**TFIDF and LSI**

***#Gensim accomplishes this by taking a corpus, a collection of text documents, and producing a vector representation of***

***#the text in the corpus. The vector representation can then be used to train a model, which is an algorithm to create***

***#different representations of the data, which are usually more semantic. These three concepts are key to understanding***

***#how Gensim works. At the same time, we'll work through a simple example that illustrates each of them.***

***#A corpus is a collection of digital documents. This corpus is fed to Gensim from which it will infer the structure of the***

***#documents and extract topics from the documents. Once the algorithm learns on how to infer topics from the training corpus,***

***#it can be used to assign topics to new documents which were not present in the training corpus.For this reason, we also refer***

***#to this collection as the training corpus. No human intervention is required - the topic classification is unsupervised.***

**​*# Reading whole corpus from the diabetes and make a big corpus***

***## Using glob we can read and write***

***import io***

***import glob***

***​***

***a = open("output.txt",'a',encoding="utf-8")***

***counter = 0***

***#var = ""***

***list\_of\_files = glob.glob(r"C:\Users\lohit\Desktop\course\KDD\FinalProject\diabetes\diabetes\\*.txt")        # create the list of file***

***for file\_name in list\_of\_files:***

***FI = open(file\_name, 'r',encoding="utf-8")***

***#print(FI.read())***

***for line in FI:***

***a.write(line)***

***counter = counter + 1***

***FI.close()***

***a.close()***

***​***

***# We first remove all the words the commonly used English words - called stop words such as 'the', ‘a’, ‘we’, etc.) and words***

***# that occur only once in the corpus.***

***# Second we are counting number of different words inside the document and if the count is more than one it will keep on adding***

***# frequency and calcuate the total frequency of all words.***

***# Now we are take the words which are more than 0 time inside the corpus.***

***# After that we are printing the words.***

***# Using NLTK library we are calculating the operation***

***#Using NLTK to appedning the whole in single file***

***# do not tun this part take a lot of time.***

**from nltk.corpus import stopwords**

**from nltk.tokenize import word\_tokenize**

***#word\_tokenize accepts a string as an input, not a file.***

**stop\_words = set(stopwords.words('english'))**

**file1 = open(r"C:\Users\lohit\Desktop\course\KDD\output.txt",'r',encoding='utf-8')**

**line = file1.read()*# Use this to read file content as a stream:***

**words = line.split()**

**for r in words:**

**if not r in stop\_words:**

**appendFile = open('filteredtext.txt','a',encoding='utf-8')**

**appendFile.write(" "+r)**

***# Remove stop words from the corpus using NLTK***

***# Remove puncuations from the corpus using NLTK***

***# Remove lemma words from the Corpus usinf NLTK***

**from nltk.corpus import stopwords**

**from nltk.stem.wordnet import WordNetLemmatizer**

**import string**

**stop = set(stopwords.words('english'))**

**exclude = set(string.punctuation)**

**lemma = WordNetLemmatizer()**

**def clean(doc):**

***#print(doc)***

**stop\_free = " ".join([i for i in doc.lower().split() if i not in stop])**

**punc\_free = ''.join(ch for ch in stop\_free if ch not in exclude)**

**normalized = " ".join(lemma.lemmatize(word) for word in punc\_free.split())**

**return normalized**

**​**

**document\_text = open(r"C:\Users\lohit\Desktop\course\KDD\output.txt",'r', encoding='utf-8')**

**x = document\_text.read()**

**doc\_clean = [clean(doc).split() for doc in x.split("\n")]**

***# Importing Gensim***

***import gensim***

***from gensim import corpora***

***​***

***# Creating the term dictionary of our courpus, where every unique term is assigned an index.***

***dictionary = corpora.Dictionary(doc\_clean)***

***​***

***# Converting list of documents (corpus) into Document Term Matrix using dictionary prepared above.***

***doc\_term\_matrix = [dictionary.doc2bow(doc) for doc in doc\_clean]***

***#Calculating the word frequency from the corpus and removing some of them***

***import re***

***import string***

***dictionary\_arr=[]***

***document\_text = open(r"C:\Users\lohit\Desktop\course\KDD\output.txt",'r',encoding = 'utf-8')***

***x = document\_text.read()***

***​***

***#text\_string = x.split()***

***# Count word frequencies***

***from collections import defaultdict***

***frequency = defaultdict(int)***

***for text in x.split():***

***for token in text.split():***

***frequency[token] += 1***

***​***

***#for words in x.lower().split('\n'):***

***#        dictionary\_arr.append(words)***

***processed\_corpus = [[token for token in text.split() if frequency[token] > 0] for text in x.split('\n')]***

***#processed\_corpus***

***#processed\_corpus***

***​***

***#doc\_clean = [clean(doc).split() for doc in dictionary\_arr]***

***#We now need to tokenize our data. This breaks the documents into words and assigns tokens: unique numbers to the words that***

***#have been repeated more than “x” times. Thus, we associate each word in the corpus with a unique integer ID. We can do this***

***#using the Gensim.corpora.Dictionary class. This dictionary defines the vocabulary of all words that our processing knows about.***

***from gensim import corpora***

***dictionary = corpora.Dictionary(processed\_corpus)***

***dictionary.save('C:/Users/lohit/AppData/Local/Temp/den.dict')***

***#print(dictionary)***

***C:\Users\lohit\Anaconda3\lib\site-packages\gensim\utils.py:862: UserWarning: detected Windows; aliasing chunkize to chunkize\_serial***

