Raaghav_94_NLP7_SentenceCompression

March 17, 2024

1 Sentence Compression

[ ]:
```
import pandas as pd
import numpy as np
import keras
import json
```

```
Using TensorFlow backend.
```

1.1 Import Dependencies

[ ]:
```
train_path = r"/kaggle/input/rl-sentence-compression/rl-sentence-compression/
↪rl-sentence-compression/data/train-data/gigaword/train.jsonl"

def jsonl_to_df(path):
    lines = []
    with open(path) as f:
        lines = f.read().splitlines()

    line_dicts = [json.loads(line) for line in lines]
    return pd.DataFrame(line_dicts)

train_1 = jsonl_to_df(train_path)
```

[ ]:
```
pre = train_1[['text', 'summary']].iloc[:100000]
pre.head()
```

[ ]:
```
                                text  \
0  australia 's current account deficit shrunk by a record #.## billion dollars
-lrb- #.## billion us -rrb- in the june quarter due to soaring commodity prices
, figures released monday showed .
1                                          at least two people were killed in
a suspected bomb attack on a passenger bus in the strife-torn southern
philippines on monday , the military said .
2                                                   australian shares
closed down #.# percent monday following a weak lead from the united states and
lower commodity prices , dealers said .
3                           south korea 's nuclear envoy kim sook urged north
```

```
korea monday to restart work to disable its nuclear plants and stop its ``
typical '' brinkmanship in negotiations .
4             south korea on monday announced sweeping tax reforms , including
income and corporate tax cuts to boost growth by stimulating sluggish private
consumption and business investment .
```

	summary
0	australian current account deficit narrows sharply
1	at least two dead in southern philippines blast
2	australian stocks close down #.# percent
3	envoy urges north korea to restart nuclear disablement
4	skorea announces tax cuts to stimulate economy

1.2 Data Preprocessing

[ ]:
```
text = [str(doc) for doc in pre['text']]
summary = ['_START_ '+ str(doc) + ' _END_' for doc in pre['summary']]
```

[ ]:
```
text[0]
```

[ ]:
```
"australia 's current account deficit shrunk by a record #.## billion dollars
-lrb- #.## billion us -rrb- in the june quarter due to soaring commodity prices
, figures released monday showed ."
```

[ ]:
```
summary[0]
```

[ ]:
```
'_START_ australian current account deficit narrows sharply _END_'
```

[ ]:
```
pre['cleaned_text'] = pd.Series(text)
pre['cleaned_summary'] = pd.Series(summary)
```

[ ]:
```
text_count = []
summary_count = []
```

[ ]:
```
for sent in pre['cleaned_text']:
    text_count.append(len(sent.split()))
for sent in pre['cleaned_summary']:
    summary_count.append(len(sent.split()))
```

[ ]:
```
max_text_len=50
max_summary_len=25
```

[ ]:
```
cleaned_text =np.array(pre['cleaned_text'])
cleaned_summary=np.array(pre['cleaned_summary'])

short_text=[]
short_summary=[]
```

```
for i in range(len(cleaned_text)):
    if(len(cleaned_summary[i].split())<=max_summary_len and len(cleaned_text[i].
 ↪split())<=max_text_len):
        short_text.append(cleaned_text[i])
        short_summary.append(cleaned_summary[i])

post_pre=pd.DataFrame({'text':short_text,'summary':short_summary})
```

[ ]:
```
post_pre.head()
```

[ ]:
```
                                text  \
0  australia 's current account deficit shrunk by a record #.## billion dollars
-lrb- #.## billion us -rrb- in the june quarter due to soaring commodity prices
, figures released monday showed .
1                                              at least two people were killed in
a suspected bomb attack on a passenger bus in the strife-torn southern
philippines on monday , the military said .
2                                                        australian shares
closed down #.# percent monday following a weak lead from the united states and
lower commodity prices , dealers said .
3                            south korea 's nuclear envoy kim sook urged north
korea monday to restart work to disable its nuclear plants and stop its ``
typical '' brinkmanship in negotiations .
4              south korea on monday announced sweeping tax reforms , including
income and corporate tax cuts to boost growth by stimulating sluggish private
consumption and business investment .

                                                summary
0      sostok _START_ australian current account deficit narrows sharply _END_
eostok
1         sostok _START_ at least two dead in southern philippines blast _END_
eostok
2                 sostok _START_ australian stocks close down #.# percent _END_
eostok
3  sostok _START_ envoy urges north korea to restart nuclear disablement _END_
eostok
4          sostok _START_ skorea announces tax cuts to stimulate economy _END_
eostok
```

