Soal nomor 1, 2502041956 - Nicholas Javier - LA05

Link Video : https://www.youtube.com/watch?v=yF9EqOwTXt0

1. [LO1, LO2, LO3] Hate speech pada media sosial adalah suatu ungkapan yang sangat mengganggu. Identifikasi hate speech secara otomatis akan membantu pihak terkait untuk melakukan suatu tindakan secara efisien. Oleh sebab itu diperlukan bantuan dari anda untuk membangun model klasifikasi multilabel Hate Speech pada data yang diberikan, dengan label Hate Speech (HS), Abusive, HS_Individual, HS_Group, HS_Religion, HS_Race, HS_Physical, HS_Gender, HS_Other.

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MultiLabelBinarizer
import re
import string
import torch
from transformers import DistilBertTokenizer,
DistilBertForSequenceClassification, Trainer, TrainingArguments
!pip install datasets
from datasets import Dataset
```

```
Requirement already satisfied: datasets in
/usr/local/lib/python3.10/dist-packages (2.20.0)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from datasets) (3.15.4)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.10/dist-packages (from datasets) (1.25.2)
Requirement already satisfied: pyarrow>=15.0.0 in
/usr/local/lib/python3.10/dist-packages (from datasets) (16.1.0)
Requirement already satisfied: pyarrow-hotfix in
/usr/local/lib/python3.10/dist-packages (from datasets) (0.6)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in
/usr/local/lib/python3.10/dist-packages (from datasets) (0.3.8)
Requirement already satisfied: pandas in
/usr/local/lib/python3.10/dist-packages (from datasets) (2.0.3)
Requirement already satisfied: requests>=2.32.2 in
/usr/local/lib/python3.10/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in
/usr/local/lib/python3.10/dist-packages (from datasets) (4.66.4)
Requirement already satisfied: xxhash in
```

```
/usr/local/lib/python3.10/dist-packages (from datasets) (3.4.1)
Requirement already satisfied: multiprocess in
/usr/local/lib/python3.10/dist-packages (from datasets) (0.70.16)
Requirement already satisfied: fsspec[http]<=2024.5.0,>=2023.1.0 in
/usr/local/lib/python3.10/dist-packages (from datasets) (2023.6.0)
Requirement already satisfied: aiohttp in
/usr/local/lib/python3.10/dist-packages (from datasets) (3.9.5)
Requirement already satisfied: huggingface-hub>=0.21.2 in
/usr/local/lib/python3.10/dist-packages (from datasets) (0.23.4)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from datasets) (24.1)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.10/dist-packages (from datasets) (6.0.1)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets)
(1.3.1)
Requirement already satisfied: attrs>=17.3.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets)
(23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets)
(1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets)
(6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets)
(1.9.4)
Requirement already satisfied: async-timeout<5.0,>=4.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets)
(4.0.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.21.2-
>datasets) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2-
>datasets) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2-
>datasets) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2-
>datasets) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2-
>datasets) (2024.6.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas->datasets)
(2.8.2)
```

```
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas->datasets)
(2023.4)
Requirement already satisfied: tzdata>=2022.1 in
/usr/local/lib/python3.10/dist-packages (from pandas->datasets)
(2024.1)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2-
>pandas->datasets) (1.16.0)
```