***warnings.warn("detected Windows; aliasing chunkize to chunkize\_serial")***

***print(dictionary.token2id)***

***IOPub data rate exceeded.***

***The notebook server will temporarily stop sending output***

***to the client in order to avoid crashing it.***

***To change this limit, set the config variable***

***`--NotebookApp.iopub\_data\_rate\_limit`.***

***bow\_corpus = [dictionary.doc2bow(text) for text in processed\_corpus]***

***corpora.MmCorpus.serialize('C:/Users/lohit/AppData/Local/Temp/den.mm', bow\_corpus)***

***#bow\_corpus***

***#Next, we need to represent the documents mathematically to be able to continue further processing, so we represent each***

***#document as a vector. We use the bag-of-words model where each document is represented by a vector containing the frequency***

***#counts of each word in the dictionary. The length of the vector is the number of entries in the dictionary. One of the main***

***#properties of the bag-of-words model is that it completely ignores the order of the tokens in the document that is encoded,***

***#hence bag-of-words.***

***#The first entry in each tuple corresponds to the ID of the token in the dictionary, the second corresponds to the count***

***#of this token. Now changing the whole courpus into vector form.***

***#We are saving this into our temporary folder called with an extension of “.mm”. Note that while this list lives entirely***

***#in memory, in most applications you will want a more scalable solution.***

***#Now that we have vectorized our corpus we can begin to transform it using models. We use model as an abstract term***

***#referring to a transformation from one document representation to another. In Gensim documents are represented as vectors***

***#so a model can be thought of as a transformation between two vector spaces. The details of this transformation are learned***

***#from the training corpus. One simple example of a model is tf-idf. The tf-idf model transforms vectors from the bag-of-words***

***#representation to a vector space where the frequency counts are weighted according to the relative rarity of each word in***

***#the corpus.***

***#Term Frequency, Inverse Document Frequency (TF-IDF) TF-IDF is a way to score the importance of words (or "terms") in a***

***#document based on how frequently they appear across multiple documents.***

***#If a word appears frequently in a document, it's important and TF-IDF gives the word a high score.***

***#But if a word appears in many documents, then it's not a unique identifier and gives the word a low score.***

***#Therefore, stop words will be scaled down. Words that appear frequently in a single document will be scaled up.***

***#For a term t in a document d, the weight Wt, d of term t in document d is given by:***

***#Wt,d = TFt,d log (N/DFt )***

***#Where,***

***#TFt,d is the number of occurrences of t in document d.***

***#DFt is the number of documents containing term t.***

***#N is the total number of documents in corpus.***

***from gensim import models***

***# train the model***

***tfidf = models.TfidfModel(bow\_corpus)***

***# transform the "system minors" string***

***#top 10 frequrn***

***print(tfidf[dictionary.doc2bow("diabetes patients type study insulin blood risk disease health research".lower().split())])***

***​***

***​***

***print(tfidf[dictionary.doc2bow("come opportunity supporting publication involving millions rapidly suggesting ".lower().split())])***

***[(53, 0.3492563305587527), (60, 0.22507556354736905), (62, 0.3512624990218378), (212, 0.30059628126992954), (394, 0.31066264090650114), (405, 0.3248702030107599), (431, 0.316902484435025), (557, 0.3622871041934056), (636, 0.3007640397348136), (712, 0.29904426990547595)]***

***[(204, 0.34994353712370985), (882, 0.34918760631019247), (3230, 0.3528768841850399), (3346, 0.34994353712370985), (4171, 0.35328037345900437), (4217, 0.3648741288092756), (4905, 0.358821439314328), (6213, 0.34918760631019247)]***

