Big Data Science - Spring 2018
Assignment 4

Daniel Rivera Ruiz
Department of Computer Science
New York University
drr342@nyu.edu

1 Similar words

Table 1 shows the top 10 most similar words to the ones given in the assignment:

Table 1

Word w	10 most similar words to w
cat	cat, mouse, squirrel, toad, hen, rabbit, wolf, rat, kitten, weasel
animal	animal, animals, vegetable, organism, reptile, insect, creature, ox, organic, deer
science	science, sciences, chemistry, physiology, physics, scientific, theology, astronomy, geology, philosophy
scientific	scientific, philosophical, chemistry, biological, physiology, science, physics, ethical, deductive, theoretical
vector	-
vendor	vendor, annum, quart, raisins, fourpence, gallon, fried, canned, stewed, a-year
car	car, cars, omnibus, cab, garage, buggy, motor, hansom, wagon, bus
hot	hot, cold, warm, fried, boiled, soda, weather, rain, cool, hotter
major	major, colonel, miss, captain, mrs, dr, lady, mr, sir, doctor
man	man, woman, gentleman, lady, feller, men, fellow, girl, ses, said
doctor	doctor, miss, dr, colonel, captain, major, yes, answered, letter, noel
flower	flower, blossoms, blossom, bloom, flowering, flowers, buds, violets, petals, lilac
capital	capital, annum, cent, income, manufacturing, exports, ayres, insurance, levied, dollars
washington	washington, sherman, aug, th, major-general, appleton, howe, congress, governor, feb

Observations on the results for similar words:

- The word *vector* was not found in our *word2vec* model and therefore no similar words were returned. This makes sense because 1) the training corpus consists mostly of literary works (novels) where the word *vector* is probably not very common and 2) the minimal word frequency threshold was set to 15, so even if *vector* did appear somewhere in the training corpus, it wouldn't have made it to the model unless it appeared 15 times or more.
- The list of similar words for *capital* suggests that it is used in the training corpus more frequently in the sense of money rather than a city.
- The last words in the lists for *man* and *doctor* don't seem to be very related to the original words, which probably means that their similarity scores are not as good as those of the more highly ranked words.
- There doesn't seem to be a clear pattern to the similar words for *washington*, which probably means that the word appeared in several different contexts throughout the training corpus.

2 Word analogies

Table 2 shows the top 5 values of x that better complete the analogies provided in the assignment:

Table 2

Analogy	Top 5 values of x
king : queen :: x : woman	man, gentleman, fellow, person, boy
paris : france :: x : italy	florence, venice, rome, london, munich
car : cars :: x : birds	bird, nest, cuckoos, twittering, kingfisher
man : woman :: x : nurse	steward, surgeon, housekeeper, physician, mate
fall : stand :: x : live	fell, fallen, meet, return, sink
important : unimportant :: x : unwilling	obliged, anxious, willing, determined, reluctant

Observations on the results for similar words:

- Rome does not appear as the first suggestion in the second analogy, which is understandable considering that the model has no way of knowing the capital cities of the countries and it relies solely in probabilities.
- The fourth analogy is clearly biased by gender, which probably means that the authors of the texts in the training corpus are too, since the model is only learning things as it sees them.
- There doesn't seem to be a clear pattern for the fifth analogy, which probably means that there is not enough information in the training corpus regarding this specific set of words.

3 Embeddings visualization

Figure 1 shows the cluster surrounding the word *bird*.
Figure 2 shows the cluster surrounding the word *brother*.
Figure 3 shows the cluster surrounding the word *cheek*.
Figure 4 shows the cluster surrounding the word *happy*.
Figure 5 shows the cluster surrounding the word *music*.
Figure 6 shows the cluster surrounding the word *salad*.
Figure 7 shows the cluster surrounding the word *president*.

