Bài tập lớn Xử lý ngôn ngữ tự nhiên (CO3085)
Học kỳ: 1, Năm học: 2024-2025

Võ Thanh Hùng

9/2024

1 Giới thiệu

Xử lý ngôn ngữ tự nhiên đóng một vai trò quan trọng trong nhiều lĩnh vực trong cuộc sống. Trong tương tác giữa người và máy tính (Human-Computer Interaction, HCI), NLP đóng vai trò ngày càng quan trọng, cùng với sự phát triển của các kỹ thuật xử lý giúp việc hiểu các câu, đoạn văn trong ngôn ngữ tự nhiên với độ chính xác tốt hơn. Trong những năm gần đây, Deep learning (DL) đạt được những kết quả ấn tượng trong nhiều lĩnh vực như xử lý ảnh, xử lý tiếng nói và đặc biệt trong xử lý văn bản (NLP) cũng đóng góp một phần lớn trong xu hướng phát triển đó. Tuy nhiên, cũng vì vậy, DL làm lu mờ phần nào vai trò của các phương pháp xử lý cổ điển của NLP. Tuy nhiên, về mặt thực tiễn, các phương pháp cổ điển, nền tảng của NLP vẫn có vai trò quan trọng và thậm chí có vai trò hỗ trợ tích cực cho các hệ thống triển khai DL cho các bài toán NLP.

Trong bài tập lớn này, sinh viên (SV) sẽ được làm quen với một bài toán NLP khá thực tế trong đó SV được yêu cầu hiện thực một số phương pháp xử lý NLP cổ điển và liên quan.

1.1 Ngữ cảnh bài toán

Customer service (CS) là một bộ phận quan trọng của tất cả các công ty bán sản phẩm. Trong bài toán tập lớn này, chúng ta xem xét ngữ cảnh CS sẽ chủ động gọi điện cho khách hàng (KH) để chào mời sản phẩm. Phần lớn các cuộc gọi sẽ kết thúc nhanh chóng khi **câu phản hồi đầu tiên** của khách hàng mang tính tiêu cực (**negative**), ví dụ: anh không có nhu cầu, tôi đang bận lắm, ... Tất nhiên, bên cạnh đó, một số khách hàng sẽ tỏ ra hào hứng và CS sẽ cần tiếp tục câu chuyện để giới thiệu sản phẩm tới KH, lúc này câu phản hồi đó có tính tích cực (**positive**).

Đóng vai trò chúng ta đang hiện thực một hệ thống tự động gọi cho KH và lấy được **câu phản hồi đầu tiên** như trên, các bạn được yêu cầu phân tích và xử lý cho câu phản hồi đó.

2 Phần I: Viết văn phạm và parse

Phần này, chúng ta chưa giới hạn dịch vụ đang giới thiệu (mở), sinh viên được yêu cầu hoàn thành các nhiệm vụ sau đây:

2.1 Viết tập rule, văn phạm cho bài toán (mở) này

- Input: None, không yêu cầu thêm input ở bên ngoài, có thể sử dụng data nội bộ (ví dụ một file Excel viết grammar theo form nào đó phù hợp để sinh ra dạng final) trong các thư mục data/ hoặc models/ (kèm với code)
- Output: output/grammar.txt

2.2 Viết giải thuật sinh câu

- Input: None, sử dụng data nội bộ (data/, models/, etc.) hoặc output/grammar.txt ở trên đều được, không yêu cầu thêm input ở ngoài
- Output: output/samples.txt

Mỗi dòng là 1 câu hợp lệ với grammar đã viết.

Hiện thực giải thuật để sinh ra tất cả các mẫu câu có thể phân tích được bởi văn phạm đã viết ở trên. (giới hạn tối đa 10,000 câu để tránh việc tập câu quá lớn.)

2.3 Xây dựng bộ phân tích cú pháp

- Input: 1 file text (sentences.txt) trong thư mục input, gồm nhiều câu, mỗi câu trên 1 dòng
- Output: output/parse-results.txt, mỗi kết quả là 1 dòng tương ứng

Hiện thực giải thuật parse cho từng câu nhận được, xuất ra cấu trúc cây văn phạm có được tương ứng. Trong trường hợp câu không hợp lệ, xuất ra 1 cây rỗng ().

