In [ ]:

```
#Devang Mehrotra
#18BCE0763
```

In [1]:

```
import nltk
#nltk.download('wordnet')
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import sent_tokenize, word_tokenize

import re
stop_words = set(stopwords.words('english'))
```

In [4]:

```
#common words
words1 = []
#Group in a list the words common for two text files and show their total count
f1 = open("AI.txt").readlines()
f2 = open("ML.txt").readlines()
if len(f1) != 0 | len(f2) != 0:
    uniq1 = set(words for line in f1 for words in line.strip().split())
    uniq2 = set(wordss for lines in f2 for wordss in lines.strip().split())
    for words in uniq1:
        for wordds in uniq2:
            if words == wordds:
                words1.append(words);


words1 = [w for w in words1 if not w in stop_words]
print(len(words1))
```

```
61
```

In [5]:

```
with open('index.txt', 'w') as f:
    for item in words1:
        f.write("%s\n" % item)

readwords = []

# opening the text file
with open('index.txt','r') as file:

    # reading each line
    for line in file:

        # reading each word
        for word in line.split():

            # displaying the words
            readwords.append(word)

print(readwords)
```

```
['used', 'computer', 'based', 'trying', 'way', 'development', 'While', 'Th
e', 'applications', 'made', 'They', 'language', 'use', 'new', 'machine', 'le
arning', 'creating', 'developed', 'Applications', 'various', 'think', 'man
y', 'technology', 'change', 'may', 'science', 'humans', 'developing', 'recog
nition', 'Intelligence', 'In', 'perform', 'provide', 'work', 'possible', 'as
sociated', 'intelligence', 'like', 'Speech', 'learning,', 'What', 'fields',
'world', 'These', 'people', 'making', 'Artificial', 'programming', 'comple
x', 'specific', 'AI', 'natural', 'world,', 'learn,', 'paper', 'different',
'data', 'tasks', 'interact', 'increasing', 'intelligent']
```

In [6]:

```
ps = PorterStemmer()
lemmatizer = WordNetLemmatizer()
stems = []
lemma = []
for w in readwords:
    print(ps.stem(w), " - ", lemmatizer.lemmatize(w))
    stems.append(ps.stem(w))
    lemma.append(lemmatizer.lemmatize(w))

frequency1 = {}
for word in stems:
    count = frequency1.get(word,0)
    frequency1[word] = count + 1
frequency_list1 = frequency1.keys()
print(len(frequency_list1))

frequency2 = {}
for word in lemma:
    count = frequency2.get(word,0)
    frequency2[word] = count + 1
frequency_list2 = frequency2.keys()
print(len(frequency_list2))

if(len(frequency_list1) <= len(frequency_list2)):
    with open('index.txt', 'w') as f:
        for item in stems:
            f.write("%s\n" % item)
```

```
use  -  used
comput  -  computer
base  -  based
tri  -  trying
way  -  way
develop  -  development
while  -  While
the  -  The
applic  -  application
made  -  made
they  -  They
languag  -  language
use  -  use
new  -  new
machin  -  machine
learn  -  learning
creat  -  creating
develop  -  developed
applic  -  Applications
variou  -  various
think  -  think
mani  -  many
technolog  -  technology
chang  -  change
may  -  may
scienc  -  science
human  -  human
develop  -  developing
recognit  -  recognition
intellig  -  Intelligence
In  -  In
```

```
perform  -  perform
provid  -  provide
work  -  work
possibl  -  possible
associ  -  associated
intellig  -  intelligence
like  -  like
speech  -  Speech
learning,  -  learning,
what  -  What
field  -  field
world  -  world
these  -  These
peopl  -  people
make  -  making
artifici  -  Artificial
program  -  programming
complex  -  complex
specif  -  specific
AI  -  AI
natur  -  natural
world,  -  world,
learn,  -  learn,
paper  -  paper
differ  -  different
data  -  data
task  -  task
interact  -  interact
increas  -  increasing
intellig  -  intelligent
55
61
```

In [7]:

```
import os


if(len(frequency_list1) > len(frequency_list2)):
    print("hello")
    with open('index.txt', 'w') as f:
        for item in lemma:
            f.write("%s\n" % item)
#os.rename('index.txt', 'final-index.txt')
```

In [9]:

```
import nltk
nltk.download('averaged_perceptron_tagger')
finalwords = []

# opening the text file
with open('index.txt','r') as file:

    # reading each line
    for line in file:

        # reading each word
        for word in line.split():

            # displaying the words
            finalwords.append(word)
tagged = nltk.pos_tag(finalwords)
print(tagged)
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Devang Mehrotra\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.

[('use', 'NN'), ('comput', 'NN'), ('base', 'NN'), ('tri', 'JJ'), ('way', 'N
N'), ('develop', 'VB'), ('while', 'IN'), ('the', 'DT'), ('applic', 'JJ'),
('made', 'VBD'), ('they', 'PRP'), ('languag', 'VBP'), ('use', 'VBP'), ('ne
w', 'JJ'), ('machin', 'NN'), ('learn', 'VBP'), ('creat', 'NN'), ('develop',
'VB'), ('applic', 'JJ'), ('variou', 'NN'), ('think', 'VBP'), ('mani', 'NN
S'), ('technolog', 'VBP'), ('chang', 'NN'), ('may', 'MD'), ('scienc', 'VB'),
('human', 'JJ'), ('develop', 'VB'), ('recognit', 'NN'), ('intellig', 'NN'),
('In', 'IN'), ('perform', 'NN'), ('provid', 'NN'), ('work', 'NN'), ('possib
l', 'NN'), ('associ', 'JJ'), ('intellig', 'NN'), ('like', 'IN'), ('speech',
'NN'), ('learning,', 'VBP'), ('what', 'WP'), ('field', 'NN'), ('world', 'N
N'), ('these', 'DT'), ('peopl', 'NNS'), ('make', 'VBP'), ('artifici', 'JJ'),
('program', 'NN'), ('complex', 'JJ'), ('specif', 'NN'), ('AI', 'NNP'), ('nat
ur', 'CC'), ('world,', 'JJ'), ('learn,', 'JJ'), ('paper', 'NN'), ('differ',
'NN'), ('data', 'NNS'), ('task', 'NN'), ('interact', 'NN'), ('increas', 'J
J'), ('intellig', 'NN')]
```

In [11]:

```
import pandas as pd
df = pd.DataFrame(tagged)
print(df)
```

```
           0    1
0        use   NN
1     comput   NN
2       base   NN
3        tri   JJ
4        way   NN
..       ...  ...
56      data  NNS
57      task   NN
58  interact   NN
59   increas   JJ
60  intellig   NN

[61 rows x 2 columns]
```

In [ ]:

```
```