
Text Generation and Spelling Correction

— Vũ Trung Hiếu —

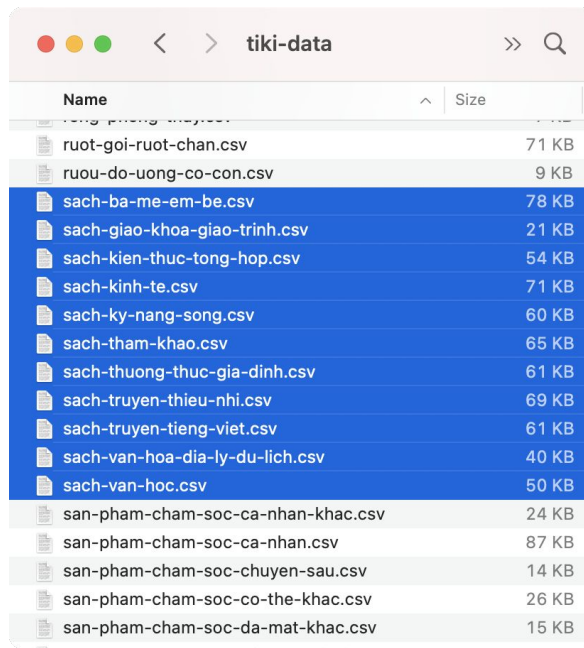
Contents

1. Dataset
2. Model
3. Text Generation
4. Spelling Correction

1. Dataset

Dữ liệu là danh sách sản phẩm được cào từ trang web của Tiki.

Đề tài này dùng dữ liệu về tên sản phẩm về sách



Name	Size
ruot-goi-ruot-chan.csv	71 KB
ruou-do-uong-co-con.csv	9 KB
sach-ba-me-em-be.csv	78 KB
sach-giao-khoa-giao-trinh.csv	21 KB
sach-kien-thuc-tong-hop.csv	54 KB
sach-kinh-te.csv	71 KB
sach-ky-nang-song.csv	60 KB
sach-tham-khao.csv	65 KB
sach-thuong-thuc-gia-dinh.csv	61 KB
sach-truyen-thieu-nhi.csv	69 KB
sach-truyen-tieng-viet.csv	61 KB
sach-van-hoa-dia-ly-du-lich.csv	40 KB
sach-van-hoc.csv	50 KB
san-pham-cham-soc-ca-nhan-khac.csv	24 KB
san-pham-cham-soc-ca-nhan.csv	87 KB
san-pham-cham-soc-chuyen-sau.csv	14 KB
san-pham-cham-soc-co-the-khac.csv	26 KB
san-pham-cham-soc-da-mat-khac.csv	15 KB

1. Dataset

Dataset được làm sạch bằng cách loại bỏ các dấu chấm câu và các ký tự lạ. Bắt đầu một chuỗi là "{" và kết thúc một chuỗi là "}". Sau đó data được làm giàu và xáo trộn.

```
# clean  
bos = "{"  
eos = }"  
regex = "[^0-9a-zAզմաթևգըծոբդօժիւղհեթկէնյնպնխքնրվ  
for i in range(len(lines)):  
    lines[i] = re.sub(regex, " ", lines[i].lower()).strip()  
    lines[i] = bos + re.sub(' ', ' ', lines[i]) + eos  
lines[:10]
```

```
➤ ['{madame chic rất thần thái rất paris}',
  '{hành trình của linh hồn}',
  '{thai giáo theo chuyên gia 280 ngày mỗi ngày đọc một trang}',
  '{eat clean thực đơn 14 ngày thanh lọc cơ thể và giảm cân}',
  '{thánh kinh dưỡng da}',
  '{green smoothies gia m cần la m đe p da tăng cường sức đề kháng vợ i 7}',
  '{brew tuyết t đi nh ca phê ta i nhà}',
  '{khởi sự ăn chay}',
  '{đừng chỉ mặc màu đen}',
  '{chào juice}']
```

```
[ ] # augment
text = []
for line in lines:
    line = [line]*10
    text.extend(line)
random.shuffle(text)
text = "".join(text)
text[:500]
```

