textmining-1

November 9, 2024

Objective : Scraping komen dari youtube, lalu cleaning data sebelum digunakan, dan pada akhirnya menjadi sebuah wordcloud yang dapat digunakan untuk mencari insight yang berguna untuk peluang bisnis ataupun riset.

[2]:
```
import googleapiclient.discovery
import pandas as pd

api_service_name = "youtube"
api_version = "v3"
DEVELOPER_KEY = "AIzaSyAmectm8AVwx9cZdJB7UKxWSHeW7rXOK7M"

#API key google
youtube = googleapiclient.discovery.build(
    api_service_name, api_version, developerKey=DEVELOPER_KEY)

#list id video untuk discrape
video_ids = ["_k2m7mKV_x0", "WH1SduDRL_Y", "aXLVJO7f21c",␣
 ↪"GhaP-u8SK9s","INh0A88iJC8","2PAu2zttES0","Ny12OVobIWc","fqF41VJufU4","lZ2rMlS9tZE","lJ6jsY
comments = []

#Looping setiap video id
for video_id in video_ids:
    request = youtube.commentThreads().list(
        part="snippet",
        videoId=video_id,
        #20 komen per video
        maxResults=20
    )
    response = request.execute()
    #kumpulkan komen
    for item in response['items']:
        comment = item['snippet']['topLevelComment']['snippet']
        comments.append([
            comment['textOriginal']
        ])

#Masukan komen yang sudah di scrape kedalam dataframe
df = pd.DataFrame(comments, columns=['text'])
```

```
df.info()  # Display DataFrame info to check the structure

df.to_csv("youtube_comments_submission.csv", index=False)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 154 entries, 0 to 153
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   text    154 non-null    object
dtypes: object(1)
memory usage: 1.3+ KB
```

[36]:
```
#Print 10 first rows of dataset.
df.head(10)
```

[36]:

	text
0	I'm a new subscriber ka. Thank you for sharing…
1	Rekomendasi ecourse machine learning dong kak
2	Kak, kuliah s2 dimana tu?
3	Aku background IT sampe sekarang masih 10% pen…
4	berarti kakak beruntung dibimbing waktu s2 ten…
5	Kita sejaman. Mohon bantuannya saya pengen bel…
6	kak helpful bangett!!
7	Great sharing
8	saya bkn background IT, sdg ikut bootcamp sela…
9	saya mau apply beasiswa antara front end dev /…

[ ]:
```
import re, string, nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
```

[37]:
```
def clean_text(text):
    #Hilangan data non-teks
    text = re.sub(r'[^A-Za-z\s]', '', text)
    #Ubah text ke huruf kecil
    text = text.lower()
    return text

def tokenize(text):
    #Tokenisasi teks
    return word_tokenize(text)
```

```
#Menghapus stopword yang tidak diperlukan dalam analsisa
def remove_stopwords(tokens):
    #Khusus stopword dalam bahasa Indonesia
    stop_words = set(stopwords.words('indonesian'))
    return [word for word in tokens if word not in stop_words]

#Memecah kata-kata menjadi kata-kata sesungguhnya tanpa tambahan.
def lemmatize(tokens):
    lemmatizer = WordNetLemmatizer()
    return [lemmatizer.lemmatize(word) for word in tokens]

#Jalankan function untuk preprocessing text
df['cleaned_text'] = df['text'].apply(clean_text)
df['tokens'] = df['cleaned_text'].apply(tokenize)
df['tokens'] = df['tokens'].apply(remove_stopwords)
df['lemmatized_text'] = df['tokens'].apply(lemmatize)

df[['text', 'cleaned_text', 'tokens', 'lemmatized_text']].head()
```

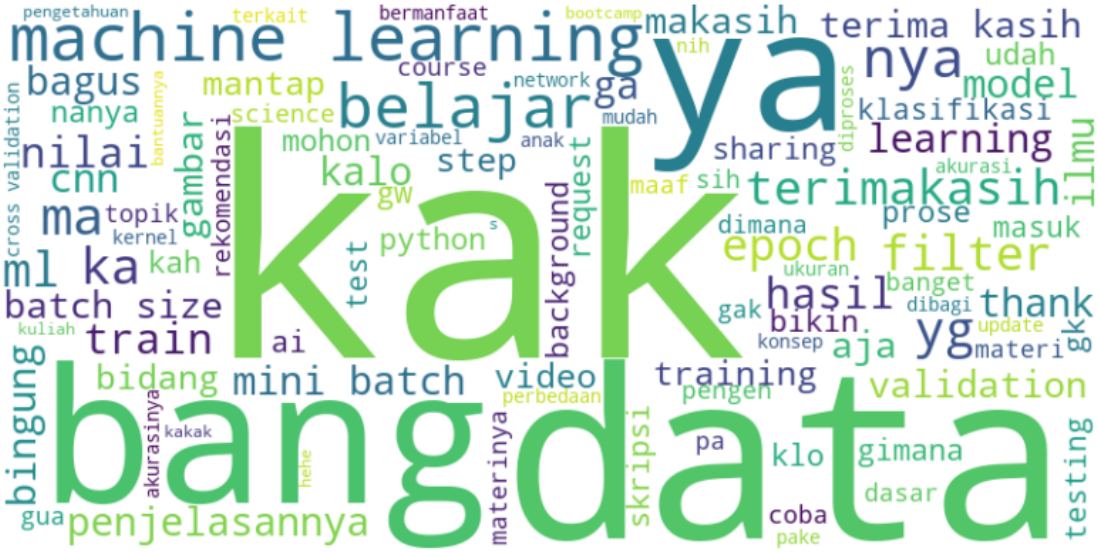
```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\cornelius\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\cornelius\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\cornelius\AppData\Roaming\nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

[37]:

```
                                                text  \
0  I'm a new subscriber ka. Thank you for sharing…
1      Rekomendasi ecourse machine learning dong kak
2                          Kak, kuliah s2 dimana tu?
3  Aku background IT sampe sekarang masih 10% pen…
4  berarti kakak beruntung dibimbing waktu s2 ten…

                                        cleaned_text  \
0  im a new subscriber ka thank you for sharing \…
1      rekomendasi ecourse machine learning dong kak
2                            kak kuliah s dimana tu
3  aku background it sampe sekarang masih  penget…
4  berarti kakak beruntung dibimbing waktu s tent…

                                              tokens  \
0  [im, a, new, subscriber, ka, thank, you, for, …
1     [rekomendasi, ecourse, machine, learning, kak]
```

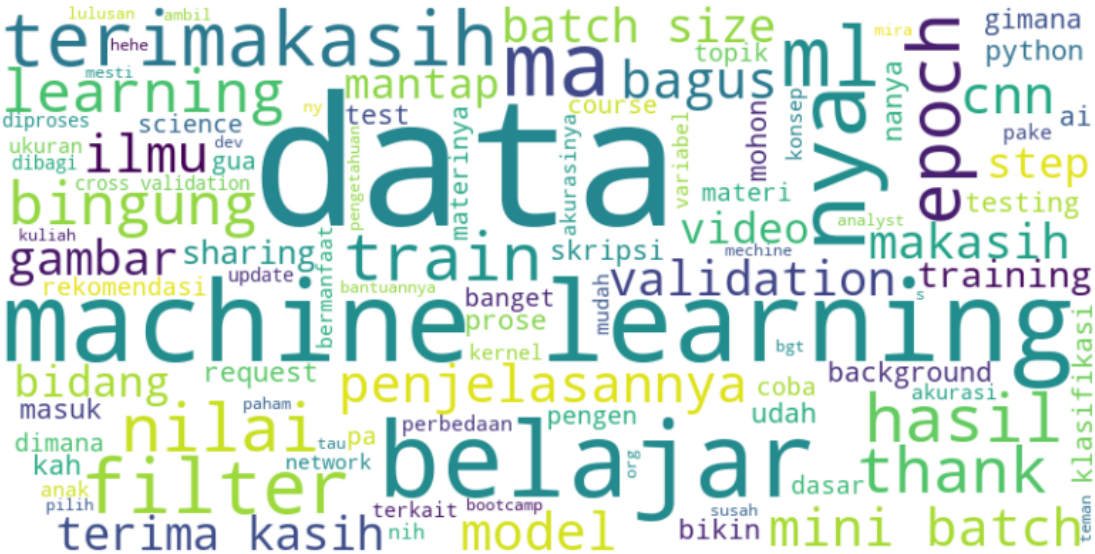
```
2                          [kak, kuliah, s, dimana, tu]
3  [background, it, sampe, pengetahuan, ai, machi…
4  [kakak, beruntung, dibimbing, s, machine, lear…

                                        lemmatized_text
0  [im, a, new, subscriber, ka, thank, you, for, …
1     [rekomendasi, ecourse, machine, learning, kak]
2                          [kak, kuliah, s, dimana, tu]
3  [background, it, sampe, pengetahuan, ai, machi…
4  [kakak, beruntung, dibimbing, s, machine, lear…
```

[39]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Gabungkan semua kata-kata yang sudah melalui proses lemmatization menjadi␣
 ↪satu string
all_words = ' '.join([' '.join(tokens) for tokens in df['lemmatized_text']])

# Buat word cloud
wordcloud = WordCloud(width=800, height=400, background_color='white',␣
 ↪max_words=100).generate(all_words)

# Tampilkan word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```



Dari hasil wordcloud yang didapatkan, ada beberapa stopword yang perlu di drop secara manual seperti "Kak", "bang", "ya", dan lainnya. Kata-kata tersebut berpotensi menjadi "noise" yang mengganggu proses analisa, dikarenakan kurangnya substansi yang dapat didapatkan dari kata-kata tersebut.

[42]:
```
# Tambahkan kata-kata stopword khusus
custom_stopwords = ["kak", "bang", "ya", "yg", "gw", "kakak",␣
 ↪"maaf","ka","gue","kalo","gak","aja","pa","sih","klo","ga","gk"]
stop_words = set(stopwords.words('indonesian')).union(custom_stopwords)

def remove_custom_stopwords(tokens):
    return [word for word in tokens if word not in stop_words]

