nursery_rhymes_LSTM

April 8, 2022

1 generative LSTM model

this model uses data from poems to create it's own. by using LSTMs, the model can learn the structure of what words follow other words.

references:

word based text generation

letter based text generation

Tarek El-Hajjaoui

Train an LSTM on the nursery rhymes below and use it to create a new nursery rhyme. The new nursery rhyme should consist of 30 lines, each of which is comprised of 20 words. As there is no specific quantitative metric to assess here, this portion of the writeup's analysis of results will consist of a human interpretation of the performance of the model's nursery rhyme generation.

Please submit your code, a README, and the writeup. Given the multiple experiments within this assignment, your report will likely need to be ~4 pages

connecting drive to google colab environment way to access files stored in google drive

[43]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

1.1 notebook variables

[44]:

```
raw_poem_path = "/content/drive/MyDrive/Colab Notebooks/poem_LSTM/data/
 ↪nursery_rhymes.txt"
cleaned_poem_path = "/content/drive/MyDrive/Colab Notebooks/poem_LSTM/data/poem.
 ↪csv"

load_weights = True
```

1.2 functions for: raw data -> table data

every four lines is a new poem, two lines to the starting verses and one line to the next verse. I'll be separating the poems into table form by taking the text in each poem and matching it with the appropriate title per record in the table.

strip_chars use this to strip funky or odd characters that provice little or no meaning in the formation of words

input: string to strip, optional array of characters to strip

returns: stripped string

[45]:

```
def strip_chars(string_strip, char_list = ['\n','\"','\'',',',';
 ↪','_','|','+','-',':','.','?','!']):
  for char in char_list:
    string_strip = string_strip.replace(char, '')
  return string_strip
```

separate_lines_into_poems using the existing spacing in the poems txt file to build a tabular form

input: file

returns: text file in a csv form

[46]:

```
def separate_lines_into_poems(lines):
    txt_file = "title,text\n"

    curr_poem = ""
    poem_title = ""

    count_new_lines = 0

    # grabbing first title line
    for line in lines:
      if(line != '\n'):
        poem_title = strip_chars(line)
        break

    # using spaces of four to break up poems
    for line in lines[0:]:
        if (count_new_lines == 4):

            curr_poem = strip_chars(string_strip = curr_poem)

            # curr_poem = curr_poem.replace("\n\n", "\n")
            curr_poem = " ".join(curr_poem.split('\n'))
            curr_poem = " ".join(curr_poem.split())

            line_data = (poem_title.replace('\n', '').replace(',','') + "," +␣
 ↪curr_poem + "\n").lower()
```

```
            txt_file += line_data

            curr_poem = ""

            poem_title = line
        else:
            curr_poem += line

        if (line == '\n'):
            count_new_lines += 1

        if (line != '\n'):
            count_new_lines = 0

    return txt_file
```

cleaning_poem_text full function that cleans the poem text and writes it to the correct file

input: file to read raw text from, file to write cleaned csv to

returns: nothing, it writes to a file

[47]:

```
def cleaning_poem_text(in_file, out_file):
  # open txt file
  with open(in_file,"r") as f_in:
    #separate the file into poems
    poem_txt = separate_lines_into_poems(f_in.readlines())

  with open(out_file, 'w') as f_out:
    f_out.write(poem_txt)
```

1.3 starting LSTM modelling

[48]:

```
cleaning_poem_text(raw_poem_path, cleaned_poem_path)
```

[49]:

```
# import nltk # getting standard stopwords
# nltk.download('stopwords')
# from nltk.corpus import stopwords
# STOPWORDS = set(stopwords.words('english'))
```

reading poems from csv

[50]:

```
import csv

# creating poem titles and poem texts
def poem_csv_reader(file_path):
  titles = []
  texts = []
  with open(file_path, 'r') as csvfile:
      reader = csv.reader(csvfile, delimiter=',')
      next(reader)
      for row in reader:
          titles.append(row[0])
          article = row[1]
          # for word in STOPWORDS: #replacing stop words with blanks
          #     token = ' ' + word + ' '
          #     article = article.replace(token, ' ')
          #     article = article.replace(' ', ' ')
          texts.append(article)

      return titles, texts
```