```
df = pd.read_csv('data_1C.csv')
df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 3292,\n  \"fields\":
[\n    {\n      \"column\": \"Unnamed: 0\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 950,\n        \"min\": 0,\n
\"max\": 3291,\n        \"num_unique_values\": 3292,\n
\"samples\": [\n          3098,\n          3161,\n          3165\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"index\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 3805,\n
\"min\": 1,\n        \"max\": 13168,\n        \"num_unique_values\":
3292,\n        \"samples\": [\n          1270,\n          8043,\n
10550\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Tweet\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 3287,\n        \"samples\": [\n          \"Dulu
kita selalu mengucap kata sayang di penghujung malam, kini tak lebih
dari dua orang asing yg terkadang diam-diam merindukan masa lalu'\",\n
\"3. Anak anak muda kreatif Indonesia menyambut gerakan ini dengan
kegiatan Ekspedisi Kapsul Waktu yang baru dibuka 70 tahun
mendatang\",\n          \"USER Mana suaranya cebong.... ??????? \\\\
nKeluar kolam dunk...dunk...dunk \\\\xf0\\\\x9f\\\\x98\\\\x81\\\\
xf0\\\\x9f\\\\x98\\\\x81\\\\xf0\\\\x9f\\\\x98\\\\x81'\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"HS\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Abusive\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"HS_Individual\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"HS_Group\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"HS_Religion\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"HS_Race\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"HS_Physical\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"HS_Gender\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"HS_Other\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3292 entries, 0 to 3291
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Unnamed: 0     3292 non-null   int64 
 1   index          3292 non-null   int64 
 2   Tweet          3292 non-null   object
 3   HS             3292 non-null   int64 
 4   Abusive        3292 non-null   int64 
 5   HS_Individual  3292 non-null   int64 
 6   HS_Group       3292 non-null   int64 
 7   HS_Religion    3292 non-null   int64 
 8   HS_Race        3292 non-null   int64 
 9   HS_Physical    3292 non-null   int64 
 10  HS_Gender      3292 non-null   int64 
 11  HS_Other       3292 non-null   int64
dtypes: int64(11), object(1)
memory usage: 308.8+ KB
```

```
df.describe()
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 8,\n \"fields\": [\n
{\n   \"column\": \"Unnamed: 0\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 1186.2337934135105,\n
\"min\": 0.0,\n   \"max\": 3292.0,\n
\"num_unique_values\": 7,\n   \"samples\": [\n   3292.0,\n
1645.5,\n   2468.25\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"index\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 4187.859035452676,\n   \"min\":
1.0,\n   \"max\": 13168.0,\n   \"num_unique_values\": 8,\n
\"samples\": [\n   6647.773390036452,\n   6745.5,\n
3292.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"HS\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 1163.7503318639951,\n   \"min\": 0.0,\n   \"max\":
3292.0,\n   \"num_unique_values\": 5,\n   \"samples\": [\n
0.4258809234507898,\n   1.0,\n   0.49455096598085396\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"Abusive\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1163.7528216635562,\n   \"min\": 0.0,\n   \"max\": 3292.0,\n
\"num_unique_values\": 5,\n   \"samples\": [\n
0.3845686512758202,\n   1.0,\n   0.4865670763072863\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"HS_Individual\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1163.7601526268763,\n   \"min\": 0.0,\n   \"max\": 3292.0,\n
\"num_unique_values\": 5,\n   \"samples\": [\n
0.27794653705953826,\n   1.0,\n   0.44805495393684475\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"HS_Group\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1163.8218983435818,\n   \"min\": 0.0,\n   \"max\": 3292.0,\n
\"num_unique_values\": 5,\n   \"samples\": [\n
0.14793438639125153,\n   1.0,\n   0.35508886927470645\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"HS_Religion\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1163.8326535001772,\n   \"min\": 0.0,\n   \"max\": 3292.0,\n
\"num_unique_values\": 5,\n   \"samples\": [\n
0.05741190765492102,\n   1.0,\n   0.23266332775451506\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"HS_Race\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":

```
1163.8360497960946,\n        \"min\": 0.0,\n        \"max\": 3292.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n
0.03614823815309842,\n          1.0,\n          0.18668725170262032\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"HS_Physical\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1163.8382363991452,\n        \"min\": 0.0,\n        \"max\": 3292.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n
0.024605103280680437,\n          1.0,\n          0.15494187521750735\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"HS_Gender\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1163.8381121428802,\n        \"min\": 0.0,\n        \"max\": 3292.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n
0.025212636695018227,\n          1.0,\n          0.1567942204818653\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"HS_Other\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1163.7590209973682,\n        \"min\": 0.0,\n        \"max\": 3292.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n
0.293134872417983,\n          1.0,\n          0.4552689102599026\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```

```
df.isnull().sum()
```

```
Unnamed: 0       0
index            0
Tweet            0
HS               0
Abusive          0
HS_Individual    0
HS_Group         0
HS_Religion      0
HS_Race          0
HS_Physical      0
HS_Gender        0
HS_Other         0
dtype: int64
```

A. Bangunlah model klasifikasi dengan menggunakan Large Language Model (LLM), anda dapat menggunakan pretrained model yang akan anda jelaskan pada penjelasan dengan melakukan minimal 2 hyperparameter tuning.

