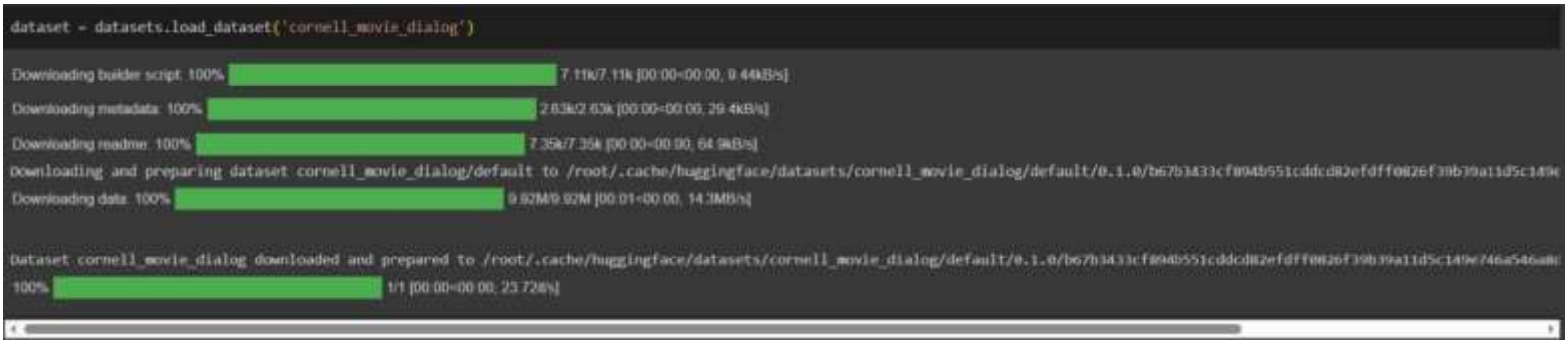
Approach#1:

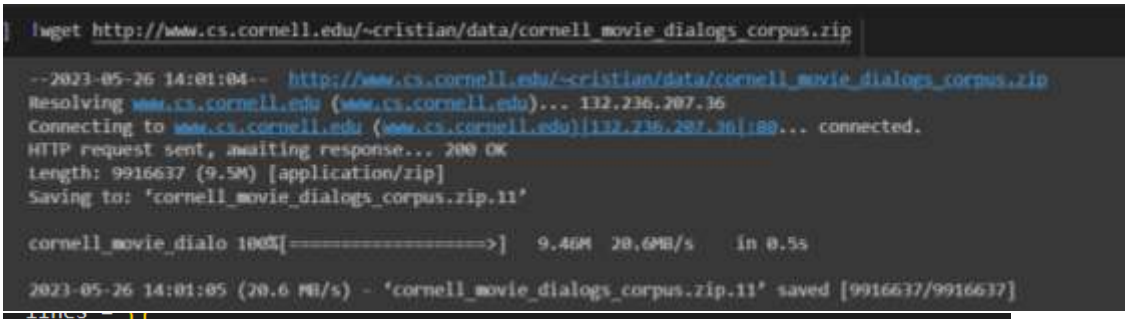
We start by load data as we did In Tensorflow Approach#1

But as we know that data of cornell_movie_dialogs_corpus is too large and tinyshakespeare is too small so the data take too long to get loaded so we start changing the the way of load to be easier for use



Approach#2:

We start by Loading and unzipping data cornell_movie_dialogs_corpus




```
with zipfile.ZipFile('cornell_movie_dialogs_corpus.zip', 'r') as zip_ref:
    zip_ref.extractall('data')
```

