```
pip install torchtext==0.15.2

Collecting torchtext==0.15.2
  Downloading torchtext-0.15.2-cp310-cp310-
manylinux1_x86_64.whl.metadata (7.4 kB)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (from torchtext==0.15.2) (4.66.6)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from torchtext==0.15.2)
(2.32.3)
Collecting torch==2.0.1 (from torchtext==0.15.2)
  Downloading torch-2.0.1-cp310-cp310-manylinux1_x86_64.whl.metadata
(24 kB)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from torchtext==0.15.2)
(1.26.4)
Collecting torchdata==0.6.1 (from torchtext==0.15.2)
  Downloading torchdata-0.6.1-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (13 kB)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch==2.0.1-
>torchtext==0.15.2) (3.16.1)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.10/dist-packages (from torch==2.0.1-
>torchtext==0.15.2) (4.12.2)
Requirement already satisfied: sympy in
/usr/local/lib/python3.10/dist-packages (from torch==2.0.1-
>torchtext==0.15.2) (1.13.1)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch==2.0.1-
>torchtext==0.15.2) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch==2.0.1-
>torchtext==0.15.2) (3.1.4)
Collecting nvidia-cuda-nvrtc-cu11==11.7.99 (from torch==2.0.1-
>torchtext==0.15.2)
  Downloading nvidia_cuda_nvrtc_cu11-11.7.99-2-py3-none-
manylinux1_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-runtime-cu11==11.7.99 (from torch==2.0.1-
>torchtext==0.15.2)
  Downloading nvidia_cuda_runtime_cu11-11.7.99-py3-none-
manylinux1_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cuda-cupti-cu11==11.7.101 (from torch==2.0.1-
>torchtext==0.15.2)
  Downloading nvidia_cuda_cupti_cu11-11.7.101-py3-none-
manylinux1_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cudnn-cu11==8.5.0.96 (from torch==2.0.1-
>torchtext==0.15.2)
  Downloading nvidia_cudnn_cu11-8.5.0.96-2-py3-none-
manylinux1_x86_64.whl.metadata (1.6 kB)
```

```
Collecting nvidia-cublas-cu11==11.10.3.66 (from torch==2.0.1-
>torchtext==0.15.2)
  Downloading nvidia_cublas_cu11-11.10.3.66-py3-none-
manylinux1_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cufft-cu11==10.9.0.58 (from torch==2.0.1-
>torchtext==0.15.2)
  Downloading nvidia_cufft_cu11-10.9.0.58-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-curand-cu11==10.2.10.91 (from torch==2.0.1-
>torchtext==0.15.2)
  Downloading nvidia_curand_cu11-10.2.10.91-py3-none-
manylinux1_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cusolver-cu11==11.4.0.1 (from torch==2.0.1-
>torchtext==0.15.2)
  Downloading nvidia_cusolver_cu11-11.4.0.1-2-py3-none-
manylinux1_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cusparse-cu11==11.7.4.91 (from torch==2.0.1-
>torchtext==0.15.2)
  Downloading nvidia_cusparse_cu11-11.7.4.91-py3-none-
manylinux1_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-nccl-cu11==2.14.3 (from torch==2.0.1-
>torchtext==0.15.2)
  Downloading nvidia_nccl_cu11-2.14.3-py3-none-
manylinux1_x86_64.whl.metadata (1.8 kB)
Collecting nvidia-nvtx-cu11==11.7.91 (from torch==2.0.1-
>torchtext==0.15.2)
  Downloading nvidia_nvtx_cu11-11.7.91-py3-none-
manylinux1_x86_64.whl.metadata (1.7 kB)
Collecting triton==2.0.0 (from torch==2.0.1->torchtext==0.15.2)
  Downloading triton-2.0.0-1-cp310-cp310-
manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (1.0 kB)
Requirement already satisfied: urllib3>=1.25 in
/usr/local/lib/python3.10/dist-packages (from torchdata==0.6.1-
>torchtext==0.15.2) (2.2.3)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from nvidia-cublas-
cu11==11.10.3.66->torch==2.0.1->torchtext==0.15.2) (75.1.0)
Requirement already satisfied: wheel in
/usr/local/lib/python3.10/dist-packages (from nvidia-cublas-
cu11==11.10.3.66->torch==2.0.1->torchtext==0.15.2) (0.45.1)
Requirement already satisfied: cmake in
/usr/local/lib/python3.10/dist-packages (from triton==2.0.0-
>torch==2.0.1->torchtext==0.15.2) (3.30.5)
Collecting lit (from triton==2.0.0->torch==2.0.1->torchtext==0.15.2)
  Downloading lit-18.1.8-py3-none-any.whl.metadata (2.5 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests-
>torchtext==0.15.2) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
```

```
/usr/local/lib/python3.10/dist-packages (from requests-
>torchtext==0.15.2) (3.10)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests-
>torchtext==0.15.2) (2024.8.30)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch==2.0.1-
>torchtext==0.15.2) (3.0.2)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy->torch==2.0.1-
>torchtext==0.15.2) (1.3.0)
Downloading torchtext-0.15.2-cp310-cp310-manylinux1_x86_64.whl (2.0
MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 26.3 MB/s eta
0:00:00
anylinux1_x86_64.whl (619.9 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 619.9/619.9 MB 2.9 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.6/4.6 MB 63.5 MB/s eta
0:00:00
anylinux1_x86_64.whl (317.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 317.1/317.1 MB 4.8 MB/s eta
0:00:00
anylinux1_x86_64.whl (11.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.8/11.8 MB 66.5 MB/s eta
0:00:00
anylinux1_x86_64.whl (21.0 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.0/21.0 MB 38.8 MB/s eta
0:00:00
e_cu11-11.7.99-py3-none-manylinux1_x86_64.whl (849 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 849.3/849.3 kB 31.3 MB/s eta
0:00:00
anylinux1_x86_64.whl (557.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 557.1/557.1 MB 1.6 MB/s eta
0:00:00
anylinux2014_x86_64.whl (168.4 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 168.4/168.4 MB 6.7 MB/s eta
0:00:00
anylinux1_x86_64.whl (54.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 54.6/54.6 MB 11.2 MB/s eta
0:00:00
anylinux1_x86_64.whl (102.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 102.6/102.6 MB 8.4 MB/s eta
0:00:00
anylinux1_x86_64.whl (173.2 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 173.2/173.2 MB 4.7 MB/s eta
0:00:00
anylinux1_x86_64.whl (177.1 MB)
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 177.1/177.1 MB 5.6 MB/s eta
0:00:00
anylinux1_x86_64.whl (98 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.6/98.6 kB 8.3 MB/s eta
0:00:00
anylinux2014_x86_64.manylinux_2_17_x86_64.whl (63.3 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 63.3/63.3 MB 10.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 96.4/96.4 kB 7.9 MB/s eta
0:00:00
e-cu11, nvidia-cuda-nvrtc-cu11, nvidia-cuda-cupti-cu11, nvidia-cublas-
cu11, nvidia-cusolver-cu11, nvidia-cudnn-cu11, triton, torch,
torchdata, torchtext
  Attempting uninstall: torch
    Found existing installation: torch 2.5.1+cu121
    Uninstalling torch-2.5.1+cu121:
      Successfully uninstalled torch-2.5.1+cu121
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
torchaudio 2.5.1+cu121 requires torch==2.5.1, but you have torch 2.0.1
which is incompatible.
torchvision 0.20.1+cu121 requires torch==2.5.1, but you have torch
2.0.1 which is incompatible.
