

```
In [ ]: import analyze as a
```

```
In [ ]: titles = ["Bible", "BUG", "GAP", "Moby Dick"]

bibleNEmbeds = a.embeds('NvectorsBible.txt')
bibleGEmbeds = a.embeds('GvectorsBible.txt')

bugNEmbeds = a.embeds('NvectorsBUG.txt')
bugGEmbeds = a.embeds('GvectorsBUG.txt')

gapNEmbeds = a.embeds('NvectorsGAP.txt')
gapGEmbeds = a.embeds('GvectorsGAP.txt')

mobyNEmbeds = a.embeds('NvectorsMoby.txt')
mobyGEmbeds = a.embeds('GvectorsMoby.txt')

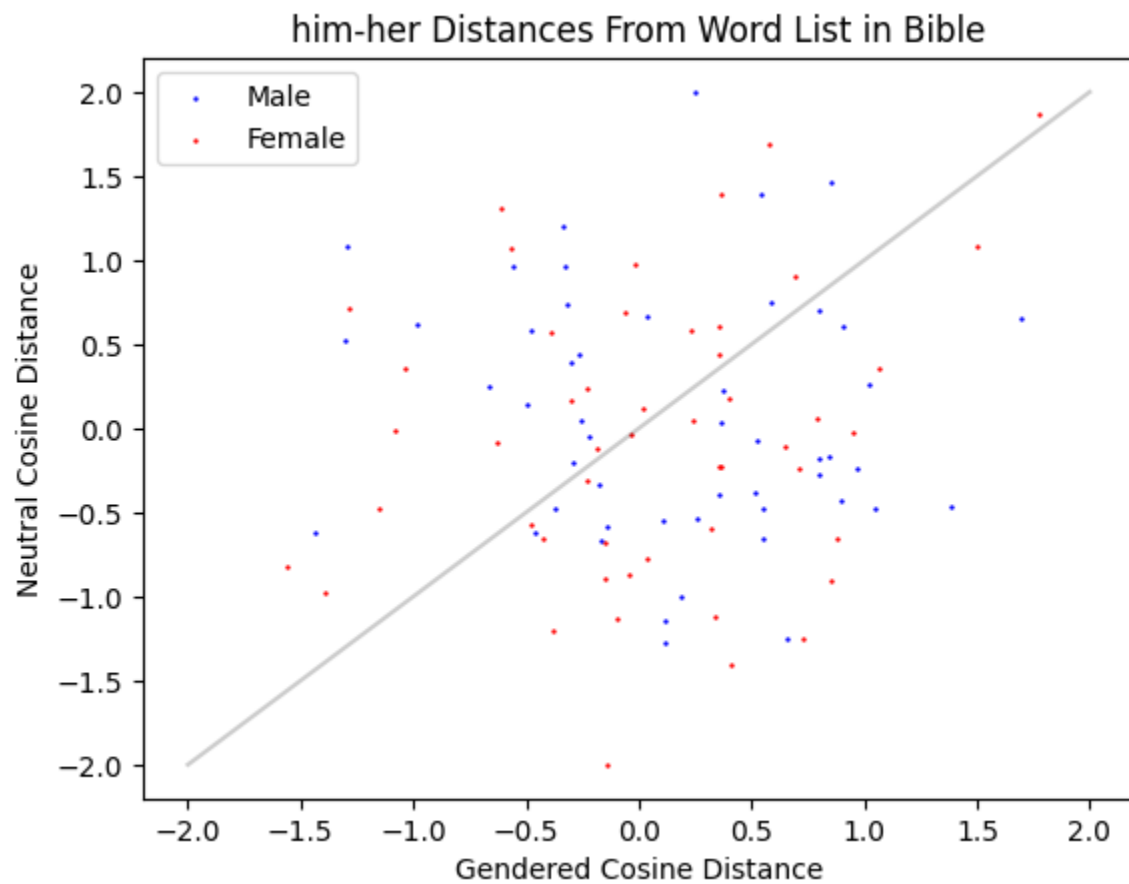
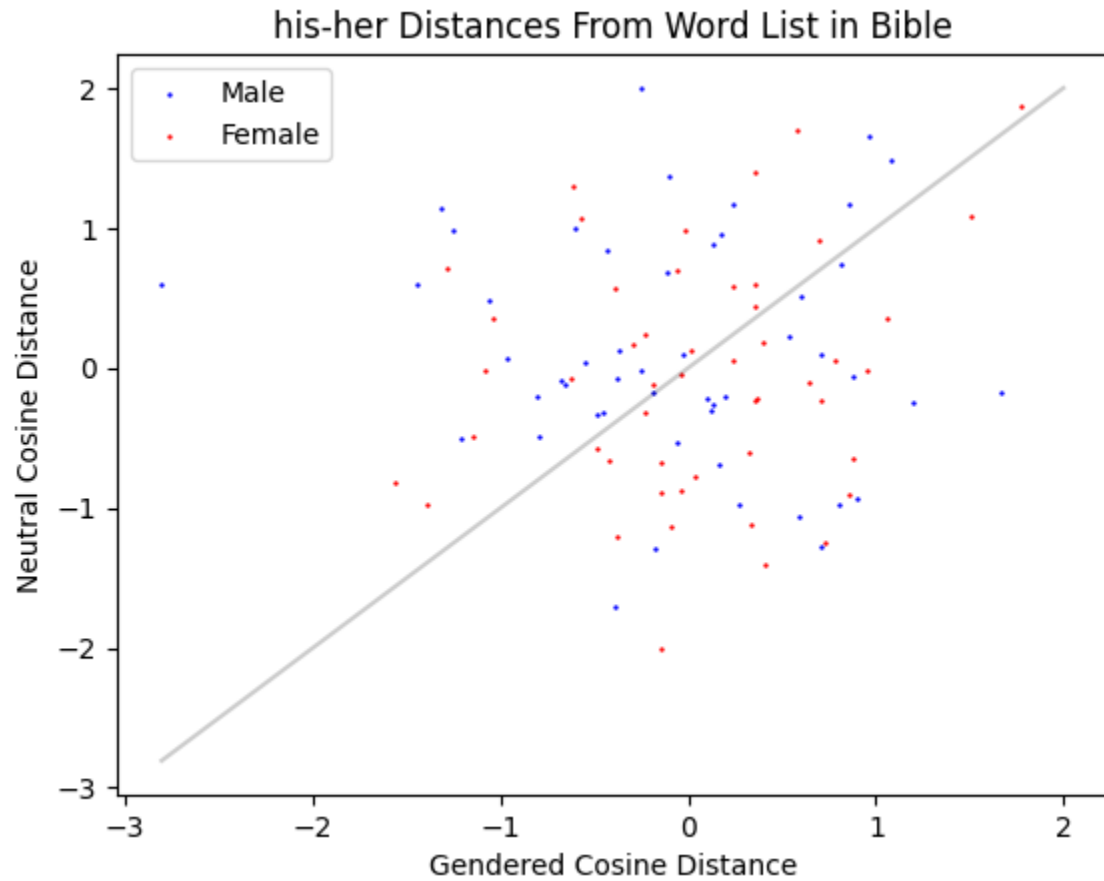
neutralEmbeds = [bibleNEmbeds, bugNEmbeds, gapNEmbeds, mobyNEmbeds]
genderEmbeds = [bibleGEmbeds, bugGEmbeds, gapGEmbeds, mobyGEmbeds]

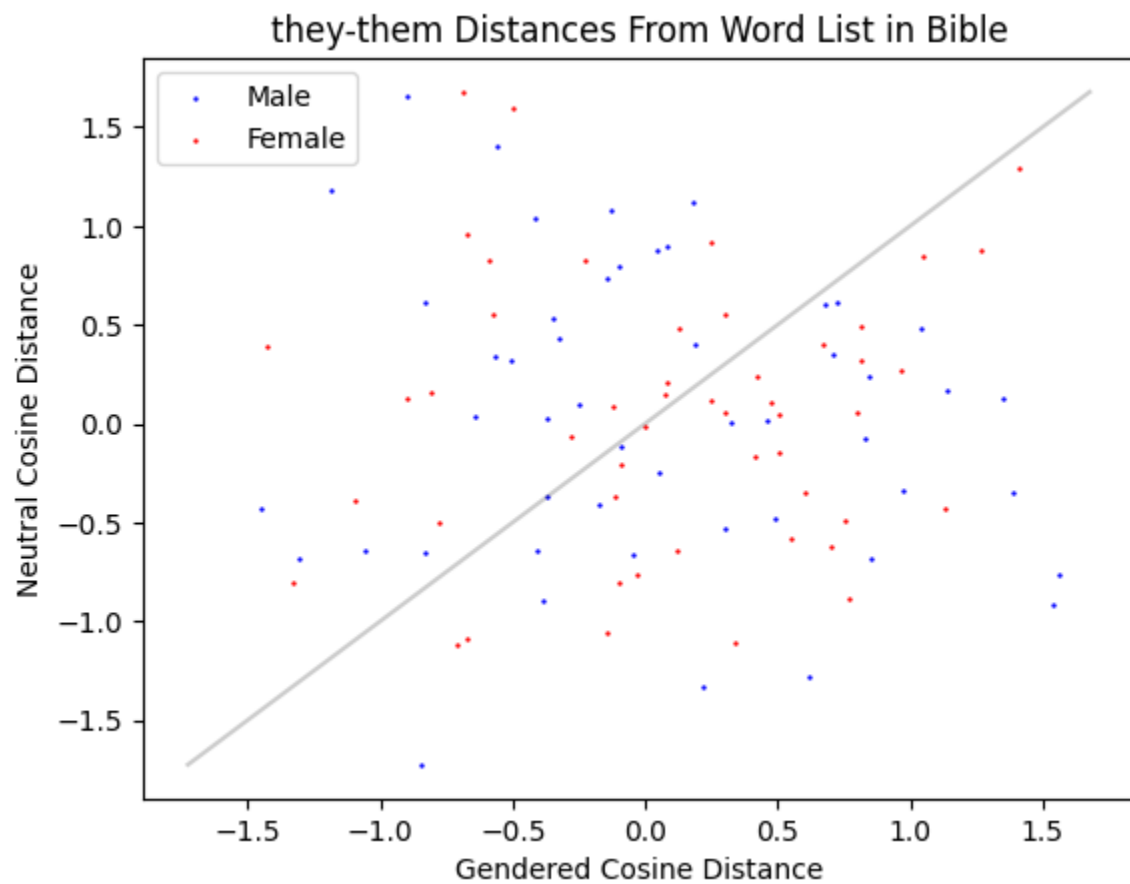
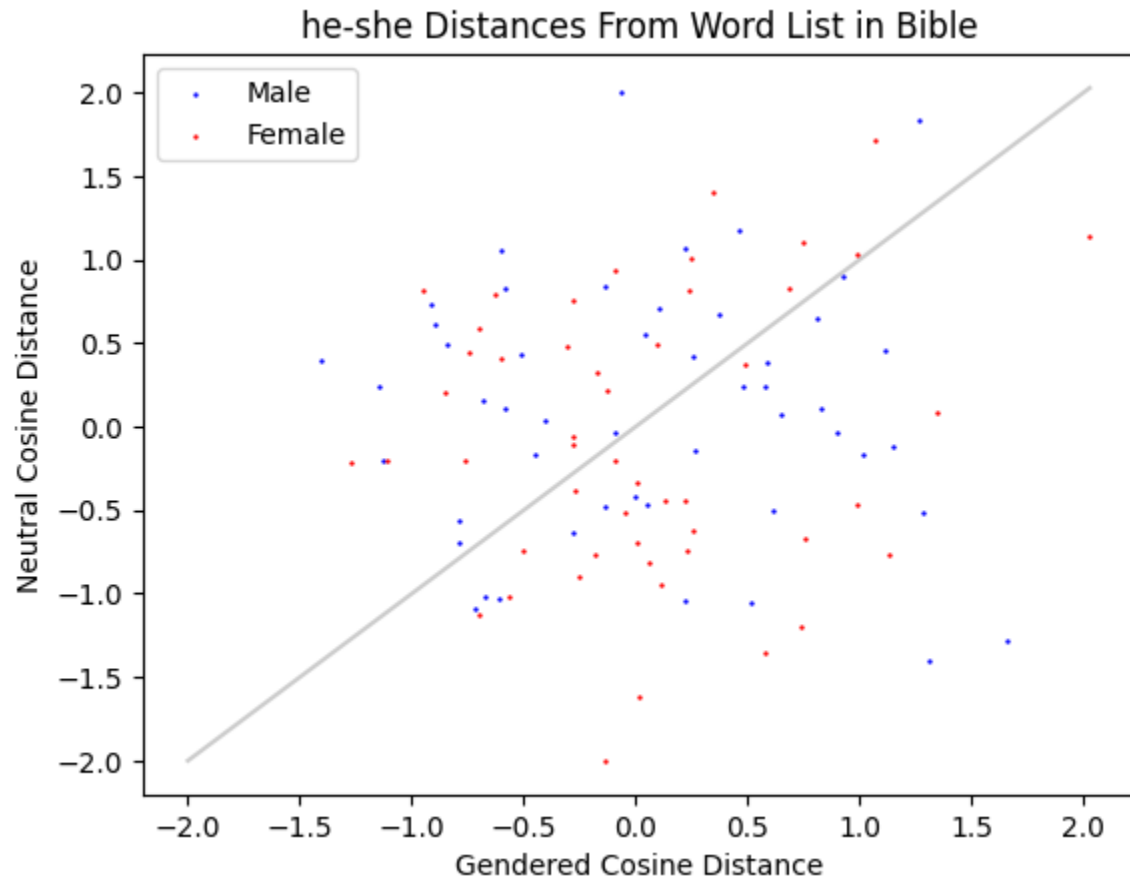
genderPairs = a.genderPairs('GenderPairs.csv')
spotlightWords = a.spotlightWords('Spotlight.csv')
```

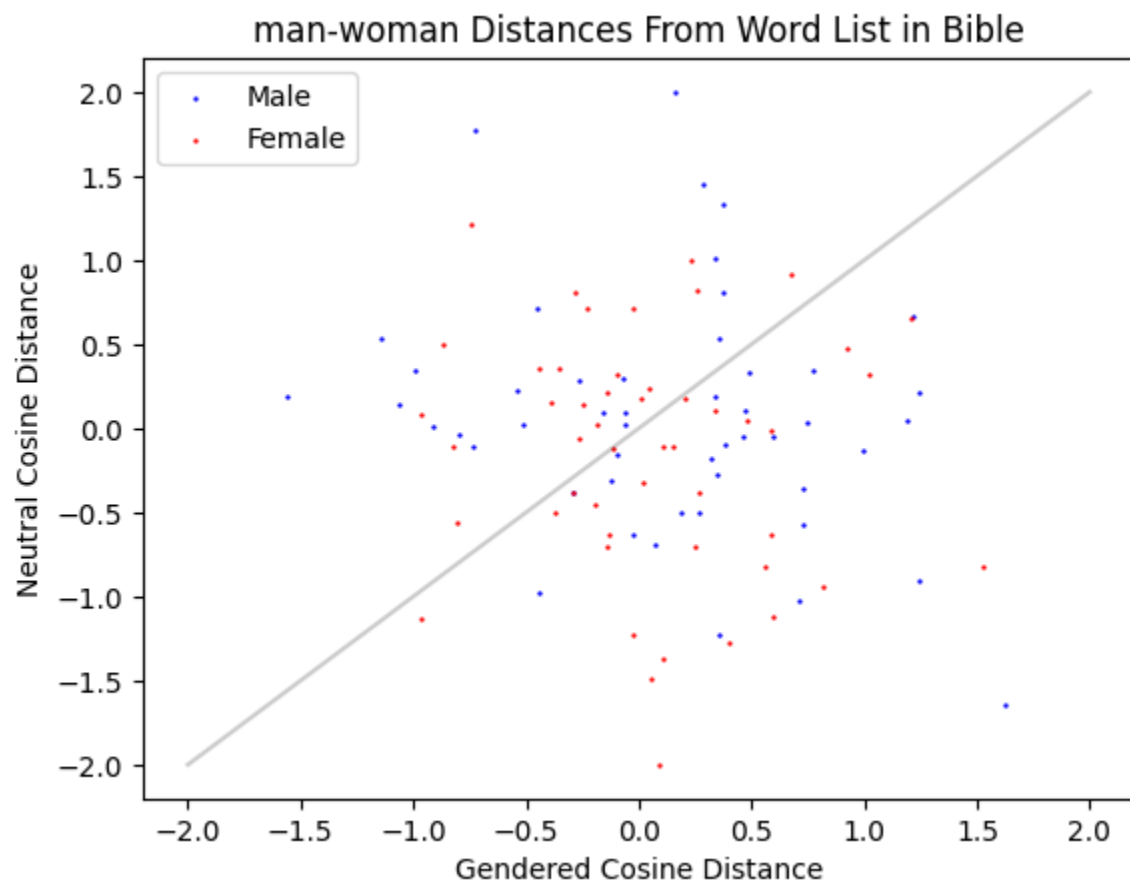
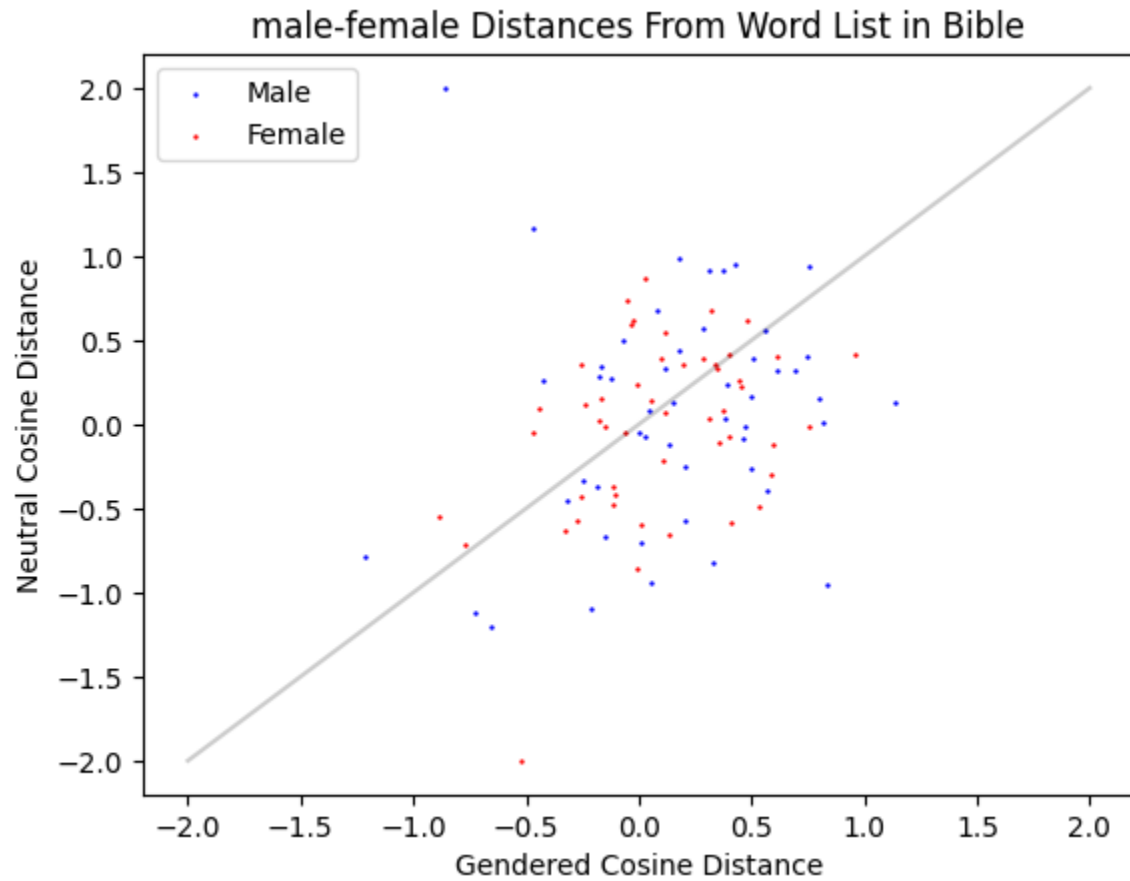
```
In [ ]: """for i, title in enumerate(titles):
        print("_____")
        a.plotter(genderPairs, spotlightWords, neutralEmbeds[i], genderEmbeds[i], title
        #delete quotes for all graphs - takes up a lot of space
```