***putx = tfidf[dictionary.doc2bow("diabetes patients type study insulin blood risk disease health research people cells glucose said also levels researchers will treatment university care clinical medical title heart control high percent sugar found diabetic results studies body medicine obesity weight cell years group drug american first association data development information women professor center diseases increased time cardiovascular mice used associated children function help important including national year factors cancer million findings diet published metabolic well says system many journal compared effects lead based patient team human therapy school among trial higher developing institute http long complications according three googletag kidney beta cause drugs protein increase known company lower adults related need using however work participants life food significant world loss less program source potential resistance number management healthy develop hospital liver population better scientists pressure author exercise chronic improve metformin without activity phase likely authors changes even early term like reduce available called prevention trials individuals obese condition test states leading common role showed gene make prevent effect pancreas death part support provide effective cholesterol immune united normal conditions evidence reported groups metabolism include surgery whether developed treated public major different looking future genetic forward healthcare diagnosed statements level much daily quality events director report problems current safety shown lifestyle addition products americans within stroke take failure colleagues genes article significantly issue analysis scientific physical greater reduced placebo international brain currently treatments tissue technology primary taking might result additional treat often education bottom syndrome display dose oral small energy similar average response push months department able physicians form hormone combination risks today week possible impact 600xflex four linked online ability specific outcomes benefits rates foundation hba1c process large suggest release product proteins diabetics received medications around several target change show identified good reduction growth inflammation five provides improved society science just rate added factor college damage serious worldwide symptoms benefit countries causes controlled global family cases therapeutic production overweight produce needed involved made older since recent although times conducted period previous general another presented humans pancreatic community know every state european increasing weeks visit attack find therapies cost molecular programs producing monitoring pain living already severe services shows medication person least given either approximately middle suggests nutrition annual following glycemic total overall approach prevalence follow understanding screening sciences action incidence access muscle diagnosis become model hypoglycemia nearly included president half identify efficacy discovery 2007 designed 2008 senior across side costs disorders discovered novel self intake hypertension young subjects still stress working adverse increases estimated best anti treating delivery fatty receptor demonstrated onset previously active determine biology eating next approved finding adult mortality centers certain inflammatory manage vessels institutes mellitus amount needs link recently multiple standard range foods gain paper understand affect pharmaceutical coronary particularly baseline laboratory review reducing free mass project meeting announced funded together longer alone individual testing market receive fasting vision intervention caused 2010 survey class news guidelines administration practice index critical california systems tests types randomized affects measure white single autoimmune second billion molecules making differences lowering pharmaceuticals dietary month associate growing affected mechanism step especially chief days regulatory problem observed history present sensitivity enough nerve 2009 press mean mechanisms example studied blindness investigator doctors potentially lives interventions vascular cure experience elevated contact animals taken skin stage 2006 believe animal continue positive meal harvard produced earlier last mouse fact goal medicines difference progression seen injections order doctor recommended ways later point hope provided division hours aged expected funding appears advert poor case endocrinology largest third members therefore must improving biological expression improvement enzyme look complex includes pathway injection focused established decrease short limited play professionals models decreased considered receiving required reports physician plan focus achieve service rise areas cellular epidemic investigators full importance organization europe training tested preventing signaling area throughout others internal populations lack doses occurs regular contribute helps supported measured strategies targets june lipid molecule advanced almost clinic moderate chemical providing safe measures assistant leads start secretion tissues 2005 nature followed independent address examined efforts approval home place canada site collaboration light live excess whose past awareness boston despite thus particular natural issues means innovative open directly clear knowledge centre written keep deaths various double achieved companies grant recommendations unique whole improvements country experts offers genetics existing makes impaired offer york versus initial samples tolerance beneficial little post commonly explained researcher wide specifically appropriate works method field rather nation usually direct please latest member induced relationship attacks david regarding central strategy experienced highest design smoking appear main date occur responsible 2011 explains effectiveness simple america plans strong performance promote highly real twice call successful along environmental looked network difficult pathways implications require policy status regulate back leader effectively turn complete government 2004 profile resources maintain dependent thought environment controlling prior indicate assess right suggested approaches size line rapid executive stop protect chair managing combined advance took great controls progress allow regulation helping evaluated processes methods industry analyzed seven vice actual give clinicians value done noted promising burden normally investigation influence underlying fewer agents challenge reduces michael signs raise ongoing resulting eight officer completed evaluate explain notes activities essential performed success comprehensive improves generally larger adds monitor necessary want properly causing march suffer estimates create forms learn option instead course revealed variety indicated though contains getting actually defined april demonstrate reach toward john ensure continued differ principal note founded think amounts options presence respond examine concluded released majority fully allows reviewed assessed investigate plays basis continues avoid dedicated come opportunity supporting publication involving millions rapidly suggesting widely remains".lower().split())]***

***import logging***

***​***

***logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)***

***import tempfile***

***import os.path***

***​***

***TEMP\_FOLDER = tempfile.gettempdir()***

***print('Folder "{}" will be used to save temporary dictionary and corpus.'.format(TEMP\_FOLDER))***

***Folder "C:\Users\lohit\AppData\Local\Temp" will be used to save temporary dictionary and corpus.***

***from gensim import corpora, models, similarities***

***if os.path.isfile(os.path.join(TEMP\_FOLDER, 'den.dict')):***

***dictionary = corpora.Dictionary.load(os.path.join(TEMP\_FOLDER, 'den.dict'))***

***corpus = corpora.MmCorpus(os.path.join(TEMP\_FOLDER, 'den.mm'))***

***print("Used files generated before ")***

***else:***

***print("Run again error")***

***2017-12-07 02:38:14,042 : INFO : loading Dictionary object from C:\Users\lohit\AppData\Local\Temp\den.dict***

***2017-12-07 02:38:14,154 : INFO : loaded C:\Users\lohit\AppData\Local\Temp\den.dict***

***2017-12-07 02:38:14,220 : INFO : loaded corpus index from C:\Users\lohit\AppData\Local\Temp\den.mm.index***

***2017-12-07 02:38:14,221 : INFO : initializing corpus reader from C:\Users\lohit\AppData\Local\Temp\den.mm***

***2017-12-07 02:38:14,222 : INFO : accepted corpus with 379126 documents, 205840 features, 5375083 non-zero entries***

***Used files generated before***

***tfidf = models.TfidfModel(corpus) # step 1 -- initialize a model***

***2017-12-07 02:38:20,307 : INFO : collecting document frequencies***

***2017-12-07 02:38:20,310 : INFO : PROGRESS: processing document #0***

***2017-12-07 02:38:20,631 : INFO : PROGRESS: processing document #10000***

***2017-12-07 02:38:20,889 : INFO : PROGRESS: processing document #20000***

***2017-12-07 02:38:21,189 : INFO : PROGRESS: processing document #30000***

***2017-12-07 02:38:21,512 : INFO : PROGRESS: processing document #40000***

***2017-12-07 02:38:21,859 : INFO : PROGRESS: processing document #50000***

***2017-12-07 02:38:22,199 : INFO : PROGRESS: processing document #60000***

***2017-12-07 02:38:22,518 : INFO : PROGRESS: processing document #70000***

***2017-12-07 02:38:22,828 : INFO : PROGRESS: processing document #80000***

***2017-12-07 02:38:23,149 : INFO : PROGRESS: processing document #90000***

***2017-12-07 02:38:23,448 : INFO : PROGRESS: processing document #100000***

***2017-12-07 02:38:23,768 : INFO : PROGRESS: processing document #110000***

***2017-12-07 02:38:24,067 : INFO : PROGRESS: processing document #120000***

***2017-12-07 02:38:32,175 : INFO : calculating IDF weights for 379126 documents and 205839 features (5375083 matrix non-zeros)***

