Grammatical Error Correction

In [5]:
```
# !pip install -q datasets tqdm pandas
# !pip install -q sentencepiece
# !pip install -q transformers
# !pip install -q wandb
#!pip install --user -U nltk
#!pip3 install datasets
#!pip install wandb
```

In [4]:
```
import argparse
import glob
import os
import json
import time
import logging
import random
import re
from itertools import chain
from string import punctuation
import warnings
warnings.filterwarnings('ignore')
import nltk
#nltk.download('punkt')
from nltk.tokenize import sent_tokenize
import numpy as np
import pandas as pd
from tqdm import tqdm
import numpy as np
import tensorflow_datasets
import torch
from torch.utils.data import Dataset, DataLoader
#import datasets
from transformers import AdamW,get_linear_schedule_with_warmup
from transformers import T5ForConditionalGeneration, T5Tokenizer
from datasets import load_metric
from transformers import Seq2SeqTrainingArguments, Seq2SeqTrainer, DataCollatorForSeq2Seq
from torch.utils.data import Dataset, DataLoader
from sklearn.model_selection import train_test_split
```

In [5]:
```
#os.environ["WANDB_DISABLED"] = "true"
```

In [7]:
```
pd.set_option('display.max_colwidth', None)
```

In [8]:
```
len_df = 18386520  # C4_200M.tsv-00000-of-00010
start =  1838651 # 919325 +
end = int(18386520/20)
```

In [9]:
```
start = end + 1
end = start + 300000
```

In [10]:
```
print(start, end, end - start)
```

```
919327 1219327 300000
```

In [11]:
```
file_name='C4_200M.tsv-00000-of-00010'
df_main = pd.read_csv(file_name, delimiter='\t', skiprows=start, nrows=end) # on_bad_lines='skip
df_main.dropna(inplace=True)
df_main.head()
```

Out[11]:

	Korean translation agency based in London, The United Kingdom offers services in addition by official Korean translators.	Korean translation agency based in London, United Kingdom offering services by official Korean translators.
0	Stab that badboy on with a stick!!	Stab that badboy with a stick!!
1	What I did for the matrics to finish at this univercity is quite a clear example of what I described in this previous post, precisely the difficulties which the Italian citizens on a daily basis - when they have to deal with the public administration.	What I did to complete the matriculation at this University is a clear example of what I described in this previous post, namely the difficulties which the Italian citizens experience on a daily basis when they have to deal with the public administration.
2	The campaign was a big test for the newly appointed Regional Commissioner (RC) as president-Magufuli and prime minister Kassim Majaliwa insisting that school desks campaign was the RCs' factor when appointed.	The campaign was a big test to the newly appointed Regional Commissioners (RC's) as President Magufuli and Prime Minister Kassim Majaliwa were insisting that the school desks campaign was one of the RCs' performance ratings after being appointed.
3	Then he looed up canister, then found that it was a box for holding teas 33387 and when he turned to tea he discovered it was sometimes made of beef, and beef was meat and meat is what human being composed of; and canister was, therefore, a box for taking meat.	Then he looked up canister, and found that it was a box for holding tea; and when he turned to tea he discovered it was sometimes made of beef, and beef was meat, and meat is what human beings are composed of; and canister was, therefore, a box for containing meat.
4	Super-Quad CLASSIC+ MM - all the classic features of our prestigious Super-Quad pickups in their traditional solid carbon fibre housing, now with Music Man string spacing and poles piece.	Super-Quad CLASSIC+ MM - all the classic features of our renowned Super-Quad pickups in their traditional solid carbon fibre housing, now with Music Man string spacing and pole pieces.

In [12]:
```
df_main.columns = ["input", "target"]
df=df_main.iloc[start:end]
```

In [14]:
```
model_name = 't5-base'
tokenizer = T5Tokenizer.from_pretrained(model_name)
model = T5ForConditionalGeneration.from_pretrained(model_name)
```

```
/usr/local/lib/python3.10/dist-packages/transformers/models/t5/tokenization_t5.py:164: FutureWarnin
g: This tokenizer was incorrectly instantiated with a model max length of 512 which will be correcte
d in Transformers v5.
For now, this behavior is kept to avoid breaking backwards compatibility when padding/encoding with
`truncation is True`.
- Be aware that you SHOULD NOT rely on t5-base automatically truncating your input to 512 when paddi
ng/encoding.
- If you want to encode/pad to sequences longer than 512 you can either instantiate this tokenizer w
ith `model_max_length` or pass `max_length` when encoding/padding.
- To avoid this warning, please instantiate this tokenizer with `model_max_length` set to your prefe
rred value.
  warnings.warn(
```

