Dialogue State Tracking

Hai Pham-Ngoc

Tóm tắt nội dung

Theo dõi trạng thái hội thoại (DST) là nhiệm vụ then chốt trong hệ thống hội thoại hướng tác vụ, nhằm theo dõi và cập nhật theo thời gian thực ý định người dùng và các tham số (slots). Báo cáo này trình bày **hai phương pháp** DST: (1) mô hình học sâu kết hợp phân loại đa nhãn hành động và phân loại token, sử dụng embedding (GloVe, TinyBERT) và LSTM, được huấn luyện trên dữ liệu MultiWOZ 2.2; (2) phương pháp dựa trên Mô hình Ngôn ngữ Lớn (LLM) Gemini 1.5 Flash, tối ưu prompt để trích xuất trạng thái hội thoại như bài toán đọc hiểu, không yêu cầu huấn luyện lại và dễ mở rộng. Báo cáo này cũng phân tích các thách thức DST (mở rộng miền, giá trị không biết, chi phí gán nhãn), tổng quan các hướng nghiên cứu (dựa trên/không dựa trên ontology, MRC, In-Context Learning), mô tả chi tiết hai phương pháp đề xuất, và trình bày kết quả thực nghiệm trên MultiWOZ 2.2, so sánh hiệu suất và kết luận về tiềm năng của LLM cho DST. **Source code** tại: GitHub. **Báo cáo cụ thể hơn** tại: Báo cáo chi tiết hơn.

1 Giới thiệu

1.1 Về bài toán Theo dõi trạng thái hội thoại (DST)

Theo dõi trạng thái hội thoại (DST) là thành phần then chốt trong hệ thống hội thoại hướng tác vụ, nhằm theo dõi và cập nhật trạng thái hội thoại theo thời gian thực dựa trên tương tác giữa người dùng và hệ thống, hỗ trợ đưa ra quyết định và phản hồi phù hợp. Trạng thái hội thoại bao gồm:

- **Ý định người dùng (User Intent):** Mong muốn/mục đích (ví dụ: đặt vé máy bay, tìm nhà hàng).
- **Tham số (Slots/Entities):** Chi tiết liên quan đến ý định (ví dụ: điểm đi, điểm đến, thời gian).

Ví dụ: "Tôi muốn đặt vé máy bay từ Hà Nội đi Sài Gòn vào ngày mai"→ Ý định: Đặt vé máy bay; Điểm đi: Hà Nội; Điểm đến: Sài Gòn; Ngày đi: Ngày mai.

1.2 Các thách thức chính trong nghiên cứu DST

Các thách thức chính bao gồm:

- **Khả năng mở rộng miền:** Yêu cầu phương pháp không dựa trên ontology hoặc học liên tục.
- **Xử lý giá trị không biết:** Cần phương pháp không ontology hoặc học chuyển tiếp.
- **Giảm chi phí gán nhãn:** Sử dụng học bán giám sát/không giám sát.

2 Công Việc Liên Quan

Dưới đây gồm 2 phần chính là các **hướng nghiên cứu điển hình hiện tại**, cũng như các **hướng nghiên cứu tiềm năng trong tương lai**.

2.1 Các Hướng Nghiên Cứu Điển Hình

Bảng 1 tóm tắt 2 hướng tiếp cận điển hình trong nghiên cứu DST.

Bảng 1: Tóm tắt các hướng phương pháp nghiên cứu điển hình về DST

Loại Phương Pháp	Ưu Điểm	Nhược Điểm	Ứng Dụng Chính
Dựa trên ontology	Đơn giản, hiệu quả	Hạn chế trong mở rộng miền	Hệ thống cố định miền
Không dựa trên ontology	Linh hoạt, mạnh mẽ	Khó kiểm soát chất lượng	Hệ thống mở miền

Mô Hình Dựa Trên Bản Thể (Ontology-Based Models)

Phương Pháp Dựa Trên Ontology (Ontology-Based Methods):

- **Ý tưởng:** Các phương pháp này sử dụng một ontology chứa tập giá trị slot được xác định trước, từ đó dự đoán giá trị slot dựa trên danh sách các giá trị khả dĩ.
- **Ưu điểm:** Phù hợp với các hệ thống có miền xác định rõ ràng, đơn giản và dễ triển khai.
- **Nhược điểm:** Hạn chế trong việc mở rộng sang miền mới và xử lý các giá trị chưa biết.

Nghiên cứu cụ thể: Các nghiên cứu về phương pháp dựa trên bản thể tĩnh bao gồm: [Lim et al., 2023], [Zhu et al., 2022], [Ouyang et al., 2020], [Zhao et al., 2021], [Zhu et al., 2020], [Gao et al., 2020], [Wang et al., 2022], [Lin et al., 2021], [Huang et al., 2020], [Zeng and Nie, 2020], [Yang et al., 2023], [Jia et al., 2024], [Le et al., 2020b], [Hu et al., 2020], [Heck et al., 2020]. Trong đó:

- [Feng et al., 2020] đề xuất Seq2Seq-DU, dùng BERT mã hóa câu nói và lược đồ, bộ giải mã tạo con trỏ trạng thái hội thoại, so sánh với các phương pháp DST khác.
- [Jeon and Lee, 2021] đề cập việc sử dụng BERT trong hệ thống hội thoại hướng mục tiêu, một số có thể dùng phương pháp dựa trên bản thể.
- [Le et al., 2020a] phân loại mô hình DST thành dựa trên từ vựng cố định (tương đương dựa trên bản thể) và từ vựng mở.