[51]:
```
poem_titles, poem_texts = poem_csv_reader(cleaned_poem_path)
print("number of poem titles: ", len(poem_titles))
print("number of poem texts: ", len(poem_texts))
```

```
number of poem titles:  287
number of poem texts:  287
```

[52]:
```
data = " ".join(poem_texts)
```

new model

[53]:
```
from numpy import array
import numpy as np
from keras.preprocessing.text import Tokenizer
from tensorflow.keras.utils import to_categorical
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Embedding
from keras.layers import Dropout
from keras.callbacks import ModelCheckpoint
```

1.4 data preparation

[54]:
```
# integer encode sequences of words
tokenizer = Tokenizer()
tokenizer.fit_on_texts([data])
encoded = tokenizer.texts_to_sequences([data])[0]
```

[55]:
```
# retrieve vocabulary size
vocab_size = len(tokenizer.word_index) + 1
print('Vocabulary Size: %d' % vocab_size)
```

```
Vocabulary Size: 2406
```

[56]:
```
# encode 2 words -> 1 word
sequences = list()
for i in range(2, len(encoded)):
        sequence = encoded[i-2:i+1]
        sequences.append(sequence)
print('Total Sequences: %d' % len(sequences))
```

```
Total Sequences: 15842
```

[57]:
```
# pad sequences
max_length = max([len(seq) for seq in sequences])
sequences = pad_sequences(sequences, maxlen=max_length, padding='pre')
print('Max Sequence Length: %d' % max_length)
```

```
Max Sequence Length: 3
```

[58]:
```
# split into input and output elements
sequences = array(sequences)
X, y = sequences[:,:-1],sequences[:,-1]
y = to_categorical(y, num_classes=vocab_size)
```

1.5 model building

[59]:
```
# define model
model = Sequential()
model.add(Embedding(vocab_size, 10, input_length=max_length-1))
model.add(LSTM(50))
model.add(Dense(vocab_size, activation='softmax'))
print(model.summary())
```

```
Model: "sequential_2"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 2, 10)             24060

 lstm_2 (LSTM)               (None, 50)                12200

 dense_2 (Dense)             (None, 2406)              122706

=================================================================
Total params: 158,966
Trainable params: 158,966
Non-trainable params: 0

_________________________________________________________________
None
```

[60]:

```
# compile network
model.compile(loss='categorical_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])
```

[61]:

```
# define the checkpoint
filepath="weights-improvement-{epoch:02d}-{loss:.4f}.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1,␣
 ↪save_best_only=True, mode='min')
callbacks_list = [checkpoint]
```

[62]:

```
# fit network
if(not load_weights):
  model.fit(X, y, epochs=500, verbose=2, callbacks=callbacks_list)
```

[63]:

```
if(load_weights):
  # load the network weights
  filename = "/content/drive/MyDrive/Colab Notebooks/poem_LSTM/
 ↪weights-improvement-500-0.8179.hdf5"
  model.load_weights(filename)
  model.compile(loss='categorical_crossentropy', optimizer='adam')
```

1.6 sequence generation

[64]:

```
next_seq = ""

import random
def get_rand_word():
    rand_word = ''
    is_valid = False
    while is_valid == False:
      rand_word = data.split(' ')[random.randint(0, len(data.split(' ')) - 1)]
      if rand_word != ' ' and len(rand_word) != 0:
        is_valid = True
    return rand_word
```

