In [ ]:
```python
import csv
import numpy as np
from gensim.models import Word2Vec
from scipy import spatial
```

In [1]:
```python
# process data
# from csv to list of "words"

filename = "output_obfuscated_13epoch_model.csv"

word = []
real_word = []

with open(filename, 'r') as csvfile:
    csvreader = csv.reader(csvfile)
    fields = next(csvreader)
    for row in csvreader:
        last = row[-1][1:-1]
        l = last.split('||')
        #if len(l) == 1:
            #real_word.append(l[0])
        #else:
        word.append(l)
```

In [3]:
```python
# check if similar methods are predicted to have different sequences

same = {}

with open(filename, 'r') as csvfile:
    csvreader = csv.reader(csvfile)
    fields = next(csvreader)
    for row in csvreader:
        last = row[-1][1:-1]
        label = row[1]+row[2]
        if label in same:
            if last not in same[label]:
                same[label].append(last)
        else:
            same[label] = [last]
```

In [5]:
```python
# apply word2vec

model = Word2Vec(word, min_count = 1)
model.save("word2vec.model")
```

```
C:\Users\lyq19\anaconda3\lib\site-packages\gensim\similarities\__init__.py:15: UserWar
ning: The gensim.similarities.levenshtein submodule is disabled, because the optional
Levenshtein package <https://pypi.org/project/python-Levenshtein/> is unavailable. Ins
tall Levenshtein (e.g. `pip install python-Levenshtein`) to suppress this warning.
  warnings.warn(msg)
```

In [7]:
```python
# test

new_model = Word2Vec.load("word2vec.model")
print(new_model)
```

```
Word2Vec(vocab=50, vector_size=100, alpha=0.025)
```

In [8]:
```python
# turn sentences into vectors

sen_vec = []
for sen in word:
    vec = new_model.wv[sen]
    data = np.array(vec).astype(np.float)
    new = np.average(data, axis = 0)
    sen_vec.append(new)

sen_vec = np.array(sen_vec)
```

```
<ipython-input-8-22ba88d1c9f8>:8: DeprecationWarning: `np.float` is a deprecated alias
for the builtin `float`. To silence this warning, use `float` by itself. Doing this wi
ll not modify any behavior and is safe. If you specifically wanted the numpy scalar ty
pe, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/rel
ease/1.20.0-notes.html#deprecations
  data = np.array(vec).astype(np.float)
```

In [9]:
```python
# build a KDTree

tree = spatial.KDTree(sen_vec)
```

In [10]:
```python
# use the KDTree to find the closest vector and then map back to sentence
# closest is the resulting list, whose order matches word's

closest = []
closest_index = []

for j in range(len(sen_vec)):
    sen = sen_vec[j]
    result = tree.query(sen, 2)[1]
    i = result[0]
    if i == j:
        i = result[1]
    c = word[i]
    closest_index.append(i)
    closest.append(c)
```

In [11]:
```python
# write closest back to csv
output = []

for c in closest:
    temp = ""
    for words in c:
        temp += "|" + words + "|"
    output.append(temp)

with open('output_obfuscated_13epoch_model.csv','r') as csvinput:
    with open('obfuscated_13epoch_final_output.csv', 'w') as csvoutput:
        writer = csv.writer(csvoutput, lineterminator='\n')
        reader = csv.reader(csvinput)

        all = []
        row = next(reader)
        row.append("most_similar")
        row.append("most_similar_index")
        all.append(row)

        i = 0
        for row in reader:
            row.append(output[i])
            row.append(closest_index[i])
            all.append(row)
            i+=1

        writer.writerows(all)
```

In [19]:
```python
# accuracy test for nonobfuscated-data-trained code2seq

filename = 'nonobfuscated_final_output.csv'

with open(filename, 'r') as csvfile:

    csvreader = csv.reader(csvfile)
    fields = next(csvreader)
    data = list(csvreader)
    s = len(data)
    a = 0

    for row in data:
        last = int(row[-1])
        label = row[1]+row[2]
        similar = data[last]
        if label == similar[1]+similar[2]:
            a += 1

    print(a/s)
```

```
0.4524236983842011
```

In [12]:
```python
# accuracy test for obfuscated-data-trained code2seq

filename = 'obfuscated_13epoch_final_output.csv'

with open(filename, 'r') as csvfile:

    csvreader = csv.reader(csvfile)
    fields = next(csvreader)
    data = list(csvreader)
    s = len(data)
    a = 0

    for row in data:
        last = int(row[-1])
        label = row[1]+row[2]
        similar = data[last]
        if label == similar[1]+similar[2]:
            a += 1

    print(a/s)
```

```
0.3080714725816389
```