Phương Pháp Không Cần Ontology (Ontology-Free Methods)

Ý tưởng: Trích xuất giá trị slot trực tiếp từ ngữ cảnh hội thoại, không cần tập giá trị xác định trước.

Ưu điểm: Linh hoạt, xử lý được các giá trị chưa biết, phù hợp với các hệ thống mở miền.

Nhược điểm: Khó kiểm soát chất lượng trích xuất giá trị slot.

Nghiên cứu cụ thể: Các nghiên cứu về phương pháp không cần ontology bao gồm:

- [Feng et al., 2020] giới thiệu COMER, CREDIT và SimpleTOD, coi DST là bài toán tạo chuỗi, dùng kiến trúc encoder-decoder. COMER dùng encoder-decoder phân cấp dựa trên BERT; CREDIT dùng encoder-decoder phân cấp không BERT; SimpleTOD dùng mô hình chuỗi-chuỗi dựa trên GPT-2.
- [Zhu et al., 2022] đề cập phương pháp không ontology để giảm phụ thuộc ontology và cải thiện khái quát hóa. Một số phương pháp trích xuất giá trị từ ngữ cảnh, số khác tạo giá trị trực tiếp.
- [Xu et al., 2023] chia phương pháp DST thành dựa trên/không dựa trên ontology. Phương pháp không ontology tạo giá trị slot từ ngữ cảnh hoặc từ vựng.
- [Ouyang et al., 2020] thảo luận phương pháp không ontology (từ vựng mở), cho phép tạo giá trị slot chỉ với slot mục tiêu.
- [Zhao et al., 2021] so sánh phương pháp đề xuất với nhiều phương pháp cơ sở, bao gồm cả không ontology.
- [Zhu et al., 2020] thảo luận phương pháp DST không ontology, dùng encoder-decoder hoặc mạng con trỏ để tạo/trích xuất giá trị từ ngữ cảnh.

- [Le et al., 2020a] phân loại mô hình DST thành từ vựng cố định và từ vựng mở (tương đương không ontology).
- [Gao et al., 2020] thảo luận phương pháp không ontology, bao gồm dùng cơ chế trỏ dựa trên attention để tìm vị trí giá trị slot.
- [Hu et al., 2022] đề xuất khung Học Trong Ngữ Cảnh (ICL) cho DST zero-shot và few-shot, dùng LM dự đoán giá trị slot từ ngữ cảnh, không cần ontology.
- [Wang et al., 2022] phân loại phương pháp DST thành phân loại và tạo. Phương pháp tạo thường không ontology, tạo trạng thái hội thoại bằng seq2seq.
- [Huang et al., 2020] thảo luận phương pháp DST không ontology, dùng bộ tạo trạng thái tạo giá trị slot từ ngữ cảnh.
- [Zeng and Nie, 2020] tập trung vào DST không ontology, giá trị không xác định trước, cần được trích xuất/tạo trực tiếp từ đầu vào.
- [Yang et al., 2023] chủ yếu tập trung vào DST dựa trên bản thể, nhưng cũng đề cập việc áp dụng attention cho mô hình tạo (có thể không ontology).
- [Jia et al., 2024] thảo luận mô hình không ontology (từ vựng mở), dựa vào tạo/trích xuất giá trị slot từ lịch sử và trạng thái hội thoại.
- [Le et al., 2020b] đề xuất phương pháp không ontology, dùng bộ giải mã không tự hồi quy tạo tất cả token trạng thái hội thoại cùng lúc.
- [Hu et al., 2020] tập trung vào chia sẻ thông tin slot trong DST dựa trên bản thể, nhưng kỹ thuật này cũng có thể áp dụng cho phương pháp không ontology.
- [Heck et al., 2020] đề xuất phương pháp không ontology, dùng chiến lược sao chép ba lần dựa vào dự đoán span và cơ chế bộ nhớ.
- [Campagna et al., 2020] tập trung vào học chuyển giao zero-shot cho DST đa miền, thường không cần ontology.

2.2 Các Hướng Nghiên Cứu Mới

Dựa Trên Hiểu Đọc Máy (MRC)

- DST được xử lý như bài toán đọc hiểu, với lịch sử hội thoại là đoạn văn và slot là câu hỏi.
- [Gao et al., 2020] chuyển đổi DST thành MRC bằng cách thiết kế câu hỏi cho mỗi slot, phân loại slot thành categorical (xử lý bằng MRC trắc nghiệm) và extractive (xử lý bằng MRC span-based).
- [Le et al., 2020a] và [Le et al., 2020b] đề cập đến mô hình DST Reader, coi DST là bài toán đọc hiểu, dự đoán mỗi slot như một khoảng văn bản trong lịch sử hội thoại, sử dụng mạng nơ-ron dựa trên attention.

Phương Pháp Học Trong Ngữ Cảnh (In-Context Learning)

- Sử dụng LLM (như GPT) cho DST zero-shot/few-shot, giải quyết các miền chưa thấy.
- [Hu et al., 2022] đề xuất khung Học Trong Ngữ Cảnh (ICL) cho DST zero-shot/few-shot. LM nhận instance kiểm tra và ví dụ, giải mã trực tiếp trạng thái hội thoại mà không cập nhật tham số, không cần ontology vì LM học dự đoán giá trị slot từ ngữ cảnh. Bài báo giới thiệu mô hình IC-DST, kết hợp: chuyển DST thành text-to-SQL (tích hợp mô tả ontology vào prompt); dùng trạng thái hội thoại (thay vì lịch sử) biểu diễn ngữ cảnh; đề xuất phương pháp học điểm tương đồng chọn ví dụ trong ngữ cảnh.

