

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Mô hình Graph2Seq cho bài toán sinh câu hỏi

NGUYỄN ĐỖ TÚ

tu.nd194200@sis.hust.edu.vn

Ngành: Khoa học máy tính

Giảng viên hướng dẫn: PGS.TS. NGUYỄN THỊ KIM ANH

Chữ ký GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 08/2023

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn đến toàn thể các thầy cô trong trường Đại học Bách khoa Hà Nội nói chung và các thầy cô trong trường Công nghệ thông tin và Truyền thông nói riêng đã tận tình chỉ bảo và truyền đạt cho em rất nhiều kiến thức mới, kinh nghiệm quý báu trong suốt quá trình 4 năm học vừa qua để giúp chúng em bước tiếp trên con đường mới đầy khó khăn.

Đặc biệt, em xin chân thành cảm ơn cô PGS.TS. Nguyễn Thị Kim Anh đã giúp đỡ em trong suốt thời gian làm đồ án, cô đã chia sẻ những kiến thức, ý tưởng thực tế để giúp em hoàn thiện đồ án của mình một cách tốt nhất có thể.

Bên cạnh đó thì em xin gửi lời cảm ơn đến gia đình đã luôn ở bên cạnh hỗ trợ, động viên và tạo điều kiện tốt nhất để em hoàn thành đồ án. Đặc biệt em xin gửi lời cảm ơn sâu sắc nhất tới mẹ em, mẹ đã tiếp thêm động lực, ý chí để giúp em vượt qua những khoảng thời gian khó khăn và gian nan nhất. Cuối cùng, em xin cảm ơn những người bạn luôn sẵn sàng chia sẻ kinh nghiệm và giúp đỡ em hoàn thành tốt quá trình học tập và rèn luyện ở đại học.

Thời gian làm đồ án không phải là dài, nên trong quá trình làm đồ án khó có thể tránh khỏi những hạn chế và thiếu sót, nên em mong nhận được những ý kiến đóng góp của thầy giáo, cô giáo và các bạn đọc để đồ án của em trở lên hoàn thiện hơn.

Em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Bài toán sinh ra câu hỏi là một vấn đề đầy thách thức trong Xử lý ngôn ngữ tự nhiên (NLP), nhằm tạo ra các câu hỏi tự nhiên và có ý nghĩa từ các định dạng đầu vào khác nhau, chẳng hạn như văn bản ngôn ngữ tự nhiên, cơ sở dữ liệu có cấu trúc, cơ sở kiến thức và hình ảnh. Trong đồ án này, em sẽ tìm hiểu bài toán sinh câu hỏi đối với dữ liệu văn bản. Đây là một lĩnh vực nghiên cứu rất tiềm năng bởi những ứng dụng đối với các lĩnh vực khác như làm giàu dữ liệu cho hệ thống hỏi đáp tự động, sinh ra các bài kiểm tra ứng dụng trong giáo dục, xác thực an toàn thông tin,... Những nghiên cứu gần đây sử dụng mạng nơ ron nhân tạo cũng đã đạt được những kết quả nhất định, đặc biệt là với mô hình học sâu dựa trên đồ thị, có thể kể đến như mô hình Graph2Seq kết hợp với học tăng cường (2020), mô hình Graph2Seq sử dụng bộ giải mã dựa trên mạng đồ thị lặp (2021). Những nghiên cứu về mạng nơ ron nhân tạo với bài toán sinh câu hỏi hầu hết sử dụng bộ dữ liệu SQuAD đã tiền xử lý, trong đó đoạn văn được xử lý phần lớn chỉ còn một câu (câu văn chứa câu trả lời). Tuy nhiên trong thực tế, những đoạn văn thường rất dài, thông tin bổ sung cho câu hỏi và câu trả lời không chỉ gói gọn trong một, hai câu mà nó có thể còn nằm ở những câu khác. Vì vậy, em sẽ sử dụng bộ dữ liệu SQuAD gốc để thực hiện tiền xử lý lại và huấn luyện mô hình. Ngoài ra, tại bước xây dựng đồ thị cho đoạn văn, phương pháp hiện tại được sử dụng là dựng cây phân tích cú pháp phụ thuộc cho từng câu, sau đó thực hiện nối các cây ở các nút tương ứng với biên của câu. Điều đó sẽ dẫn đến trong khi lan truyền trong mạng đồ thị, nhưng của nút có thể sẽ thiếu thông tin về ngữ cảnh hay cấu trúc cú pháp của câu. Trong đồ án này, em sẽ thực hiện thử nghiệm trên bộ dữ liệu SQuAD gốc phiên bản 1.1 các phương pháp khác xây dựng đồ thị cho đoạn văn: (i) xây dựng đồ thị dựa trên HEAD của câu và (ii) xây dựng đồ thị sử dụng phân giải đồng tham chiếu. Phương pháp sử dụng biên của câu được chọn là phương pháp baseline để so sánh, và kết quả cho thấy phương pháp dựa trên HEAD của câu và phương pháp sử dụng phân giải đồng tham chiếu đạt được kết quả tốt hơn so với baseline trên tập SQuAD phiên bản 1.1.

Sinh viên thực hiện
(Ký và ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Các giải pháp hiện tại và hạn chế	2
1.3 Mục tiêu và định hướng giải pháp	3
1.4 Đóng góp của đồ án	3
1.5 Bố cục đồ án	4
CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT	5
2.1 Phát biểu của bài toán	5
2.2 Các nghiên cứu liên quan	5
2.3 Mô hình Graph2Seq với IGND	6
2.3.1 Bộ mã hóa quan hệ (Relational Encoder).....	6
2.3.2 Bộ giải mã quan hệ (Iterative Graph Network-based Decoder).....	10
2.3.3 Hàm mất mát	13
2.4 Kết chương.....	13
CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT.....	15
3.1 Tổng quan giải pháp.....	15
3.2 Bộ dữ liệu SQuAD phiên bản 1.1	15
3.3 Phương pháp xây dựng đồ thị từ đoạn văn	29
3.3.1 Phương pháp xây dựng đồ thị dựa trên biên của câu.....	29
3.3.2 Phương pháp xây dựng đồ thị phụ thuộc dựa trên HEAD của câu....	31
3.3.3 Phương pháp nối các cây cú pháp phụ thuộc sử dụng phân giải đồng tham chiếu.....	33
3.4 Kết chương.....	35

CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM.....	36
4.1 Các tham số đánh giá	36
4.1.1 BLEU	36
4.1.2 ROUGE-L.....	37
4.2 Phương pháp thí nghiệm.....	38
4.2.1 Dữ liệu.....	38
4.2.2 Phương pháp baseline.....	38
4.2.3 Tham số huấn luyện	38
4.3 Kết quả thực nghiệm	39
4.4 Kết chương.....	45
CHƯƠNG 5. KẾT LUẬN	46
5.1 Kết luận	46
5.2 Hướng phát triển trong tương lai	46
TÀI LIỆU THAM KHẢO.....	51

DANH MỤC HÌNH VẼ

Hình 1.1 Mô hình dựa trên đồ thị ([7] Zhang et al., 2021)	2
Hình 2.1 Kiến trúc tổng quát của mô hình Graph2Seq với bộ giải mã IGND. Chú thích trong ảnh: những nút màu vàng là từ ứng với câu trả lời, màu xanh hay tím chỉ những copied word và mức độ màu nhạt đậm tượng trưng cho điểm cao thấp của copy score. ([8] Fei et al., 2021)	6
Hình 2.2 Mô tả luồng thực thi trong bộ giải mã Decoder trong bước thời gian thứ t	11
Hình 3.1 Phân bố của thể loại câu trả lời trong bộ dữ liệu SQuAD 1.1. .	16
Hình 3.2 Phân bố về số lượng câu của đoạn văn trong bộ dữ liệu SQuAD 1.1.	17
Hình 3.3 Phân bố về số lượng câu của đoạn văn trong bộ dữ liệu SQuAD 1.1.	18
Hình 3.4 Biểu đồ về phân bố về số lượng token trong đoạn văn trong bộ dữ liệu SQuAD 1.1.	19
Hình 3.5 Biểu đồ boxplot về phân bố về số lượng token trong đoạn văn trong bộ dữ liệu SQuAD 1.1.	19
Hình 3.6 Biểu đồ về phân bố về số lượng token của câu hỏi trong bộ dữ liệu SQuAD 1.1.	20
Hình 3.7 Biểu đồ boxplot về phân bố về số lượng token của câu hỏi trong bộ dữ liệu SQuAD 1.1.	20
Hình 3.8 Biểu đồ về phân bố về số lượng token của câu trả lời trong bộ dữ liệu SQuAD 1.1.	21
Hình 3.9 Biểu đồ boxplot về phân bố về số lượng token của câu trả lời trong bộ dữ liệu SQuAD 1.1.	21
Hình 3.10 Các từ vựng xuất hiện trong đoạn văn trong bộ dữ liệu SQuAD 1.1.	23
Hình 3.11 Các từ vựng xuất hiện trong câu hỏi trong bộ dữ liệu SQuAD 1.1.	24
Hình 3.12 Các từ vựng xuất hiện trong câu trả lời trong bộ dữ liệu SQuAD 1.1.	25
Hình 3.13 Biểu đồ thống kê phần trăm số lượng câu hỏi theo mục đích hỏi trong bộ dữ liệu SQuAD 1.1.	27