***corpus\_tfidf = tfidf[corpus]***

***lsi = models.LsiModel(corpus\_tfidf, id2word=dictionary, num\_topics=2) # initialize an LSI transformation***

***corpus\_lsi = lsi[corpus\_tfidf] # create a double wrapper over the original corpus: bow->tfidf->fold-in-lsi***

***2017-12-07 02:38:59,332 : INFO : using serial LSI version on this node***

***2017-12-07 02:38:59,333 : INFO : updating model with new documents***

***2017-12-07 02:39:00,134 : INFO : preparing a new chunk of documents***

***2017-12-07 02:39:00,226 : INFO : using 100 extra samples and 2 power iterations***

***2017-12-07 02:39:00,228 : INFO : 1st phase: constructing (205840, 102) action matrix***

***2017-12-07 02:39:00,345 : INFO : orthonormalizing (205840, 102) action matrix***

***2017-12-07 02:39:04,021 : INFO : 2nd phase: running dense svd on (102, 20000) matrix***

***2017-12-07 02:39:04,484 : INFO : computing the final decomposition***

***2017-12-07 02:39:04,486 : INFO : keeping 2 factors (discarding 85.645% of energy spectrum)***

***2017-12-07 02:39:04,563 : INFO : processed documents up to #20000***

***2017-12-07 02:39:04,570 : INFO : topic #0(13.687): 0.390\*"the" + 0.318\*"of" + 0.297\*"and" + 0.243\*"to" + 0.215\*"in" + 0.184\*"diabetes" + 0.179\*"a" + 0.162\*"is" + 0.156\*"with" + 0.145\*"for"***

***2017-12-07 02:39:04,576 : INFO : topic #1(12.329): 0.641\*"googletag.cmd.push(function()" + 0.641\*"{" + -0.421\*"});" + 0.000\*"as" + 0.000\*"University" + 0.000\*"from" + 0.000\*"an" + 0.000\*"aspirin" + -0.000\*"," + -0.000\*"Institute"***

***2017-12-07 02:39:05,684 : INFO : preparing a new chunk of documents***

***2017-12-07 02:39:05,799 : INFO : using 100 extra samples and 2 power iterations***

***lsi.print\_topics(2)***

***2017-12-07 02:41:06,224 : INFO : topic #0(58.360): 0.394\*"the" + 0.321\*"of" + 0.265\*"and" + 0.246\*"to" + 0.235\*"in" + 0.196\*"a" + 0.161\*"that" + 0.155\*"with" + 0.151\*"is" + 0.150\*"diabetes"***

***2017-12-07 02:41:06,231 : INFO : topic #1(37.328): 0.706\*"{" + 0.706\*"googletag.cmd.push(function()" + -0.067\*"});" + -0.000\*"Source:" + 0.000\*"concluded:" + -0.000\*"lower" + -0.000\*"our" + -0.000\*"therapy" + 0.000\*"women" + 0.000\*"patients"***

***Out[16]:***

***[(0,***

***'0.394\*"the" + 0.321\*"of" + 0.265\*"and" + 0.246\*"to" + 0.235\*"in" + 0.196\*"a" + 0.161\*"that" + 0.155\*"with" + 0.151\*"is" + 0.150\*"diabetes"'),***

***(1,***

***'0.706\*"{" + 0.706\*"googletag.cmd.push(function()" + -0.067\*"});" + -0.000\*"Source:" + 0.000\*"concluded:" + -0.000\*"lower" + -0.000\*"our" + -0.000\*"therapy" + 0.000\*"women" + 0.000\*"patients"')]***

***#for doc in corpus\_lsi: # both bow->tfidf and tfidf->lsi transformations are actually executed here, on the fly***

***#print(doc)***

***[(0, -1.0)]***

***[(1, 1.0)]***

***lsi.save(os.path.join(TEMP\_FOLDER, 'model.lsi')) # same for tfidf, lda, ...***

***#lsi = models.LsiModel.load(os.path.join(TEMP\_FOLDER, 'model.lsi'))***

***2017-12-07 02:41:27,994 : INFO : saving Projection object under C:\Users\lohit\AppData\Local\Temp\model.lsi.projection, separately None***

***2017-12-07 02:41:28,052 : INFO : saved C:\Users\lohit\AppData\Local\Temp\model.lsi.projection***

***2017-12-07 02:41:28,054 : INFO : saving LsiModel object under C:\Users\lohit\AppData\Local\Temp\model.lsi, separately None***

***2017-12-07 02:41:28,056 : INFO : not storing attribute projection***

***2017-12-07 02:41:28,057 : INFO : not storing attribute dispatcher***

***2017-12-07 02:41:28,213 : INFO : saved C:\Users\lohit\AppData\Local\Temp\model.lsi***

***from gensim import corpora, models, similarities***

***dictionary = corpora.Dictionary.load('C:/Users/lohit/AppData/Local/Temp/den.dict')***

***corpus = corpora.MmCorpus('C:/Users/lohit/AppData/Local/Temp/den.mm') # comes from the first tutorial, "From strings to vectors"***