2.2.1 Các Xu Hướng Nghiên Cứu Mới Khác

Bảng 2: Các Xu Hướng Nghiên Cứu Mới Khác

Xu Hướng Nghiên Cứu
Học Tập Ít Mẫu (Giảm phụ thuộc vào dữ liệu gắn nhãn nhờ meta-learning và fine-tuning)
Học Tập Liên Tục (Phát triển mô hình thích nghi với miền mới mà không cần huấn luyện lại)
DST Đa Phương Thức (Kết hợp thông tin từ văn bản, hình ảnh, âm thanh)
Xử Lý Nhiễu Dữ Liệu (Áp dụng các framework kháng nhiễu)

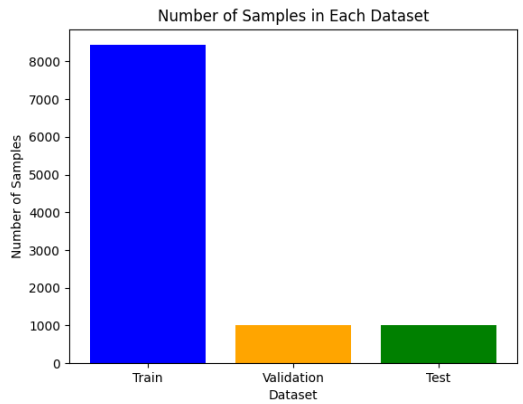
3 Về Dữ Liệu

Sự phát triển của các tập dữ liệu hội thoại đa miền quy mô lớn, chẳng hạn như MultiWOZ 2.0, đã góp phần đáng kể vào việc thúc đẩy nghiên cứu trong lĩnh vực Hệ thống Đối thoại (DST). Tuy nhiên, các phiên bản đầu tiên của MultiWOZ chứa nhiều nhiễu trong các chú thích trạng thái, gây khó khăn trong việc đánh giá hiệu suất của các mô hình một cách chính xác và công bằng.

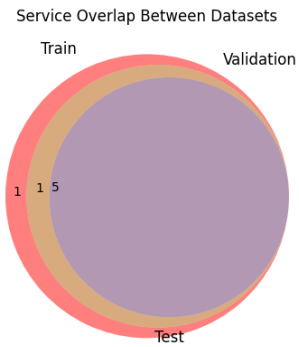
Để khắc phục vấn đề này, MultiWOZ 2.4 được giới thiệu như một phiên bản tinh chỉnh của MultiWOZ 2.1, tập trung vào việc sửa chữa các lỗi chú thích trong các tập validation và test. Các nghiên cứu cho thấy rằng MultiWOZ 2.4 mang lại hiệu suất cao hơn cho các mô hình DST tiên tiến và cung cấp một môi trường đánh giá chính xác và đáng tin cậy hơn.

Tuy nhiên, do MultiWOZ 2.4 (cùng với phiên bản 2.3) chưa được nhóm phát hành gốc cập nhật nhanh chóng, họ vẫn đang duy trì và phát triển phiên bản 2.2. Vì vậy, nghiên cứu này sẽ tập trung vào phiên bản 2.2 của MultiWOZ.

Dữ liệu của các tập train, validation và test trong MultiWOZ 2.2 được giữ nguyên từ các phiên bản trước. Điều này tạo điều kiện thuận lợi cho việc so sánh các kết quả giữa các phiên bản khác nhau. Thông tin chi tiết về phân chia các tập train, validation và test của MultiWOZ 2.2 được trình bày dưới đây:



(a) Số lượng phân chia train, validation, test



(b) Sự giao nhau giữa các miền trong các tập dữ liệu

Hình 1: Phân bố dữ liệu và sự giao nhau giữa các miền

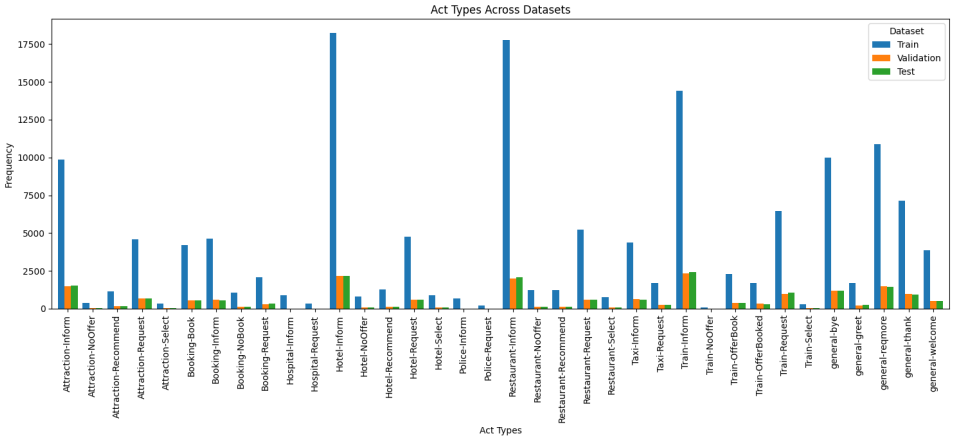
Nhận xét: Hình ảnh 1a thể hiện số lượng mẫu trong từng tập dữ liệu (train, validation, test) một cách trực quan. Tập train có số lượng mẫu lớn nhất với gần 9,000 mẫu, trong khi các tập validation và test có số lượng mẫu nhỏ hơn nhiều, chỉ khoảng 1,000 mẫu mỗi tập. Sự chênh lệch này cho thấy việc huấn luyện mô hình sẽ có nhiều dữ liệu hơn để học hỏi, trong khi việc đánh giá hiệu suất có thể gặp khó khăn do số lượng mẫu hạn chế trong các tập validation và test. Điều này có thể ảnh hưởng đến tính chính xác của việc đánh giá mô hình nếu không thực hiện cẩn thận.

