Video Link (dua soal sama):
https://drive.google.com/file/d/1g6zunjKlR_FJnqGw4VWsucqZvNUatZUc/view?usp=sharing

```
import pandas as pd
import numpy as np

df = pd.read_csv("Scraped Articles.csv")

df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 109,\n  \"fields\": [\n    {\n      \"column\": \"Unnamed: 0\",\n      \"properties\": {\n\"dtype\": \"number\",\n        \"std\": 7,\n        \"min\": 0,\n\"max\": 24,\n        \"num_unique_values\": 25,\n        \"samples\": [\n          8,\n          16,\n          0\n        ],\n\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Title\",\n      \"properties\": {\n        \"dtype\": \"string\",\n        \"num_unique_values\": 107,\n\"samples\": [\n          \"Prediksi Final Supercoppa Italiana, Inter Milan lawan AC Milan: Derbi Sengit di Riyadh\",\n\"Pergantian Amorim Kunci MU Taklukkan Southampton!\",\n\"Guardiola Harus Menjilat Ludahnya Sendiri\"\n        ],\n\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Text\",\n      \"properties\": {\n\"dtype\": \"string\",\n        \"num_unique_values\": 103,\n\"samples\": [\n          \"Borneo FC menyudahi kerja sama dengan Pieter Huistra. Hasil buruk yang diraih Pesut Etam menjadi penyebabnya.\\n\\nDi Liga 1, Borneo FC sedang dalam laju buruk. Dalam tiga pertandingan terakhir, tim kebanggaan publik Samarinda itu selalu kalah.\\n\\nLaju buruk Borneo FC dimulai saat takluk dari Persebaya Surabaya pada 20 Desember 2024. Setelah itu, Borneo FC kalah dari Persik Kediri dan Semen Padang.\\n\\nADVERTISEMENT SCROLL TO CONTINUE WITH CONTENT\\n\\nKekalahan dari Persik dan Kabau Sirah ditelan Borneo FC saat berlaga di kandang. Oleh karena itu, manajemen tim langsung memecat pelatih asal Belanda itu.\\n\\n\\\"Hasil evaluasi menyeluruh, Borneo FC dan Pieter Huistra sepakat untuk mengakhiri kerja sama. Hal ini terkait dengan rentetan hasil buruk yang diterima tim dalam beberapa pertandingan terakhir,\\\" kata pernyataan resmi Borneo FC di situs resmi klub.\\n\\nADVERTISEMENT\\n\\n\\\"Selama melatih Borneo FC, Pieter Huistra telah memimpin 72 pertandingan, meraih 38 kemenangan, 15 hasil imbang, dan 19 kekalahan. Kami menghargai kontribusinya, dan keputusan ini diambil dengan profesionalitas demi kepentingan tim,\\\" demikian pernyataan manajemen Borneo FC.\\n\\nDengan rentetan hasil buruk itu, Borneo FC kini menduduki posisi ke-10 di klasemen Liga 1. Mereka mengumpulkan 26 angka hasil 18 kali bermain.\\n\\nUntuk putaran kedua, Borneo FC sudah mendatangkan beberapa pemain. Ricky Cawor dan Kenzo Nambu yang digaet.\\n\\nUntuk menggantikan Huistra, Borneo FC sudah menunjuk pelatih baru. Joaquin Gomes, pelatih asal Spanyol, yang dipilih menjadi juru taktik anyar di
```