```
def preprocess_text(text):
    text = re.sub(r'http\S+', '', text)
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = text.lower()
    return text
```

```
df['Tweet'] = df['Tweet'].apply(preprocess_text)

labels = ['HS', 'Abusive', 'HS_Individual', 'HS_Group', 'HS_Religion',
'HS_Race', 'HS_Physical', 'HS_Gender', 'HS_Other']
df['labels'] = df[labels].values.tolist()

train_data, test_data = train_test_split(df, test_size=0.2,
random_state=42)
train_data, val_data = train_test_split(train_data, test_size=0.1,
random_state=42)

train_dataset = Dataset.from_pandas(train_data)
val_dataset = Dataset.from_pandas(val_data)
test_dataset = Dataset.from_pandas(test_data)

mlb = MultiLabelBinarizer()
train_labels = mlb.fit_transform(train_data['labels'])
val_labels = mlb.transform(val_data['labels'])
test_labels = mlb.transform(test_data['labels'])

tokenizer = DistilBertTokenizer.from_pretrained('distilbert-base-
uncased')

def tokenize(batch):
    return tokenizer(batch['Tweet'], padding=True, truncation=True)

train_dataset = train_dataset.map(tokenize, batched=True)
val_dataset = val_dataset.map(tokenize, batched=True)
test_dataset = test_dataset.map(tokenize, batched=True)

train_dataset.set_format('torch', columns=['input_ids',
'attention_mask'], output_all_columns=True)
val_dataset.set_format('torch', columns=['input_ids',
'attention_mask'], output_all_columns=True)
test_dataset.set_format('torch', columns=['input_ids',
'attention_mask'], output_all_columns=True)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/
_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your
settings tab (https://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to
access public models or datasets.
  warnings.warn(
```

```
{"model_id":"d5259aabe548411297cb3500d96b65f1","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"31847de2d5c04ef5a3b2ba215bfdbd84","version_major":2,"vers
ion_minor":0}

{"model_id":"3fb7a01387524471b38d09e0673a7dab","version_major":2,"vers
ion_minor":0}
```

Untuk teks dalam dataset df, kode melakukan preprocessing, pembagian data, dan tokenisasi. Pertama, fungsi preprocess_text membersihkan teks dengan menghapus URL, tanda baca, dan mengubah teks menjadi huruf kecil. Fungsi ini kemudian diterapkan pada kolom "Tweet" dalam dataframe. Selanjutnya, kolom yang disebut "label" diisi dengan daftar label yang terdapat di setiap baris. Dengan menggunakan train_test_split, data dibagi menjadi set pelatihan, validasi, dan pengujian. Dataset ini kemudian diformat seperti dataset library. Untuk mengubah label teks menjadi format biner, multilabel binarizer digunakan. Selanjutnya, teks diubah menjadi token yang dapat dimasukkan ke dalam model dengan tokenizer model DistilBert. Tokenisasi dilakukan pada ketiga set data: pelatihan, validasi, dan pengujian. Selanjutnya, set data diformat menggunakan PyTorch dengan kolom output utama input_ids dan attention_mask.

