Praktikum 1.2 Natural Language Processing

Nama : Ronggur Mahendra Widya Putra

NIM : 13519008

```
!pip install datasets
!pip install transformers==4.28.0
!pip install evaluate
```

```
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/d
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: transformers==4.28.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: huggingface-hub<1.0,>=0.11.0 in /usr/local/lib/python
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in /usr/local/lib/p
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: evaluate in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: datasets>=2.0.0 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: dill in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: multiprocess in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: fsspec[http]>=2021.05.0 in /usr/local/lib/python3.10/
Requirement already satisfied: huggingface-hub>=0.7.0 in /usr/local/lib/python3.10/d
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: responses<0.19 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: pyarrow>=8.0.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/d
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in /usr/local/lib/python3
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-pa
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
#Import Libraries
import tensorflow as tf
import pandas as pd
import numpy as np
from tensorflow.keras.layers import Embedding, LSTM, Dense, Bidirectional
from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split
from datasets import load_dataset
from keras.layers import Dense
# from keras.utils.vis_utils import plot_model
from tensorflow.keras.utils import plot_model
```

Resources ×

You are subscribed to Colab Pro. Learn more.
Available: 99.74 compute units
Usage rate: approximately 0.16 per hour
You have 2 active sessions. Manage sessions

Want even more memory and disk space?
Upgrade to Colab Pro+

Not connected to runtime.

Change runtime type

```
# from transformers import BertTokenizer, TFBertModel
from tensorflow import keras
from sklearn.metrics import accuracy_score


# Load Train data
train_df = pd.read_parquet('./train-00000-of-00001-04b49ae22f595095.parquet', engine='pyarrow
train_df.head(10)
```

	text	label
0	– Scope 3: Optional scope that includes indire...	1
1	The Group is not aware of any noise pollution ...	0
2	Global climate change could exacerbate certain...	0
3	Setting an investment horizon is part and parc...	0
4	Climate change the physical impacts of climate...	0
5	Projects with potential limited adverse social...	0
6	We emitted 13.4 million tonnes CO2 of Scope 2 ...	1
7	We do not provide normalised figures for our C...	1
8	We anticipate that the potential effects of cl...	0
9	Enhancing our responsible screening criteria N...	0

```
train_df.describe()
```

	label
count	1000.000000
mean	0.908000
std	0.764278
min	0.000000
25%	0.000000
50%	1.000000
75%	1.250000

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
train_df_data  = train_df['text'].to_list()
train_df_label = train_df['label'].to_list()


# Split data
train_data, val_data, train_label, val_label  = train_test_split(train_df_data, train_df_labe


# load test data
test_df = pd.read_parquet('./test-00000-of-00001-3f9f7af4f5914b8e.parquet', engine='pyarrow')
test_df.head(10)
```

	text	label
0	Sustainable strategy ‘red lines’ For our susta...	0
1	Verizon’s environmental, health and safety man...	1
2	In 2019, the Company closed a series of transa...	1
3	In December 2020, the AUC approved the Electri...	0
4	Finally, there is a reputational risk linked t...	0
5	Ecoefficiency Eco-efficiency management provid...	1
6	The Group and its customers are exposed to cli...	0
7	Both our Board and executive leadership team r...	1
8	Although it is intended that governments will ...	1
9	Climate-related risks and opportunities have g...	0

```
test_data  = test_df['text'].to_list()
test_label = test_df['label'].to_list()


print("train_label : ", len(train_data))
print("train_label : ",len(train_label))

print("val_label : ", len(val_data))
print("val_label : ",len(val_label))

print("test_data : ",len(test_data))
print("test_label :",len(test_label))
```

```
train_label :  800
train_label :  800
val_label :  200
val_label :  200
test_data :  320
test_label : 320
```

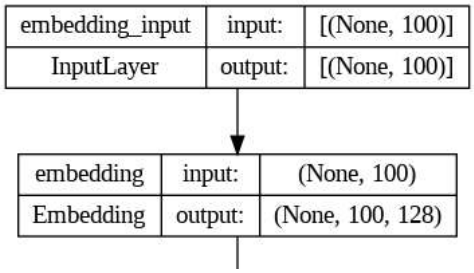
```
# Preprocess & Tokenize
MAX_WORDS = 10000
tokenizer = Tokenizer(num_words=MAX_WORDS)
tokenizer.fit_on_texts(texts = train_data)

train_sequences = tokenizer.texts_to_sequences(train_data)
val_sequences = tokenizer.texts_to_sequences(val_data)
test_sequences = tokenizer.texts_to_sequences(test_data)

train_label = np.array(train_label)
val_label = np.array(val_label)
test_label = np.array(test_label)


# Tokenize
train_data_tokenized = pad_sequences(train_sequences, maxlen = 100)
val_data_tokenized = pad_sequences(val_sequences, maxlen = 100)
test_data_tokenized = pad_sequences(test_sequences, maxlen = 100)

# Cast into numpy array
train_data_tokenized = np.array(train_data_tokenized)
val_data_tokenized = np.array(val_data_tokenized)
test_data_tokenized = np.array(test_data_tokenized)
```


RNN/LSTM MODEL

```
# Define Model
# Hyper parameter sama dengan contoh di slide
model_rnn = Sequential()
model_rnn.add(Embedding(input_dim = MAX_WORDS, output_dim = 128, input_length = train_data_t
model_rnn.add(Bidirectional(LSTM(64, return_sequences=True)))
model_rnn.add(Bidirectional(LSTM(32)))
model_rnn.add(Dense(1,activation='sigmoid'))



#compile model
model_rnn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model_rnn.summary())
print("\n\nModel Visualize")
plot_model(model_rnn, to_file='model_plot.png', show_shapes=True, show_layer_names=True)
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 100, 128)          1280000

 bidirectional (Bidirection  (None, 100, 128)          98816
 al)

 bidirectional_1 (Bidirecti  (None, 64)                41216
 onal)

 dense (Dense)               (None, 1)                 65

=================================================================
Total params: 1420097 (5.42 MB)
Trainable params: 1420097 (5.42 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

Model Visualize




```
# Train
model_rnn.fit(train_data_tokenized, train_label, epochs=10, batch_size=32, validation_data=(\

Epoch 1/10
25/25 [==============================] - 28s 605ms/step - loss: 0.3810 - accuracy: 0.40
Epoch 2/10
25/25 [==============================] - 8s 308ms/step - loss: -0.5538 - accuracy: 0.40
Epoch 3/10
25/25 [==============================] - 12s 474ms/step - loss: -1.5585 - accuracy: 0.4
Epoch 4/10
25/25 [==============================] - 10s 409ms/step - loss: -2.6264 - accuracy: 0.6
Epoch 5/10
```



```
- accuracy: 0.62
25/25 [==============================] - 5s 220ms/step - loss: -3.7078 - accuracy: 0.62
Epoch 7/10
25/25 [==============================] - 6s 222ms/step - loss: -4.2268 - accuracy: 0.59
Epoch 8/10
25/25 [==============================] - 5s 183ms/step - loss: -4.0046 - accuracy: 0.65
Epoch 9/10
25/25 [==============================] - 6s 251ms/step - loss: -2.3637 - accuracy: 0.58
Epoch 10/10
25/25 [==============================] - 5s 188ms/step - loss: -4.4952 - accuracy: 0.66
<keras.src.callbacks.History at 0x79e925fa7160>
```

```
# Evaluate

loss, acc = model_rnn.evaluate(test_data_tokenized, test_label)
print("loss: ", loss)
print("accuracy: ", acc)

10/10 [==============================] - 0s 47ms/step - loss: -1.1683 - accuracy: 0.653
loss:  -1.1683006286621094
accuracy:  0.653124988079071
```

```
# Prediction
prediction = model_rnn.predict(test_data_tokenized[:5])

for text, prediction, groundtruth in zip(tokenizer.sequences_to_texts(test_data_tokenized), p
    sentiment = "positive" if prediction > 0.5 else "negative"
    groundtruth = "positive" if groundtruth == 0.5 else "negative"
    print(f"Text: {text} \n Predicted Sentiment: {sentiment}\n Groundtruth: {groundtruth}\n\n

 1/1 [==============================] - 2s 2s/step
 Text: sustainable strategy for our sustainable strategy range we incorporate a series o
  Predicted Sentiment: positive
  Groundtruth: negative
```

```
Text: environmental health and safety management system provides a framework for identi
 Predicted Sentiment: positive
 Groundtruth: negative


Text: in 2019 the company a series of transactions related to the sale of its canadian
 Predicted Sentiment: positive
 Groundtruth: negative


Text: which would normally come into effect on january 1 2021 for both businesses the r
 Predicted Sentiment: negative
 Groundtruth: negative


Text: finally there is a reputational risk linked to the possibility that oil companies
 Predicted Sentiment: negative
 Groundtruth: negative
```

Word2Vec Embedding

```
from gensim.models import Word2Vec


word2vec_model = Word2Vec(sentences=train_data, vector_size=128, window = 5, min_count=1, sg=

word2vec_model.save("word2vec.model")
```

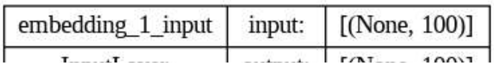
```
WARNING:gensim.models.word2vec:Each 'sentences' item should be a list of words (usually
```

```
embedding_matrix = np.zeros((MAX_WORDS, 128))
for word,i in tokenizer.word_index.items():
    if i < MAX_WORDS:
        if word in word2vec_model.wv:
            embedding_matrix[i] = word2vec_model.wv[word]


# define model
```



```
word2vec_model.add(Embedding(input_dim = MAX_WORDS, output_dim = 128, input_length = train_da
word2vec_model.add(Bidirectional(LSTM(64, return_sequences=True)))
word2vec_model.add(Bidirectional(LSTM(32)))
word2vec_model.add(Dense(1,activation='sigmoid'))


#compile model
word2vec_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(word2vec_model.summary())
print("\n\nModel Visualize")
plot_model(word2vec_model, to_file='model_plot.png', show_shapes=True, show_layer_names=True)
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 100, 128)          1280000

 bidirectional_2 (Bidirecti  (None, 100, 128)          98816
 onal)

 bidirectional_3 (Bidirecti  (None, 64)                41216
 onal)

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 1420097 (5.42 MB)
Trainable params: 1420097 (5.42 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None


Model Visualize
```




```
# Train
word2vec_model.fit(train_data_tokenized, train_label, epochs=10, batch_size=32, validation_da
```

```
Epoch 1/10
25/25 [==============================] - 14s 253ms/step - loss: 0.3516 - accuracy: 0.40
Epoch 2/10
25/25 [==============================] - 7s 273ms/step - loss: -0.5236 - accuracy: 0.43
Epoch 3/10
25/25 [==============================] - 4s 179ms/step - loss: -2.1807 - accuracy: 0.59
Epoch 4/10
25/25 [==============================] - 4s 179ms/step - loss: -3.1236 - accuracy: 0.66
Epoch 5/10
25/25 [==============================] - 7s 292ms/step - loss: -3.9444 - accuracy: 0.68
Epoch 6/10
25/25 [==============================] - 4s 178ms/step - loss: -4.5340 - accuracy: 0.69
Epoch 7/10
25/25 [==============================] - 5s 197ms/step - loss: -4.9905 - accuracy: 0.68
Epoch 8/10
25/25 [==============================] - 6s 244ms/step - loss: -5.4836 - accuracy: 0.68
Epoch 9/10
25/25 [==============================] - 4s 180ms/step - loss: -6.0214 - accuracy: 0.67
Automatic saving failed. This file was updated remotely or in another tab. Show diff - accuracy: 0.69
<keras.src.callbacks.History at 0x79e925284910>
```

```
# Evaluate

loss, acc = word2vec_model.evaluate(test_data_tokenized, test_label)
print("loss: ", loss)
print("accuracy: ", acc)
```

```
10/10 [==============================] - 1s 82ms/step - loss: -2.0320 - accuracy: 0.678
loss:  -2.0319786071777344
accuracy:  0.6781250238418579
```

```
# Prediction
prediction = word2vec_model.predict(test_data_tokenized[:5])

for text, prediction, groundtruth in zip(tokenizer.sequences_to_texts(test_data_tokenized), p
    sentiment = "positive" if prediction > 0.5 else "negative"
    groundtruth = "positive" if groundtruth == 0.5 else "negative"
    print(f"Text: {text} \n Predicted Sentiment: {sentiment}\n Groundtruth: {groundtruth}\n\n
```

```
1/1 [==============================] - 2s 2s/step
Text: sustainable strategy for our sustainable strategy range we incorporate a series o
 Predicted Sentiment: positive
 Groundtruth: negative


Text: environmental health and safety management system provides a framework for identi
 Predicted Sentiment: positive
 Groundtruth: negative


Text: in 2019 the company a series of transactions related to the sale of its canadian
 Predicted Sentiment: positive
 Groundtruth: negative
```

```
Text: which would normally come into effect on january 1 2021 for both businesses the r
 Predicted Sentiment: negative
 Groundtruth: negative


Text: finally there is a reputational risk linked to the possibility that oil companies
 Predicted Sentiment: negative
 Groundtruth: negative
```

Attention Based Model

```
import tensorflow as tf
import torch
from transformers import AutoTokenizer, AutoModelForSequenceClassification
from transformers import Trainer, TrainingArguments

from sklearn.model_selection import train_test_split
from datasets import load_dataset

from transformers import TFAutoModel, AutoTokenizer
import transformers
from tensorflow.keras.layers import Input, Dense, GlobalAveragePooling1D, Attention, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam

import random


if torch.cuda.is_available():
    device = torch.device("cuda")
    print("GPU is available and being used")
else:
    device = torch.device("cpu")
    print("GPU is not available, using CPU instead")


from datasets import load_dataset
from datasets import Dataset, DatasetDict
```



```
parquet', engine=

# dataset_val_pd = pd.read_parquet('./train-00000-of-00001-04b49ae22f595095.parquet', engine=
# dataset_val = Dataset.from_pandas(dataset_val_pd)
dataset_test_pd = pd.read_parquet('./test-00000-of-00001-3f9f7af4f5914b8e.parquet', engine='p
dataset_test = Dataset.from_pandas(dataset_test_pd)
dataset_train_pd
```

	text	label
0	– Scope 3: Optional scope that includes indire...	1
1	The Group is not aware of any noise pollution ...	0
2	Global climate change could exacerbate certain...	0
3	Setting an investment horizon is part and parc...	0
4	Climate change the physical impacts of climate...	0
5	Projects with potential limited adverse social...	0
6	We emitted 13.4 million tonnes CO2 of Scope 2 ...	1
7	We do not provide normalised figures for our C...	1
8	We anticipate that the potential effects of cl...	0
9	Enhancing our responsible screening criteria N...	0

```
dataset_train['label'].nunique()


dataset_test_pd
```

	text	label
0	Sustainable strategy ‘red lines’ For our susta...	0
1	Verizon’s environmental, health and safety man...	1
2	In 2019, the Company closed a series of transa...	1
3	In December 2020, the AUC approved the Electri...	0
4	Finally, there is a reputational risk linked t...	0
...	...	...
315	Indirect emissions result from operational act...	1
316	All data in this TCFD report is as of, or for ...	1
317	Outcome: The bank explained that it would be w...	1
319	Climate change is producing changes in weather...	0

```
import datasets
```

```
labels_train = np.array(dataset_train["label"])
# labels_val = np.array(dataset_val["label"])  # Label is already an array of 0 and 1
labels_test = np.array(dataset_test["label"])  # Label is already an array of 0 and 1


my_dataset = datasets.DatasetDict({"train":dataset_train, "test":dataset_test})


my_dataset
```

```
DatasetDict({
    train: Dataset({
        features: ['text', 'label'],
        num_rows: 10
    })
    test: Dataset({
        features: ['text', 'label'],
        num_rows: 320
    })
})
```

```
tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
tokenized_datasets = my_dataset.map(tokenize_function, batched=True)
```

```
Map:                    10/10
100%                    [00:00<00:00,
                        170.34 examples/s]
```

```
model = AutoModelForSequenceClassification.from_pretrained("bert-base-cased", num_labels=2)
```

```
Some weights of the model checkpoint at bert-base-cased were not used when initializing
- This IS expected if you are initializing BertForSequenceClassification from the check
- This IS NOT expected if you are initializing BertForSequenceClassification from the c
Some weights of BertForSequenceClassification were not initialized from the model check
You should probably TRAIN this model on a down-stream task to be able to use it for pre
```

```
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)


from transformers import TrainingArguments, Trainer
# from transformers import AutoTokenizer, AutoModelForCausalLM
training_args = TrainingArguments(output_dir="test_trainer", evaluation_strategy="epoch")
```

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_datasets['train'],
    eval_dataset=tokenized_datasets['test'],
    compute_metrics=compute_metrics,
)


trainer.train()
```

[5/6 09:57 < 03:19, 0.01 it/s, Epoch 2/3]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	1.082815	0.331250

[34/40 07:13 < 01:18, 0.08 it/s]

[6/6 19:42, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Accuracy

```
# Eval
result = trainer.evaluate()


result


# make prediction

result = trainer.predict(tokenized_datasets['test'])

for i in range(10):
    print(f"Text: {tokenized_datasets['test']['content'][i]}")
    print(f"label: {tokenized_datasets['test']['label'][i]}")
    print(f"Prediction: {np.argmax(result[0][i])} \n")
```

Report

Automatic saving failed. This file was updated remotely or in another tab. Show diff

Performance

- LSTM MODEL

  Training Accuracy : 0.71

  Test Accuracy : 0.70

- Word2Vec Embedding

  Training Accuracy : 0.70

  Test Accuracy : 0.64

- Attention - Based Model

  Training Accuracy : 0.

  Test Accuracy :

Reference :

- https://huggingface.co/distilbert-base-uncased
- https://www.tensorflow.org/text/tutorials/classify_text_with_bert

Double-click (or enter) to edit

Waiting to finish the current execution.

Automatic saving failed. This file was updated remotely or in another tab. Show diff