sisa Liga 1.\",\n          \"Bola.com, Jakarta - Carlo Ancelotti, pelatih Real Madrid, menyatakan bahwa menjadi seorang Vinicius Junior bukanlah hal yang mudah. Hal ini terutama terlihat setelah Vinicius mendapatkan kartu merah dalam pertandingan tunda jornada 12 La Liga Spanyol melawan Valencia, yang berlangsung pada Sabtu (4/1/2025) dini hari WIB.\\n\\nPemain sayap asal Brasil tersebut diusir dari lapangan setelah terlibat insiden dengan kiper Stole Dimitrievski. Insiden ini terjadi ketika Vinicius mendorong bagian belakang leher Dimitrievski, menambah beban tekanan yang sudah mengelilinginya, baik di dalam maupun di luar lapangan.\\n\\nPertandingan yang digelar di Stadion Mestalla menjadi ajang lain bagi Vinicius untuk menghadapi ejekan dari pendukung tim lawan. Bahkan, ia harus menghadapi sebagian penggemar yang sebelumnya pernah melakukan tindakan rasis terhadapnya dalam pertandingan pada tahun 2022.\\n\\nAdvertisement\\n\\nDalam konferensi pers setelah pertandingan, Carlo Ancelotti menyoroti tantangan besar yang dihadapi Vinicius Junior dalam pertandingan tersebut.\\n\\n\\n\\\"Saya pikir sulit untuk menjadi Vinicius. Saya tidak berada di posisinya, tetapi saya pikir itu sulit. Untuk menghadapi semua yang telah terjadi, hinaan, semuanya, itu tidak sederhana,\\\" ujar pelatih asal Italia tersebut.\\n\\nWalaupun demikian, Ancelotti menambahkan bahwa Vinicius merasa menyesal atas insiden yang menyebabkan kartu merah dan telah meminta maaf kepada tim. Pelatih tersebut juga mengajak semua pihak untuk bergerak maju dan melupakan kejadian tersebut.\",\n          \"Bola.com, Madrid - Real Madrid dilaporkan telah menargetkan tiga kandidat sebagai pengganti potensial untuk Carlo Ancelotti. Salah satu nama yang sering disebut adalah Xabi Alonso.\\n\\nTekanan mengenai masa depan Carlo Ancelotti di Real Madrid semakin meningkat setelah kekalahan telak dari Barcelona di final Piala Super Spanyol, Senin (13/1/2025) dini hari WIB. Los Blancos harus menerima kekalahan dengan skor 2-5.\\n\\nDi samping kekalahan menyakitkan dari Barcelona tersebut, performa Real Madrid memang kurang memuaskan sepanjang awal musim ini. Mereka tertinggal satu poin di belakang pemimpin klasemen La Liga, yaitu Atletico Madrid. Selain itu, El Real hanya berada di posisi ke-20 klasemen Liga Champions, karena baru memenangkan tiga pertandingan sejauh ini.\\n\\nAdvertisement\\n\\nYang paling baru, tentu saja adalah kekalahan telak dari Barcelona tersebut. Seruan untuk menggantikan Carlo Ancelotti mulai terdengar di berbagai kalangan.\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Source\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 2,\n        \"samples\": [\n          \"https://www.liputan6.com\",\n          \"https://sport.detik.com\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"URL\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 5,\n        \"samples\": [\n          \"https://sport.detik.com/sepakbola/liga-indonesia/d-7735069/shin-tae-yong-bangga-bisa-cetak-banyak-sejarah-bersama-indonesia\",\n          \"https://sport.detik.com/sport-lain/d-

7735806/mau-nonton-proliga-2025-di-surabaya-amankan-tiketnya-di-livin-sukha\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Label\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 5,\n        \"samples\": [\n          \"Liga Indonesia\",\n          \"Non sepak bola\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe","variable_name":"df"}

```
df.drop(columns= df.columns[0], inplace=True)
df = df.astype(str)
```

```
df["Text"].apply(len).max()
```

```
4944
```

```
df.loc[59]
```

```
Title     Foto Liga Spanyol Hari Ini - Foto Terbaru Terkini
Text                                                    nan
Source                                https://www.liputan6.com
URL       https://www.liputan6.com/hot/read/5712885/biog...
Label                                             Liga Spanyol
Name: 59, dtype: object
```

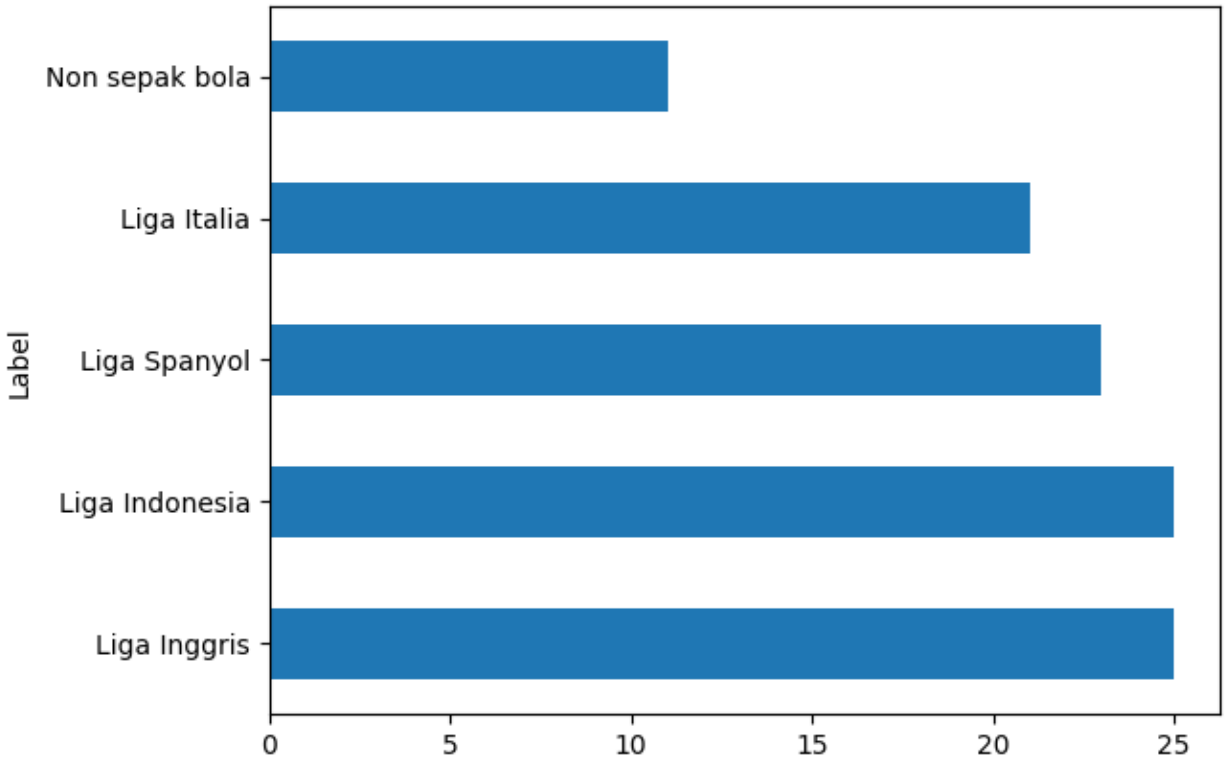
```
df = df[df["Text"] != "nan"]
```

```
len(df)
```

```
105
```

```
df["Label"].value_counts().plot(kind='barh')
```

```
<Axes: ylabel='Label'>
```




```
pip install wordcloud
```

```
Collecting wordcloud
  Downloading wordcloud-1.9.4-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.4 kB)
Requirement already satisfied: numpy>=1.6.1 in
/usr/local/lib/python3.11/dist-packages (from wordcloud) (1.26.4)
Requirement already satisfied: pillow in
/usr/local/lib/python3.11/dist-packages (from wordcloud) (11.1.0)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.11/dist-packages (from wordcloud) (3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)
(1.3.1)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)
(0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)
(4.55.3)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)
(1.4.8)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)
(24.2)
```

```
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)
(3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)
(2.9.0.post0)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7-
>matplotlib->wordcloud) (1.17.0)
Downloading wordcloud-1.9.4-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (547 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/547.9 kB ? eta -:--:--
━━━━━━━ ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 112.6/547.9 kB 3.2 MB/s eta
0:00:01 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 542.7/547.9 kB 9.1
MB/s eta 0:00:01 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 547.9/547.9
kB 7.5 MB/s eta 0:00:00
```

```python
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Create the word cloud
text = " ".join(df["Text"].astype(str))
wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(text)

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

## Preprocessing

### Text cleaning

