ΠΡΟΑΙΡΕΤΙΚΗ ΕΡΓΑΣΙΑ ΣΤΟ ΜΑΘΗΜΑ ΜΥΕ003: ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ, ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2022

A STUDY OF BIAS IN WORD EMBEDDINGS

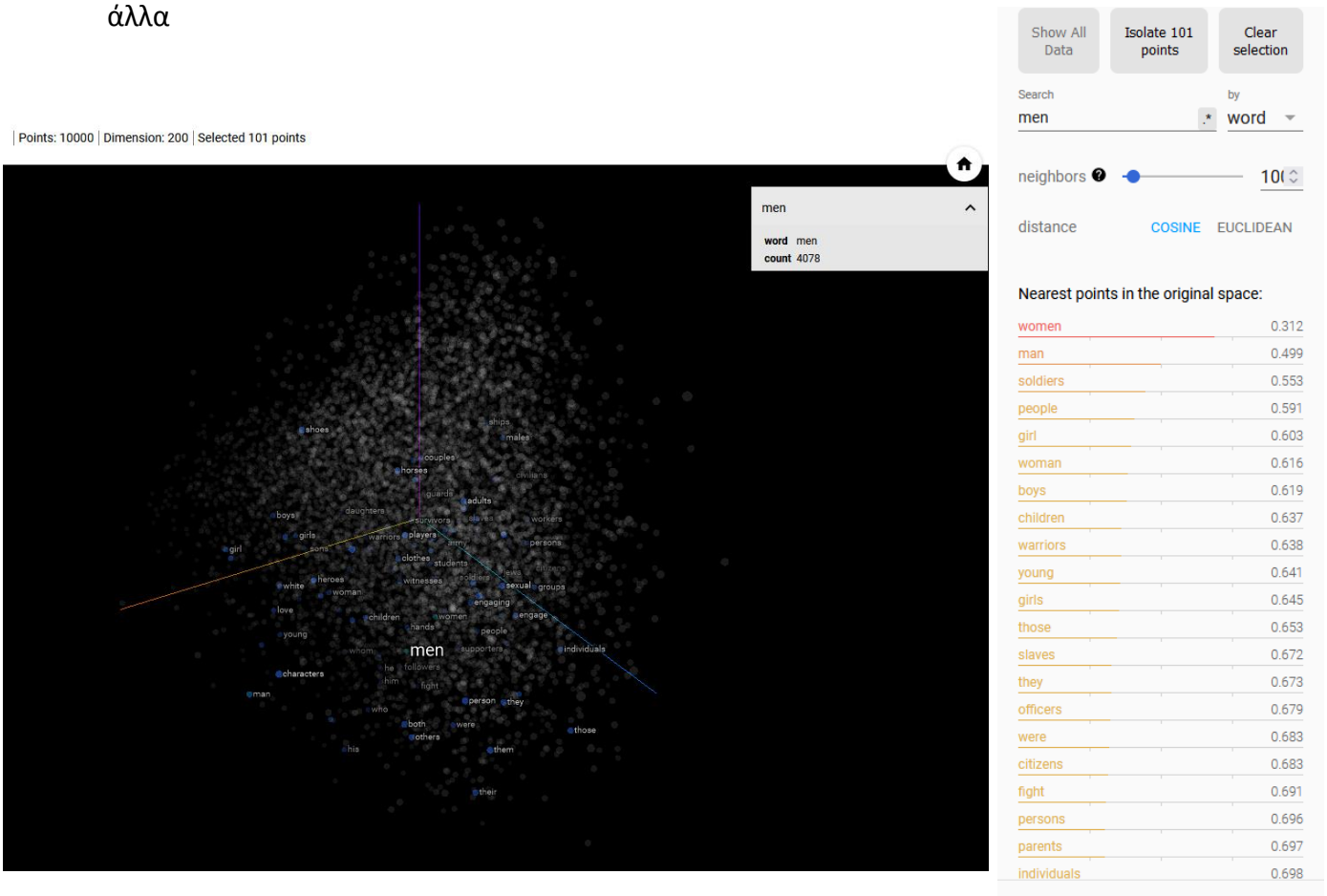
Σωτήρης Παναγιώτου, ΑΜ: 4456

Νικολέττα Μπιντζηλαίου, ΑΜ: 4435

1) Χρησιμοποιώντας τον embedding Projector κάναμε τις αναζητήσεις για τις λέξεις men, women, professionals

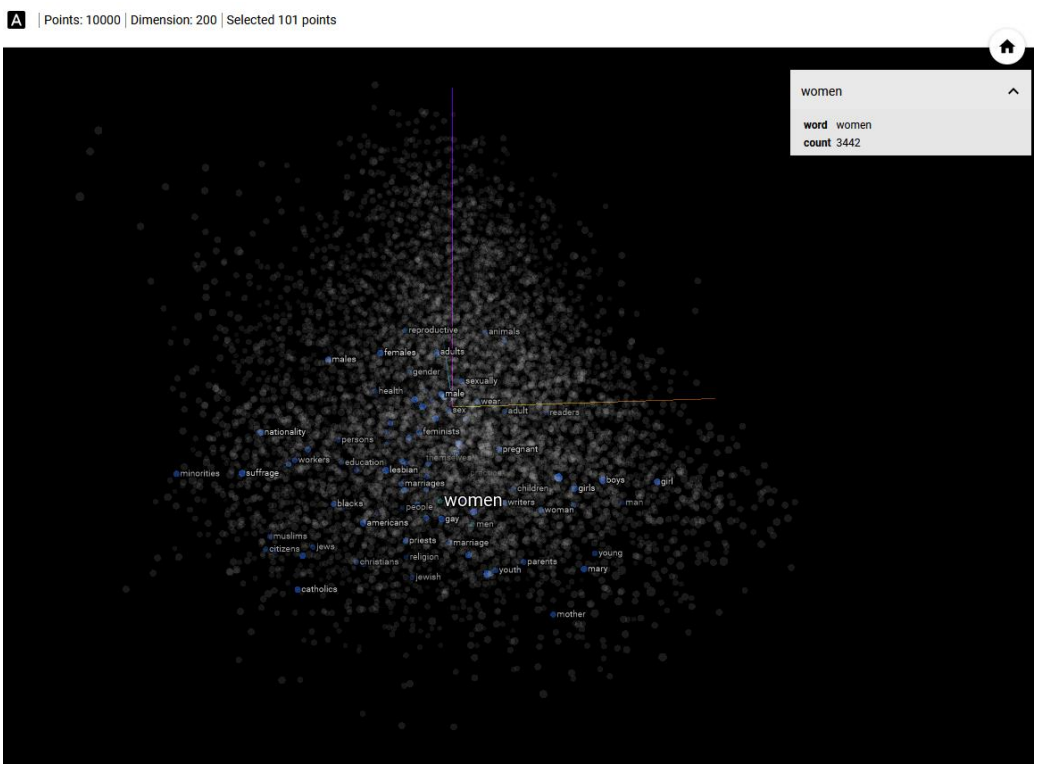
Για τη λέξη ‘men’ παρατηρήσαμε ότι σαν γειτονικά σημεία έχει τις λέξεις:

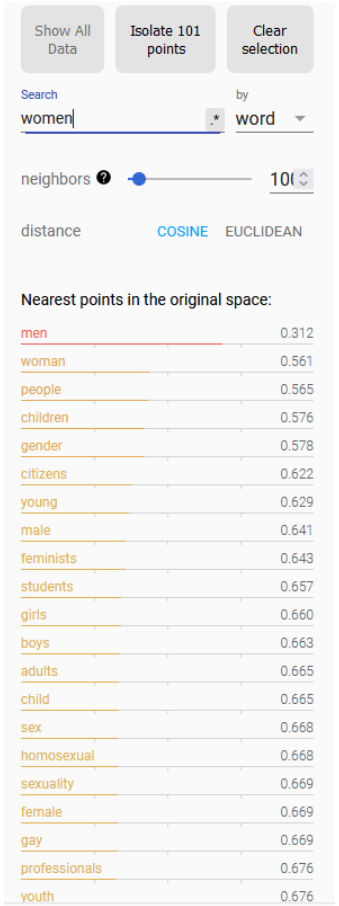
women, man, soldiers, people,…, children, young, warriors, officers, citizens και άλλα



Αντίστοιχα, για τη λέξη ‘women’ παρατηρήσαμε ότι σαν γειτονικές έχει τις λέξεις:

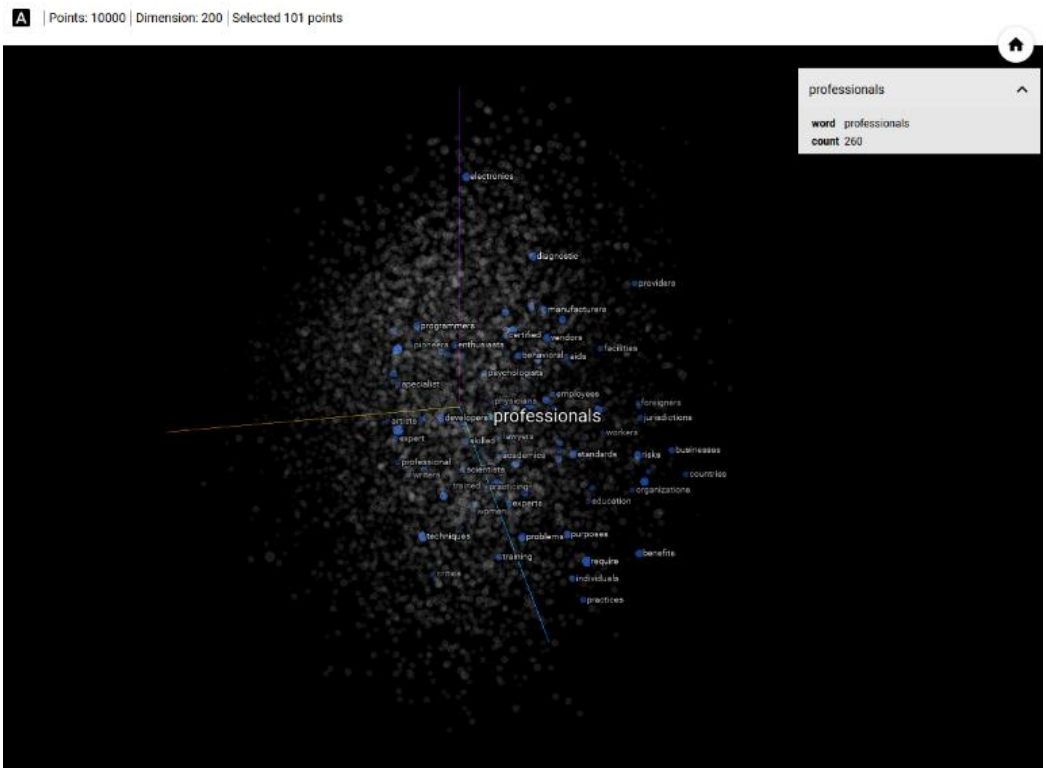
men, woman, people, children ,…,feminists, sex, sexuality και άλλα.





Για τη λέξη ‘professionals’ παρατηρήσαμε ότι σαν γειτονικές έχει τις λέξεις:

experts, standards, academics,…, physicians, women, programmers κτλπ.