4 Debiasing algorithms

The debiasing algorithms proposed in Bolukbasi et al. (2016) are used to neutralize a set of words that are likely to be biased in different contexts such as gender, race or religion. As described in the paper, it is easier to define the set of words to neutralize N as the complement set $W \setminus S$, where W

is the set of all words in the vocabulary and S is the set of gender specific words that must not be neutralized, e.g. *mother* and *father* or *nun* and *priest*. The reason behind this is that the cardinality of S is usually much smaller than that of N.

The first algorithm proposed is called *Hard debiasing* or *Neutralize and Equalize* which can be summarize in the following steps:

1) *Identify gender subspace.* In this step a matrix C is built using the bias-defining sets $D_1, D_2, \ldots, D_n$ along with the embeddings of all the words in each set. SVD is applied to this matrix to select the k first (most significant) rows as the bias subspace B.
2) *Neutralize and Equalize*. In the second step the embeddings of the words in N are neutralized by projecting them onto the orthogonal subspace of B and equalized by centering them along predefined equality sets $E_1, E_2, \ldots, E_m$ and normalizing them.

The second algorithm proposed is called *Soften* and can be described as follows:

1) *Identify gender subspace.* Exactly the same as for *Hard debiasing*.
2) *Soft bias correction*. In this case, the desired debiasing is defined as a linear transformation T that seeks to preserve pairwise inner products between all the word vectors while minimizing the projection of the neutral words onto B. To achieve this, a tuning parameter λ is used to balance the objective of preserving the original embedding inner products with the goal of reducing bias.

In the first algorithm, *neutralize* ensures that neutral words are zero in the bias subspace and *equalize* perfectly equalizes sets of words outside the subspace and thereby enforces the property that any neutral word is equidistant to all words in each equality set.
The disadvantage of *equalize* is that it removes certain distinctions that are valuable in some applications, in which case the *soften* algorithm can be used. *Soften* reduces the differences between equality sets while maintaining as much similarity to the original embedding as possible, with a parameter (λ) that controls this trade-off.