```
import regex as re
def clean(text):
 return re.sub(r'[^A-Za-z0-9. s]', '', text)
```

### Pipeline

```
pipeline = [str.lower, clean]
def prepare(text, pipeline):
  tokens = text
  for transform in pipeline:
    tokens = transform(tokens)

  return tokens

df['Text_Processed'] = df['Text'].apply(prepare,pipeline=pipeline)
```

# Train Test Split

```
df_copy = df.copy()

df_copy = df_copy[["Text_Processed", "Label"]]

# Encode labels as integers
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df_copy["Label"] = label_encoder.fit_transform(df_copy["Label"])

df_train = df_copy.sample(frac=0.75, random_state=98)
df_test = df_copy.drop(df_train.index)
df_val = df_test.sample(frac = 0.5, random_state = 30)
df_test = df_test.drop(df_val.index)
```

# Model Training

## LLM Task 1 V2

```
import torch
from torch.utils.data import DataLoader
from transformers import BertTokenizer, BertForSequenceClassification,
AdamW, Trainer, TrainingArguments, AutoModelForSequenceClassification
```

## Load pretrained

```
# Load the tokenizer
tokenizer = BertTokenizer.from_pretrained('indolem/indobert-base-uncased')
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: 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":"5383c54a3661403f84fdf02971a59969","version_major":2,"version_minor":0}
```

```
{"model_id":"012f76ee2f3e452cb53b6fcea1b1d040","version_major":2,"version_minor":0}
```

```
{"model_id":"ef04614ef4a84a358aaa213b0cfef067","version_major":2,"version_minor":0}
```

```
{"model_id":"74a99cfec286452f892ad1e1f1066fcc","version_major":2,"version_minor":0}
```

```
{"model_id":"e3e5f5a3c46d497e84af5aa48ac9c47a","version_major":2,"version_minor":0}
```


## Tokenizer

```
from transformers import AutoTokenizer
from datasets import Dataset, DatasetDict

tokenizer = AutoTokenizer.from_pretrained("indolem/indobert-base-uncased")
dataset_train = Dataset.from_pandas(df_train)
dataset_test = Dataset.from_pandas(df_test)
dataset_val = Dataset.from_pandas(df_val)

# RENAME COLUMN TO 'labels'
dataset_train = dataset_train.rename_column("Label", "labels")
dataset_test = dataset_test.rename_column("Label", "labels")
dataset_val = dataset_val.rename_column("Label", "labels")
MAX_LEN = 500
def tokenize_function(examples):
    return tokenizer(examples["Text_Processed"], padding="max_length", truncation=True, max_length = MAX_LEN)
```