Hình 3.15 Biểu đồ thống kê số lượng câu hỏi tương ứng với từ để hỏi trong bộ dữ liệu SQuAD 1.1.	27
Hình 3.14 Biểu đồ thống kê phần trăm số lượng câu hỏi tương ứng với từ để hỏi trong bộ dữ liệu SQuAD 1.1.	28
Hình 3.16 Ví dụ về kết nối các cây phân tích cú pháp qua biên của câu. Đoạn văn "The light of the lava lamp deepens my experience of my environment . It transforms the room into a peaceful refuge." gồm hai câu, trong đó biên cuối của câu thứ nhất là "environment", biên đầu của câu thứ hai là "It".	31
Hình 3.17 Ví dụ về kết nối các cây phân tích cú pháp qua HEAD của câu. Đoạn văn "The light of the lava lamp deepens my experience of my environment. It transforms the room into a peaceful refuge." gồm hai câu, trong đó HEAD của mỗi câu được chọn là các đỉnh không có cung phụ thuộc nào trỏ đến, ở đây là hai nút "deepens" và "transforms".	32
Hình 3.18 Ví dụ về kết nối các cây phân tích cú pháp sử dụng phân giải đồng tham chiếu và HEAD của câu.	34
Hình 4.1 Biểu đồ giá trị của hàm mất mát qua từng epoch.	40
Hình 4.2 Biểu đồ giá trị độ đo BLEU-1, BLEU-2, BLEU-3, BLEU-4 qua từng epoch.	41
Hình 4.3 Biểu đồ giá trị độ đo ROUGE-L F1 qua từng epoch.	42
Hình 4.4 Trong ví dụ đầu tiên, từ "its" tham chiếu đến từ "France", từ đó câu hỏi liên quan đến "France" sẽ có được thông tin quanh từ "its" là "worldwide empire", "rebuild".	45

DANH MỤC BẢNG BIỂU

Bảng 3.1 Sô lượng các nhãn NER xuất hiện trong các tập train, dev và test.	22
Bảng 4.1 Kết quả độ đo BLEU và ROUGE-L thực hiện trên tập test.	42
Bảng 4.2 Kết quả "Human Evaluation" thực hiện trên 100 mẫu ngẫu nhiên.	42
Bảng 4.3 Một số ví dụ về kết quả câu hỏi sinh ra của mô hình Graph2Seq với ba phương pháp xây dựng đồ thị từ văn bản.	44

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

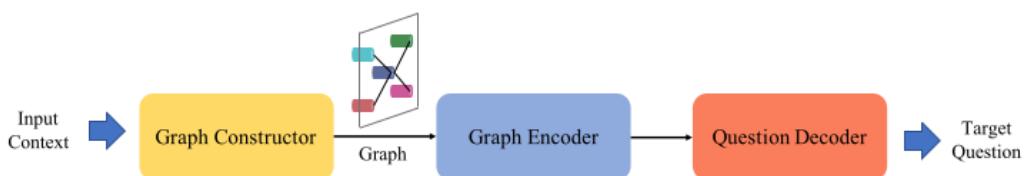
Thuật ngữ	Ý nghĩa
BERT	Bidirectional Encoder Representation from Transformer
bi-GGNN	Mạng Nơ ron Đồ thị Có Cổng hai chiều (bidirectional GGNN)
bi-LSTM	bidirectional LSTM)
GGNN	Mạng Nơ ron Đồ thị Có Cổng (Gated Graph Neural Network)
GNN	Mạng nơ ron đồ thị (Graph Neural Network)
IGND	Bộ giải mã dựa trên mạng đồ thị lặp (Iterative Graph Network-based Decoder)
LSTM	Mô hình bộ nhớ Ngắn hạn Dài (Long Short-Term Memory)
NER	Name Entity Recognition
POS	Part of Speech
QA	Hệ thống hỏi đáp (Question Answering)
QG	Hệ thống sinh câu hỏi (Question Generation)
RNN	Mạng nơ ron hồi quy (Recurrent Neural Network)
SOTA	State of the art

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trước những khối kiến thức khổng lồ của nhân loại, con người luôn luôn đặt ra những câu hỏi về mọi sự vật sự việc xung quanh chúng, và việc nghiên cứu về hệ thống đặt câu hỏi (Question Generation - QG) là một lĩnh vực nghiên cứu quan trọng trong Trí tuệ Nhân tạo. Gần đây, những nghiên cứu về bài toán sinh câu hỏi được mở rộng với đầu vào không chỉ là dữ liệu văn bản ngôn ngữ tự nhiên mà còn bao gồm cơ sở trí thức ([1] ElSahar et al., 2018; [2] Khapra et al., 2017; [3] Kumar et al., 2019) và hình ảnh ([4] Fan et al., 2018a; [5] Fan et al., 2018b; [6] Li et al., 2018). Trong đó, bài toán sinh câu hỏi đối với dữ liệu văn bản ngôn ngữ tự nhiên đang rất được quan tâm bởi những ứng dụng rộng rãi của nó, ví dụ như QG có thể được tận dụng như một chiến lược tăng cường dữ liệu, giảm công sức của con người trong việc tạo ra các cặp câu hỏi và câu trả lời cho các hệ thống hỏi đáp quy mô lớn, hay QG cũng có thể coi như một mô-đun trong các hệ thống hỏi đáp, trợ lý ảo để giúp cuộc hội thoại trở nên tự nhiên và có sự tương tác tốt hơn, hay như ứng dụng dễ thấy nhất của QG là trong giáo dục, hệ thống QG có thể giúp các thầy cô sinh ra những bài kiểm tra theo yêu cầu nội dung, giảm bớt sức lực của giáo viên. Trong thập kỷ vừa qua, có hai phương pháp tiếp cận chủ yếu cho bài toán sinh câu hỏi là dựa trên luật (rule based) và sử dụng mạng nơ ron nhân tạo (neural network based), trong đó những nghiên cứu sử dụng mạng nơ ron nhân tạo đã đạt được những kết quả tốt nhất.

Các mô hình mạng nơ ron tiếp cận bài toán QG với dữ liệu văn bản thường có thiết kế gồm một bộ mã hóa (Encoder) và một bộ giải mã (Decoder). Để câu hỏi sinh ra khi trả lời phải liên quan đến đoạn văn cho trước, ta phải tìm cách biểu diễn được sự liên kết các câu văn trong đoạn văn và mối quan hệ giữa đoạn văn và câu trả lời cho trước. Đối với mô hình sequence-to-sequence (Seq2Seq) hay được sử dụng trong các tác vụ có đầu vào văn bản thì lại không đáp ứng được yêu cầu liên quan đến sự liên kết giữa các câu văn bởi điểm yếu của Seq2Seq khi đoạn văn gồm nhiều câu dài là vấn đề phụ thuộc xa. Kể cả khi sử dụng mạng LSTM nhưng vẫn không thể giải quyết được triệt để vấn đề phụ thuộc xa đối với đoạn văn dài mà trong thực tế, đoạn văn đầu vào của hệ thống thường phức tạp hơn với nhiều câu hơn. Vì vậy, một số nghiên cứu đã sử dụng mô hình dựa trên đồ thị với ưu điểm mô hình hóa tốt những dữ liệu có tính cấu trúc cao và có thể mã hóa được mối quan hệ xa giữa các câu.



Hình 1.1: Mô hình dựa trên đồ thị ([7] Zhang et al., 2021)

Hình 1.1 mô tả cấu trúc chung của mô hình dựa trên đồ thị, gồm ba bước chính (i) xây dựng đồ thị từ đầu vào có thể là dữ liệu văn bản (phương pháp phân giải đồng tham chiếu, phương pháp dùng cây phân tích cú pháp, ...) hay dữ liệu hình ảnh (phương pháp sử dụng phát hiện đối tượng, phương pháp mô tả hình ảnh, ...), (ii) Mã hóa đồ thị để học ra biểu diễn nhúng của nút, và (iii) giải mã để đưa ra câu hỏi. Một nghiên cứu được công bố năm 2021 của tác giả ([8] Fei et al., 2021) đã cải tiến mô hình dựa trên đồ thị graph-to-sequence (Graph2Seq) ([9] Xu et al., 2018) với bộ giải mã dựa trên mạng đồ thị lặp (Iterative Graph Network-based Decoder - IGND) và kết quả thực nghiệm trên bộ dữ liệu SQuAD và MS MARCO đã đạt được kết quả SOTA. Tuy nhiên, bộ dữ liệu SQuAD mà nghiên cứu trên sử dụng lại có phần đơn giản khi số câu trong đoạn văn đã được rút gọn xuống và phần xây dựng đồ thị từ đoạn văn sử dụng phương án dùng cây phân tích cú pháp phụ thuộc đối với từng câu và sau đó ghép các câu lại với nhau thông qua từ ở biên mỗi câu. Điều này khi thực hiện lan truyền để học nhúng của nút, các câu có thể sẽ không có đầy đủ thông tin ngữ cảnh ở các câu khác.

1.2 Các giải pháp hiện tại và hạn chế

Đối với những mô hình dựa trên đồ thị, mạng nơ ron đồ thị (GNN) đã thể hiện được ưu điểm so với mạng nơ ron hồi quy (RNN) trong việc biểu diễn cấu trúc thông tin ẩn của văn bản như những phương pháp phân tích cú pháp (syntactic parsing) ([10] Xu et al., 2018b), phân tích ngữ nghĩa (semantic parsing) ([11] Song et al., 2018b). Hơn nữa, mạng GNN còn có thể mô hình hóa mối quan hệ toàn cục giữa các từ để cải thiện biểu diễn với khả năng truyền thông tin qua các kết nối giữa các từ hoặc cụm từ trong câu và kết hợp thông tin biểu diễn của các hàng xóm. Điều này cho phép thông tin ở một nút có thể được tổng hợp từ những nút ở xa nó, giúp cải thiện vấn đề phụ thuộc xa trong các mạng RNN.

Trong bài toán sinh câu hỏi, với đầu vào của mô hình gồm một đoạn văn bản, ta cần thực hiện bước xây dựng đồ thị từ dữ liệu văn bản và cũng đã có những giải pháp được đưa ra. Một trong những giải pháp sớm nhất được đề xuất là phân tích cú pháp sử dụng thông tin về vị trí của từ, cây phân tích cú pháp và cấu trúc ngữ pháp trong câu (Constituency Parsing) để tạo ra đồ thị cú pháp ([10] Xu et al., 2018b) và

đã đạt những kết quả nhất định trong tác vụ tạo ra dữ liệu cấu trúc logic từ dữ liệu văn bản. Ngoài ra còn có hai phương pháp khác là xây dựng đồ thị kiểu tĩnh dựa trên cú pháp và kiểu động hướng ngữ nghĩa (static and dynamic graph) ([12] Chen et al., 2020), cụ thể phương pháp tĩnh áp dụng cây phân tích cú pháp cho từng câu và nối các cây lại với nhau thông qua từ biên của câu; phương pháp động xây dựng đồ thị có hướng và có trọng số dựa trên ma trận chú ý cho nhúng ở mức từ của đoạn văn, lợi thế của phương pháp là có thể học được cấu trúc thông tin ẩn của đoạn văn mà không cần miền tri thức cụ thể. Kết quả thực nghiệm hai phương pháp trên với tập dữ liệu SQuAD đã qua xử lý squad-split ([13] Zhou et al., 2017) cho thấy phương pháp tĩnh có kết quả tốt hơn phương pháp động trong tất cả các câu hình thực nghiệm ([12] Chen et al., 2020).

Phương pháp xây dựng đồ thị tĩnh ở trên đối với đầu vào là đoạn văn gồm nhiều câu thì ta sẽ dựng đồ thị tương ứng bằng cách nối các cây phân tích cú pháp của mỗi câu tại nút tương ứng với biên của câu (từ bắt đầu và từ kết thúc của câu). Điều này khi thực hiện lan truyền trong mạng GNN, khi tổng hợp thông tin ở các nút biên có thể sẽ không có được thông tin liên quan xuất hiện ở những câu khác.

1.3 Mục tiêu và định hướng giải pháp

Mục tiêu chính của đồ án là thử nghiệm các giải pháp xây dựng đồ thị từ đoạn văn và kết hợp huấn luyện với mô hình Graph2Seq với bộ giải mã IGND ([8] Fei et al., 2021) trên bộ dữ liệu SQuAD. Phương pháp xây dựng đồ thị cho đoạn văn vẫn dựa trên cây phân tích cú pháp cho mỗi câu trong đoạn, nhưng sẽ có các cách khác nhau để nối ghép các cây đó lại. Trong đồ án này ngoài phương pháp nối dựa trên biên của câu, em sẽ trình bày thêm hai phương án nối khác là phương pháp nối dựa trên HEAD của câu và phương pháp nối sử dụng phân giải đồng tham chiếu.

Vì đồ án tập trung chính vào phần tiền xử lý xây dựng đồ thị cho đoạn văn gồm nhiều câu nên bộ dữ liệu sử dụng trong đồ án là bộ dữ liệu gốc SQuAD phiên bản 1.1 ([14] Rajpurkar et al., 2016) thay vì sử dụng một phiên bản "squad-split" ([15] Du et al., 2017) đã được tiền xử lý sẵn, thu gọn số lượng câu của đoạn văn xuống chỉ còn đa phần là một câu (xử lý bằng cách chỉ lấy câu chứa câu trả lời hoặc nếu câu trả lời nằm ở nhiều hơn một câu thì sẽ lấy những câu đó ghép lại).

1.4 Đóng góp của đồ án

Đồ án này có 2 đóng góp chính như sau:

1. Đồ án thử nghiệm phương án mới xây dựng đồ thị từ văn bản cho trước để cải thiện tính cấu trúc trong biểu diễn đồ thị.
2. Đồ án thực hiện thực nghiệm trên bộ dữ liệu SQuAD phiên bản 1.1 chưa qua

xử lý theo định dạng "squad-split" để đánh giá ảnh hưởng của phương pháp xây dựng đồ thị.

1.5 Bố cục đồ án

Chương 2 trình bày về kiến trúc mô hình học sâu Graph2Seq sử dụng trong việc huấn luyện gồm hai bộ phận chính: (i) bộ mã hóa với tầng làm giàu thông tin đầu vào và mã hóa đồ thị quan hệ; (ii) bộ giải mã IGND sử dụng kết hợp mô hình LSTM và mô hình đồ thị bi-GGNN để mô hình hóa cấu trúc thông tin ẩn và cơ chế sao chép (copy mechanism).

Chương 3 trình bày về phần phân tích bộ dữ liệu SQuAD phiên bản 1.1 để nhìn ra những đặc trưng của bộ dữ liệu hỏi đáp, như thống kê về kích thước bộ dữ liệu, sự đa dạng về kiểu câu hỏi, ... các phương pháp xây dựng đồ thị từ đoạn văn bản cho trước. Có ba phương pháp được sử dụng trong đồ án: (i) phương pháp xây dựng đồ thị dựa trên biên của câu (phương án baseline); (ii) phương pháp xây dựng đồ thị dựa trên HEAD của câu và (iii) phương pháp xây dựng đồ thị sử dụng phân giải đồng tham chiếu.

Chương 4 trình bày về kết quả thực nghiệm khi chạy mô hình IGND với ba phương pháp kể trên với dữ liệu SQuAD phiên bản 1.1.

CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT

2.1 Phát biểu của bài toán

Bài toán sinh câu hỏi dựa trên hướng tiếp cận sử dụng mô hình học sâu sẽ thực hiện sinh từ một đoạn văn cho trước và yêu cầu câu hỏi sinh ra phải trả lời được dựa trên đoạn văn đã cho. Giả sử đoạn văn đã cho là một chuỗi các từ, kí hiệu X^P và $X^P = \{x_1^P, x_2^P, \dots, x_N^P\}$; câu trả lời cho trước là chuỗi các từ, kí hiệu X^A và $X^A = \{x_1^A, x_2^A, \dots, x_L^A\}$. Kết quả đầu ra mong muốn của mô hình là một câu hỏi Y cũng là chuỗi các từ $Y = \{y_1, y_2, \dots, y_T\}$.

Chú thích: N, L, T lần lượt là độ dài của đoạn văn, câu trả lời và câu hỏi sinh ra tính theo đơn vị từ. Mục tiêu của bài toán là cực đại hóa hàm *likelihood*:

$$\operatorname{argmax}_Y P(Y|X^P, X^A)$$

2.2 Các nghiên cứu liên quan

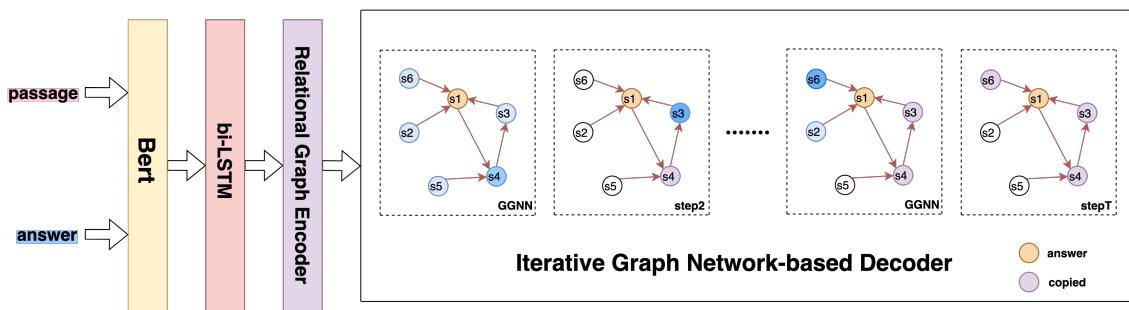
Những nghiên cứu sớm nhất về bài toán sinh câu hỏi thường tiếp cận theo hướng sử dụng luật (rule-based), dựa trên heuristic hay những bản mẫu được tạo thủ công ([16] Mostow and Chen, 2009; [17] Heilman and Smith, 2010) nhưng lại có khả năng mở rộng và khả năng sinh từ thấp, không hiệu quả. Một hướng tiếp cận khác để giải quyết bài toán này là sử dụng học sâu dựa trên những bước phát triển lớn ở công nghệ phần cứng. Những nghiên cứu dựa trên hướng sử dụng học sâu có thể kể đến như mô hình mạng nơ ron Seq2Seq kết hợp với cơ chế chú ý ([15] Du et al., 2017); nghiên cứu cải thiện đầu vào của bộ mã hóa và làm giàu thông tin (feature-enriched) bằng cách nối thêm vào đầu vào các thông tin về từ vựng trong câu ([13] Zhou et al., 2018); những nghiên cứu của các tác giả ([18] Sun et al., 2018; [19] Kim et al., 2019; [20] Song et al., 2018) đã sử dụng thêm thông tin về vị trí câu trả lời trong đoạn văn, từ đó câu hỏi sinh ra sẽ có ý nghĩa sát với câu trả lời hơn; phương pháp áp dụng đặc trưng về mặt cú pháp để biểu diễn từ và quyết định xem từ nào là quan trọng trong quá trình sinh từ ([21] Liu et al., 2019; [12] Chen et al., 2020); nghiên cứu của ([12] Chen et al., 2020) kết hợp học có giám sát và học tăng cường trong quá trình huấn luyện để cực đại hóa độ đo chất lượng của câu hỏi sinh ra. Và kết quả gần đây của nhóm tác giả ([22] Chan and Fan, 2020) áp dụng mô hình BERT với chiến lược sử dụng mô hình huấn luyện trước (pretrain) đã đạt đến hiệu năng SOTA (state-of-the-art).

Một nghiên cứu năm 2021 của tác giả Zichu Fei và các cộng sự về bài toán sinh câu hỏi công bố ở hội nghị “Conference on Empirical Methods in Natural Language Processing (2021)” có tên “Iterative GNN-based Decoder for Question

Generation” ([8]) đã đạt được những kết quả SOTA trên hai tập dữ liệu SQuAD và MARCO. Trong nghiên cứu trên, tác giả đã cải tiến bộ mã hóa với nhúng quan hệ để lấy thông tin kết nối giữa câu trả lời (Relational Encoder) và đoạn văn và phát triển bộ giải mã LSTM kết hợp với mạng đồ thị bi-GGNN giúp giải quyết các vấn đề về thông tin cấu trúc trong lúc sinh từ và ảnh hưởng của từ sao chép (copied word).

2.3 Mô hình Graph2Seq với IGND

Mô hình Graph2Seq với IGND gồm hai thành phần chính là bộ mã hóa quan hệ và bộ giải mã dựa trên mạng đồ thị lặp IGND. Dữ liệu đầu vào gồm đoạn văn bản và câu trả lời cho trước, sau khi qua pha tính toán biểu diễn và xây dựng đồ thị từ dữ liệu văn bản, bộ giải mã học ra vector nhúng ở mức đồ thị của văn bản đầu vào và đưa giá trị đó qua bộ giải mã để thực hiện sinh câu hỏi phù hợp (Hình 2.1).



Hình 2.1: Kiến trúc tổng quát của mô hình Graph2Seq với bộ giải mã IGND. Chú thích trong ảnh: những nút màu vàng là ứng với câu trả lời, màu xanh hay tím chỉ những copied word và mức độ nhạt đậm tương ứng cho điểm cao thấp của copy score. ([8] Fei et al., 2021)

2.3.1 Bộ mã hóa quan hệ (Relational Encoder)

a, Mã hóa thông tin ngữ cảnh của đoạn văn và câu trả lời

Đầu tiên, đầu vào của mô hình gồm đoạn văn X^P và câu trả lời X^A sẽ đi qua tầng làm giàu thông tin sử dụng mô hình Bộ nhớ Ngắn hạn Dài hai chiều (bi-LSTM), sau đó sẽ qua bộ mã hóa đồ thị quan hệ (Relational Graph Encoder) thực hiện tổng hợp phụ thuộc giữa các từ trong câu trả lời và đoạn văn. Đối với đoạn văn X^P , ở tầng đầu tiên bi-LSTM, đầu vào được làm giàu thông tin thêm bằng cách thêm các thông tin về từ vựng như từ loại (part-of-speech - POS), nhận diện thực thể có tên (Name Entity Recognition - NER), thông tin về vị trí của câu trả lời, chữ viết hoa viết thường, và BERT để lấy thông tin ngữ cảnh của câu. Cụ thể, kí hiệu đầu vào tầng bi-LSTM là e_i , ta có:

$$e_i = [w_i, b_i, a_i, p_i, n_i, u_i] \quad (2.1)$$

CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT

trong đó, w_i là nhúng của từ thứ i sử dụng GloVe vector 300 chiều. b_i là nhúng BERT của từ, a_i là nhúng vị trí của câu trả lời xuất hiện trong đoạn văn, n_i là nhúng của NER, p_i là nhúng của POS, u_i là nhúng "word case".

Vị trí của câu trả lời xuất hiện trong đoạn văn đóng vai trò quan trọng trong việc sinh câu hỏi có ý nghĩa, và ở đây, a_i sẽ được tính bằng cách sử dụng cách đánh nhãn BIO. Trong đó, B là đánh dấu vị trí bắt đầu của câu trả lời trong đoạn văn, I để đánh dấu rằng từ đó vẫn nằm trong câu trả lời, O là đánh dấu từ không nằm trong câu trả lời. Còn đối với u_i là nhúng word case, thực hiện đánh nhãn nhị phân với hai trường hợp từ là chữ thường hay có chữ cái viết hoa.

Thực hiện lan truyền tiến và lan truyền ngược trên mạng bi-LSTM, ta sẽ có biểu diễn vector ẩn theo hai chiều \vec{h}_i và \overleftarrow{h}_i .

$$\vec{h}_i = LSTM(e_i, \overrightarrow{h}_{i-1}) \quad (2.2)$$

$$\overleftarrow{h}_i = LSTM(e_i, \overleftarrow{h}_{i-1}) \quad (2.3)$$

trong đó, i là từ thứ i trong đoạn văn X^P . Tổng hợp kết quả hai chiều lan truyền tiến và lan truyền ngược ta có vector ẩn cho từ thứ i trong X^P là:

$$h_i = [\vec{h}_i, \overleftarrow{h}_i] \quad (2.4)$$

Thực hiện nối các vector ẩn của các từ trong đoạn văn X^P ta có đầu ra qua tầng mạng bi-LSTM của X^P là:

$$H = [h_1, h_2, \dots, h_N] \quad (2.5)$$

Đối với câu trả lời X^A , áp dụng mạng bi-LSTM và tính toán tương tự từ công thức (2.2) đến (2.5) với đầu vào khi đưa vào mạng sẽ gồm nhúng từ kết hợp với nhúng BERT và ta được đầu ra tương ứng là H^a .

Trong đoạn văn X^P không phải từ nào cũng có vai trò quan trọng trong biểu diễn vector ẩn, vì vậy sử dụng thêm cơ chế chú ý để lấy được ảnh hưởng của các từ quan trọng với câu trả lời. Đầu tiên sử dụng hàm $tanh$ để lấy điểm e_i^a giữa H^a và vector ẩn h_i của từ thứ i .

$$e_i^a = v_a^T \tanh(W_a H^a + W_h h_i) \quad (2.6)$$

$$\alpha_i^a = \frac{\exp(e_i^a)}{\sum_{j=1}^N e_j^a} \quad (2.7)$$

$$H^P = \sum_{i=1}^N \alpha_i^a h_i \quad (2.8)$$

chú thích ở công thức 2.7, j là chỉ số chỉ từ thứ j trong đoạn văn. Sau đó ta sẽ chuẩn hóa điểm số của cơ chế chú ý α_i^a cho từ thứ i thông qua hàm softmax và thực hiện nhân vector ẩn của từ thứ i với trọng số chú ý tương ứng ta có vector trạng thái ẩn có thông tin ngữ cảnh hướng câu trả lời (answer-aware weighted context hidden states) H^P .

b, Mã hóa mối quan hệ giữa các từ với mạng thần kinh đồ thị

Trong khi mạng hồi quy tuyến tính RNNs xử lý tốt các tác vụ bắt phụ thuộc cục bộ giữa các từ liên tục trong văn bản, mạng thần kinh đồ thị Graph Neural Network (GNNs) lại thể hiện tốt hơn ở việc khai thác tốt các cấu trúc ẩn giàu thông tin phân tích cú pháp hoặc phân tích cú pháp ngữ nghĩa. Hơn nữa, mạng thần kinh đồ thị GNNs còn có thể mô hình hóa mối quan hệ toàn cục giữa các từ để cải thiện biểu diễn vector của đoạn văn. Vì vậy, thay vì sử dụng mạng hồi quy tuyến tính RNNs để mã hóa đoạn văn bản, ta sẽ mã hóa đoạn văn thành một đồ thị với mỗi đỉnh là tương ứng với từ trong đoạn văn đó, sau đó sử dụng mô hình Graph2Seq để mã hóa đồ thị tương ứng với đoạn văn và câu trả lời để giải mã ra câu hỏi.

Đầu tiên, ta cần dựng đồ thị từ dữ liệu văn bản. Phương pháp được sử dụng là xây dựng cây phân tích cú pháp phụ thuộc (dependency parsing) để xây dựng đồ thị có hướng với mỗi câu trong đoạn văn, sau đó ta sẽ kết nối những đồ thị cú pháp phụ thuộc đó với nhau.

Để có thể học được nhúng của nút đồ thị (node embedding), ta sẽ áp dụng mạng thần kinh đồ thị có cỗng hai chiều (Bidirectional Gated Graph Neural Network - bi-GGNN) để học ra nhúng của nút đồ thị từ cạnh với cả chiều vào và chiều ra. Ý tưởng tương tự đã được áp dụng trong mô hình GraphSAGE ([23] Hamilton et al., 2017) (một cải tiến của mô hình mạng thần kinh đồ thị) nhưng thay vì học ra nhúng của nút theo hướng cạnh đi vào và hướng cạnh đi ra một cách độc lập và thực hiện tổng hợp chúng vào cuối vòng lặp thì ta sẽ tính toán đồng thời nhúng của nút theo cả hai hướng và tổng hợp chúng trong mỗi bước lặp. Ngoài ra, ta sẽ sử dụng thêm một yếu tố nữa là nhúng quan hệ (relational embedding) cho mỗi đỉnh để tổng hợp mối quan hệ giữa các từ trong đoạn văn.

Cụ thể, trong mạng bi-GGNN, nhúng của nút sẽ được khởi tạo bằng vector trạng

thái ẩn có thông tin ngữ cảnh hướng câu trả lời H^P mà ta tính toán được ở mô hình bi-LSTM (công thức 2.8), nhúng quan hệ sẽ được khởi tạo ngẫu nhiên. Trong mỗi bước tính toán (graph hop), với mỗi nút trong đồ thị, ta sẽ áp dụng hàm kết tập (aggregation function) để tổng hợp thông tin của các nút hàng xóm có cạnh nối theo chiều đi vào (hoặc đi ra) và của chính nút đó. Trong đồ án này, em sử dụng hàm kết tập MEANS để tính toán.

$$h_{\mathcal{N}_v}^{k+1} = \frac{h_{\mathcal{N}_v}^k + \sum_{j \in \mathcal{N}_v} h_{v_j}^k}{|\mathcal{N}_v|} \quad (2.9)$$

$$h_{\mathcal{N}_v}^{k+1} = \frac{h_{\mathcal{N}_v}^k + \sum_{j \in \mathcal{N}_v} h_{v_j}^k}{|\mathcal{N}_v|} \quad (2.10)$$

$$h_{\mathcal{N}_{rela}}^{k+1} = \frac{h_{\mathcal{N}_{rela}}^k + \sum_{j \in \mathcal{N}_{rela}} r_{ij}}{|\mathcal{N}_{rela}|} \quad (2.11)$$

$$h_{\mathcal{N}_{rela}}^{k+1} = \frac{h_{\mathcal{N}_{rela}}^k + \sum_{j \in \mathcal{N}_{rela}} r_{ij}}{|\mathcal{N}_{rela}|} \quad (2.12)$$

$$h_{\mathcal{N}_v}^{k+1} = FUSE(h_{\mathcal{N}_v}^{k+1}, h_{\mathcal{N}_{rela}}^{k+1}) \quad (2.13)$$

$$h_{\mathcal{N}_{rela}}^{k+1} = FUSE(h_{\mathcal{N}_{rela}}^{k+1}, h_{\mathcal{N}_{rela}}^{k+1}) \quad (2.14)$$

trong đó, k là chỉ số vòng lặp thứ k , i là chỉ số số thứ tự của nút trong đồ thị; v_i là đỉnh thứ i của đồ thị; $\mathcal{N}_v, \mathcal{N}_{rela}$ là tập các nút kề với nút v_i tương ứng với cạnh nối hai đỉnh có hướng đi vào và đi ra; $h_{\mathcal{N}_v}^{k+1}, h_{\mathcal{N}_{rela}}^{k+1}$ là vector nhúng nút v_i theo từng chiều vào và chiều ra ở bước lặp $k+1$; $h_{\mathcal{N}_v}^k, h_{\mathcal{N}_{rela}}^k$ là vector nhúng quan hệ theo từng chiều vào và chiều ra của nút v_i ở bước lặp $k+1$; $h_{v_i}^k$ là nhúng của nút v_i tại bước lặp k ; r_{ij} là nhúng quan hệ giữa hai nút i và j , được khởi tạo ngẫu nhiên; $h_{\mathcal{N}_v}^{k+1}$ là vector nhúng của nút v_i theo cả chiều vào và chiều ra ở bước lặp $k+1$; $h_{\mathcal{N}_{rela}}^{k+1}$ là vector nhúng quan hệ của nút v_i theo cả chiều vào, chiều ra ở bước lặp $k+1$;

FUSE là hàm kết hợp thông tin, được định nghĩa:

$$FUSE(a, b) = z \odot a + (1 - z) \odot b \quad (2.15)$$

$$z = \sigma(W_z[a; b; a \odot b; a - b] + b_z) \quad (2.16)$$

với \odot là phép tính element-wise; σ là hàm *sigmoid*; W_z, b_z là các tham số, và z được gọi là "gated vector".

Cuối mỗi vòng lặp k , ta sẽ thực hiện cập nhật nhúng của nút v_i và nhúng quan hệ của nút v_i thông qua mạng hồi tiếp với nút có cổng (Gated Recurrent Unit - GRU) với trạng thái ẩn là thông tin vector nhúng ở vòng lặp trước, đầu vào là kết quả tổng hợp vector nhúng của nút v_i và kết quả tổng hợp vector nhúng quan hệ của nút v_i vừa tính qua hàm *FUSE*. Cụ thể ta có:

$$h_{v_i}^k = GRU(h_{v_i}^{k-1}, h_{\mathcal{N}_{v_i}}^k) \quad (2.17)$$

$$h_{rela_i}^k = GRU(h_{rela_i}^{k-1}, h_{\mathcal{N}_{rela_i}}^k) \quad (2.18)$$

Sau n vòng lặp, ta sẽ thu được nhúng của nút thứ i và nhúng quan hệ của nút thứ i lần lượt kí hiệu là $h_{c_i}^n$, $h_{rela_i}^n$. Trong đó, $h_{c_i}^n$ mang thông tin về ngữ cảnh của nút thứ i , $h_{rela_i}^n$ mang thông tin về cấu trúc cú pháp. Tổng hợp hai thông tin trên ta có đầu ra sau cùng là vector nhúng nút h_i^n với:

$$h_i^n = h_{c_i}^n + h_{rela_i}^n \quad (2.19)$$

Cuối cùng ta cần tính toán ra vector nhúng đồ thị (graph embedding) thay vì dùng lại ở mức nhúng của nút vì nhúng đồ thị sẽ truyền tải được thông tin của toàn bộ đồ thị và điều này giúp ích cho phần giải mã ở pha sau. Phương pháp tính toán nhúng đồ thị ở đây là sử dụng phương pháp gộp (Pooling), trong nghiên cứu của (Xu et al., 2018) về mạng Graph2Seq đã thấy rằng không có sự khác biệt nào về hiệu năng giữa ba phương pháp Pooling là gộp cực đại (Max Pooling), gộp cực tiểu (Min Pooling) và gộp trung bình (Average Pooling). Vì vậy trong bài sẽ sử dụng phương pháp tính toán gộp cực đại để tổng hợp ra nhúng đồ thị.

Thực hiện đưa toàn bộ các vector nhúng h_i^n của từng nút qua mạng thần kinh đặc (fully-connected neural network) và áp dụng thuật toán Max Pooling, ta sẽ thu được vector nhúng mức đồ thị h^G .

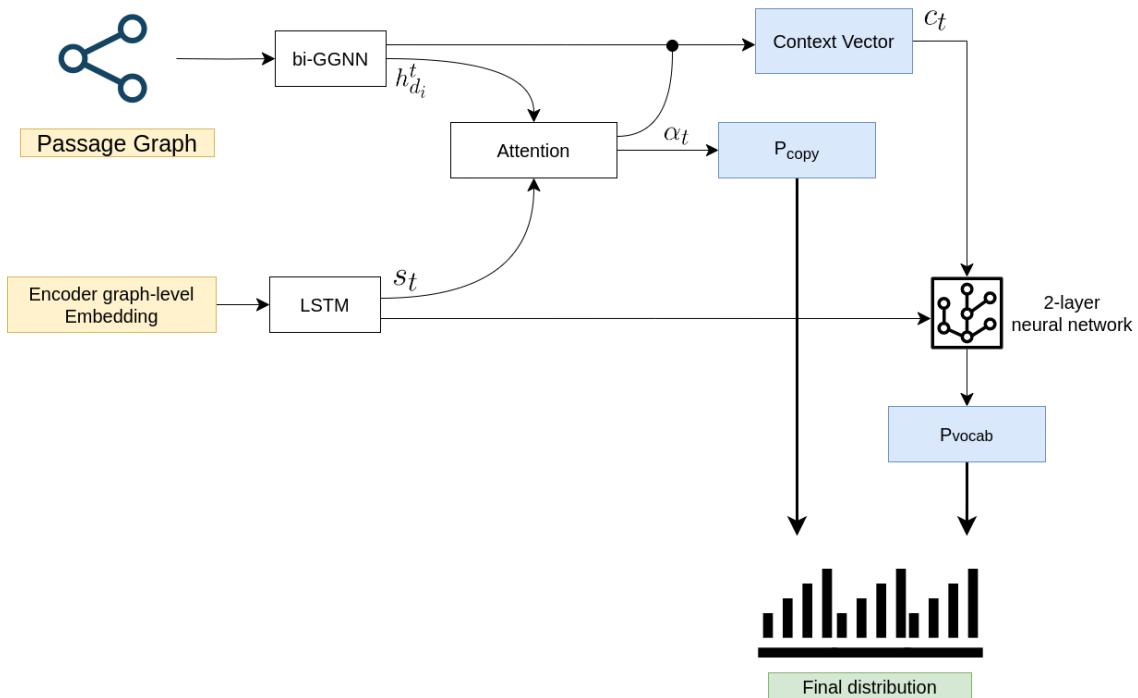
$$h^G = MaxPool(W^G h_i^n) \quad (2.20)$$

trong đó, W^G là trọng số huấn luyện.

2.3.2 Bộ giải mã quan hệ (Iterative Graph Network-based Decoder)

Bộ giải mã (decoder) sử dụng trong đồ án là mô hình bộ nhớ Ngắn hạn Dài kết hợp cơ chế chú ý và cơ chế sao chép (copy mechanism) (Sun et al., 2018). Tuy nhiên, những nghiên cứu trước đây về bài toán sinh câu hỏi hầu hết bỏ qua hai điểm

đáng chú ý có thể đóng góp vào chất lượng câu hỏi sinh ra, đó là cấu trúc thông tin ẩn ở những từ đã sinh trước đó trong quá trình giải mã và ảnh hưởng của từ sao chép trong đoạn văn. Vì vậy để giải quyết hai vấn đề trên, nhà nghiên cứu Zichu Fei và các cộng sự năm 2021 đã đề xuất một bộ giải mã dựa trên mạng đồ thị lặp (Iterative Graph Network-based Decoder - IGND).



Hình 2.2: Mô tả luồng thực thi trong bộ giải mã Decoder trong bước thời gian thứ t .

Đầu tiên, ta sẽ khởi tạo vector trạng thái ẩn s_0 và vector ngữ cảnh c_0 (vector sẽ được cập nhật theo cơ chế chú ý) dựa trên vector nhúng mức độ thị h^G đã tính toán ở bộ mã hóa quan hệ (công thức 2.20).

$$s_0 = \tanh(W_{t2}\tanh(W_{t1}h^G + b_{t1}) + b_{t2}) \quad (2.21)$$

$$c_0 = s_0 \quad (2.22)$$

với $W_{t2}, W_{t1}, b_{t1}, b_{t2}$ là các trọng số huấn luyện. Trong mỗi bước giải mã thứ t , mạng LSTM sẽ nhận đầu vào gồm nhúng w_{t-1} của từ đã sinh ra trước đó y_{t-1} , vector ngữ cảnh c_{t-1} và trạng thái ẩn s_{t-1} , để tính ra trạng thái ẩn ở bước thứ t đó:

$$s_t = LSTM([w_{t-1}; c_{t-1}], s_{t-1}) \quad (2.23)$$

Cùng với mạng LSTM, ta sử dụng đồ thị G_d là đồ thị tạo ra từ đoạn văn ở phần "Mã hóa mối quan hệ giữa các từ với mạng thần kinh đồ thị". Nhận thấy những từ được sao chép từ đoạn văn sang câu hỏi đóng vai trò quan trọng trong ngữ nghĩa

của cả câu hỏi, ta sẽ sử dụng thêm nhãn vai trò (role tags) ([8] Fei et al., 2021), trong đó những nhãn đó sẽ có giá trị {“no-copy”, “answer”, “copied”}. Ban đầu, các nhãn được khởi tạo mặc định cho các từ trong đoạn văn là “no-copy”, và với những từ nằm trong đoạn văn mà nó nằm trong câu trả lời là nhãn “answer” và nhãn “answer” sẽ không thay đổi trong toàn bộ quá trình giải mã. Trong mỗi bước giải mã, các nhãn trên sẽ được cập nhật lại, nếu một từ được sao chép vào câu hỏi thì nhãn vai trò của nút đó sẽ được đổi thành “copied”.

$$tag_i = \begin{cases} 0 & \text{nếu từ thứ } i \text{ là từ được sao chép} \\ 1 & \text{nếu từ thứ } i \text{ nằm trong câu trả lời} \\ 2 & \text{nếu từ thứ } i \text{ không phải từ sao chép} \end{cases}$$

Tại bước giải mã thứ t (Hình 2.2), bộ giải mã thực hiện tính toán ở hai mô đun: (i) bi-GGNN để lấy nhúng ở mức độ thị với thông tin về nhãn vai trò và (ii) LSTM để lấy vector trạng thái ẩn. Sau đó, áp dụng cơ chế chú ý và các phép tính toán cần thiết để tính ra phân phối xác suất trên tập từ điển sau cùng.

Cụ thể, ta áp dụng mạng bi-GGNN với hàm kết tập MEAN để tính toán ra nhúng biểu diễn của nút. Tại mỗi bước giải mã t , ta khởi tạo nhúng của nút thứ i bằng vector nhúng của nút thứ i cuối cùng trong bộ mã hóa là h_i^n kết hợp với nhãn vai trò.

$$h_{d_i}^t = [h_i^n, r_i^t] \quad (2.24)$$

trong đó, d_i là chỉ số chỉ nút thứ i ở bước giải mã t , r_i^t là nhúng của nhãn vai trò (role tag) của nút i ở bước giải mã t . h_i^n là nhúng nút được tính ở công thức (2.19).

Áp dụng tương tự như các công thức (2.9) đến (2.18), sau n vòng lặp ở mạng bi-GGNN, ta có nhúng nút thứ i là $h_{d_i}^t$.

Sau đó, ta sẽ áp dụng cơ chế chú ý Perceptron đa tầng và thực hiện tính toán ra vector ngữ cảnh c_t , cụ thể:

$$e_{t,i} = v^T \tanh(W_s s_t + W_h h_{d_i}^t) \quad (2.25)$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^N \exp(e_{t,j})} \quad (2.26)$$

$$c_t = \sum_{i=1}^N \alpha_{t,i} h_{d_i}^t \quad (2.27)$$

trong đó, $e_{t,i}$ là điểm số cơ chế chú ý, $\alpha_{t,i}$ là điểm số chuẩn hóa qua hàm *softmax*;

W_s, W_h, v^T là các trọng số huấn luyện.

Bộ giải mã của mô hình được phát triển dựa trên framework sinh con trỏ (pointer-generator) với hai chế độ sinh (generation mode) và sao chép (copy mode). Trong quá trình sinh từ ở bước giải mã thứ t , ta có phân bố xác suất trên toàn tập từ điển P_{vocab} được tính dựa trên vector trạng thái ẩn s_t và vector ngữ cảnh c_t qua mạng thần kinh đặc hai tầng.

$$P_{vocab} = \text{softmax}(g(s_t; c_t)) \quad (2.28)$$

với g là mạng thần kinh lan truyền tiền hai tầng với hàm kích hoạt là hàm kích hoạt nội bộ tối đa (maxout internal activation).

Trong đoạn văn cho trước có thể tồn tại những từ hiếm gặp như tên người, tên địa danh, hay các từ phiên âm nước ngoài. Vì vậy, cơ chế sao chép (copy mechanism) được sử dụng để sao chép trực tiếp các từ ngoài từ điển đó (Out-of-vocabulary) vào câu hỏi và tích hợp thêm thông tin về sự chú ý của từ vào phân bố xác suất trên tập từ điển trong quá trình sinh từ P_{vocab} . Công thức tính phân bố xác suất trên tập từ điển cuối cùng P_{final} tại bước thứ t :

$$P_{final} = g_t P_{vocab} + (1 - g_t) P_{copy} \quad (2.29)$$

trong đó, $P_{copy} = \alpha_t$ vì nhận thấy, trọng số chú ý α_t đo mối quan hệ giữa các từ và trạng thái ẩn của bộ giải mã nên có thể coi trọng số α_t như xác suất sao chép từ; g_t được tính toán từ vector trạng thái ẩn s_t , vector ngữ cảnh c_t , nhưng từ thứ t là w_t theo công thức:

$$g_t = \sigma(W_g(c_t + s_t + w_t)) \quad (2.30)$$

với W_g là tham số huấn luyện và σ là hàm sigmoid.

2.3.3 Hàm mất mát

Hàm mất mát sử dụng trong mô hình Graph2Seq là hàm mất mát log âm (negative log likelihood) cho câu hỏi sinh ra y :

$$\frac{1}{L} \sum_{t=1}^T \log(P(y_t)) \quad (2.31)$$

2.4 Kết chương

Trong chương 2, em đã trình bày về kiến trúc mô hình học sâu Graph2Seq gồm bộ phận chính là bộ mã hóa với tầng làm giàu thông tin và mã hóa đồ thị quan hệ,

CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT

bộ giải mã IGND với khả năng mô hình cấu trúc thông tin ẩn của chuỗi từ đã sinh ra trước đó. Trong chương 3, em sẽ trình bày về bộ dữ liệu SQuAD phiên bản 1.1 và các phương pháp xây dựng đồ thị từ đoạn văn bản.

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

3.1 Tổng quan giải pháp

Trong các nghiên cứu về mô hình học sâu cho bài toán sinh câu hỏi, bộ dữ liệu "squad-split" ([15] Du et al., 2017) là bộ dữ liệu SQuAD đã được tiền xử lý trước được sử dụng rất nhiều trong nghiên cứu. Tuy nhiên bộ dữ liệu đó đã xử lý thu gọn đoạn văn lại bằng cách chỉ giữ lại câu chứa câu trả lời hoặc nếu câu trả lời nằm ở nhiều câu thì sẽ nối các câu đó lại (dựa trên thông tin về vị trí câu trả lời trong đoạn văn). Điều này có thể sẽ làm mất thông tin ngữ cảnh của toàn bộ đoạn văn vì đoạn văn thực tế có kích thước lớn và nhiều thông tin liên quan đến câu trả lời có thể nằm các câu văn khác. Vì vậy đồ án này sử dụng bộ dữ liệu SQuAD phiên bản 1.1 ([14] Rajpurkar et al., 2016) gốc chưa qua tiền xử lý để huấn luyện.

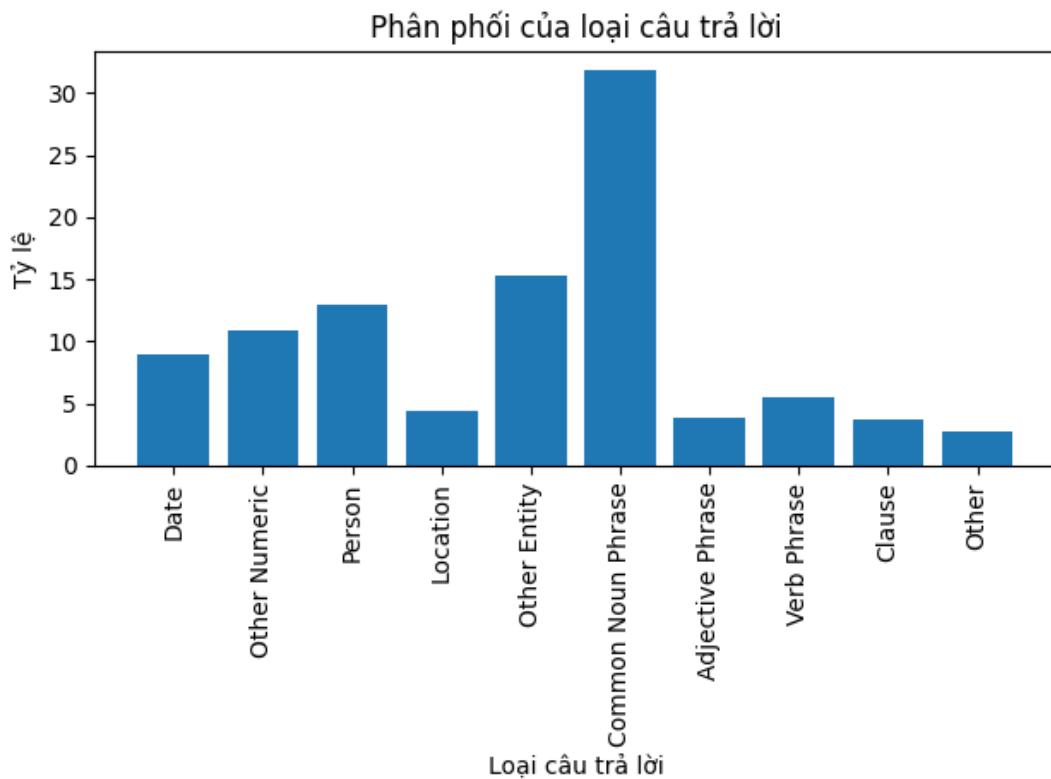
Phương pháp xây dựng đồ thị cho đoạn văn vẫn sẽ dựa trên cây phân tích cú pháp cho từng câu, sau đó nối các cây đó lại. Ngoài phương pháp nối cây dựa trên nút tương ứng với biên của câu sử dụng trong nghiên cứu mô hình "Graph2seq +RL+ BERT" ([12] Chen et al., 2020), ta có thể sử dụng nút tương ứng với HEAD của mỗi câu hoặc sử dụng phân giải đồng tham chiếu để biểu diễn thực thể với đầy đủ thông tin hơn.

3.2 Bộ dữ liệu SQuAD phiên bản 1.1

Bài toán sinh câu hỏi QG cũng giống như bài toán hỏi đáp (Question Answering) về yêu cầu dữ liệu phải chứa một số lượng lớn câu hỏi và đoạn văn. Ta nhận thấy cấu trúc dữ liệu của hai bài toán trên đều bao gồm ba yếu tố chính: đoạn văn (paragraph), câu hỏi (question) và câu trả lời (answer) nên ta hoàn toàn có thể xử lý dữ liệu cho bài toán hỏi đáp để sử dụng cho bài toán sinh câu hỏi. Một số tập dữ liệu đã được các nghiên cứu về sinh câu hỏi sử dụng như SQuAD, MS MARCO, NewsQA, RACE, ... và mỗi bộ dữ liệu đều thuộc kiểu câu hỏi khác nhau, ví dụ như sinh câu hỏi trắc nghiệm, điền vào chỗ trống trong câu (fill-in-the-blank questions), factoid/non-factoid. Trong đó, bộ dữ liệu "squad-split" là bộ dữ liệu được sử dụng để đánh giá thực nghiệm trong rất nhiều nghiên cứu về QG như mô hình "Graph2seq +RL+ BERT" ([12] Chen et al., 2020), mô hình NQG++ ([13] Zhou et al., 2018). "squad-split" ([15] Du et al., 2017) là bộ dữ liệu đã thông qua tiền xử lý của bộ dữ liệu Stanford Question Answering Dataset (SQuAD) ([14] Rajpurkar et al., 2016). Đầu tiên sử dụng thư viện Stanford CoreNLP ([24] Manning et al., 2014) để tách token (tokenization), tách câu và lấy thông tin về NER, POS; sau đó với thông tin về câu trả lời trong đoạn văn cho trước, ta sẽ lấy câu chứa câu trả lời đó, còn nếu câu trả lời nằm ở hai câu trở lên thì ta sẽ thực hiện

nối các câu đó lại thành một câu. Điều này sẽ giúp giảm tải tính toán cho hệ thống nhờ đoạn văn đầu vào đã được cắt gọn đi, nhưng khi chỉ lựa chọn những câu chứa câu trả lời thì có thể làm mất thông tin bổ trợ cho câu trả lời. Vì đồ án thực hiện việc xây dựng đồ thị với đầu vào là một đoạn văn bản nên bộ dữ liệu sử dụng trong đồ án là bộ dữ liệu SQuAD phiên bản 1.1 chưa qua tiền xử lý.

Bộ dữ liệu SQuAD phiên bản 1.1 là bộ dữ liệu phục vụ cho tác vụ đọc hiểu tổng quát được tổng hợp từ gần 500 bài viết trên Wikipedia và gồm hơn 107,000 cặp câu hỏi và câu trả lời. Bộ dữ liệu cũng rất đa dạng về thể loại câu trả lời như cụm danh từ, địa điểm, cụm tính từ, ... và có phân bố như hình vẽ 3.1. Điều này dẫn đến sự đa dạng về cấu trúc cú pháp của câu hỏi và có thể phải thực hiện suy diễn để đưa ra câu trả lời chính xác.



Hình 3.1: Phân bố của thể loại câu trả lời trong bộ dữ liệu SQuAD 1.1.

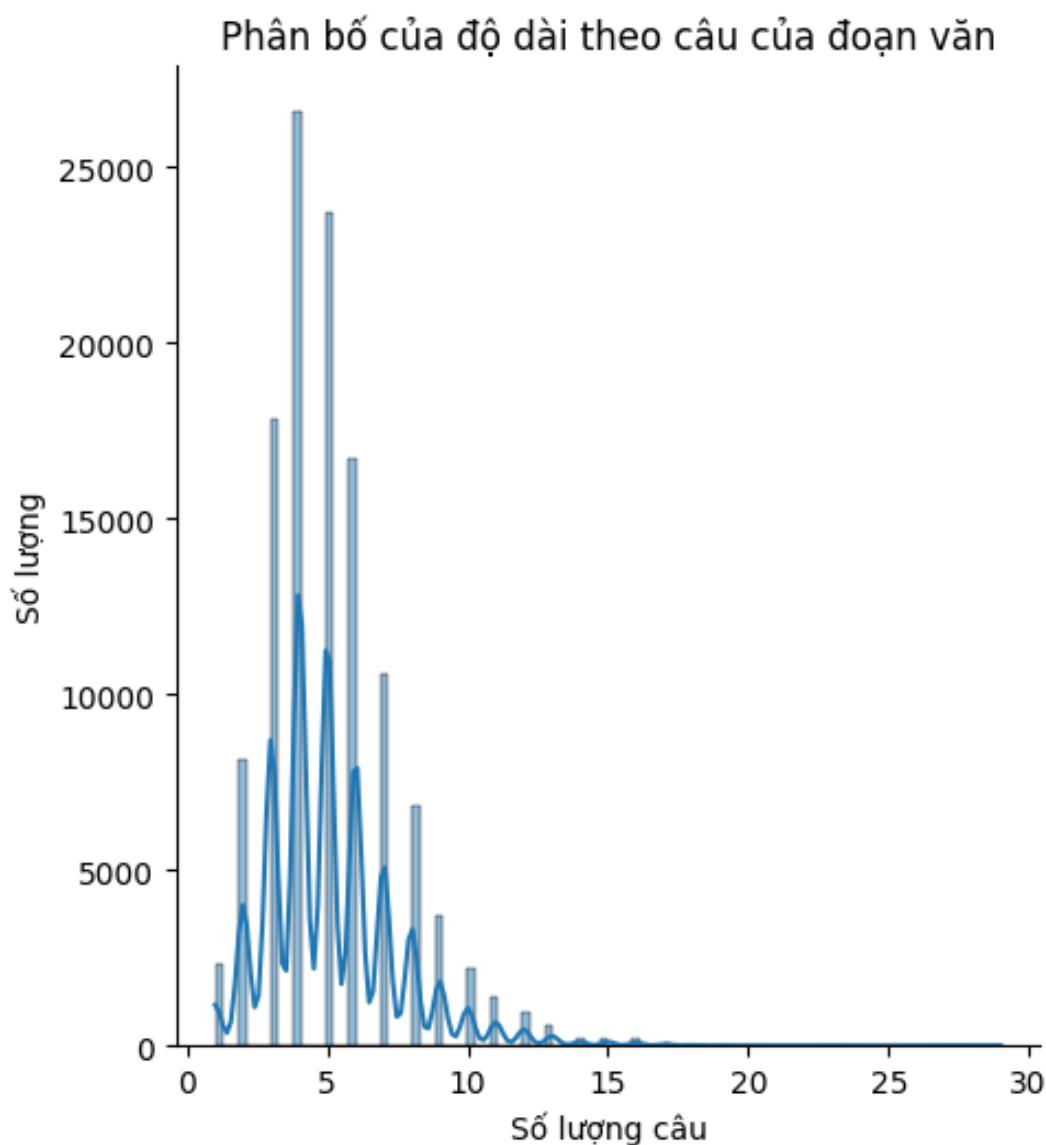
Một số thống kê khác về bộ dữ liệu SQuAD 1.1

Bộ dữ liệu ba gồm ba tập (train, dev, test) được chia lại theo tỉ lệ 8:1:1 với số lượng tương ứng là 97861, 12232, 12232.

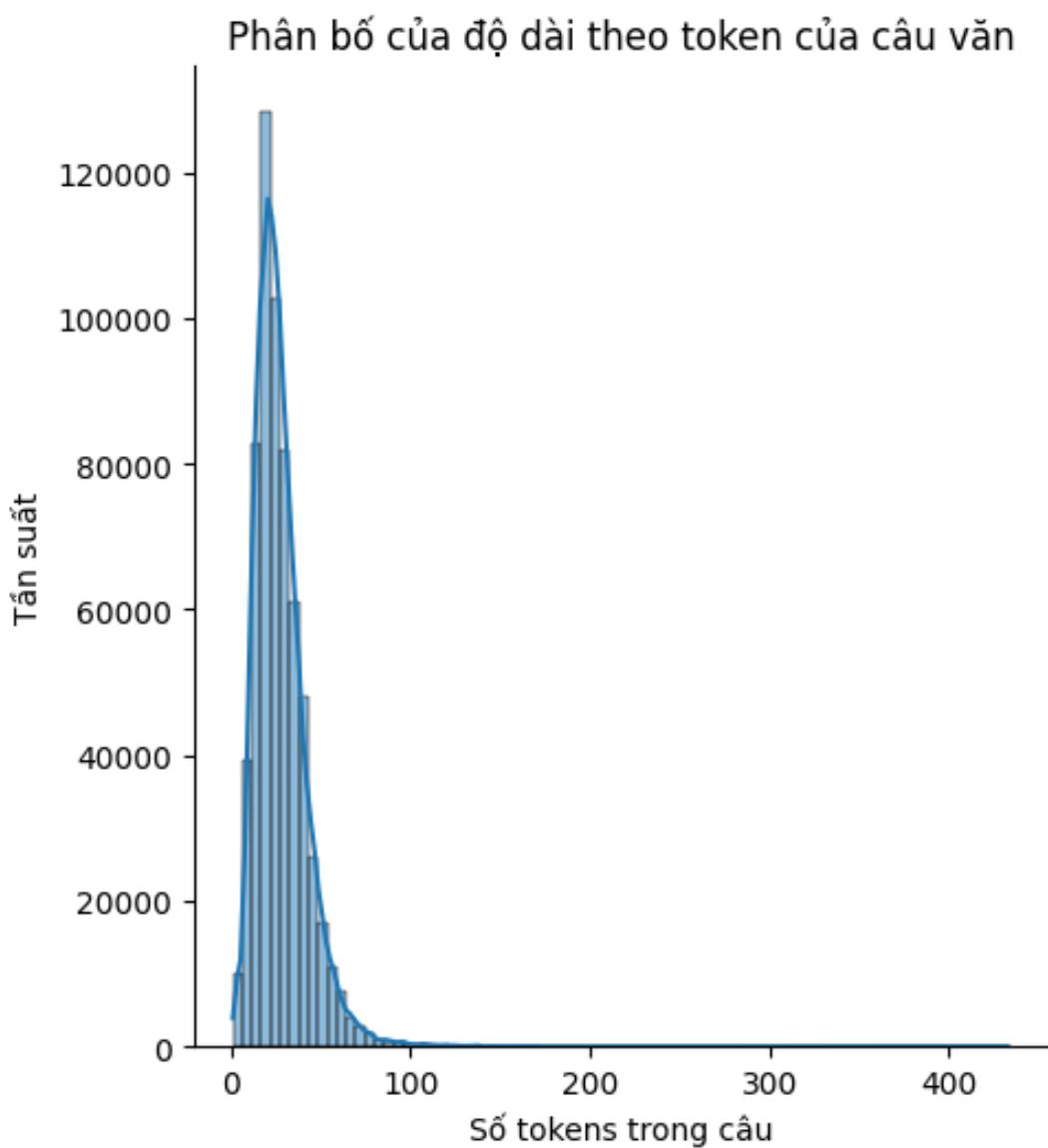
Trong bộ dữ liệu SQuAD, mỗi văn bản đầu vào gồm duy nhất một đoạn văn và số lượng câu trong đoạn văn có phân phối như hình vẽ 3.2, trong đó số lượng câu văn chủ yếu từ 2 đến 8 câu trong một đoạn, mỗi câu trung bình có 27 tokens (Hình 3.3) và số lượng token trung bình trong đoạn văn là khoảng 141 tokens (phân bố

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

nhiều trong hình vẽ 3.4 và 3.5). Điều này cho thấy sự phức tạp về mặt cú pháp và ngữ nghĩa và thách thức khi có quá nhiều thông tin nhiễu của đầu vào. Đối với câu hỏi, số lượng token trong một câu nằm trong khoảng từ 3 đến nhiều nhất là 60 tokens (Hình 3.6, 3.7); còn thống kê với câu trả lời đa phần là câu ngắn, độ dài trung bình là 3 tokens (Hình 3.8, 3.9).

