In [355]:

```python
import numpy as np
import pandas as pd
import math
import random
import matplotlib.pyplot as plt
```

In [373]:

```python
# Train, test, and cache the basic classifier for Obfuscation HW
import pandas
import sklearn
import argparse
from sklearn.metrics import accuracy_score
from nltk import tokenize
import pickle

def get_preds(cache_name, test):
    m,v = pickle.load(open(cache_name, 'rb'))
    test = [" ".join(tokenize.word_tokenize(t)) for t in test]
    test_data_features = v.transform(test)
    preds = m.predict(test_data_features)
    return preds
```

Testing with the original dataset

In [445]:

```python
test_data = pandas.read_csv('/Users/nehakardam/Documents/UWclasses /CSE

cache_name = 'gender_classifier.pickle'
test_preds = get_preds(cache_name, list(test_data["post_text"]))
gold_test = list(test_data["op_gender"])
gaccuracy0 = accuracy_score(list(test_preds), gold_test)
print("Gender classification accuracy", accuracy_score(list(test_preds)

cache_name = 'subreddit_classifier.pickle'
test_preds = get_preds(cache_name, list(test_data["post_text"]))
gold_test = list(test_data["subreddit"])
saccuracy0 = accuracy_score(list(test_preds), gold_test)
print("Subreddit accuracy", accuracy_score(list(test_preds), gold_test)
```

```
Gender classification accuracy 0.6495

/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator LogisticRegression from version 1.0.2 wh
en using version 0.24.2. This might lead to breaking code or invalid resu
lts. Use at your own risk.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator CountVectorizer from version 1.0.2 when
using version 0.24.2. This might lead to breaking code or invalid result
s. Use at your own risk.
  warnings.warn(

Subreddit accuracy 0.8585
```

In [375]:

```
#Male and female word test
# male =open(r'/Users/nehakardam/Documents/UWclasses /CSE NLP/A5/male.txt
# female = open(r'/Users/nehakardam/Documents/UWclasses /CSE NLP/A5/femal

male = pd.read_csv (r'/Users/nehakardam/Documents/UWclasses /CSE NLP/A5/m
male.to_csv (r'/Users/nehakardam/Documents/UWclasses /CSE NLP/A5/male.csv
female = pd.read_csv (r'/Users/nehakardam/Documents/UWclasses /CSE NLP/A5
female.to_csv (r'/Users/nehakardam/Documents/UWclasses /CSE NLP/A5/female


#Main dataset from reddit post
df = pd.read_csv('/Users/nehakardam/Documents/UWclasses /CSE NLP/A5/datas
df.head()
```

```
<ipython-input-375-b0b4f34e6690>:5: ParserWarning: Falling back to the 'p
ython' engine because the 'c' engine does not support regex separators (s
eparators > 1 char and different from '\s+' are interpreted as regex); yo
u can avoid this warning by specifying engine='python'.
  male = pd.read_csv (r'/Users/nehakardam/Documents/UWclasses /CSE NLP/A
5/male.txt', sep='delimiter', header=None)
<ipython-input-375-b0b4f34e6690>:7: ParserWarning: Falling back to the 'p
ython' engine because the 'c' engine does not support regex separators (s
eparators > 1 char and different from '\s+' are interpreted as regex); yo
u can avoid this warning by specifying engine='python'.
  female = pd.read_csv (r'/Users/nehakardam/Documents/UWclasses /CSE NLP/
A5/female.txt', sep='delimiter', header=None)
```

Out[375]:

	Unnamed: 0	op_id	op_gender	post_id	post_text	subreddit	op_gender_visi
0	1200978	MexicanSpaceProgram	M	1200978	It really comes down to the circumstances unde...	relationships	Fa
1	747542	urmyheartBeatStopR	M	747542	S.Korea, Japan, & China have tons of boy bands...	funny	Fa
2	721771	MadHatter69	M	721771	Those eyes.	funny	Fa
3	727114	on_the_redpill	M	727114	you need shades (Its not my fault if you keep...	funny	Fa
4	737662	oranjeeleven	M	737662	Nope.	funny	Fa

Model1

First, build a baseline obfuscation model: For each post in dataset.csv, if the post was written by a man (“M”) and it contains words from male.txt, replace these words with a random word from female.txt. Obfuscate posts written by women (“W”) in the same way (i.e., by replacing words from female.txt with random words from male.txt). Test classify.py on your obfuscated data and report what happens to the two accuracy measurements discussed above.

In [376]:

```
dict_male = {}
dict_female= {}
list_male = []
list_female = []
for index, value in male[0].items():
    dict_male[value] = "M"
    list_male.append(value)
for index, value in female[0].items():
    dict_female[value] = "F"
    list_female.append(value)
```

In [ ]:

```
list_male
```

In [377]:

```
len(list_female)
```

Out[377]:

```
3000
```

In [378]:

```
import random

def anonymize(words, gender):
    res = []
    if gender == "M":
        for word in words:
            if word in dict_male:
                r_index = random.randint(1, 3000) - 1
#                 print(list_female[r_index])
                res.append(list_female[r_index])
            else:
                res.append(word)
    else:
        for word in words:
            if word in dict_female:
                r_index = random.randint(1, 3000) - 1
#                 print(list_male[r_index])
                res.append(list_male[r_index])
            else:
                res.append(word)

#     print ("TESTING:", len(words), len(res))
    return res


df_copy = df.copy()
for i in range (df.shape[0]):
#     print("ORIGINAL: ", df_copy["post_text"][i])
#     print("ANONYMIZED: ", anonymize(df["post_text"][i], df["op_gender
    df_copy["post_text"][i] = anonymize(df["post_text"][i], df["op_gend

```

```
<ipython-input-378-6926280320ab>:30: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  df_copy["post_text"][i] = anonymize(df["post_text"][i], df["op_gender"]
[i])
```

In [379]:

```
df_copy.to_csv('/Users/nehakardam/Documents/UWclasses /CSE NLP/A5/datas
```

In [435]:

```
test_data = pandas.read_csv('/Users/nehakardam/Documents/UWclasses /CSE

cache_name = 'gender_classifier.pickle'
test_preds = get_preds(cache_name, list(test_data["post_text"]))
gold_test = list(test_data["op_gender"])
gaccuracy1 = accuracy_score(list(test_preds), gold_test)
print("Gender classification accuracy", accuracy_score(list(test_preds)

cache_name = 'subreddit_classifier.pickle'
test_preds = get_preds(cache_name, list(test_data["post_text"]))
gold_test = list(test_data["subreddit"])
saccuracy1 = accuracy_score(list(test_preds), gold_test)
print("Subreddit accuracy", accuracy_score(list(test_preds), gold_test)
```

```
/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator LogisticRegression from version 1.0.2 wh
en using version 0.24.2. This might lead to breaking code or invalid resu
lts. Use at your own risk.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator CountVectorizer from version 1.0.2 when
using version 0.24.2. This might lead to breaking code or invalid result
s. Use at your own risk.
  warnings.warn(

Gender classification accuracy 0.289

/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator LogisticRegression from version 1.0.2 wh
en using version 0.24.2. This might lead to breaking code or invalid resu
lts. Use at your own risk.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator CountVectorizer from version 1.0.2 when
using version 0.24.2. This might lead to breaking code or invalid result
s. Use at your own risk.
  warnings.warn(

Subreddit accuracy 0.517
```

Model 2

Second, improve your obfuscation model: Instead of replacing words from male.txt with randomly chosen words from female.txt, choose a semantically similar word from female.txt. Do the same in reverse. You may use any metric you like for identifying semantically similar words, but you should explain why you chose it. We recommend starting with cosine distance between pretrained word embeddings (available, for example, here) Test classify.py on data obfuscated using your improved model and analyze the results. The classifier should perform close to random at identifying gender and should obtain at least 79% accuracy on classifying the subreddit.

In [382]:

```python
import torchtext
import torch
import torch.nn as nn
from torch.autograd import Variable
import matplotlib.pyplot as plt
import numpy as np
from torchtext.vocab import Vectors
from tqdm.notebook import tqdm

# The first time you run this will download a ~823MB file
glove = torchtext.vocab.GloVe(name="6B", # trained on Wikipedia 2014 co
                              dim=100)    # embedding size = 50
```

In [383]:

```python
x = glove['england']
y = glove['beer']
torch.cosine_similarity(glove['england'].unsqueeze(0),
                        glove['beer'].unsqueeze(0))
```

Out[383]:

```
tensor([0.2118])
```

In [384]:

```python
word = 'dog'
other = ['cat', 'puppy', 'kitten', 'mouse', 'kite', 'lion', 'doggy']
for w in other:
    dist = torch.norm(glove[word] - glove[w]) # euclidean distance
    print(w, float(dist))
```

```
cat 2.6811304092407227
puppy 3.9500551223754883
kitten 5.06204080581665
mouse 5.034541130065918
kite 6.637244701385498
lion 5.573644638061523
doggy 6.244095802307129
```