```
small_train_dataset = dataset_train.map(tokenize_function,
batched=True)
small_test_dataset = dataset_test.map(tokenize_function, batched=True)
small_val_dataset = dataset_val.map(tokenize_function, batched=True)
```

{"model_id":"4fcead83b06442b9b021eb52e05dfa0b","version_major":2,"version_minor":0}

{"model_id":"dc273324f3dc49babec509b898861e51","version_major":2,"version_minor":0}

{"model_id":"2b47f490c91d4f16b0fd6f5a14cc6ca8","version_major":2,"version_minor":0}

## Create Evaluation Function

```
import numpy as np
import evaluate

metric = evaluate.load("accuracy")
def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions = np.argmax(logits, axis=-1)
    return metric.compute(predictions=predictions, references=labels)
```

{"model_id":"ee9cb4c0f54e4beda6293b738be1f35d","version_major":2,"version_minor":0}

## Hyperparameter Tuning

```
# pip install optuna

import optuna
from transformers import Trainer, TrainingArguments
from transformers import BertForSequenceClassification, BertTokenizer


def objective(trial):

    learning_rate = trial.suggest_loguniform('learning_rate', 1e-5,
1e-3)
    warmup_steps = trial.suggest_categorical('warmup_steps', [0, 3,
5])

    model =
AutoModelForSequenceClassification.from_pretrained('indolem/indobert-
base-uncased', num_labels=5)
```

```
    training_args = TrainingArguments(
        output_dir="./res",
        warmup_steps = warmup_steps,
        learning_rate= learning_rate,
        evaluation_strategy="epoch",
        num_train_epochs = 5,
        logging_dir = './logs'
    )
    trainer = Trainer(
        model=model,
        args=training_args,
        train_dataset=small_train_dataset,
        eval_dataset=small_val_dataset,
        compute_metrics=compute_metrics,
    )
    trainer.train()
    result = trainer.evaluate()

    return result['eval_accuracy']

study = optuna.create_study(direction="maximize")  # We want to
maximize accuracy
study.optimize(objective, n_trials=5)  # Number of trials to run

print(f"Best trial: {study.best_trial}")
```

```
[I 2025-01-24 07:00:53,088] A new study created in memory with name:
no-name-202b2123-f52a-4500-9876-4e1e90c72f3c
<ipython-input-21-da8533bc3e41>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
```


```
{"model_id":"13aad3be133b49b7a09e0d7287e1cfd7","version_major":2,"vers
ion_minor":0}
```


```
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: FutureWarning: `evaluation_strategy` is deprecated and will be
removed in version 4.46 of  Transformers. Use `eval_strategy` instead
  warnings.warn(
wandb: WARNING The `run_name` is currently set to the same value as
`TrainingArguments.output_dir`. If this was not intended, please
```

```
specify a different run name by setting the
`TrainingArguments.run_name` parameter.

<IPython.core.display.Javascript object>

wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
wandb: Using wandb-core as the SDK backend.  Please refer to
https://wandb.me/wandb-core for more information.

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

[I 2025-01-24 07:02:15,315] Trial 0 finished with value:
0.07692307692307693 and parameters: {'learning_rate':
0.000972747902993506, 'warmup_steps': 5}. Best is trial 0 with value:
0.07692307692307693.
<ipython-input-21-da8533bc3e41>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:03:03,250] Trial 1 finished with value:
0.8461538461538461 and parameters: {'learning_rate':
9.415159868379277e-05, 'warmup_steps': 3}. Best is trial 1 with value:
0.8461538461538461.
<ipython-input-21-da8533bc3e41>:8: FutureWarning: suggest_loguniform
```

```
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:03:48,910] Trial 2 finished with value:
0.8461538461538461 and parameters: {'learning_rate':
0.00015987232175668372, 'warmup_steps': 3}. Best is trial 1 with
value: 0.8461538461538461.
<ipython-input-21-da8533bc3e41>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:04:41,109] Trial 3 finished with value:
0.8461538461538461 and parameters: {'learning_rate':
0.0002503455336046591, 'warmup_steps': 3}. Best is trial 1 with value:
0.8461538461538461.
<ipython-input-21-da8533bc3e41>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
```

```
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:05:35,730] Trial 4 finished with value:
0.07692307692307693 and parameters: {'learning_rate':
0.0003917406178694047, 'warmup_steps': 3}. Best is trial 1 with value:
0.8461538461538461.

Best trial: FrozenTrial(number=1, state=1,
values=[0.8461538461538461], datetime_start=datetime.datetime(2025, 1,
24, 7, 2, 15, 318265), datetime_complete=datetime.datetime(2025, 1,
24, 7, 3, 3, 249826), params={'learning_rate': 9.415159868379277e-05,
'warmup_steps': 3}, user_attrs={}, system_attrs={},
intermediate_values={}, distributions={'learning_rate':
FloatDistribution(high=0.001, log=True, low=1e-05, step=None),
'warmup_steps': CategoricalDistribution(choices=(0, 3, 5))},
trial_id=1, value=None)
```

## Training