{việtnam cùng có giới tự do toán học tập 1}{rich habits thì quen thành
cùng của những triệu phú tự thân}{giải mặt ngoại hạng anh}{đời sống b ản c
ủ a cây}{sự giàu và nghèo của các dân tộc}{brew tuy t đi nh ca phê ta i nha
{tư chiến lược marketing đến doanh nghiệp thành công}{science encyclopedia
bách khoa thứ v khoa học trái đất và vũ trụ}{triệu phú thức tỉnh bị kíp đ
khôi dòng suối nguồn think trong tâm thức}{bạn đắt giá bao nhiêu tặng
kèm bộ bookmark tiki love books}{thi đ 3 0}{k'}

1. Dataset

Từ điển bao gồm tập ký tự từ 0-9, a-z và các nguyên âm có dấu trong tiếng việt (gồm 106 ký tự)

Tập dữ liệu train là seq2seq, một seq dài 100 ký tự, nhãn của một seq là cũng là một seq 100 nhưng được dịch phải một ký tự

Ký tự được mã hoá thành số trước khi đưa vào train.

2. Model

Model 1: Embedding layer, GRU layer, dense layer

Hàm loss là cross entropy, có early stopping

Model: "sequential_2"

Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(30, None, 256)	27136
gru_2 (GRU)	(30, None, 1024)	3938304
dense_2 (Dense)	(30, None, 106)	108650

Total params: 4,074,090

Trainable params: 4,074,090

Non-trainable params: 0

Epoch 1/30	1583/1583 [=====]	- 49s 31ms/step - loss: 1.2216
Epoch 2/30	1583/1583 [=====]	- 49s 31ms/step - loss: 0.5236
Epoch 3/30	1583/1583 [=====]	- 49s 31ms/step - loss: 0.4420
Epoch 4/30	1583/1583 [=====]	- 49s 31ms/step - loss: 0.4282
Epoch 5/30	1583/1583 [=====]	- 49s 31ms/step - loss: 0.4324
Epoch 6/30	1583/1583 [=====]	- 49s 31ms/step - loss: 0.4628
Epoch 7/30	1583/1583 [=====]	- 49s 31ms/step - loss: 0.6685

2. Model

Model 2: Embedding layer, LSTM layer, dense layer

Hàm loss là cross entropy, có early stopping

Model: "sequential_20"

Layer (type)	Output Shape	Param #
embedding_20 (Embedding)	(30, None, 256)	27136
lstm_5 (LSTM)	(30, None, 1024)	5246976
dense_20 (Dense)	(30, None, 106)	108650

Total params: 5,382,762
Trainable params: 5,382,762
Non-trainable params: 0

```
Epoch 1/30
1583/1583 [=====] - 59s 37ms/step - loss: 1.1461
Epoch 2/30
1583/1583 [=====] - 59s 37ms/step - loss: 0.5018
Epoch 3/30
1583/1583 [=====] - 59s 37ms/step - loss: 0.4193
Epoch 4/30
1583/1583 [=====] - 59s 37ms/step - loss: 0.3967
Epoch 5/30
1583/1583 [=====] - 59s 37ms/step - loss: 0.3848
Epoch 6/30
1583/1583 [=====] - 59s 37ms/step - loss: 0.3776
Epoch 7/30
1583/1583 [=====] - 59s 37ms/step - loss: 0.3725
Epoch 8/30
1583/1583 [=====] - 59s 37ms/step - loss: 0.3700
Epoch 9/30
1583/1583 [=====] - 59s 37ms/step - loss: 0.3680
Epoch 10/30
1583/1583 [=====] - 59s 37ms/step - loss: 0.3687
Epoch 11/30
1583/1583 [=====] - 59s 37ms/step - loss: 0.3693
<tensorflow.python.training.tracking.util.CheckpointLoadStatus at 0x7f57236
```

3. Text Generation

Giải thuật text generation

Input: Một chuỗi ký tự bắt đầu

Chuỗi ký tự bắt đầu được forward qua model, mỗi bước model sẽ cho ra bảng phân phối xác suất của 106 ký tự và ta sẽ chọn ký tự có xác suất xuất hiện cao nhất.