```
!pip install transformers[torch] accelerate -U
```

```
Requirement already satisfied: transformers[torch] in
/usr/local/lib/python3.10/dist-packages (4.42.3)
Requirement already satisfied: accelerate in
/usr/local/lib/python3.10/dist-packages (0.32.1)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from transformers[torch])
(3.15.4)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.2 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch])
(0.23.4)
Requirement already satisfied: numpy<2.0,>=1.17 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch])
(1.25.2)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch])
(24.1)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch])
(6.0.1)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch])
(2024.5.15)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from transformers[torch])
(2.32.3)
Requirement already satisfied: safetensors>=0.4.1 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch])
(0.4.3)
Requirement already satisfied: tokenizers<0.20,>=0.19 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch])
```

```
(0.19.1)
Requirement already satisfied: tqdm>=4.27 in
/usr/local/lib/python3.10/dist-packages (from transformers[torch])
(4.66.4)
Requirement already satisfied: torch in
/usr/local/lib/python3.10/dist-packages (from transformers[torch])
(2.3.0+cu121)
Requirement already satisfied: psutil in
/usr/local/lib/python3.10/dist-packages (from accelerate) (5.9.5)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.23.2->transformers[torch]) (2023.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.23.2->transformers[torch]) (4.12.2)
Requirement already satisfied: sympy in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (1.12.1)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (3.3)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (3.1.4)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (12.1.105)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105
in /usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (12.1.105)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (12.1.105)
Requirement already satisfied: nvidia-cudnn-cu12==8.9.2.26 in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (8.9.2.26)
Requirement already satisfied: nvidia-cublas-cu12==12.1.3.1 in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (12.1.3.1)
Requirement already satisfied: nvidia-cufft-cu12==11.0.2.54 in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (11.0.2.54)
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (10.3.2.106)
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (11.4.5.107)
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in
```

```
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (12.1.0.106)
Requirement already satisfied: nvidia-nccl-cu12==2.20.5 in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (2.20.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (12.1.105)
Requirement already satisfied: triton==2.3.0 in
/usr/local/lib/python3.10/dist-packages (from torch-
>transformers[torch]) (2.3.0)
Requirement already satisfied: nvidia-nvjitlink-cu12 in
/usr/local/lib/python3.10/dist-packages (from nvidia-cusolver-
cu12==11.4.5.107->torch->transformers[torch]) (12.5.82)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests-
>transformers[torch]) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests-
>transformers[torch]) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests-
>transformers[torch]) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests-
>transformers[torch]) (2024.6.2)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch-
>transformers[torch]) (2.1.5)
Requirement already satisfied: mpmath<1.4.0,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy->torch-
>transformers[torch]) (1.3.0)
```

Kode dibawah ini mendefinisikan model klasifikasi sekuens multi-label berbasis DistilBert dan melatihnya menggunakan Trainer transformer library. Pertama, definisi kelas DistilBertForSequenceClassification dengan mewarisi dan mengganti lapisan klasifikasinya. Fungsi maju diubah untuk menangani tugas klasifikasi berbagai label menggunakan BCEWithLogitsLoss. Setelah itu, model diaktifkan dengan memuat parameter pra-terlatih dari distilbert-base-uncased dan mengubah jumlah label sesuai dengan jumlah kelas dalam dataset. Pengaturan direktori keluaran, strategi evaluasi, laju pembelajaran, ukuran batch, jumlah epoch, dan pengurangan berat adalah argumen pelatihan. Model, argumen pelatihan, dataset evaluasi, dan tokenizer kemudian digunakan untuk menginisialisasi instruktur. Terakhir, model dilatih menggunakan metode pelatihan pelatih, yang secara otomatis mengelola proses pelatihan.

```python
from transformers import DistilBertForSequenceClassification, Trainer,
TrainingArguments
import torch.nn as nn

class
```

```
DistilBertForMultiLabelSequenceClassification(DistilBertForSequenceCla
ssification):
    def __init__(self, config):
        super().__init__(config)
        self.classifier = nn.Linear(config.hidden_size,
config.num_labels)

    def forward(self, input_ids=None, attention_mask=None,
labels=None):
        outputs = self.distilbert(input_ids=input_ids,
attention_mask=attention_mask)
        hidden_state = outputs[0]
        pooled_output = hidden_state[:, 0]
        logits = self.classifier(pooled_output)
        loss = None
        if labels is not None:
            loss_fct = nn.BCEWithLogitsLoss()
            loss = loss_fct(logits, labels.float())
        output = (logits,) + outputs[1:]
        return ((loss,) + output) if loss is not None else output