Successfully installed lit-18.1.8 nvidia-cublas-cu11-11.10.3.66
nvidia-cuda-cupti-cu11-11.7.101 nvidia-cuda-nvrtc-cu11-11.7.99 nvidia-
cuda-runtime-cu11-11.7.99 nvidia-cudnn-cu11-8.5.0.96 nvidia-cufft-
cu11-10.9.0.58 nvidia-curand-cu11-10.2.10.91 nvidia-cusolver-cu11-
11.4.0.1 nvidia-cusparse-cu11-11.7.4.91 nvidia-nccl-cu11-2.14.3
nvidia-nvtx-cu11-11.7.91 torch-2.0.1 torchdata-0.6.1 torchtext-0.15.2
triton-2.0.0
```

```python
import os # Import the 'os' module to use operating system related
functions
import tarfile

# Define the extract_path variable with the actual path
# where your 'ag_news_csv.tar.gz' file is located.
# For example, if it is in your current working directory:
extract_path = os.getcwd()
# Or if it's in a specific folder, provide the full path:
# extract_path = "/path/to/your/folder"


# Now you can proceed with the rest of your code:
ag_news_folder = os.path.join(extract_path, 'ag_news_csv.tar.gz')

# Open the tar.gz file
with tarfile.open(ag_news_folder, 'r:gz') as tar:
    # Get a list of members (files and directories) within the archive
```

```
    files_in_ag_news = tar.getnames()

# Print the list of files
print(files_in_ag_news)
```

```
['ag_news_csv', 'ag_news_csv/train.csv', 'ag_news_csv/test.csv',
'ag_news_csv/classes.txt', 'ag_news_csv/readme.txt']
```

```
# Ekstrak file csv dari arsip tar.gz
with tarfile.open(ag_news_folder, 'r:gz') as tar:
    tar.extractall(path=extract_path)  # Ekstrak semua file ke folder
lokal

# Path ke file yang diekstrak
train_csv_path = os.path.join(extract_path, 'ag_news_csv/train.csv')
test_csv_path = os.path.join(extract_path, 'ag_news_csv/test.csv')

# Import the pandas library
import pandas as pd

# Membaca file CSV dengan pandas
train_data = pd.read_csv(train_csv_path, header=None)
test_data = pd.read_csv(test_csv_path, header=None)

# Cek data
print(train_data.head())
print(test_data.head())
```

```
   0                                                  1  \
0  3  Wall St. Bears Claw Back Into the Black (Reuters)
1  3  Carlyle Looks Toward Commercial Aerospace (Reu...
2  3    Oil and Economy Cloud Stocks' Outlook (Reuters)
3  3  Iraq Halts Oil Exports from Main Southern Pipe...
4  3  Oil prices soar to all-time record, posing new...

                                                   2
0  Reuters - Short-sellers, Wall Street's dwindli...
1  Reuters - Private investment firm Carlyle Grou...
2  Reuters - Soaring crude prices plus worries\ab...
3  Reuters - Authorities have halted oil export\f...
4  AFP - Tearaway world oil prices, toppling reco...
   0                                                  1  \
0  3                  Fears for T N pension after talks
1  4  The Race is On: Second Private Team Sets Launc...
2  4      Ky. Company Wins Grant to Study Peptides (AP)
3  4      Prediction Unit Helps Forecast Wildfires (AP)
4  4        Calif. Aims to Limit Farm-Related Smog (AP)

                                                   2
0  Unions representing workers at Turner   Newall...
1  SPACE.com - TORONTO, Canada -- A second\team o...
```

```
2  AP - A company founded by a chemistry research...
3  AP - It's barely dawn when Mike Fitzpatrick st...
4  AP - Southern California's smog-fighting agenc...
```

```
import pandas as pd

# Path ke file
train_path = 'ag_news_csv/train.csv'
test_path = 'ag_news_csv/test.csv'
classes_path = 'ag_news_csv/classes.txt'

# Load dataset
train_data = pd.read_csv(train_path, header=None)
test_data = pd.read_csv(test_path, header=None)

# Load classes
with open(classes_path, 'r') as f:
    classes = f.read().splitlines()

# Cek data
print("Sample Data Train:")
print(train_data.head())
print("\nClasses:")
print(classes)
```

```
Sample Data Train:
   0                                                  1  \
0  3  Wall St. Bears Claw Back Into the Black (Reuters)
1  3  Carlyle Looks Toward Commercial Aerospace (Reu...
2  3    Oil and Economy Cloud Stocks' Outlook (Reuters)
3  3  Iraq Halts Oil Exports from Main Southern Pipe...
4  3  Oil prices soar to all-time record, posing new...

                                                   2
0  Reuters - Short-sellers, Wall Street's dwindli...
1  Reuters - Private investment firm Carlyle Grou...
2  Reuters - Soaring crude prices plus worries\ab...
3  Reuters - Authorities have halted oil export\f...
4  AFP - Tearaway world oil prices, toppling reco...

Classes:
['World', 'Sports', 'Business', 'Sci/Tech']
```

```
import pandas as pd

# Path ke file
train_path = 'ag_news_csv/train.csv'
test_path = 'ag_news_csv/test.csv'
classes_path = 'ag_news_csv/classes.txt'

# Load dataset
```

```
train_data = pd.read_csv(train_path, header=None)
test_data = pd.read_csv(test_path, header=None)

# Load classes
with open(classes_path, 'r') as f:
    classes = f.read().splitlines()

# Cek data
print("Sample Data Test:")
print(train_data.head())
print("\nClasses:")
print(classes)
```

```
Sample Data Test:
   0                                                  1  \
0  3  Wall St. Bears Claw Back Into the Black (Reuters)
1  3  Carlyle Looks Toward Commercial Aerospace (Reu...
2  3    Oil and Economy Cloud Stocks' Outlook (Reuters)
3  3  Iraq Halts Oil Exports from Main Southern Pipe...
4  3  Oil prices soar to all-time record, posing new...

                                                   2
0  Reuters - Short-sellers, Wall Street's dwindli...
1  Reuters - Private investment firm Carlyle Grou...
2  Reuters - Soaring crude prices plus worries\ab...
3  Reuters - Authorities have halted oil export\f...
4  AFP - Tearaway world oil prices, toppling reco...

Classes:
['World', 'Sports', 'Business', 'Sci/Tech']
```

```
import nltk
from nltk.tokenize import word_tokenize

# Download the 'punkt_tab' data
nltk.download('punkt_tab')

# Download the 'punkt' data if you haven't already
nltk.download('punkt')

def preprocess_text(text):
    return word_tokenize(text.lower())

# Gabungkan title dan description
train_texts = (train_data[1] + " " +
train_data[2]).apply(preprocess_text).tolist()
train_labels = train_data[0].tolist()  # Label

test_texts = (test_data[1] + " " +
test_data[2]).apply(preprocess_text).tolist()
test_labels = test_data[0].tolist()
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
import gensim.downloader as api
import torch

# Load GloVe embeddings
glove = api.load("glove-wiki-gigaword-50")

# Konversi teks ke embeddings
def text_to_embedding(text, model, max_len=50):
    embeddings = [torch.tensor(model[word]) if word in model else
torch.zeros(50) for word in text] # Convert to PyTorch tensor
    if len(embeddings) < max_len:
        embeddings += [torch.zeros(50)] * (max_len - len(embeddings))
    return torch.stack(embeddings[:max_len])

train_embeddings = [text_to_embedding(text, glove) for text in
train_texts]
test_embeddings = [text_to_embedding(text, glove) for text in
test_texts]
```

Modeling: LSTM

```
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader

# Dataset
class TextDataset(Dataset):
    def __init__(self, texts, labels):
        self.texts = texts
        self.labels = torch.tensor(labels) - 1  # Ubah label ke 0-
index

    def __len__(self):
        return len(self.texts)

    def __getitem__(self, idx):
        return self.texts[idx], self.labels[idx]

train_dataset = TextDataset(train_embeddings, train_labels)
test_dataset = TextDataset(test_embeddings, test_labels)

train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=32)

# LSTM Model
class LSTMClassifier(nn.Module):
```

```
    def __init__(self, input_dim, hidden_dim, output_dim):
        super(LSTMClassifier, self).__init__()
        self.lstm = nn.LSTM(input_dim, hidden_dim, batch_first=True)
        self.fc = nn.Linear(hidden_dim, output_dim)

    def forward(self, x):
        _, (hidden, _) = self.lstm(x)
        out = self.fc(hidden[-1])
        return out

# Model setup
input_dim = 50  # Dimensi GloVe
hidden_dim = 128
output_dim = 4  # Jumlah kelas

model = LSTMClassifier(input_dim, hidden_dim, output_dim)
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)

# Training Loop
for epoch in range(5):
    model.train()
    total_loss = 0
    for texts, labels in train_loader:
        optimizer.zero_grad()
        outputs = model(texts)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()

    print(f"Epoch {epoch+1}, Loss: {total_loss/len(train_loader)}")
```

```
Epoch 1, Loss: 0.46592242943644524
Epoch 2, Loss: 0.2954050063172976
Epoch 3, Loss: 0.25779516075303155
Epoch 4, Loss: 0.22949459938357275
Epoch 5, Loss: 0.20651824945807457
```

Evaluasi Model

```
def evaluate(model, data_loader):
    model.eval()
    correct = 0
    total = 0
    with torch.no_grad():
        for texts, labels in data_loader:
            outputs = model(texts)
            _, predicted = torch.max(outputs, 1)
            correct += (predicted == labels).sum().item()
```

```
            total += labels.size(0)
    return correct / total

accuracy = evaluate(model, test_loader)
print(f"Test Accuracy: {accuracy:.4f}")
```

```
Test Accuracy: 0.9186
```

Model Fast Text

```
!pip install fasttext
```

```
Requirement already satisfied: fasttext in
/usr/local/lib/python3.10/dist-packages (0.9.3)
Requirement already satisfied: pybind11>=2.2 in
/usr/local/lib/python3.10/dist-packages (from fasttext) (2.13.6)
Requirement already satisfied: setuptools>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from fasttext) (75.1.0)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from fasttext) (1.26.4)
```

```
!wget https://dl.fbaipublicfiles.com/fasttext/vectors-
crawl/cc.en.300.bin.gz
!gunzip cc.en.300.bin.gz
```

```
--2024-12-15 02:33:48--
https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.bin.gz
Resolving dl.fbaipublicfiles.com (dl.fbaipublicfiles.com)...
3.163.189.14, 3.163.189.51, 3.163.189.96, ...
Connecting to dl.fbaipublicfiles.com (dl.fbaipublicfiles.com)|
3.163.189.14|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4503593528 (4.2G) [application/octet-stream]
Saving to: ‘cc.en.300.bin.gz’

cc.en.300.bin.gz    100%[===================>]   4.19G  32.4MB/s    in
1m 59s

2024-12-15 02:35:48 (36.0 MB/s) - ‘cc.en.300.bin.gz’ saved
[4503593528/4503593528]

gzip: cc.en.300.bin already exists; do you wish to overwrite (y or n)?
n
     not overwritten
```

```
# 4. Load FastText Pretrained Embeddings
import fasttext
fasttext_model = fasttext.load_model('cc.en.300.bin')  # Path to the
downloaded FastText model
```

```
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import Dataset, DataLoader
import pandas as pd
import os

# 1. Dataset Class
class AGNewsFastTextDataset(Dataset):
    def __init__(self, file_path, fasttext_model):
        # Load CSV
        self.data = pd.read_csv(file_path, header=None)
        self.texts = self.data[1]
        self.labels = self.data[0] - 1  # Make labels 0-indexed

        # Preload FastText embeddings
        self.fasttext_model = fasttext_model

    def __len__(self):
        return len(self.labels)

    def __getitem__(self, idx):
        # Convert text to FastText embeddings
        tokens = self.texts[idx].split()
        embeddings = [self.fasttext_model.get_word_vector(word) for
word in tokens]
        embeddings_tensor = torch.tensor(embeddings)

        # Return padded embeddings and label
        return embeddings_tensor, torch.tensor(self.labels[idx])

# 2. Collate Function for Padding
def collate_fn(batch):
    texts, labels = zip(*batch)
    lengths = torch.tensor([len(text) for text in texts])
    padded_texts = nn.utils.rnn.pad_sequence(texts, batch_first=True)
    return padded_texts, torch.stack(labels), lengths

# 3. Model Definition
class TextClassificationModel(nn.Module):
    def __init__(self, embedding_dim, hidden_dim, num_classes):
        super(TextClassificationModel, self).__init__()
        self.lstm = nn.LSTM(embedding_dim, hidden_dim,
batch_first=True)
        self.fc = nn.Linear(hidden_dim, num_classes)

    def forward(self, x, lengths):
        # Pack padded sequence
        packed_input = nn.utils.rnn.pack_padded_sequence(x, lengths,
batch_first=True, enforce_sorted=False)
```

```
        packed_output, (hidden, _) = self.lstm(packed_input)
        out = self.fc(hidden[-1])
        return out

# 4. Load FastText Pretrained Embeddings
import fasttext
fasttext_model = fasttext.load_model('cc.en.300.bin')  # Adjust path
to your FastText model

# 5. Paths and Parameters
train_path = "ag_news_csv/train.csv"
test_path = "ag_news_csv/test.csv"
embedding_dim = 300
hidden_dim = 128
num_classes = 4
batch_size = 32
epochs = 5
learning_rate = 0.001

# 6. Dataset and DataLoader
train_dataset = AGNewsFastTextDataset(train_path, fasttext_model)
test_dataset = AGNewsFastTextDataset(test_path, fasttext_model)

train_loader = DataLoader(train_dataset, batch_size=batch_size,
shuffle=True, collate_fn=collate_fn)
test_loader = DataLoader(test_dataset, batch_size=batch_size,
collate_fn=collate_fn)

# 7. Model, Loss, Optimizer
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = TextClassificationModel(embedding_dim, hidden_dim,
num_classes).to(device)
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=learning_rate)

# 8. Training Function
def train(model, dataloader, optimizer, criterion):
    model.train()
    total_loss = 0
    for texts, labels, lengths in dataloader:
        texts, labels, lengths = texts.to(device), labels.to(device),
lengths.to(device)
        optimizer.zero_grad()
        outputs = model(texts, lengths)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
    return total_loss / len(dataloader)
```

```python
# 9. Evaluation Function
def evaluate(model, dataloader):
    model.eval()
    correct = 0
    total = 0
    with torch.no_grad():
        for texts, labels, lengths in dataloader:
            texts, labels, lengths = texts.to(device),
labels.to(device), lengths.to(device)
            outputs = model(texts, lengths)
            _, predicted = torch.max(outputs, 1)
            correct += (predicted == labels).sum().item()
            total += labels.size(0)
    return correct / total

# 10. Training Loop
for epoch in range(epochs):
    train_loss = train(model, train_loader, optimizer, criterion)
    accuracy = evaluate(model, test_loader)
    print(f"Epoch {epoch+1}, Loss: {train_loss:.4f}, Test Accuracy:
{accuracy:.4f}")
```

```
<ipython-input-2-2730197cf07b>:26: UserWarning: Creating a tensor from
a list of numpy.ndarrays is extremely slow. Please consider converting
the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:245.)
  embeddings_tensor = torch.tensor(embeddings)

Epoch 1, Loss: 0.4952, Test Accuracy: 0.8479
Epoch 2, Loss: 0.3988, Test Accuracy: 0.8572
Epoch 3, Loss: 0.3611, Test Accuracy: 0.8667
Epoch 4, Loss: 0.3301, Test Accuracy: 0.8718
Epoch 5, Loss: 0.3029, Test Accuracy: 0.8759
```

Model Bertt

```
pip install torch==1.11.0 transformers==4.12.0

Collecting torch==1.11.0
  Downloading torch-1.11.0-cp310-cp310-manylinux1_x86_64.whl.metadata
(24 kB)
Collecting transformers==4.12.0
  Using cached transformers-4.12.0-py3-none-any.whl.metadata (56 kB)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.10/dist-packages (from torch==1.11.0) (4.12.2)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from transformers==4.12.0)
(3.16.1)
Requirement already satisfied: huggingface-hub>=0.0.17 in
```

```
/usr/local/lib/python3.10/dist-packages (from transformers==4.12.0)
(0.26.5)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.10/dist-packages (from transformers==4.12.0)
(1.26.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from transformers==4.12.0)
(24.2)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.10/dist-packages (from transformers==4.12.0)
(6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from transformers==4.12.0)
(2024.9.11)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from transformers==4.12.0)
(2.32.3)
Collecting sacremoses (from transformers==4.12.0)
  Using cached sacremoses-0.1.1-py3-none-any.whl.metadata (8.3 kB)
Collecting tokenizers<0.11,>=0.10.1 (from transformers==4.12.0)
  Using cached tokenizers-0.10.3.tar.gz (212 kB)
  Installing build dependencies ... ents to build wheel ... etadata
(pyproject.toml) ... ent already satisfied: tqdm>=4.27 in
/usr/local/lib/python3.10/dist-packages (from transformers==4.12.0)
(4.66.6)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.0.17-
>transformers==4.12.0) (2024.10.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests-
>transformers==4.12.0) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests-
>transformers==4.12.0) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests-
>transformers==4.12.0) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests-
>transformers==4.12.0) (2024.8.30)
Requirement already satisfied: click in
/usr/local/lib/python3.10/dist-packages (from sacremoses-
>transformers==4.12.0) (8.1.7)
Requirement already satisfied: joblib in
/usr/local/lib/python3.10/dist-packages (from sacremoses-
>transformers==4.12.0) (1.4.2)
Downloading torch-1.11.0-cp310-cp310-manylinux1_x86_64.whl (750.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 750.6/750.6 MB 375.5 kB/s eta
0:00:00
```

```
ers-4.12.0-py3-none-any.whl (3.1 MB)
Using cached sacremoses-0.1.1-py3-none-any.whl (897 kB)
Building wheels for collected packages: tokenizers
  error: subprocess-exited-with-error

  × Building wheel for tokenizers (pyproject.toml) did not run
successfully.
  │ exit code: 1
  ╰─> See above for output.

  note: This error originates from a subprocess, and is likely not a
problem with pip.
  Building wheel for tokenizers (pyproject.toml) ...   ERROR: Failed
building wheel for tokenizers
Failed to build tokenizers
ERROR: ERROR: Failed to build installable wheels for some
pyproject.toml based projects (tokenizers)
```

```python
import pandas as pd

# Membaca file train.csv dan test.csv
train_data = pd.read_csv('ag_news_csv/train.csv', header=None)
test_data = pd.read_csv('ag_news_csv/test.csv', header=None)

# Melihat beberapa sampel data
print(train_data.head())

# Struktur: Kolom pertama adalah label, kolom kedua adalah judul
berita, kolom ketiga adalah teks berita
```

```
   0                                                  1  \
0  3  Wall St. Bears Claw Back Into the Black (Reuters)
1  3  Carlyle Looks Toward Commercial Aerospace (Reu...
2  3    Oil and Economy Cloud Stocks' Outlook (Reuters)
3  3  Iraq Halts Oil Exports from Main Southern Pipe...
4  3  Oil prices soar to all-time record, posing new...

                                                   2
0  Reuters - Short-sellers, Wall Street's dwindli...
1  Reuters - Private investment firm Carlyle Grou...
2  Reuters - Soaring crude prices plus worries\ab...
3  Reuters - Authorities have halted oil export\f...
4  AFP - Tearaway world oil prices, toppling reco...
```

```python
# Menggabungkan judul dan teks berita
train_data[1] = train_data[1] + ". " + train_data[2]  # Gabungkan
judul (kolom 1) dan teks (kolom 2)
test_data[1] = test_data[1] + ". " + test_data[2]

# Hapus kolom yang tidak diperlukan (kolom 2, karena sudah
```

```
digabungkan)
train_data = train_data[[0, 1]]  # Hanya gunakan label dan teks
gabungan
test_data = test_data[[0, 1]]

# Mengecek struktur data setelah gabungan
print(train_data.head())
```

```
   0                                                  1
0  3  Wall St. Bears Claw Back Into the Black (Reute...
1  3  Carlyle Looks Toward Commercial Aerospace (Reu...
2  3  Oil and Economy Cloud Stocks' Outlook (Reuters...
3  3  Iraq Halts Oil Exports from Main Southern Pipe...
4  3  Oil prices soar to all-time record, posing new...
```

```
import torch
from torch.utils.data import Dataset
from transformers import BertTokenizer # Import the BertTokenizer

# Load tokenizer
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased') #
Define tokenizer

class AGNewsDataset(Dataset):
    def __init__(self, texts, labels, tokenizer, max_len=128):
        self.texts = texts
        self.labels = labels
        self.tokenizer = tokenizer
        self.max_len = max_len

    def __len__(self):
        return len(self.texts)

    def __getitem__(self, idx):
        text = self.texts[idx]
        label = self.labels[idx]

        # Tokenisasi dengan padding dan truncation
        inputs = self.tokenizer(text, max_length=self.max_len,
padding='max_length', truncation=True, return_tensors="pt")

        input_ids = inputs['input_ids'].squeeze(0)  # Menghilangkan
batch dimension
        attention_mask = inputs['attention_mask'].squeeze(0)

        return {
            'input_ids': input_ids,
            'attention_mask': attention_mask,
            'labels': torch.tensor(label, dtype=torch.long)
        }
```

```
# Memproses dataset
train_texts = train_data[1].tolist()
train_labels = train_data[0].tolist()

test_texts = test_data[1].tolist()
test_labels = test_data[0].tolist()

# Membuat dataset PyTorch
train_dataset = AGNewsDataset(train_texts, train_labels, tokenizer)
test_dataset = AGNewsDataset(test_texts, test_labels, tokenizer)
```

```
/usr/local/lib/python3.10/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":"f6ad4987ffe14a87a92cb2d17ca4e845","version_major":2,"vers
ion_minor":0}

{"model_id":"f54a481721384f109ee82cec52dbe776","version_major":2,"vers
ion_minor":0}

{"model_id":"f821507239524c7eaebd0d20c9bdd660","version_major":2,"vers
ion_minor":0}

{"model_id":"a0c025f4ac304227a3ba30d0897a38f7","version_major":2,"vers
ion_minor":0}
```


```
!pip install --upgrade torch
!pip install --upgrade transformers
```

```
Requirement already satisfied: torch in
/usr/local/lib/python3.10/dist-packages (2.5.1)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch) (4.12.2)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch) (3.1.4)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch) (2024.10.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in
/usr/local/lib/python3.10/dist-packages (from torch) (12.4.127)
```

```
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127
in /usr/local/lib/python3.10/dist-packages (from torch) (12.4.127)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in
/usr/local/lib/python3.10/dist-packages (from torch) (12.4.127)
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in
/usr/local/lib/python3.10/dist-packages (from torch) (9.1.0.70)
Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in
/usr/local/lib/python3.10/dist-packages (from torch) (12.4.5.8)
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in
/usr/local/lib/python3.10/dist-packages (from torch) (11.2.1.3)
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in
/usr/local/lib/python3.10/dist-packages (from torch) (10.3.5.147)
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in
/usr/local/lib/python3.10/dist-packages (from torch) (11.6.1.9)
Requirement already satisfied: nvidia-cusparse-cu12==12.3.1.170 in
/usr/local/lib/python3.10/dist-packages (from torch) (12.3.1.170)
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in
/usr/local/lib/python3.10/dist-packages (from torch) (2.21.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in
/usr/local/lib/python3.10/dist-packages (from torch) (12.4.127)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in
/usr/local/lib/python3.10/dist-packages (from torch) (12.4.127)
Requirement already satisfied: triton==3.1.0 in
/usr/local/lib/python3.10/dist-packages (from torch) (3.1.0)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.10/dist-packages (from torch) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch)
(1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch) (3.0.2)
Requirement already satisfied: transformers in
/usr/local/lib/python3.10/dist-packages (4.47.0)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from transformers) (3.16.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.24.0 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.26.5)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.10/dist-packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from transformers)
(2024.9.11)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in
```

```
/usr/local/lib/python3.10/dist-packages (from transformers) (0.21.0)
Requirement already satisfied: safetensors>=0.4.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.4.5)
Requirement already satisfied: tqdm>=4.27 in
/usr/local/lib/python3.10/dist-packages (from transformers) (4.66.6)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.24.0->transformers) (2024.10.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.24.0->transformers) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(2024.8.30)

import torch
import transformers

print(torch.__version__)          # Versi PyTorch
print(transformers.__version__)   # Versi Transformers

2.0.1+cu117
4.47.0

!pip install torch==1.13.1 torchvision==0.14.1 torchaudio==0.13.1
!pip install --upgrade transformers

Collecting torch==1.13.1
  Downloading torch-1.13.1-cp310-cp310-manylinux1_x86_64.whl.metadata
(24 kB)
Collecting torchvision==0.14.1
  Downloading torchvision-0.14.1-cp310-cp310-
manylinux1_x86_64.whl.metadata (11 kB)
Collecting torchaudio==0.13.1
  Downloading torchaudio-0.13.1-cp310-cp310-
manylinux1_x86_64.whl.metadata (1.2 kB)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.10/dist-packages (from torch==1.13.1) (4.12.2)
Requirement already satisfied: nvidia-cuda-runtime-cu11==11.7.99 in
/usr/local/lib/python3.10/dist-packages (from torch==1.13.1) (11.7.99)
Requirement already satisfied: nvidia-cudnn-cu11==8.5.0.96 in
```

```
/usr/local/lib/python3.10/dist-packages (from torch==1.13.1)
(8.5.0.96)
Requirement already satisfied: nvidia-cublas-cu11==11.10.3.66 in
/usr/local/lib/python3.10/dist-packages (from torch==1.13.1)
(11.10.3.66)
Requirement already satisfied: nvidia-cuda-nvrtc-cu11==11.7.99 in
/usr/local/lib/python3.10/dist-packages (from torch==1.13.1) (11.7.99)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from torchvision==0.14.1)
(1.26.4)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from torchvision==0.14.1)
(2.32.3)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/usr/local/lib/python3.10/dist-packages (from torchvision==0.14.1)
(11.0.0)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from nvidia-cublas-
cu11==11.10.3.66->torch==1.13.1) (75.1.0)
Requirement already satisfied: wheel in
/usr/local/lib/python3.10/dist-packages (from nvidia-cublas-
cu11==11.10.3.66->torch==1.13.1) (0.45.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests-
>torchvision==0.14.1) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests-
>torchvision==0.14.1) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests-
>torchvision==0.14.1) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests-
>torchvision==0.14.1) (2024.8.30)
Downloading torch-1.13.1-cp310-cp310-manylinux1_x86_64.whl (887.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 887.5/887.5 MB 2.0 MB/s eta
0:00:00
anylinux1_x86_64.whl (24.2 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.2/24.2 MB 70.5 MB/s eta
0:00:00
anylinux1_x86_64.whl (4.2 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.2/4.2 MB 82.9 MB/s eta
0:00:00
pting uninstall: torch
    Found existing installation: torch 2.0.1
    Uninstalling torch-2.0.1:
      Successfully uninstalled torch-2.0.1
  Attempting uninstall: torchvision
    Found existing installation: torchvision 0.20.1+cu121
```

```
    Uninstalling torchvision-0.20.1+cu121:
      Successfully uninstalled torchvision-0.20.1+cu121
  Attempting uninstall: torchaudio
    Found existing installation: torchaudio 2.5.1+cu121
    Uninstalling torchaudio-2.5.1+cu121:
      Successfully uninstalled torchaudio-2.5.1+cu121
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
torchdata 0.6.1 requires torch==2.0.1, but you have torch 1.13.1 which
is incompatible.
torchtext 0.15.2 requires torch==2.0.1, but you have torch 1.13.1
which is incompatible.
Successfully installed torch-1.13.1 torchaudio-0.13.1 torchvision-
0.14.1
Requirement already satisfied: transformers in
/usr/local/lib/python3.10/dist-packages (4.46.3)
Collecting transformers
  Downloading transformers-4.47.0-py3-none-any.whl.metadata (43 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.5/43.5 kB 1.8 MB/s eta
0:00:00
ent already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from transformers) (3.16.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.24.0 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.26.5)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.10/dist-packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from transformers)
(2024.9.11)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from transformers) (2.32.3)
Collecting tokenizers<0.22,>=0.21 (from transformers)
  Downloading tokenizers-0.21.0-cp39-abi3-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.7 kB)
Requirement already satisfied: safetensors>=0.4.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.4.5)
Requirement already satisfied: tqdm>=4.27 in
/usr/local/lib/python3.10/dist-packages (from transformers) (4.66.6)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.24.0->transformers) (2024.10.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.24.0->transformers) (4.12.2)
```

```
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(2024.8.30)
Downloading transformers-4.47.0-py3-none-any.whl (10.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.1/10.1 MB 56.9 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.0 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.0/3.0 MB 60.4 MB/s eta
0:00:00
ers
  Attempting uninstall: tokenizers
    Found existing installation: tokenizers 0.20.3
    Uninstalling tokenizers-0.20.3:
      Successfully uninstalled tokenizers-0.20.3
  Attempting uninstall: transformers
    Found existing installation: transformers 4.46.3
    Uninstalling transformers-4.46.3:
      Successfully uninstalled transformers-4.46.3
Successfully installed tokenizers-0.21.0 transformers-4.47.0
```

```python
import torch
from transformers import DistilBertTokenizer,
DistilBertForSequenceClassification, AdamW
from torch.utils.data import Dataset, DataLoader
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import pandas as pd
from torch.cuda.amp import autocast, GradScaler

# Cek apakah menggunakan GPU
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print(f"Using device: {device}")

# Load tokenizer untuk DistilBERT
tokenizer = DistilBertTokenizer.from_pretrained('distilbert-base-
uncased')

# Dataset
class AGNewsBERTDataset(Dataset):
    def __init__(self, data):
        # Menggabungkan judul dan deskripsi untuk input teks
```

```
        self.encodings = tokenizer(list(data[1] + " " + data[2]),
truncation=True, padding=True, max_length=64, return_tensors='pt')
        self.labels = torch.tensor(data[0].tolist()) - 1  # Label
dikurangi 1 karena dimulai dari 1 di AG News

    def __getitem__(self, idx):
        item = {key: val[idx] for key, val in self.encodings.items()}
        item['labels'] = self.labels[idx]
        return item

    def __len__(self):
        return len(self.labels)

# Model DistilBERT untuk Klasifikasi
model =
DistilBertForSequenceClassification.from_pretrained('distilbert-base-
uncased', num_labels=4).to(device)

# Optimizer menggunakan versi torch AdamW
optimizer = AdamW(model.parameters(), lr=5e-5)

# Loss function
criterion = torch.nn.CrossEntropyLoss()

# Mixed precision scaler
scaler = GradScaler()

# Train function with mixed precision
def train(model, dataloader, optimizer, criterion):
    model.train()
    total_loss = 0
    for i, batch in enumerate(dataloader):
        optimizer.zero_grad()
        # Move batch to device
        for key in batch:
            batch[key] = batch[key].to(device)

        # Mixed precision
        with autocast():
            outputs = model(**batch)
            loss = criterion(outputs.logits, batch['labels'])

        # Backpropagation with scaling
        scaler.scale(loss).backward()
        scaler.step(optimizer)
        scaler.update()

        total_loss += loss.item()

        # Print every 10 batches
```

```
        if i % 10 == 0:
            print(f"Batch {i}, Loss: {loss.item()}")

    return total_loss / len(dataloader)

# Evaluate function
def evaluate(model, dataloader):
    model.eval()
    correct = 0
    total = 0
    all_preds = []
    all_labels = []
    with torch.no_grad():
        for batch in dataloader:
            for key in batch:
                batch[key] = batch[key].to(device)
            outputs = model(**batch)
            _, predicted = torch.max(outputs.logits, 1)
            correct += (predicted == batch['labels']).sum().item()
            total += len(batch['labels'])
            all_preds.extend(predicted.cpu().numpy())
            all_labels.extend(batch['labels'].cpu().numpy())

    accuracy = correct / total
    return accuracy, all_preds, all_labels

# Membaca data
train_data = pd.read_csv('ag_news_csv/train.csv', header=None)
test_data = pd.read_csv('ag_news_csv/test.csv', header=None)

# Membatasi jumlah data untuk eksperimen cepat
train_data = train_data.sample(200, random_state=42)  # Menggunakan
200 data untuk training
val_data = train_data.sample(50, random_state=42)     # Menggunakan 50
data untuk validasi
test_data = test_data.sample(50, random_state=42)     # Menggunakan 50
data untuk testing

# Membuat Dataset dan DataLoader
train_dataset = AGNewsBERTDataset(train_data)
val_dataset = AGNewsBERTDataset(val_data)
test_dataset = AGNewsBERTDataset(test_data)

# Menggunakan batch size yang lebih kecil
train_loader = DataLoader(train_dataset, batch_size=2, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=2)
test_loader = DataLoader(test_dataset, batch_size=2)

# Training dan Evaluasi
for epoch in range(3):  # Kita mulai dengan 3 epoch
```

```
    train_loss = train(model, train_loader, optimizer, criterion)
    accuracy, _, _ = evaluate(model, val_loader)
    print(f"Epoch {epoch+1}, Loss: {train_loss:.4f}, Validation
Accuracy: {accuracy:.4f}")

# Evaluasi pada dataset test
test_accuracy, test_preds, test_labels = evaluate(model, test_loader)
print(f"Test Accuracy: {test_accuracy:.4f}")

# Menghitung laporan klasifikasi menggunakan sklearn
print(classification_report(test_labels, test_preds))
```

```
Using device: cpu

/usr/local/lib/python3.10/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":"e778471097894cfab9f5ca8ec4d2b29c","version_major":2,"vers
ion_minor":0}

{"model_id":"ab01bf8cc01c4a0595635d2e123e4bd6","version_major":2,"vers
ion_minor":0}

{"model_id":"0777a429562043b09135475632df184a","version_major":2,"vers
ion_minor":0}

{"model_id":"c89ea32f673e40a8a6d07bb4a27aff55","version_major":2,"vers
ion_minor":0}

{"model_id":"aad23a6e7a8f4e84be6416ba5ec7af04","version_major":2,"vers
ion_minor":0}
```


```
Some weights of DistilBertForSequenceClassification 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/optimization.py:5
91: FutureWarning: This implementation of AdamW is deprecated and will
be removed in a future version. Use the PyTorch implementation
torch.optim.AdamW instead, or set `no_deprecation_warning=True` to
disable this warning
```

```
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torch/cuda/amp/grad_scaler.py:118: UserWarning: torch.cuda.amp.GradScaler is enabled, but CUDA is not available.  Disabling.
  warnings.warn("torch.cuda.amp.GradScaler is enabled, but CUDA is not available.  Disabling.")
/usr/local/lib/python3.10/dist-packages/torch/amp/autocast_mode.py:202: UserWarning: User provided device_type of 'cuda', but CUDA is not available. Disabling
  warnings.warn('User provided device_type of \'cuda\', but CUDA is not available. Disabling')

Batch 0, Loss: 1.352712631225586
Batch 10, Loss: 1.9594247341156006
Batch 20, Loss: 1.4823451042175293
Batch 30, Loss: 0.8037070631980896
Batch 40, Loss: 1.2655599117279053
Batch 50, Loss: 0.9209012985229492
Batch 60, Loss: 0.11161141097545624
Batch 70, Loss: 1.0713199377059937
Batch 80, Loss: 0.4819076359272003
Batch 90, Loss: 0.05450010299682617
Epoch 1, Loss: 0.9026, Validation Accuracy: 0.8800
Batch 0, Loss: 0.23489952087402344
Batch 10, Loss: 0.39673376083374023
Batch 20, Loss: 0.2568333148956299
Batch 30, Loss: 1.0112831592559814
Batch 40, Loss: 0.13880038261413574
Batch 50, Loss: 0.13768735527992249
Batch 60, Loss: 1.3489625453948975
Batch 70, Loss: 0.1452878713607788
Batch 80, Loss: 0.37681642174720764
Batch 90, Loss: 0.7816606760025024
Epoch 2, Loss: 0.3918, Validation Accuracy: 0.9200
Batch 0, Loss: 0.14666102826595306
Batch 10, Loss: 0.8269717693328857
Batch 20, Loss: 0.06775273382663727
Batch 30, Loss: 0.01092933677136898
Batch 40, Loss: 0.347779244184494
Batch 50, Loss: 0.017392387613654137
Batch 60, Loss: 0.02455771341919899
Batch 70, Loss: 0.029635775834321976
Batch 80, Loss: 0.03560473769903183
Batch 90, Loss: 0.0841277539730072
Epoch 3, Loss: 0.1732, Validation Accuracy: 0.9800
Test Accuracy: 0.8000
              precision    recall  f1-score   support

           0       0.80      0.89      0.84         9
           1       1.00      1.00      1.00        15
```

```
           2       0.75      0.56      0.64        16