[ ]:
```
post_pre['summary'] = post_pre['summary'].apply(lambda x : 'sostok '+ x + '␣
 ↪eostok')
```

1.3 Data Splitting and Tokenizing

[ ]:
```
from sklearn.model_selection import train_test_split
x_tr,x_val,y_tr,y_val=train_test_split(np.array(post_pre['text']),np.
  ↪array(post_pre['summary']),test_size=0.1,random_state=0,shuffle=True)
```

[ ]:
```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

x_tokenizer = Tokenizer()
x_tokenizer.fit_on_texts(list(x_tr))
```

[ ]:
```
thresh=4

cnt=0
tot_cnt=0
freq=0
tot_freq=0

for key,value in x_tokenizer.word_counts.items():
    tot_cnt=tot_cnt+1
    tot_freq=tot_freq+value
    if(value<thresh):
        cnt=cnt+1
        freq=freq+value
```

```
% of rare words in vocabulary: 52.61714733388826
Total Coverage of rare words: 1.3035643725717203
```

[ ]:
```
x_tokenizer = Tokenizer(num_words=tot_cnt-cnt)
x_tokenizer.fit_on_texts(list(x_tr))

x_tr_seq    =   x_tokenizer.texts_to_sequences(x_tr)
x_val_seq   =   x_tokenizer.texts_to_sequences(x_val)

x_tr    =   pad_sequences(x_tr_seq,  maxlen=max_text_len, padding='post')
x_val   =   pad_sequences(x_val_seq, maxlen=max_text_len, padding='post')

x_voc   =  x_tokenizer.num_words + 1

print("Size of vocabulary in X = {}".format(x_voc))
```

```
Size of vocabulary in X = 19355
```

[ ]:
```
y_tokenizer = Tokenizer()
y_tokenizer.fit_on_texts(list(y_tr))
```

[ ]:
```
thresh=6

cnt=0
tot_cnt=0
freq=0
tot_freq=0

for key,value in y_tokenizer.word_counts.items():
    tot_cnt=tot_cnt+1
    tot_freq=tot_freq+value
    if(value<thresh):
        cnt=cnt+1
        freq=freq+value
```

```
% of rare words in vocabulary: 64.36004373000769
Total Coverage of rare words: 2.9634977047238267
```

[ ]:
```
y_tokenizer = Tokenizer(num_words=tot_cnt-cnt)
y_tokenizer.fit_on_texts(list(y_tr))

y_tr_seq    =   y_tokenizer.texts_to_sequences(y_tr)
y_val_seq   =   y_tokenizer.texts_to_sequences(y_val)

y_tr    =   pad_sequences(y_tr_seq, maxlen=max_summary_len, padding='post')
y_val   =   pad_sequences(y_val_seq, maxlen=max_summary_len, padding='post')

y_voc  =   y_tokenizer.num_words +1
print("Size of vocabulary in Y = {}".format(y_voc))
```

```
Size of vocabulary in Y = 8803
```

[ ]:
```
ind=[]
for i in range(len(y_tr)):
    cnt=0
    for j in y_tr[i]:
        if j!=0:
            cnt=cnt+1
    if(cnt==2):
        ind.append(i)

y_tr=np.delete(y_tr,ind, axis=0)
x_tr=np.delete(x_tr,ind, axis=0)
```

[ ]:
```
ind=[]
for i in range(len(y_val)):
    cnt=0
    for j in y_val[i]:
        if j!=0:
```

```
            cnt=cnt+1
    if(cnt==2):
        ind.append(i)

y_val=np.delete(y_val,ind, axis=0)
x_val=np.delete(x_val,ind, axis=0)
```