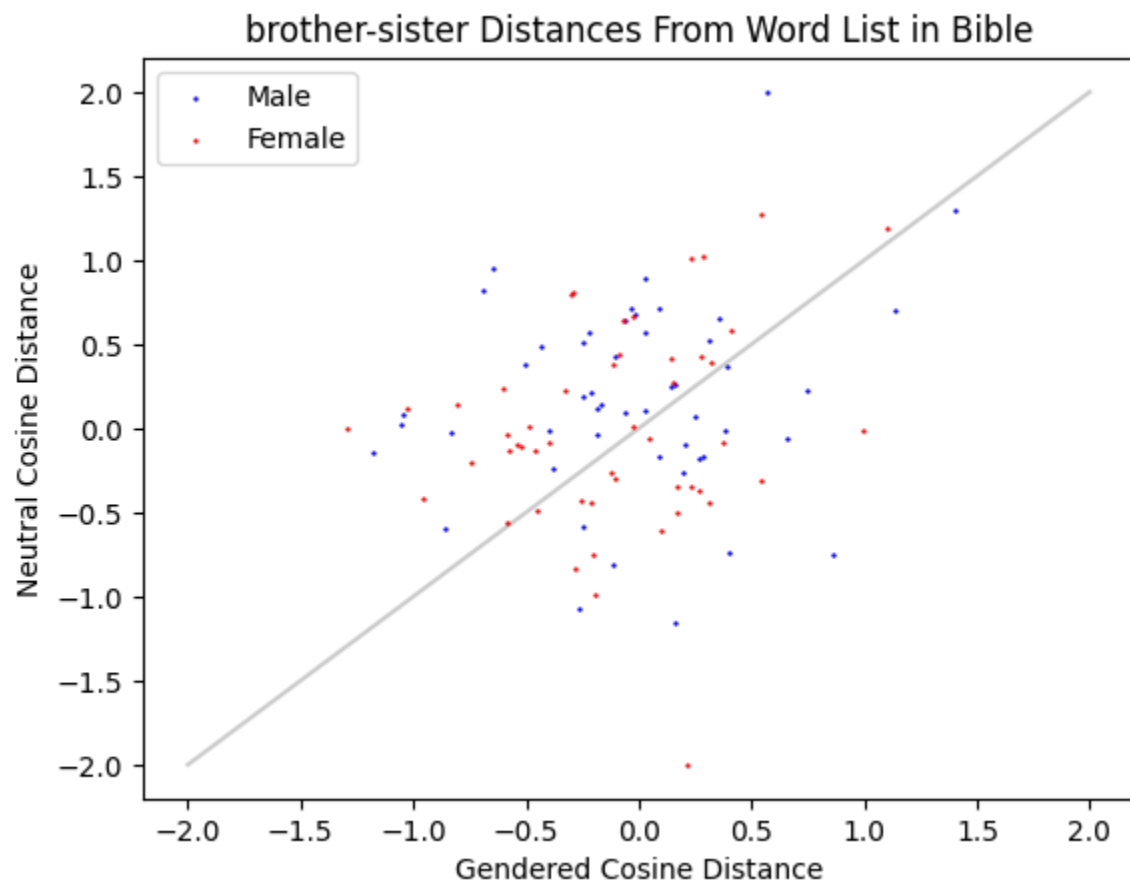
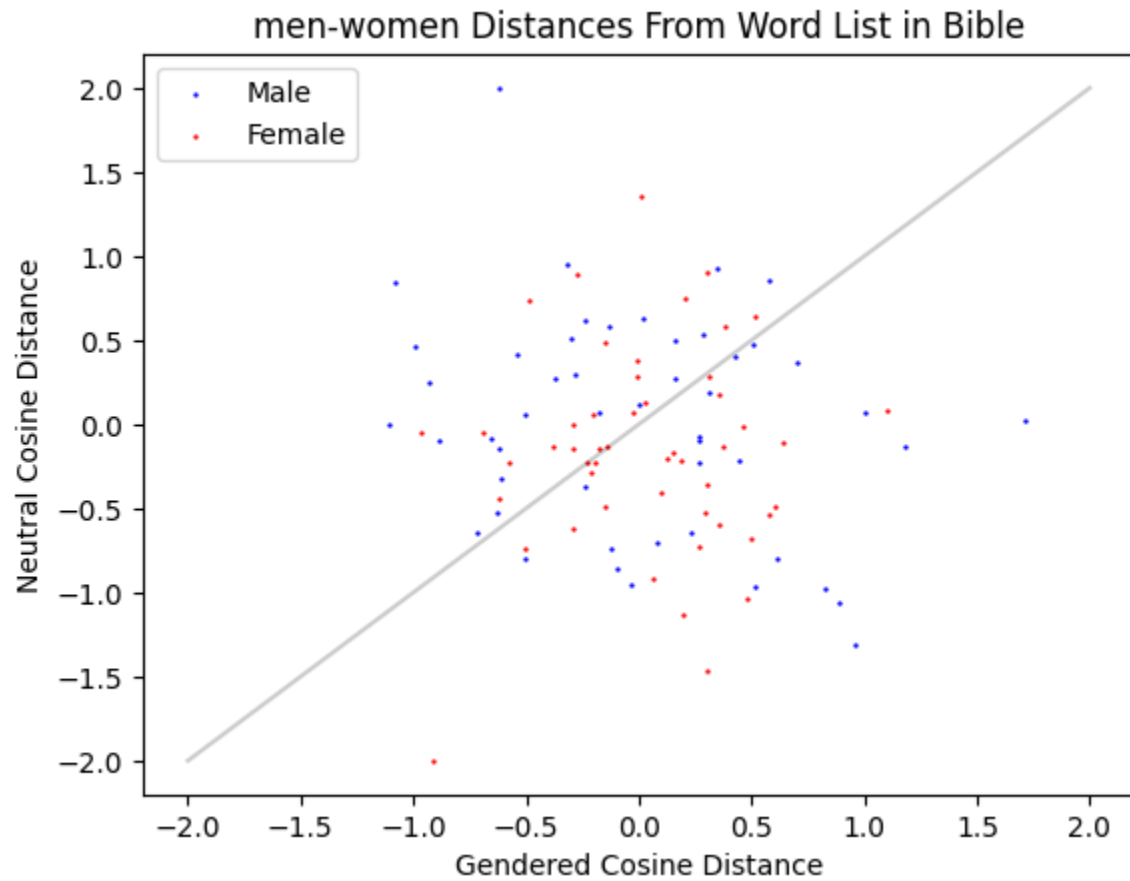
```
Out[ ]: 'for i, title in enumerate(titles):\n    print("_____\n    ___")\n    a.plotter(genderPairs, spotlightWords, neutralEmbeds[i], genderEmbeds\n    [i], title)'
```

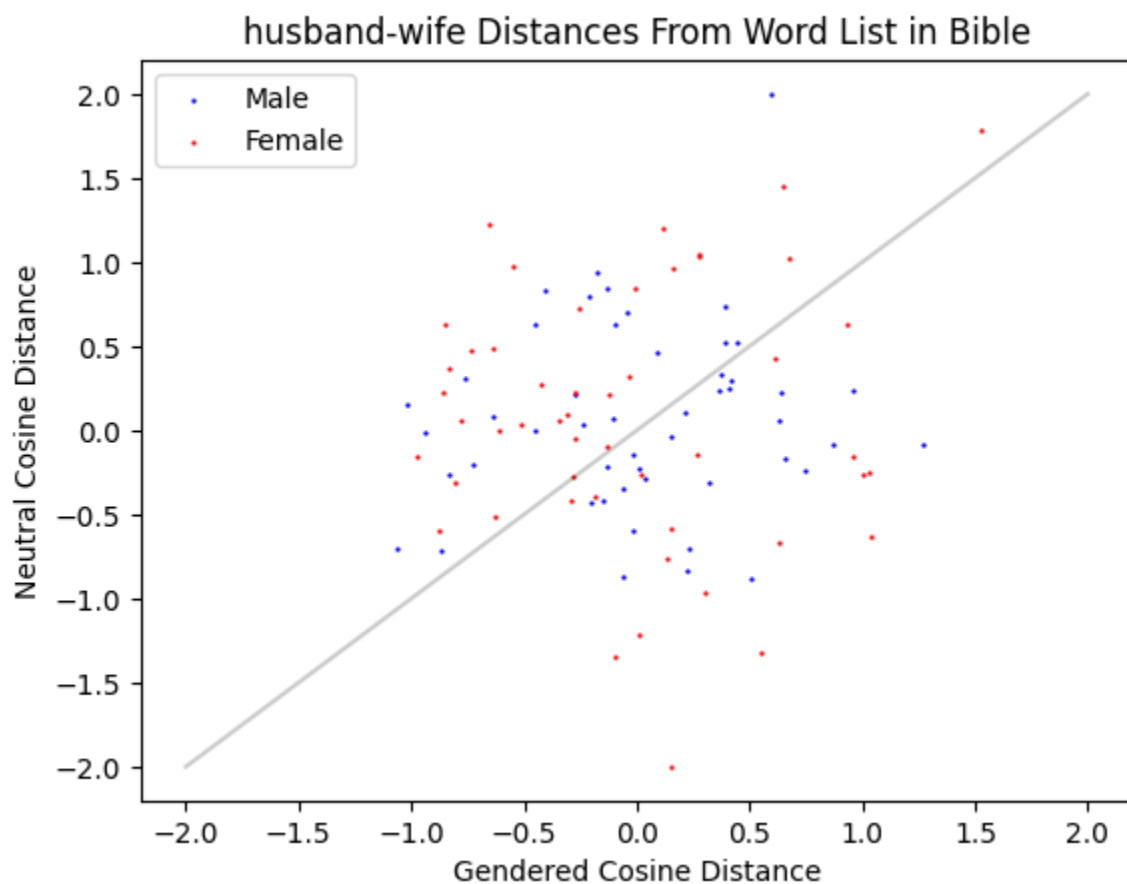
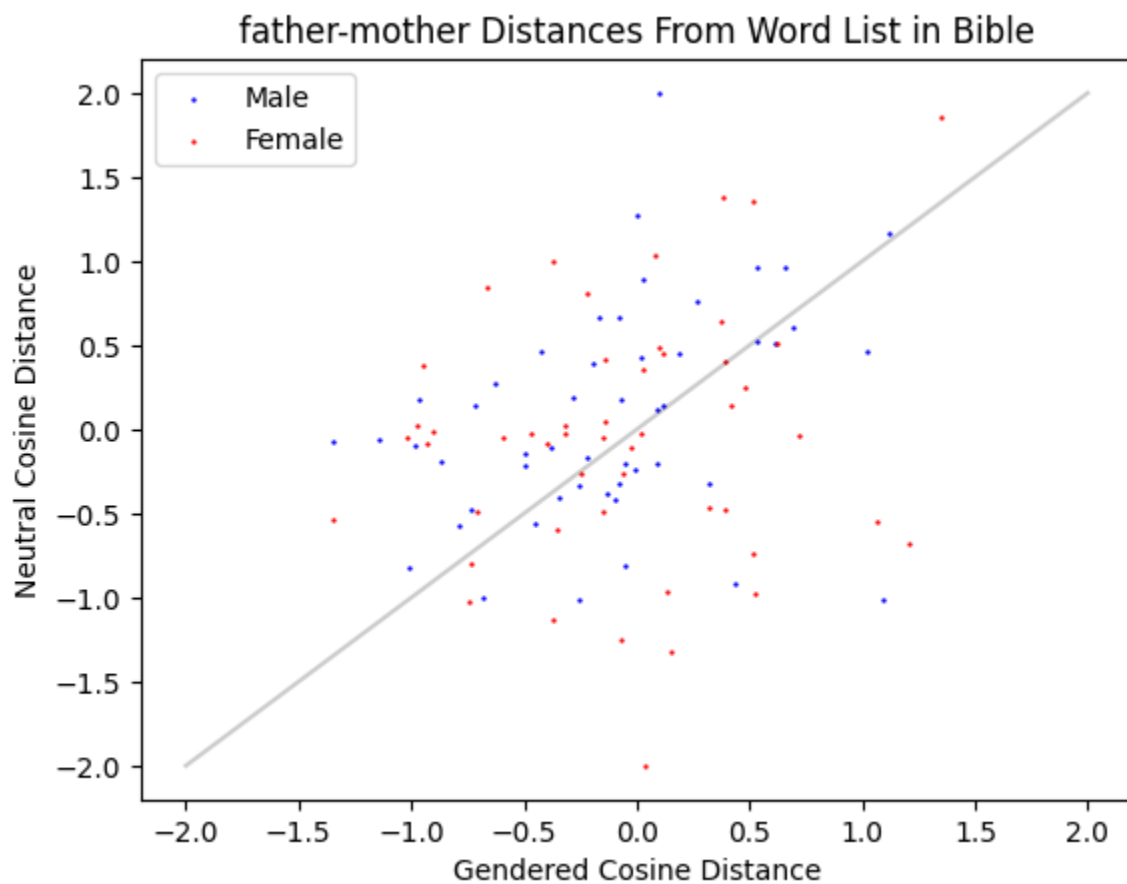
```
In [ ]: a.plotter(genderPairs, spotlightWords, bibleNEmbeds, bibleGEmbeds, titles[0])
```



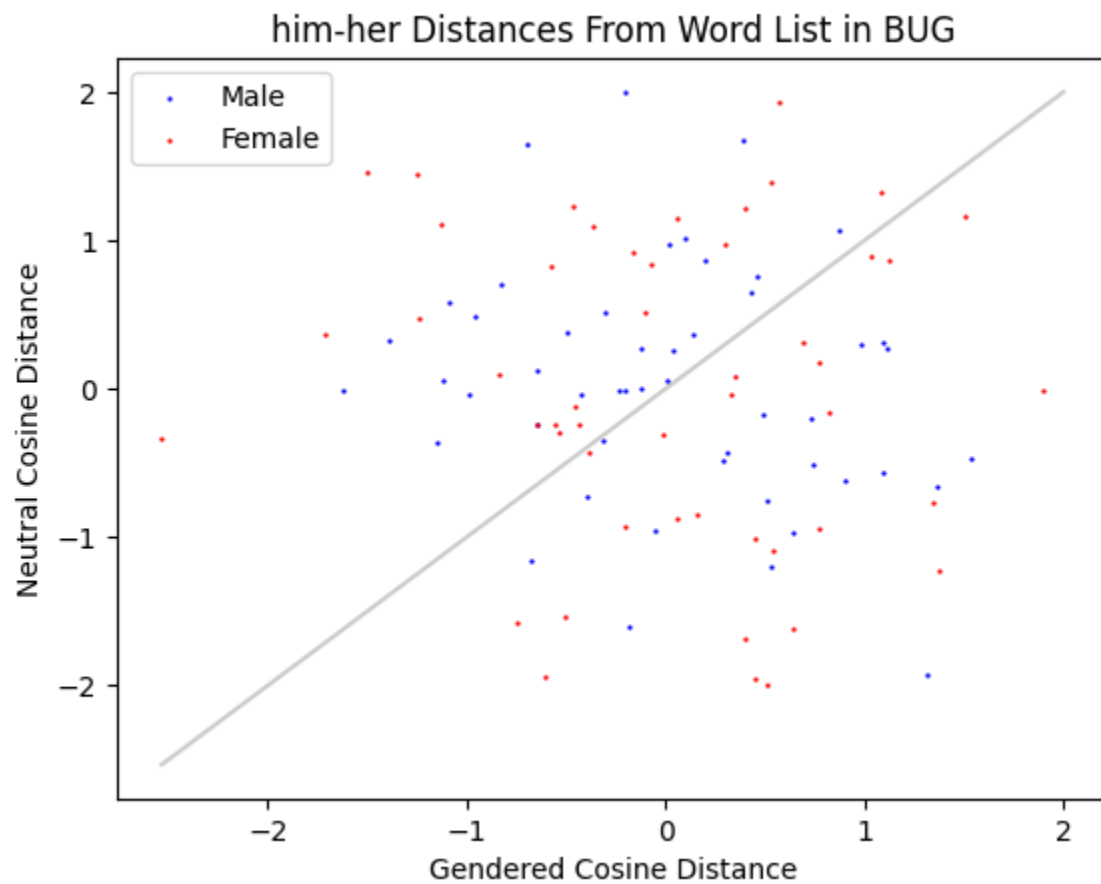
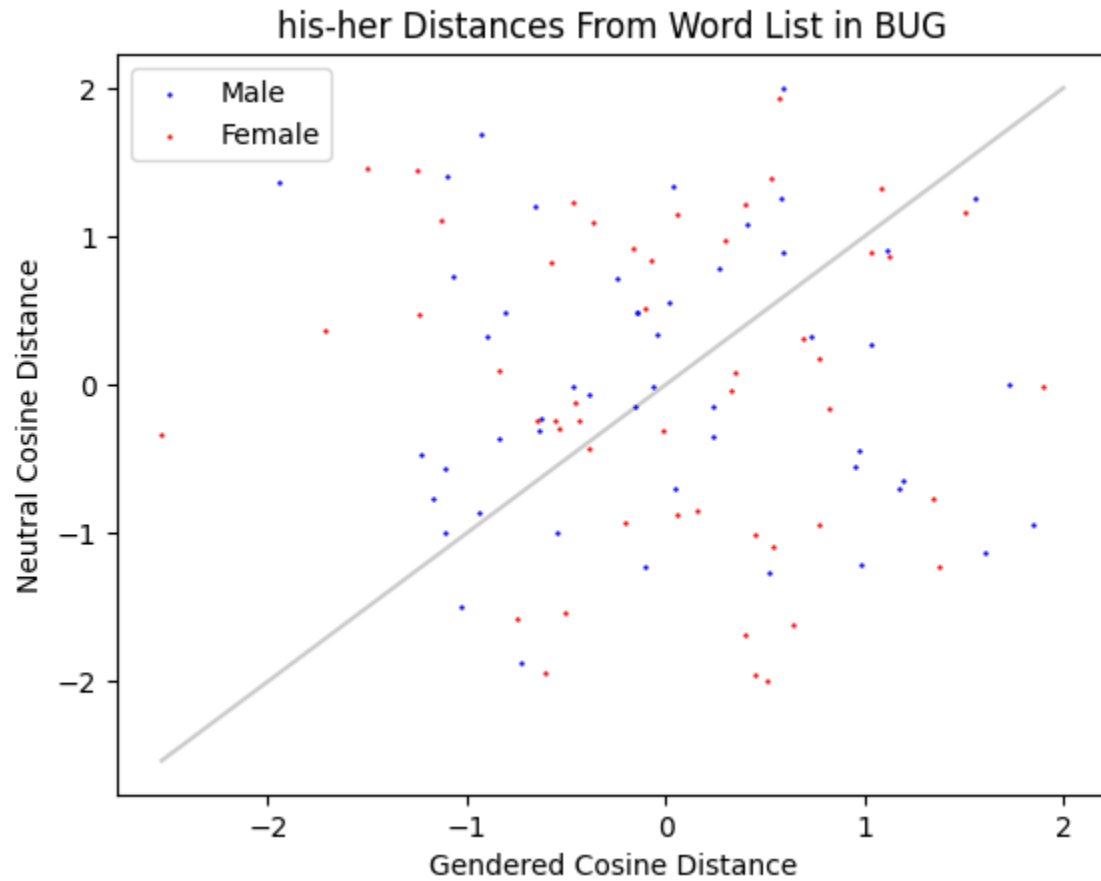