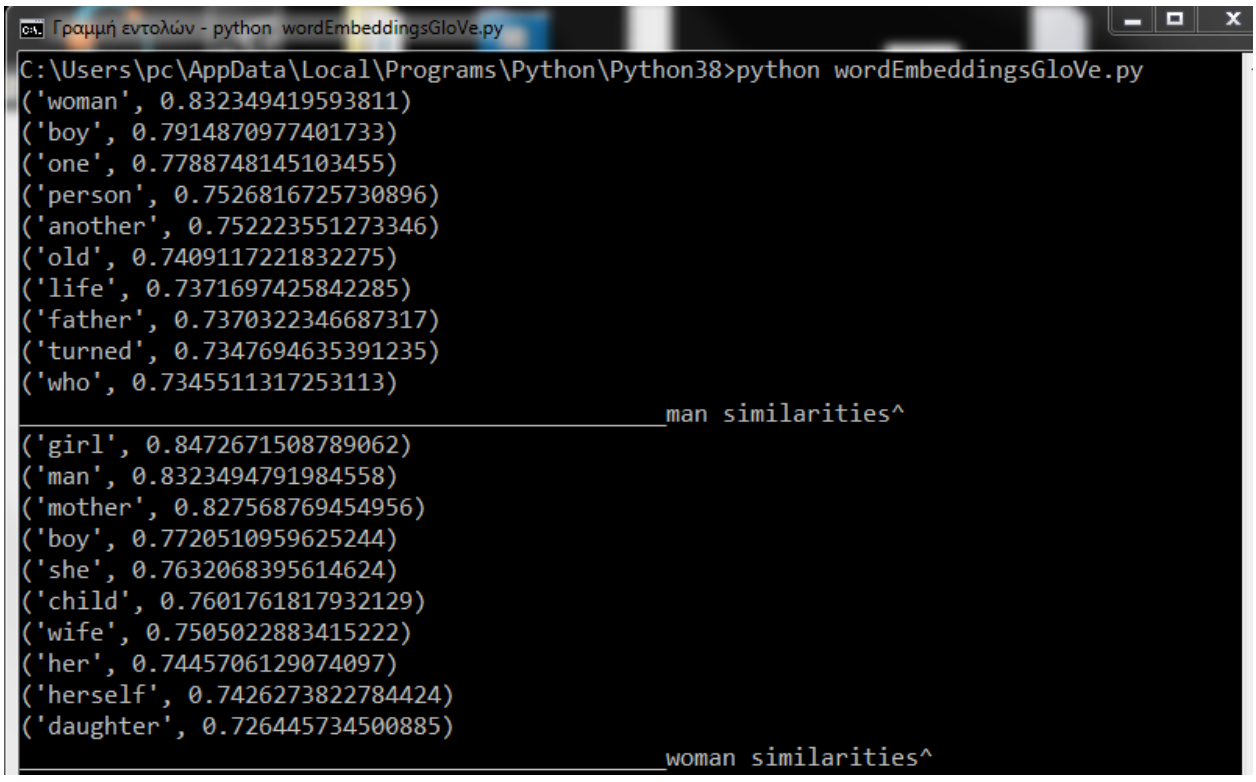
2) Στην συνέχεια, ψάξαμε pre-trained word embeddings που να μπορούμε να χρησιμοποιήσουμε σε project online. Η αναζήτηση μας μας έφερε στον GloVe: Global Vectors for Word Representation, των Jeffrey Pennington, Richard Socher, Christopher D. Manning, από το Stanford, 2014. [1]

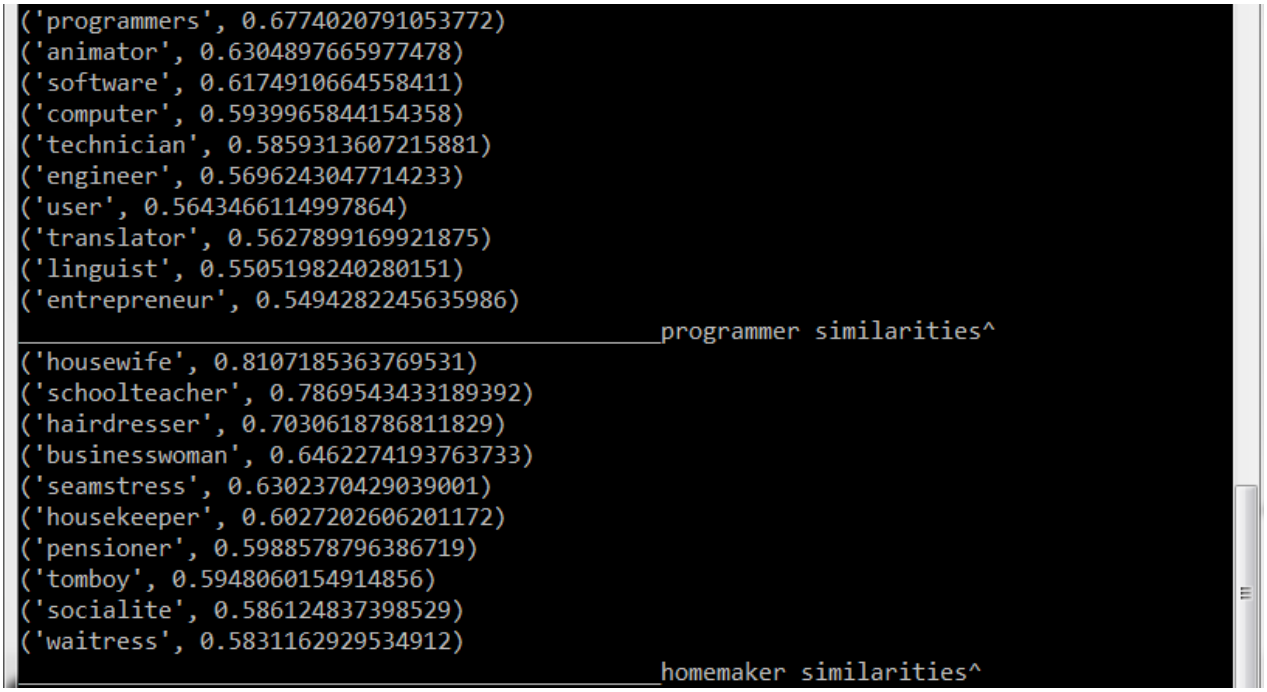
Αναρωτηθήκαμε λοιπόν πως θα χρησιμοποιήσουμε και θα χειριστούμε αυτά τα embeddings. Λόγω έλλειψης χρόνου, δεν φτιάξαμε νέο project from scratch, αλλά πήραμε έμπνευση όπως και βασιστήκαμε σε κάποια ήδη δημοσιευμένα που υπάρχουν στο διαδίκτυο.