1.4 Model Building

[ ]:
```
from keras import backend as K
import gensim
from numpy import *
import numpy as np
import pandas as pd
import re
from bs4 import BeautifulSoup
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from nltk.corpus import stopwords
from tensorflow.keras.layers import Input, LSTM, Embedding, Dense, Concatenate,␣
 ↪TimeDistributed
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import EarlyStopping
import warnings
pd.set_option("display.max_colwidth", 200)
warnings.filterwarnings("ignore")

K.clear_session()

latent_dim = 300
embedding_dim=200

# Encoder
encoder_inputs = Input(shape=(max_text_len,))

enc_emb =  Embedding(x_voc, embedding_dim,trainable=True)(encoder_inputs)

encoder_lstm1 =␣
 ↪LSTM(latent_dim,return_sequences=True,return_state=True,dropout=0.
 ↪4,recurrent_dropout=0.4)
encoder_output1, state_h1, state_c1 = encoder_lstm1(enc_emb)

encoder_lstm2 =␣
 ↪LSTM(latent_dim,return_sequences=True,return_state=True,dropout=0.
 ↪4,recurrent_dropout=0.4)
encoder_output2, state_h2, state_c2 = encoder_lstm2(encoder_output1)
```

```
encoder_lstm3=LSTM(latent_dim, return_state=True,␣
 ↪return_sequences=True,dropout=0.4,recurrent_dropout=0.4)
encoder_outputs, state_h, state_c= encoder_lstm3(encoder_output2)

decoder_inputs = Input(shape=(None,))

dec_emb_layer = Embedding(y_voc, embedding_dim,trainable=True)
dec_emb = dec_emb_layer(decoder_inputs)

decoder_lstm = LSTM(latent_dim, return_sequences=True,␣
 ↪return_state=True,dropout=0.4,recurrent_dropout=0.2)
decoder_outputs,decoder_fwd_state, decoder_back_state =␣
 ↪decoder_lstm(dec_emb,initial_state=[state_h, state_c])

decoder_dense =  TimeDistributed(Dense(y_voc, activation='softmax'))
decoder_outputs = decoder_dense(decoder_outputs)

model = Model([encoder_inputs, decoder_inputs], decoder_outputs)

model.summary()
```

```
Size of vocabulary from the w2v model = 19355
Model: "model"
__________________________________________________________________________________
__________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================
==================
input_1 (InputLayer)            [(None, 50)]         0
__________________________________________________________________________________
__________________
embedding (Embedding)           (None, 50, 200)      3871000     input_1[0][0]
__________________________________________________________________________________
__________________
lstm (LSTM)                     [(None, 50, 300), (N 601200      embedding[0][0]
__________________________________________________________________________________
__________________
input_2 (InputLayer)            [(None, None)]       0
__________________________________________________________________________________
__________________
lstm_1 (LSTM)                   [(None, 50, 300), (N 721200      lstm[0][0]
__________________________________________________________________________________
__________________
embedding_1 (Embedding)         (None, None, 200)    1760600     input_2[0][0]
__________________________________________________________________________________
__________________
lstm_2 (LSTM)                   [(None, 50, 300), (N 721200      lstm_1[0][0]
```

```
__________________________________________________________________________________
__________________
lstm_3 (LSTM)                   [(None, None, 300),  601200
embedding_1[0][0]
                                                                 lstm_2[0][1]
                                                                 lstm_2[0][2]
__________________________________________________________________________________
__________________
time_distributed (TimeDistribut (None, None, 8803)   2649703     lstm_3[0][0]
==================================================================================
==================
Total params: 10,926,103
Trainable params: 10,926,103
Non-trainable params: 0
__________________________________________________________________________________
__________________
```

[ ]:
```
model.compile(optimizer='rmsprop', loss='sparse_categorical_crossentropy')
```