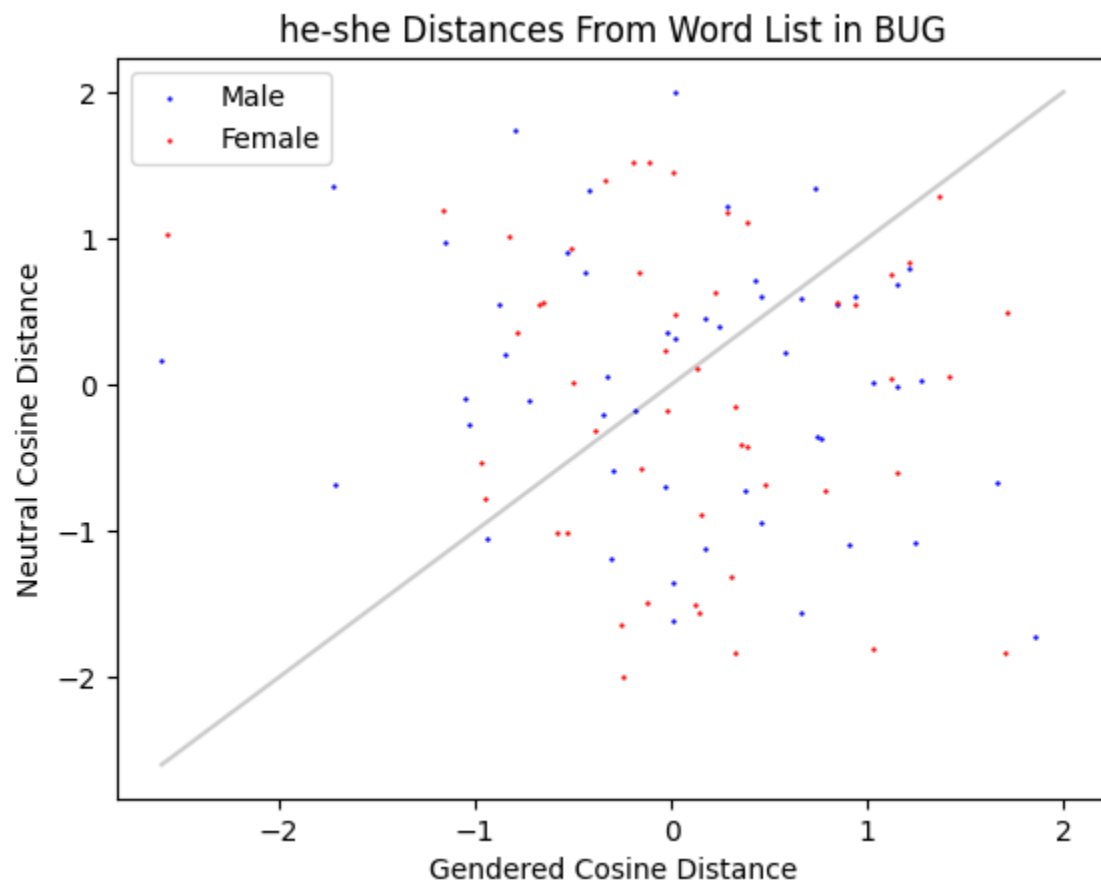
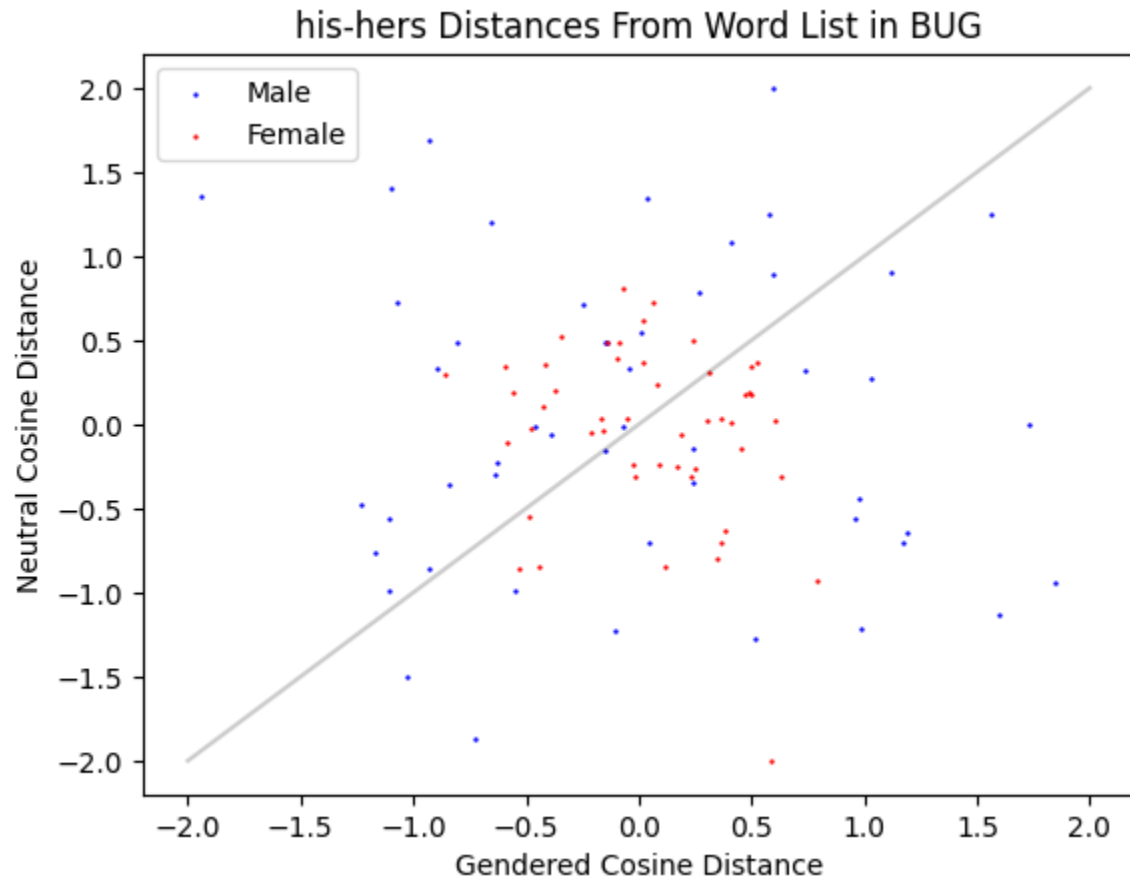


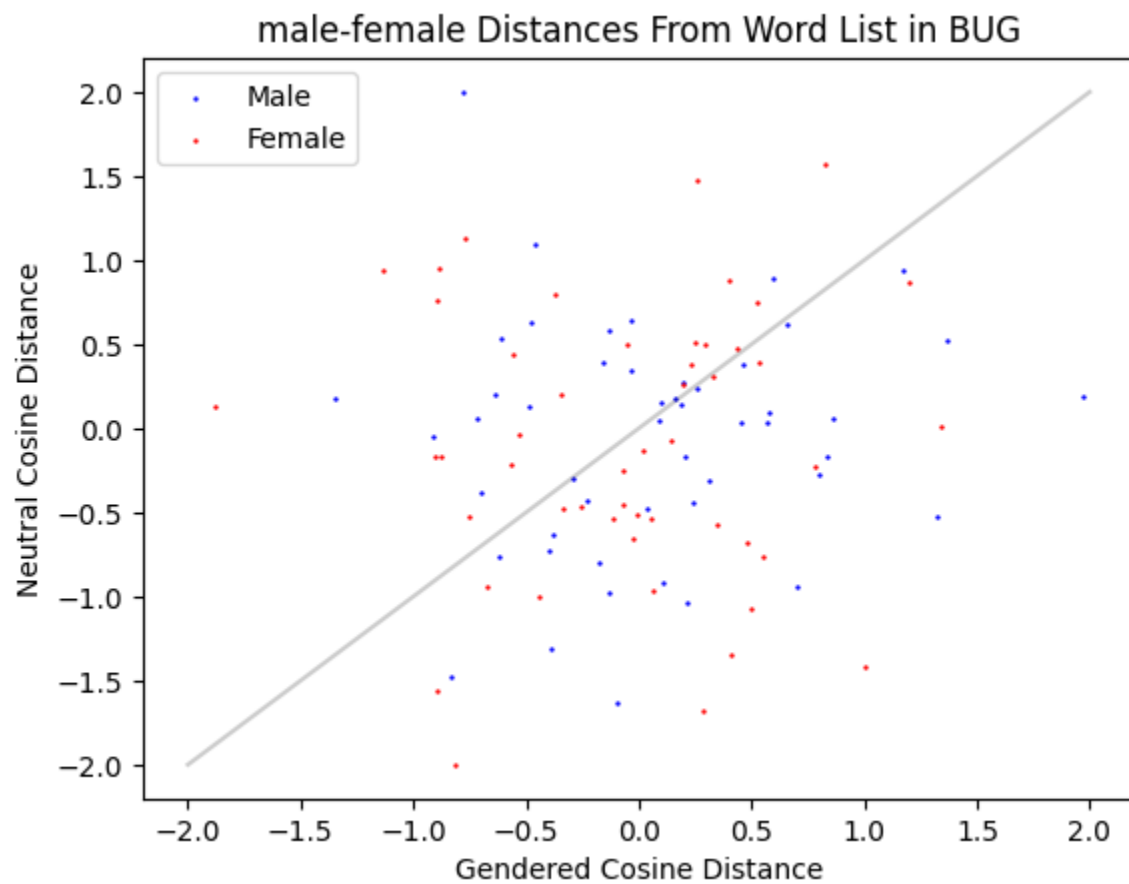
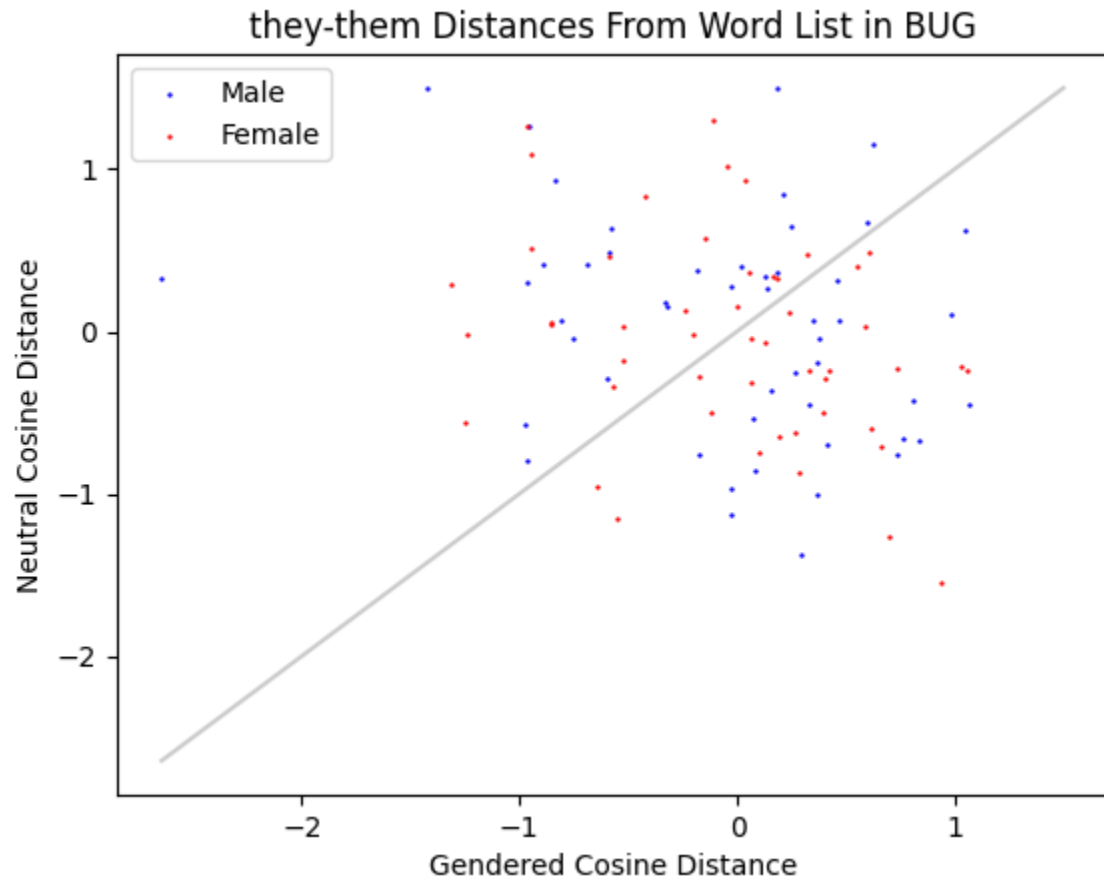


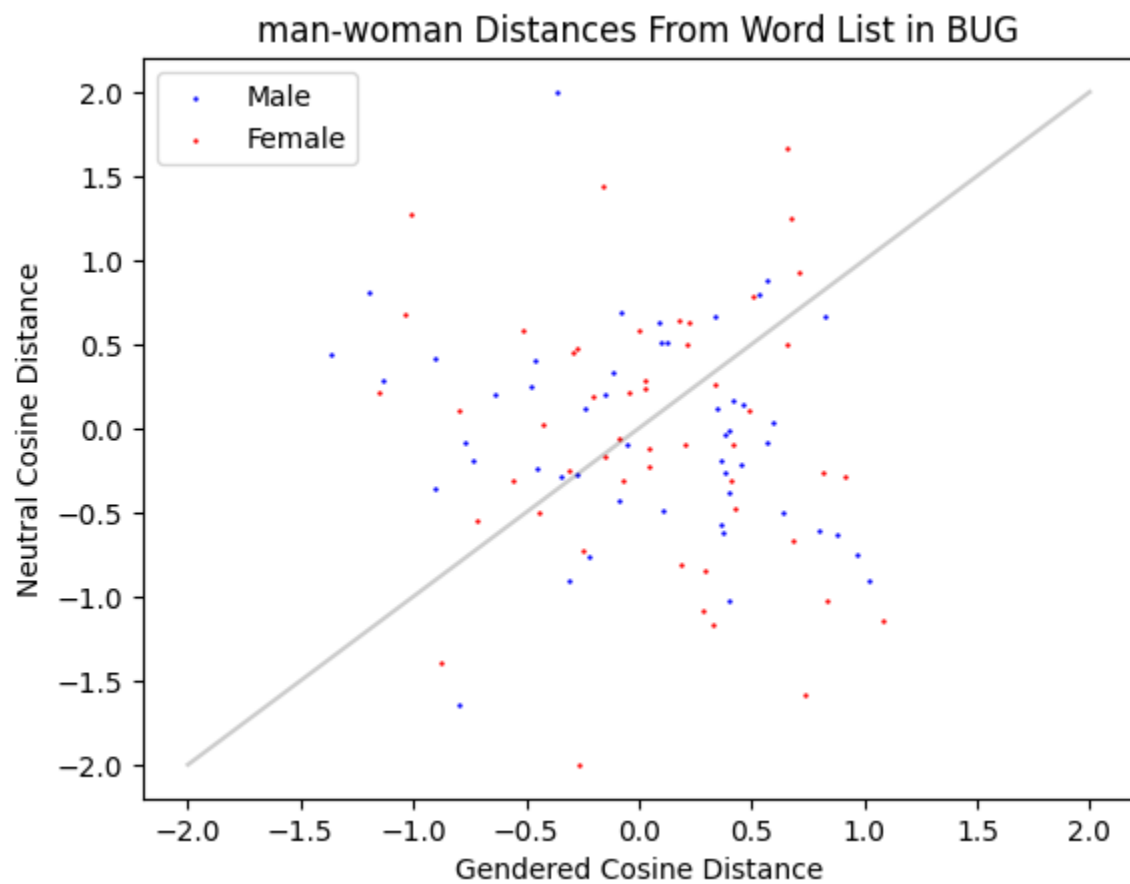
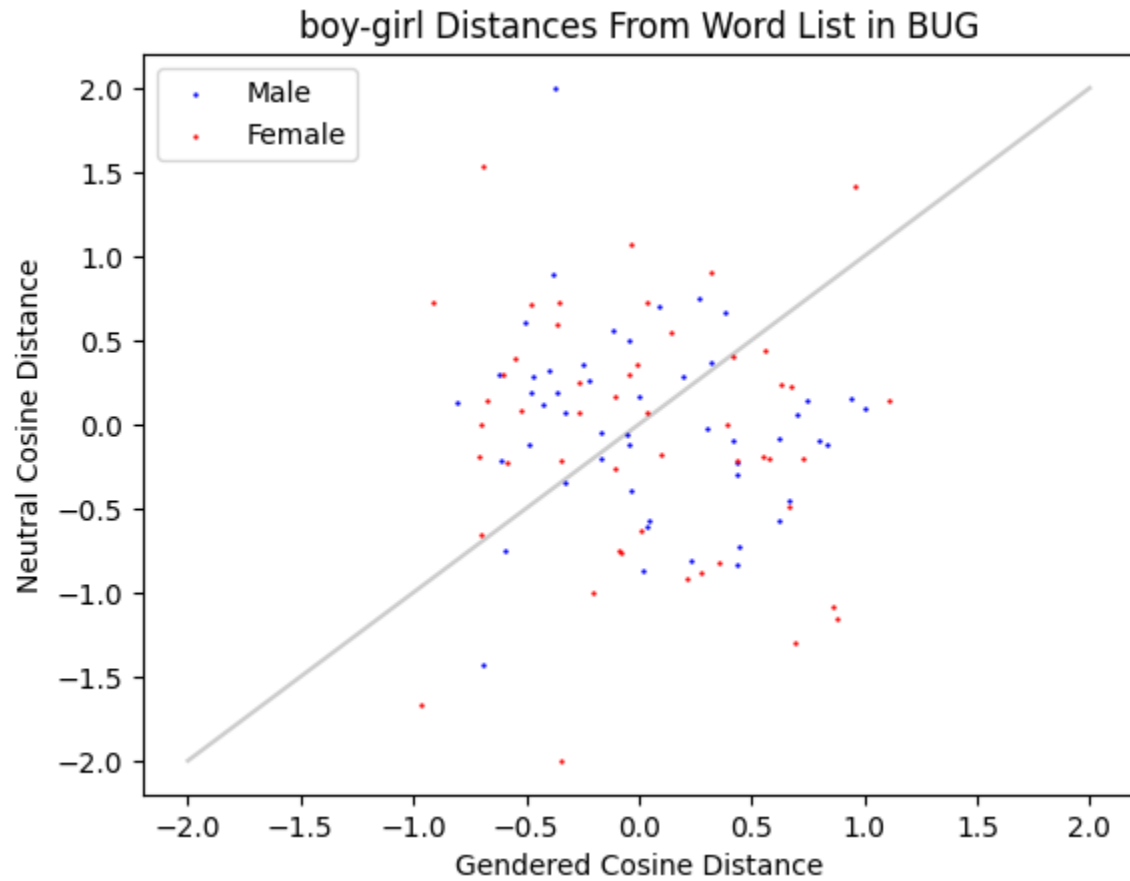