Các miền trong MultiWOZ 2.2 bao gồm: Nhà hàng (restaurant), Khách sạn (hotel), Điểm tham quan (attraction), Tàu (train), Taxi (taxi), Bệnh viện (hospital), Cảnh sát (police), Xe buýt (bus), Đặt chỗ (booking), Thông tin chung (general)

Sự giao nhau giữa các miền này trong ba tập dữ liệu train, validation và test được thể hiện trong hình 1b.

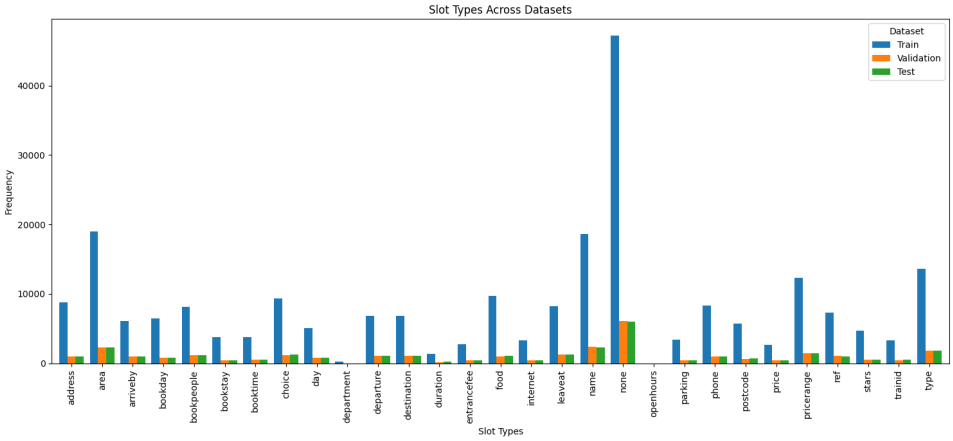
Hình ảnh minh họa sự chồng chéo giữa các miền trong các tập dữ liệu train, validation và test. Cụ thể, tập test chỉ chứa một lượng nhỏ dữ liệu từ ít miền hơn.

Các loại hành động (act types) có mặt trong các cuộc hội thoại như sau:



Hình 2: Phân phối hành động trong các tập train, validation, test

Ngoài ra, phân phối các slot trong các tập dữ liệu cũng được minh họa như sau:



Hình 3: Phân phối slot trong các tập train, validation, test

Nhìn chung, phân phối các hành động giữa các nhãn là khá đồng đều, trong khi phân phối giữa các nhãn slot thì không đồng đều. Điều này cho thấy rằng mô hình phân loại slot có thể đạt được macro F1 score thấp, nhưng nếu tính trong weighted F1 score, kết quả sẽ cao hơn do mô hình ưu tiên các nhãn có tần suất xuất hiện cao hơn.

3.1 Phân Chia Dữ Liệu

Dữ liệu được phân tách từ các cuộc hội thoại thành các turn. Mỗi câu nói sẽ tương ứng với một danh sách các act types, cùng với một từ điển chứa các slot và giá trị của chúng. Khi mã hóa các câu này, mỗi từ sẽ được tách ra dưới dạng các token cách nhau bằng khoảng trắng, sau đó gán nhãn BIO cho từng từ.

Kết quả thu được sau khi xử lý dữ liệu sẽ có dạng:

- **Vector X:** ['Token 1', 'Token 2', ..., 'Token n', 'Padding', ..., 'Padding'] - một câu (USER hoặc SYSTEM)
- **Vector Y_act:** [0, 0, 1, 1, 0, ..., 1] - vector dài 36 cho 36 hành động có thể xảy ra.
- **Vector Y_slot:** [[O, O, ..., B-..., I-..., ...], ..., [...]] - ma trận có kích thước (số_token * 53 slot)

Chi tiết về tiền xử lý dữ liệu và phân chia có thể tham khảo tại GitHub.