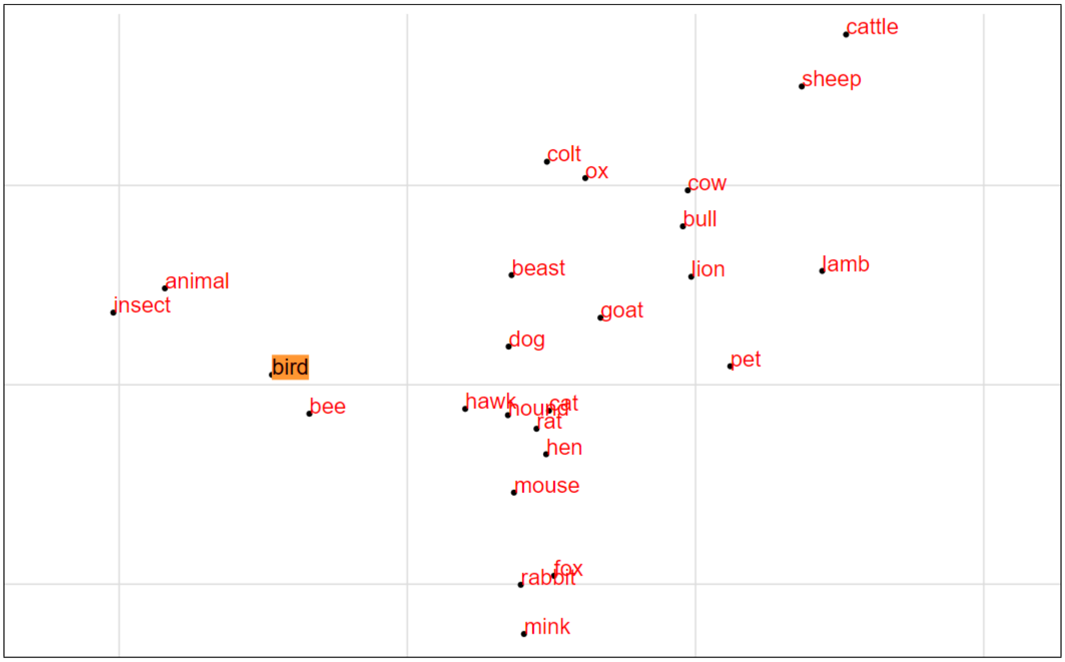

Figure 1

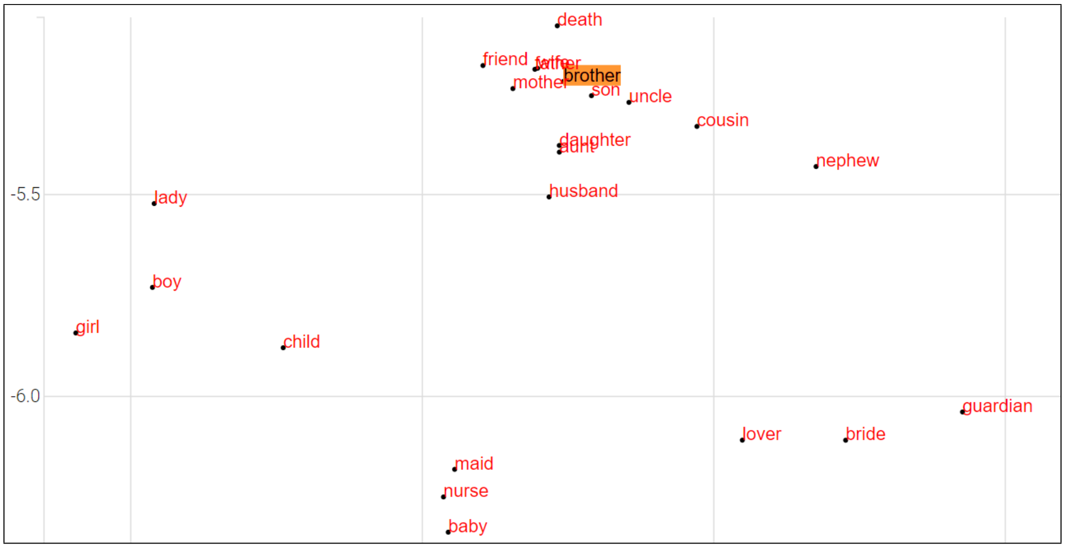

Figure 2

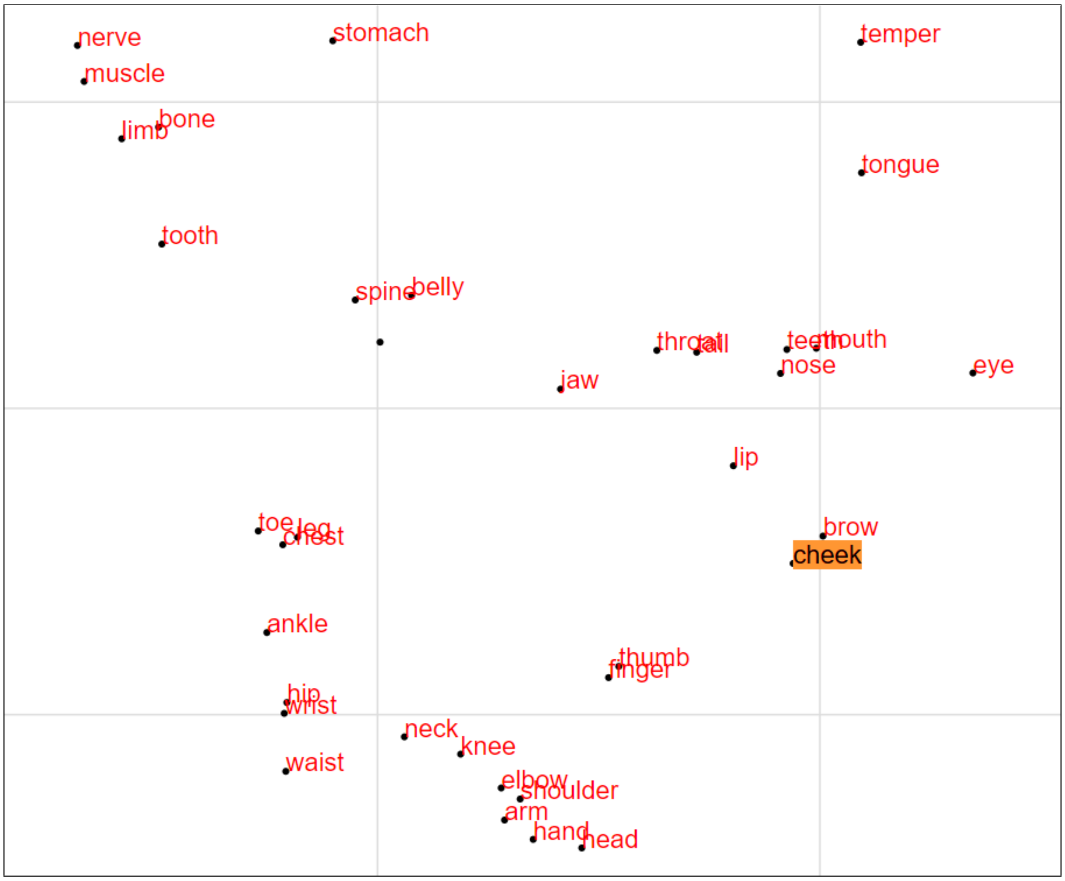

Figure 3

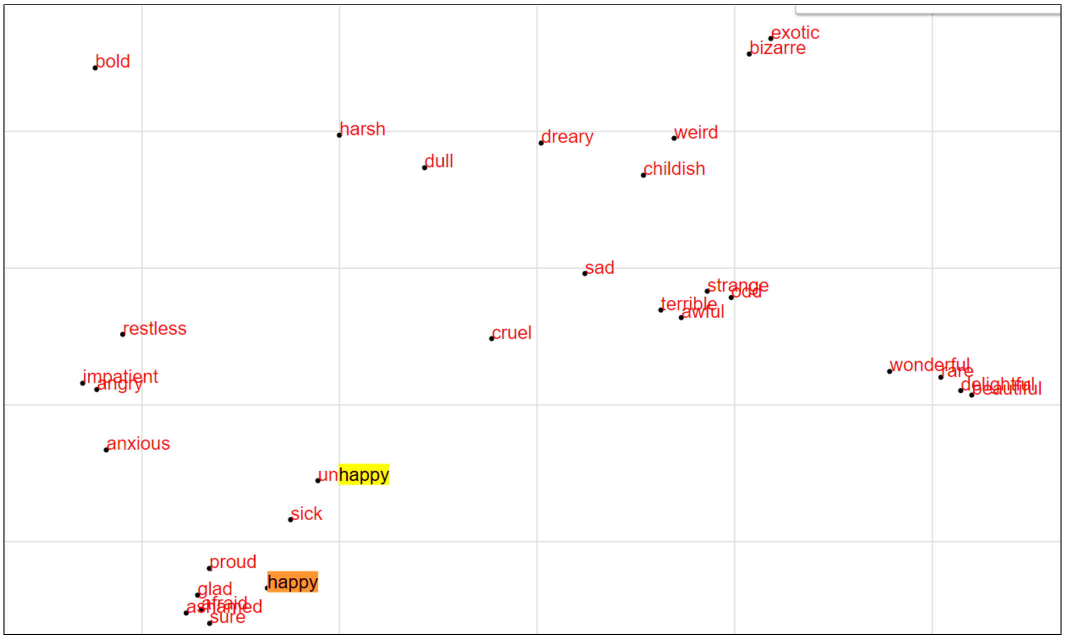

Figure 4

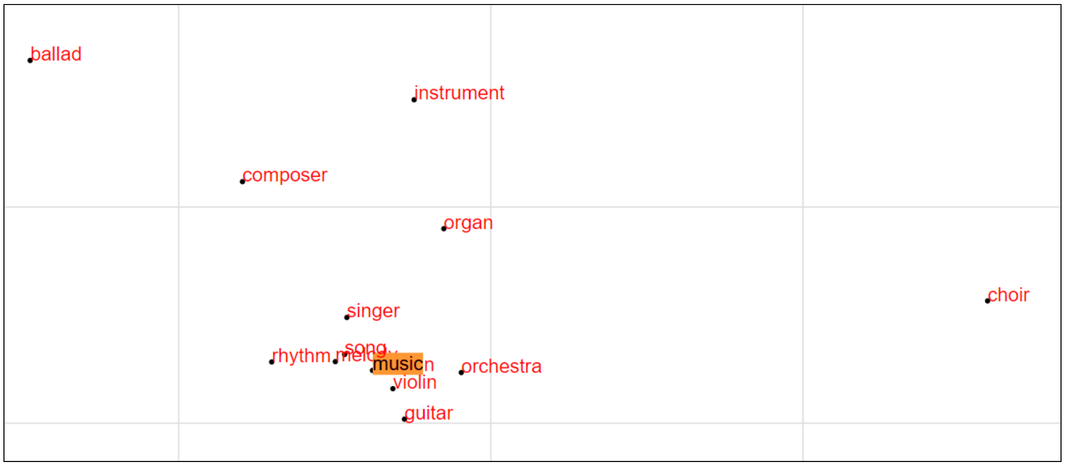

Figure 5

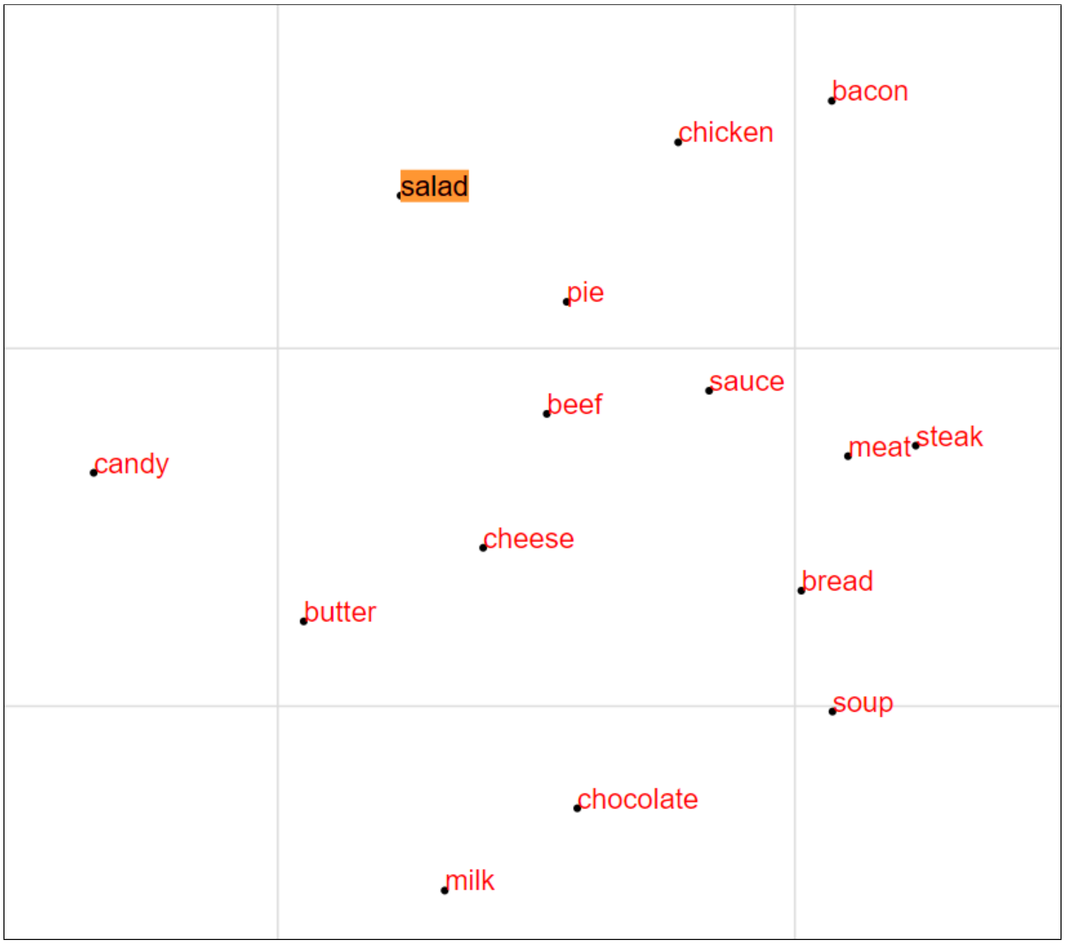
bacon
chicken
salad
pie
sauce
beef
meat
steak
candy
cheese
bread
butter
soup
chocolate
milk