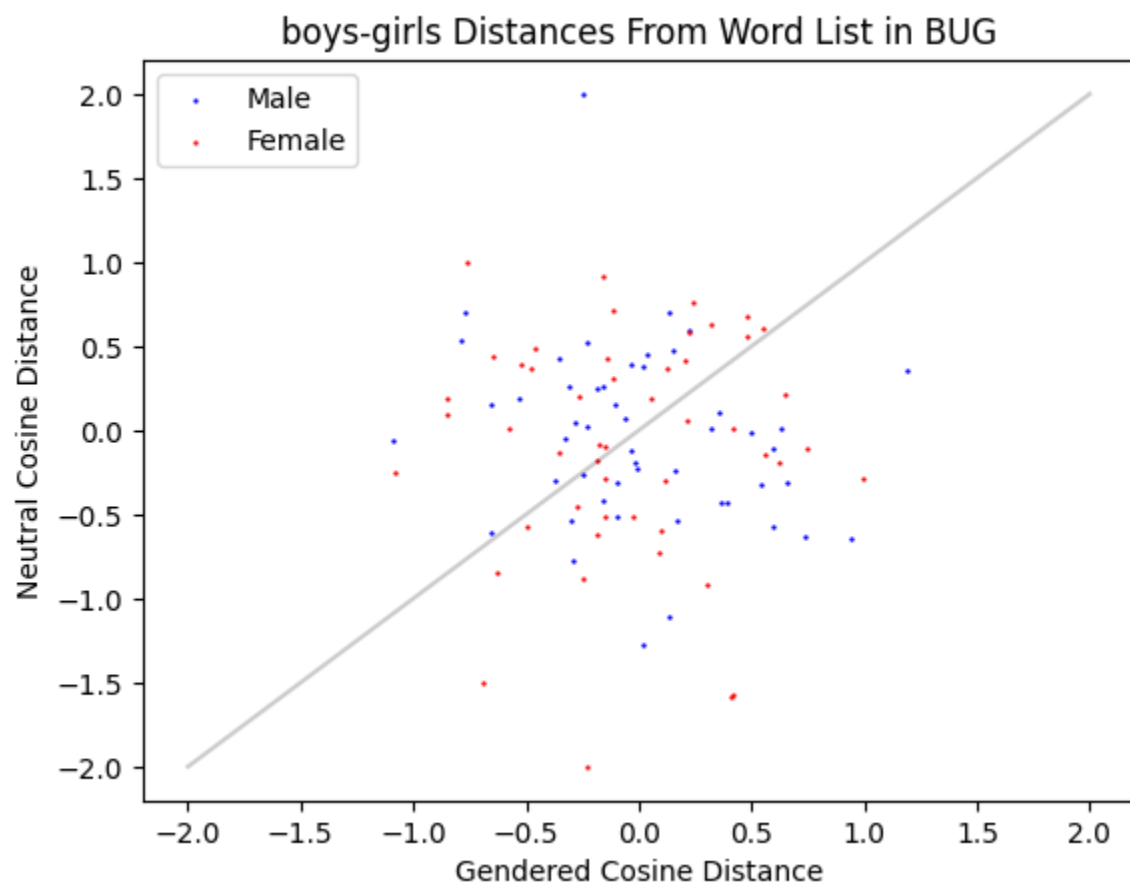
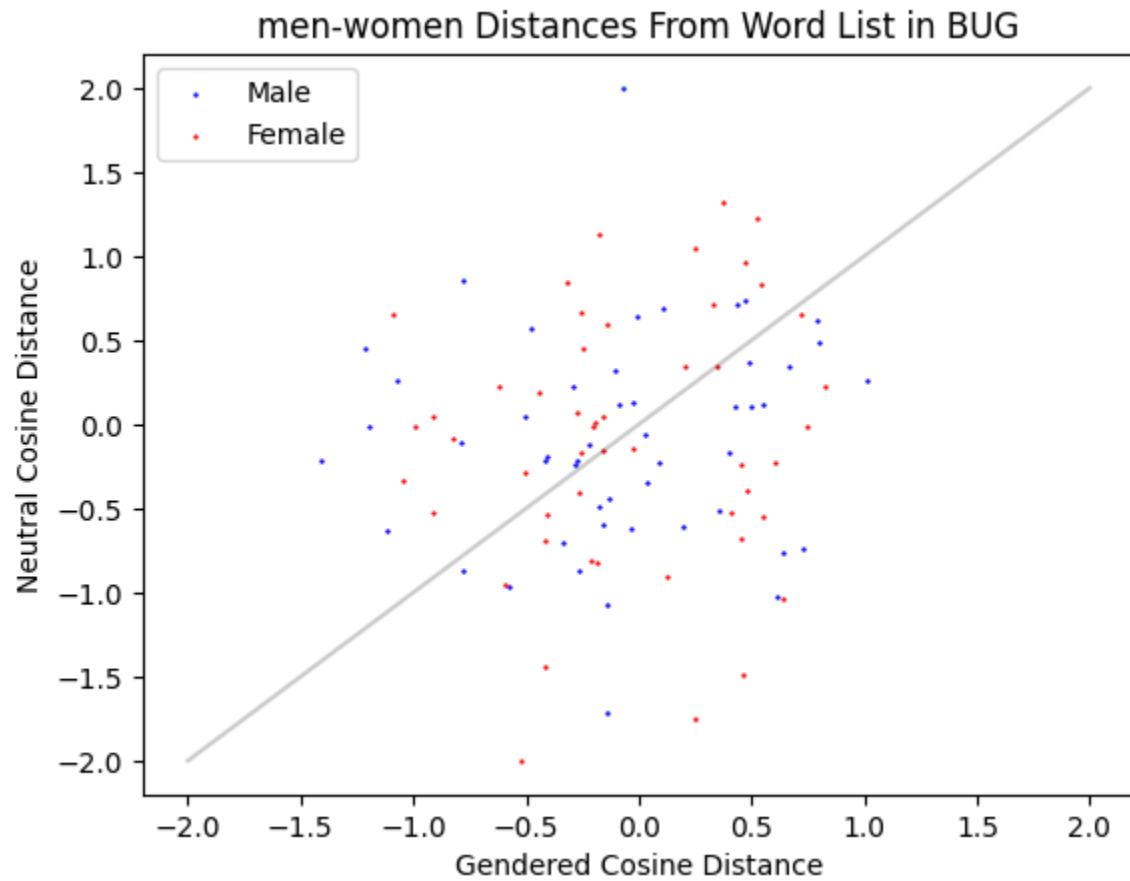
```
In [ ]: a.plotter(genderPairs, spotlightWords, bugNEmbeds, bugGEmbeds, titles[1])
```

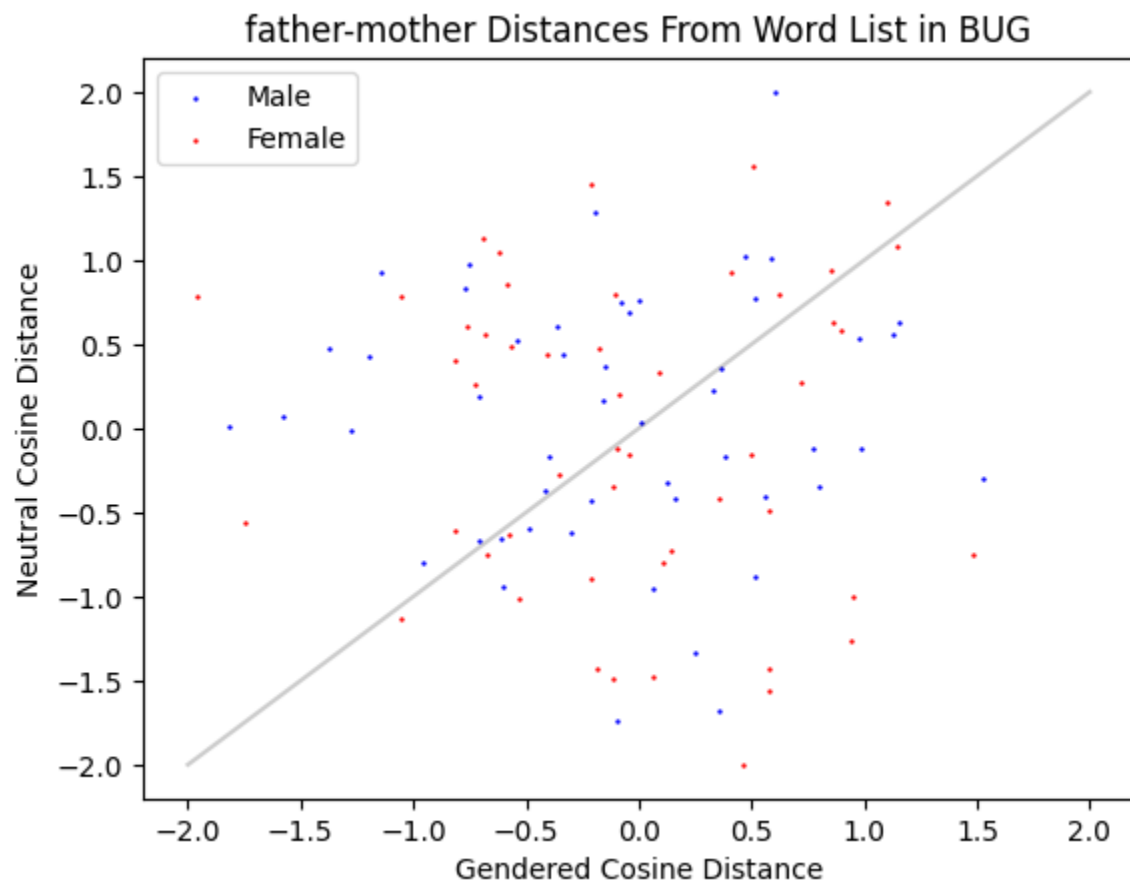
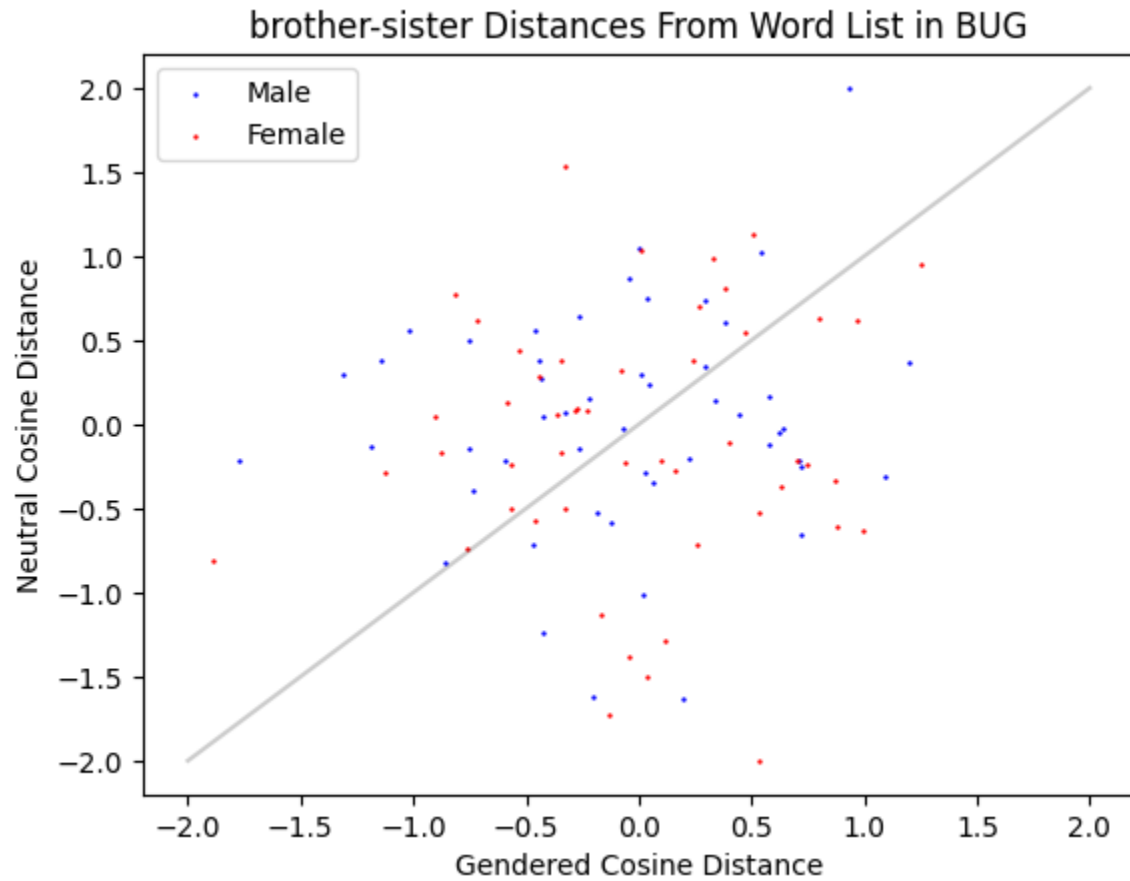


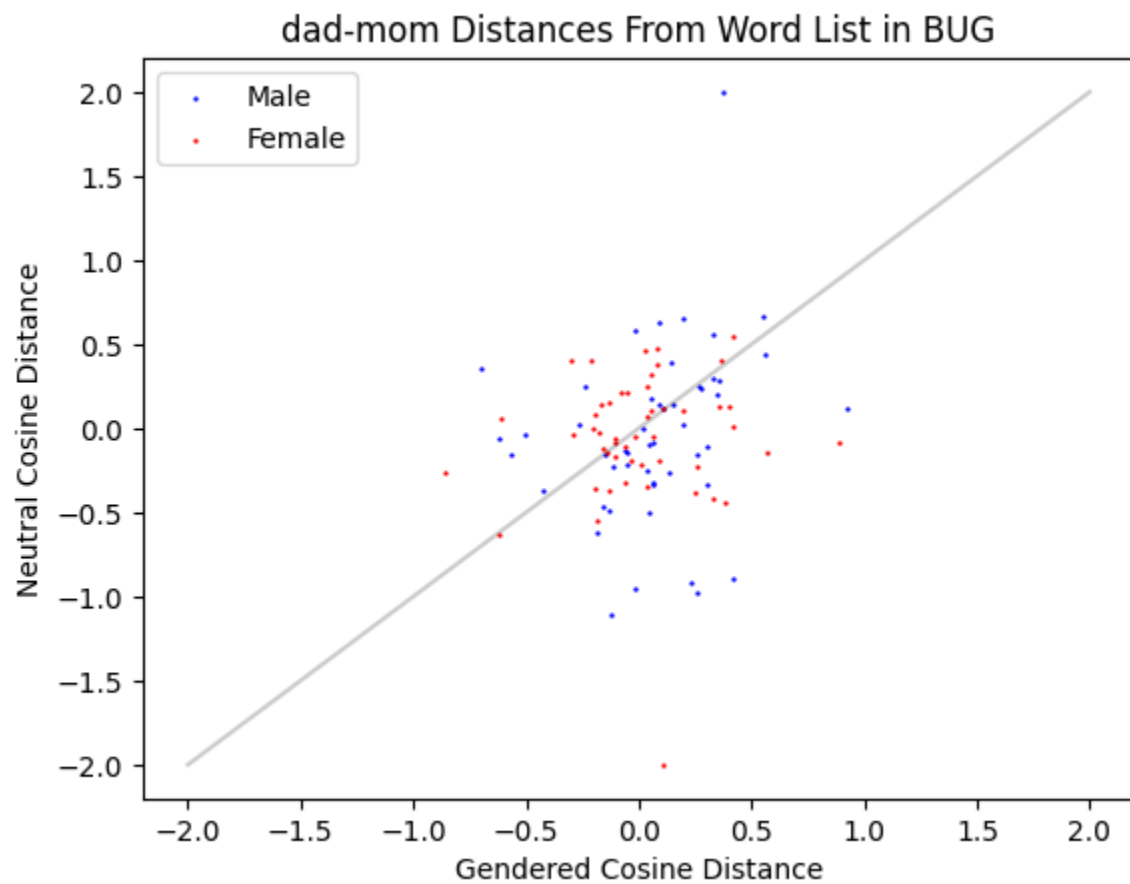
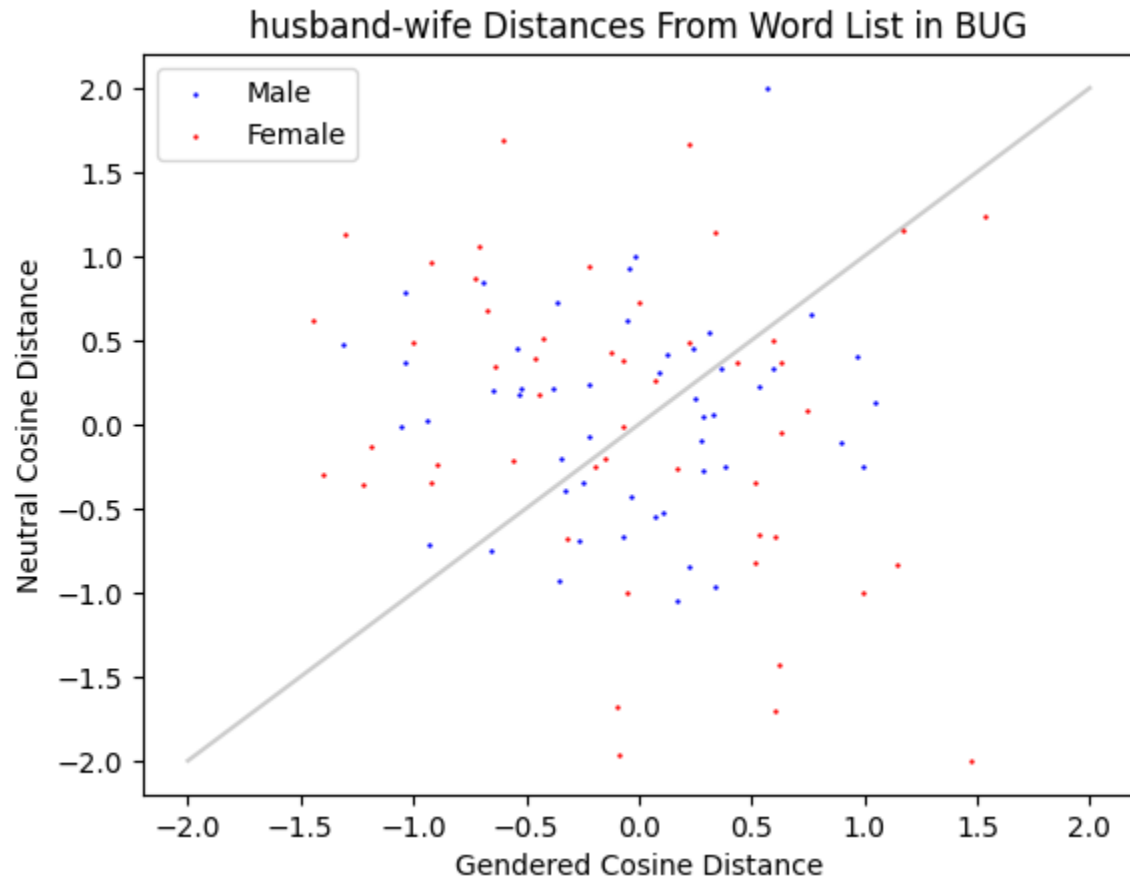


