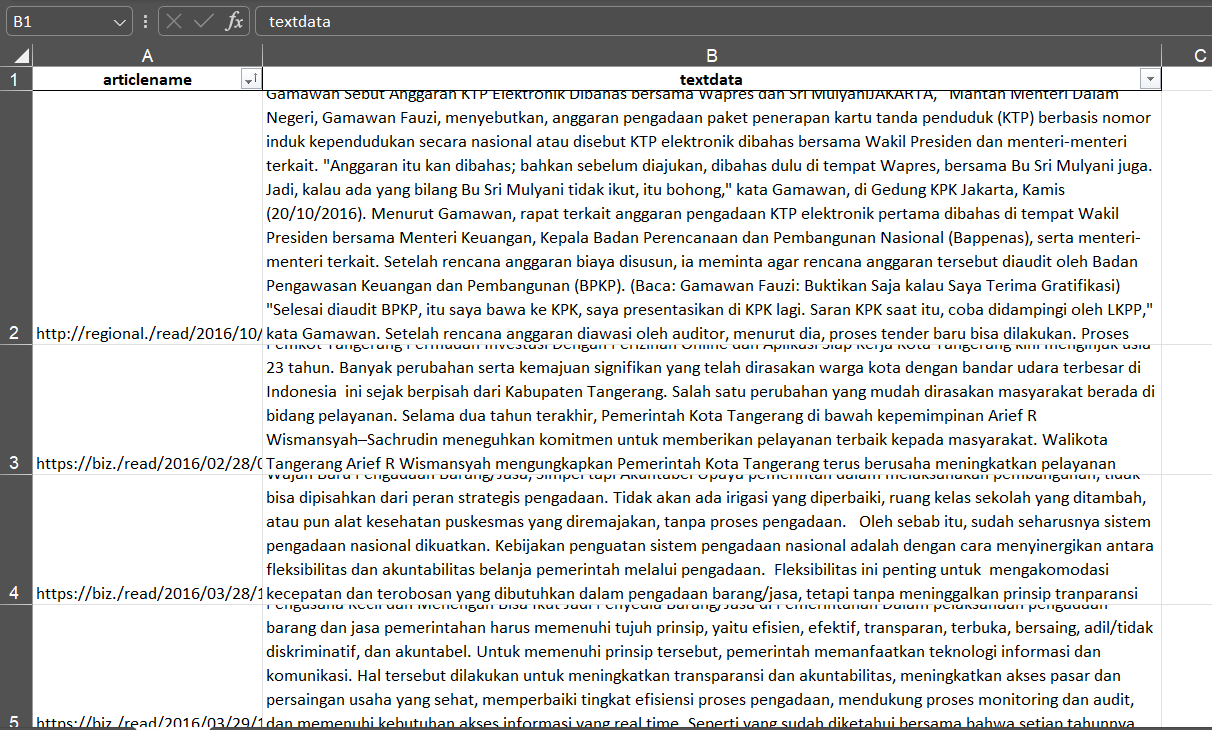
MODELING ARTICLE NEWS DENGAN LDA

Berikut hasil pengolahan data kumpulan dari beberapa artikel dengan menggunakan py



Berikut scrypt pengolahan data dengan menggunakan Python

import gensim

from gensim.utils import simple\_preprocess

from gensim.corpora import Dictionary

import pandas as pd

import numpy as np

import nltk

import string

import re #regex library

# import word\_tokenize & FreqDist from NLTK

from nltk.tokenize import word\_tokenize

# from nltk.probability import FreqDist

from nltk.corpus import stopwords

import pyLDAvis.gensim\_models as gensimvis

import pyLDAvis

import warnings

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

data = pd.read\_excel("dataBerita.xlsx")

data = pd.DataFrame(data)

data = data.drop(columns=["articlename"])

# print(df\_imp\_wcount)

# pre-processing

# bikin jadi lower semua data

data["textdata"]= data["textdata"].str.lower()

# TOKENISASI

nltk.download('punkt')

def remove\_tweet\_special(text):

    # remove tab, new line, ans back slice

    text = text.replace('\\t'," ").replace('\\n'," ").replace('\\u'," ").replace('\\',"")

    # remove non ASCII (emoticon, chinese word, .etc)

    text = text.encode('ascii', 'replace').decode('ascii')

    # remove mention, link, hashtag

    # text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)"," ", text).split())

    # remove incomplete URL

    return text.replace("http://", " ").replace("https://", " ")

data['textdata'] = data['textdata'].apply(remove\_tweet\_special)

#menghilankan number pada text

def remove\_number(text):

    return  re.sub(r"\d+", "", text)

data['textdata'] = data['textdata'].apply(remove\_number)

#remove tanda baca

def remove\_punctuation(text):

    return text.translate(str.maketrans("","",string.punctuation))

data['textdata'] = data['textdata'].apply(remove\_punctuation)