***#print(corpus)***

***2017-12-07 02:41:41,397 : INFO : loading Dictionary object from C:/Users/lohit/AppData/Local/Temp/den.dict***

***2017-12-07 02:41:41,521 : INFO : loaded C:/Users/lohit/AppData/Local/Temp/den.dict***

***2017-12-07 02:41:41,567 : INFO : loaded corpus index from C:/Users/lohit/AppData/Local/Temp/den.mm.index***

***2017-12-07 02:41:41,568 : INFO : initializing corpus reader from C:/Users/lohit/AppData/Local/Temp/den.mm***

***2017-12-07 02:41:41,569 : INFO : accepted corpus with 379126 documents, 205840 features, 5375083 non-zero entries***

***doc = "diabetes research medicine"***

***vec\_bow = dictionary.doc2bow(doc.lower().split())***

***vec\_lsi = lsi[vec\_bow] # convert the query to LSI space***

***print(vec\_lsi)***

***​***

***[(0, 0.20682855845805731), (1, -1.7022889433835045e-05)]***

***index = similarities.MatrixSimilarity(lsi[corpus]) # transform corpus to LSI space and index it***

***2017-12-07 02:42:08,296 : WARNING : scanning corpus to determine the number of features (consider setting `num\_features` explicitly)***

***2017-12-07 02:42:23,292 : INFO : creating matrix with 379126 documents and 2 features***

***index.save('C:/Users/lohit/AppData/Local/Temp/den.index')***

***index = similarities.MatrixSimilarity.load('C:/Users/lohit/AppData/Local/Temp/den.index')***

***2017-12-07 02:43:04,791 : INFO : saving MatrixSimilarity object under C:/Users/lohit/AppData/Local/Temp/den.index, separately None***

***2017-12-07 02:43:04,839 : INFO : saved C:/Users/lohit/AppData/Local/Temp/den.index***

***2017-12-07 02:43:04,841 : INFO : loading MatrixSimilarity object from C:/Users/lohit/AppData/Local/Temp/den.index***

***2017-12-07 02:43:04,862 : INFO : loaded C:/Users/lohit/AppData/Local/Temp/den.index***

***sims = index[vec\_lsi] # perform a similarity query against the corpus***

***print(list(enumerate(sims))) # print (document\_number, document\_similarity) 2-tuples***

***IOPub data rate exceeded.***

***The notebook server will temporarily stop sending output***

***to the client in order to avoid crashing it.***

***To change this limit, set the config variable***

***`--NotebookApp.iopub\_data\_rate\_limit`.***

**sims = sorted(enumerate(sims), key=lambda item: -item[1])**

**print(sims) *# print sorted (document number, similarity score) 2-tuples***

**IOPub data rate exceeded.**

**The notebook server will temporarily stop sending output**

**to the client in order to avoid crashing it.**

**To change this limit, set the config variable**

**`--NotebookApp.iopub\_data\_rate\_limit`.**

***# this time we are reading file one by one and apply tfIdf-> LSI***

***# calculating words from document to document to get the interesting topic***

***import glob***

***list\_of\_files = glob.glob(r"C:\Users\lohit\Desktop\course\KDD\FinalProject\diabetes\diabetes\\*.txt")        # create the list of file***

***for file\_name in list\_of\_files:***

***FI = open(file\_name, 'r',encoding="utf-8")***

***x = FI.read()***

***​***

***#text\_string = x.split()***

***# Count word frequencies***

***from nltk.corpus import stopwords***

***from nltk.stem.wordnet import WordNetLemmatizer***

***import string***

***stop = set(stopwords.words('english'))***

***exclude = set(string.punctuation)***

***lemma = WordNetLemmatizer()***

***def clean(doc):***

***#print(doc)***

***stop\_free = " ".join([i for i in doc.lower().split() if i not in stop])***

***punc\_free = ''.join(ch for ch in stop\_free if ch not in exclude)***

***normalized = " ".join(lemma.lemmatize(word) for word in punc\_free.split())***

***return normalized***

***processed\_corpus = [clean(doc).split() for doc in x.split("\n")]***

***#print(doc\_clean)***

***from gensim import corpora, models, utils***

***dictionary = corpora.Dictionary(processed\_corpus)***

***dictionary.save('C:/Users/lohit/AppData/Local/Temp/den.dict')***

***# print(dictionary)***

***print(dictionary.token2id)***

***bow\_corpus = [dictionary.doc2bow(text) for text in processed\_corpus]***

***corpora.MmCorpus.serialize('C:/Users/lohit/AppData/Local/Temp/den.mm', bow\_corpus)***

***#print(bow\_corpus)***

***from gensim import models***

***# train the model***

***tfidf = models.TfidfModel(bow\_corpus)***

***# transform the "system minors" string***

***tfidf[dictionary.doc2bow("diabetes patients type study insulin blood risk disease health research ".lower().split())]***

***import logging***

***​***

***logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)***

***import tempfile***

***import os.path***

***​***

***TEMP\_FOLDER = tempfile.gettempdir()***

***print('Folder "{}" will be used to save temporary dictionary and corpus.'.format(TEMP\_FOLDER))***

***from gensim import corpora, models, similarities***

***if os.path.isfile(os.path.join(TEMP\_FOLDER, 'den.dict')):***

***dictionary = corpora.Dictionary.load(os.path.join(TEMP\_FOLDER, 'den.dict'))***

***corpus = corpora.MmCorpus(os.path.join(TEMP\_FOLDER, 'den.mm'))***

***print("Used files generated before ")***

***else:***

***print("Run again error")***

***tfidf = models.TfidfModel(corpus) # step 1 -- initialize a model***

***corpus\_tfidf = tfidf[corpus]***

***#for doc in corpus\_tfidf:***

***#   print(doc)***

***lsi = models.LsiModel(corpus\_tfidf, id2word=dictionary, num\_topics=50) # initialize an LSI transformation***

***corpus\_lsi = lsi[corpus\_tfidf] # create a double wrapper over the original corpus: bow->tfidf->fold-in-lsi***

***lsi.print\_topics(50)***

***#for doc in corpus\_lsi: # both bow->tfidf and tfidf->lsi transformations are actually executed here, on the fly***

***#   print(doc)***

***lsi.save(os.path.join(TEMP\_FOLDER, 'model.lsi')) # same for tfidf, lda, ...***

***#lsi = models.LsiModel.load(os.path.join(TEMP\_FOLDER, 'model.lsi'))***

***from gensim import corpora, models, similarities***

***dictionary = corpora.Dictionary.load('C:/Users/lohit/AppData/Local/Temp/den.dict')***

***corpus = corpora.MmCorpus('C:/Users/lohit/AppData/Local/Temp/den.mm') # comes from the first tutorial, "From strings to vectors"***

