

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN,  
ĐẠI HỌC QUỐC GIA HÀ NỘI



---

**Assignment 3: Dialogue Systems**

---

Ngày 31 tháng 12 năm 2024

**Giảng viên**

TS. Lê Hồng Phương

**Họ và tên**

Dương Nguyễn Việt Hải

**Mã sinh viên**

21002137

---

# 1 Tóm tắt lý thuyết:

## 1.1 Chatbot

Chatbot: Là hệ thống mô phỏng hội thoại giống con người, ví dụ ChatGPT. Chatbot tập trung vào các cuộc trò chuyện không có cấu trúc cụ thể.

Hai loại kiến trúc chatbot phổ biến:

- Rule-based: trả lời dựa trên câu trả lời được thiết lập sẵn khi đối chiếu với câu hỏi. Phải thiết lập cụ thể câu trả lời sẵn cho từng câu hỏi.
- Corpus-based: sử dụng dữ liệu mẫu lớn (corpus) để học các mẫu, mối quan hệ, hoặc quy tắc từ ngữ trong ngôn ngữ và sẽ tạo ra câu trả lời không cần thiết lập thủ công như Rule-based

Hệ thống hội thoại định hướng nhiệm vụ (Task-oriented dialogue systems): Được thiết kế để hoàn thành một nhiệm vụ cụ thể (vd: đặt vé máy bay, đặt lịch hẹn).

## 1.2 Đặc tính hội thoại ở người:

Bao gồm 3 phần chính:

- Turn: Lượt nói, A nói rồi đến B rồi quay lại A là lượt nói của một hội thoại cơ bản
- Speech acts: Hành động trong câu nói, ví dụ "Đặt vé máy bay" là một hành động yêu cầu.
- Grounding: Hiểu ý, xác định rằng đã hiểu ý của đối phương muốn nói gì

## 1.3 Hệ thống hội thoại:

Cấu trúc Frame với hai phần Frame và Slots:

- Frame: Chứa thông tin tổng ví dụ: Thông tin chuyên
- Slots: Chứa các thông tin nhỏ cụ thể ví dụ như: Ngày, giờ, địa điểm, tên hành khách,...

Quy trình hoạt động gồm 3 bước chính:

- Bước 1: Domain Classification giúp xác định chủ đề của cuộc hội thoại, xem xem đang nói về vấn đề gì,...
- Bước 2: Intent Detection hiểu ý định và xác định mục đích.

- 
- Bước 3: Slots filling từ thông tin trong hội thoại trích xuất và gắn vào các slot.

Đánh giá hiệu suất:

$$\text{Slot Error Rate for a Sentence} = \frac{\text{Tổng các slot bị điền sai}}{\text{Toàn bộ slot của câu}}$$

## 2 Mục tiêu

- Trích chọn dữ liệu (x, y): x là câu nói (utterance), y là dialogue act tương ứng với câu nói đó.
- Cài đặt thuật toán sequence labelling
- Huấn luyện, đánh giá mô hình phát hiện dialogue acts

## 3 Tiến hành thực tiễn:

### 3.1 Tổng quan dữ liệu:

Để hiểu dữ liệu ta xem xét file data.json như sau:

- Mỗi đối tượng JSON là một cuộc hội thoại.
- Mỗi cuộc hội thoại có một mảng các lượt hội thoại.
- Trong mỗi hội thoại sẽ bao gồm:
  - speaker: là người nói câu đó có thể là user hoặc system.
  - utterance: nội dung câu thoại
  - frames: một mảng các đối tượng JSON, mỗi đối tượng tương ứng với một "service"
  - state: (trong mỗi frame) chứa thông tin về active\_intent

### 3.2 Task 1: Trích xuất dữ liệu:

Yêu cầu: Trích chọn dữ liệu (x, y): x là câu nói (utterance), y là dialogue act tương ứng với câu nói đó.