3 Phần II: Biểu diễn ngữ nghĩa và trả lời

Ở phần này, sinh viên được lựa chọn **một trong hai hướng khác nhau** để làm sau đây: 3.1 hoặc 3.2.

3.1 Lựa chọn làm từng bước hệ thống hỏi đáp đơn giản

Ở phần này, chúng ta sẽ giới hạn lại những câu trả lời mang tính tích cực (positive) của khách hàng để tiếp tục câu chuyện giới thiệu. Dịch vụ được cung cấp sẽ được hạn chế lại (để dễ hiện thực) theo dữ liệu được cung cấp sau đây (database):

(TOUR PQ Phú_Quốc) (TOUR DN Đà_Nẵng) (TOUR NT Nha_Trang)

(DTIME PQ HCMC "7AM 1/7") (ATIME PQ PQ "9AM 1/7")

(DTIME PQ HCMC "8AM 5/7") (ATIME PQ PQ "10AM 5/7")

(DTIME DN HCMC "7AM 1/7") (ATIME DN DN "9AM 1/7")

(DTIME DN HCMC "7AM 4/7") (ATIME DN DN "9AM 4/7")

(DTIME NT HCMC "7AM 1/7") (ATIME NT NT "12AM 1/7")

(DTIME NT HCMC "7AM 5/7") (ATIME NT NT "12AM 5/7")

(RUN-TIME PQ HCM PQ 2:00 HR)

(RUN-TIME DN HCM PQ 2:00 HR)

(RUN-TIME NT HCM PQ 5:00 HR)

(BY PQ airplane)

(BY DN airplane)

(BY NT train)

Các bạn được yêu cầu hoàn thành các nhiệm vụ bên dưới để trả lời được những câu hỏi sau của khách hàng:

1. em có thể nhắc lại tất cả các tour được không?
2. đi từ Hồ Chí Minh tới Nha Trang hết bao lâu?
3. đi từ Hồ Chí Minh tới Đà Nẵng hết bao lâu?
4. có bao nhiêu tour đi Phú Quốc vậy bạn?
5. tour Nha Trang đi bằng phương tiện gì vậy?
6. đi Nha Trang có những ngày nào nhỉ?

Sinh viên được yêu cầu làm từng bước dưới đây:

1. Xây dựng bộ phân tích cú pháp của văn phạm phụ thuộc (* hoặc văn phạm theo hướng cấu trúc)
2. Phân tích cú pháp và xuất ra các quan hệ ngữ nghĩa của các câu truy vấn
3. Tạo các quan hệ văn phạm với cơ sở dữ liệu đã cho ở trên
4. Tạo dạng luận lý và ngữ nghĩa thủ tục
5. Truy xuất cơ sở dữ liệu để tìm thông tin và trả lời cho các câu truy vấn (hỏi) ở trên.

- Input: None, sử dụng data đã được cung cấp như là local data, không cung cấp thêm data bên ngoài
- Output: output/p2-q-$i.txt với $i là câu với thứ tự 1..5 ở trên. Ví dụ, với câu 1, output sẽ là "p1-q-1.txt"

3.2 Lựa chọn xây dựng chatbot đầy đủ

Với sinh viên lựa chọn giải pháp này, sinh viên được yêu cầu xây dựng một chatbot hoàn chỉnh (stand-alone). Topic của chatbot có thể tùy chọn, ưu tiên chatbot hỗ trợ khách hàng (customer support).

Sinh viên được tùy chọn thư viện, framework để phát triển, một số gợi ý như sau:

1. RASA
2. botPress
3. ChatterBot
4. etc.

Ngoài ra, sinh viên cũng có thể sử dụng các LLM để phát triển tiếp các ứng dụng chatbot của mình. Lưu ý rằng, trong trường hợp này, cần có những điều chỉnh phù hợp để bot có thể trả lời dựa vào tri thức theo domain, tham khảo từ khóa "**retrieval augmented generation**".