model =
DistilBertForMultiLabelSequenceClassification.from_pretrained('distilb
ert-base-uncased', num_labels=len(labels))

training_args = TrainingArguments(
    output_dir='./results',
    evaluation_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=5,
    weight_decay=0.01,
)


trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    tokenizer=tokenizer,
)

trainer.train()
```

```
Some weights of DistilBertForMultiLabelSequenceClassification were not
initialized from the model checkpoint at distilbert-base-uncased and
are newly initialized: ['classifier.bias', 'classifier.weight',
'pre_classifier.bias', 'pre_classifier.weight']
```

```
You should probably TRAIN this model on a down-stream task to be able
to use it for predictions and inference.
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:
1494: FutureWarning: `evaluation_strategy` is deprecated and will be
removed in version 4.46 of □ Transformers. Use `eval_strategy` instead
  warnings.warn(
```

```
<IPython.core.display.HTML object>
```

```
TrainOutput(global_step=745, training_loss=0.29505649029008496,
metrics={'train_runtime': 169.6674, 'train_samples_per_second':
69.813, 'train_steps_per_second': 4.391, 'total_flos':
422960655897252.0, 'train_loss': 0.29505649029008496, 'epoch': 5.0})
```

```python
from sklearn.metrics import classification_report

train_preds = trainer.predict(train_dataset)
train_pred_labels = (train_preds.predictions > 0).astype(int)

val_preds = trainer.predict(val_dataset)
val_pred_labels = (val_preds.predictions > 0).astype(int)

test_preds = trainer.predict(test_dataset)
test_pred_labels = (test_preds.predictions > 0).astype(int)

train_report = classification_report(train_preds.label_ids,
train_pred_labels, target_names=labels)
val_report = classification_report(val_preds.label_ids,
val_pred_labels, target_names=labels)
test_report = classification_report(test_preds.label_ids,
test_pred_labels, target_names=labels)

print("Training Set Classification Report:")
print(train_report)

print("Validation Set Classification Report:")
print(val_report)

print("Testing Set Classification Report:")
print(test_report)
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
Training Set Classification Report:
              precision    recall  f1-score   support

          HS       0.93      0.91      0.92      1009
     Abusive       0.93      0.83      0.88       920
```

```
HS_Individual       0.78      0.75      0.76       657
     HS_Group       0.73      0.45      0.55       352
  HS_Religion       0.85      0.21      0.33       136
      HS_Race       0.88      0.37      0.52        75
  HS_Physical       0.00      0.00      0.00        59
    HS_Gender       0.00      0.00      0.00        62
     HS_Other       0.81      0.82      0.82       701

    micro avg       0.87      0.75      0.80      3971
    macro avg       0.66      0.48      0.53      3971
 weighted avg       0.84      0.75      0.78      3971
  samples avg       0.45      0.41      0.42      3971

Validation Set Classification Report:
               precision    recall  f1-score   support

           HS       0.78      0.68      0.73       107
      Abusive       0.76      0.73      0.74        97
HS_Individual       0.65      0.49      0.56        63
     HS_Group       0.63      0.27      0.38        44
  HS_Religion       1.00      0.07      0.13        14
      HS_Race       0.67      0.12      0.21        16
  HS_Physical       0.00      0.00      0.00         7
    HS_Gender       0.00      0.00      0.00         5
     HS_Other       0.60      0.50      0.54        68

    micro avg       0.71      0.53      0.61       421
    macro avg       0.56      0.32      0.37       421
 weighted avg       0.69      0.53      0.58       421
  samples avg       0.34      0.29      0.30       421

Testing Set Classification Report:
               precision    recall  f1-score   support

