**LATENT DRICHITEL ALLOCATION:(Python File)**

By applying this algorithm the words are categorised into similar topics

#!/usr/bin/env python

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

import warnings

warnings.filterwarnings(action='ignore', category=UserWarning, module='gensim')

import nltk

import gensim

from gensim import corpora, models

from nltk.corpus import stopwords

from nltk.stem.wordnet import WordNetLemmatizer

import string

filepath = 'E:/project/preprocess4/pp1/pp2/pp3/pp4/pp5/tt25.txt'

## Open the file with read only permit

f = open(filepath, encoding="utf8")

## use readlines to read all lines in the file

## The variable "lines" is a list containing all lines

doc = f.readlines()

texts = [text for text in doc if len(text) > 2]

doc\_clean = [doc.split() for doc in texts]

dictionary = corpora.Dictionary(doc\_clean)

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

ldamodel = models.ldamodel.LdaModel(doc\_term\_matrix, num\_topics=5, id2word = dictionary, passes=5)

#print(ldamodel.show\_topics)

topicfilepath = 'E:/project/preprocess4/pp1/pp2/pp3/pp4/pp5/final/fp25.txt'

topicf = open(topicfilepath, "w")

i=0

for topic in ldamodel.show\_topics(num\_topics=5, formatted=False, num\_words=6):

print("Topic {}: Words: ".format(topic[0]))

#print(ldamodel.print\_topics(num\_topics=2, num\_words=4))

topicf.write("\nTopic "+str(i)+": Words: \n")

topicwords = [w for (w, val) in topic[1]]

for j in range(0,6):

print(topicwords[j]+ " ")

topicf.writelines(topicwords[j]+ " ")

i = i+1

f.close()

topicf.close()