A. Με αφορμή τον κώδικα του Normalized Nerd [3], γράψαμε την παρακάτω μικρή εφαρμογή:

```
import numpy as np
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt


from gensim.scripts.glove2word2vec import glove2word2vec
glove_input_file = 'glove.6B.100d.txt'


from gensim.models import KeyedVectors
model = KeyedVectors.load_word2vec_format(glove_input_file, binary=False, no_header=True)

resultMan=model.most_similar("man")
for i in resultMan:
       print(i)
print("_____________________________________________man similarities^")

resultWoman=model.most_similar("woman")
for e in resultWoman:
       print(e)
print("_____________________________________________woman similarities^")

resultProg=model.most_similar("programmer")
for x in resultProg:
       print(x)
print("_____________________________________________programmer similarities^")

resultHome=model.most_similar("homemaker")
for y in resultHome:
       print(y)
print("_____________________________________________homemaker similarities^")

vocab = ["boy", "mother", "man", "woman", "father", "child", "businessman","businesswoman",
"wife",
"teacher","programmer","homemaker","ceo","lawyer","doctor","husband","president","beautiful","ha
rdworking","whining"]

def tsne_plot(model):
    labels = []
    wordvecs = []

    for word in vocab:
```

```
        wordvecs.append(model[word])
        labels.append(word)

    tsne_model = TSNE(perplexity=3, n_components=2, init='pca', random_state=42)
    coordinates = tsne_model.fit_transform(wordvecs)

    x = []
    y = []
    for value in coordinates:
        x.append(value[0])
        y.append(value[1])

    plt.figure(figsize=(8,8))
    for i in range(len(x)):
        plt.scatter(x[i],y[i])
        plt.annotate(labels[i],
                     xy=(x[i], y[i]),
                     xytext=(2, 2),
                     textcoords='offset points',
                     ha='right',
                     va='bottom')
    plt.show()

tsne_plot(model)
```





Θέλουμε να δούμε τα πιο κοντινά embeddings των λέξεων man, woman, programmer και homemaker.

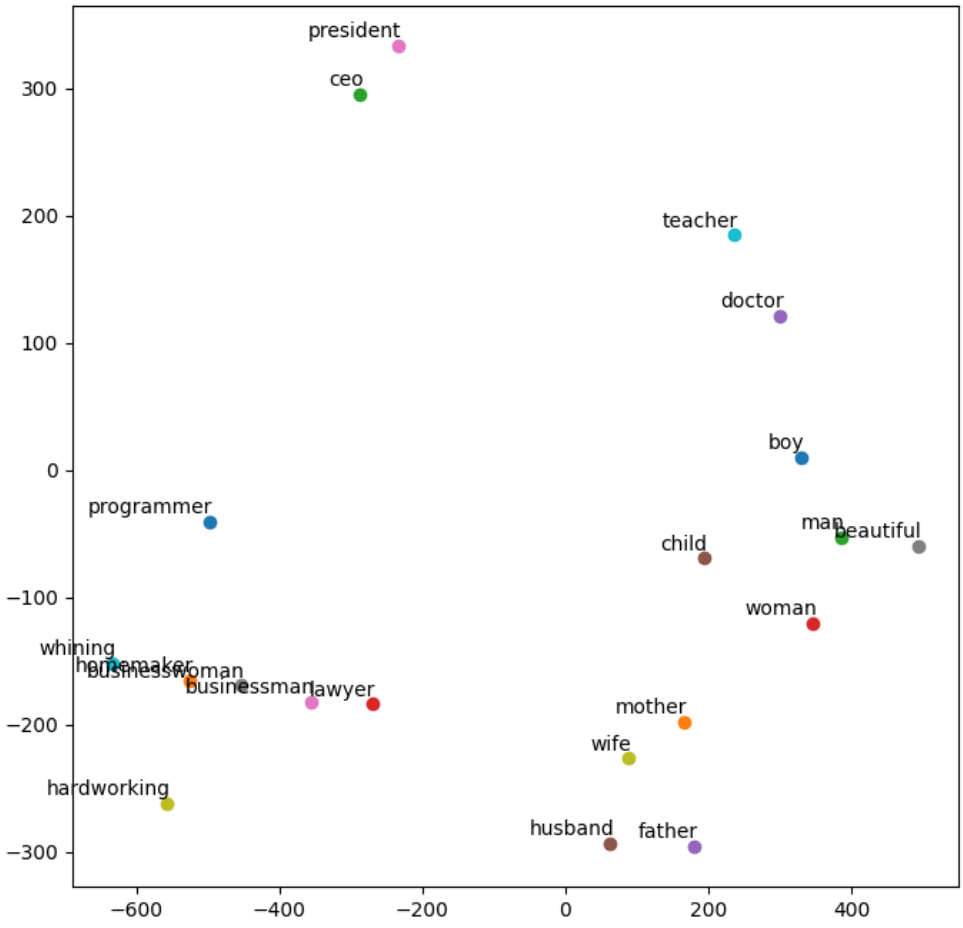
Για τη λέξη man, παρατηρούμε embeddings λέξεων που έχουν να κάνουν με τη ζωή γενικά (boy, person, old, life) αλλά και κάποια για την οικογένεια (father).

Για τη λέξη woman, παρατηρούμε embeddings λέξεων που έχουν να κάνουν κυρίως με την οικογένεια (mother, wife, child, daughter).