In [15]:
```
def calc_token_len(example):
    return len(tokenizer(example).input_ids)
```

In [16]:
```
train_df, test_df = train_test_split(df, test_size=0.10, shuffle=True)
train_df.shape, test_df.shape
```

Out[16]:
```
((269997, 2), (30000, 2))
```

In [17]:
```
test_df['input_token_len'] = test_df['input'].apply(calc_token_len)
```

```
Token indices sequence length is longer than the specified maximum sequence length for this model (9
11 > 512). Running this sequence through the model will result in indexing errors
/tmp/ipykernel_15226/2133573097.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexi
ng.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/inde
xing.html#returning-a-view-versus-a-copy)
  test_df['input_token_len'] = test_df['input'].apply(calc_token_len)
```

In [18]:
```
test_df.head()
```

Out[18]:

	input	target	input_token_len
924269	Add the parsley (you can use the stems too, as long as you thinly chop them) to the maison.	Add the parsley (you can use the stems too, as long as you thinly chop them) to the bowl too.	27
971103	ICE"s 2009 report noted that despite the rapid growth of immigration detention genrally, number of convicted criminals located and detained its people barely increased.	ICE"s 2009 report noted that despite the rapid growth of immigration detention generally, the number of convicted criminals located and detained had barely increased.	38
1116905	This link with BUILA proved invaluable with the University hosting BUILA's annual conference over two consecutive years where Bobby had been first 'enrolled' as a USW Conference Ambassador!	This link with BUILA proved invaluable with the University hosting BUILA's annual conference over two consecutive years where Bobby was first 'enrolled' as a USW Conference Ambassador!	40
1218648	We focused and committed to making individuals improved lives or personal performance in a specific areas like stress management, starting up a small business and helping it grow time management, health coaching, personal relationships or other areas.	We focused and committed to helping individuals improve their lives or personal performance in a specific areas like stress management, starting up a small business and helping it grow, time management, health coaching, personal relationships or other areas.	44
1003253	Watch this video - our client fan about her perfect appointment to prestigious position - we pair it up with her Press Release.	Watch this video - our client talking about her appointment to a prestigious position - we paired it with her Press Release.	28

In [19]:
```
test_df['input_token_len'].describe()
```

Out[19]:
```
count    30000.000000
mean        33.849933
std         26.730257
min          3.000000
25%         17.000000
50%         27.000000
75%         42.000000
max        917.000000
Name: input_token_len, dtype: float64
```

In [20]:
```
train_dataset = Dataset.from_pandas(train_df)
test_dataset = Dataset.from_pandas(test_df)
```

In [21]:
```
test_dataset
```

Out[21]:
```
Dataset({
    features: ['input', 'target', 'input_token_len', '__index_level_0__'],
    num_rows: 30000
})
```

In [22]:
```
class GrammarDataset(Dataset):
    def __init__(self, dataset, tokenizer,print_text=False):
        self.dataset = dataset
        self.pad_to_max_length = False
        self.tokenizer = tokenizer
        self.print_text = print_text
        self.max_len = 64

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


    def tokenize_data(self, example):
        input_, target_ = example['input'], example['target'] # output

        # tokenize inputs
        tokenized_inputs = tokenizer(input_, pad_to_max_length=self.pad_to_max_length,
                                    max_length=self.max_len,
                                    return_attention_mask=True)

        tokenized_targets = tokenizer(target_, pad_to_max_length=self.pad_to_max_length,
                                    max_length=self.max_len,
                                    return_attention_mask=True)

        inputs={"input_ids": tokenized_inputs['input_ids'],
            "attention_mask": tokenized_inputs['attention_mask'],
            "labels": tokenized_targets['input_ids']
        }

        return inputs


    def __getitem__(self, index):
        inputs = self.tokenize_data(self.dataset[index])

        if self.print_text:
            for k in inputs.keys():
                print(k, len(inputs[k]))

        return inputs
```