```
from transformers import TrainingArguments, Trainer

# Best Hyperparameters : {'learning_rate': 9.415159868379277e-05,
'warmup_steps': 3}
model =
AutoModelForSequenceClassification.from_pretrained('indolem/indobert-
base-uncased', num_labels=5)

training_args = TrainingArguments(
        output_dir="./results",
        warmup_steps = 3,
        learning_rate= 9.415159868379277e-05,
        evaluation_strategy="epoch",
        num_train_epochs = 6
    )
trainer = Trainer(
    model=model,
```

```
    args=training_args,
    train_dataset=small_train_dataset,
    eval_dataset=small_val_dataset,
    compute_metrics=compute_metrics,
)
trainer.train()
```

```
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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=60, training_loss=0.40739790598551434,
metrics={'train_runtime': 67.7936, 'train_samples_per_second': 6.992,
'train_steps_per_second': 0.885, 'total_flos': 1217949214140000.0,
'train_loss': 0.40739790598551434, 'epoch': 6.0})
```

## Results

```
from sklearn.metrics import classification_report
LLM_task1_prediction = trainer.predict(small_test_dataset)

print(classification_report(df_test["Label"],
LLM_task1_prediction.predictions.argmax(axis=1)))
```

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

```
              precision    recall  f1-score   support

           0       0.50      1.00      0.67         1
           1       1.00      1.00      1.00         3
           2       0.75      0.75      0.75         4
           3       0.67      0.67      0.67         3
           4       1.00      0.50      0.67         2

    accuracy                           0.77        13
   macro avg       0.78      0.78      0.75        13
weighted avg       0.81      0.77      0.77        13
```

# LLM Task 2

## Tokenizer

```
# pip install datasets

# !pip install --upgrade evaluate

from transformers import AutoTokenizer
from datasets import Dataset, DatasetDict

# LOAD TOKENIZER
tokenizer = AutoTokenizer.from_pretrained("indolem/indobert-base-
uncased")

MAX_LEN = 500
def tokenize_function(examples):
    return tokenizer(examples["Text_Processed"], padding="max_length",
truncation=True, max_length = MAX_LEN)
```

## Evaluation Function

```
import numpy as np
import evaluate

metric = evaluate.load("accuracy")
def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions = np.argmax(logits, axis=-1)
    return metric.compute(predictions=predictions, references=labels)
```

## Sepak Bola vs Non Sepak Bola

### Pretrained model

```
import torch
from torch.utils.data import DataLoader
from transformers import BertTokenizer, BertForSequenceClassification,
AdamW, Trainer, TrainingArguments, AutoModelForSequenceClassification
```

### Transform Labels

```
from datasets import Dataset, DatasetDict

# SPLIT TO 'non sepak bola' AND 'sepak bola'
df_train_binary = df_train.copy()
df_test_binary = df_test.copy()
df_val_binary = df_val.copy()


# Non sepak bola is 4 after Label Encoded
```

```
df_train_binary["Label"] = df_train_binary["Label"].apply(lambda x:
"Non sepak bola" if x == 4 else "Sepak bola")
df_train_binary = df_train_binary[["Text_Processed", "Label"]]
df_test_binary["Label"] = df_test_binary["Label"].apply(lambda x: "Non
sepak bola" if x == 4 else "Sepak bola")
df_test_binary = df_test_binary[["Text_Processed", "Label"]]
df_val_binary["Label"] = df_val_binary["Label"].apply(lambda x: "Non
sepak bola" if x == 4 else "Sepak bola")
df_val_binary = df_val_binary[["Text_Processed", "Label"]]

# 1 = sepak bola
# 0 = non sepak bola

# Encode labels as integers
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df_train_binary["Label"] =
label_encoder.fit_transform(df_train_binary["Label"])
df_test_binary["Label"] =
label_encoder.transform(df_test_binary["Label"])
df_val_binary["Label"] =
label_encoder.transform(df_val_binary["Label"])

# CONVERT TO DATASET OBJECTS
dataset_train_binary = Dataset.from_pandas(df_train_binary)
dataset_test_binary = Dataset.from_pandas(df_test_binary)
dataset_val_binary = Dataset.from_pandas(df_val_binary)

# RENAME COLUMN TO 'labels'
dataset_train_binary = dataset_train_binary.rename_column("Label",
"labels")
dataset_test_binary = dataset_test_binary.rename_column("Label",
"labels")
dataset_val_binary = dataset_val_binary.rename_column("Label",
"labels")
```

Tokenize

```
tokenized_train_dataset_binary =
dataset_train_binary.map(tokenize_function, batched=True)
tokenized_test_dataset_binary =
dataset_test_binary.map(tokenize_function, batched=True)
tokenized_val_dataset_binary =
dataset_val_binary.map(tokenize_function, batched=True)
```


```
{"model_id":"a2ab255952e146a8b80c386fe3ab3e42","version_major":2,"vers
ion_minor":0}

{"model_id":"38a7e5d2617843f2b4993f40e5bffea6","version_major":2,"vers
ion_minor":0}
```

{"model_id":"a713ad56867d4490bda6eacdf868b11c","version_major":2,"version_minor":0}

Hyperparameter Tuning