4 Phương pháp

4.1 Tóm tắt

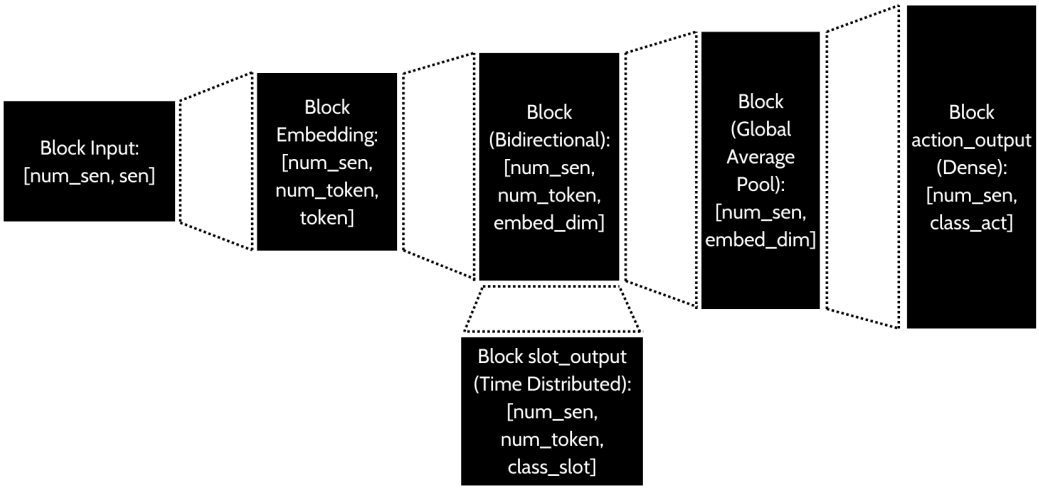
Trong nghiên cứu này, em triển khai hai phương pháp chính như sau:

- **Phương pháp 1:** Sử dụng dữ liệu đã phân chia thành train, validation và test để huấn luyện và kiểm thử mô hình. Phương pháp huấn luyện được chọn kết hợp hai bài toán:
  - Phân loại câu (Sentence Classification) - bài toán xác định act type: phân loại đa nhãn (multi-label classification).
  - Phân loại token (Token Classification hoặc Seq2Seq) - bài toán xác định loại slot của token.

  Thay vì tạo ra hai mô hình độc lập, em sẽ xây dựng một mô hình tổng hợp thực hiện đồng thời cả hai nhiệm vụ trên. Cách tiếp cận này sẽ tiết kiệm nguồn lực tính toán và giải quyết cả hai nhiệm vụ một cách đồng thời, tuy nhiên, mô hình có thể khó tối ưu hóa hơn vì phải giải quyết hai nhiệm vụ cùng lúc.
- **Phương pháp 2:** Sử dụng LLM (hoặc SLM) để thực hiện giống như ý tưởng đọc hiểu, sau đó thực hiện hai bài toán trắc nghiệm: chọn nhiều đáp án (cho bài toán xác định act), chọn một (hoặc không chọn) đáp án (cho mỗi slot). Ưu điểm của phương pháp này là tận dụng được sức mạnh của mô hình ngôn ngữ đã được huấn luyện trên nhiều nguồn tài liệu lớn. Do đó, không cần phải huấn luyện lại, chỉ cần tối ưu prompt. Phương pháp này cũng cho phép mở rộng mô hình nhanh chóng cho nhiều lĩnh vực khác nhau khi có cập nhật mới. Ngoài ra, phương pháp này có thể áp dụng cho nhiều ngôn ngữ khác nhau, không chỉ tiếng Anh (ví dụ: bộ dữ liệu MultiWOZ hiện tại chỉ hỗ trợ tiếng Anh).

4.2 Phương pháp 1: Huấn luyện mô hình bài toán phân loại đa nhãn và phân loại class cho từng token

Hình 4 mô tả mô hình hai nhánh kết hợp cho cả hai nhiệm vụ xác định act và xác định slot.



Hình 4: Mô hình hai nhánh kết hợp cho bài toán xác định act và xác định slot

Quy trình tổng quan là từ n câu đầu vào, qua block Embedding để thu được ma trận ba chiều [n_câu, n_token_mỗi_câu, d_embed_token]. Sau đó, dữ liệu này được đưa qua block Bidirectional, thu được ma trận ba chiều [n_câu, n_token_mỗi_câu, d_embed_mới_token]. Mô hình sau đó chia thành hai nhánh:

- Nhánh 1 thực hiện pooling về hai chiều [n_câu, d_embed_câu] và qua lớp Dense để phân loại đa nhãn (block `action_output`).
- Nhánh 2 vẫn giữ dạng ba chiều [n_câu, n_token_mỗi_câu, vector_phân_loại_class_slot].

Cài đặt Hàm Loss
Hàm loss cho mỗi đầu ra có thể được định nghĩa như sau:

$$\text{Loss}_{\text{hành động}} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$
$$\text{Loss}_{\text{slot}} = -\sum_{j=1}^{M} y_j \log(\hat{y}_j)$$

Trong đó:
- N là số lớp cho đầu ra hành động.
- M là số token cho đầu ra slot.
- y_i và y_j là nhãn thật.
- $\hat{y}_i$ và $\hat{y}_j$ là xác suất dự đoán.

Tổng hàm loss cho mô hình hybrid có thể được tính toán như một tổng trọng số của cả hai hàm loss:

$$\text{Tổng Loss} = \alpha \cdot \text{Loss}_{\text{hành động}} + \beta \cdot \text{Loss}_{\text{slot}}$$

Trong đó α và β là các siêu tham số để cân bằng sự đóng góp của mỗi hàm loss.
Chi tiết về mô hình S1:
Chi tiết về quá trình huấn luyện và kiểm thử có thể tham khảo tại GitHub.

- Sử dụng Block Embedding là Glove 50 chiều, số lượng token tối đa cho mỗi câu là 50.
- Block Bidirectional sử dụng một lớp LSTM với số lượng unit là 128.