***#print(corpus)***

***doc = "diabetes research medicine discovert prevention cure "***

***vec\_bow = dictionary.doc2bow(doc.lower().split())***

***vec\_lsi = lsi[vec\_bow] # convert the query to LSI space***

***print(vec\_lsi)***

***index = similarities.MatrixSimilarity(lsi[corpus]) # transform corpus to LSI space and index it***

***index.save('C:/Users/lohit/AppData/Local/Temp/den.index')***

***index = similarities.MatrixSimilarity.load('C:/Users/lohit/AppData/Local/Temp/den.index')***

***sims = index[vec\_lsi] # perform a similarity query against the corpus***

***print(list(enumerate(sims))) # print (document\_number, document\_similarity) 2-tuples***

***sims = sorted(enumerate(sims), key=lambda item: -item[1])***

***print(sims) # print sorted (document number, similarity score) 2-tuples***

***​***

***2017-12-07 14:05:24,066 : INFO : adding document #0 to Dictionary(0 unique tokens: [])***

***2017-12-07 14:05:24,067 : INFO : built Dictionary(143 unique tokens: ['title', 'wyeth', 'symposium', 'metabolic', 'dysregulation']...) from 21 documents (total 232 corpus positions)***

***2017-12-07 14:05:24,079 : INFO : saving Dictionary object under C:/Users/lohit/AppData/Local/Temp/den.dict, separately None***

***2017-12-07 14:05:24,112 : INFO : saved C:/Users/lohit/AppData/Local/Temp/den.dict***

***2017-12-07 14:05:24,420 : INFO : storing corpus in Matrix Market format to C:/Users/lohit/AppData/Local/Temp/den.mm***

***2017-12-07 14:05:24,424 : INFO : saving sparse matrix to C:/Users/lohit/AppData/Local/Temp/den.mm***

***2017-12-07 14:05:24,425 : INFO : PROGRESS: saving document #0***

***2017-12-07 14:05:24,426 : INFO : saved 21x143 matrix, density=5.661% (170/3003)***

***2017-12-07 14:05:24,433 : INFO : saving MmCorpus index to C:/Users/lohit/AppData/Local/Temp/den.mm.index***

***2017-12-07 14:05:24,440 : INFO : collecting document frequencies***

***2017-12-07 14:05:24,440 : INFO : PROGRESS: processing document #0***

***2017-12-07 14:05:24,440 : INFO : calculating IDF weights for 21 documents and 142 features (170 matrix non-zeros)***

***2017-12-07 14:05:24,443 : INFO : loading Dictionary object from C:\Users\lohit\AppData\Local\Temp\den.dict***

***2017-12-07 14:05:24,445 : INFO : loaded C:\Users\lohit\AppData\Local\Temp\den.dict***

***2017-12-07 14:05:24,447 : INFO : loaded corpus index from C:\Users\lohit\AppData\Local\Temp\den.mm.index***

***2017-12-07 14:05:24,448 : INFO : initializing corpus reader from C:\Users\lohit\AppData\Local\Temp\den.mm***

***2017-12-07 14:05:24,449 : INFO : accepted corpus with 21 documents, 143 features, 170 non-zero entries***

***2017-12-07 14:05:24,456 : INFO : collecting document frequencies***

***2017-12-07 14:05:24,461 : INFO : PROGRESS: processing document #0***

***2017-12-07 14:05:24,464 : INFO : calculating IDF weights for 21 documents and 142 features (170 matrix non-zeros)***

***2017-12-07 14:05:24,466 : INFO : using serial LSI version on this node***

***2017-12-07 14:05:24,469 : INFO : updating model with new documents***

***2017-12-07 14:05:24,474 : INFO : preparing a new chunk of documents***

***2017-12-07 14:05:24,475 : INFO : using 100 extra samples and 2 power iterations***

***2017-12-07 14:05:24,477 : INFO : 1st phase: constructing (143, 150) action matrix***

***2017-12-07 14:05:24,482 : INFO : orthonormalizing (143, 150) action matrix***

***2017-12-07 14:05:24,488 : INFO : 2nd phase: running dense svd on (143, 21) matrix***

***2017-12-07 14:05:24,492 : INFO : computing the final decomposition***

***2017-12-07 14:05:24,494 : INFO : keeping 8 factors (discarding 0.000% of energy spectrum)***

***2017-12-07 14:05:24,496 : INFO : processed documents up to #21***

***2017-12-07 14:05:24,497 : INFO : topic #0(1.398): 0.575\*"boston" + 0.522\*"university" + 0.190\*"wyeth" + 0.183\*"dysregulation" + 0.183\*"symposium" + 0.176\*"title" + 0.176\*"presented" + 0.162\*"metabolic" + 0.113\*"department" + 0.113\*"therapeutic"***

***2017-12-07 14:05:24,500 : INFO : topic #1(1.014): -0.305\*"chairman" + -0.305\*"hosted" + -0.305\*"h" + -0.305\*"farb" + -0.271\*"dr" + -0.248\*"experimental" + -0.248\*"therapeutic" + -0.248\*"department" + -0.241\*"david" + 0.234\*"university"***