           3       0.62      0.80      0.70        10

    accuracy                           0.80        50
   macro avg       0.79      0.81      0.80        50
weighted avg       0.81      0.80      0.80        50
```

Metode Transfomer

```
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader, Dataset
from transformers import BertTokenizer
import pandas as pd
from sklearn.model_selection import train_test_split

# Hyperparameters
MAX_LEN = 128  # Maksimum panjang sequence input
EMBED_SIZE = 512  # Ukuran embedding
NUM_CLASSES = 4  # Jumlah kelas di dataset AG News
NUM_HEADS = 8  # Jumlah attention heads
NUM_ENCODER_LAYERS = 6  # Jumlah layer encoder
BATCH_SIZE = 32
LEARNING_RATE = 1e-4
EPOCHS = 5

# Custom Dataset class
class AGNewsDataset(Dataset):
    def __init__(self, data, tokenizer):
        # Tokenisasi teks input (menggabungkan judul dan deskripsi
berita)
        self.encodings = tokenizer(list(data[1] + " " + data[2]),
truncation=True, padding=True, max_length=MAX_LEN,
return_tensors='pt')
        # Label dikurangi 1 agar dimulai dari 0
        self.labels = torch.tensor(data[0].tolist()) - 1

    def __getitem__(self, idx):
        item = {key: val[idx] for key, val in self.encodings.items()}
        item['labels'] = self.labels[idx]
        return item

    def __len__(self):
        return len(self.labels)

# Transformer Model
class TransformerClassifier(nn.Module):
    def __init__(self, embed_size, num_heads, num_encoder_layers,
```

```
num_classes):
        super(TransformerClassifier, self).__init__()

        # Embedding layer dan positional encoding
        self.embedding = nn.Embedding(30522, embed_size)  # Vocabulary
size dari tokenizer BERT
        self.positional_encoding = nn.Parameter(torch.zeros(1,
MAX_LEN, embed_size))

        # Layer encoder transformer
        encoder_layer = nn.TransformerEncoderLayer(d_model=embed_size,
nhead=num_heads)
        self.transformer_encoder =
nn.TransformerEncoder(encoder_layer, num_layers=num_encoder_layers)

        # Layer fully connected untuk klasifikasi
        self.fc = nn.Linear(embed_size, num_classes)

    def forward(self, x):
        # Embed token input dan tambahkan positional encoding
        embedding_output = self.embedding(x) +
self.positional_encoding[:, :x.size(1), :]

        # Output dari transformer encoder
        transformer_output =
self.transformer_encoder(embedding_output)

        # Mean pooling di sepanjang dimensi sequence length
        cls_token = transformer_output.mean(dim=1)

        # Layer fully connected untuk menghasilkan logits kelas
        logits = self.fc(cls_token)
        return logits

# Load tokenizer dari Huggingface (BERT tokenizer)
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')

# Load dataset AG News
train_data = pd.read_csv('ag_news_csv/train.csv', header=None)
test_data = pd.read_csv('ag_news_csv/test.csv', header=None)

# Membatasi ukuran dataset agar lebih cepat (misalnya 1000 contoh
train, 200 contoh test)
train_data = train_data.sample(1000, random_state=42)
test_data = test_data.sample(200, random_state=42)

# Split dataset ke training dan validasi (90% training, 10% validasi)
train_data, val_data = train_test_split(train_data, test_size=0.1,
random_state=42)
```

```
# Buat dataset dan DataLoader
train_dataset = AGNewsDataset(train_data, tokenizer)
val_dataset = AGNewsDataset(val_data, tokenizer)
test_dataset = AGNewsDataset(test_data, tokenizer)

train_loader = DataLoader(train_dataset, batch_size=BATCH_SIZE,
shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=BATCH_SIZE)
test_loader = DataLoader(test_dataset, batch_size=BATCH_SIZE)

# Inisialisasi model, optimizer, dan loss function
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = TransformerClassifier(embed_size=EMBED_SIZE,
num_heads=NUM_HEADS, num_encoder_layers=NUM_ENCODER_LAYERS,
num_classes=NUM_CLASSES).to(device)

optimizer = optim.Adam(model.parameters(), lr=LEARNING_RATE)
criterion = nn.CrossEntropyLoss()

# Fungsi untuk melatih model
def train(model, train_loader, optimizer, criterion):
    model.train()
    total_loss = 0
    for batch in train_loader:
        inputs = batch['input_ids'].to(device)
        labels = batch['labels'].to(device)

        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

        total_loss += loss.item()

    return total_loss / len(train_loader)

# Fungsi untuk evaluasi model
def evaluate(model, val_loader):
    model.eval()
    correct = 0
    total = 0
    with torch.no_grad():
        for batch in val_loader:
            inputs = batch['input_ids'].to(device)
            labels = batch['labels'].to(device)

            outputs = model(inputs)
            _, predicted = torch.max(outputs, 1)
```

```
            correct += (predicted == labels).sum().item()
            total += labels.size(0)

    accuracy = correct / total
    return accuracy

# Training loop
for epoch in range(EPOCHS):
    train_loss = train(model, train_loader, optimizer, criterion)
    val_accuracy = evaluate(model, val_loader)
    print(f"Epoch {epoch + 1}/{EPOCHS}, Loss: {train_loss:.4f},
Validation Accuracy: {val_accuracy:.4f}")

# Evaluasi model di dataset test
test_accuracy = evaluate(model, test_loader)
print(f"Test Accuracy: {test_accuracy:.4f}")
```

```
Epoch 1/5, Loss: 1.5777, Validation Accuracy: 0.2900
Epoch 2/5, Loss: 1.4092, Validation Accuracy: 0.2900
Epoch 3/5, Loss: 1.4015, Validation Accuracy: 0.2900
Epoch 4/5, Loss: 1.3790, Validation Accuracy: 0.3400
Epoch 5/5, Loss: 1.3728, Validation Accuracy: 0.3000
Test Accuracy: 0.2400
```

Buat Analisis Perbandingan model di atas dengan parameter:

Dataset (Apakah membutuhkan yang lebih besar?)

Waktu dan Sumber Daya Komputasi

Jelaskan Generalisas

Dataset LSTM : Membutuhkan dataset besar FEST TEXT : Cocok untuk dataset kecil hingga besar. DistilBERT : Cocok untuk dataset kecil hingga menengah. TRANSFOMER : Membutuhkan dataset besar

Waktu dan sumberdaya komputasi LSTM : Pelatihan cepat, cocok untuk GPU sederhana. FEST TEXT : Cepat dilatih bahkan pada CPU. DistilBERT : Sangat berat, membutuhkan GPU/TPU canggih. TRANSFOMER : Lumayan berat mungkin karna faktor sinyal

Generalisasi : LSTM : Kurang baik pada data kecil, rentan underfit. FEST TEXT : Generalisasi baik untuk kata-kata umum, kurang baik menangani konteks kompleks. DistilBERT : Generalisasi hampir setara dengan BERT TRANSFOMER : Generalisasi nya agak kurang baik

disini saya meggunakan DistilBERT karna mencoba YangBertnya itu bebrapa jam tidak jalan