Search Engine : Will Perform Searches in this Notebook

We will use Pairwise distance between query and questions stored in our database

In [109]:
```python
import warnings
warnings.filterwarnings("ignore")
import pandas as pd
import sqlite3
import csv
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from wordcloud import WordCloud
import re
import os
from sqlalchemy import create_engine # database connection
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem.snowball import SnowballStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import metrics
from sklearn.metrics import f1_score,precision_score,recall_score
from datetime import datetime
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics import pairwise_distances
```

In [135]:
```python
con = sqlite3.connect('dataset/processed.db')
processed = pd.read_sql_query("""SELECT * FROM processed""", con)
con.close()
```

In [136]:
```python
processed = processed.drop(["index"], axis=1)
```

In [144]:
```
processed.head()
```

Out[144]:

	Title	Body	Tags
0	implementing boundary value analysis software ...	<pre><code>#include&lt;iostream&gt;\n#include&...	c++
1	dynamic datagrid binding silverlight	<p>I should do binding for datagrid dynamicall...	c#
2	dynamic datagrid binding silverlight	<p>I should do binding for datagrid dynamicall...	c#
3	java lang nosuchmethoderror javax servlet serv...	<p>i want to have a servlet to process inputs ...	java
4	specified initialization vector iv match block...	<p>I've had troubles using an CryptoStream for...	c#

In [133]:
```
vectorizer = CountVectorizer()
bow_features = vectorizer.fit_transform(processed['Title'])
bow_features.get_shape()
```

Out[133]:
```
(572406, 68851)
```

In [145]:
```
# TFIDF vectorizer
tfidf = TfidfVectorizer()
tfidf_features = tfidf.fit_transform(processed.Title)
tfidf_features.get_shape()
```

Out[145]:
```
(572406, 68851)
```

In [115]:
```
# We will check for this String : dynamic datagrid binding silverlight
```

In [116]:
```
def process_query(query):
    preprocessed_reviews = []
    sentance = re.sub("\S*\d\S*", "", query).strip()
    sentance = re.sub('[^A-Za-z]+', ' ', sentance)
    sentance = ' '.join(e.lower() for e in sentance.split() if e.lower() not in stopwords.words('english'))
    preprocessed_reviews.append(sentance.strip())
    return preprocessed_reviews
```

In [117]:
```
def tfidf_search(tfidf, query):
    query = process_query(query)
    query_trans = tfidf.transform(query)
    pairwise_dist = pairwise_distances(tfidf_features, query_trans)

    indices = np.argsort(pairwise_dist.flatten())[0:10]
    df_indices = list(processed.index[indices])
    return df_indices
```

In [118]:
```
def bow_search(vectorizer, query):
    query = process_query(query)
    query_trans = vectorizer.transform(query)
    pairwise_dist = pairwise_distances(bow_features, query_trans)

    indices = np.argsort(pairwise_dist.flatten())[0:10]
    df_indices = list(processed.index[indices])
    return df_indices
```

In [119]:
```
def search(query, typ = "tfidf"):
    if typ == "tfidf":
        val = tfidf_search(tfidf, query)
    else :
        val = bow_search(vectorizer, query)
    return val
```

In [147]:
```
query = "synchronization "
df_indices = search(query)
```

In [148]:
```python
print("The Query is :    ", query)
print("Top Results : ")
for i in (df_indices):
    print("Title : ", processed.Title.iloc[i])
```

```
The Query is :    synchronization
Top Results :
Title :  synchronization problems c using pthreads mutexes
Title :  java excel date formatting
Title :  java application servlet io exception
Title :  java tabbed pane display icon close title
Title :  android onsensorchanged wont work
Title :  parse java date
Title :  add runtime created playlist designer code runtime
Title :  c abstract classes incomplete types
Title :  c using library
Title :  c threading memory leaks
```

In [134]:
```python
df_indices = search(query, "bow")
print("The Query is :    ", query)
print("Top Results : ")
for i in (df_indices):
    print("Title : ", processed.Title.iloc[i])
```

```
The Query is :    static variable issue
Top Results :
Title :  static variable value different background agent
Title :  onclick return true false properly working
Title :  c vector based two dimensional array objects
Title :  operator definition arrays c
Title :  static object initialisation
Title :  run command emacs get output clickable buffer
Title :  statusbar frame sticks portrait orientation occludes window view
Title :  c string escape
Title :  c many ways compiler optimizes away code
Title :  c strings strlen valgrind
```

Our results are getting better but not Very Good :

So Lets use some machine learning to get better result