Mô hình của S1 có cấu trúc như sau:

Bảng 3: Cấu trúc mô hình S1

Layer (type)	**Output Shape**	**Param #**
input_layer_1	(None, 50)	0
embedding_1	(None, 50, 50)	730,150
bidirectional_1	(None, 50, 128)	58,880
global_average_pooling	(None, 128)	0
slot_output	(None, 50, 53)	6,837
action_output	(None, 24)	3,096

Tổng số tham số: 798,963 (3.05 MB). Tham số huấn luyện được: 68,813 (268.80 KB). Tham số không huấn luyện: 730,150 (2.79 MB).
Chi tiết về mô hình S2:

- Sử dụng Block Embedding là TinyBert của Huawei với số lượng token tối đa cho mỗi câu là 20 để giảm chi phí tính toán, và số chiều cho mỗi token là 312.
- Block Bidirectional sử dụng một lớp LSTM nhưng với số unit là 256.

Chi tiết về kiến trúc có thể tham khảo tại GitHub. Việc chạy huấn luyện và kết quả kiểm thử có thể thực hiện nhờ main.py tại folder root.

```
pip install -r requirements.txt
nohup python main.py --tasks download_data > download_data.log 2>&1 &
nohup python main.py --tasks process_and_save > process_and_save.log 2>&1 &
nohup python main.py --tasks load_process_and_train > load_process_and_train.log 2>&1 &
nohup python main.py --tasks evaluate > evaluate.log 2>&1 &
nohup python main.py --tasks demo #> demo.log 2>&1 &
```

Mô hình của S2 có cấu trúc như sau:

Bảng 4: Cấu trúc mô hình

Layer (type)	**Output Shape**	**Param #**
input_layer	(None, 20, 312)	0
bidirectional	(None, 20, 256)	451,584
dropout	(None, 20, 256)	0
global_average_pooling	(None, 256)	0
dropout_1	(None, 256)	0
slot_output	(None, 20, 53)	13,621
action_output	(None, 24)	6,168

Tổng số tham số: 471,373 (1.80 MB). Tham số huấn luyện được: 471,373 (1.80 MB). Tham số không huấn luyện: 0.

Tuy nhiên, tổng tham số này chưa tính đến lớp embedding TinyBert của Huawei (4 lớp Transformer: 14.5 triệu tham số), vì vậy tổng số tham số của mô hình S2 gần 15 triệu.

Mô hình S1 đơn giản hơn mô hình S2 (bao gồm cả bước tokenizer + embedding và cho qua lớp Bidirectional). Tuy nhiên, cả hai mô hình S1 và S2 vẫn là các mô hình nhỏ khi so với các tác vụ xử lý ngôn ngữ tự nhiên phức tạp. Đặc biệt trước sự nhập nhằng trong quá trình gán nhãn (4 phiên bản của MultiWOZ 2 đều đang cố gắng cải thiện lại việc gán nhãn dữ liệu). Do đó, cả hai mô hình này chỉ là "small model"để kiểm thử cách thuật toán hoạt động. **Sẽ rất khó để hai "small model"này có thể đạt lại các kết quả SOTA.**

4.3 Phương pháp 2: Hướng tiếp cận sử dụng LLM (hoặc SLM)

Thay vì phải xây dựng và huấn luyện mô hình từ bộ dữ liệu còn hạn chế (chỉ một ngôn ngữ là tiếng Anh, chiến lược gán nhãn dữ liệu chưa đủ tốt), phương pháp này sẽ coi toàn bộ hội thoại như một đoạn văn đọc hiểu. Tại mỗi turn, LLM sẽ thực hiện bài toán đọc hiểu và đưa vào prompt đã yêu cầu để giải quyết câu hỏi.

- Chọn một (hoặc nhiều) hoặc không chọn hành động nào cho câu turn đó.
- Chọn một (hoặc không chọn) giá trị cụ thể cho các slot.

Ưu điểm của cách này như đã đề cập ở trên:

- Khắc phục hạn chế trong việc huấn luyện mô hình, thu thập, xử lý, và gán nhãn dữ liệu.
- Khi cần mở rộng chương trình sang nhiều lĩnh vực khác, hoặc khi cần thêm, xóa loại hành động, hoặc khi cần thêm, xóa loại slot thì chỉ cần thay đổi prompt, không cần huấn luyện lại mô hình.
- LLM hiện đại nhất sẽ được cập nhật liên tục bởi lượng dữ liệu khổng lồ từ Internet.

Em sử dụng Gemini 1.5 Flash của Google. Chi tiết về mã triển khai mô hình có thể xem tại GitHub. Lưu ý rằng yêu cầu khi chạy là cần tải API key trước. Prompt được đưa ra như sau:

```
prompt = f"""
    You are a system that tracks conversation states.
    Given the following conversation history and current turn,
    extract the current conversation state as a JSON dictionary:
```

```
    Act Type List:
    {self.act_type_list}

    Slot Type List:
    {self.slot_type_list}

    Conversation History:
    {self.dialog_history}

    Current Turn:
    {current_turn}

    Return Conversation state (JSON) for current turn:
    {{act_type: [act_value, ...],
    slot_type: {{slot_1: slot_value_1, slot_2: slot_value_2, ...}}}}
"""
```

5 Results

Nhìn chung cả phương pháp 1 và phương pháp 2 được dùng trong báo cáo này đều là không sử dụng ontology. Phương pháp 1 sử dụng embedding (GloVe và TinyBERT) và LSTM để giải quyết 2 nhiệm vụ phân loại act type và slot type. Phương pháp 2 sử dụng LLM để giải quyết 2 nhiệm vụ này thông qua tối ưu hóa prompt mà không trải qua quá trình huấn luyện.