***2017-12-07 14:05:24,502 : INFO : topic #2(1.001): -0.346\*"public" + -0.173\*"event" + -0.173\*"together" + -0.173\*"leading" + -0.173\*"am" + -0.173\*"obesity" + -0.173\*"impact" + -0.173\*"recent" + -0.173\*"inflammation" + -0.173\*"diabetes"***

***2017-12-07 14:05:24,504 : INFO : topic #3(1.000): 0.548\*"register" + 0.548\*"httpwwwbumcbuedubusmpmwyeth" + 0.548\*"go" + -0.183\*"digravio" + -0.183\*"source" + -0.183\*"gina" + 0.000\*"public" + 0.000\*"leading" + 0.000\*"am" + 0.000\*"500"***

***2017-12-07 14:05:24,506 : INFO : topic #4(1.000): 0.548\*"gina" + 0.548\*"source" + 0.548\*"digravio" + 0.183\*"httpwwwbumcbuedubusmpmwyeth" + 0.183\*"register" + 0.183\*"go" + 0.000\*"hosted" + 0.000\*"chairman" + 0.000\*"h" + 0.000\*"farb"***

***2017-12-07 14:05:24,540 : INFO : topic #0(1.398): 0.575\*"boston" + 0.522\*"university" + 0.190\*"wyeth" + 0.183\*"dysregulation" + 0.183\*"symposium" + 0.176\*"title" + 0.176\*"presented" + 0.162\*"metabolic" + 0.113\*"department" + 0.113\*"therapeutic"***

***2017-12-07 14:05:24,544 : INFO : topic #1(1.014): -0.305\*"chairman" + -0.305\*"hosted" + -0.305\*"h" + -0.305\*"farb" + -0.271\*"dr" + -0.248\*"experimental" + -0.248\*"therapeutic" + -0.248\*"department" + -0.241\*"david" + 0.234\*"university"***

***2017-12-07 14:05:24,546 : INFO : topic #2(1.001): -0.346\*"public" + -0.173\*"event" + -0.173\*"together" + -0.173\*"leading" + -0.173\*"am" + -0.173\*"obesity" + -0.173\*"impact" + -0.173\*"recent" + -0.173\*"inflammation" + -0.173\*"diabetes"***

***2017-12-07 14:05:24,547 : INFO : topic #3(1.000): 0.548\*"register" + 0.548\*"httpwwwbumcbuedubusmpmwyeth" + 0.548\*"go" + -0.183\*"digravio" + -0.183\*"source" + -0.183\*"gina" + 0.000\*"public" + 0.000\*"leading" + 0.000\*"am" + 0.000\*"500"***

***2017-12-07 14:05:24,549 : INFO : topic #4(1.000): 0.548\*"gina" + 0.548\*"source" + 0.548\*"digravio" + 0.183\*"httpwwwbumcbuedubusmpmwyeth" + 0.183\*"register" + 0.183\*"go" + 0.000\*"hosted" + 0.000\*"chairman" + 0.000\*"h" + 0.000\*"farb"***

***2017-12-07 14:05:24,550 : INFO : topic #5(0.957): -0.404\*"phd" + -0.354\*"professor" + -0.202\*"ma" + -0.202\*"md" + -0.191\*"dr" + -0.184\*"research" + -0.152\*"biology" + -0.152\*"medicine" + -0.152\*"school" + -0.152\*"neurology"***

***2017-12-07 14:05:24,552 : INFO : topic #6(0.801): 0.492\*"title" + 0.492\*"presented" + 0.237\*"wyeth" + -0.228\*"boston" + 0.204\*"symposium" + 0.200\*"dysregulation" + 0.175\*"metabolic" + -0.153\*"university" + -0.138\*"one" + -0.138\*"trustee"***

***2017-12-07 14:05:24,553 : INFO : topic #7(0.676): -0.390\*"university" + -0.265\*"boston" + 0.230\*"ballroom" + 0.230\*"street" + 0.230\*"october" + 0.230\*"present" + 0.230\*"mass" + 0.230\*"sherborn" + 0.230\*"trustee" + 0.230\*"22nd"***

***2017-12-07 14:05:24,554 : INFO : saving Projection object under C:\Users\lohit\AppData\Local\Temp\model.lsi.projection, separately None***

***2017-12-07 14:05:24,569 : INFO : saved C:\Users\lohit\AppData\Local\Temp\model.lsi.projection***

***2017-12-07 14:05:24,571 : INFO : saving LsiModel object under C:\Users\lohit\AppData\Local\Temp\model.lsi, separately None***

***2017-12-07 14:05:24,573 : INFO : not storing attribute projection***

***2017-12-07 14:05:24,575 : INFO : not storing attribute dispatcher***

***2017-12-07 14:05:24,584 : INFO : saved C:\Users\lohit\AppData\Local\Temp\model.lsi***

***2017-12-07 14:05:24,584 : INFO : loading Dictionary object from C:/Users/lohit/AppData/Local/Temp/den.dict***

***2017-12-07 14:05:24,588 : INFO : loaded C:/Users/lohit/AppData/Local/Temp/den.dict***

***2017-12-07 14:05:24,590 : INFO : loaded corpus index from C:/Users/lohit/AppData/Local/Temp/den.mm.index***

***2017-12-07 14:05:24,591 : INFO : initializing corpus reader from C:/Users/lohit/AppData/Local/Temp/den.mm***

***2017-12-07 14:05:24,593 : INFO : accepted corpus with 21 documents, 143 features, 170 non-zero entries***

***2017-12-07 14:05:24,598 : WARNING : scanning corpus to determine the number of features (consider setting `num\_features` explicitly)***