           HS       0.80      0.71      0.75       286
      Abusive       0.80      0.70      0.75       249
HS_Individual       0.71      0.51      0.59       195
     HS_Group       0.62      0.25      0.36        91
  HS_Religion       0.60      0.08      0.14        39
      HS_Race       0.83      0.18      0.29        28
  HS_Physical       0.00      0.00      0.00        15
    HS_Gender       0.00      0.00      0.00        16
     HS_Other       0.72      0.59      0.65       196

    micro avg       0.76      0.56      0.64      1115
    macro avg       0.57      0.33      0.39      1115
 weighted avg       0.73      0.56      0.62      1115
  samples avg       0.39      0.33      0.33      1115
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in samples with no predicted labels. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined
and being set to 0.0 in samples with no true labels. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in samples with no predicted labels. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined
and being set to 0.0 in samples with no true labels. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in samples with no predicted labels. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined
and being set to 0.0 in samples with no true labels. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

Menurut laporan klasifikasi untuk set validasi dan pengujian, model menunjukkan kinerja yang baik pada beberapa kelas, seperti kelas "HS" dan "Abusive", yang memiliki nilai precision, recall, dan f1-score yang relatif tinggi. Namun, kinerja model yang sangat rendah pada kelas lain, seperti "HS_Physical", "HS_Gender", dan "HS_Religion", menunjukkan bahwa model kesulitan Singkatnya, model ini dapat menemukan beberapa jenis ujaran kebencian dan konten pelecehan dengan cukup baik. Ini terutama berlaku untuk kelas dengan data yang lebih kecil.

```
from sklearn.metrics import classification_report, accuracy_score

train_accuracy = accuracy_score(train_preds.label_ids,
train_pred_labels)
val_accuracy = accuracy_score(val_preds.label_ids, val_pred_labels)
test_accuracy = accuracy_score(test_preds.label_ids, test_pred_labels)

print(f"Testing Set Accuracy: {test_accuracy:.2f}")
```

```
Testing Set Accuracy: 0.53
```