5.1 Phương pháp 1:

Với bài toán slot classification - Seq2Seq:

	S1 model	S2 model
Accuracy	0.96	0.93
Macro F1	0.10	0.41
Weighted F1	0.94	0.92

Bảng 5: Kết quả Slot Classification

Với bài toán act classification - Multi-label classification:

	S1 model	S2 model
Accuracy	0.21	0.28

Bảng 6: Kết quả Act Classification

Bình luận: Kết quả thu được trên tập test của MultiWOZ 2.2 cho thấy:

- Nhìn chung model S2 với 15 triệu tham số cho kết quả tốt hơn so với model S1 với 800k tham số.
- Do các nhãn dữ liệu mất cân bằng nên macro F1 thấp, trong khi weighted F1 cao. Cần cải thiện model để cân bằng các nhãn thiểu số hơn.
- Độ chính xác phân loại đa nhãn cho bài toán act type classification cũng thấp. Một mặt vì chưa sử dụng vector hóa toàn bộ câu mà mới chỉ dùng các token riêng lẻ. Mặt khác vì là model kết hợp 2 nhánh nên sẽ khó tối ưu hơn cho 1 nhánh cụ thể của model.
- Giải pháp ở đây là cải thiện hàm loss tổng thể để model tập trung học tốt bên nhánh act type nhiều hơn.

5.2 Phương pháp 2:

Việc sử dụng prompt hợp lý với 1 model tạo sinh LLM cho kết quả (cảm nhận) tốt hơn so với phương pháp 1. Lý do đầu tiên chắc chắn do số lượng tham số nhiều hơn, model đã được học qua rất nhiều văn bản của nhiều ngôn ngữ khác nhau rồi.

Sau đây là kết quả demo việc sử dụng phương pháp 2.

Cho ngôn ngữ là tiếng Việt:

```
python main.py --tasks use_llm
Input current turn (or type 'q' to quit): Chào hệ thống.
Tôi muốn đặt dịch vụ cho thứ Hai tới.
Tôi sẽ đi tàu từ Hải Dương đến Hà Nội lúc 9h sáng,
sau đó khám bệnh tại bệnh viện 108 vào lúc 15h, khám xong tầm 18h.
Sau đó tôi muốn đặt một nhà hàng Pháp ở phố cổ cho bữa tối.
Extracted dialog state for current turn:
{
  "act_type": ["Train-Inform", "Hospital-Inform", "Restaurant-Inform"],
  "slot_type": {
    "day": "thứ Hai tới",
    "departure": "Hải Dương",
    "destination": "Hà Nội",
    "leaveat": "9h sáng",
    "bookday": "thứ Hai tới",
    "department": "khám bệnh",
    "booktime": "15h",
    "type": "Pháp",
    "area": "phố cổ",
    "food": "bữa tối"
  }
}
```

Cho ngôn ngữ là tiếng Anh:

```
python main.py --tasks use_llm
Input current turn (or type 'q' to quit): Hello system.
I'd like to book some services for next Monday.
I'll be taking the train from Hai Duong to Hanoi at 9 am,
then I have a medical appointment at Hospital 108 at 3 pm,
which should finish around 6 pm.
After that, I'd like to book a French restaurant in the Old Quarter for dinner.
Extracted dialog state for current turn:
{
  "act_type": ["Train-Inform", "Hospital-Inform", "Restaurant-Inform"],
  "slot_type": {
    "day": "Monday",
    "departure": "Hai Duong",
    "destination": "Hanoi",
    "leaveat": "9 am",
    "department": "medical",
    "name": "Hospital 108",
    "booktime": "3 pm",
    "arriveby": "6 pm",
    "food": "French",
    "area": "Old Quarter"
  }
}
```

6 Kết luận

Việc lựa chọn phương pháp dựa trên ontology hay không phụ thuộc vào đặc điểm của sản phẩm/dịch

vụ. Phương pháp dựa trên ontology phù hợp với các hệ thống dịch vụ khép kín, ít hoặc không liên kết với các dịch vụ bên ngoài. Ngược lại, các phương pháp không dựa trên ontology, đặc biệt là các phương pháp sử dụng học sâu, thể hiện tính linh hoạt và khả năng mở rộng tốt hơn cho các hệ thống dịch vụ phức tạp, liên kết với nhiều lĩnh vực khác nhau.

Trong bối cảnh các sản phẩm và dịch vụ ngày càng phát triển và tích hợp đa dạng, xu hướng sử dụng các phương pháp không dựa trên ontology ngày càng trở nên phổ biến. Điều này cho phép hệ thống dễ dàng thích ứng với các thay đổi và mở rộng sang các lĩnh vực mới mà không cần phải xây dựng lại ontology từ đầu.

Kết quả nghiên cứu cho thấy việc xây dựng một mô hình kết hợp cả hai tác vụ phân loại đa nhãn hành động (act classification) và phân loại token (slot classification) để dự đoán giá trị slot là một hướng đi tiềm năng. Hiệu quả của mô hình này có thể được nâng cao đáng kể bởi các yếu tố sau:

- **Kích thước mô hình:** Mô hình lớn hơn với nhiều tham số hơn thường có khả năng biểu diễn và học dữ liệu tốt hơn.
- **Mối quan hệ giữa các tác vụ:** Việc tận dụng mối quan hệ giữa hai tác vụ phân loại hành động và phân loại token có thể giúp mô hình học hiệu quả hơn.
- **Chất lượng và số lượng dữ liệu:** Dữ liệu gán nhãn sạch và đa dạng là yếu tố then chốt để huấn luyện một mô hình học sâu thành công.