```
import optuna
from transformers import Trainer, TrainingArguments
from transformers import BertForSequenceClassification, BertTokenizer


def objective(trial):

    learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-3)
    warmup_steps = trial.suggest_categorical('warmup_steps', [0, 3, 5])

    model = AutoModelForSequenceClassification.from_pretrained('indolem/indobert-base-uncased', num_labels=2)

    training_args = TrainingArguments(
        output_dir="./results",
        warmup_steps = warmup_steps,
        learning_rate=learning_rate,
        evaluation_strategy="epoch",
        num_train_epochs = 5
    )
    trainer = Trainer(
        model=model,
        args=training_args,
        train_dataset=tokenized_train_dataset_binary,
        eval_dataset=tokenized_val_dataset_binary,
        compute_metrics=compute_metrics,
    )
    trainer.train()

    result = trainer.evaluate()
    return result['eval_accuracy']

study = optuna.create_study(direction="maximize")  # We want to maximize accuracy
study.optimize(objective, n_trials=5)  # Number of trials to run

print(f"Best trial: {study.best_trial}")
```

```
[I 2025-01-24 07:14:41,432] A new study created in memory with name: no-name-80c27805-cfc6-4ba2-b29f-df33e73d2b5c
<ipython-input-31-4012c63b4548>:8: FutureWarning: suggest_loguniform has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
```

```
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:15:29,753] Trial 0 finished with value:
0.7692307692307693 and parameters: {'learning_rate':
0.0005898049478958887, 'warmup_steps': 3}. Best is trial 0 with value:
0.7692307692307693.
<ipython-input-31-4012c63b4548>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:16:34,663] Trial 1 finished with value:
0.7692307692307693 and parameters: {'learning_rate':
0.000799423992358785, 'warmup_steps': 3}. Best is trial 0 with value:
0.7692307692307693.
<ipython-input-31-4012c63b4548>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
```

```
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:17:39,516] Trial 2 finished with value:
0.9230769230769231 and parameters: {'learning_rate':
0.0004008705752504907, 'warmup_steps': 3}. Best is trial 2 with value:
0.9230769230769231.
<ipython-input-31-4012c63b4548>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:19:08,140] Trial 3 finished with value:
0.7692307692307693 and parameters: {'learning_rate':
1.3982907602859736e-05, 'warmup_steps': 0}. Best is trial 2 with
value: 0.9230769230769231.
<ipython-input-31-4012c63b4548>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
```

```
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:20:03,424] Trial 4 finished with value:
0.7692307692307693 and parameters: {'learning_rate':
2.9076323834869918e-05, 'warmup_steps': 3}. Best is trial 2 with
value: 0.9230769230769231.

Best trial: FrozenTrial(number=2, state=1,
values=[0.9230769230769231], datetime_start=datetime.datetime(2025, 1,
24, 7, 16, 34, 666472), datetime_complete=datetime.datetime(2025, 1,
24, 7, 17, 39, 515936), params={'learning_rate':
0.0004008705752504907, 'warmup_steps': 3}, user_attrs={},
system_attrs={}, intermediate_values={},
distributions={'learning_rate': FloatDistribution(high=0.001,
log=True, low=1e-05, step=None), 'warmup_steps':
CategoricalDistribution(choices=(0, 3, 5))}, trial_id=2, value=None)
```

## Training

Turns out using the default learning rate with 0 warm up steps gives the best results

learning_rate = 0.00005

```
from transformers import TrainingArguments, Trainer

model =
AutoModelForSequenceClassification.from_pretrained('indolem/indobert-
base-uncased', num_labels=2)

training_args = TrainingArguments(output_dir="test_trainer",
eval_strategy="epoch", num_train_epochs = 6)

binary_trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_train_dataset_binary,
    eval_dataset=tokenized_val_dataset_binary,
    compute_metrics=compute_metrics,
)

binary_trainer.train()
```

```
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able
to use it for predictions and inference.

<IPython.core.display.HTML object>

TrainOutput(global_step=60, training_loss=0.094495956103007,
metrics={'train_runtime': 62.902, 'train_samples_per_second': 7.536,
'train_steps_per_second': 0.954, 'total_flos': 121791640860000.0,
'train_loss': 0.094495956103007, 'epoch': 6.0})
```

## Sepak Bola Subcategory

### Transform Labels

```
# SPLIT WITHIN SEPAKBOLA CATEGORIES
df_train_multiple = df_train.copy()
df_test_multiple = df_test.copy()
df_val_multiple = df_val.copy()

df_train_multiple = df_train_multiple[df_train_multiple["Label"] != 4]
[["Text_Processed", "Label"]]
df_test_multiple = df_test_multiple[df_test_multiple["Label"] != 4]
[["Text_Processed", "Label"]]
df_val_multiple = df_val_multiple[df_val_multiple["Label"] != 4]
[["Text_Processed", "Label"]]