Với lựa chọn này, sinh viên được yêu cầu nộp source code cùng với Dockerfile để build docker image và deploy với docker-compose.yml (sinh viên tự đọc thêm theo hướng dẫn đầu kỳ).

Video demo/short report cần có và nộp trong thư mục bài làm.

Lưu ý: Với hướng này, sinh viên tùy chọn ngôn ngữ để hiện thực.

4 Các yêu cầu khác

4.1 Ngôn ngữ, môi trường

- Sinh viên sử dụng Java hoặc Python để làm bài tập lớn. (trừ trường hợp chọn 3.2). Chỉ sử dụng MỘT trong hai ngôn ngữ cho bài tập lớn của mình. **Không sử dụng đồng thời cả hai**.
- Bài sẽ được chấm trên Java 8 đối với ngôn ngữ và Python 3.8 đối với ngôn ngữ Python.
- Docker sẽ được sử dụng để chấm bài. File Dockerfile đã được cung cấp sẵn, và hầu như là đủ dùng, SV chỉ cần điều chỉnh khi cần thiết.
- Sinh viên được yêu cầu nộp cùng với bài làm Dockerfile (ở thư mục gốc) để build môi trường mà trong đó code SV được thực thi, gồm toàn bộ code, những thư viện, etc. đầy đủ để chạy code.

Tham khảo thư mục sample để follow theo và biết cách build môi trường chấm.

Script *util.sh* cung cấp sẵn các hàm dùng để build, chạy trích xuất kết quả, sinh viên tham khảo và sử dụng nó để test xem cách build, implement của mình đã đúng chưa.

Quy trình chấm:

- Quy ước $ROOT là thư mục gốc bài làm của bạn (sau khi giải nén ra); $S_OUT là thư mục sẽ chứa kết quả output của SV ở máy chấm bài (máy host, không phải trong container).

- Chạy các lệnh theo thứ tự:

```
cd $ROOT

docker build -t student-image .

docker run --rm -v $S\_OUT:/nlp/output student-image
```

- lấy kết quả ở thư mục $S_OUT để chấm
- Vui lòng tham khảo hàm *run_test* trong *util.sh* để biết cách đánh giá, nếu gọi hàm này thành công ra các output trong thư mục *output* là được.

Trong **student-image**:
- source code ở /nlp
- output chạy ra ở /nlp/output

4.2 Nộp bài

- Sinh viên nén toàn bộ các file/thư mục (bao gồm source code, data, ...) vào một file nén theo dạng MSSV.zip, trong đó MSSV chính là mã số sinh viên của sinh viên. Sinh viên không nén file theo các định dạng khác. Chú ý: khi giải nén file thì sẽ xuất hiện các thư mục. Tham khảo hàm *run_submit* trong *util.sh* để biết cách đóng gói, hàm đã được cung cấp sẵn, SV chỉ cần gọi lệnh để đóng gói nếu không có những yêu cầu đặc biệt.
- Trong thư mục của mỗi sinh viên sẽ có một file README.md bao gồm các thông tin về bài tập lớn cũng như các ghi chú khác về thực thi ứng dụng nếu cần.
- Đính kèm thư mục OUTPUT của sinh viên tự chạy, sử dụng trong trường hợp code không chạy được.

4.3 Giới hạn và xử lý gian lận

- Đây là bài tập cá nhân, sinh viên phải TỰ MÌNH làm bài
- Được phép sử dụng các thư viện **PUBLIC** của bên thứ ba không hạn chế, bao gồm cả trọng số của network. Tuy nhiên, sinh viên phải tự deploy, không được phép sử dụng API có sẵn.
- Có thể trao đổi ý tưởng với các sinh viên khác trong và ngoài lớp, tuy nhiên, không được share/chép lại code, kết quả.
- Mọi hình thức GIAN LẬN nếu bị phát hiện sẽ bị xử lý NGHIÊM KHẮC theo quy định học vụ.