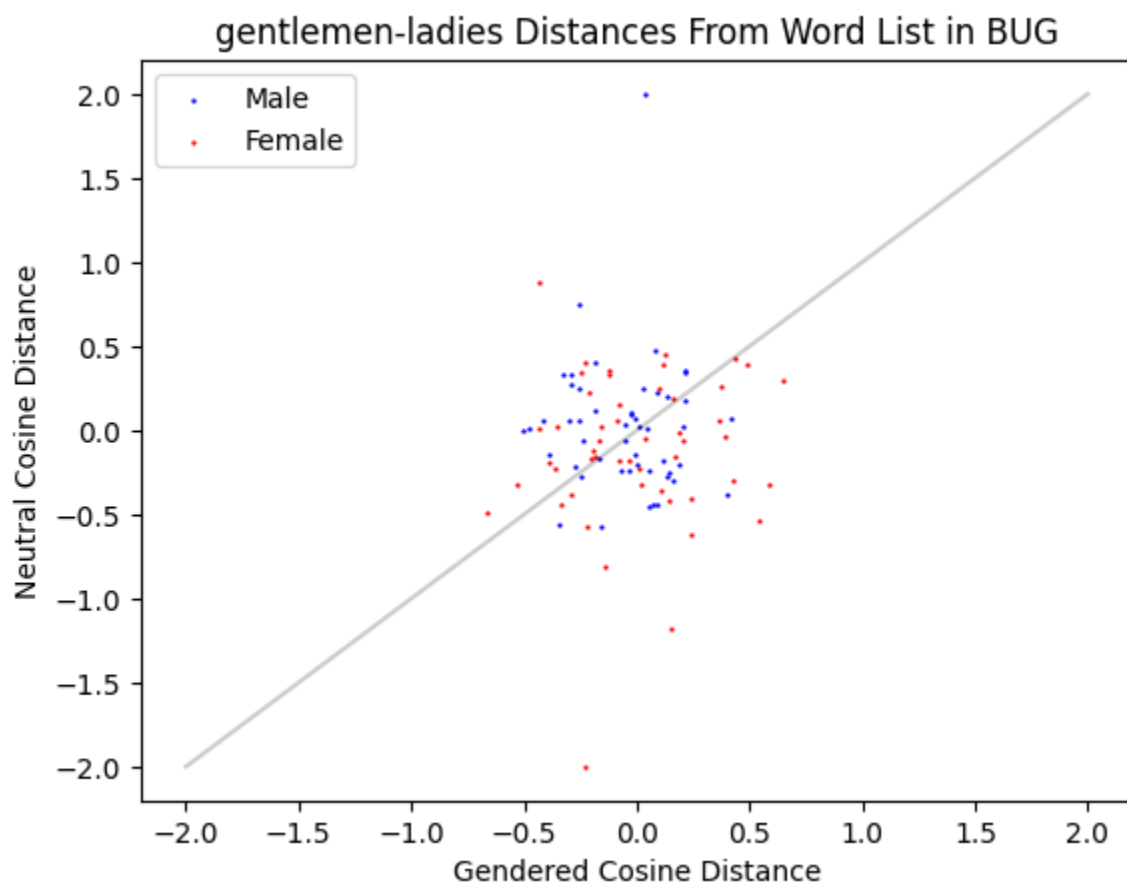
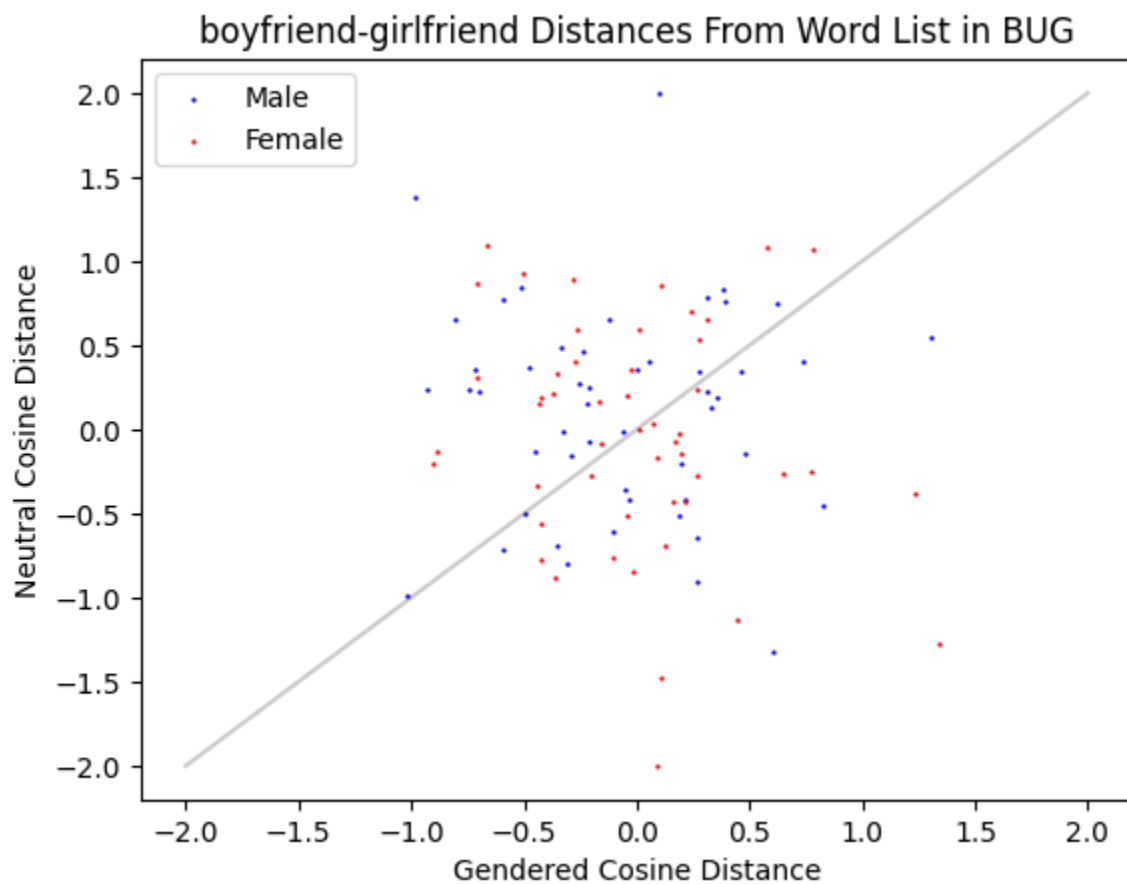




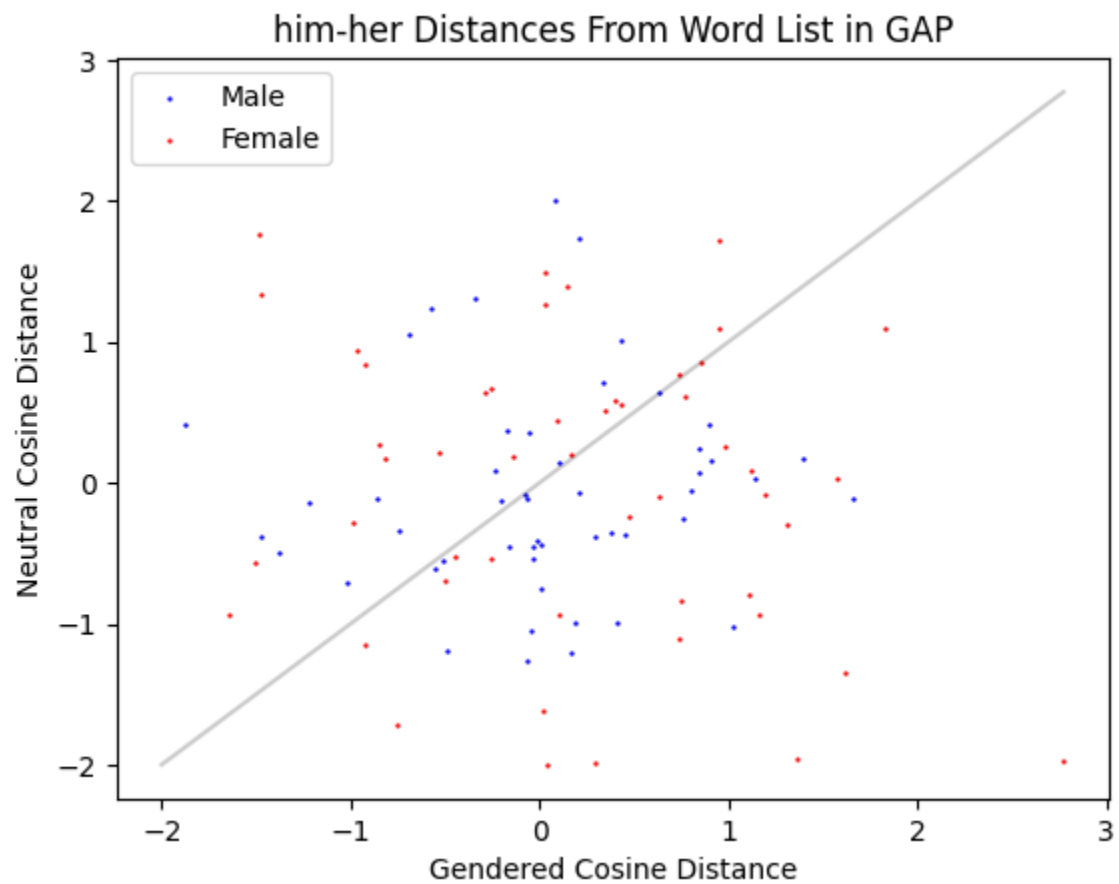
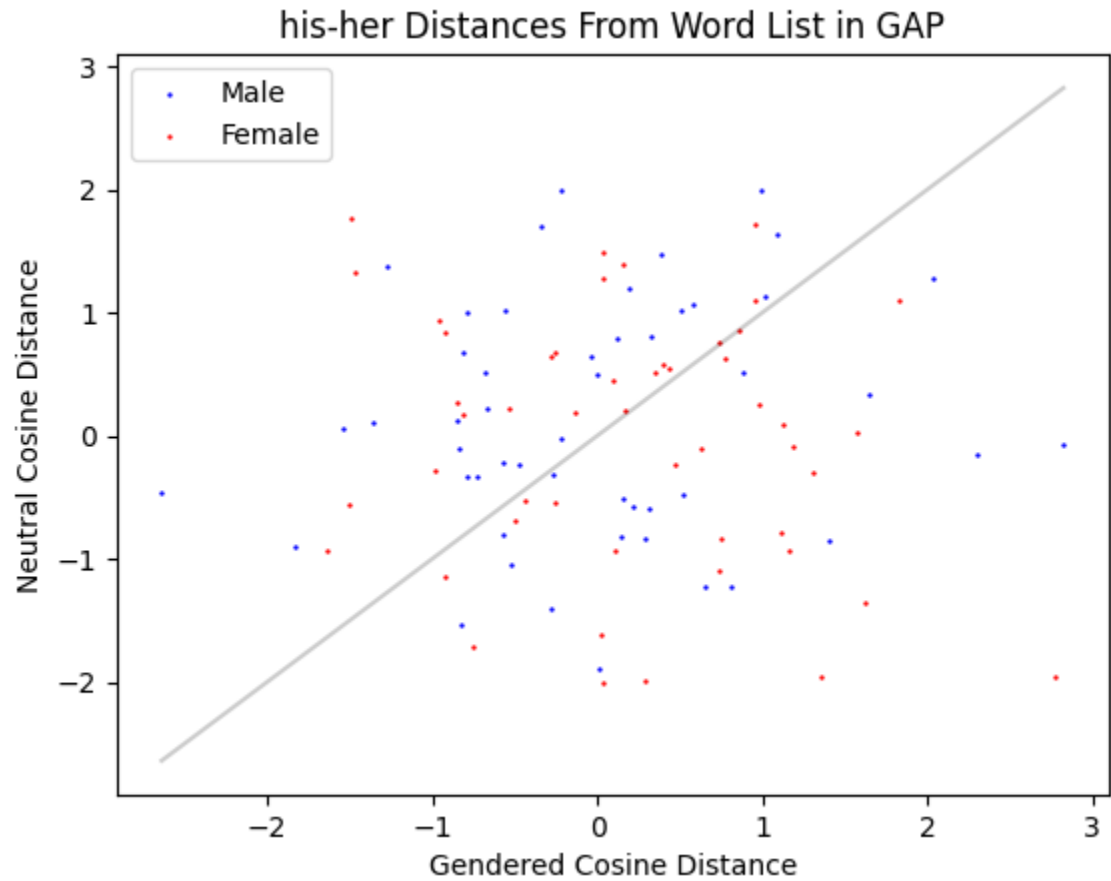


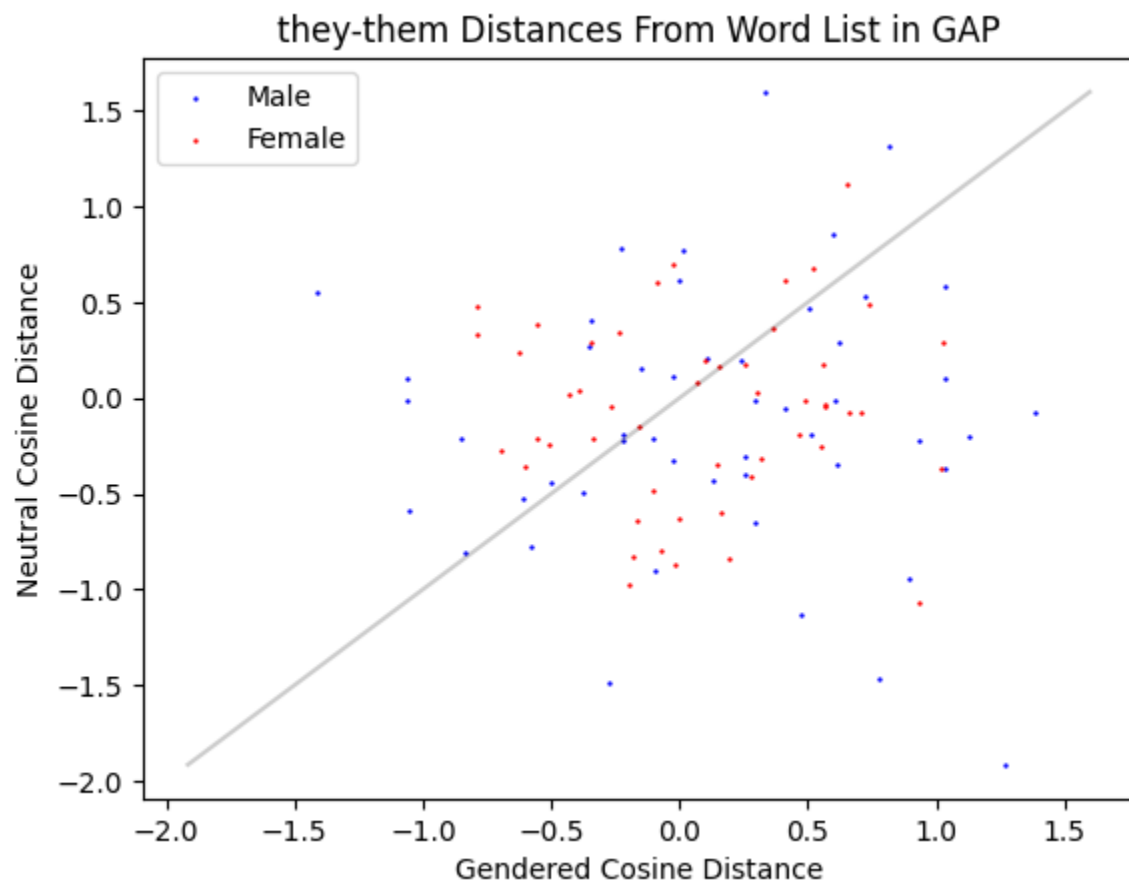
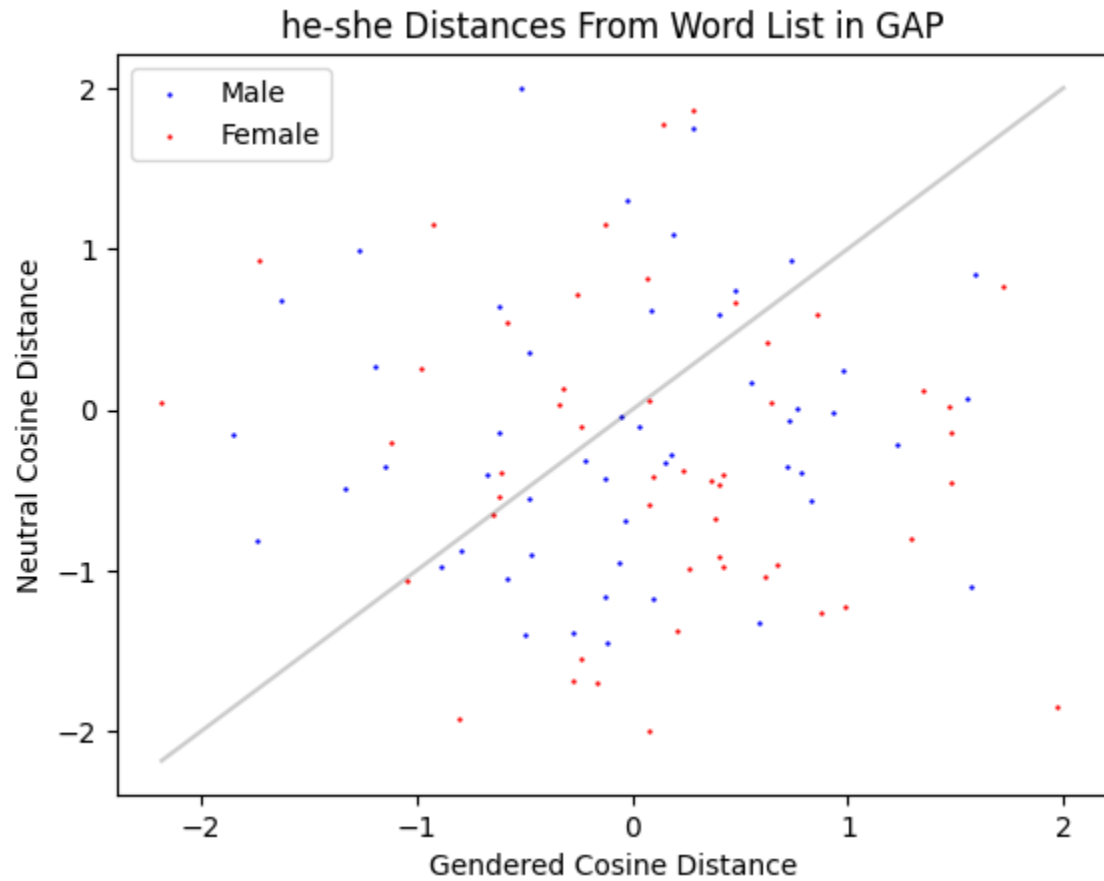


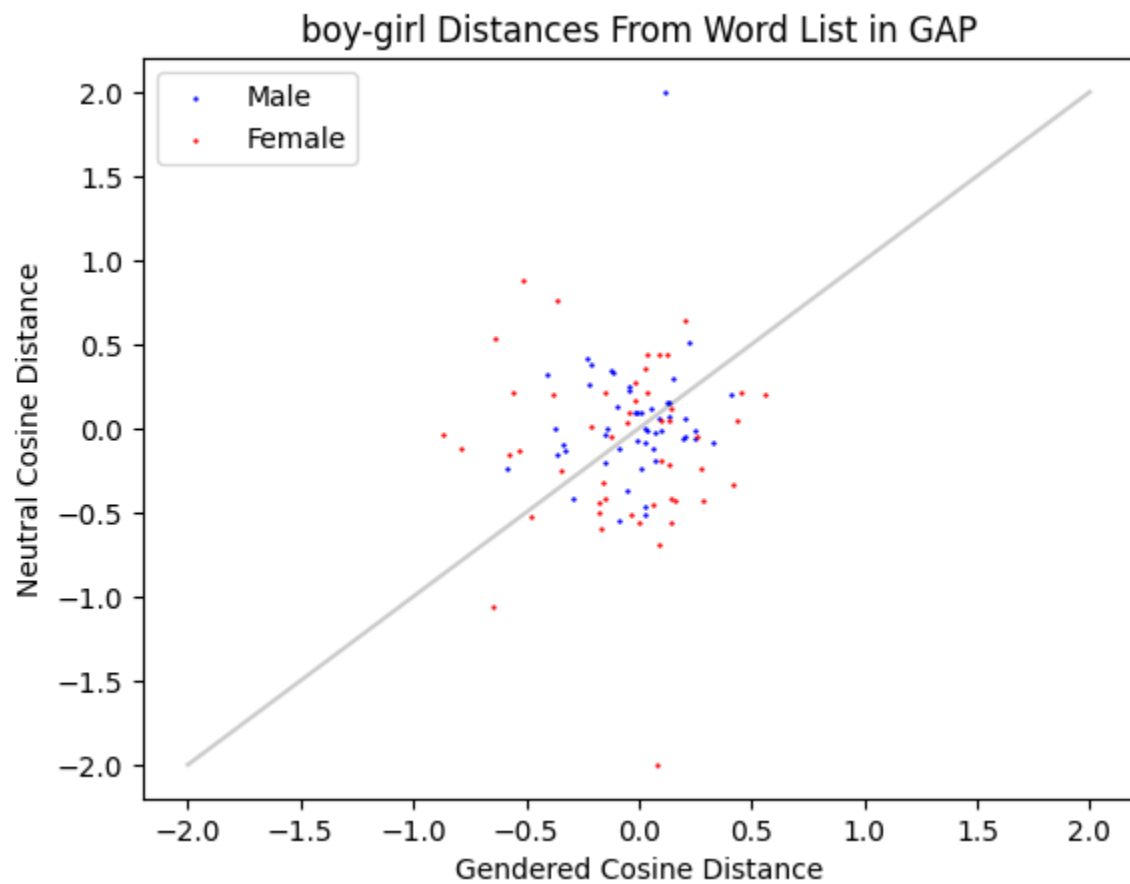
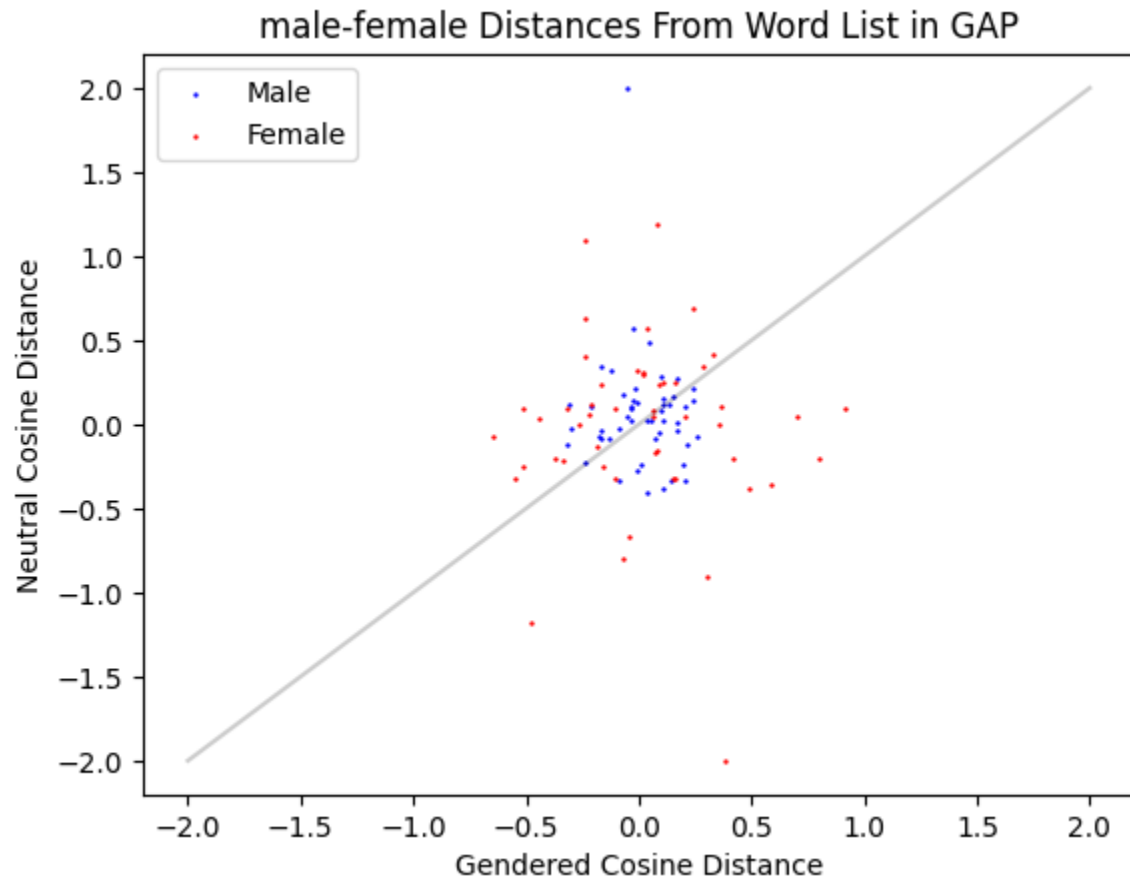