# CONVERT TO DATASET OBJECTS
dataset_train_multiple = Dataset.from_pandas(df_train_multiple)
dataset_test_multiple = Dataset.from_pandas(df_test_multiple)
dataset_val_multiple = Dataset.from_pandas(df_val_multiple)

# RENAME COLUMN TO 'labels'
dataset_train_multiple = dataset_train_multiple.rename_column("Label",
"labels")
dataset_test_multiple = dataset_test_multiple.rename_column("Label",
"labels")
dataset_val_multiple = dataset_val_multiple.rename_column("Label",
"labels")
```

### Tokenize

```
tokenized_train_dataset_multiple =
dataset_train_multiple.map(tokenize_function, batched=True)
tokenized_test_dataset_multiple =
dataset_test_multiple.map(tokenize_function, batched=True)
tokenized_val_dataset_multiple =
dataset_val_multiple.map(tokenize_function, batched=True)
```

```
{"model_id":"b4e082fcad5d40178b87073d25d24f88","version_major":2,"version_minor":0}

{"model_id":"6a43a719b70b463a9979b3582094f825","version_major":2,"version_minor":0}

{"model_id":"cdce21ada8b1431cad8d1827b320650c","version_major":2,"version_minor":0}
```


Hyperparameter Tuning

```
import optuna
from transformers import Trainer, TrainingArguments
from transformers import BertForSequenceClassification, BertTokenizer


def objective(trial):

    learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-3)
    warmup_steps = trial.suggest_categorical('warmup_steps', [0, 3, 5])

    model = AutoModelForSequenceClassification.from_pretrained('indolem/indobert-base-uncased', num_labels=4)

    training_args = TrainingArguments(
        output_dir="./results",
        warmup_steps = warmup_steps,
        learning_rate=learning_rate,
        evaluation_strategy="epoch",
        num_train_epochs = 5
    )
    trainer = Trainer(
        model=model,
        args=training_args,
        train_dataset=tokenized_train_dataset_multiple,
        eval_dataset=tokenized_val_dataset_multiple,
        compute_metrics=compute_metrics,
    )
    trainer.train()

    result = trainer.evaluate()
    return result['eval_accuracy']

study = optuna.create_study(direction="maximize")  # We want to maximize accuracy
study.optimize(objective, n_trials=5)  # Number of trials to run

print(f"Best trial: {study.best_trial}")
```

```
[I 2025-01-24 07:21:38,148] A new study created in memory with name:
no-name-d1c218b0-d9e1-4d19-8aef-7b41aeef735d
<ipython-input-35-ef003e458814>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:22:31,872] Trial 0 finished with value: 0.6 and
parameters: {'learning_rate': 2.7030599864663558e-05, 'warmup_steps':
5}. Best is trial 0 with value: 0.6.
<ipython-input-35-ef003e458814>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:23:35,655] Trial 1 finished with value: 0.9 and
parameters: {'learning_rate': 0.0005243212796975114, 'warmup_steps':
5}. Best is trial 1 with value: 0.9.
<ipython-input-35-ef003e458814>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
```

```
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:24:36,181] Trial 2 finished with value: 0.9 and
parameters: {'learning_rate': 0.00011876878238608558, 'warmup_steps':
0}. Best is trial 1 with value: 0.9.
<ipython-input-35-ef003e458814>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:25:31,846] Trial 3 finished with value: 0.9 and
parameters: {'learning_rate': 0.00011883868784746048, 'warmup_steps':
3}. Best is trial 1 with value: 0.9.
<ipython-input-35-ef003e458814>:8: FutureWarning: suggest_loguniform
has been deprecated in v3.0.0. This feature will be removed in v6.0.0.
See https://github.com/optuna/optuna/releases/tag/v3.0.0. Use
suggest_float(..., log=True) instead.
  learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-
3)
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
```

```
newly initialized: ['classifier.bias', '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/python3.11/dist-packages/transformers/training_args.py:
1575: 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>

<IPython.core.display.HTML object>

[I 2025-01-24 07:26:34,491] Trial 4 finished with value: 0.4 and
parameters: {'learning_rate': 0.00036501670937317903, 'warmup_steps':
3}. Best is trial 1 with value: 0.9.

Best trial: FrozenTrial(number=1, state=1, values=[0.9],
datetime_start=datetime.datetime(2025, 1, 24, 7, 22, 31, 874294),
datetime_complete=datetime.datetime(2025, 1, 24, 7, 23, 35, 655283),
params={'learning_rate': 0.0005243212796975114, 'warmup_steps': 5},
user_attrs={}, system_attrs={}, intermediate_values={},
distributions={'learning_rate': FloatDistribution(high=0.001,
log=True, low=1e-05, step=None), 'warmup_steps':
CategoricalDistribution(choices=(0, 3, 5))}, trial_id=1, value=None)
```

Training