In [385]:

```
import random

def closest_word_in_female(word):
    closest_word = ""
    min_distance = 100000
    for w in list_female:
        dis = torch.norm(glove[word] - glove[w])
        if(dis < min_distance):
            min_distance = dis
            closest_word = w
    return closest_word


def closest_word_in_male(word):
    closest_word = ""
    min_distance = 100000
    for w in list_male:
        dis = torch.norm(glove[word] - glove[w])
        if(dis < min_distance):
            min_distance = dis
            closest_word = w
    return closest_word

def anonymize2(words, gender):
    res = []
    if gender == "M":
        for word in words:
            if word in dict_male:
                res.append(closest_word_in_female(word))
            else:
                res.append(word)
    else:
        for word in words:
            if word in dict_female:
                res.append(closest_word_in_male(word))
            else:
                res.append(word)

    print ("TESTING:", len(words), len(res))
    return res

```

In [386]:

```
df_copy_model2 = df.copy()
for i in range (df.shape[0]):
#     print("ORIGINAL: ", df_copy["post_text"][i])
#     print("ANONYMIZED: ", anonymize(df["post_text"][i], df["op_gender
    df_copy_model2["post_text"][i] = anonymize2(df["post_text"][i], df[
```

```
TESTING: 195 195
TESTING: 325 325
TESTING: 36 36
TESTING: 29 29
TESTING: 172 172
TESTING: 164 164
TESTING: 39 39
TESTING: 245 245
TESTING: 49 49
TESTING: 86 86
TESTING: 2202 2202
TESTING: 324 324
TESTING: 363 363
TESTING: 552 552
TESTING: 170 170
TESTING: 200 200
TESTING: 224 224
TESTING: 128 128
TESTING: 346 346
TESTING: 68 68
```

In [388]:

```
df_copy_model2.to_csv('/Users/nehakardam/Documents/UWclasses /CSE NLP/A
```

In [436]:

```
test_data = pandas.read_csv('/Users/nehakardam/Documents/UWclasses /CSE

cache_name = 'gender_classifier.pickle'
test_preds = get_preds(cache_name, list(test_data["post_text"]))
gold_test = list(test_data["op_gender"])
gaccuracy2 = accuracy_score(list(test_preds), gold_test)
print("Gender classification accuracy", accuracy_score(list(test_preds)

cache_name = 'subreddit_classifier.pickle'
test_preds = get_preds(cache_name, list(test_data["post_text"]))
gold_test = list(test_data["subreddit"])
saccuracy2 = accuracy_score(list(test_preds), gold_test)
print("Subreddit accuracy", accuracy_score(list(test_preds), gold_test)
```

```
/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator LogisticRegression from version 1.0.2 wh
en using version 0.24.2. This might lead to breaking code or invalid resu
lts. Use at your own risk.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator CountVectorizer from version 1.0.2 when
using version 0.24.2. This might lead to breaking code or invalid result
s. Use at your own risk.
  warnings.warn(

Gender classification accuracy 0.082

/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator LogisticRegression from version 1.0.2 wh
en using version 0.24.2. This might lead to breaking code or invalid resu
lts. Use at your own risk.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator CountVectorizer from version 1.0.2 when
using version 0.24.2. This might lead to breaking code or invalid result
s. Use at your own risk.
  warnings.warn(

Subreddit accuracy 0.6525
```

Model3

Third, experiment with some basic modifications to your obfuscation models. For example, what if you randomly decide whether or not to replace words instead of replacing every lexicon word? What if you only replace words that have semantically similar enough counterparts?

In [450]:

```python
import random

def closest_word_in_female(word):
    closest_word = ""
    min_distance = 100000
    for w in list_female:
        dis = torch.norm(glove[word] - glove[w])
        if(dis < min_distance):
            min_distance = dis
            closest_word = w
    return closest_word


def closest_word_in_male(word):
    closest_word = ""
    min_distance = 100000
    for w in list_male:
        dis = torch.norm(glove[word] - glove[w])
        if(dis < min_distance):
            min_distance = dis
            closest_word = w
    return closest_word

def anonymize3(words, gender):
    res = []
    print(gender)
    if gender == "M":
        for word in words:
            if word in dict_male and len(word) > 2:
                res.append(closest_word_in_female(word))
            else:
                res.append(word)
    else:
        for word in words:
            if word in dict_female and len(word) > 2:
                res.append(closest_word_in_male(word))
            else:
                res.append(word)

    print ("TESTING:", len(words), len(res))
    return res

