

BÁO CÁO BÀI TẬP 3

Tuần thứ 3

Họ và tên: Nguyễn Hoàng Minh

Lớp: K67A5

Mã sinh viên: 22001273

Trong tuần này, em tiến hành cài đặt cụ thể mô hình phát hiện dialogue acts và các slot-value pairs tương ứng của chúng. Ngoài ra, em tiến hành cả việc phát hiện intent của người dùng sử dụng kiến trúc sequence2sequence. Các thang đo em sử dụng cho dialogue acts và intent là điểm F1, precision và recall. Riêng việc đo lường slot-value pairs, do em muốn mô hình phải phát hiện chính xác cả slot và value nên em gộp chúng lại thành string có dạng “slot=value” và kiểm tra chính xác trên string đó, chỉ cần slot hoặc value sai thì coi như dự đoán đó là sai. Từ đây thì em thấy rằng cứ mỗi một cặp slot value thì sẽ thành một class riêng, thành ra cả bộ dữ liệu có n cặp slot-value thì sẽ có n classes. Em nghĩ rằng việc đo F1 trên n classes như thế này cũng không chính xác hơn việc đếm phần trăm nhận diện đúng. Vậy nên để đo sự hiệu quả của mô hình trong việc phát hiện slot-value thì em đếm số cặp đúng trên tổng số cặp trong bộ dữ liệu.

I. Biến đổi dữ liệu.

So với tuần 2, em có điều chỉnh bước biến đổi dữ liệu một chút để phù hợp cho việc phát hiện slot-value và intent của người dùng. Các trường của dữ liệu sau khi được biến đổi là như sau:

dialogue_id | turn | utterance | bio_tag | dialogue_act | speaker | active_intent

Trong đó, các trường được thêm vào là **bio_tag** và **active_intent**. **bio_tag** là các nhãn bio cho một câu, ví dụ:

```
0 0    0 0    0 B-DES I-DES    0 B-DEPTIME I-DEPTIME 0    AIRLINE-SHOWFLIGHT
I want to fly to San Francisco on Monday afternoon please EOS
```

active_intent là intent của người dùng tại thời điểm nói câu đó. Ở các bước xử lý sau, em cũng ghép cả **active_intent** vào **bio_tag** và coi nó như là các nhãn BIO. Ý tưởng của em là sẽ sử dụng các nhãn BIO này làm cặp slot-value cho dialogue acts ở trong câu đấy luôn, giống như các tác giả trình bày trong cuốn SLP, Stanford. Ngoài ra, trong các bộ từ vựng em có thêm các token đặc biệt là <eos>, <pad>=0, <unk>=1.

II. Mô hình

Mô hình của em có kiến trúc như sau:

```
MyGRU(
    (embeddings): Embedding(7752, 512, padding_idx=0)
    (flatten): Flatten(start_dim=1, end_dim=-1)
    (grus): GRU(512, 256, num_layers=3, batch_first=True, dropout=0.2)
    (bio_head): Linear(in_features=256, out_features=47, bias=True)
    (act_head): Linear(in_features=256, out_features=128, bias=True)
    (dense_2): Linear(in_features=128, out_features=64, bias=True)
    (dense_3): Linear(in_features=64, out_features=18, bias=True)
)
```

Trong đó, layer **embeddings** biến các tokens thành các dense vector để bắt được ngữ nghĩa. Các layers **grus** là encoder được em sử dụng chung cho cả **bio_head** và **act_head**. **bio_head** sẽ có nhiệm vụ decode vector trả về từ encoder thành các chuỗi nhãn BIO tương ứng, bao gồm cả **active_intent** của người dùng. Còn **act_head** sẽ lấy timestep cuối cùng của **grus** để thực hiện phân loại acts. Các lớp **dense_2** và **dense_3** là các lớp FC được dùng tiếp nối sau **act_head**, do trong quá trình train thì em nhận thấy mô hình bị underfit trong việc nhận diện các acts nên em tăng độ phức tạp cho nhánh này. Mô hình sẽ trả về cả chuỗi nhãn BIO trực tiếp từ **bio_head** và act từ **act_head** sau khi qua các lớp FC.