[65]:

```
def generate_seq(seed_seq='', supress_msg=True):
    if seed_seq == '':
      if supress_msg == False:
        print('No sequence was chosen, a random pair of words is being \
      chosen as the seed text.')
      seed_seq = get_rand_word() + " " + get_rand_word()
      if supress_msg == False:
        print(f"Random pair chosen: {seed_seq}")
    # when provide the model with a pair of words
    in_text = seed_seq
    # set the 2nd wor of current sequence as the first for the next sequence
    next_seq = seed_seq.split(' ')[1]
    # encode the text as integer
    encoded = tokenizer.texts_to_sequences([in_text])[0]
    # pre-pad sequences to a fixed length
    encoded = pad_sequences([encoded], maxlen=max_length-1, padding='pre')
    # predict probabilities for each word
    predict = model.predict(encoded, verbose=0)
    yhat=np.argmax(predict,axis=1)
    # map predicted word index to word
    out_word = ''
    for word, index in tokenizer.word_index.items():
      if index == yhat:
        out_word = word
        break
    # append to input
    in_text = out_word
    # set the out word as the 2nd word for next_seq variable
    next_seq += ' ' + out_word
    return in_text
```

[66]:

```
# generate a sequence from a language model
def generate_seq_individual(model, tokenizer, max_length, seed_text, n_words):
  in_text = seed_text
  # generate a fixed number of words
  for _ in range(n_words):
    # encode the text as integer
    encoded = tokenizer.texts_to_sequences([in_text])[0]
    # pre-pad sequences to a fixed length
    encoded = pad_sequences([encoded], maxlen=max_length, padding='pre')
    # predict probabilities for each word
    yhat = model.predict(encoded, verbose=0)
    # yhat = np.random.choice(len(yhat),)
    yhat = yhat.argmax(axis=-1)
    # map predicted word index to word
    out_word = ''
    for word, index in tokenizer.word_index.items():
      if index == yhat:
        out_word = word
        break
                # append to input
    in_text += ' ' + out_word
  return in_text
```

[67]:

```
# evaluate model
print(generate_seq_individual(model, tokenizer, max_length-1, 'Kyle did', 5))
print(generate_seq_individual(model, tokenizer, max_length-1, 'Connor does', 3))
print(generate_seq_individual(model, tokenizer, max_length-1, 'Arman has', 5))
print(generate_seq_individual(model, tokenizer, max_length-1, 'pail of', 5))
```

```
Kyle did our girls or as little
Connor does z and heres
Arman has like the lamb so mild
pail of water the water and over
```

[68]:

```python
def gen_poem(trained_model,
                        n_lines=30,
                        n_words = 20,
                        output_file="lstm_rhymes.txt"):
  with open(output_file, "w") as out_file:
    for i in range(n_lines):
      msg = "{:<4}".format(f"{i + 1}")
      seed_pair = get_rand_word() + " " + get_rand_word()
      curr_line = generate_seq(seed_pair)
      for n in range(n_words):
        curr_line += " " + generate_seq(next_seq)
      msg += curr_line
      print(msg)
      out_file.write(msg + '\n')
      msg = ""
      curr_line = ""
    print("")

gen_poem(model, output_file="lstm_rhymes1.txt")
gen_poem(model, output_file="lstm_rhymes2.txt")
```

```
1   and a the fish her out the a little heigho you soon with froggy then
daughter the go better day for
2   a learned say jack and wrap other miller that cow the again of you and from
away of poor come water
3   was which yourself skilful day down news when and together up you with your
and says jolly a the been and
4   so robin burden the with with in of got that lived many to love
daffydowndilly must house and i cold in
5   rowley ladle to mother so a fishes mice they horn the in up the eight was
every get boy and on
6   twenty legs but a milk all and bad a and goose the coo thomas and far
cinders off as with shall
7   dont be so had an old wrap then a s she had cat the on naughty house be for
i pretty
8   the they he got and leave rat the come then will off going or wife johnny
back you in was nor
9   or the my there so shant shall in down you at mowing and she to said reason
of so the about
10  king if two the i till read pin the my gulp in bouncing with not boy good
wand play to pig
11  peter tie asleep the queen his two away stole went she he on had said leave
what nothing what i the
```