```
# A lot of ties but this one converged faster
# best hyperparameters : {'learning_rate': 0.00011883868784746048,
'warmup_steps': 3}

from transformers import TrainingArguments, Trainer


model_cat =
AutoModelForSequenceClassification.from_pretrained('indolem/indobert-
base-uncased', num_labels=4)


training_args = TrainingArguments(
    output_dir="test_trainer",
    learning_rate = 0.0005243212796975114,
    warmup_steps = 5,
    eval_strategy="epoch",
    num_train_epochs = 6
)

sepak_bola_trainer = Trainer(
    model=model_cat,
    args=training_args,
    train_dataset=tokenized_train_dataset_multiple,
```

```
    eval_dataset=tokenized_val_dataset_multiple,
    compute_metrics=compute_metrics,
)

sepak_bola_trainer.train()
```

```
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at indolem/indobert-base-uncased and are
newly initialized: ['classifier.bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able
to use it for predictions and inference.
```

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

```
TrainOutput(global_step=60, training_loss=0.8304917653401692,
metrics={'train_runtime': 47.4819, 'train_samples_per_second': 9.225,
'train_steps_per_second': 1.264, 'total_flos': 112543663752000.0,
'train_loss': 0.8304917653401692, 'epoch': 6.0})
```

## Results

```
from sklearn.metrics import classification_report

# USE THE FIRST MODEL TO GET BINARY PREDICTIONS
binary_prediction =
binary_trainer.predict(tokenized_test_dataset_binary).predictions.argm
ax(axis=1)

# CREATE A DATASET OBJECT USING DF_TEST AND THE BINARY PREDICTIONS
df_binary_prediction =
pd.concat([df_test["Text_Processed"].reset_index(),
pd.DataFrame(binary_prediction, columns = ["labels"])], axis = 1)

# CREATE FINAL PREDICTION DATASET
dataset_FINAL = df_binary_prediction.copy()

# TAKE SEPAKBOLA CATEGORIES PREDICTION
# Sepakbola = 1
# Non Sepakbola = 0

dataset_binary_prediction = Dataset.from_pandas(df_binary_prediction)
dataset_binary_prediction = dataset_binary_prediction.filter(lambda
example: example["labels"] == 1)

# TOKENIZE
tokenized_binary_prediction =
dataset_binary_prediction.map(tokenize_function, batched=True)

# PREDICT USING THE SUBCATOGERIZER
subcategories_prediction =
sepak_bola_trainer.predict(tokenized_binary_prediction).predictions.ar
```

```
gmax(axis = 1)

# REPLACE THE "Non Sepakbola" CLASS WITH ITS ORIGINAL LABEL (4)
dataset_FINAL["labels"] = dataset_FINAL["labels"].apply(lambda x: 4 if
x == 0 else x)

# Fill in the final dataset
for i in range(len(subcategories_prediction)):
    if dataset_FINAL.iloc[i]["labels"] == 1:
      dataset_FINAL.loc[i,"labels"] = subcategories_prediction[i]


# GET CLASSIFICATION REPORT

print(classification_report(df_test['Label'],
dataset_FINAL['labels']))
```

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


```
{"model_id":"da1a1e2581db45eca0a6276e01614e77","version_major":2,"vers
ion_minor":0}
```



```
{"model_id":"ce56ce86c8004d099728eb82d20486e5","version_major":2,"vers
ion_minor":0}
```


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

```
              precision    recall  f1-score   support

           0       0.50      1.00      0.67         1
           1       0.75      1.00      0.86         3
           2       0.75      0.75      0.75         4
           3       1.00      0.67      0.80         3
           4       1.00      0.50      0.67         2

    accuracy                           0.77        13
   macro avg       0.80      0.78      0.75        13
weighted avg       0.83      0.77      0.77        13
```

```
from sklearn.metrics import classification_report
LLM_task1_prediction = trainer.predict(small_test_dataset)

print(classification_report(df_test["Label"],
LLM_task1_prediction.predictions.argmax(axis=1)))
```

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

```
              precision    recall  f1-score   support

           0       0.50      1.00      0.67         1
           1       1.00      1.00      1.00         3
```

```
           2       0.75      0.75      0.75         4
           3       0.67      0.67      0.67         3
           4       1.00      0.50      0.67         2

    accuracy                           0.77        13
   macro avg       0.78      0.78      0.75        13
weighted avg       0.81      0.77      0.77        13
```

## Analysis

Both approaches have very similar performances with only small variances in the Sepak Bola classes.

The effects of using the method of doing a binary classification before a multilabel isn't apparent. This might be because of a lack of representation of the Non Sepak Bola class in the testing dataset (only two entries). So even if the second model is slightly better at recognizing Sepak Bola and Non Sepak Bola categories, it might not reflect this in the results.

```
list(df_test["Label"])
```

```
[1, 1, 1, 0, 3, 3, 3, 2, 2, 2, 2, 4, 4]
```

```
list(LLM_task1_prediction.predictions.argmax(axis=1))
```

```
[1, 1, 1, 0, 2, 3, 3, 3, 2, 2, 2, 4, 0]
```

```
list( dataset_FINAL['labels'])
```

```
[1, 1, 1, 0, 2, 3, 3, 0, 2, 2, 2, 4, 1]
```