In [23]:
```
dataset = GrammarDataset(test_dataset, tokenizer, True)
print(dataset[121])
```

```
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `t
runcation=True` to explicitly truncate examples to max length. Defaulting to 'longest_first' truncat
ion strategy. If you encode pairs of sequences (GLUE-style) with the tokenizer you can select this s
trategy more precisely by providing a specific strategy to `truncation`.

input_ids 40
attention_mask 40
labels 38
{'input_ids': [11107, 6, 16, 131, 7643, 767, 16, 828, 6, 13346, 12524, 5073, 23, 141, 131, 263, 231
4, 8305, 21, 31786, 6, 17029, 11, 3411, 2348, 57, 8, 907, 1323, 6, 27274, 6, 52, 2051, 6, 11, 377, 1
7279, 5, 1], 'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], 'labels': [11107, 6, 16, 131, 7643, 767, 16, 82
8, 6, 13346, 12524, 5073, 23, 65, 263, 2314, 8305, 12, 31786, 6, 17029, 11, 3411, 2348, 57, 8, 907,
1323, 6, 27274, 6, 2051, 6, 11, 377, 17279, 5, 1]}
```

In [25]:
```
rouge_metric = load_metric("rouge")
```

```
/tmp/ipykernel_15226/2048908469.py:2: FutureWarning: load_metric is deprecated and will be removed i
n the next major version of datasets. Use 'evaluate.load' instead, from the new library 🤗 Evaluate:
https://huggingface.co/docs/evaluate (https://huggingface.co/docs/evaluate)
  rouge_metric = load_metric("rouge")
```

In [27]:
```
data_collator = DataCollatorForSeq2Seq(tokenizer, model=model, padding='longest', return_tensors='pt'
```

In [28]:
```
batch_size = 16
args = Seq2SeqTrainingArguments(
                        output_dir="./kaggle/working/c4_200m/weights",
                        evaluation_strategy="steps",
                        per_device_train_batch_size=batch_size,
                        per_device_eval_batch_size=batch_size,
                        learning_rate=2e-5,
                        num_train_epochs=1,
                        weight_decay=0.01,
                        save_total_limit=2,
                        predict_with_generate=True,
                        #fp16 = True, # only while using CUDA
                        gradient_accumulation_steps = 6,
                        eval_steps = 500,
                        save_steps = 500,
                        load_best_model_at_end=True,
                        logging_dir="./logs",
                        report_to=None
                        #report_to="wandb", # report  to wandb
    )
```

```
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --r
eport_to flag to control the integrations used for logging result (for instance --report_to none).
```

In [5]:
```
def compute_metrics(eval_pred):
    predictions, labels = eval_pred
    decoded_preds = tokenizer.batch_decode(predictions, skip_special_tokens=True)
    labels = np.where(labels != -100, labels, tokenizer.pad_token_id)
    decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens=True)

    decoded_preds = ["\n".join(nltk.sent_tokenize(pred.strip())) for pred in decoded_preds]
    decoded_labels = ["\n".join(nltk.sent_tokenize(label.strip())) for label in decoded_labels]

    result = rouge_metric.compute(predictions=decoded_preds, references=decoded_labels, use_stemmer=T
    # Extract a few results
    result = {key: value.mid.fmeasure * 100 for key, value in result.items()}

    # Add mean generated length
    prediction_lens = [np.count_nonzero(pred != tokenizer.pad_token_id) for pred in predictions]
    result["gen_len"] = np.mean(prediction_lens)
    return {k: round(v, 4) for k, v in result.items()}
```

In [30]:
```
trainer = Seq2SeqTrainer(model=model,
                args=args,
                train_dataset= GrammarDataset(train_dataset, tokenizer),
                eval_dataset=GrammarDataset(test_dataset, tokenizer),
                tokenizer=tokenizer,
                data_collator=data_collator,
                compute_metrics=compute_metrics)
```

```
In [31]: trainer.train()
```

```
/usr/local/lib/python3.10/dist-packages/transformers/optimization.py:306: FutureWarning: This implem
entation of AdamW is deprecated and will be removed in a future version. Use the PyTorch implementat
ion torch.optim.AdamW instead, or set `no_deprecation_warning=True` to disable this warning
  warnings.warn(
***** Running training *****
  Num examples = 269997
  Num Epochs = 1
  Instantaneous batch size per device = 16
  Total train batch size (w. parallel, distributed & accumulation) = 96
  Gradient Accumulation steps = 6
  Total optimization steps = 2812
  Number of trainable parameters = 222903552
```

[2812/2812 2:47:59, Epoch 0/1]