Sự phát triển mạnh mẽ của các Mô hình Ngôn ngữ Lớn (LLM) đã mở ra một hướng tiếp cận mới đầy hứa hẹn. Bằng cách thiết kế prompt hiệu quả và quản lý phiên (session) thông minh, chúng ta có thể tận dụng khả năng của LLM mà không cần tốn nhiều công sức vào việc thu thập, gán nhãn dữ liệu, huấn luyện và kiểm thử mô hình. Phương pháp này có tiềm năng triển khai rộng rãi cho nhiều lĩnh vực khác nhau, đáp ứng nhu cầu phát triển không ngừng của các sản phẩm và dịch vụ trong tương lai.

Tài liệu

[Campagna et al., 2020] Campagna, G., Foryciarz, A., Moradshahi, M., and Lam, M. S. (2020). Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. *arXiv preprint arXiv:2005.00891*.

[Feng et al., 2020] Feng, Y., Wang, Y., and Li, H. (2020). A sequence-to-sequence approach to dialogue state tracking. *arXiv preprint arXiv:2011.09553*.

[Gao et al., 2020] Gao, S., Agarwal, S., Chung, T., Jin, D., and Hakkani-Tur, D. (2020). From machine reading comprehension to dialogue state tracking: Bridging the gap. *arXiv preprint arXiv:2004.05827*.

[Heck et al., 2020] Heck, M., van Niekerk, C., Lubis, N., Geishauser, C., Lin, H.-C., Moresi, M., and Gašić, M. (2020). Trippy: A triple copy strategy for value independent neural dialog state tracking. *arXiv preprint arXiv:2005.02877*.

[Hu et al., 2020] Hu, J., Yang, Y., Chen, C., He, L., and Yu, Z. (2020). Sas: Dialogue state tracking via slot attention and slot information sharing. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 6366–6375.

[Hu et al., 2022] Hu, Y., Lee, C.-H., Xie, T., Yu, T., Smith, N. A., and Ostendorf, M. (2022). In-context learning for few-shot dialogue state tracking. *arXiv preprint arXiv:2203.08568*.

[Huang et al., 2020] Huang, Y., Feng, J., Hu, M., Wu, X., Du, X., and Ma, S. (2020). Meta-reinforced multi-domain state generator for dialogue systems. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7109–7118.

[Jeon and Lee, 2021] Jeon, H. and Lee, G. G. (2021). Domain state tracking for a simplified dialogue system. *arXiv preprint arXiv:2103.06648*.

[Jia et al., 2024] Jia, X., Zhang, R., and Peng, M. (2024). Multi-domain gate and interactive dual attention for multi-domain dialogue state tracking. *Knowledge-Based Systems*, 286:111383.

[Le et al., 2020a] Le, H., Sahoo, D., Liu, C., Chen, N. F., and Hoi, S. C. (2020a). End-to-end multi-domain task-oriented dialogue systems with multi-level neural belief tracker.

[Le et al., 2020b] Le, H., Socher, R., and Hoi, S. C. (2020b). Non-autoregressive dialog state tracking. *arXiv preprint arXiv:2002.08024*.

[Lim et al., 2023] Lim, J., Whang, T., Lee, D., and Lim, H. (2023). Adaptive multi-domain dialogue state tracking on spoken conversations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

[Lin et al., 2021] Lin, Z., Liu, B., Moon, S., Crook, P., Zhou, Z., Wang, Z., Yu, Z., Madotto, A., Cho, E., and Subba, R. (2021). Leveraging slot descriptions for zero-shot cross-domain dialogue state tracking. *arXiv preprint arXiv:2105.04222*.

[Ouyang et al., 2020] Ouyang, Y., Chen, M., Dai, X., Zhao, Y., Huang, S., and Chen, J. (2020). Dialogue state tracking with explicit slot connection modeling. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 34–40.

[Wang et al., 2022] Wang, Y., Zhao, J., Bao, J., Duan, C., Wu, Y., and He, X. (2022). Luna: Learning slot-turn alignment for dialogue state tracking. *arXiv preprint arXiv:2205.02550*.

[Xu et al., 2023] Xu, J., Song, D., Liu, C., Hui, S. C., Li, F., Ju, Q., He, X., and Xie, J. (2023). Dialogue state distillation network with inter-slot contrastive learning for dialogue state tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13834–13842.

[Yang et al., 2023] Yang, L., Li, J., Li, S., and Shinozaki, T. (2023). Multi-domain dialogue state tracking with disentangled domain-slot attention. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4928–4938.

[Zeng and Nie, 2020] Zeng, Y. and Nie, J.-Y. (2020). Multi-domain dialogue state tracking based on state graph. *arXiv preprint arXiv:2010.11137*.

[Zhao et al., 2021] Zhao, J., Mahdieh, M., Zhang, Y., Cao, Y., and Wu, Y. (2021). Effective sequence-to-sequence dialogue state tracking. *arXiv preprint arXiv:2108.13990*.

[Zhu et al., 2022] Zhu, Q., Li, B., Mi, F., Zhu, X., and Huang, M. (2022). Continual prompt tuning for dialog state tracking. *arXiv preprint arXiv:2203.06654*.

[Zhu et al., 2020] Zhu, S., Li, J., Chen, L., and Yu, K. (2020). Efficient context and schema fusion networks for multi-domain dialogue state tracking. *arXiv preprint arXiv:2004.03386*.