Web Mining – Lab 6

Clustering of Web documents using KMeans

By Abhijeet Ambadekar (16BCE1156)

Problem statement:

Part A -

You are given 9 one-line documents here. Consider the following keywords to represent the documents in the vector space model:

[1] Automotive [2] Car [3] motorcycles [4] self-drive [5] IoT [6] hire [7] Dhoni
Represent the documents in vector space Model using these keywords and use it as input to cluster the
documents using Manhattan distance as parameter. Ignore case differences.
You also need to do K- means clustering with K=4.

Part B -

Use the same program which you have developed for part A to do " K-means clustering" of the following web documents with K=4. Use the keywords

[1] Tesla [2] Electric [3] Car/Vehicle/Automobile [4] pollution [5] de-monetisation [6] GST [7] black money

Download the webpage into a .txt file [ignore images,tables and limit the size of the document to 500 words Max] and build your vector space model using Term frequency.

Ignore case differences. Treat singular and plural of nouns as same. Treat Car/vehicle/automobile as one word [synonyms]. Treat "black money" as a single word.

Program Code:

```
import numpy as np
import re

import requests
from bs4 import BeautifulSoup

def get_web_text(url):
	try:
		page = requests.get(url)
	except Exception as e:
		try:
			print("Failed to reached {}".format(url))
		except UnicodeEncodeError:
			print("Failed to reached and cant show the URL")
		return None

	soup = BeautifulSoup(page.text, 'html.parser')

	# Removing style blocks
	[tag.decompose() for tag in soup("style")]

	# Removing scripts
	[tag.decompose() for tag in soup("script")]

	text = re.sub('\n+', ' ', soup.get_text()).strip().lower()
	return text


def get_word_count(words, text):
	if type(words) == str:
		return text.count(words)
	elif type(words) == list:
		return sum([text.count(wordi) for wordi in words])



class KMeans_Clusterer():
	def __init__(self, k = 4, num_iter = 20):
		self.k = k
		self.num_iter = num_iter

		self.wordlist = []
		self.doclist = {}
		self.doc_vecs = {}

		self.clusters = {}
		self.cluster_centroids = {}
```

```
      def get_text_data(self, docfile):
            with open(docfile) as fp:
                  content = list(map(str.lower, fp.readlines()))

            for textline in content:
                  docname, doctext = textline.split(" : ")
                  self.doclist[docname] = doctext


      def get_URL_data(self, docfile):
            with open(docfile) as fp:
                  urls = list(map(str.strip, fp.readlines()))

            for url in urls:
                  self.doclist[url] = get_web_text(url)


      def get_word_freq(self):
            for doc in self.doclist:
                  self.doc_vecs[doc] = np.array([get_word_count(word,
self.doclist[doc]) for word in self.wordlist])


      def get_doc_vec(self, doc):
            return self.doc_vecs[doc]


      def fit(self, wordlist, docfile, input_type = None):
            self.wordlist = wordlist
            if input_type == "TEXT":
                  self.get_text_data(docfile)
            elif input_type == "URL":
                  self.get_URL_data(docfile)
            else:
                  print("Input type unspecified while fitting the data.")

            self.get_word_freq()
