

Wlildan Miftakhurahman

21110040

Convert Dataset

```
import csv

# Baca file CSV
csv_file_path = 'dataset_mentalhealth.csv'

with open(csv_file_path, 'r', encoding='utf-8') as csv_file:
    csv_reader = csv.DictReader(csv_file)

    # Tentukan kolom yang ingin diambil
    kolom_jawaban = 'Jawaban'

    # Buat list untuk menyimpan nilai kolom Jawaban
    jawaban_list = [row[kolom_jawaban] for row in csv_reader]

# Buat file teks dengan nilai kolom Jawaban
txt_file_path = 'output.txt'

with open(txt_file_path, 'w', encoding='utf-8') as txt_file:
    for jawaban in jawaban_list:
        txt_file.write(jawaban + '\n')

print(f'File teks telah berhasil dibuat di: {txt_file_path}')

File teks telah berhasil dibuat di: output.txt
```

Import Libraries

```
import io
import random
import string
import warnings
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import warnings
warnings.filterwarnings('ignore')

import nltk #import library NLTK
from nltk.stem import WordNetLemmatizer #import library untuk
lemmatization
nltk.download('popular', quiet=True) # for downloading packages
```

```

[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data] C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data] Package wordnet is already up-to-date!

True

f=open('output.txt','r',errors = 'ignore') #membuka file corpus dari
wikipedia
raw=f.read() #raw kini berisi semua data dari corpus per baris (raw)
raw = raw.lower()# converts to lowercase

```

Tokenisasi

```

#tokenisasi adalah memilah-milah dokumen ke kalimat-kalimat,
#kemudian memilah setiap kalimat menjadi sekumpulan kata kata
sent_tokens = nltk.sent_tokenize(raw) # converts dokumen corpus ke
kalimat-kalimat
word_tokens = nltk.word_tokenize(raw)# converts dokumen corpus ke
kata-kata

```

Preprocessing

```

lemmer = nltk.stem.WordNetLemmatizer()
#WordNet is a semantically-oriented dictionary of English included in
NLTK.
def LemTokens(tokens):
    return [lemmer.lemmatize(token) for token in tokens]
remove_punct_dict = dict((ord(punct), None) for punct in
string.punctuation)
def LemNormalize(text):
    return
LemTokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)
))

```

Keyword matching

```

# kata-kata pembuka didaftar terlebih dulu dan kemudian secara acak
diberikan respon jawabannya
GREETING_INPUTS = ("hello", "hi", "greetings", "sup", "what's
up","hey", "hai")
GREETING_RESPONSES = ["hi", "hey", "hi there", "hello"]
def greeting(sentence):
    for word in sentence.split():
        if word.lower() in GREETING_INPUTS:
            return random.choice(GREETING_RESPONSES)

```

Modeling dengan TF-IDF dan Cosine Similarity

```
def response(user_response):
    robo_response='' #pada tahap awal respon mesin diisi karakter
    kosong
    sent_tokens.append(user_response) #pertanyaan user ditokenisasi
    dan ditambahkan di corpus
    TfidfVec = TfidfVectorizer(tokenizer=LemNormalize,
    stop_words='indonesian')
    tfidf = TfidfVec.fit_transform(sent_tokens) #token dari pertanyaan
    user di vektorisasi
    vals = cosine_similarity(tfidf[-1], tfidf) #hitung similarity
    setiap token corpus dengan token pertanyaan
    idx=vals.argsort()[0][-2] #sort jarak similariti setiap token
    corpus dengan token pertanyaan
    flat = vals.flatten()
    flat.sort()
    req_tfidf = flat[-2]
    if(req_tfidf==0): #jika pertanyaan dan semua token corpus jaraknya
    tinggi maka
        #berarti pertanyaan tidak ada jawabannya
        robo_response=robo_response+"Mohon maaf, saya tidak paham
    pertanyaan anda"
        return robo_response
    else:
        robo_response = robo_response+sent_tokens[idx] #jika jaraknya
    terrendah maka dipakai sebagai jawaban
        return robo_response
```

Uji Coba / Evaluasi Model

```
flag=True
print("===== \n Mesin: Ini adalah layanan autochat,
    untuk mengakhiri sesi, ketik selesai")
while(flag==True):
    user_response = input("Masukkan pertanyaan :")
    user_response=user_response.lower()
    print('user :', user_response)
    if(user_response!='selesai'): #jika user tidak keluar
        if(user_response=='thanks' or user_response=='thank you' ):
        #jika ucapkan thanks/thankyou
            flag=False #tanda proses berhenti
            print("Mesin: You are welcome..") #balasan thank you
        else:
            if(greeting(user_response)!=None): #jika response adalah
            kalimat greeting
                print("Mesin: "+greeting(user_response)) #tampilkan
            kalimat greeting
            else: #jika bukan kalimat greeting
```

```

        print("Mesin: ",end="")
        print(response(user_response)) #memproses user answer
disini
        sent_tokens.remove(user_response) #user answer dihapus
setelah dimunculkan
    else:
        flag=False
        print("Mesin: Terimakasih sudah menggunakna layanan chatbot")
        print("=====")