Step	Training Loss	Validation Loss	Rouge1	Rouge2	Rougel	Rougelsum	Gen Len
500	0.763400	0.628792	71.224700	60.843600	70.482700	70.513800	17.331700
1000	0.678100	0.603869	71.444600	61.254500	70.707600	70.743500	17.316600
1500	0.656000	0.591053	71.598400	61.518200	70.867500	70.903300	17.300400
2000	0.644400	0.585335	71.653500	61.625700	70.919300	70.954400	17.299000
2500	0.637800	0.582086	71.696400	61.697600	70.964500	71.000800	17.296600

```
***** Running Evaluation *****
  Num examples = 30000
  Batch size = 16
Saving model checkpoint to ./kaggle/working/c4_200m/weights/checkpoint-500
Configuration saved in ./kaggle/working/c4_200m/weights/checkpoint-500/config.json
Model weights saved in ./kaggle/working/c4_200m/weights/checkpoint-500/pytorch_model.bin
tokenizer config file saved in ./kaggle/working/c4_200m/weights/checkpoint-500/tokenizer_config.json
Special tokens file saved in ./kaggle/working/c4_200m/weights/checkpoint-500/special_tokens_map.json
Deleting older checkpoint [kaggle/working/c4_200m/weights/checkpoint-1000] due to args.save_total_li
mit
***** Running Evaluation *****
  Num examples = 30000
  Batch size = 16
Saving model checkpoint to ./kaggle/working/c4_200m/weights/checkpoint-1000
Configuration saved in ./kaggle/working/c4_200m/weights/checkpoint-1000/config.json
Model weights saved in ./kaggle/working/c4_200m/weights/checkpoint-1000/pytorch_model.bin
tokenizer config file saved in ./kaggle/working/c4_200m/weights/checkpoint-1000/tokenizer_config.jso
n
Special tokens file saved in ./kaggle/working/c4_200m/weights/checkpoint-1000/special_tokens_map.jso
n
Deleting older checkpoint [kaggle/working/c4_200m/weights/checkpoint-1500] due to args.save_total_li
mit
***** Running Evaluation *****
  Num examples = 30000
  Batch size = 16
Saving model checkpoint to ./kaggle/working/c4_200m/weights/checkpoint-1500
Configuration saved in ./kaggle/working/c4_200m/weights/checkpoint-1500/config.json
Model weights saved in ./kaggle/working/c4_200m/weights/checkpoint-1500/pytorch_model.bin
tokenizer config file saved in ./kaggle/working/c4_200m/weights/checkpoint-1500/tokenizer_config.jso
n
Special tokens file saved in ./kaggle/working/c4_200m/weights/checkpoint-1500/special_tokens_map.jso
n
Deleting older checkpoint [kaggle/working/c4_200m/weights/checkpoint-500] due to args.save_total_lim
it
***** Running Evaluation *****
  Num examples = 30000
  Batch size = 16
Saving model checkpoint to ./kaggle/working/c4_200m/weights/checkpoint-2000
Configuration saved in ./kaggle/working/c4_200m/weights/checkpoint-2000/config.json
Model weights saved in ./kaggle/working/c4_200m/weights/checkpoint-2000/pytorch_model.bin
tokenizer config file saved in ./kaggle/working/c4_200m/weights/checkpoint-2000/tokenizer_config.jso
n
Special tokens file saved in ./kaggle/working/c4_200m/weights/checkpoint-2000/special_tokens_map.jso
n
Deleting older checkpoint [kaggle/working/c4_200m/weights/checkpoint-1000] due to args.save_total_li
mit
***** Running Evaluation *****
  Num examples = 30000
  Batch size = 16
Saving model checkpoint to ./kaggle/working/c4_200m/weights/checkpoint-2500
Configuration saved in ./kaggle/working/c4_200m/weights/checkpoint-2500/config.json
Model weights saved in ./kaggle/working/c4_200m/weights/checkpoint-2500/pytorch_model.bin
tokenizer config file saved in ./kaggle/working/c4_200m/weights/checkpoint-2500/tokenizer_config.jso
n
Special tokens file saved in ./kaggle/working/c4_200m/weights/checkpoint-2500/special_tokens_map.jso
n
Deleting older checkpoint [kaggle/working/c4_200m/weights/checkpoint-1500] due to args.save_total_li
mit


Training completed. Do not forget to share your model on huggingface.co/models =)


Loading best model from ./kaggle/working/c4_200m/weights/checkpoint-2500 (score: 0.582085549831390
4).
```

```
Out[31]: TrainOutput(global_step=2812, training_loss=0.6711784297677226, metrics={'train_runtime': 10081.218
5, 'train_samples_per_second': 26.782, 'train_steps_per_second': 0.279, 'total_flos': 1.986856424497
152e+16, 'train_loss': 0.6711784297677226, 'epoch': 1.0})
```