            self.freq_matrix = [[0 for i in range(self.k)] for i in
range(len(self.doc_vecs.keys()))]


      def get_doc_dist(self, doc1, doc2, metric="manhattan"):
            if metric == "manhattan":
                  return round(np.sum(abs(doc1 - doc2)), 4)
            elif metric == "euclidean":
                  return round((sum((self.doc_vecs[doc1] -
self.doc_vecs[doc2])**2))**(0.5), 4)


      def update_cluster_centroids(self):
            for curr_clus in sorted(self.clusters.keys()):
```

```
                  clus_doc_vecs = [self.doc_vecs[doc] for doc in
self.clusters[curr_clus]]
                  # print(clus_doc_vecs)
                  self.cluster_centroids[curr_clus] =
np.around(np.mean(clus_doc_vecs, axis = 0), 4)


      def update_freq_matrix(self):
            for i, doci in enumerate(sorted(self.doc_vecs.keys())):
                  for j, clusj in enumerate(sorted(self.clusters.keys())):
                        self.freq_matrix[i][j] =
self.get_doc_dist(self.doc_vecs[doci], self.cluster_centroids[clusj],
metric="manhattan")


      def cluster_init(self):
            self.clusters = {"Cluster"+str(i): [sorted(self.doclist.keys())[i]]
for i in range(self.k)}

            # Adding nearest Neighbour for the remaining documents
            for i, doci in enumerate(sorted(self.doclist.keys())[self.k:]):
                  self.clusters[sorted(self.clusters.keys())[i %
self.k]].append(doci)
            self.update_cluster_centroids()
            self.update_freq_matrix()


      def cluster(self):
            self.cluster_init()

            for i in range(self.num_iter):

                  temp_clus = {clus : [] for clus in
sorted(self.clusters.keys())}

                  for i, doci in enumerate(sorted(self.doc_vecs.keys())):
                        j = self.freq_matrix[i].index(min(self.freq_matrix[i]))
                        temp_clus[sorted(temp_clus.keys())[j]].append(doci)

                  if temp_clus == self.clusters:
                        print("Clusters converged at {} iterations of
{}.".format(i, self.num_iter))
                        break
                  else:
                        self.clusters = temp_clus

                  self.update_cluster_centroids()
                  self.update_freq_matrix()


      def print_details(self):
```

```
            print("The clusters created are as follows:")
            for clus in self.clusters:
                  print(clus, ":", self.clusters[clus])

            print("\n\nThe document vectors of these clusters are:")
            for clus in self.cluster_centroids:
                  print(clus, ":", self.cluster_centroids[clus])



print("(a) Clustering given one line documents")

wordlist = ["automotive", "car", "motorcycle", "self-drive", "iot", "hire",
"dhoni"]
docfile = "docfile.txt"

textcls = KMeans_Clusterer(k=4, num_iter=20)
textcls.fit(wordlist, docfile, input_type="TEXT")
textcls.cluster()
textcls.print_details()


print("\n\n(b) Clustering given URLs")

urlwordlist = ["tesla", "electric", ["car", "vehicle", "automobile"],
"pollution", "de-monetisation" , "gst" , "black money"]
urldocfile = "urldocfile.txt"

urlcls = KMeans_Clusterer()
urlcls.fit(urlwordlist, urldocfile, input_type="URL")
urlcls.cluster()
urlcls.print_details()
```