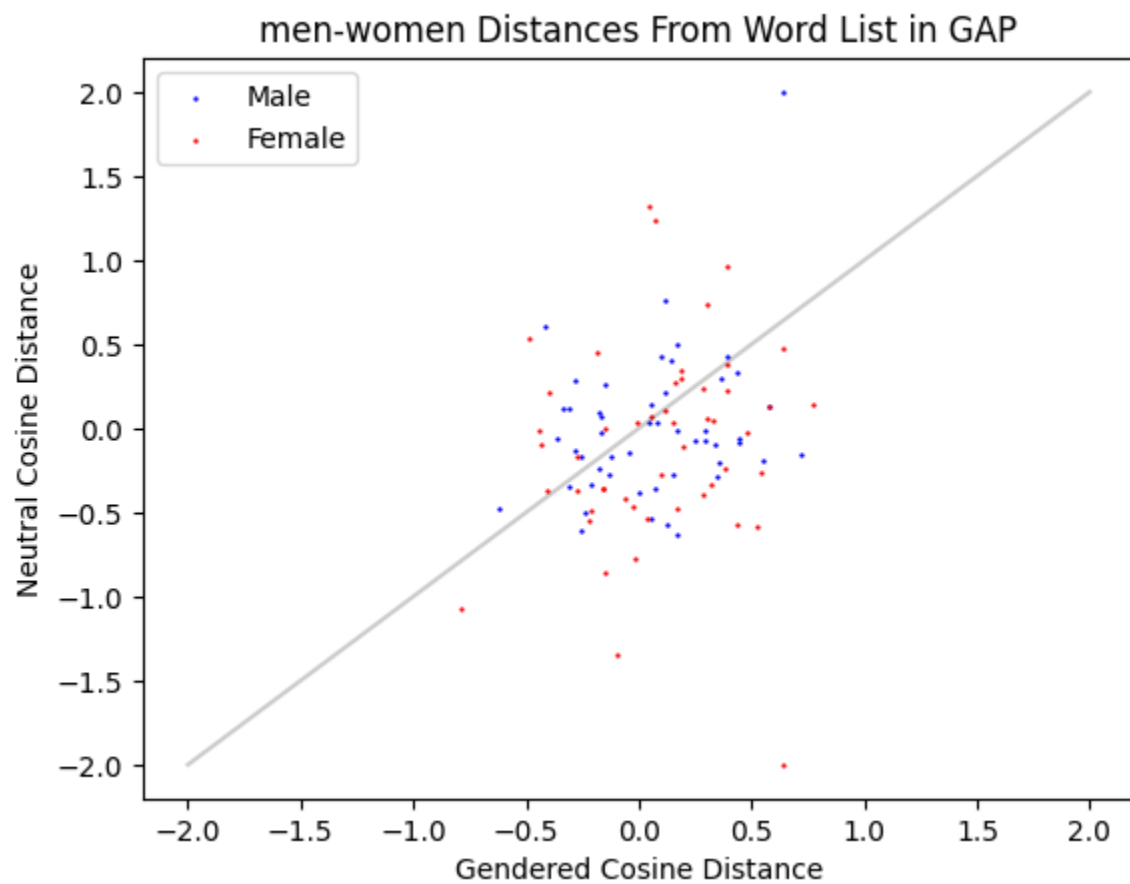
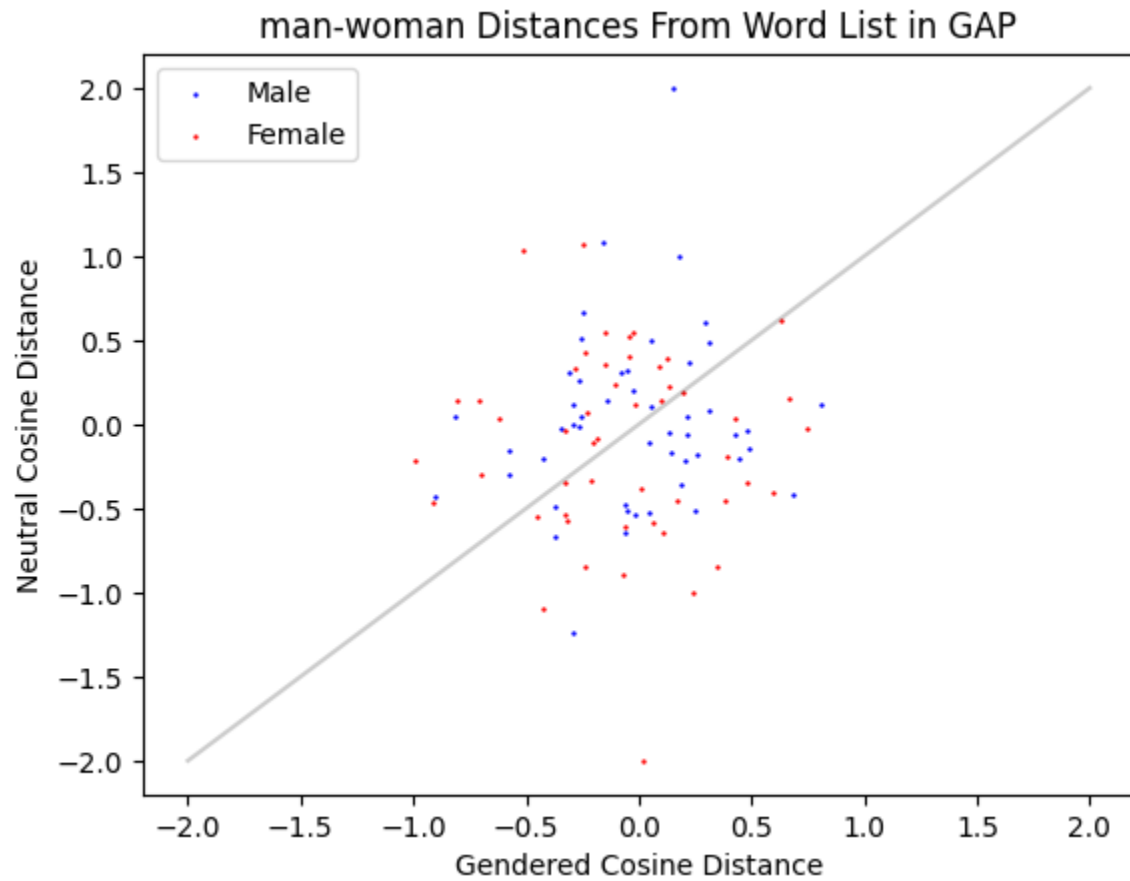


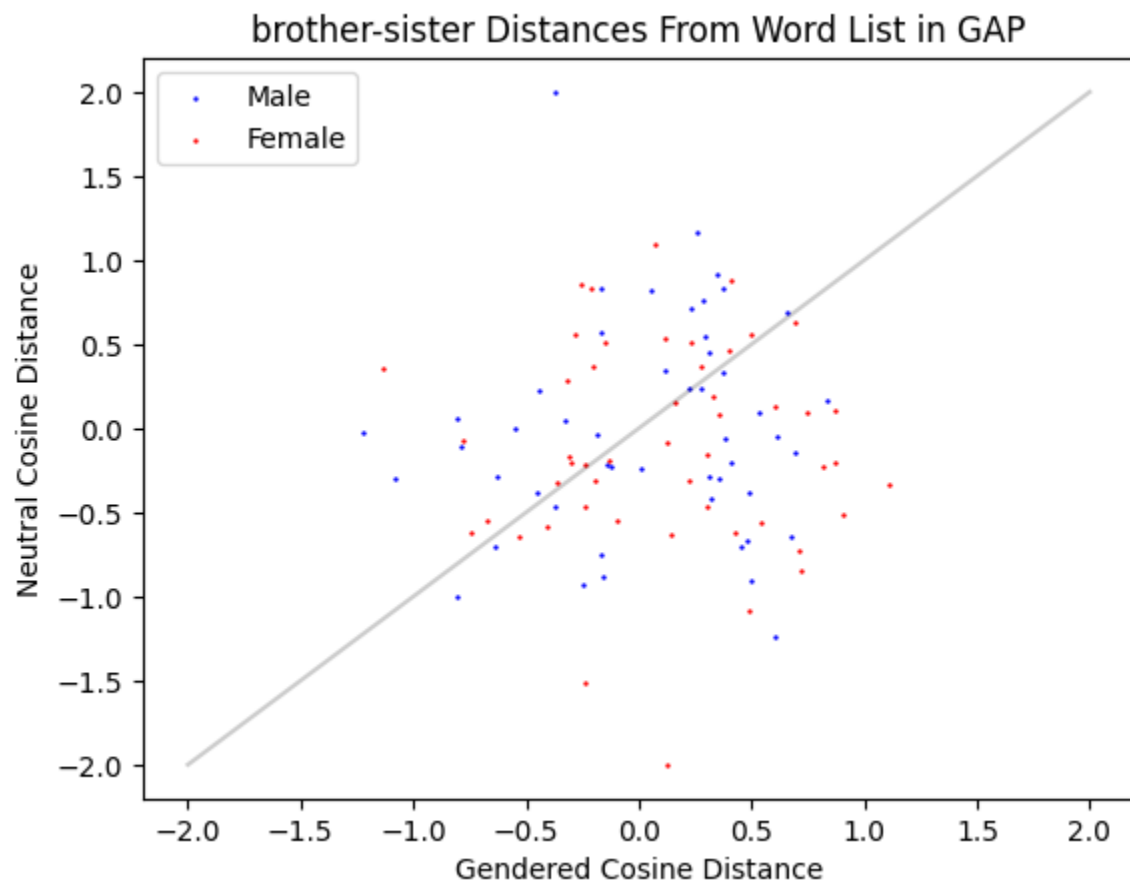
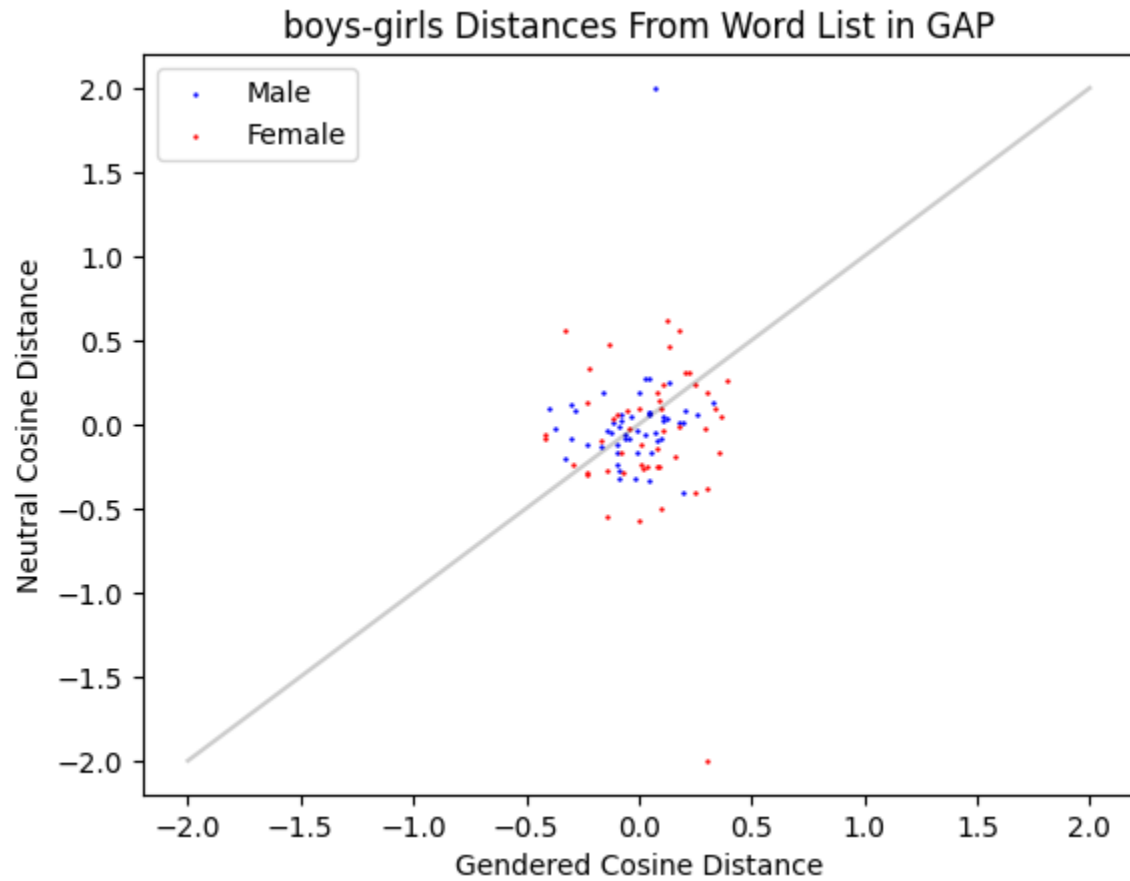
```
In [ ]: a.plotter(genderPairs, spotlightWords, gapNEmbeds, gapGEmbeds, titles[2])
```

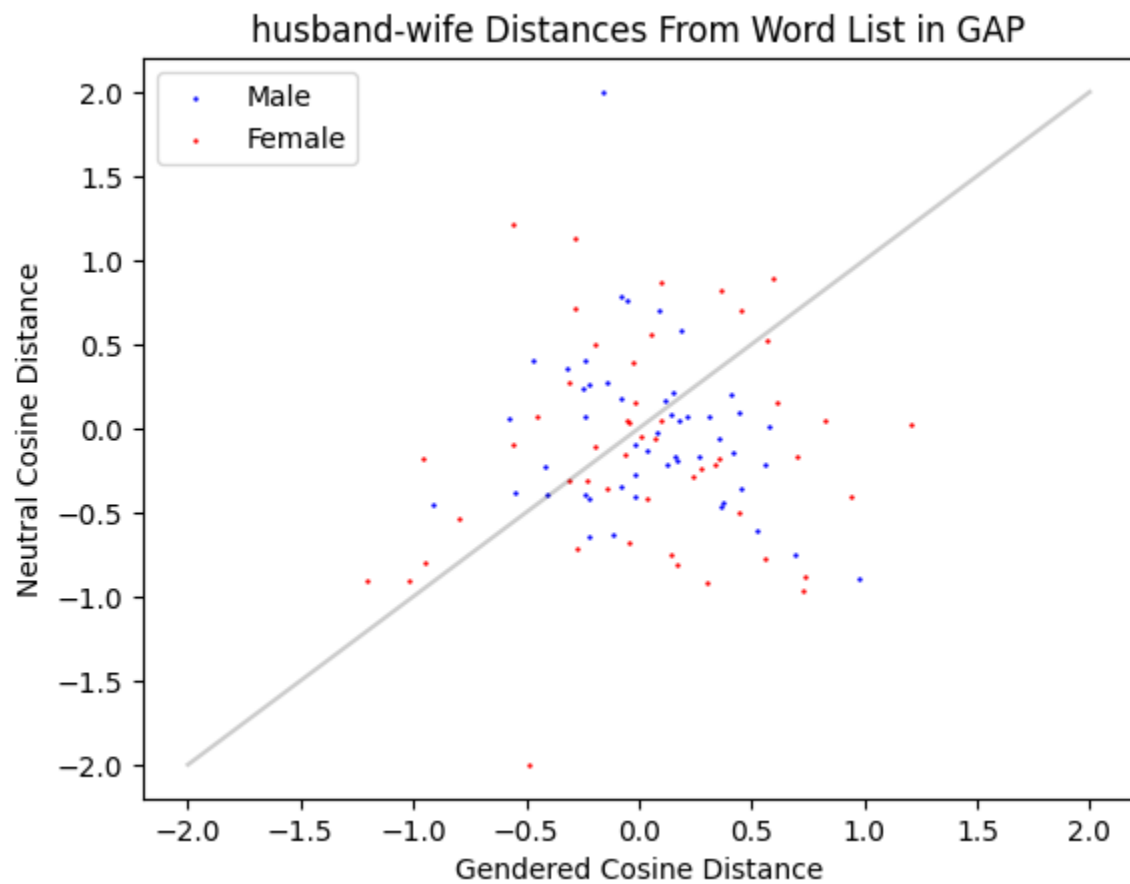
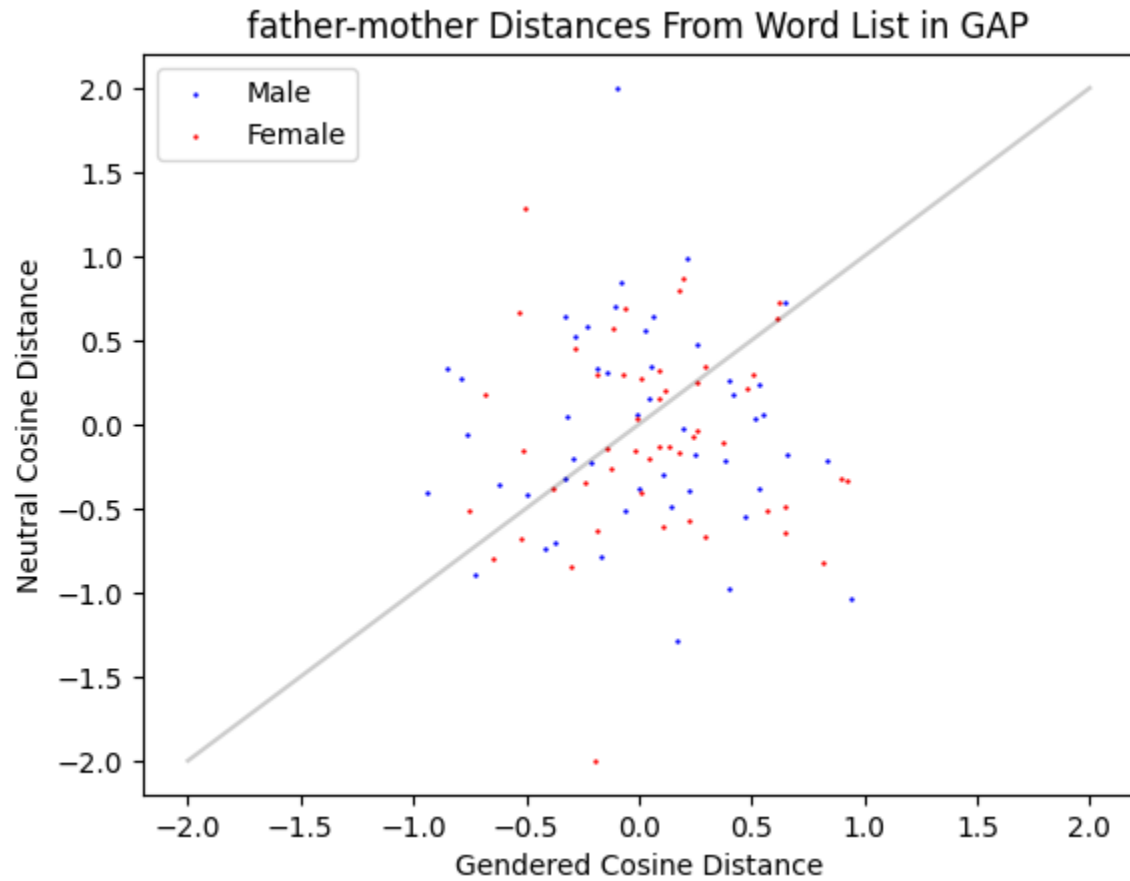


