In [1]:

```
#Projeto Clusterização para Forense Computacional
#Disciplina de Processamento de Linguagem Natural - Prof. Dra. Nádia

#Importação das API's
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction import text
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from nltk.tokenize import RegexpTokenizer
from nltk.stem.snowball import SnowballStemmer
from nltk import word_tokenize
from nltk.corpus import stopwords
import string
import os
import re
import timeit
import nltk
%matplotlib inline
start = timeit.default_timer()
```

In [2]:

```
#Carregamento dos arquivos a serem processados

path="D:/Projeto/doc2txt"
data = {}
i=0
for subdir, dirs, files in os.walk(path):
    for file in files:
        file_path = subdir + os.path.sep + file
        arquivo = open(file_path, 'r',encoding='utf8', errors='ignore')
        text = arquivo.read()
        lowers = text.lower()
        data[i] = lowers
        i=i+1

#Verificar se todos os arquivos foram carregados
print(len(data))
print(data[0])
```

512
before the
federal communications commission
washington, d.c. 20554

in the matter of

amendment of section 73.202(b)
fm table of allotments,
fm broadcast stations.
(tenino, washington)
)
)
)
)
)
)
)

mb docket no. 05-185
rm-11236

report and order
(proceeding terminated)
adopted: august 24, 2005 released: august 26, 200
5

by the assistant chief, audio division, media bureau:

1. the audio division has before it a notice of proposed rulemaking 1 issued at the request of dr. sandra l. woodruff (“petitioner”) requestin g the allotment of channel 229c3 at tenino, washington, as the community’s first local service. petitioner did not file comments reiterating her con tinuing interest in applying for channel 229c3 at tenino.2 no other comme nts or counterproposals were received.

2. as stated in the notice, a showing of continuing interest is re quired before a channel will be allotted. it is the commission’s policy t o refrain from making an allotment to a community absent an expression of interest. therefore, since no such continuing interest has been expressed for an allotment at tenino, washington, we will dismiss petitioner’s propo sal.

3. this document is not subject to the congressional review act. (the commission, is, therefore, not required to submit a copy of this repo rt and order to government accountability office, pursuant to the congress ional review act, see 5 u.s.c. section 801(a)(1)(a) since this proposed ru le is dismissed, herein.)

4. in view of the above, it is ordered that the petition for rule m aking filed by dr. sandra l. woodruff (rm-11236), requesting the allotment of channel 229c3 at tenino, washington, is dismissed.

5. it is further ordered, that this proceeding is terminated.

6.
for further information concerning this proceeding, contact helen mclean, media bureau, (202) 418-2738.

federal communications commission

john a. karousos
assistant chief
audio division
media bureau

1 see tenino, washington, 20 fcc rcd 8835 (mb 2005).

2 the proposed reference coordinates were 46-51-22 nl and 122-52-30 wl wit
h a site restriction of 1.9 kilometers (1.1 miles) west of tenino.

(...continued from previous page)

(continued....)

---

federal communications commission da 05-2340

In [3]:

```
#Usando expressão regular para remover o que não for letras e transformar tudo em texto
 minúsculo
i=0
u=len(data)
for i in range(0,len(data)):
    letters_only = re.sub("[^a-zA-Z]", # The pattern to search for
      " ", # The pattern to replace it with
      data[i] ) # The text to search
    lower_case = letters_only.lower()
    re.sub(r'\b\w{1,3}\b', '',lower_case)
    data[i]=lower_case
print(data[0])
```

```
 before the federal communications commission washington  d c          in
the matter of  amendment of section        b  fm table of allotments   fm
broadcast stations   tenino  washington                    mb docket no
     rm           report and order  proceeding terminated  adopted  august
                           released  august            by the assistant c
hief  audio division  media bureau             the audio division has befo
re it a notice of proposed rulemaking  issued at the request of dr  sandra
l  woodruff   petitioner   requesting the allotment of channel    c  at te
nino  washington  as the community s first local service   petitioner did
not file comments reiterating her continuing interest in applying for chan
nel    c  at tenino    no other comments or counterproposals were received
             as stated in the notice  a showing of continuing interest is
required before a channel will be allotted   it is the commission s policy
to refrain from making an allotment to a community absent an expression of
interest   therefore  since no such continuing interest has been expressed
for an allotment at tenino  washington  we will dismiss petitioner s propo
sal            this document is not subject to the congressional review ac
t    the commission  is  therefore  not required to submit a copy of this
report and order to government accountability office  pursuant to the cong
ressional review act  see   u s c  section     a     a  since this propose
d rule is dismissed  herein             in view of the above  it is ordere
d that the petition for rule making filed by dr  sandra l  woodruff  rm
      requesting the allotment of channel    c  at tenino  washington  is
dismissed            it is further ordered  that this proceeding is termin
ated              for further information concerning this proceeding  cont
act helen mclean  media bureau                          federal communicati
ons commission                                  john a  karousos          a
ssistant chief         audio division         media bureau   see tenino  w
ashington     fcc rcd       mb          the proposed reference coordinates
were          nl and           wl with a site restriction of     kilometer
s      miles  west of tenino                                        continu
ed from previous page    continued
                                                federal communications comm
ission     da                federal communications commission da
```