Trong cuốn sách, tác giả cũng nói về việc truyền thêm acts của câu trước đó vào có thể giúp mô hình nhận diện acts tốt hơn. Em cũng thử nghiệm huấn luyện mô hình với cả hai trường hợp: có acts trước đó và không có. Nếu có thêm acts trước đó thì em sẽ ghép timestep cuối cùng của **grus** cùng với cả vector biểu diễn acts trước đó (phép concatenate) rồi truyền vào **act_head**.

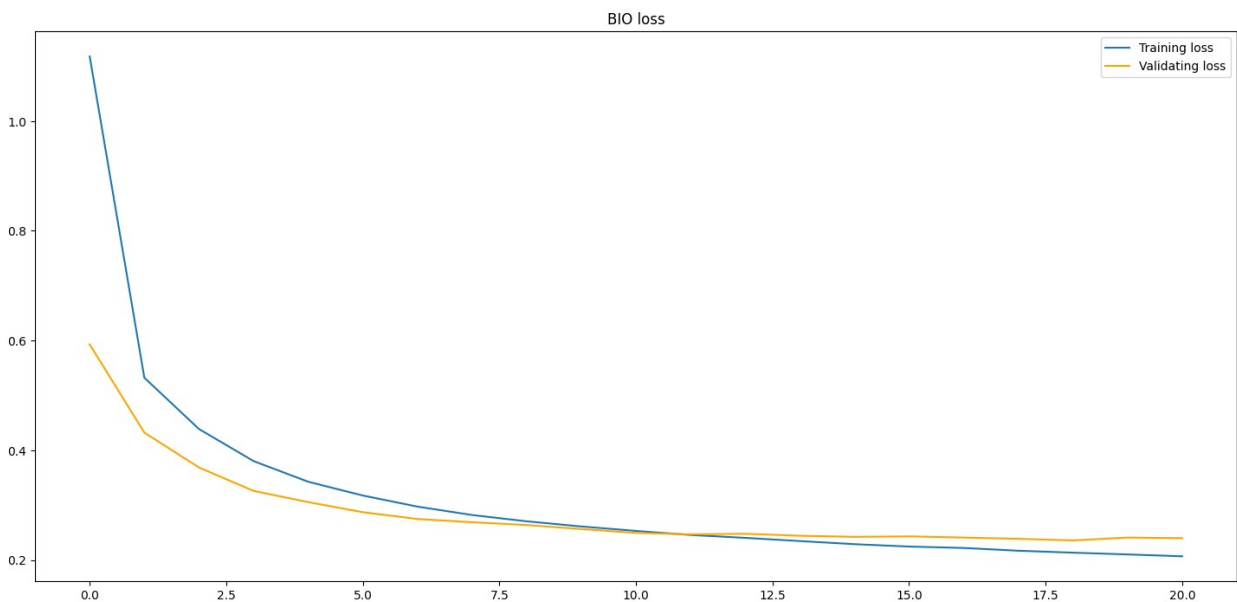
III. Huấn luyện mô hình

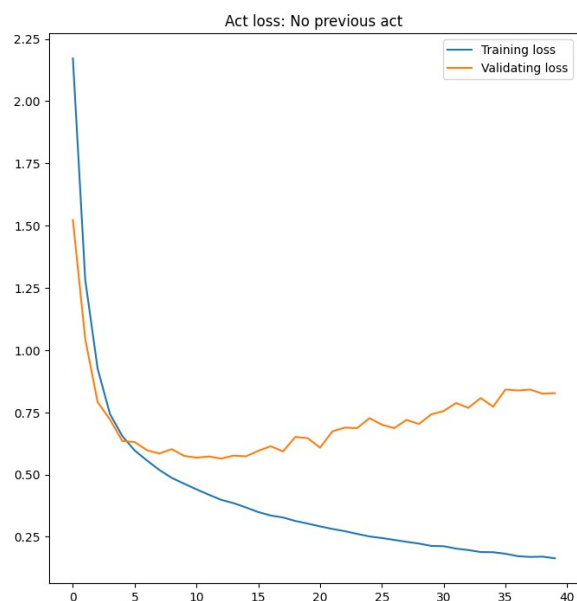
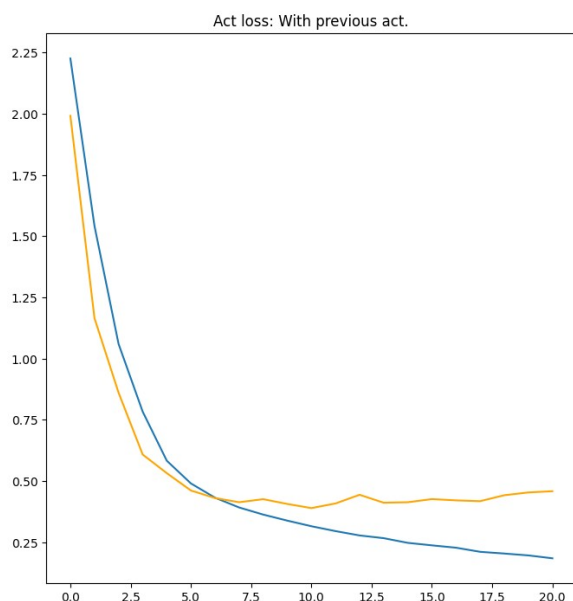
Do các nhãn BIO trong bộ dữ liệu chứa khá nhiều class “O” nên em sẽ xử lý imbalance bằng cách áp dụng class weights trừng phạt các dự đoán sai về class “O” ít đi, chú trọng hơn vào các nhãn khác. Thang đo loss em sử dụng là Cross Entropy Loss, optimizer là Adam với learning rate là 1e-4. Trong quá trình train, em nhận thấy rằng nhiều khi loss của việc nhận diện BIO đã đủ tốt nhưng loss của Acts chưa tốt. Khi đó, khi mô hình tiếp tục cập nhật weights thì lại vô tình làm kết quả của nhận diện BIO xấu đi, hoặc thậm chí là gây ra overfit. Kể cả đối với acts cũng hoàn toàn có thể như vậy. Vậy nên, em thiết lập một cơ chế là: nếu loss của act hoặc bio detection không cải thiện sau một số epoch nào đó thì sẽ freeze nhánh đó lại và chỉ train nhánh kia. Còn nếu như cả hai loss đều không cải thiện sau một số epoch cụ thể thì sẽ dừng train để tránh overfit.

IV. Kết quả

A. Loss

Loss trong quá trình train được đo lại như sau, do việc thêm previous act vào không ảnh hưởng đến BIO detection nên em không so sánh:





Từ hai biểu đồ loss trên, em có những nhận xét rằng:

- Việc nhận diện BIO diễn ra khá tốt, training loss hội tụ sau 20 epoch nhưng sau epoch 10 thì có vẻ như validating loss không cải thiện.
- Việc thêm previous act vào thực sự cải thiện act detection. Bằng chứng là khi có previous act, loss đã giảm xuống còn 0.5 tại epoch thứ 6, còn khi không có previous act thì loss không thể giảm xuống 0.5. Tuy nhiên, từ sau epoch thứ 6, cả hai loss đều không có xu hướng cải thiện.
- Cả biểu đồ Act loss và BIO loss đều gợi ý cho ta rằng có thể mô hình tốt nhất là tại epoch thứ 6 do training loss và validating loss sát nhau, nên em sẽ chọn mô hình thứ 6 để đo sự hiệu quả.

B. Performance

Bảng dưới đây tóm tắt các kết quả của em:

Dataset	Measurements	With previous act	No previous act
Training set	Act F1	0.8282	0.8176
	Intent F1	0.7523	0.7794
	Slot-Value accuracy	0.7188	0.7293
Validating set	Act F1	0.5927	0.5359
	Intent F1	0.6938	0.6811
	Slot-Value accuracy	0.6953	0.7227
Testing set	Act F1	0.6052	0.5348
	Intent F1	0.6903	0.6815
	Slot-Value accuracy	0.6902	0.7155

Nhìn vào bảng trên, em có một số nhận xét như sau:

- Nhìn vào điểm F1 của Act, có thể thấy rằng rõ ràng là việc truyền vào previous act thực sự đã cải thiện nhận diện Acts.
- Các điểm Intent F1, slot-value accuracy của việc không truyền vào previous act là cao hơn. Tuy nhiên, như em đã nói ở trên, việc có hay không truyền vào previous act không hề ảnh hưởng tới gán nhãn BIO hay intent detection, vì em chỉ truyền previous act vào **act_head**, không hề liên quan tới **bio_head**. Các điểm này cao hơn có thể là do tại epoch thứ 6 lúc train không có previous act, **bio_head** may mắn đạt được trạng thái tốt hơn.