12 the he he named is still the a i did nodded he pussy woman gold has on when they eat were
13 such a my to jack are will a and back meeow to of dale away bone jane cow as can and
14 powley you a you fol poor fly rowley begins it weather and and it ship none in as of and mouse
15 yes and there wind or simon cat naughty he and fire it them ill the all bouncing a quoth girls she
16 ever will dance and queen with cat high let other and bridge sun this farmer off ye your woodcock ring you
17 did what jack blind what points and dog she a there this anon spinach bouncing dame young the rogue gave is
18 five hector the the i i mouse full rowley king are a little a them dont notshe spread was poor it
19 crown me a and never garden back i and old kittens gave of and woolly going and it fire says home
20 one to as for they will was he your money stands he gave went would burden and there the a old
21 an parson the a some have nag king had pretty crumpled two beef his slip for dickery little starts come bessy
22 dears says away two hanged with if saturday as were all not he and ho oh bade taken bade old bright
23 to and be tried find he his queen is the his penny when by jack will ring had gilly it and
24 she this and dance they thrive young the betty woman go a olin of jack shoot climbed what heigh well thou
25 a a rogue shant a little cock as have horses old crumpled up ate tell beating went the it is the
26 he you a head in dog town there wren spain it come will i little sack his to roast man to
27 carries house morning a of not with says a i clothes yonder babylon goosey a by shall dont into as he
28 home she ill she carrion toll i and little what to she diller charley tried about and news and with home
29 a gaily at a wood drake knows milk would is bit an went and spark he johnny i till pig come
30 the were up fly fought in them the with some back tell then up he wee to king his buble my

1 called she pin all and he one fly bleed joyful wedding when neer going go dine mother says a on if
2 to children as he her hop i and tune milk said night and your then and white at come think love
3 a rigs going says a ate nothing often and will may write made barley sake all to hop you up buble
4 her hay sun then frog or the and the upon to swim woman notshe well go will the all burned take
5 say clout hole either as john as her all thou by the the and lift any and

little young went as
6 it both song up full will a man them miller spoke three or his together do rat and you made it
7 way put the i little thou wont quoth joy what man and matter in was at he do quiet they and
8 milk bridge and i a a up the the he a mother she broom his remedy have the and silver safe
9 a the clothes his school is that hill his it he he how ride off came wand lady to and could
10 upon said i the had made may clean little black the lived and yellow for down them dogs his is gave
11 the down side little cant nail little gone little i every are nest that the to look want out yonder the
12 applepie has with you your he pulled mother cats did i one keep dig dolly if news there said and for
13 was she carrion walls locket a cock behind but got or little rope merry for jolly she a iron her the
14 drums frog swarm purrr i on the it not there it the and suns plum the what in such new drink
15 shall an that he carrion must bread a be ho cake bit queen what the got his soon penny if gently
16 wont were came over on dame he would it raised the it thief their purrr mammy ah found and the wheelbarrow
17 water x a a old the she one the she pig jenny too eight the he pig his on care not
18 he betty the to going pray you burden nanny high dont little to i maids the a of with why clothes
19 dressed bit away and loved do dont try fell my good your in raven pair through three way mice was very
20 sea get and b a old took oh of i out tune a head miller down me run fire reason fire
21 and my a quoth the remain and burden calf mother he burden for both little a of make for i the
22 to to so taffy she it to loved her will not wilt see says nancy was of if sung money up
23 was this heigho the but kill with jack miller both he brought cats going and i here about mother all find
24 the with and to tails cow not in out creep go with was heigh pigs heigh gave the had and killed
25 the made his came little too do my and will robin the but her then is the found the it again
26 flew neither there moon or to i the little and and the toes and over little he was you you battle
27 back some do it heigh a and a minds h to clothes plainlooking to nobody heigh i kittens going old you
28 wives mistress you i fiddle sing robin tail and was the and the pig carried is man and a pig a
29 mittens there pussy and she kittens tried pounds not ill the is down cook

```
eho can ding with old storms for
30  making whip it does carried and to about and penny a this out at black news
two of him with you
```

Previously I tokenized the letters an not the words. It tried to spell the words rather than using them out of a ‘dictionary’. Splitting the words into a dicitonary solved this.