Ký tự được chọn sẽ được tiếp tục đưa vào model và qua trình cứ tiếp tục diễn ra đến khi gặp ký tự kết thúc chuỗi hoặc xác suất xuất hiện của tất cả các ký tự quá bé.

3. Text Generation

Kết quả cho model GRU:

```
▶ #Build new model to generate
result_of_gru_char = generate_text(generate_model, start_string=u"dế mèn phiê")
print(result_of_gru_char)
result_of_gru_char = generate_text(generate_model, start_string=u"nhà kh")
print(result_of_gru_char)
result_of_gru_char = generate_text(generate_model, start_string=u"sách tập làm v")
print(result_of_gru_char)
result_of_gru_char = generate_text(generate_model, start_string=u"thanh lợ")
print(result_of_gru_char)
```

```
↳ (1, 11)
dế mèn phiêu lưu ký tái nhà ăn cơm học
(1, 6)
nhà khi đúng b
(1, 14)
sách tập làm việc nhà thuật x
(1, 8)
thanh lọc ốc diêu của philập tư duy vệ sách mẹ nhà trường chứng khoán nhật kỳ lực chi kháng kèm s
```

3. Text Generation

Kết quả cho model LSTM:

```
▶ result_of_gru_char = generate_text(generate_model_lstm, start_string=u"dế mèn phiê")  
print(result_of_gru_char)  
result_of_gru_char = generate_text(generate_model_lstm, start_string=u"nhà kh")  
print(result_of_gru_char)  
result_of_gru_char = generate_text(generate_model_lstm, start_string=u"sách tập làm v")  
print(result_of_gru_char)  
result_of_gru_char = generate_text(generate_model_lstm, start_string=u"thanh lợ")  
print(result_of_gru_char)
```

```
↳ (1, 11)  
dế mèn phiêu lưu ký khi những điều lấp lánh được gọi tên tái bản  
(1, 6)  
nhà khoa học  
(1, 14)  
sách tập làm văn  
(1, 8)  
thanh lọc cơ thể và giảm cân
```

Có thể thấy model dùng LSTM cho ra kết quả tốt hơn model dùng GRU.

4. Spelling Correction: No Lookahead

Giải thuật không dùng lookahead

Input: Một chuỗi các từ

Chấp nhận một số lượng từ bắt đầu nào đó là không sai chính tả, ví dụ 7

Từng ký tự sẽ chạy qua model, nếu ký tự tiếp theo có xác suất trong bảng phân phối xác suất mà model dự đoán quá thấp (dưới một ngưỡng), ta sẽ kết luận ký tự đó sai và thay thế bằng ký tự có xác suất xuất hiện cao nhất

4. Spelling Correction: No Lookahead

Giải thuật không dùng lookahead

Input: Một chuỗi các từ

Chấp nhận một số lượng từ bắt đầu nào đó là không sai chính tả, ví dụ 7

Từng ký tự sẽ chạy qua model, nếu ký tự tiếp theo có xác suất trong bảng phân phối xác suất mà model dự đoán quá thấp (dưới một ngưỡng), ta sẽ kết luận ký tự đó sai và thay thế bằng ký tự có xác suất xuất hiện cao nhất

4. Spelling Correction: No Lookahead

Giải thuật không dùng lookahead.