Saya melakukan hyperparameter tuning untuk soal training dibawah dengan mengubah epoch, learning rate, dan batch size.

```
training_args = TrainingArguments(
    output_dir='./results',
    evaluation_strategy="epoch",
    learning_rate=3e-5,
    per_device_train_batch_size=32,
    per_device_eval_batch_size=32,
    num_train_epochs=7,
    weight_decay=0.01,
)


trainer_hyper = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    tokenizer=tokenizer,
)

trainer_hyper.train()
```

```
/usr/local/lib/python3.10/dist-packages/transformers/
training_args.py:1494: FutureWarning: `evaluation_strategy` is
deprecated and will be removed in version 4.46 of 🤗 Transformers. Use
`eval_strategy` instead
  warnings.warn(
```

```
<IPython.core.display.HTML object>
```

```
TrainOutput(global_step=525, training_loss=0.138319027310326,
metrics={'train_runtime': 201.6882, 'train_samples_per_second':
```

```
82.221, 'train_steps_per_second': 2.603, 'total_flos':
592147298826864.0, 'train_loss': 0.138319027310326, 'epoch': 7.0})
```

```
trainh_preds = trainer_hyper.predict(train_dataset)
trainh_pred_labels = (trainh_preds.predictions > 0).astype(int)

valh_preds = trainer_hyper.predict(val_dataset)
valh_pred_labels = (valh_preds.predictions > 0).astype(int)

testh_preds = trainer_hyper.predict(test_dataset)
testh_pred_labels = (testh_preds.predictions > 0).astype(int)

trainh_report = classification_report(trainh_preds.label_ids,
trainh_pred_labels, target_names=labels)
valh_report = classification_report(valh_preds.label_ids,
valh_pred_labels, target_names=labels)
testh_report = classification_report(testh_preds.label_ids,
testh_pred_labels, target_names=labels)

print("Training Set Classification Report:")
print(trainh_report)

print("Validation Set Classification Report:")
print(valh_report)

print("Testing Set Classification Report:")
print(testh_report)
```

```
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

Training Set Classification Report:
               precision    recall  f1-score   support

           HS       1.00      0.99      1.00      1009
      Abusive       0.99      0.97      0.98       920
HS_Individual       0.96      0.95      0.95       657
     HS_Group       0.91      0.91      0.91       352
  HS_Religion       0.96      0.87      0.91       136
      HS_Race       0.91      0.77      0.83        75
  HS_Physical       0.89      0.14      0.24        59
    HS_Gender       0.83      0.79      0.81        62
     HS_Other       0.97      0.99      0.98       701

    micro avg       0.97      0.95      0.96      3971
    macro avg       0.93      0.82      0.85      3971
 weighted avg       0.97      0.95      0.95      3971
```

```
  samples avg       0.53      0.52      0.53      3971

Validation Set Classification Report:
               precision    recall  f1-score   support

           HS       0.78      0.74      0.76       107
      Abusive       0.80      0.73      0.76        97
HS_Individual       0.60      0.48      0.53        63
     HS_Group       0.58      0.50      0.54        44
  HS_Religion       0.58      0.50      0.54        14
      HS_Race       0.91      0.62      0.74        16
  HS_Physical       0.00      0.00      0.00         7
    HS_Gender       1.00      0.20      0.33         5
     HS_Other       0.65      0.53      0.59        68

    micro avg       0.72      0.61      0.66       421
    macro avg       0.66      0.48      0.53       421
 weighted avg       0.70      0.61      0.65       421
  samples avg       0.35      0.33      0.32       421

Testing Set Classification Report:
               precision    recall  f1-score   support

           HS       0.80      0.72      0.76       286
      Abusive       0.82      0.75      0.78       249
HS_Individual       0.69      0.54      0.60       195
     HS_Group       0.56      0.46      0.51        91
  HS_Religion       0.63      0.31      0.41        39
      HS_Race       0.76      0.57      0.65        28
  HS_Physical       0.00      0.00      0.00        15
    HS_Gender       0.62      0.31      0.42        16
     HS_Other       0.71      0.61      0.66       196

    micro avg       0.74      0.62      0.68      1115
    macro avg       0.62      0.48      0.53      1115
 weighted avg       0.73      0.62      0.67      1115
  samples avg       0.39      0.36      0.36      1115


/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in samples with no predicted
labels. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined
and being set to 0.0 in samples with no true labels. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
```

```
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in samples with no predicted labels. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined
and being set to 0.0 in samples with no true labels. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in samples with no predicted labels. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined
and being set to 0.0 in samples with no true labels. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```python
from sklearn.metrics import classification_report, accuracy_score

trainh_accuracy = accuracy_score(trainh_preds.label_ids,
trainh_pred_labels)
valh_accuracy = accuracy_score(valh_preds.label_ids, valh_pred_labels)
testh_accuracy = accuracy_score(testh_preds.label_ids,
testh_pred_labels)

print(f"Testing Set Accuracy: {testh_accuracy:.2f}")
```

```
Testing Set Accuracy: 0.57
```

Laporan klasifikasi untuk set pelatihan menunjukkan kinerja yang sangat baik untuk model, dengan nilai precision, recall, dan f1-score yang tinggi untuk hampir semua kelas. Ini menunjukkan bahwa model mampu mengenali dan mengklasifikasikan data pelatihan dengan sangat akurat. Namun, performa model menurun secara signifikan pada set validasi dan pengujian, terutama pada kelas dengan jumlah data yang lebih sedikit seperti "HS_Physical" dan "HS_Gender".

Dibandingkan dengan penelitian sebelumnya, terlihat bahwa model menghadapi masalah yang serupa dalam mengklasifikasikan beberapa kelas yang sama pada set pengujian dan validasi. Misalnya, kelas "HS_Physical" dan "HS_Gender" menunjukkan skor f1 yang sangat rendah atau nol di kedua laporan. Namun, kelas lain, seperti "HS_Religion" dan "HS_Race", menunjukkan peningkatan kinerja pada set pengujian dan data pelatihan, yang menunjukkan bahwa model mungkin membutuhkan penyesuaian tambahan, mungkin dengan menambahkan lebih banyak data untuk kelas-kelas yang lebih kecil