# Terapkan fungsi ini pada kolom token yang sudah ada
df['tokens'] = df['tokens'].apply(remove_custom_stopwords)
df['lemmatized_text'] = df['tokens'].apply(lemmatize)

# Perbarui word cloud dengan stopwords yang sudah dihapus
all_words = ' '.join([' '.join(tokens) for tokens in df['lemmatized_text']])
wordcloud = WordCloud(width=800, height=400, background_color='white',␣
 ↪max_words=100).generate(all_words)

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```



Pada hasil wordcloud yang sudah menghapus stopword tambahan, lebih banyak insight yang bisa didapatkan, seperti kata "data" yang sangat berkaitan erat dengan

materi machine learning, lalu "belajar" yang dapat diartikan sebagai banyaknya audiens youtube yang tertarik untuk pembelajaran machine learning, hal ini dapat menjadi peluang bisnis yang bagus seperti membuat bootcamp, konten-konten singkat di sosial media, dan bisnis edukasi lainnya.

[43]:
```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

# Ubah kolom 'lemmatized_text' menjadi kalimat kembali
df['lemmatized_text_str'] = df['lemmatized_text'].apply(lambda x: ' '.join(x))

# 1. Count Vectorizer
count_vectorizer = CountVectorizer()
count_vectors = count_vectorizer.fit_transform(df['lemmatized_text_str'])
count_vector_df = pd.DataFrame(count_vectors.toarray(),␣
 ↪columns=count_vectorizer.get_feature_names_out())

# 2. TF-IDF Vectorizer
tfidf_vectorizer = TfidfVectorizer()
tfidf_vectors = tfidf_vectorizer.fit_transform(df['lemmatized_text_str'])
tfidf_vector_df = pd.DataFrame(tfidf_vectors.toarray(),␣
 ↪columns=tfidf_vectorizer.get_feature_names_out())

# Display the first few rows of each vectorization result
print("Count Vectorizer Result:\n", count_vector_df.head())
print("\nTF-IDF Vectorizer Result:\n", tfidf_vector_df.head())
```

```
Count Vectorizer Result:
    abang  acak  adam  ademmm  adu  ai  aiml  ajaib  aktif  akurasi  …  wkwk
\
0      0     0     0       0    0   0     0      0      0        0  …     0
1      0     0     0       0    0   0     0      0      0        0  …     0
2      0     0     0       0    0   0     0      0      0        0  …     0
3      0     0     0       0    0   1     0      0      0        0  …     0
4      0     0     0       0    0   0     0      0      0        0  …     0

   worth  wow  xgboost  xy  yaa  yng  you  yt  zero
0      0    0        0   0    0    0    1   0     0
1      0    0        0   0    0    0    0   0     0
2      0    0        0   0    0    0    0   0     0
3      0    0        0   0    0    0    0   0     0
4      0    0        0   0    0    0    0   0     0

[5 rows x 651 columns]

TF-IDF Vectorizer Result:
    abang  acak  adam  ademmm  adu        ai  aiml  ajaib  aktif  akurasi  …
\
0    0.0   0.0   0.0     0.0  0.0  0.000000   0.0    0.0    0.0      0.0  …
```

```
1    0.0   0.0   0.0     0.0  0.0  0.000000   0.0    0.0    0.0      0.0  …
2    0.0   0.0   0.0     0.0  0.0  0.000000   0.0    0.0    0.0      0.0  …
3    0.0   0.0   0.0     0.0  0.0  0.395625   0.0    0.0    0.0      0.0  …
4    0.0   0.0   0.0     0.0  0.0  0.000000   0.0    0.0    0.0      0.0  …

   wkwk  worth  wow  xgboost   xy  yaa  yng       you   yt  zero
0   0.0    0.0  0.0      0.0  0.0  0.0  0.0  0.166064  0.0   0.0
1   0.0    0.0  0.0      0.0  0.0  0.0  0.0  0.000000  0.0   0.0
2   0.0    0.0  0.0      0.0  0.0  0.0  0.0  0.000000  0.0   0.0
3   0.0    0.0  0.0      0.0  0.0  0.0  0.0  0.000000  0.0   0.0
4   0.0    0.0  0.0      0.0  0.0  0.0  0.0  0.000000  0.0   0.0

[5 rows x 651 columns]
```

Link Video Penjelasan : https://drive.google.com/file/d/1gFALrxrej3SoznL89nv-_OtAramoNFUL/view?usp=sharing