Kết quả

```
▶ # Good cases
correct_text(generate_model_lstm, "để mèn phiêu lưu ký táo bản")
correct_text(generate_model_lstm, "dòng suoi nguồn thịnh vương")
correct_text(generate_model_lstm, "dòng suối nguồn thịnh vượng")
print()
```

☞ Assume the first 7 chars are correct
để mèn phi(e) --> để mèn phiê
để mèn phi(e)u lưu ký tá(o) --> để mèn phiêu lưu ký tái
misspell: để mèn phi(e)u lưu ký tá(o) bản
correct: để mèn phiêu lưu ký tái bản

Assume the first 7 chars are correct
dòng su(o) --> dòng suố
dòng su(o)i nguồn thịnh v(u) --> dòng suối nguồn thịnh vư
dòng su(o)i nguồn thịnh v(u)(o) --> dòng suối nguồn thịnh vượ
misspell: dòng su(o)i nguồn thịnh v(u)(o)ng
correct: dòng suối nguồn thịnh vượng

Assume the first 7 chars are correct
misspell:
correct: dòng suối nguồn thịnh vượng

4. Spelling Correction: No Lookahead

Giải thuật không dùng lookahead.

Kết quả:

Chạy không tốt đối với những trường hợp phải xoá ký tự sai. Điều này có thể giải quyết bằng phương pháp look ahead.



```
# bad case
correct_text(generate_model_lstm, "dòng suối nguồn thịnh vượng")
correct_text(generate_model_lstm, "dòng suối naguồn thịnh vượng")
print()
```



```
Assume the first 7 chars are correct
dòng suối n(n) --> dòng suối ng
dòng suối n(n)(g) --> dòng suối ngu
dòng suối n(n)(g)(u) --> dòng suối nguổ
dòng suối n(n)(g)(u)(ổ) --> dòng suối nguồn
dòng suối n(n)(g)(u)(ổ)(n) --> dòng suối nguồn
dòng suối n(n)(g)(u)(ổ)(n)( ) --> dòng suối nguốn t
dòng suối n(n)(g)(u)(ổ)(n)( ) (t) --> dòng suối nguốn th
dòng suối n(n)(g)(u)(ổ)(n)( ) (t)(h) --> dòng suối nguốn thỉ
dòng suối n(n)(g)(u)(ổ)(n)( ) (t)(h)(i) --> dòng suối nguốn thịn
dòng suối n(n)(g)(u)(ổ)(n)( ) (t)(h)(i)(n) --> dòng suối nguốn thịnh
dòng suối n(n)(g)(u)(ổ)(n)( ) (t)(h)(i)(n)(h) --> dòng suối nguốn thịnh h
dòng suối n(n)(g)(u)(ổ)(n)( ) (t)(h)(i)(n)(h)( ) --> dòng suối nguốn thịnh v
dòng suối n(n)(g)(u)(ổ)(n)( ) (t)(h)(i)(n)(h)( ) (v) --> dòng suối nguốn thịnh vư
dòng suối n(n)(g)(u)(ổ)(n)( ) (t)(h)(i)(n)(h)( ) (v)(u) --> dòng suối nguốn thịnh vượ
dòng suối n(n)(g)(u)(ổ)(n)( ) (t)(h)(i)(n)(h)( ) (v)(u)(q) --> dòng suối nguốn thịnh vượn
dòng suối n(n)(g)(u)(ổ)(n)( ) (t)(h)(i)(n)(h)( ) (v)(u)(q)(n) --> dòng suối nguốn thịnh vượn
dòng suối n(n)(g)(u)(ổ)(n)( ) (t)(h)(i)(n)(h)( ) (v)(u)(q)(n)(g) --> dòng suối nguốn thịnh vượng
misspell: dòng suối n(n)(g)(u)(ổ)(n)( ) (t)(h)(i)(n)(h)( ) (v)(u)(q)(n)(g)
correct: dòng suối nguốn thịnh vượng
```

```
Assume the first 7 chars are correct
dòng suối n(a) --> dòng suối ng
dòng suối n(a)(a) --> dòng suối ngu
dòng suối n(a)(a)(g) --> dòng suối nguổ
dòng suối n(a)(a)(g)(u) --> dòng suối nguồn
dòng suối n(a)(a)(g)(u)(ổ) --> dòng suối nguốn
dòng suối n(a)(a)(g)(u)(ổ)(n) --> dòng suối nguốn t
dòng suối n(a)(a)(g)(u)(ổ)(n)( ) --> dòng suối nguốn th
dòng suối n(a)(a)(g)(u)(ổ)(n)( ) (t) --> dòng suối nguốn thỉ
dòng suối n(a)(a)(g)(u)(ổ)(n)( ) (t)(h) --> dòng suối nguốn thịn
dòng suối n(a)(a)(g)(u)(ổ)(n)( ) (t)(h)(i) --> dòng suối nguốn thịnh
dòng suối n(a)(a)(g)(u)(ổ)(n)( ) (t)(h)(i)(n) --> dòng suối nguốn thịnh h
dòng suối n(a)(a)(g)(u)(ổ)(n)( ) (t)(h)(i)(n)(h) --> dòng suối nguốn thịnh hư
dòng suối n(a)(a)(g)(u)(ổ)(n)( ) (t)(h)(i)(n)(h)( ) (v) --> dòng suối nguốn thịnh hư
misspell: dòng suối n(a)(a)(g)(u)(ổ)(n)( ) (t)(h)(i)(n)(h)( ) (v)uợng
correct: dòng suối nguốn thịnh hư ượng
```