#remove whitespace

def remove\_whitespace\_LT(text):

    return text.strip()

data['textdata'] = data['textdata'].apply(remove\_whitespace\_LT)

# remove single char

def remove\_singl\_char(text):

    return re.sub(r"\b[a-zA-Z]\b", "", text)

data['textdata'] = data['textdata'].apply(remove\_singl\_char)

# NLTK word tokenize

def word\_tokenize\_wrapper(text):

    return word\_tokenize(text)

data['textdata\_tokens'] = data['textdata'].apply(word\_tokenize\_wrapper)

nltk.download('stopwords')

from nltk.corpus import stopwords

list\_stopwords = stopwords.words('indonesian')

list\_stopwords.extend(["yg", "dg", "rt", "dgn", "ny", "d", 'klo',

                       'kalo', 'amp', 'biar', 'bikin', 'bilang',

                       'gak', 'ga', 'krn', 'nya', 'nih', 'sih',

                       'si', 'tau', 'tdk', 'tuh', 'utk', 'ya',

                       'jd', 'jgn', 'sdh', 'aja', 'n', 't',

                       'nyg', 'hehe', 'pen', 'u', 'nan', 'loh', 'rt',

                       '&amp', 'yah', 'bisnis', 'pandemi', 'indonesia',

                       "ada", "tan", "ton", "pt", "komentar", "juta", "unit", "menang", "artikel",

                       "smartphone", "tagar", "sedia", "kaskus", "seksi"])

# convert list to dictionary

list\_stopwords = set(list\_stopwords)

#remove stopword pada list token

def stopwords\_removal(words):

    return [word for word in words if word not in list\_stopwords]

data['textdata\_tokens\_WSW'] = data['textdata\_tokens'].apply(stopwords\_removal)

from gensim import corpora

doc\_clean = data['textdata\_tokens\_WSW']

dictionary = corpora.Dictionary(doc\_clean)

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

# print(dictionary)

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

# Creating the object for LDA model using gensim library

Lda = gensim.models.ldamodel.LdaModel

total\_topics = 3 # jumlah topik yang akan di extract

number\_words = 10 # jumlah kata per topik

# Running and Trainign LDA model on the document term matrix.

lda\_model = Lda(doc\_term\_matrix, num\_topics=total\_topics, id2word = dictionary, passes=50)

for idx, topic in lda\_model.print\_topics(-1):

    print(f"Topik {idx}: {topic}")

# Visualisasi dengan pyLDAvis

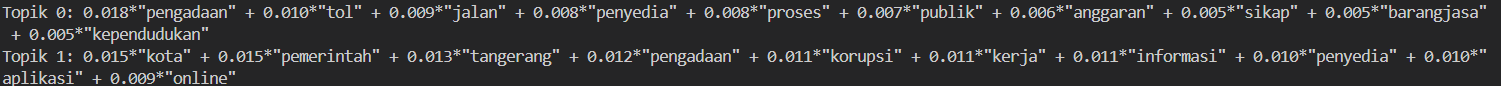
vis\_data = gensimvis.prepare(lda\_model, corpus, dictionary)

pyLDAvis.display(vis\_data)

# Menyimpan visualisasi ke dalam file HTML

pyLDAvis.save\_html(vis\_data, 'lda\_visualization.html')

output dengan ditentukan 2 topik pada lda sebagai berikut:



secara jelas seperti ini:

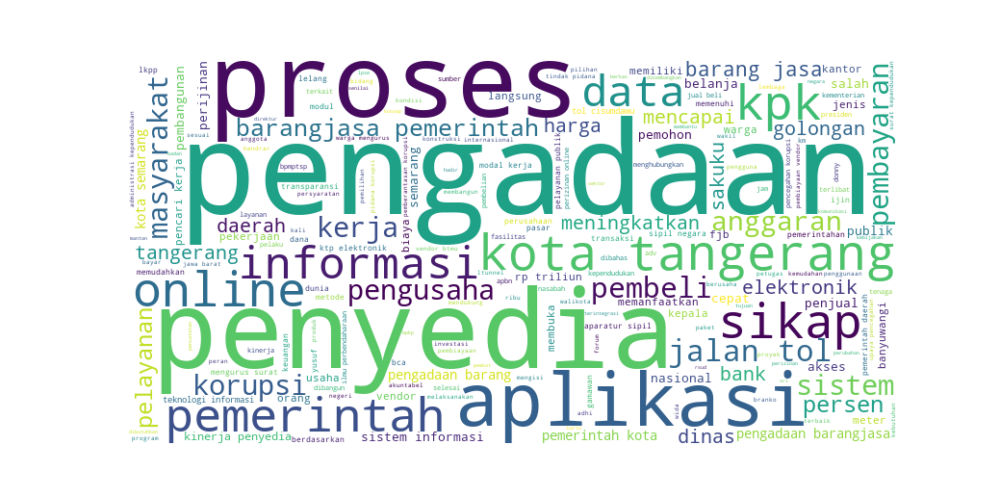
Topik 0:

0.018\*"pengadaan" + 0.010\*"tol" + 0.009\*"jalan" + 0.008\*"penyedia" + 0.008\*"proses" + 0.007\*"publik" + 0.006\*"anggaran" + 0.005\*"sikap" + 0.005\*"barangjasa" + 0.005\*"kependudukan"

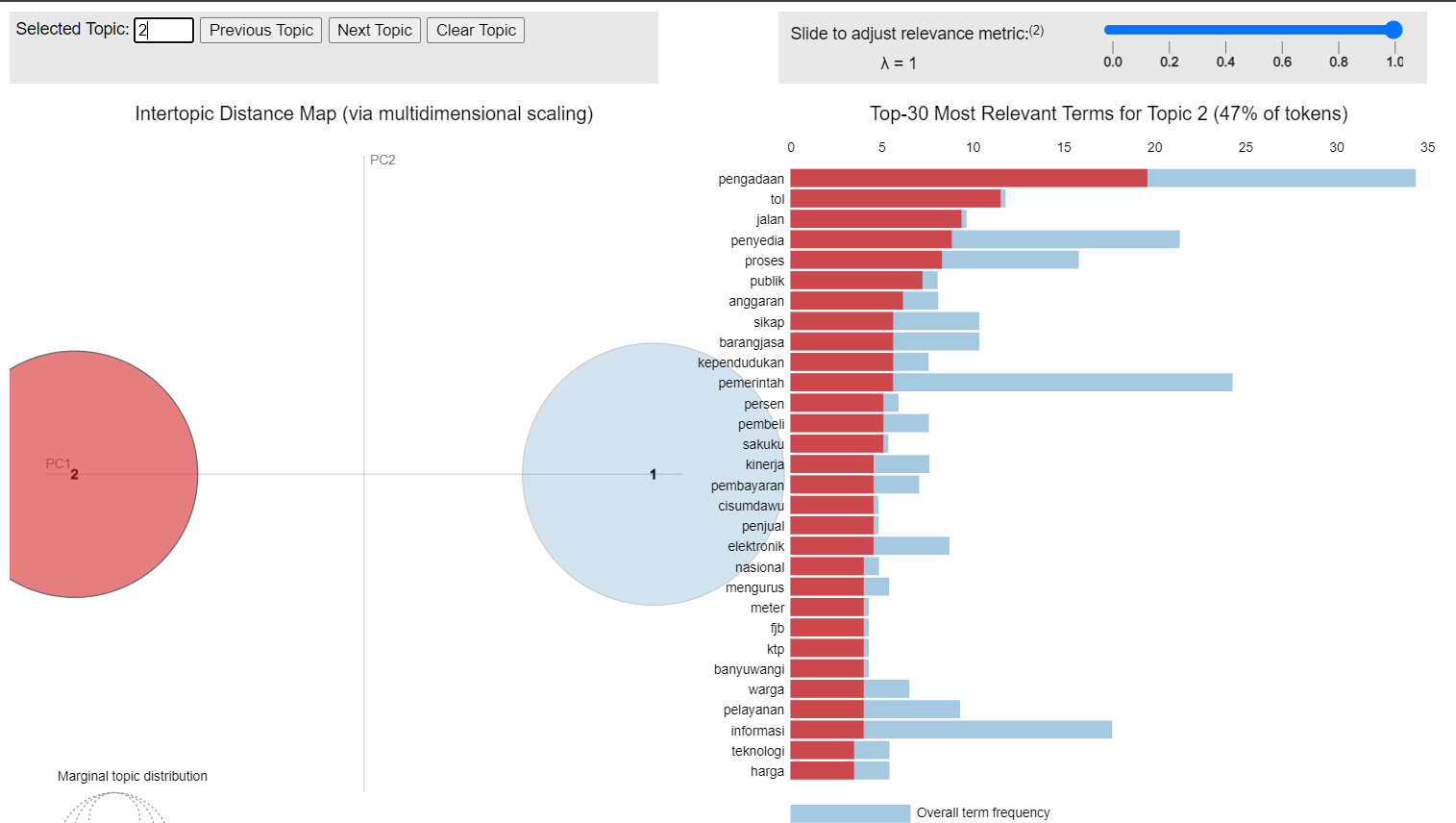
Topik 1:

0.015\*"kota" + 0.015\*"pemerintah" + 0.013\*"tangerang" + 0.012\*"pengadaan" + 0.011\*"korupsi" + 0.011\*"kerja" + 0.011\*"informasi" + 0.010\*"penyedia" + 0.010\*"aplikasi" + 0.009\*"online"

Bagword yang dihasilkan:



Dan penggambaran secara visualisasi nya seperti dibawah ini



Dari output gambar bisa dilihat frekuensi topik yang yang relevant dan perlu mencari sumber yang lebih banyak