Hướng giải quyết:

- Hàm này duyệt qua từng đối tượng hội thoại trong data. Với mỗi lượt hội thoại, nó kiểm tra nếu speaker là "USER". Lấy utterance.

- Duyệt qua các frames và trích xuất active\_intent từ state của frame tương ứng nếu active\_intent khác "NONE".
- Tạo cặp (utterance, dialogue\_act) và thêm vào danh sách samples.
- Trả về các cặp x,y tương ứng.

```
Utterance: I am looking for the Addenbrookes Hospital with hepatology department
Dialogue Act: ['hospital-find_hospital']
-----
Utterance: I'm planning a trip to cambridge and need a place to dine. Can you find something that serves jamaican food in the centre?
Dialogue Act: ['restaurant-find_restaurant', 'hotel-find_hotel']
-----
Utterance: Are you sure? It should be expensive.
Dialogue Act: ['restaurant-find_restaurant', 'hotel-find_hotel']
-----
Utterance: I really want jamaican food. Can you check another area? If you find one it should be expensive. If not I'll try Thai in the centre.
Dialogue Act: ['restaurant-find_restaurant', 'hotel-find_hotel']
-----
Utterance: No thanks. Just go ahead and book a table for 3 people at 14:30 on thursday.
Dialogue Act: ['restaurant-book_restaurant']
-----
Utterance: I'm sorry to be a bother. Instead of that reservation, I would be ok with British food. Is there a British place in the centre?
Dialogue Act: ['restaurant-book_restaurant', 'hotel-find_hotel']
-----
Utterance: Okay, can I please have a reservation for Midsummer House Restaurant? Thank you so much for your patience with me!
Dialogue Act: ['restaurant-book_restaurant']
-----
Utterance: i am also looking for a place to stay in the centre that offers free wifi.
Dialogue Act: ['restaurant-book_restaurant', 'hotel-find_hotel']
```

Hình 1: Kết quả

### 3.3 Task 2: Phát hiện hành động trong hội thoại

Mô hình sử dụng LSTM

Cách hoạt động:

- Đầu vào x (một batch các chuỗi index từ vựng) được chuyển đổi thành vector embedding tương ứng.
- Các vector embedding được đưa vào lớp LSTM. Lớp LSTM trả về hai thành phần:
  - Chuỗi output của lớp LSTM tại mỗi vị trí thời gian.
  - Trạng thái ẩn (hidden state) của LSTM ở lớp cuối cùng, chứa thông tin tóm tắt về chuỗi đầu vào.
- Ghép nối trạng thái ẩn của hai chiều ở lớp cuối cùng theo chiều thứ nhất (chiều hidden\_dim). Việc ghép nối này tạo ra một vector tổng hợp chứa thông tin từ cả hai chiều.
- Trạng thái ẩn đã được ghép nối được đưa qua lớp fully connected để tạo ra kết quả dự đoán nhãn cho chuỗi.

Thiết lập thêm earlystopping để huấn luyện mô hình thoải mái với lượng lớn epoch đến khi đạt yêu cầu.

```
Epoch 7/20, Training Loss: 0.09116062521934509, Validation Loss: 0.1289001529475049, F1: 0.7940585322809942
Early stopping at epoch 7
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
C:\Users\vieth\AppData\Local\Temp\ipykernel_9240\3553865515.py:83: FutureWarning: You are using `torch.load` with `weights_only=False` (the current def
model.load_state_dict(torch.load('best_model.pth'))
Sample predictions:
Predicted Labels: ['train-find_train']
Actual Labels: ['train-find_train']
Predicted Labels: []
Actual Labels: ['hotel-find_hotel', 'restaurant-find_restaurant']
Predicted Labels: ['hotel-book_hotel']
Actual Labels: ['restaurant-find_restaurant']
Predicted Labels: ['hotel-find_hotel']
Actual Labels: ['hotel-find_hotel']
Predicted Labels: ['train-find_train']
Actual Labels: ['train-find_train']
Final F1 score: 0.7944314602174576
```

Hình 2: Kết quả

### 3.4 Task 3: Trích xuất slot/value

Cấu trúc dữ liệu:

- data.json: Chứa thông tin về các lượt nói và metadata.
- dialogue\_acts.json: Chứa thông tin về các action và slot cho các câu hội thoại.
- valid\_files.txt, test\_files.txt: Chứa danh sách các tên file JSON cho tập valid và test.

