p/2}(\kappa)}{I_{p/2-1}(\kappa)} = \bar{r}$
- For large p
- $\hat{\kappa} \approx \dfrac{\bar{r}(p - \bar{r}^2)}{1 - \bar{r}^2}$ when p is large

Watson & Williams (1956) "On the construction of significance tests on the circle and the sphere" Biometrika 43:344-352

ON THE CONSTRUCTION OF SIGNIFICANCE TESTS ON THE CIRCLE AND THE SPHERE

By G. S. WATSON* and E. J. WILLIAMS†

1. Introduction

A number of recent papers have dealt with the probability density, in two and three dimensions, proportional to

$$\exp(\kappa \cos\theta),$$

where κ is a precision constant, and θ is the angle between an observed unit vector and the population mean unit vector or polar vector. The purposes of these investigations have been (i) to derive, from observed results, limits within which the unknown polar vector is likely to lie, and (ii) to test the homogeneity of different sets of observations, both in their precision and in their direction. These distributions and the tests associated with them thus produce the circular and spherical analogues of the usual tests in Euclidean space, based on the normal distribution.

Hypothesis testing on the p dimensional sphere

- ANOVA style: $H_o: \mu_1 = \mu_2 = \ldots = \mu_k$ vs $H_a: \mu_i \neq \mu_j$ for at least one pair (i,j) of observations
- For k populations with sample sizes $n_1, n_2, \ldots, n_k$ where $n = n_1 + n_2 + \ldots + n_k$
- Resultant length: $R_i = \| \sum_{j=1}^{n_i} x_{ij} \|$
- Total Resultant length: $R = \| \sum_{i=1}^{k} \sum_{j=1}^{n_i} x_{ij} \|$
- $W = \dfrac{(n-k)\left(\sum_{i=1}^{k} R_i - R\right)}{(k-1)(n - \sum_{i=1}^{k} R_i)} \sim F_{(k-1)(p-1),(n-k)(p-1)}$

- $$W = \frac{(n-k)\left(\sum_{i=1}^{k} R_i - R\right)}{(k-1)(n-\sum_{i=1}^{k} R_i)} \sim F_{(k-1)(p-1),(n-k)(p-1)}$$

- Usual ANOVA assumptions apply: common (unknown) concentration parameter

More variations

- Jammalamadaka, S. Rao., and Ambar Sengupta. (2001) "Topics in Circular Statistics". River Edge, N.J: World Scientific
- https://ocul-crl.primo.exlibrisgroup.com/permalink/01OCUL_CRL/1gorbd6/alma991022668573505153
- See Chapter 5 for an excellent overview of the different test options

- #non-equal concentration parameters likelihood ratio test for 2 or more groups

- het.aov(x, ina) # x has unit vectors, ina has indicator of groupings

In R; set up

- library(ggplot2)
- library(tidyverse)
- library(tidytext)
- library(wordVectors)
- **library(gutenbergr)**
- **library(Directional)**
- model = read.vectors("GoogleNews-vectors-negative300.bin")

Do these books start in the same way?

- Jane Austen "Sense and Sensibility"
- Jane Austen "Persuasion"

- Convert Chapter 1 into vectors. Compare the means

In R; set up

- SS = gutenberg_download(gutenberg_works(str_detect(title, fixed("Sense and Sensibility", ignore_case=TRUE)))) %>%
- mutate(chapter = cumsum(str_detect(text, regex("^chapter [\\divxlc]", ignore_case = TRUE)))) %>%
- unnest_tokens(word, text, token = "words") %>%
- anti_join(stop_words,by="word")

- PE = gutenberg_download(gutenberg_works(str_detect(title, fixed("Persuasion", ignore_case=TRUE)))) %>%
- mutate(chapter = cumsum(str_detect(text, regex("^chapter [\\divxlc]", ignore_case = TRUE)))) %>%
- unnest_tokens(word, text, token = "words") %>%
- anti_join(stop_words,by="word")

- #Extract just Chapter 1 of each:
- SS1 = SS%>% filter(chapter ==1)
- PE1 = PE%>% filter(chapter ==1)

- modelNormed = normalize_lengths(model)
- #Extracting the vectors from the word2Vec model
- modelNormed
- modelNormed[c("in","that","for","in"),]
- #Remove missing words
- Allwordsinbooks = unique(SS1$word,PE1$word)
- AllInUse = intersect(Allwordsinbooks,rownames(modelNormed))

Word vector matrices

- SS1vec = modelNormed[intersect(SS1$word,AllInUse),]
- PE1vec = modelNormed[intersect(PE1$word,AllInUse),]
- Bookname = **factor**(c(rep("SS",dim(SS1vec)[1]), rep("PE",dim(PE1vec)[1])))
- het.aov(rbind(SS1vec,PE1vec), Bookname)