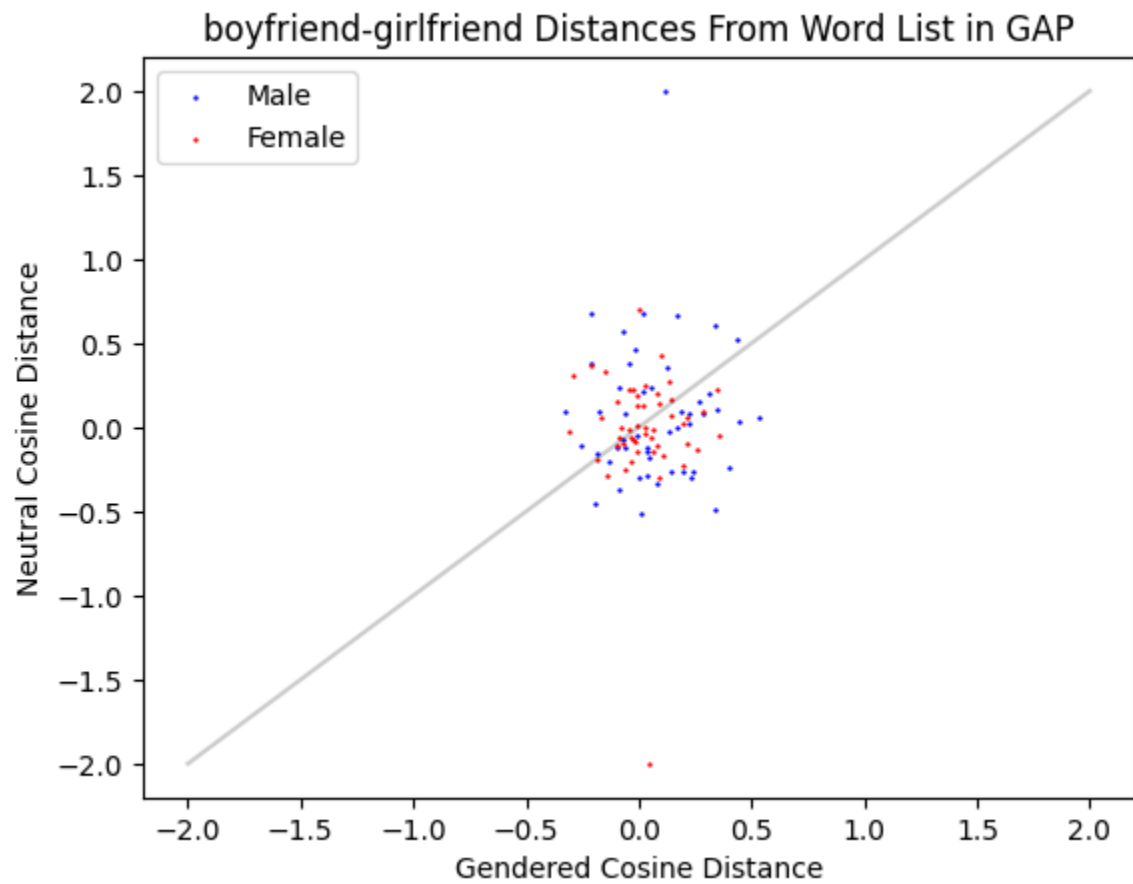




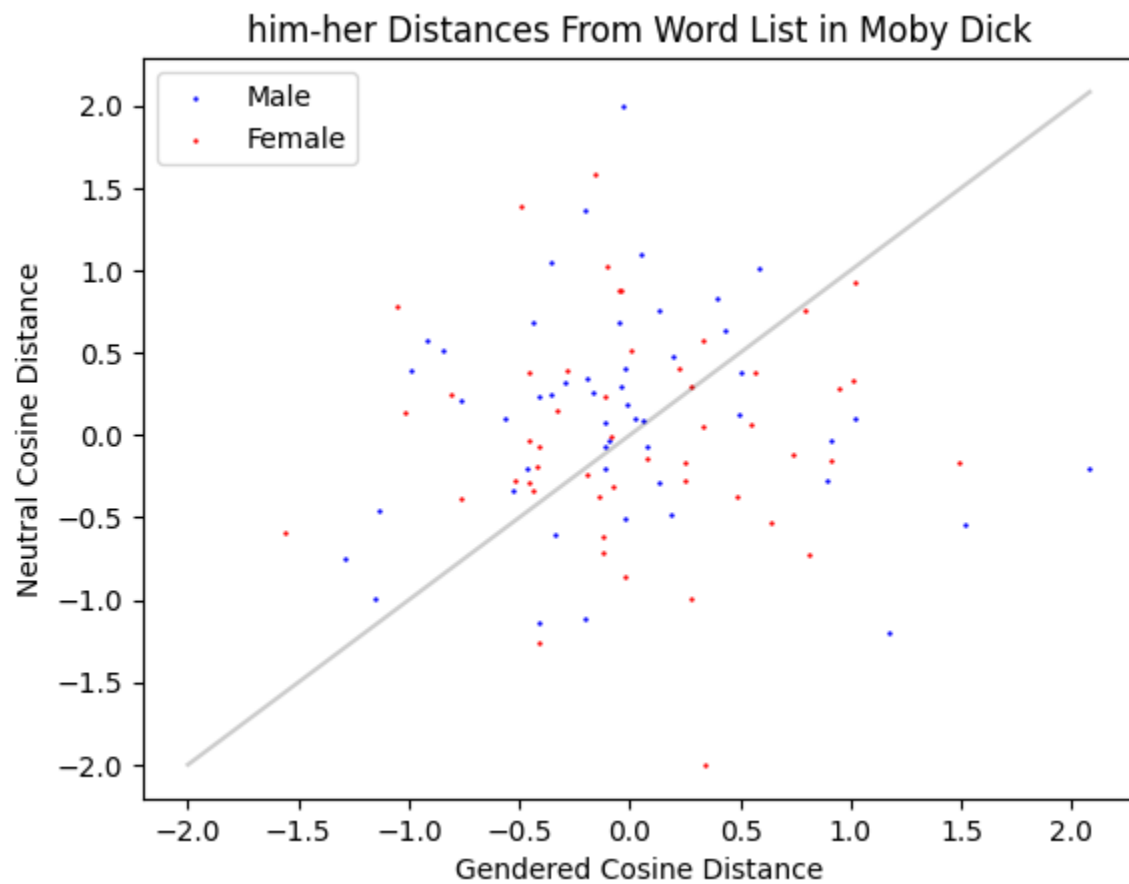
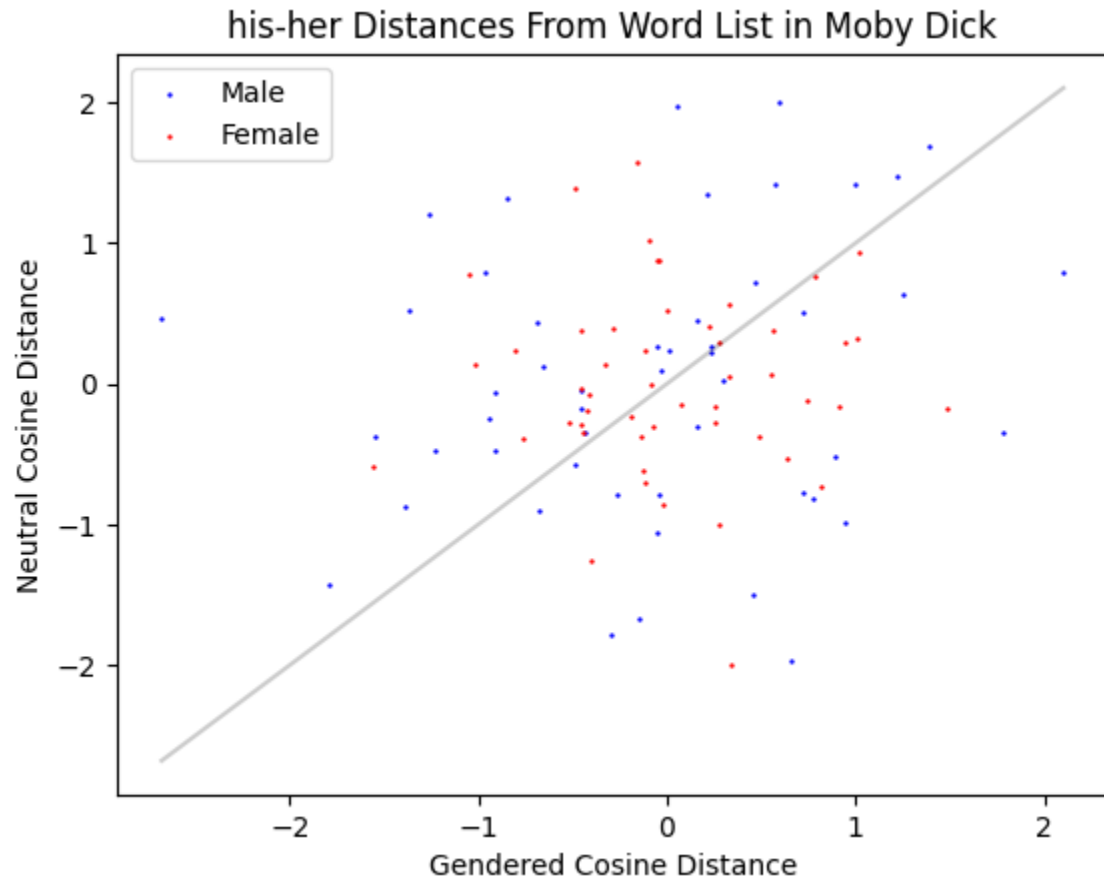


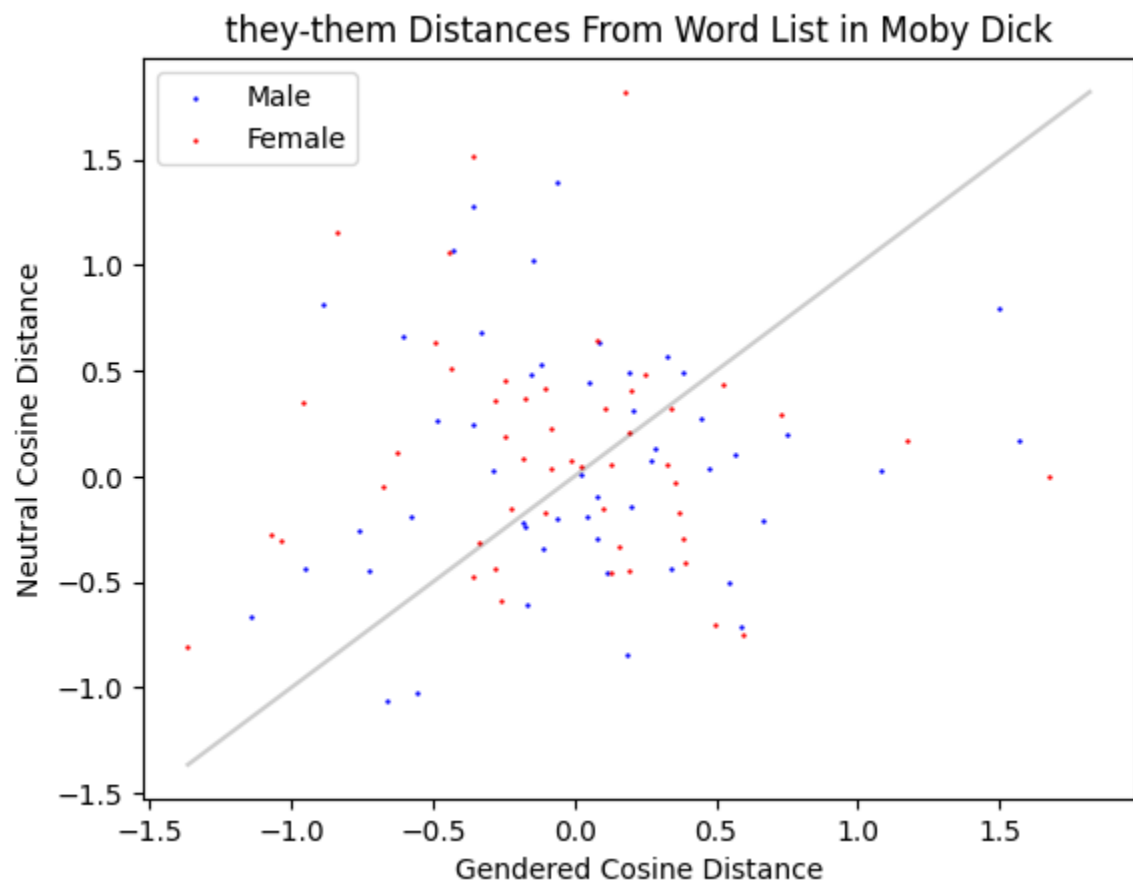
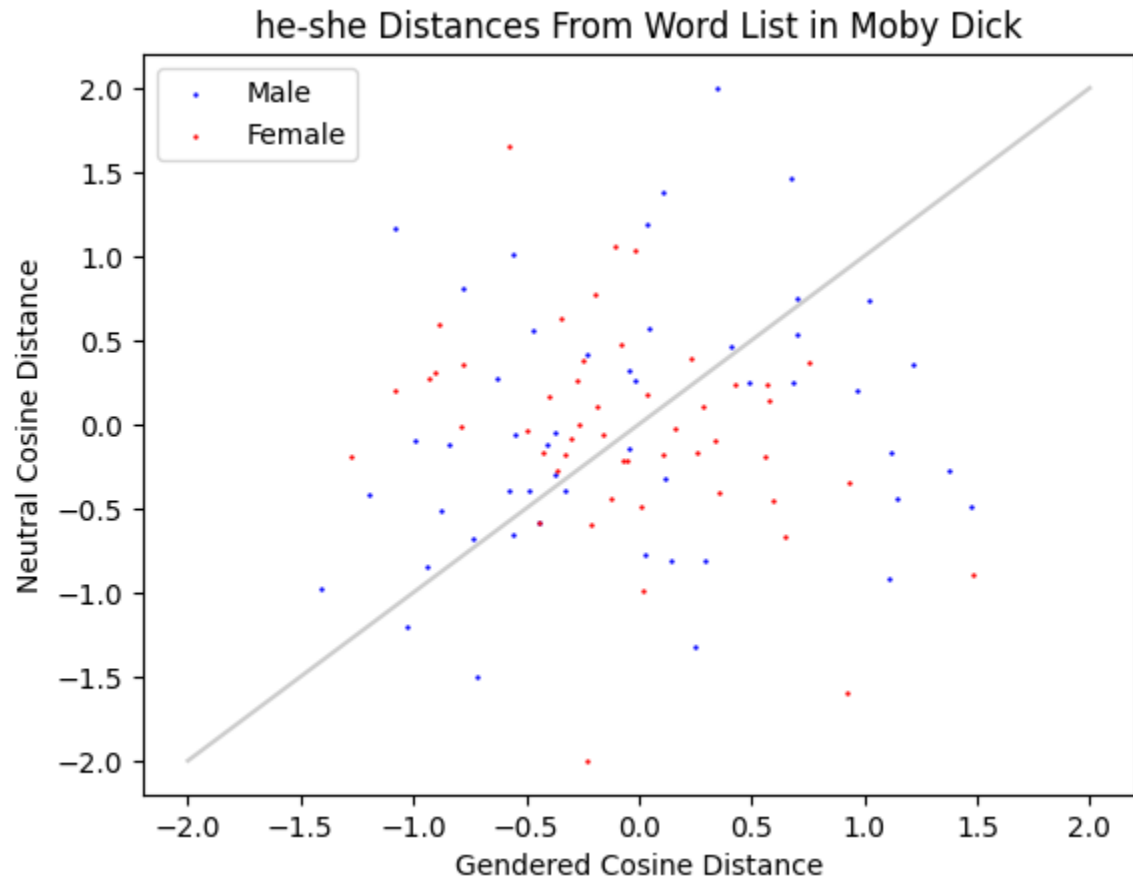


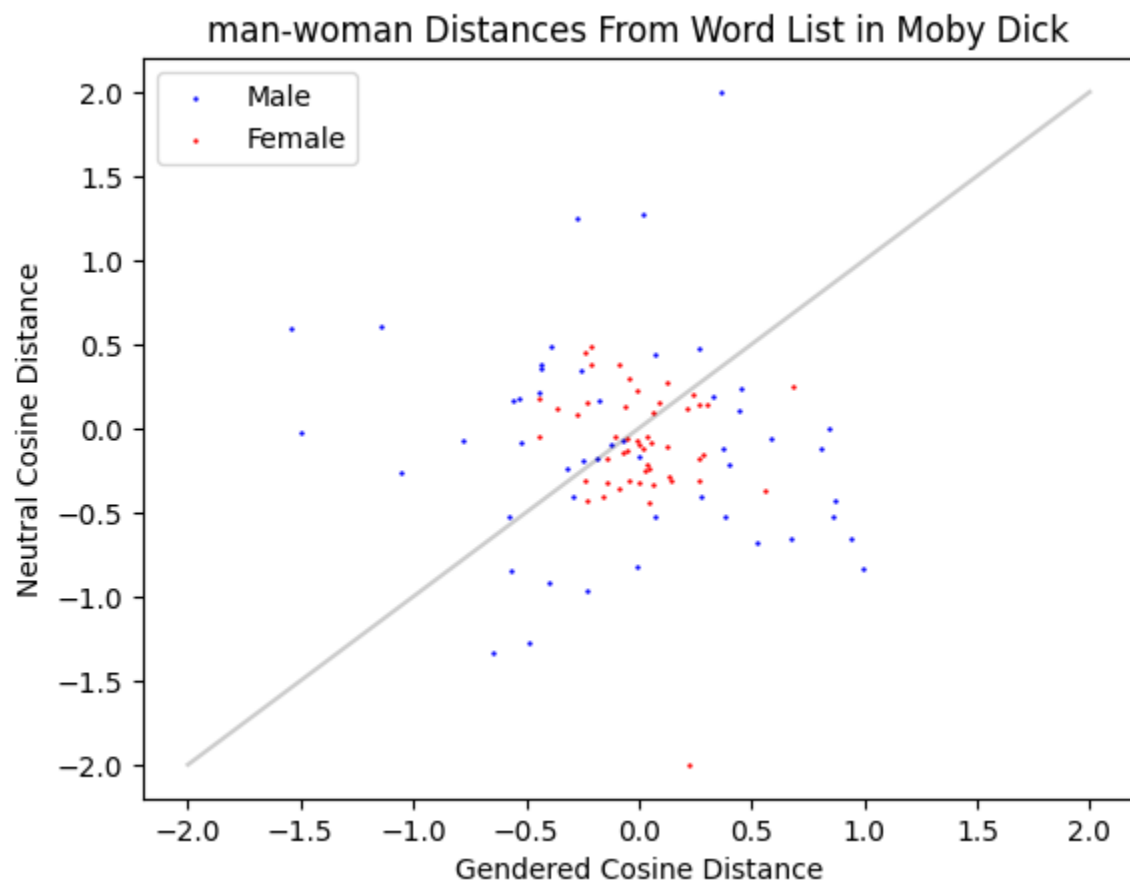
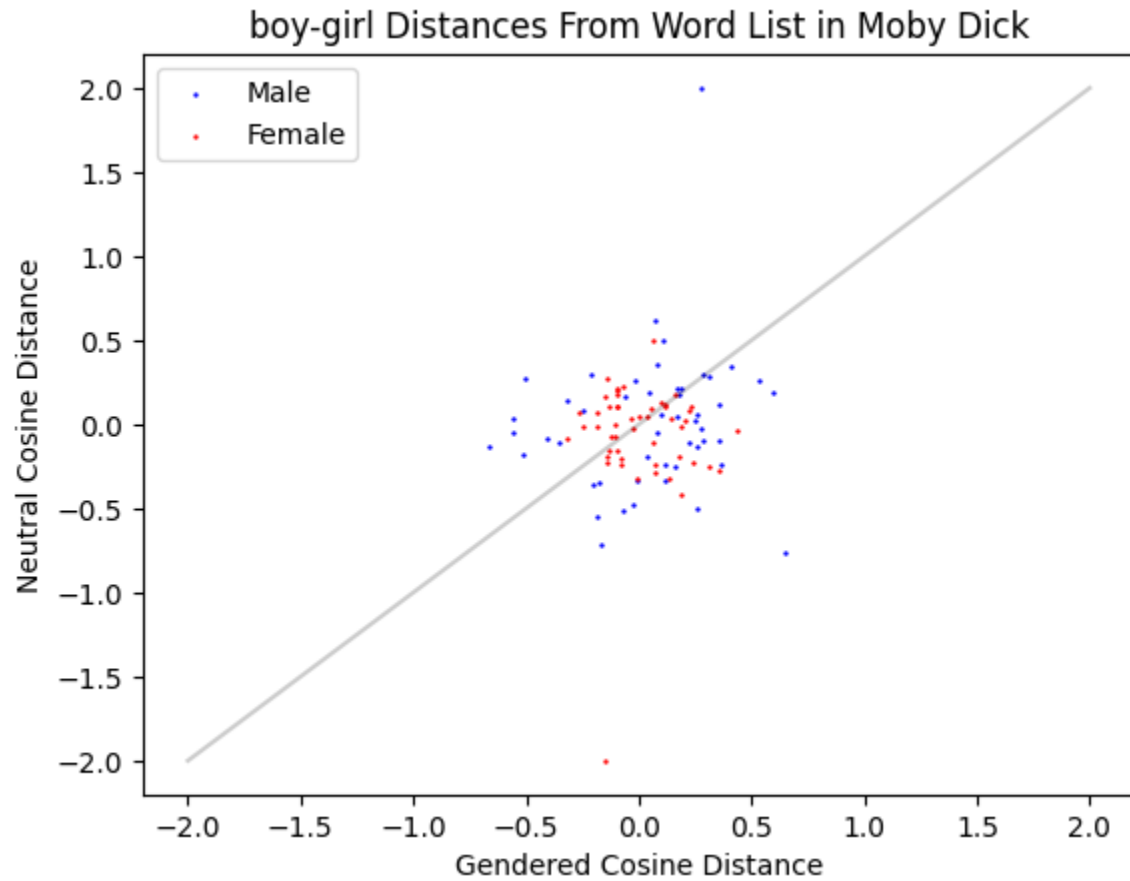