Gắn nhãn BIO (Begin, Inside, Other)

- Kỹ thuật gắn nhãn BIO:
  - Mục đích để xác định vị trí và ranh giới của các thực thể (entities) trong câu.
  - Có ba loại tag chính: B-slot ( bắt đầu của một slot), I-slot ( bên trong của một slot), O ( từ không thuộc slot nào)
- Cách gắn nhãn BIO cho từng slot:
  - Trong quá trình xử lý dữ liệu, xác định các vị trí của slot trong câu bằng cách so sánh với giá trị của slot trong acts dictionary.
  - Sau khi xác định được vị trí, các tag tương ứng (B- hoặc I-) được gán cho các từ trong giá trị slot.
  - Các từ còn lại trong câu sẽ được gán tag "O".

Mô hình sử dụng: LSTM

```
Epoch: 4, Train Loss: 0.0858, Val Loss: 0.0681, F1 Score: 0.9750, Slot Error Rate: 0.1850
Epoch: 5, Train Loss: 0.0752, Val Loss: 0.0593, F1 Score: 0.9777, Slot Error Rate: 0.1633
Epoch: 6, Train Loss: 0.0656, Val Loss: 0.0510, F1 Score: 0.9806, Slot Error Rate: 0.1397
Epoch: 7, Train Loss: 0.0570, Val Loss: 0.0420, F1 Score: 0.9849, Slot Error Rate: 0.1101
Epoch: 8, Train Loss: 0.0490, Val Loss: 0.0361, F1 Score: 0.9869, Slot Error Rate: 0.0957
Epoch: 9, Train Loss: 0.0420, Val Loss: 0.0303, F1 Score: 0.9893, Slot Error Rate: 0.0783
Epoch: 10, Train Loss: 0.0357, Val Loss: 0.0258, F1 Score: 0.9909, Slot Error Rate: 0.0669
Epoch: 11, Train Loss: 0.0314, Val Loss: 0.0214, F1 Score: 0.9930, Slot Error Rate: 0.0503
Epoch: 12, Train Loss: 0.0268, Val Loss: 0.0187, F1 Score: 0.9938, Slot Error Rate: 0.0455
Epoch: 13, Train Loss: 0.0237, Val Loss: 0.0166, F1 Score: 0.9948, Slot Error Rate: 0.0381
Epoch: 14, Train Loss: 0.0210, Val Loss: 0.0154, F1 Score: 0.9951, Slot Error Rate: 0.0353
Epoch: 15, Train Loss: 0.0187, Val Loss: 0.0134, F1 Score: 0.9958, Slot Error Rate: 0.0305
Epoch: 16, Train Loss: 0.0179, Val Loss: 0.0120, F1 Score: 0.9963, Slot Error Rate: 0.0273
Epoch: 17, Train Loss: 0.0157, Val Loss: 0.0104, F1 Score: 0.9968, Slot Error Rate: 0.0238
Epoch: 18, Train Loss: 0.0153, Val Loss: 0.0108, F1 Score: 0.9966, Slot Error Rate: 0.0250
Early stopping

Sample sentence: I want to book a hotel in the east that has 4 stars
Predicted tags: ['o', 'o', 'o', 'o', 'o', 'o', 'o', 'o', 'o', 'o', 'B-stars', 'o']
Extracted slots: [{'slot': 'Stars', 'value': '4'}]
```

### Hình 3: Kết quả

**F1-Score:** 0.9966

**SER:** 0.0250

## Tham khảo

Bài làm tham khảo cách gắn nhãn BIO cho task 3 từ link kaggle: "<https://www.kaggle.com/code/.../slot/value>"