***2017-12-07 14:05:24,601 : INFO : creating matrix with 21 documents and 8 features***

***2017-12-07 14:05:24,605 : INFO : saving MatrixSimilarity object under C:/Users/lohit/AppData/Local/Temp/den.index, separately None***

***2017-12-07 14:05:24,612 : INFO : saved C:/Users/lohit/AppData/Local/Temp/den.index***

***2017-12-07 14:05:24,612 : INFO : loading MatrixSimilarity object from C:/Users/lohit/AppData/Local/Temp/den.index***

***2017-12-07 14:05:24,617 : INFO : loaded C:/Users/lohit/AppData/Local/Temp/den.index***

***{'title': 0, 'wyeth': 1, 'symposium': 2, 'metabolic': 3, 'dysregulation': 4, 'presented': 5, 'boston': 6, 'university': 7, 'department': 8, 'pharmacology': 9, 'experimental': 10, 'therapeutic': 11, 'present': 12, 'october': 13, '22nd': 14, 'trustee': 15, 'ballroom': 16, 'one': 17, 'sherborn': 18, 'street': 19, 'mass': 20, 'free': 21, 'event': 22, '830': 23, 'am': 24, '500': 25, 'pm': 26, 'bring': 27, 'together': 28, 'leading': 29, 'scientist': 30, 'pharmaceutical': 31, 'company': 32, 'executive': 33, 'public': 34, 'forum': 35, 'discus': 36, 'recent': 37, 'research': 38, 'impact': 39, 'obesity': 40, 'diabetes': 41, 'heart': 42, 'disease': 43, 'cancer': 44, 'inflammation': 45, 'open': 46, 'lecturer': 47, 'include': 48, 'dr': 49, 'steven': 50, 'a': 51, 'kliewer': 52, 'md': 53, 'nancy': 54, 'b': 55, 'jake': 56, 'l': 57, 'hamon': 58, 'distinguished': 59, 'chair': 60, 'basic': 61, 'professor': 62, 'molecular': 63, 'biology': 64, 'texas': 65, 'southwestern': 66, 'medical': 67, 'center': 68, 'dallas': 69, 'tx': 70, 'gary': 71, 'e': 72, 'landreth': 73, 'phd': 74, 'neuroscience': 75, 'neurology': 76, 'case': 77, 'western': 78, 'reserve': 79, 'cleveland': 80, 'oh': 81, 'timothy': 82, 'f': 83, 'osborne': 84, 'biochemistry': 85, 'california': 86, 'irvine': 87, 'ca': 88, 'giulio': 89, 'm': 90, 'pasinetti': 91, 'psychiatry': 92, 'geriatrics': 93, 'adult': 94, 'development': 95, 'mount': 96, 'sinai': 97, 'school': 98, 'medicine': 99, 'new': 100, 'york': 101, 'ny': 102, 'pere': 103, 'puigserver': 104, 'assistant': 105, 'cell': 106, 'dana': 107, 'farber': 108, 'institute': 109, 'ma': 110, 'david': 111, 'r': 112, 'riddell': 113, 'principal': 114, 'ii': 115, 'discovery': 116, 'princeton': 117, 'nj': 118, 'sudha': 119, 'seshadri': 120, 'associate': 121, 'andrei': 122, 'ruckenstein': 123, 'provost': 124, 'vice': 125, 'president': 126, 'james': 127, 'tobin': 128, 'cardiovascular': 129, 'cambridge': 130, 'benjamin': 131, 'wolozin': 132, 'hosted': 133, 'h': 134, 'farb': 135, 'chairman': 136, 'register': 137, 'go': 138, 'httpwwwbumcbuedubusmpmwyeth': 139, 'source': 140, 'gina': 141, 'digravio': 142}***

***Folder "C:\Users\lohit\AppData\Local\Temp" will be used to save temporary dictionary and corpus.***

***Used files generated before***

***[(0, 0.10438020609840562), (1, -0.15367473858110067), (2, -0.31691870729279276), (5, -0.32189135004841918), (6, -0.016264913126184739), (7, 0.065483330271116857)]***

***2017-12-07 14:05:24,700 : INFO : adding document #0 to Dictionary(0 unique tokens: [])***

***2017-12-07 14:05:24,705 : INFO : built Dictionary(301 unique tokens: ['title', 'fructose', 'hamper', 'hormone', 'control']...) from 43 documents (total 539 corpus positions)***

***2017-12-07 14:05:24,705 : INFO : saving Dictionary object under C:/Users/lohit/AppData/Local/Temp/den.dict, separately None***

***2017-12-07 14:05:24,713 : INFO : saved C:/Users/lohit/AppData/Local/Temp/den.dict***

***2017-12-07 14:05:24,713 : INFO : storing corpus in Matrix Market format to C:/Users/lohit/AppData/Local/Temp/den.mm***

***2017-12-07 14:05:24,717 : INFO : saving sparse matrix to C:/Users/lohit/AppData/Local/Temp/den.mm***

***2017-12-07 14:05:24,719 : INFO : PROGRESS: saving document #0***

***2017-12-07 14:05:24,720 : INFO : saved 43x301 matrix, density=3.554% (460/12943)***

***2017-12-07 14:05:24,734 : INFO : saving MmCorpus index to C:/Users/lohit/AppData/Local/Temp/den.mm.index***

***2017-12-07 14:05:24,740 : INFO : collecting document frequencies***

***2017-12-07 14:05:24,747 : INFO : PROGRESS: processing document #0***

***2017-12-07 14:05:24,749 : INFO : calculating IDF weights for 43 documents and 300 features (460 matrix non-zeros)***