4. Spelling Correction: Lookahead

Giải thuật dùng lookahead

Input: Một chuỗi các từ. Chấp nhận một số lượng từ bắt đầu nào đó là không sai chính tả, ví dụ 7

Từng ký tự sẽ chạy qua model, nếu ký tự tiếp theo có xác suất trong bảng phân phối xác suất mà model dự đoán quá thấp (dưới một ngưỡng), ta sẽ tạo ra 4 lựa chọn gồm bỏ ký tự hiện tại, thay ký tự sai bằng một trong 3 ký tự có xác suất cao nhất.

Tính xác suất cả chuỗi về sau đúng nếu dùng 4 lựa chọn này. Chọn lựa chọn có xác suất cao nhất. Và tiếp tục quá trình

4. Spelling Correction: Lookahead

Giải thuật dùng lookahead

Kết quả: Chạy tốt với trường hợp phải xóa ký tự sai hoặc thay thế ký tự đó

```
▶ correct_text_lookahead(generate_model_lstm, "để mèn phiêu lưu ký táo bản")
correct_text_lookahead(generate_model_lstm, "dòng suoi nguồn thịnh vượng")
correct_text_lookahead(generate_model_lstm, "dòng suối nguồn thịnh vượng")
```

```
↳ Assume the first 7 chars are correct
để mèn phi(e) --> để mèn phiê
để mèn phi(e)u lưu ký tá(o) --> để mèn phiêu lưu ký tái
Misspell: để mèn phi(e)u lưu ký tá(o) bản
Correct: để mèn phiêu lưu ký tái bản
```

```
Assume the first 7 chars are correct
dòng su(o) --> dòng suố
dòng su(o)i nguồn thịnh v(u) --> dòng suối nguồn thịnh vu
Misspell: dòng su(o)i nguồn thịnh v(u)ợng
Correct: dòng suối nguồn thịnh vượng
```

```
Assume the first 7 chars are correct
Misspell:
Correct: dòng suối nguồn thịnh vượng

('dòng suối nguồn thịnh vượng', '')
```

```
▶ correct_text_lookahead(generate_model_lstm, "dòng suối nnguồn thịnh vượng")
correct_text_lookahead(generate_model_lstm, "dòng suối naguồn thịnh vượng")
print()
```

```
Assume the first 7 chars are correct
dòng suối n(n) --> dòng suối n
Misspell: dòng suối n(n)guồn thịnh vượng
Correct: dòng suối nguồn thịnh vượng
```

```
Assume the first 7 chars are correct
dòng suối n(a) --> dòng suối n
dòng suối n(a)(a) --> dòng suối n
Misspell: dòng suối n(a)(a)guồn thịnh vượng
Correct: dòng suối nguồn thịnh vượng
```