In [32]:
```
model_name='t5_gec_model_03'
trainer.save_model(model_name)
```

```
Saving model checkpoint to t5_gec_model_03
Configuration saved in t5_gec_model_03/config.json
Model weights saved in t5_gec_model_03/pytorch_model.bin
tokenizer config file saved in t5_gec_model_03/tokenizer_config.json
Special tokens file saved in t5_gec_model_03/special_tokens_map.json
```

In [34]:
```
model_name = 't5_gec_model_03'
torch_device = 'cuda' if torch.cuda.is_available() else 'cpu'
tokenizer = T5Tokenizer.from_pretrained(model_name)
model = T5ForConditionalGeneration.from_pretrained(model_name).to(torch_device)

def correct_grammar(input_text,num_return_sequences):
    batch = tokenizer([input_text],truncation=True,padding='max_length',max_length=64, return_tensors
    translated = model.generate(**batch,max_length=64,num_beams=4, num_return_sequences=num_return_se
    tgt_text = tokenizer.batch_decode(translated, skip_special_tokens=True)
    return tgt_text
```

```
loading file spiece.model
loading file added_tokens.json
loading file special_tokens_map.json
loading file tokenizer_config.json
loading configuration file t5_gec_model_03/config.json
Model config T5Config {
  "_name_or_path": "t5-base",
  "architectures": [
    "T5ForConditionalGeneration"
  ],
  "d_ff": 3072,
  "d_kv": 64,
  "d_model": 768,
  "decoder_start_token_id": 0,
  "dense_act_fn": "relu",
  "dropout_rate": 0.1,
  "eos_token_id": 1,
  "feed_forward_proj": "relu",
  "initializer_factor": 1.0,
  "is_encoder_decoder": true,
  "is_gated_act": false,
  "layer_norm_epsilon": 1e-06,
  "model_type": "t5",
  "n_positions": 512,
  "num_decoder_layers": 12,
  "num_heads": 12,
  "num_layers": 12,
  "output_past": true,
  "pad_token_id": 0,
  "relative_attention_max_distance": 128,
  "relative_attention_num_buckets": 32,
  "task_specific_params": {
    "summarization": {
      "early_stopping": true,
      "length_penalty": 2.0,
      "max_length": 200,
      "min_length": 30,
      "no_repeat_ngram_size": 3,
      "num_beams": 4,
      "prefix": "summarize: "
    },
    "translation_en_to_de": {
      "early_stopping": true,
      "max_length": 300,
      "num_beams": 4,
      "prefix": "translate English to German: "
    },
    "translation_en_to_fr": {
      "early_stopping": true,
      "max_length": 300,
      "num_beams": 4,
      "prefix": "translate English to French: "
    },
    "translation_en_to_ro": {
      "early_stopping": true,
      "max_length": 300,
      "num_beams": 4,
      "prefix": "translate English to Romanian: "
    }
  },
  "torch_dtype": "float32",
  "transformers_version": "4.24.0",
  "use_cache": true,
  "vocab_size": 32128
}

loading weights file t5_gec_model_03/pytorch_model.bin
All model checkpoint weights were used when initializing T5ForConditionalGeneration.

All the weights of T5ForConditionalGeneration were initialized from the model checkpoint at t5_gec_m
odel_03.
If your task is similar to the task the model of the checkpoint was trained on, you can already use
T5ForConditionalGeneration for predictions without further training.
```

In [35]:
```
input_text = "I am enjoys, writtings Articles ons AI and I also enjoyed write articling on AI."
num_return_sequences = 1
corrected_text = correct_grammar(input_text, num_return_sequences)
print(corrected_text)
```

```
['I enjoy writing articles on AI and I also enjoy writing articles on AI.']
```

In [36]:
```
text =  """Today gift shows are popular in many countries, and purpose of these shows finds talented

Firstly, result this programme has  a massive effect on the society, because many people get a chance

secondly, many audiences, and viewers watch this shows, so it is a big chance for companies by sponso

As a result, the aim of  producing this shows impressive, so part of the society following this shows

"""
print(correct_grammar(text, num_return_sequences))
```

```
['Today gift shows are popular in many countries, and the purpose of these shows is to find talented
people, and help them to introduce themselves to each other.Actually, many people now watch these sh
ows, and during this years find more fans that increase the Viewer, and many sponsors.']
```

In [ ]: