sentence-compression

March 17, 2024

[1]:
```
!pip3 install opendatasets
```

```
Requirement already satisfied: opendatasets in /usr/local/lib/python3.10/dist-
packages (0.1.22)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from opendatasets) (4.66.2)
Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages
(from opendatasets) (1.5.16)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages
(from opendatasets) (8.1.7)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-
packages (from kaggle->opendatasets) (1.16.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-
packages (from kaggle->opendatasets) (2024.2.2)
Requirement already satisfied: python-dateutil in
/usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from kaggle->opendatasets) (2.31.0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-
packages (from kaggle->opendatasets) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-
packages (from kaggle->opendatasets) (2.0.7)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages
(from kaggle->opendatasets) (6.1.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-
packages (from bleach->kaggle->opendatasets) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in
/usr/local/lib/python3.10/dist-packages (from python-
slugify->kaggle->opendatasets) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->kaggle->opendatasets)
(3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->kaggle->opendatasets) (3.6)
```

[2]:
```
import opendatasets as od
```

[3]:
```
od.download("https://www.kaggle.com/datasets/tongtrantiendung/
 ↪rl-sentence-compression")
```

```
Skipping, found downloaded files in "./rl-sentence-compression" (use force=True
to force download)
```

[4]:
```
import pandas as pd
import numpy as np
import csv

import string
import json

from pathlib import Path
import os

import re
import pickle

import nltk
from nltk.corpus import stopwords

from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline

from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer,␣
 ↪TfidfTransformer

import tensorflow as tf
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Embedding, LSTM, Dense, Input,␣
 ↪GlobalMaxPooling1D, Dropout, SimpleRNN
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.utils import to_categorical
import tensorflow_hub as hub
from tensorflow.keras.preprocessing.text import Tokenizer


from sklearn.metrics import accuracy_score, balanced_accuracy_score
```

[5]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

[5]:
```
True
```

[6]:
```
data_old = pd.read_json('/content/rl-sentence-compression/
 ↪rl-sentence-compression/rl-sentence-compression/data/train-data/gigaword/
 ↪train.jsonl', lines = True)
```

[7]:
```
file = '/content/rl-sentence-compression/rl-sentence-compression/
 ↪rl-sentence-compression/data/train-data/gigaword/train.jsonl'

with open(file,'r') as f:
  train_data=[json.loads(line) for line in f]
```

[8]:
```
data_old
```

[8]:

	id	text	summary
0	gigaword-train-0	australia 's current account deficit shrunk by…	australian current account deficit narrows sha…
1	gigaword-train-1	at least two people were killed in a suspected…	at least two dead in southern philippines blast
2	gigaword-train-2	australian shares closed down #.# percent mond…	australian stocks close down #.# percent
3	gigaword-train-3	south korea 's nuclear envoy kim sook urged no…	envoy urges north korea to restart nuclear dis…
4	gigaword-train-4	south korea on monday announced sweeping tax r…	skorea announces tax cuts to stimulate economy
…	…	…	…
999995	gigaword-train-999995	after proclaiming a special relationship with …	indian leader vajpayee to meet with bush to di…
999996	gigaword-train-999996	a group of people expelled by the british from…	former residents of indian ocean island demand…
999997	gigaword-train-999997	a mix of profit-taking and cautiousness guided…	
999998	gigaword-train-999998	hungary 's air carrier , malev , has grounded …	
999999	gigaword-train-999999	a ##-year-old-girl who struck prince charles i…	

```
999997                        stocks lower in early trading
999998    hungarian air carrier grounds flights to bosnia
999999  teen-ager who struck prince charles with carna…

[1000000 rows x 3 columns]
```

[9]:
```
data = data_old[:1000].copy()
```

[10]:
```
# data['text'] = data['text'].str.lower()
# data['text'] = data['text'].str.replace('[{}]'.format(string.punctuation), '')
# data['text'] = data['text'].str.replace('\n', '')
# data['text'] = data['text'].str.replace('\r', '')

# embed = hub.load("https://tfhub.dev/google/universal-sentence-encoder/4")
# data['text_embedding_X'] = data['text'].apply(lambda x: embed([x])[0])
```

[11]:
```
print(data['text'].apply(lambda x: len(x.split(' '))).sum())
```

```
31153
```

[12]:
```
special_character_remover = re.compile('[/(){}\[\]\|@,;]')
extra_symbol_remover = re.compile('[^0-9a-z #+_]')
STOPWORDS = set(stopwords.words('english'))
```

[13]:
```
def clean_text(text):
  text = text.lower()
  text = special_character_remover.sub(' ', text)
  text = extra_symbol_remover.sub('', text)
  text = ' '.join(word for word in text.split() if word not in STOPWORDS)
  return text

data['text'] = data['text'].apply(clean_text)
```