In [4]:

```
#Stopword and punctuation
stopwords = nltk.corpus.stopwords.words('english') + list(string.punctuation)
```

In [5]:

```
# load nltk's SnowballStemmer as variabled 'stemmer'
from nltk.stem.snowball import SnowballStemmer
stemmer = SnowballStemmer("english")
```

In [6]:

```
# define a tokenizer and stemmer which returns the set of stems in the text that it is
 passed

def tokenize_and_stem(text):
    # first tokenize by sentence, then by word to ensure that punctuation is caught as
 it's own token
    tokens = [word for sent in nltk.sent_tokenize(text) for word in nltk.word_tokenize(
sent)]
    filtered_tokens = []
    # filter out any tokens not containing letters (e.g., numeric tokens, raw punctuati
on)
    for token in tokens:
        if re.search('[a-zA-Z]', token):
            filtered_tokens.append(token)
    stems = [stemmer.stem(t) for t in filtered_tokens]
    return stems


def tokenize_only(text):
    # first tokenize by sentence, then by word to ensure that punctuation is caught as
 it's own token
    tokens = [word.lower() for sent in nltk.sent_tokenize(text) for word in nltk.word_t
okenize(sent)]
    filtered_tokens = []
    # filter out any tokens not containing letters (e.g., numeric tokens, raw punctuati
on)
    for token in tokens:
        if re.search('[a-zA-Z]', token):
            filtered_tokens.append(token)
    return filtered_tokens
```

In [7]:

```
#use extend so it's a big flat list of vocab
totalvocab_stemmed = []
totalvocab_tokenized = []
for i in data.values():
    allwords_stemmed = tokenize_and_stem(i)
    totalvocab_stemmed.extend(allwords_stemmed)
    allwords_tokenized = tokenize_only(i)
    totalvocab_tokenized.extend(allwords_tokenized)
print("finalizou")
```

```
finalizou
```

In [8]:

```
vocab_frame = pd.DataFrame({'words': totalvocab_tokenized}, index = totalvocab_stemmed)
print('there are ' + str(vocab_frame.shape[0]) + ' items in vocab_frame')
```

```
there are 3967445 items in vocab_frame
```

In [9]:

```
print(vocab_frame.head())
```

```
                 words
befor             before
the                 the
feder           federal
communic  communications
commiss       commission
```

In [10]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

#define vectorizer parameters
tfidf_vectorizer = TfidfVectorizer(max_df=0.8, max_features=200000,
                                   min_df=0.2, stop_words='english',
                                   use_idf=True, tokenizer=tokenize_and_stem, ngram_range
=(1,3))

%time tfidf_matrix = tfidf_vectorizer.fit_transform(data.values()) #fit the vectorizer
 to synopses

print(tfidf_matrix.shape)
```

```
c:\users\albernaz\appdata\local\programs\python\python37-32\lib\site-packa
ges\sklearn\feature_extraction\text.py:300: UserWarning: Your stop_words m
ay be inconsistent with your preprocessing. Tokenizing the stop words gene
rated tokens ['abov', 'afterward', 'alon', 'alreadi', 'alway', 'ani', 'ano
th', 'anyon', 'anyth', 'anywher', 'becam', 'becaus', 'becom', 'befor', 'be
sid', 'cri', 'describ', 'dure', 'els', 'elsewher', 'empti', 'everi', 'ever
yon', 'everyth', 'everywher', 'fifti', 'forti', 'henc', 'hereaft', 'hereb
i', 'howev', 'hundr', 'inde', 'mani', 'meanwhil', 'moreov', 'nobodi', 'noo
n', 'noth', 'nowher', 'onc', 'onli', 'otherwis', 'ourselv', 'perhap', 'ple
as', 'sever', 'sinc', 'sincer', 'sixti', 'someon', 'someth', 'sometim', 's
omewher', 'themselv', 'thenc', 'thereaft', 'therebi', 'therefor', 'toget
h', 'twelv', 'twenti', 'veri', 'whatev', 'whenc', 'whenev', 'wherea', 'whe
reaft', 'wherebi', 'wherev', 'whi', 'yourselv'] not in stop_words.
  'stop_words.' % sorted(inconsistent))
```

```
Wall time: 2min 5s
(512, 760)
```

In [11]:

```
terms = tfidf_vectorizer.get_feature_names()
```

In [12]:

```
from sklearn.metrics.pairwise import cosine_similarity
dist = 1 - cosine_similarity(tfidf_matrix)
```

In [13]:

```
from sklearn.cluster import KMeans

num_clusters = 8

km = KMeans(n_clusters=num_clusters,n_init = 20, n_jobs = 1)

%time km.fit(tfidf_matrix)

clusters = km.labels_.tolist()
```

```
Wall time: 49 s
```

In [14]:

```
from sklearn.externals import joblib

#uncomment the below to save your model
#since I've already run my model I am loading from the pickle

joblib.dump(km,  'doc_cluster.pkl')

km = joblib.load('doc_cluster.pkl')
clusters = km.labels_.tolist()
```

In [15]:

```
result = { 'termo': data.values(), 'cluster': clusters }

frame = pd.DataFrame(result, index = [clusters] , columns = ['termo', 'cluster'])
```

In [16]:

```
frame['cluster'].value_counts()
```

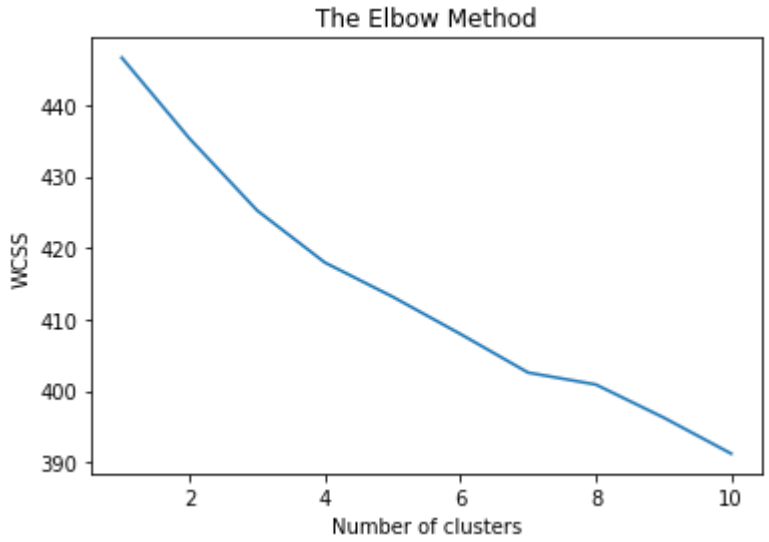
Out[16]:

```
2    158
1     91
5     85
4     44
3     43
0     43
7     39
6      9
Name: cluster, dtype: int64
```

In [17]:

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1,11):
    print("Iniciando", i)
    kmeans = KMeans(n_clusters=i,init='k-means++',max_iter=300,n_init=10,random_state=0
)
    kmeans.fit(tfidf_matrix)
    wcss.append(kmeans.inertia_)
    print("Finalizando", i)
plt.plot(range(1,11),wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.savefig('elbow.png')
plt.show()
```

```
Iniciando 1
Finalizando 1
Iniciando 2
Finalizando 2
Iniciando 3
Finalizando 3
Iniciando 4
Finalizando 4
Iniciando 5
Finalizando 5
Iniciando 6
Finalizando 6
Iniciando 7
Finalizando 7
Iniciando 8
Finalizando 8
Iniciando 9
Finalizando 9
Iniciando 10
Finalizando 10
```



In [18]:

```
from __future__ import print_function

print("Top terms per cluster:")
print()
#sort cluster centers by proximity to centroid
order_centroids = km.cluster_centers_.argsort()[:,-1:-26:-1]

for i in range(num_clusters):
    print("Cluster %d words:" % i, end='')

    for ind in order_centroids[i, :10]: #replace 6 with n words per cluster
        print(' %s' % vocab_frame.loc[terms[ind].split(' ')].values.tolist()[0][0].enco
de('utf-8', 'ignore'), end=',')
    print() #add whitespace
    print() #add whitespace

    #print("Cluster %d titles:" % i, end='')
    #for termo in frame.ix[i]['termo'].values.tolist():
    #    print(' %s,' % termo, end='')
    #print() #add whitespace
    #print() #add whitespace

print()
print()
```

```
Top terms per cluster:

Cluster 0 words: b'shall', b'section', b'any', b'required', b'permit',
b'b', b'contract', b'c', b'authors', b'department',

Cluster 1 words: b'measured', b'data', b'figure', b'testing', b'sampled',
b'energy', b'studies', b'device', b'temperatures', b'model',

Cluster 2 words: b'program', b'health', b'stated', b'service', b'include',
b'required', b'planning', b'provides', b'information', b'management',

Cluster 3 words: b'j', b'e', b'm', b'd', b'l', b'r', b'c', b'h', b't',
b'f',

Cluster 4 words: b'commission', b'cost', b'file', b'rate', b'productive',
b'order', b'consumers', b'service', b'section', b'labeled',

Cluster 5 words: b'time', b't', b'stated', b'just', b'year', b'like', b've
ry', b'meets', b'people', b'member',

Cluster 6 words: b'n', b'o', b'al', b'c', b'm', b'p', b'v', b'l', b'et',
b'legal',

Cluster 7 words: b'site', b'plant', b'water', b'area', b'national', b'par
k', b'north', b'covered', b'located', b'resource',
```

In [19]:

```
stop = timeit.default_timer()
execution_time = stop - start

print(execution_time/60) #It returns time in min
```

```
34.235403902916666
```

In [ ]:

```

```