```

In [451]:

```
df_copy_model3 = df.copy()
for i in range (df.shape[0]):
#     print("ORIGINAL: ", df_copy["post_text"][i])
#     print("ANONYMIZED: ", anonymize(df["post_text"][i], df["op_gender
    df_copy_model3["post_text"][i] = anonymize3(df["post_text"][i], df[
```

```
TESTING: 60 60
M
TESTING: 20 20
M
TESTING: 512 512
M
TESTING: 67 67
M
TESTING: 101 101
M
TESTING: 632 632
M
TESTING: 145 145
M
TESTING: 22 22
M
TESTING: 12 12
M
TESTING: 169 169
M
TESTING: 414 414
```

In [452]:

```
df_copy_model3.to_csv('/Users/nehakardam/Documents/UWclasses /CSE NLP/A
```

In [453]:

```
test_data = pandas.read_csv('/Users/nehakardam/Documents/UWclasses /CSE

cache_name = 'gender_classifier.pickle'
test_preds = get_preds(cache_name, list(test_data["post_text"]))
gold_test = list(test_data["op_gender"])
gaccuracy3 = accuracy_score(list(test_preds), gold_test)
print("Gender classification accuracy", accuracy_score(list(test_preds)

cache_name = 'subreddit_classifier.pickle'
test_preds = get_preds(cache_name, list(test_data["post_text"]))
gold_test = list(test_data["subreddit"])
saccuracy3 = accuracy_score(list(test_preds), gold_test)
print("Subreddit accuracy", accuracy_score(list(test_preds), gold_test)
```

```
/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator LogisticRegression from version 1.0.2 wh
en using version 0.24.2. This might lead to breaking code or invalid resu
lts. Use at your own risk.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator CountVectorizer from version 1.0.2 when
using version 0.24.2. This might lead to breaking code or invalid result
s. Use at your own risk.
  warnings.warn(

Gender classification accuracy 0.5

/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator LogisticRegression from version 1.0.2 wh
en using version 0.24.2. This might lead to breaking code or invalid resu
lts. Use at your own risk.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarni
ng: Trying to unpickle estimator CountVectorizer from version 1.0.2 when
using version 0.24.2. This might lead to breaking code or invalid result
s. Use at your own risk.
  warnings.warn(

Subreddit accuracy 0.526
```

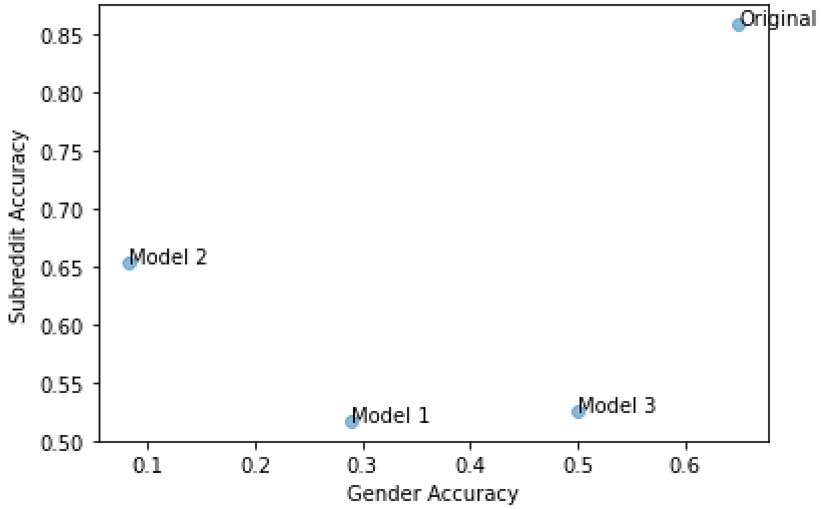
In [454]:

```
import matplotlib.pyplot as plt

x = [gaccuracy0, gaccuracy1, gaccuracy2, gaccuracy3]
y = [saccuracy0, saccuracy1, saccuracy2, saccuracy3]
n = ["Original", "Model 1", "Model 2", "Model 3"]

plt.xlabel("Gender Accuracy")
plt.ylabel("Subreddit Accuracy")

plt.scatter(x, y, alpha=0.5)
for i, txt in enumerate(n):
    plt.annotate(txt, (x[i], y[i]))
plt.show()
```



In [ ]:

```

```