[ ]:
```
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1,patience=2)
```

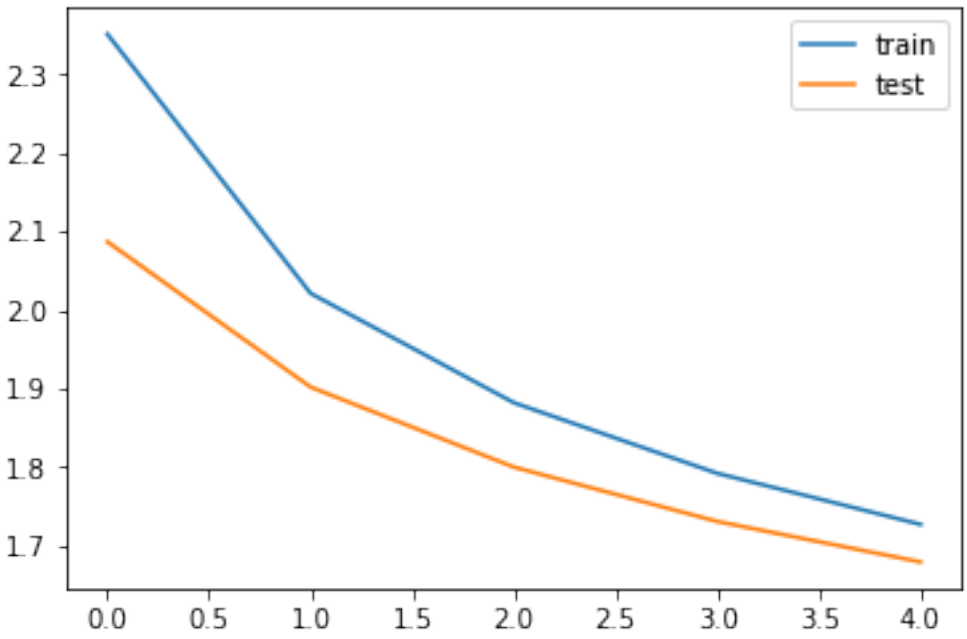
Start fitting the model with the data

[ ]:
```
history=model.fit([x_tr,y_tr[:,:-1]], y_tr.reshape(y_tr.shape[0],y_tr.shape[1],␣
↪1)[:,1:] ,epochs=5,callbacks=[es],batch_size=128,␣
↪validation_data=([x_val,y_val[:,:-1]], y_val.reshape(y_val.shape[0],y_val.
↪shape[1], 1)[:,1:]))
```

```
Train on 89710 samples, validate on 9968 samples
Epoch 1/5
89710/89710 [==============================] - 204s 2ms/sample - loss: 2.3516 -
val_loss: 2.0868
Epoch 2/5
89710/89710 [==============================] - 200s 2ms/sample - loss: 2.0210 -
val_loss: 1.9015
Epoch 3/5
89710/89710 [==============================] - 199s 2ms/sample - loss: 1.8815 -
val_loss: 1.7995
Epoch 4/5
89710/89710 [==============================] - 199s 2ms/sample - loss: 1.7915 -
val_loss: 1.7302
Epoch 5/5
89710/89710 [==============================] - 199s 2ms/sample - loss: 1.7264 -
val_loss: 1.6786
```

1.5 Visualizing the Loss

[ ]:
```
from matplotlib import pyplot
pyplot.plot(history.history['loss'], label='train')
pyplot.plot(history.history['val_loss'], label='test')
pyplot.legend()
pyplot.show()
```



1.6 Converting Index to Word

[ ]:
```
reverse_target_word_index=y_tokenizer.index_word
reverse_source_word_index=x_tokenizer.index_word
target_word_index=y_tokenizer.word_index
```

[ ]:
```
encoder_model = Model(inputs=encoder_inputs,outputs=[encoder_outputs, state_h,␣
 ↪state_c])

decoder_state_input_h = Input(shape=(latent_dim,))
decoder_state_input_c = Input(shape=(latent_dim,))
decoder_hidden_state_input = Input(shape=(max_text_len,latent_dim))

dec_emb2= dec_emb_layer(decoder_inputs)
decoder_outputs2, state_h2, state_c2 = decoder_lstm(dec_emb2,␣
 ↪initial_state=[decoder_state_input_h, decoder_state_input_c])
```

```
decoder_outputs2 = decoder_dense(decoder_outputs2)

decoder_model = Model(
    [decoder_inputs] + [decoder_hidden_state_input,decoder_state_input_h,␣
 ↪decoder_state_input_c],
    [decoder_outputs2] + [state_h2, state_c2])
```

1.7 Helper Functions

[ ]:
```
def decode_sequence(input_seq):
    e_out, e_h, e_c = encoder_model.predict(input_seq)

    target_seq = np.zeros((1,1))

    target_seq[0, 0] = target_word_index['sostok']

    stop_condition = False
    decoded_sentence = ''
    while not stop_condition:

        output_tokens, h, c = decoder_model.predict([target_seq] + [e_out, e_h,␣
 ↪e_c])

        sampled_token_index = np.argmax(output_tokens[0, -1, :])
        sampled_token = reverse_target_word_index[sampled_token_index]

        if(sampled_token!='eostok'):
            decoded_sentence += ' '+sampled_token

        if (sampled_token == 'eostok'  or len(decoded_sentence.split()) >=␣
 ↪(max_summary_len-1)):
            stop_condition = True

        target_seq = np.zeros((1,1))
        target_seq[0, 0] = sampled_token_index

        e_h, e_c = h, c

    return decoded_sentence
```