we are going to work on two files in this dataset movie_lines.txt which contains { "lineID","chartcterID","movieID","charcter","text"} and movie_conversations.txt each one of them contains {"charcter1ID","charcter2ID","movieID","utteranceIDs"} so we had to merge them to get all text in list to start working on it

```
with open('data/cornell movie-dialogs corpus/movie_lines.txt', 'r', encoding='iso-8859-1') as f:
    for line in f:
        values = line.split(" +++$+++ ")
            # Extract fields
        lineObj = {}
        for i, field in enumerate(line_fields):
            lineObj[field] = values[i]
        lines[lineObj['lineID']] = lineObj
```

```
lines
```

```
{'L1045': {'lineID': 'L1045',
  'chartcterID': 'u0',
  'movieID': 'm0',
  'charcter': 'BIANCA',
  'text': 'They do not!\n'},
 'L1044': {'lineID': 'L1044',
  'chartcterID': 'u2',
  'movieID': 'm0',
  'charcter': 'CAMERON',
  'text': 'They do to!\n'},
 'L985': {'lineID': 'L985',
  'chartcterID': 'u0',
  'movieID': 'm0',
```

```
conv_fields=["charcterID","charcterID","movieID","utteranceIDs"]
conversations = []

with open('data/cornell movie-dialogs corpus/movie_conversations.txt', 'r', encoding='iso-8859-1') as f:
    for line in f:
        values = line.split(" +++$+++ ")
        # Extract fields
        convObj = {}
        for i, field in enumerate(conv_fields):

            convObj[field] = values[i]
        # Convert string to list (convObj["utteranceIDs"] == "['L598485', 'L598486', ...]")
        lineIds = eval(convObj["utteranceIDs"])
        # Reassemble lines
        convObj["lines"] = []
        for lineId in lineIds:
            convObj["lines"].append(lines[lineId])
        conversations.append(convObj)
```

```
conversations
```

```
[{'charcterID': 'u0',
  'charcterID': 'u2',
  'movieID': 'm0',
  'utteranceIDs': "['L194', 'L195', 'L196', 'L197']\n",
  'lines': [{'lineID': 'L194',
    'chartcterID': 'u0',
    'movieID': 'm0',
    'charcter': 'BIANCA',
    'text': 'Can we make this quick?  Roxanne Korrine and Andrew Barrett are having an incredibly horrendous public break- up on the quad.  Again.\n'},
   {'lineID': 'L195',
    'chartcterID': 'u2',
    'movieID': 'm0',
```

```
qa_pairs = []
for conversation in conversations:
    # Iterate over all the lines of the conversation
    for i in range(len(conversation["lines"]) - 1):  # We ignore the last line (no answer for it)
        inputLine = conversation["lines"][i]["text"].strip()
        targetLine = conversation["lines"][i+1]["text"].strip()
        # Filter wrong samples (if one of the lists is empty)
        if inputLine and targetLine:
            qa_pairs.append([inputLine, targetLine])
```

```
qa_pairs
```

```
[['Can we make this quick?  Roxanne Korrine and Andrew Barrett are having an incredibly horrendous public break- up on the quad.  Again.',
  "Well, I thought we'd start with pronunciation, if that's okay with you."],
 ["Well, I thought we'd start with pronunciation, if that's okay with you.",
  'Not the hacking and gagging and spitting part.  Please.'],
 ['Not the hacking and gagging and spitting part.  Please.',
  "Okay... then how 'bout we try out some French cuisine.  Saturday?  Night?"],
 ["You're asking me out.  That's so cute. What's your name again?",
```

It gives vocb_size=136 and we notice that the data wasn`t clean enough

```
          !"#$%&')*+,-./0123456789:;<=>?ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_`abcdefghijklmnopqrstuvwxyz{|}~¡¢£¤¥¦§¨©ª«¬®¯°±²³´µ¶·¸¹º»¼½¾¿ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞßàáâãäåæçèéêëìíîïðñòóôõöùúûü
136
0.218888 M parameters
```

BUT the results was not too bad

```
step 0: train loss 5.2369, val loss 5.2341
step 100: train loss 2.6833, val loss 2.6853
step 200: train loss 2.5178, val loss 2.5382
step 300: train loss 2.4238, val loss 2.4317
step 400: train loss 2.3136, val loss 2.3243
step 500: train loss 2.2305, val loss 2.2388
step 600: train loss 2.1721, val loss 2.1837
step 700: train loss 2.1252, val loss 2.1198
```

```
step 4300: train loss 1.6823, val loss 1.7242
step 4400: train loss 1.6714, val loss 1.7013
step 4500: train loss 1.6742, val loss 1.6988
step 4600: train loss 1.6810, val loss 1.7035
step 4700: train loss 1.6816, val loss 1.6867
step 4800: train loss 1.6581, val loss 1.6966
step 4900: train loss 1.6747, val loss 1.7012
step 4999: train loss 1.6734, val loss 1.6849
```

```
's hinds.  Ot. I bring him <u>met you stake I! You're gonno know. Mls the here faten soidet. Resind's abo
```

Approach#3:

We need to clean data to be more easier for the model to train and predict more accurate so we used processes_sentences that was used in tensorflow to handel the data

```
def preprocess_sentence(sentence):
    #sentence=list(itertools.chain(*sentence))
    sentence=" ".join(sentence)
    # creating a space between a word and the punctuation following it
    # eg: "he is a boy." => "he is a boy ."
    sentence = re.sub(r"([?.!,])", r" \1 ", sentence)
    sentence = re.sub(r'[" "]+', " ", sentence)
    # removing contractions
    sentence = re.sub(r"i'm", "i am", sentence)
    sentence = re.sub(r"he's", "he is", sentence)
    sentence = re.sub(r"she's", "she is", sentence)
    sentence = re.sub(r"it's", "it is", sentence)
    sentence = re.sub(r"that's", "that is", sentence)
    sentence = re.sub(r"what's", "that is", sentence)
    sentence = re.sub(r"where's", "where is", sentence)
    sentence = re.sub(r"how's", "how is", sentence)
    sentence = re.sub(r"\'ll", " will", sentence)
    sentence = re.sub(r"\'ve", " have", sentence)
    sentence = re.sub(r"\'re", " are", sentence)
    sentence = re.sub(r"\'d", " would", sentence)
    sentence = re.sub(r"\'re", " are", sentence)
    sentence = re.sub(r"won't", "will not", sentence)
    sentence = re.sub(r"can't", "cannot", sentence)
    sentence = re.sub(r"n't", " not", sentence)
    sentence = re.sub(r"n'", "ng", sentence)
    sentence = re.sub(r"'bout", "about", sentence)
    # replacing everything with space except (a-z, A-Z, ".", "?", "!", ",")
    sentence = re.sub(r"[^a-zA-Z?.!,]+", " ", sentence)
    sentence = sentence.strip()
    return sentence
```

So the output was more clean:

```
qa_pairs
```

```
['Can we make this quick ? Roxanne Korrine and Andrew Barrett are having an incredibly horrendous public break up on the quad . Again . Well , I thought we would start with
pronunciation , if that is okay with you .',
 'Well , I thought we would start with pronunciation , if that is okay with you . Not the hacking and gagging and spitting part . Please .',
 'Not the hacking and gagging and spitting part . Please . Okay . . . then how about we try out some French cuisine . Saturday ? Night ?',
 'You are asking me out . That s so cute . What s your name again ? Forget it .',
 'No , no , it is my fault we did not have a proper introduction Cameron .',
 'Cameron . The thing is , Cameron I m at the mercy of a particularly hideous breed of loser . My sister . I cannot date until she does .',
 'The thing is , Cameron I m at the mercy of a particularly hideous breed of loser . My sister . I cannot date until she does . Seems like she could get a date easy enough . . .',
 'Why ? Unsolved mystery . She used to be really popular when she started high school , then it was just like she got sick of it or something .',
 'Unsolved mystery . She used to be really popular when she started high school , then it was just like she got sick of it or something . That s a shame .',
 'Gosh , if only we could find Kat a boyfriend . . . Let me see what I can do .',
```

So vocab_size=57

```
 !,.?ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
57
0.208697 M parameters
step 0: train loss 4.2410, val loss 4.2412
```

The output was getting better:

```
step 0: train loss 4.2410, val loss 4.2412
step 100: train loss 2.4309, val loss 2.4301
step 200: train loss 2.3255, val loss 2.3325
step 300: train loss 2.2262, val loss 2.2424
step 400: train loss 2.1612, val loss 2.1693
step 500: train loss 2.1090, val loss 2.0984
step 600: train loss 2.0653, val loss 2.0735
step 700: train loss 2.0202, val loss 2.0140
step 800: train loss 1.9791, val loss 1.9838
step 900: train loss 1.9576, val loss 1.9429
step 1000: train loss 1.8933, val loss 1.9049
step 1100: train loss 1.8875, val loss 1.8731
```

```
step 3900: train loss 1.5933, val loss 1.6079
step 4000: train loss 1.5990, val loss 1.5987
step 4100: train loss 1.5896, val loss 1.5978
step 4200: train loss 1.5770, val loss 1.5880
step 4300: train loss 1.5703, val loss 1.5883
step 4400: train loss 1.5770, val loss 1.5936
step 4500: train loss 1.5704, val loss 1.5743
step 4600: train loss 1.5615, val loss 1.5798
step 4700: train loss 1.5572, val loss 1.5675
step 4800: train loss 1.5641, val loss 1.5872
step 4900: train loss 1.5649, val loss 1.5696
step 4999: train loss 1.5594, val loss 1.5655
```

```
. To reawle ownlywhichead you one migh . We if at to look more , Caso . Lookes as you we and just bave of hope e prostage like wanted back croom cutching facted that . I m plean
```