Επίσης, για τη λέξη programmer, παρατηρούμε embeddings λέξεων που έχουν να κάνουν κυρίως με άλλα τεχνικά επαγγέλματα, χωρίς να υπάρχει κάποια ιδιαίτερη ένδειξη για το φύλο (κατάληξη -ter).

Αντίθετα, για τη λέξη homemaker, παρατηρούμε embeddings λέξεων που έχουν να κάνουν κυρίως με «γυναικεία» επαγγέλματα (κατάληξη –tress). Ιδιαίτερο ενδιαφέρον έχει η 2^{η} λέξη housewife, που βγαίνει σαν κοντινότερη, σε αντίθεση με τη housekeeper, που είναι 6^{η} και η 4^{η}, businesswoman.

Θέλαμε να δούμε στο plot αν μπορούμε να βρούμε κάποια εμφανή διαφορά. Βλέπουμε ότι το mother, wife είναι πιο κοντά στο woman , από ότι το husband, father στο man. Επίσης, βλέπουμε ότι το whining και το housemaker είναι πολύ κοντά στο businesswoman και μετά στο businessman.



B. Πειραματιστήκαμε πάνω στον κώδικα των IG Tech Team [2].

```
import numpy as np
import pandas as pd
import os
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt


def read_data(file_name):
    with open(file_name,'r', encoding='utf-8') as f:
        word_vocab = set()
        word2vector = dict()
        for line in f:
            line = line.strip() #Remove white space
            words_Vec = line.split()
            word_vocab.add(words_Vec[0])
            word2vector[words_Vec[0]] = np.array(words_Vec[1:],dtype=float)
    print("Total Words in DataSet:",len(word_vocab))
    return word_vocab,word2vector

vocab, w2v = read_data("glove.6B.100d.txt")

# Cosine-similarity
```

```
def cos_sim(u,v):
    """
    u: vector of 1st word
    v: vector of 2nd Word
    """
    numerator = np.dot(u,v)
    denominator= np.sqrt(np.sum(np.square(u))) * np.sqrt(np.sum(np.square(v)))
    temp= numerator/denominator
    if temp>=0.5:
           print('Similarity score is greater than 0.5, words are similar!')
    return numerator/denominator

print("Similarity Score of King and Queen",cos_sim(w2v['king'],w2v['queen']))
print("Similarity Score of Man and Woman",cos_sim(w2v['man'],w2v['woman']))
print('----------------------------------------------------------')
print("Similarity Score of man and businessman",cos_sim(w2v['man'],w2v['businessman']))
print("Similarity Score of woman and businessman",cos_sim(w2v['woman'],w2v['businessman']))
print('----------------------------------------------------------')
print("Similarity Score of man and nurse",cos_sim(w2v['man'],w2v['nurse']))
print("Similarity Score of woman and nurse",cos_sim(w2v['woman'],w2v['nurse']))
print('----------------------------------------------------------')
print("Similarity Score of man and teacher",cos_sim(w2v['man'],w2v['teacher']))
print("Similarity Score of woman and teacher",cos_sim(w2v['woman'],w2v['teacher']))
print('----------------------------------------------------------')
print("Similarity Score of man and programmer",cos_sim(w2v['man'],w2v['programmer']))
print("Similarity Score of woman and programmer",cos_sim(w2v['woman'],w2v['programmer']))
print('----------------------------------------------------------')
print("Similarity Score of man and homemaker",cos_sim(w2v['man'],w2v['homemaker']))
print("Similarity Score of woman and homemaker",cos_sim(w2v['woman'],w2v['homemaker']))
print('----------------------------------------------------------')
print("Similarity Score of man and ceo",cos_sim(w2v['man'],w2v['ceo']))
print("Similarity Score of woman and ceo",cos_sim(w2v['woman'],w2v['ceo']))
print('----------------------------------------------------------')
print("Similarity Score of man and doctor",cos_sim(w2v['man'],w2v['doctor']))
print("Similarity Score of woman and doctor",cos_sim(w2v['woman'],w2v['doctor']))
print('----------------------------------------------------------')
print("Similarity Score of man and lawyer",cos_sim(w2v['man'],w2v['lawyer']))
print("Similarity Score of woman and lawyer",cos_sim(w2v['woman'],w2v['lawyer']))
print('----------------------------------------------------------')
print("Similarity Score of man and president",cos_sim(w2v['man'],w2v['president']))
print("Similarity Score of woman and president",cos_sim(w2v['woman'],w2v['president']))
print('----------------------------------------------------------')
print("Similarity Score of man and beautiful",cos_sim(w2v['man'],w2v['beautiful']))
print("Similarity Score of woman and beautiful",cos_sim(w2v['woman'],w2v['beautiful']))
print('----------------------------------------------------------')
print("Similarity Score of man and hardworking",cos_sim(w2v['man'],w2v['hardworking']))
print("Similarity Score of woman and hardworking",cos_sim(w2v['woman'],w2v['hardworking']))
print('----------------------------------------------------------')
print("Similarity Score of man and whining",cos_sim(w2v['man'],w2v['whining']))
print("Similarity Score of woman and whining",cos_sim(w2v['woman'],w2v['whining']))
```

Στις δοκιμές μας παρατηρήσαμε τα εξής:

```
= RESTART: C:\Users\pc\AppData\Local\Programs\Python\Python38\wordEmbeddings.py
Total Words in DataSet: 400000
Similarity score is greater than 0.5, words are similar!
Similarity Score of King and Queen 0.7507690793623849
Similarity score is greater than 0.5, words are similar!
Similarity Score of Man and Woman 0.8323494204818741
----------------------------------------------------------
Similarity score is greater than 0.5, words are similar!
Similarity Score of man and businessman 0.523272963448835
Similarity Score of woman and businessman 0.47550381002390957
----------------------------------------------------------
Similarity Score of man and nurse 0.456238782531558
Similarity score is greater than 0.5, words are similar!
Similarity Score of woman and nurse 0.6139442265853785
----------------------------------------------------------
Similarity score is greater than 0.5, words are similar!
Similarity Score of man and teacher 0.5256064982738996
Similarity score is greater than 0.5, words are similar!
Similarity Score of woman and teacher 0.5791158655094495
----------------------------------------------------------
Similarity Score of man and programmer 0.2648719166376066
Similarity Score of woman and programmer 0.20142349417873726
----------------------------------------------------------
Similarity Score of man and homemaker 0.23557389792162103
Similarity Score of woman and homemaker 0.4257913224540586
----------------------------------------------------------
Similarity Score of man and ceo 0.30490661043585304
Similarity Score of woman and ceo 0.20037556477242932
----------------------------------------------------------
```

Οι λέξεις man και businessman είναι πιο κοντά από ότι οι λέξεις woman, businessman, όπως είναι λογικό αφού η μία περιέχεται μέσα στην άλλη.

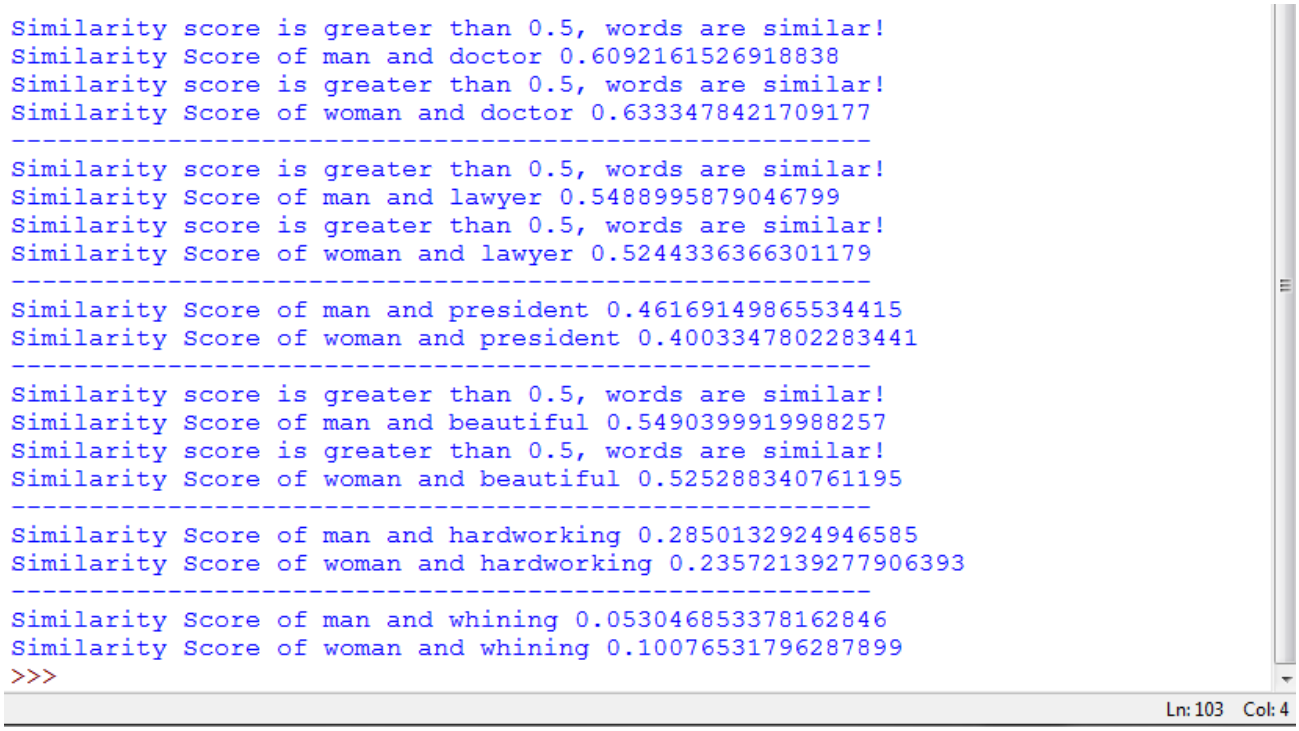
Οι λέξεις ,όμως, man και nurse είναι αρκετά μακριά σε σχέση με τις λέξεις woman και nurse. Μάλιστα, παρατηρούμε μία διαφορά περίπου 0.16.

Οι λέξεις man, teacher και woman,teacher είναι σχεδόν εξίσου κοντά, αν και το δεύτερο ζευγάρι υπερισχύει κατά 0.05.

Οι λέξεις man, programmer και woman, programmer είναι σχεδόν εξίσου μακριά, αν και το πρώτο ζευγάρι υπερισχύει κατά 0.06.

Ενδιαφέρον έχουν οι λέξεις man, homemaker & woman, homemaker καθώς αν κανένα ζευγάρι δεν είναι ιδιαίτερα κοντά (κάτω από 0.5), παρατηρούμε αισθητή διαφορά κατά περίπου 0.2 στο woman, homemaker.

Αντίστοιχα, οι λέξεις man , ceo & woman, ceo, δεν είναι κοντά αλλά το πρώτο ζευγάρι υπερισχύει κατά 0.1.



Συνεχίζοντας, δοκιμάσαμε και άλλα ζευγάρια με επαγγέλματα αλλά και επίθετα, χαρακτηρισμούς.

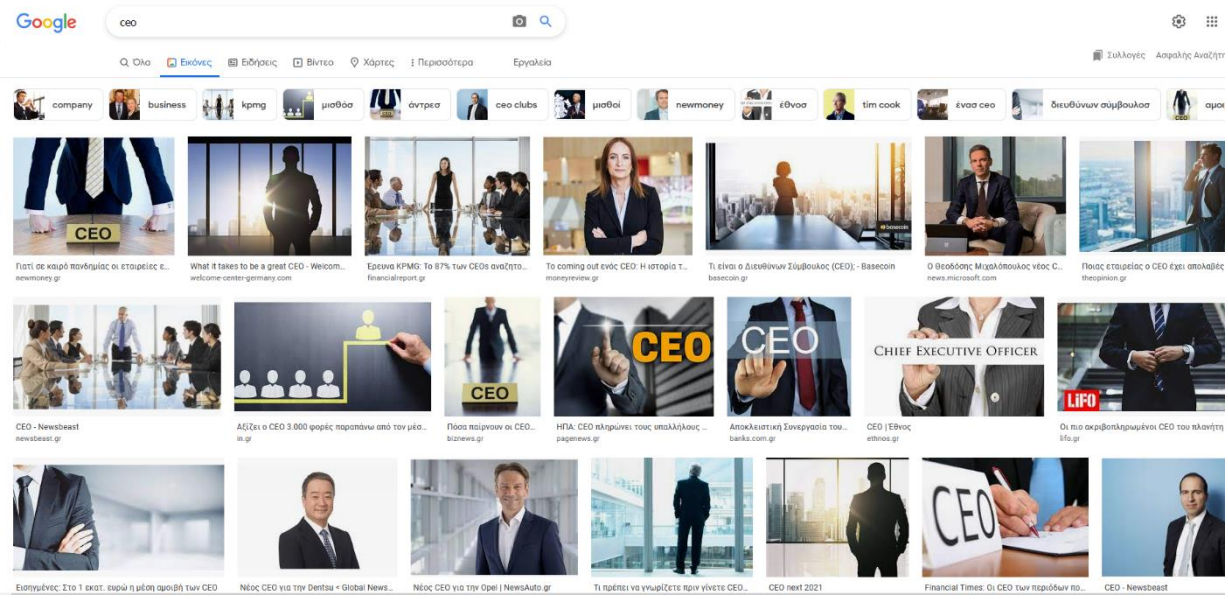
Οι λέξεις man και doctor, woman και doctor είναι αρκετά κοντά και οι δύο. Το δεύτερο ζευγάρι υπερισχύει κατά 0.03.

Οι λέξεις man, lawyer και woman, lawyer είναι σχεδόν εξίσου κοντά επίσης, αν και το πρώτο ζευγάρι υπερισχύει κατά 0.02.

Μεγαλύτερη διαφορά παρατηρούμε στις λέξεις, man& president και woman & president, όπου αν και οι δύο είναι μακριά, το πρώτο ζευγάρι υπερτερεί με 0.06.

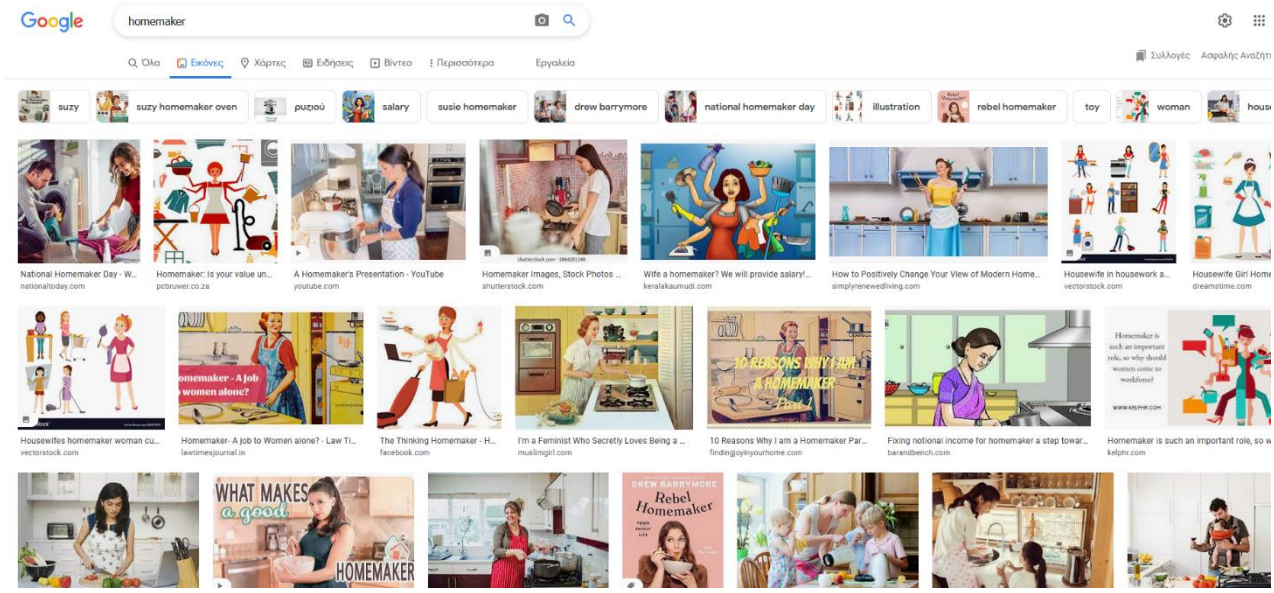
Τέλος, στα επίθετα παρατηρούμε πολύ μικρές διαφορές, της τάξης του 0.02 στο beautiful «υπερ» των αντρών, της τάξης του 0.05 στο hardworking «υπερ» των αντρών και της τάξης του 0.05 στο whining «υπερ» των γυναικών.

C. Τέλος, από περιέργεια κάναμε κάποιες αναζητήσεις εικόνων στο google. Ψάξαμε τις λέξεις ceo, homemaker, nurse:



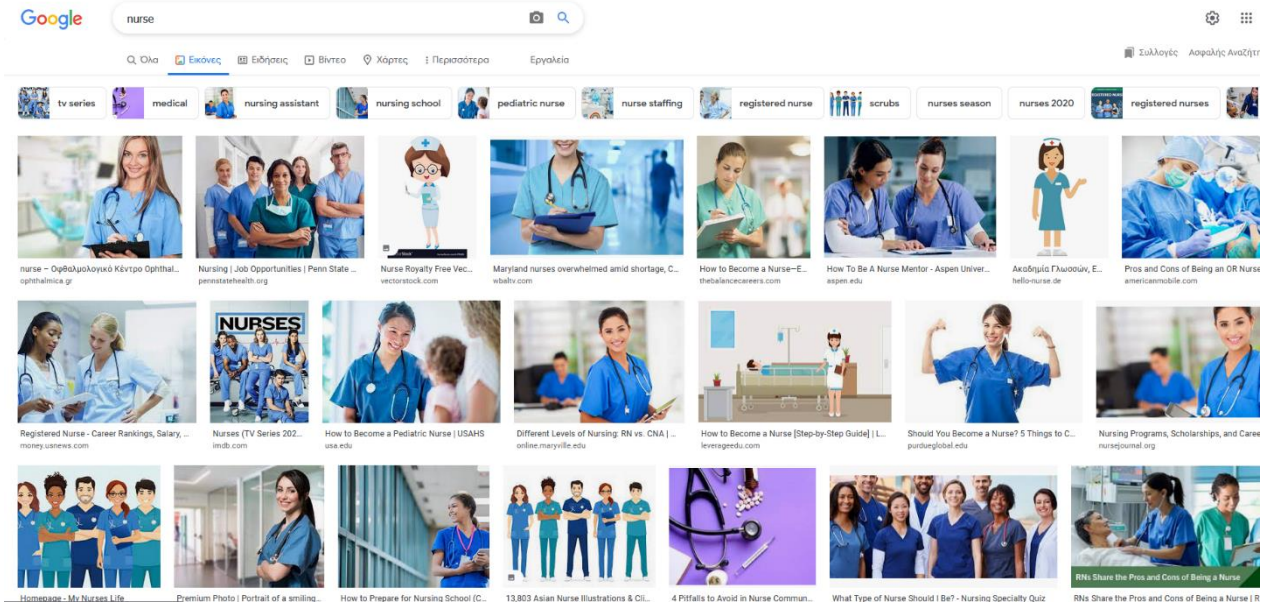
Προτεινόμενες εικόνες: άντρες

Πλειοψηφία εικονιζόμενων ατόμων: άντρες

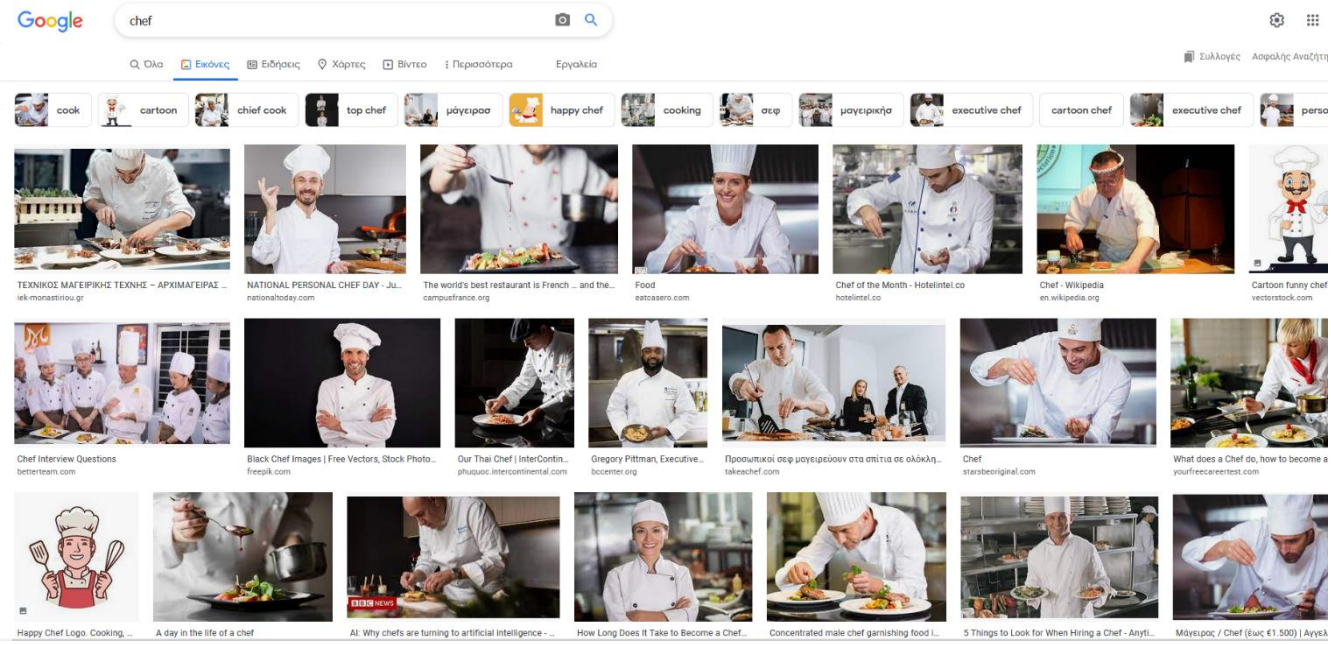


Προτεινόμενες εικόνες: γυναίκα

Πλειοψηφία εικονιζόμενων ατόμων: γυναίκες



Πλειοψηφία εικονιζόμενων ατόμων: γυναίκες



Πλειοψηφία εικονιζόμενων ατόμων: άντρες

Πηγές:

https://projector.tensorflow.org/

https://proceedings.neurips.cc/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6835575.pdf

https://www.kaggle.com/code/rtatman/gender-bias-in-word-embeddings

https://www.kaggle.com/code/jeffd23/visualizing-word-vectors-with-t-sne/notebook

https://nlp.stanford.edu/pubs/glove.pdf

[1] https://nlp.stanford.edu/projects/glove/

[2] https://www.youtube.com/watch?v=7ahKnJTF_9Y

[3] https://www.youtube.com/watch?v=Fn_U2OG1uqI