# -\*- coding: utf-8 -\*-

"""Topic\_modelling.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1onywzH0WkzZEAiXYNfZOjNelLBuvQ3rV

"""

from bs4 import BeautifulSoup as bs

import requests

import pandas as pd

import matplotlib.pyplot as plt

urls=["https://prsindia.org/policy/vital-stats/overview-road-accidents-india",

"https://timesofindia.indiatimes.com/topic/car-accident/news"]

headers\_={'user-agent':'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)'}

for url in urls:

page=requests.get(url, headers=headers\_)

soup=bs(page.content,'html.parser')

Contents=soup.findAll("p")

Scrapped\_data=[]

for content in Contents:

Text=content.get\_text()

Scrapped\_data.append([Text])

df=pd.DataFrame(Scrapped\_data,columns=["Text"])

display(df)

df.to\_json("ins\_company.json")

import string

string.punctuation

def remove\_punctuation(text):

punctuationfree="".join([i for i in text if i not in string.punctuation])

return punctuationfree

#storing the puntuation free text

df['puntuation free text']= df['Text'].apply(lambda x:remove\_punctuation(x))

#Lowering the book name

df['lower']= df['puntuation free text'].apply(lambda x: x.lower())

#Tokenization

from nltk.tokenize import TweetTokenizer as tt

#applying function to the column

tokenizer = tt()

df['tokenized\_books'] = df['lower'].apply(lambda x: tokenizer.tokenize(x))

#Removing stop words

import nltk

from nltk.corpus import stopwords

nltk.download('stopwords')

stopword = stopwords.words('english')

def remove\_stopwords(text):

output= [i for i in text if i not in stopword]

return output

df['no\_stopwords']= df['tokenized\_books'].apply(lambda x:remove\_stopwords(x))

#Stemming

from nltk.stem.porter import PorterStemmer

porter\_stemmer = PorterStemmer()

#Defining a function for stemming

def stemming(text):

stem\_tweet = [porter\_stemmer.stem(word) for word in text]

return stem\_tweet

df['stemmed\_books']=df['no\_stopwords'].apply(lambda x: stemming(x))

#Lemmatization

from nltk.stem import WordNetLemmatizer

nltk.download('wordnet')

nltk.download('omw-1.4')

wordnet\_lemmatizer = WordNetLemmatizer()

def lemmatizer(stemmed\_books):

lemm\_text = [wordnet\_lemmatizer.lemmatize(word) for word in stemmed\_books]

return lemm\_text

df['lemmatized\_books']=df['stemmed\_books'].apply(lambda x:lemmatizer(x))

display(df.head())

!pip install pyLDAvis

import numpy as np

import json

import glob

#Gensim

import gensim

import gensim.corpora as corpora

from gensim.utils import simple\_preprocess

from gensim.models import CoherenceModel

#spacy

import spacy

import nltk

from nltk.corpus import stopwords

#Vis

import pyLDAvis

import pyLDAvis.gensim

import warnings

warnings.filterwarnings("ignore",category=DeprecationWarning)

def load\_data(file):

with open(file,"r",encoding="utf-8") as f:

data=json.load(f)

return(data)

def write\_data(file,data):

with open(file,"w",encoding="utf-8") as f:

json.dump(data,f,indent=4)

nltk.download("stopwords")

print(stopwords)

data=load\_data("//content/ins\_company.json")["Text"]

data=data["5"]

data=data.split(".")

data

def lemmatization(texts,allowed\_postages=["NOUN","ADJ","VERB","ADV"]):

nlp=spacy.load("en\_core\_web\_sm",disable=["senter","ner","parser"])

texts\_out=[]

for text in texts:

doc=nlp(text)

new\_text=[]

for token in doc:

if token.pos\_ in allowed\_postages:

new\_text.append(token.lemma\_)

final=" ".join(new\_text)

texts\_out.append(final)

return (texts\_out)

lemmatized\_texts=lemmatization(data)

lemmatized\_texts

def gen\_words(texts):

final=[]

for text in texts:

new=gensim.utils.simple\_preprocess(text,deacc=True)

final.append(new)

return(final)

data\_words=gen\_words(lemmatized\_texts)

print(data\_words)

id2word=corpora.Dictionary(data\_words)

corpus=[]

for text in data\_words:

new = id2word.doc2bow(text)

corpus.append(new)

print(corpus)

word = id2word

print(word)

lda\_model= gensim.models.ldamodel.LdaModel(corpus=corpus,id2word=id2word,num\_topics=30,random\_state=100,update\_every=1,

chunksize=100,passes=10,alpha="auto")

pyLDAvis.enable\_notebook()

vis=pyLDAvis.gensim.prepare(lda\_model,corpus,id2word,mds="mmds",R=30)

vis