Figure 6

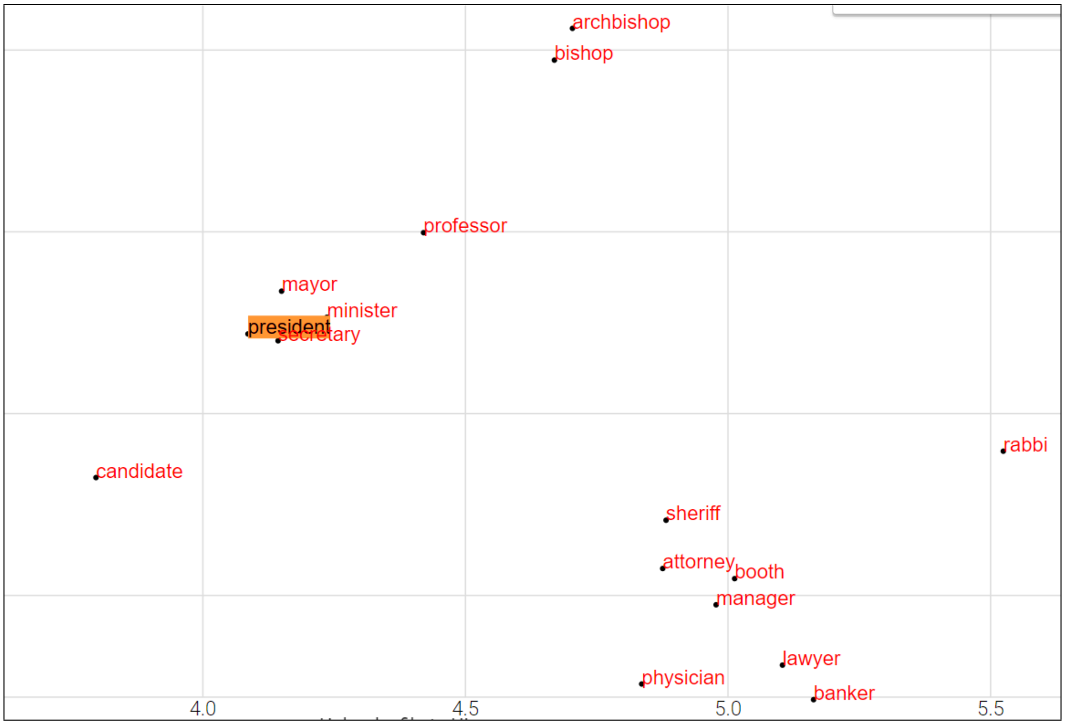
archbishop
bishop
professor
mayor
minister
president
secretary
candidate
rabbi
sheriff
attorney
booth
manager
lawyer
physician
banker
4.0
4.5
5.0
5.5

Figure 7