Source Documents:

L7_KMeansClustering.py docfile.txt

```
Doc 1 : Electric automotive maker Tesla Inc. is likely to introduce its products in India sometime in the summer of 2017.
Doc 2 : Automotive major Mahindra likely to introduce driverless cars
Doc 3 : BMW plans to introduce its own motorcycles in india
Doc 4 : Just drive, a self-drive car rental firm uses smart vehicle technology based on IoT
Doc 5 : Automotive industry going to hire thousands in 2018
Doc 6 : Famous cricket player  Dhoni brought his priced car Hummer which is an SUV
Doc 7 : Dhoni led india to its second world cup victory
Doc 8 : IoT in cars will lead to more safety and make driverless vehicle revolution possible
Doc 9 : Sachin recommended Dhoni for the indian skipper post
```

L7_KMeansClustering.py urldocfile.txt

```
https://www.zigwheels.com/newcars/Tesla
https://www.financialexpress.com/auto/car-news/
mahindra-to-launch-indias-first-electric-suv-in-2019-all-new-e-verito-sedan-on-cards/1266853/
https://en.wikipedia.org/wiki/Toyota_Prius
https://economictimes.indiatimes.com/industry/auto/auto-news/government-plans-new-policy-to-promote-electric-vehicles/
articleshow/65237123.cms
https://indianexpress.com/article/india/india-news-india/
demonetisation-hits-electric-vehicles-industry-society-of-manufacturers-of-electric-vehicles-4395104/
https://www.livemint.com/Politics/ySbMKTIC4MINsz1btccBJO/How-demonetisation-affected-the-Indian-economy-in-10-charts.html
https://www.hrblock.in/blog/impact-gst-automobile-industry-2/
https://inc42.com/buzz/
electric-vehicles-this-week-centre-reduces-gst-on-lithium-ion-batteries-hyundai-to-launch-electric-suv-in-india-and-more/
https://www.youthkiawaaz.com/2017/12/impact-of-demonetisation-on-the-indian-economy/
https://indianexpress.com/article/india/demonetisation-effects-cash-crisis-mobile-wallets-internet-banking-4406005/
https://www.news18.com/news/business/how-gst-will-curb-tax-evasion-1446035.html
https://economictimes.indiatimes.com/small-biz/policy-trends/is-gst-helping-the-indian-economy-for-the-better/
articleshow/65319874.cms
```

Output:

```
/media/anonymous/Work/Vit/Semester 5/WM/Lab/L7_KMeansClustering  python3 L7_KMeansClustering.py
(a) Clustering given one line documents
Clusters converged at 8 iterations of 20.
The clusters created are as follows:
Cluster0 : ['doc 1', 'doc 5']
Cluster1 : ['doc 2', 'doc 6']
Cluster2 : ['doc 3', 'doc 7', 'doc 9']
Cluster3 : ['doc 4', 'doc 8']


The document vectors of these clusters are:
Cluster0 : [1.  0.  0.  0.  0.  0.5 0. ]
Cluster1 : [0.5 1.  0.  0.  0.  0.  0.5]
Cluster2 : [0.     0.     0.3333 0.     0.     0.     0.6667]
Cluster3 : [0.  1.  0.  0.5 1.  0.  0. ]


(b) Clustering given URLs
Clusters converged at 11 iterations of 20.
The clusters created are as follows:
```

```
(b) Clustering given URLs
Clusters converged at 11 iterations of 20.
The clusters created are as follows:
Cluster0 : ['https://economictimes.indiatimes.com/industry/auto/auto-news/government-plans-new-policy-to-promote-electric-vehicles/articleshow/65237123.cms', 'https://www.financialexpress.com/auto/car-news/mahindra-to-launch-indias-first-electric-suv-in-2019-all-new-e-verito-sedan-on-cards/1266853/', 'https://www.zigwheels.com/newcars/Tesla']
Cluster1 : ['https://economictimes.indiatimes.com/small-biz/policy-trends/is-gst-helping-the-indian-economy-for-the-better/articleshow/65319874.cms', 'https://inc42.com/buzz/electric-vehicles-this-week-centre-reduces-gst-on-lithium-ion-batteries-hyundai-to-launch-electric-suv-in-india-and-more/', 'https://indianexpress.com/article/india/demonetisation-effects-cash-crisis-mobile-wallets-internet-banking-4406005/', 'https://indianexpress.com/article/india/india-news-india/demonetisation-hits-electric-vehicles-industry-society-of-manufacturers-of-electric-vehicles-4395104/', 'https://www.livemint.com/Politics/ySbMKTIC4MINsz1btccBJO/How-demonetisation-affected-the-Indian-economy-in-10-charts.html', 'https://www.news18.com/news/business/how-gst-will-curb-tax-evasion-1446035.html', 'https://www.youthkiawaaz.com/2017/12/impact-of-demonetisation-on-the-indian-economy/']
Cluster2 : ['https://en.wikipedia.org/wiki/Toyota_Prius']
Cluster3 : ['https://www.hrblock.in/blog/impact-gst-automobile-industry-2/']


The document vectors of these clusters are:
Cluster0 : [ 8.3333 20.3333 44.      0.      0.      0.3333  0.    ]
Cluster1 : [0.     2.1429 6.4286 0.     0.     8.     0.5714]
Cluster2 : [  3.  50. 298.   3.   0.   0.   0.]
Cluster3 : [ 0.  0. 28.  0.  0. 61.  0.]
 /media/anonymous/Work/Vit/Semester 5/WM/Lab/L7_KMeansClustering
```