```

=====

Mesin: Ini adalah layanan autochat, untuk mengakhiri sesi, ketik selesai

user : adakah kelompok pendukung bagi penyandang gangguan mental?

Mesin: ada kelompok pendukung untuk orang-orang dengan pengalaman penyakit mental, kelompok pendukung untuk orang dengan diagnosis khusus, kelompok pendukung untuk anggota keluarga dan teman, dan banyak lagi.

user : apa yang dimaksud dengan bersepeda cepat?

Mesin: bersepeda yang cepat dapat terjadi kapan saja seseorang mengalami gangguan bipolar-sekitar 10-20% orang yang didiagnosis dengan gangguan bipolar mengalami bersepeda cepat di beberapa titik.

user : apa perbedaan kesehatan mental vs penyakit mental?

Mesin: sama seperti itu mungkin untuk memiliki kesehatan mental yang buruk tetapi tidak ada penyakit mental, sangat mungkin untuk memiliki kesehatan mental yang baik bahkan dengan diagnosis penyakit mental.

user : apa saja sisi buruk asap ganja?

Mesin: asap ganja, misalnya, mengandung racun penyebab kanker.

user : apakah psikosis dapat diobati?

Mesin: psikosis dan skizofrenia dapat diobati.

user : penyakit mental adalah

Mesin: sama seperti itu mungkin untuk memiliki kesehatan mental yang buruk tetapi tidak ada penyakit mental, sangat mungkin untuk memiliki kesehatan mental yang baik bahkan dengan diagnosis penyakit mental.

user : definisi vaping

Mesin: "vaping" adalah istilah untuk menggunakan perangkat di mana cairan, sering dibumbui, diubah menjadi uap (karenanya, vaping) dan dihirup.

user : status minyak cbd?

Mesin: setelah meneliti keamanan dan efektivitas minyak cbd untuk mengobati epilepsi, pada tahun 2018, fda amerika serikat menyetujui cbd (epidiolex) sebagai terapi untuk dua kondisi langka yang ditandai dengan kejang epilepsi.

user : prodrome adalah

Mesin: psikosis, sekelompok gejala yang ditemukan dalam gangguan seperti skizofrenia, adalah satu penyakit dengan prodrome tertentu.

user : gangguan jiwa identitas adalah

Mesin: gangguan identitas disosiatif sebenarnya lebih tentang identitas yang terfragmentasi daripada banyak kepribadian berbeda yang berkembang sendiri.

```
user : saya suka main game
Mesin: bos saya memberi tahu saya bahwa saya melakukan kesalahan.
user : bandung lautan api
Mesin: Mohon maaf, saya tidak paham pertanyaan anda
user : selesai
Mesin: Terimakasih sudah menggunakna layanan chatbot
=====
```

Evaluasi Pertanyaan

Model dapat dianggap sukses karena memberikan jawaban yang relevan dan informatif dalam banyak kasus. Namun, preprocessing yang digunakan masih membiarkan banyak stopwords yang lolos, sehingga nilai TF-IDF kurang optimal. Ini dapat mengakibatkan jawaban yang kurang spesifik dan tidak sepenuhnya memanfaatkan informasi yang tersedia.

Selain itu, terdapat kasus di mana input pengguna dan jawaban mesin tidak sesuai. Misalnya, pada pertanyaan "saya suka main game", jawaban mesin tidak relevan dan mungkin disebabkan oleh ketidaksesuaian model terhadap jenis pertanyaan tersebut. Hal ini menunjukkan bahwa model mungkin perlu peningkatan untuk menangani variasi jenis pertanyaan dengan lebih baik.

Pada bagian akhir, pada pertanyaan "bandung lautan api", jawaban mesin tidak relevan dan menyebutkan kesalahan yang tidak terkait. Ini menunjukkan bahwa model tidak sepenuhnya memahami jenis pertanyaan tersebut dan memutuskan bahwa pertanyaan tidak sesuai sehingga memutuskan untuk mengeluarkan output "Mohon Maaf, saya tidak paham pertanyaan anda". yang bisa dipahami karna memang didataset pertanyaan seperti itu tidak ada