[ ]:
```
def seq2summary(input_seq):
    newString=''
    for i in input_seq:
        if((i!=0 and i!=target_word_index['sostok']) and i!
 ↪=target_word_index['eostok']):
            newString=newString+reverse_target_word_index[i]+' '
    return newString
```

```
def seq2text(input_seq):
    newString=''
    for i in input_seq:
        if(i!=0):
            newString=newString+reverse_source_word_index[i]+' '
    return newString
```

1.8 Results

[ ]:
```
for i in range(0,15):
    print("Review:",seq2text(x_tr[i]))
    print("Original summary:",seq2summary(y_tr[i])[6:-4])
    print("Predicted summary:",decode_sequence(x_tr[i].
 ↪reshape(1,max_text_len))[6:-4])
    print("\n")
```

```
Review: the european commission on wednesday ordered its civil servants to cut
down on unk documents to head off a looming translation crisis in the unk eu
which now has official languages
Original summary: ordered to cut output to prevent crisis
Predicted summary:  eu to launch eu 's largest bank of aids


Review: scorecard at the close of the second day in the four day cricket tour
match between new south wales and the west indies here friday
Original summary: new south wales v west indies cricket scorecard
Predicted summary:  south africa v england scoreboard


Review: the incoming speaker of the house of representatives said sunday the
united nations had proven itself an incompetent body and said the united states
should radically alter its involvement it
Original summary: us house leader blasts un on bosnia
Predicted summary:  us to send to un force in bosnia


Review: us president george w bush will host italian prime minister silvio
berlusconi on october white house spokeswoman dana perino said in a statement
Original summary: bush berlusconi to meet october
Predicted summary:  bush to meet with eu president


Review: a us airways pilots ' group represented by the air line pilots
association international said thursday it had agreed to an percent pay in order
to help the struggling carrier emerge from bankruptcy protection
Original summary: us airways pilots agree to pay cuts in bid to shake off
bankruptcy
```

```
Predicted summary:  us airlines to build million dollars in aid


Review: thriller changing lanes '' cut in front of panic room '' to take the top
spot at the north american box office last weekend official figures showed
monday
Original summary: changing cuts off panic room at north american box office
Predicted summary:  toyota to launch first time in south korea


Review: the governor of the bank of england eddie george and chancellor of the
exchequer kenneth clarke disagreed about pressure for an increase in interest
rates at their monthly meeting on november the minutes showed on wednesday
Original summary: clarke and george on british rate outlook
Predicted summary:  new zealand 's economy in


Review: china made sweeping changes to its leadership thursday as president
jiang zemin and almost all the country 's other communist bosses began handing
over power to a younger generation
Original summary: jiang steps down as chinese communist party changes generation
Predicted summary:  china 's jiang to visit china


Review: commonwealth leaders resolved tuesday to renew their fight against aids
which has ravaged many of the nation body 's members
Original summary: commonwealth pledges renewed offensive against aids
Predicted summary:  new zealand to visit to fight


Review: british foreign secretary malcolm rifkind said tuesday he was delighted
'' at the release of two french pilots by the bosnian serbs after three months
in captivity
Original summary: britain delighted at release of french pilots
Predicted summary:  britain 's first term for iraq


Review: thailand 's ambassadors to six major trading partners have been ordered
to promote the kingdom effectively in a modern unk '' or face removal reports
and officials said tuesday
Original summary: thai premier tells ceo ambassadors to perform or face removal
Predicted summary:  thai government to be in


Review: the un mission sacked five bosnian policemen for their involvement in
atrocities committed during the war a un spokesman said here tuesday
Original summary: five bosnian policemen sacked for wartime atrocities
Predicted summary:  un envoy to visit afghanistan
```

```
Review: liverpool goalkeeper jerzy has revealed the english giants have a very
special fan pope john paul ii
Original summary: pope support for liverpool
Predicted summary:  ferguson 's unk


Review: one of egypt 's greatest ever footballers saleh has died aged from
cancer in a london hospital press reports said on monday
Original summary: unk dead at
Predicted summary:  former jackson dies in hospital


Review: too many old men and no women have been appointed to conduct a review of
the pacific 's main regional political body a non government organization said
tuesday
Original summary: too many old men reviewing pacific regional body says lobby
group
Predicted summary:  unk to be in the
```