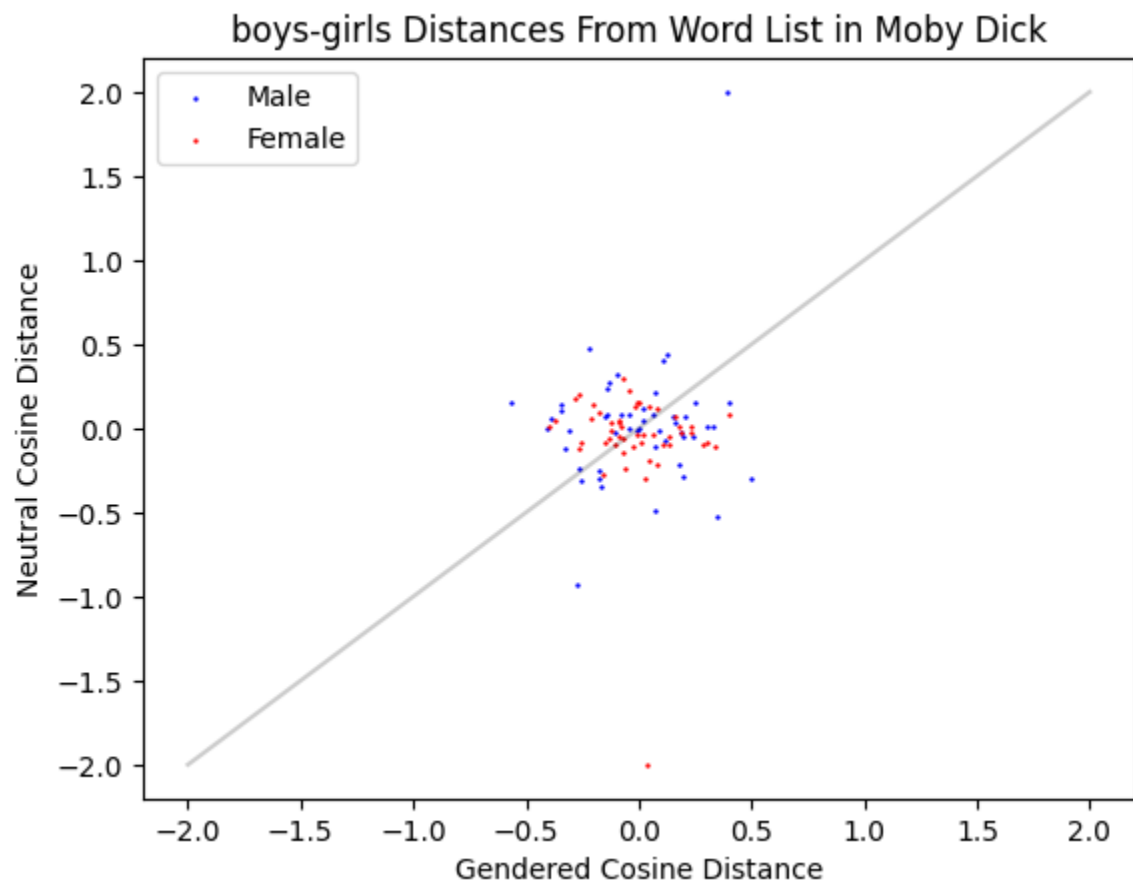
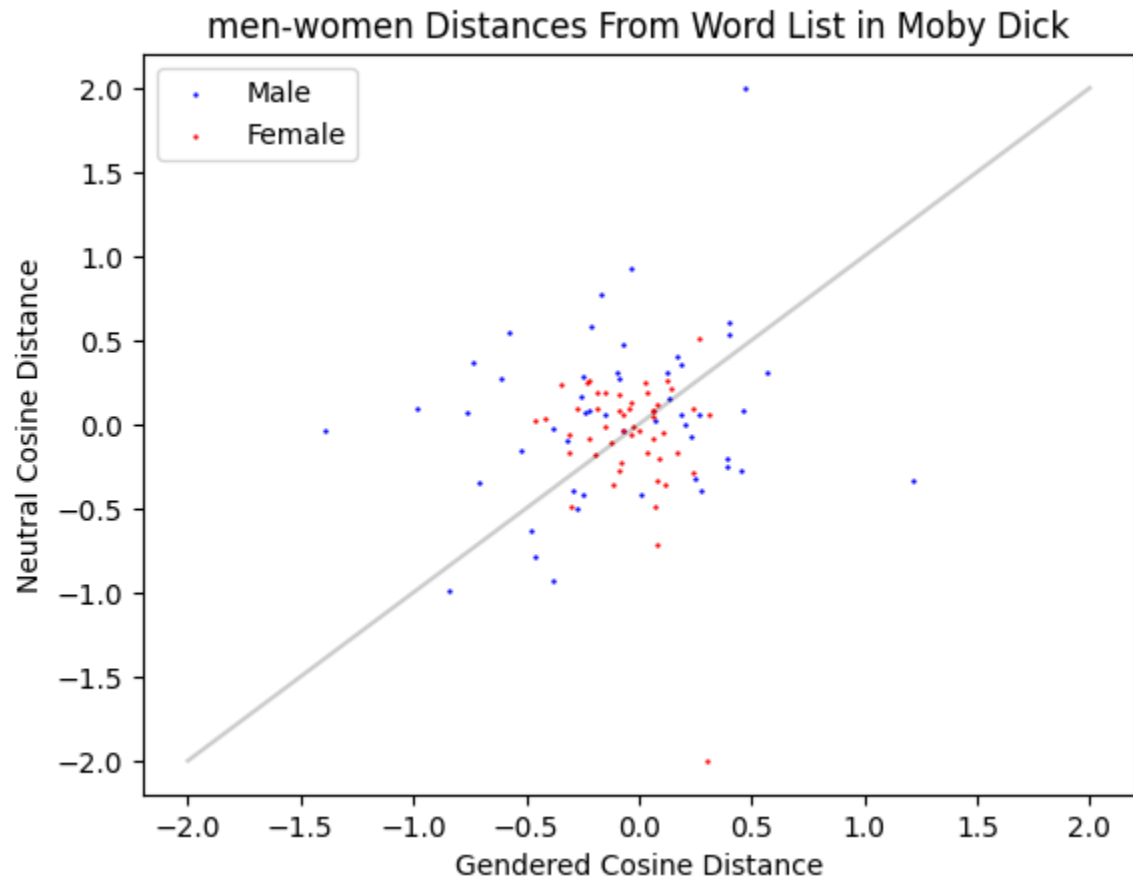


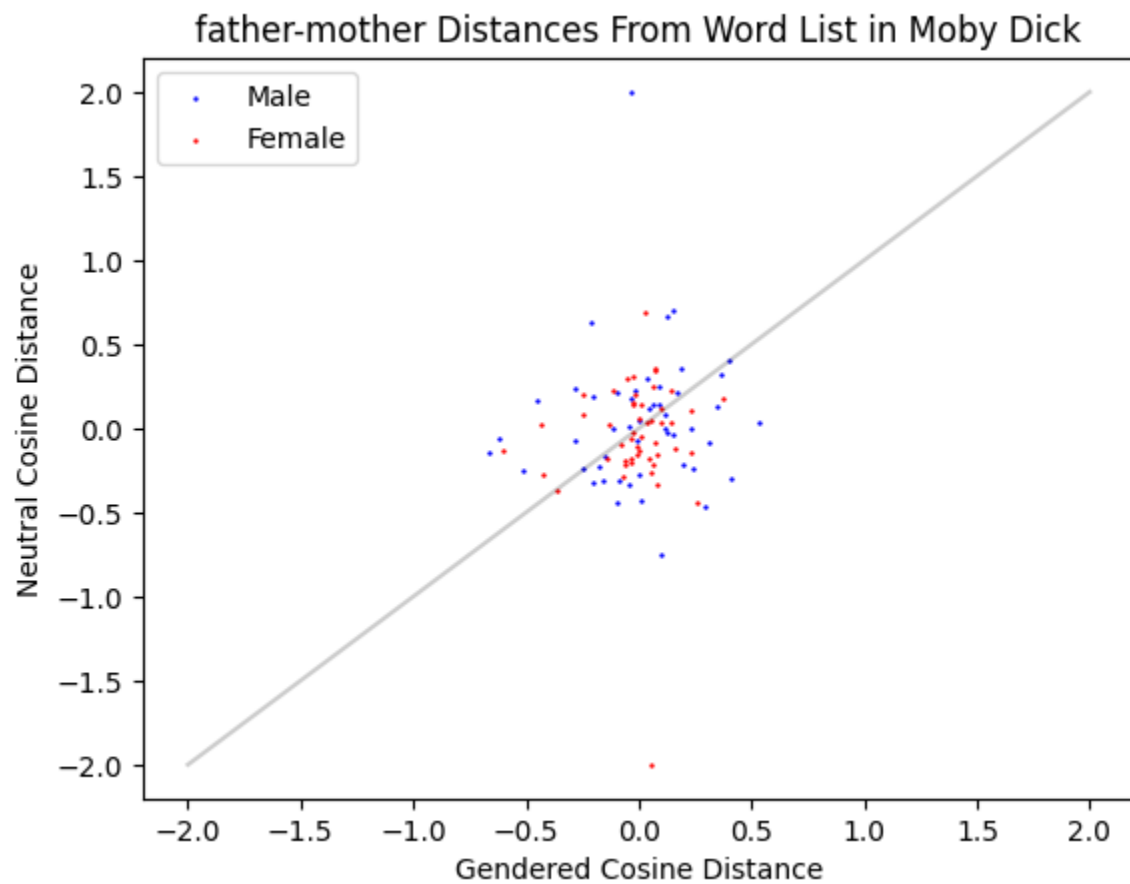
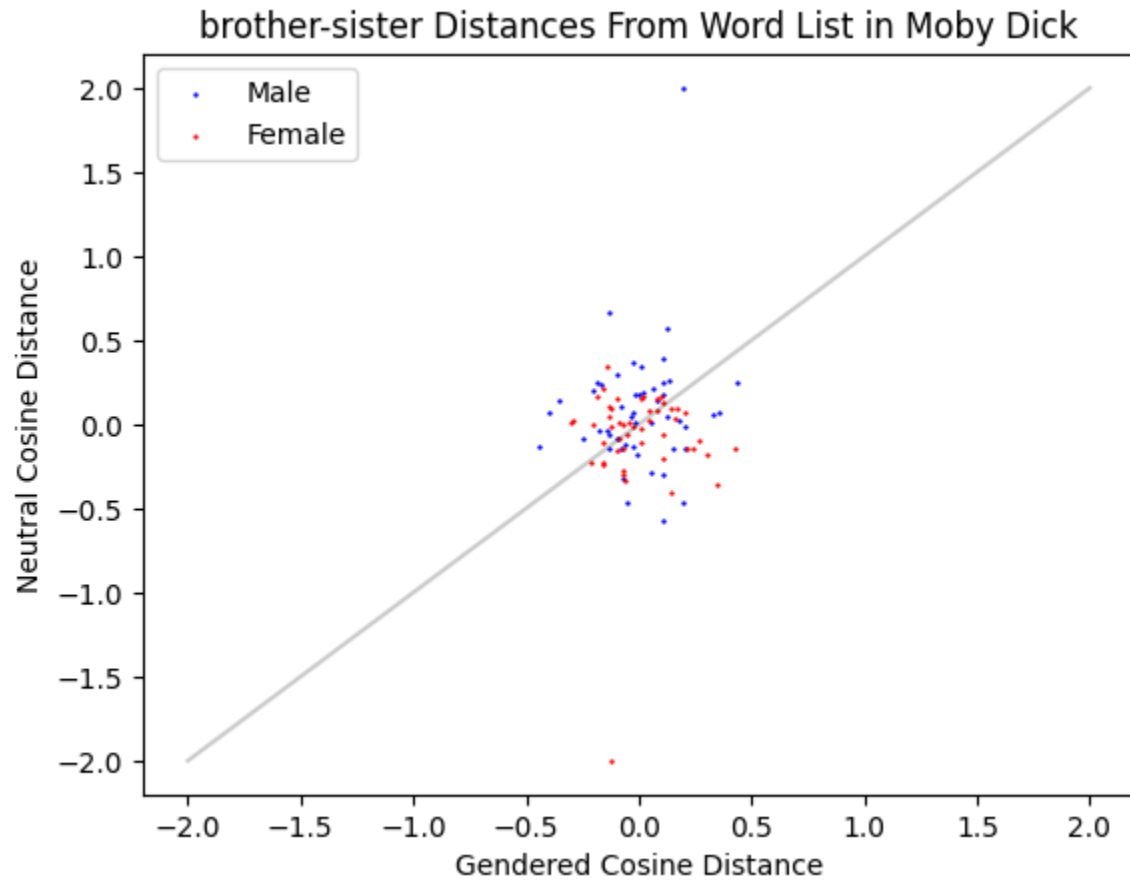
```
In [ ]: a.plotter(genderPairs, spotlightWords, mobyNEmbeds, mobyGEmbeds, titles[3])
```

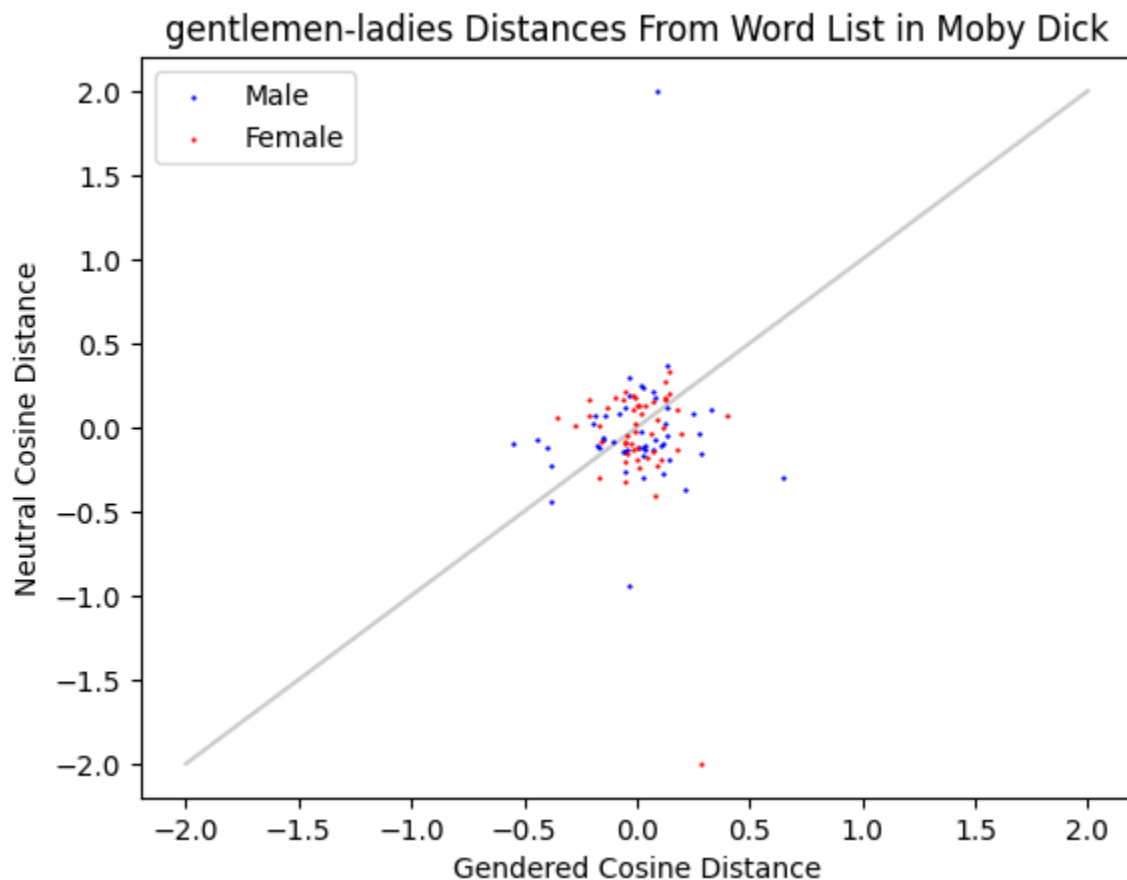












```
In [ ]: """for i, (gen, neu) in enumerate(list(zip(genderEmbeds, neutralEmbeds))):
    avg = 0
    count = 0
    for word in spotlightWords:
        if word in gen.keys() and word in neu.keys():
            count += 1
            top3gen = set(a.closestWordsNorm(gen, gen[word], 110))
            top3neu = set(a.closestWordsNorm(neu, neu[word], 110))

            top3set = top3gen.union(top3neu)

            diff = round((len(top3set) - len(top3gen))/100, 3)
            avg += diff
    avg = round(avg / count, 3)
    print(f'Top 3 rank change for {titles[i]}: {avg}')"""

#Doesn't work right

for i, (gen, neu, title) in enumerate(list(zip(genderEmbeds, neutralEmbeds, titles))):
    print(f'{title} Cosine Differences between Neutral and Gendered Embeddings')
    diff = a.closestWordsDistances(spotlightWords, gen, neu, 10)
    print(diff)
```

Bible Cosine Differences between Neutral and Gendered Embeddings

0.158

BUG Cosine Differences between Neutral and Gendered Embeddings

-0.06

GAP Cosine Differences between Neutral and Gendered Embeddings

0.001

Moby Dick Cosine Differences between Neutral and Gendered Embeddings

0.065