[14]:
```
print(data['text'].apply(lambda x: len(x.split(' '))).sum())
```

```
19219
```

[15]:
```
data['summary'] = data['summary'].apply(clean_text)
```

[16]:
```
print(data.head(10))
```

```
                id                                               text  \
0  gigaword-train-0  australia current account deficit shrunk recor…
1  gigaword-train-1  least two people killed suspected bomb attack …
2  gigaword-train-2  australian shares closed ## percent monday fol…
3  gigaword-train-3  south korea nuclear envoy kim sook urged north…
4  gigaword-train-4  south korea monday announced sweeping tax refo…
5  gigaword-train-5  taiwan share prices closed ### percent monday …
```

```
6  gigaword-train-6  australian shares closed ## percent monday fol…
7  gigaword-train-7  spanish property group colonial struggling hug…
8  gigaword-train-8  libyan leader moamer kadhafi monday promised w…
9  gigaword-train-9  united nations humanitarian chief john holmes …

                                                summary
0  australian current account deficit narrows sha…
1          least two dead southern philippines blast
2                 australian stocks close ## percent
3  envoy urges north korea restart nuclear disabl…
4        skorea announces tax cuts stimulate economy
5                    taiwan shares close ### percent
6                 australian stocks close ## percent
7          spain colonial posts ### billion euro loss
8   kadhafi promises wide political economic reforms
9    un top aid official arrives droughthit ethiopia
```

[17]:
```
max_train = 0
for i in range(len(data)):
  # print(len(data['text'][i]))
  if len(data['text'][i]) > max_train :
    max_train = len(data['text'][i])


max_target = 0
for i in range(len(data)):
  # print(len(data['text'][i]))
  if len(data['summary'][i]) > max_target :
    max_target = len(data['summary'][i])
```

[18]:
```
max_train, max_target
```

[18]:
```
(208, 64)
```

[19]:
```
# cvect = CountVectorizer()
# sparse_matrix = cvect.fit_transform(data['text'])
# print(sparse_matrix)
```

[20]:
```
# from scipy.sparse import csr_matrix

# tfidf_transformer = TfidfTransformer()

# sparse_matrix = csr_matrix(cvect.fit_transform(data['text']))
# data['text_cvect'] = tfidf_transformer.fit_transform(sparse_matrix)
```

[21]:
```
# pipe = Pipeline([
#     ('vect', CountVectorizer()),
```

```
#     ('tfidf', TfidfTransformer()),
# ])

# data['text_cvect'] = pipe.fit_transform(data['text'])
```

[24]:
```
import gc

class GarbageCollectorCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs=None):
        gc.collect()
```

[25]:
```
token = Tokenizer()
token.fit_on_texts(data['text'] for data in train_data)
```

[26]:
```
train_seq = token.texts_to_sequences(data['text'] for data in train_data)
train_sum = token.texts_to_sequences(data['summary'] for data in train_data)
```

[27]:
```
train_seq = pad_sequences(train_seq, maxlen = 300, padding = 'post')
train_sum = pad_sequences(train_sum, maxlen = 300, padding = 'post')
```

[28]:
```
train_seq
```

[28]:
```
array([[ 241,    9,  907, …,    0,    0,    0],
       [  20,  197,   40, …,    0,    0,    0],
       [ 177,  200,  106, …,    0,    0,    0],
       …,
       [   2, 9233,    3, …,    0,    0,    0],
       [2184,    9,  295, …,    0,    0,    0],
       [   2,   45,  248, …,    0,    0,    0]], dtype=int32)
```

[29]:
```
vocab_size=len(token.word_index)+1
embedding_dim=100
hidden_units=128

rnn=Sequential()
rnn.add(Embedding(vocab_size,300))
rnn.add(SimpleRNN(hidden_units,return_sequences=True))
rnn.add(Dense(vocab_size,activation='softmax'))

rnn.
 ↪compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

[ ]:
```
rnn.fit(train_seq[:1000],train_sum[:1000],epochs=10,batch_size=32)
```

```
Epoch 1/10
```

[ ]:
```
import gc

class GarbageCollectorCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs=None):
        gc.collect()
```

[ ]:
```
lstm=Sequential()
lstm.add(Embedding(vocab_size,300))
lstm.add(LSTM(hidden_units,return_sequences=True))
lstm.add(Dense(vocab_size,activation='softmax'))
lstm.
 ↪compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

[ ]:
```
lstm.fit(train_seq[:1000],train_sum[:1000],epochs=10,batch_size=32)
```