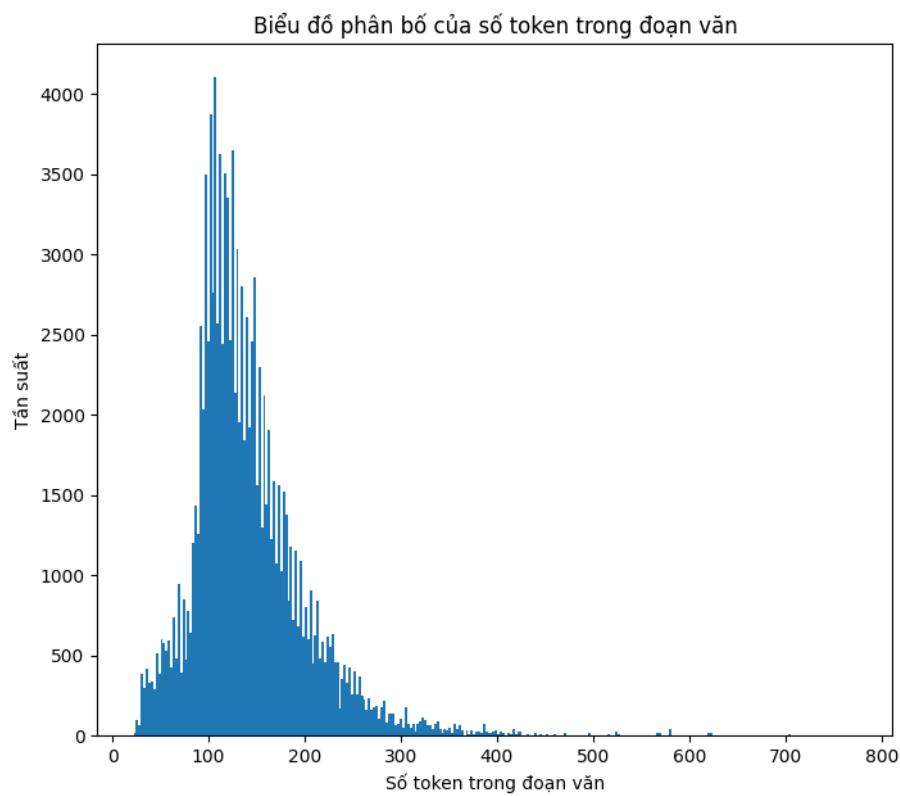


Hình 3.2: Phân bố về số lượng câu của đoạn văn trong bộ dữ liệu SQuAD 1.1.

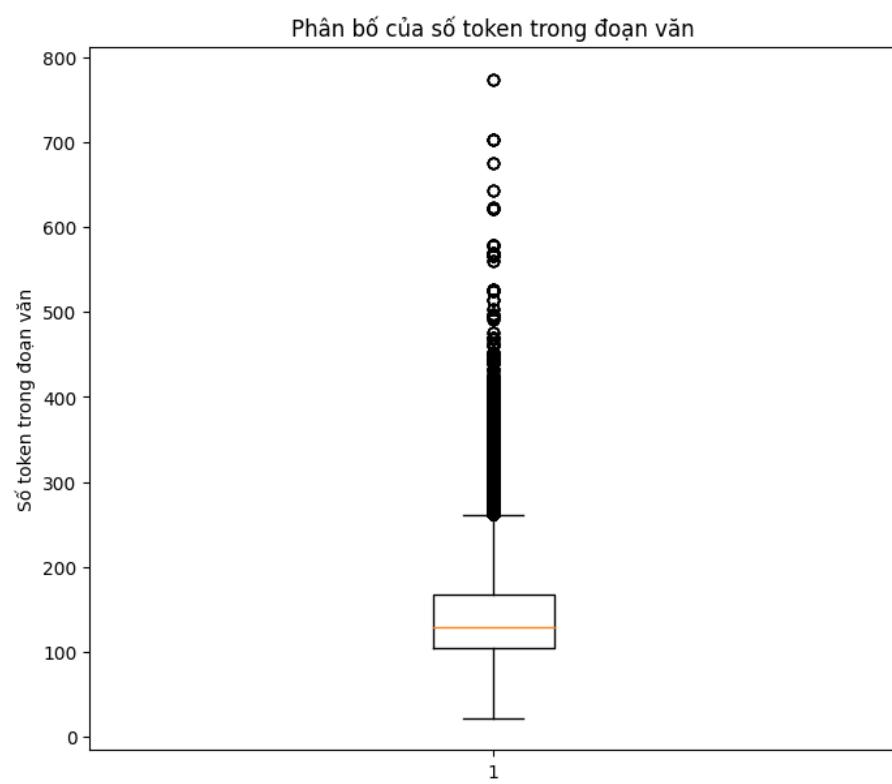


Hình 3.3: Phân bố về số lượng câu của đoạn văn trong bộ dữ liệu SQuAD 1.1.

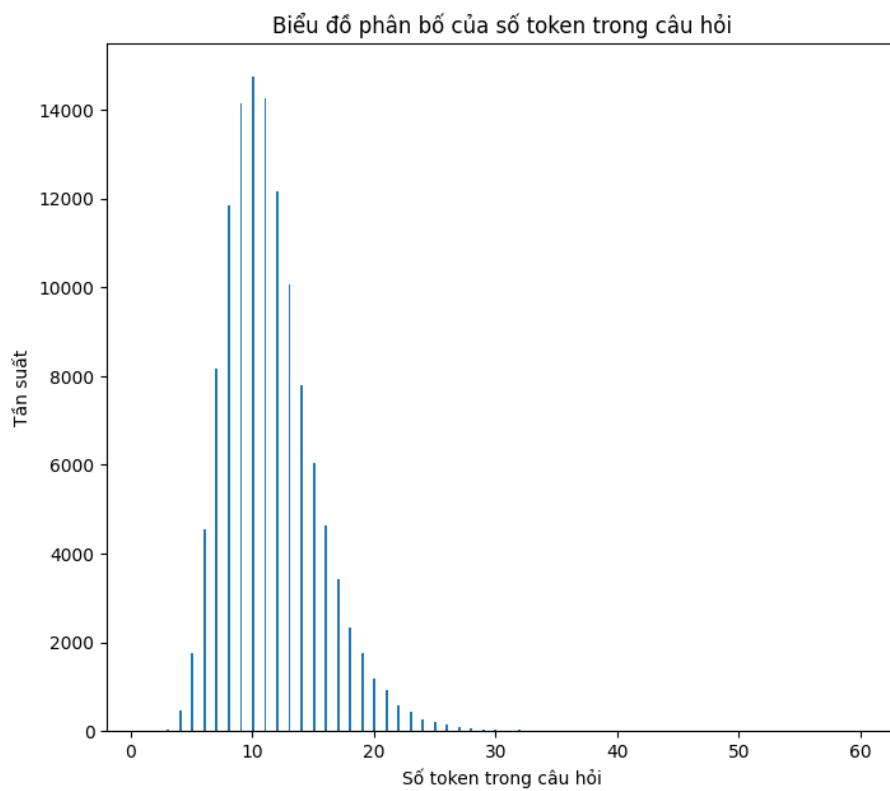
CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT



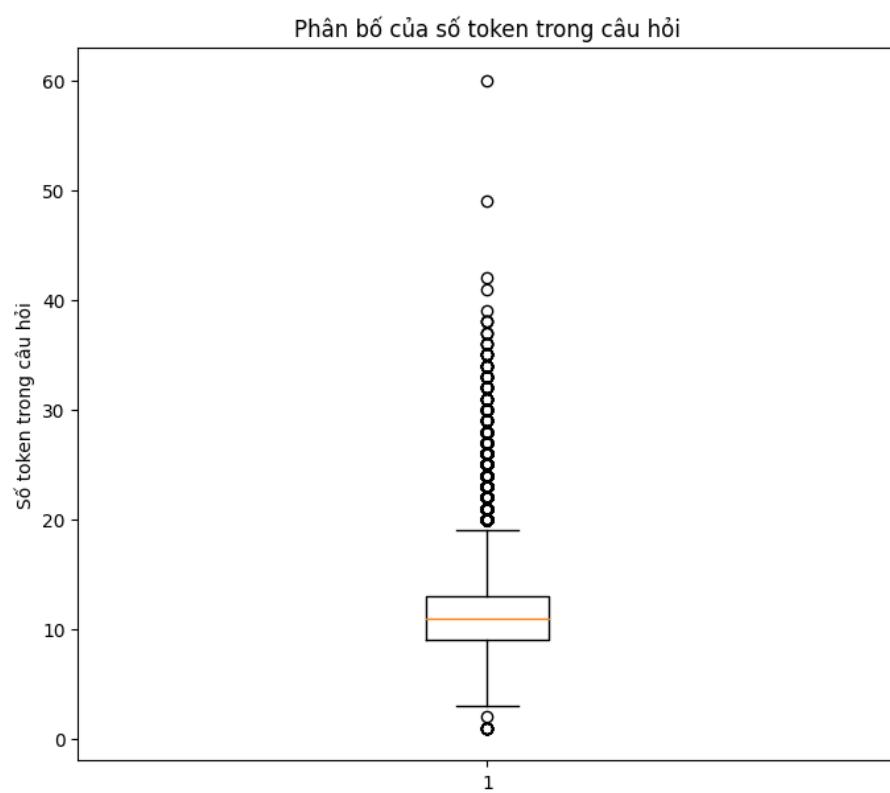
Hình 3.4: Biểu đồ về phân bố về số lượng token trong đoạn văn trong bộ dữ liệu SQuAD 1.1.



Hình 3.5: Biểu đồ boxplot về phân bố về số lượng token trong đoạn văn trong bộ dữ liệu SQuAD 1.1.

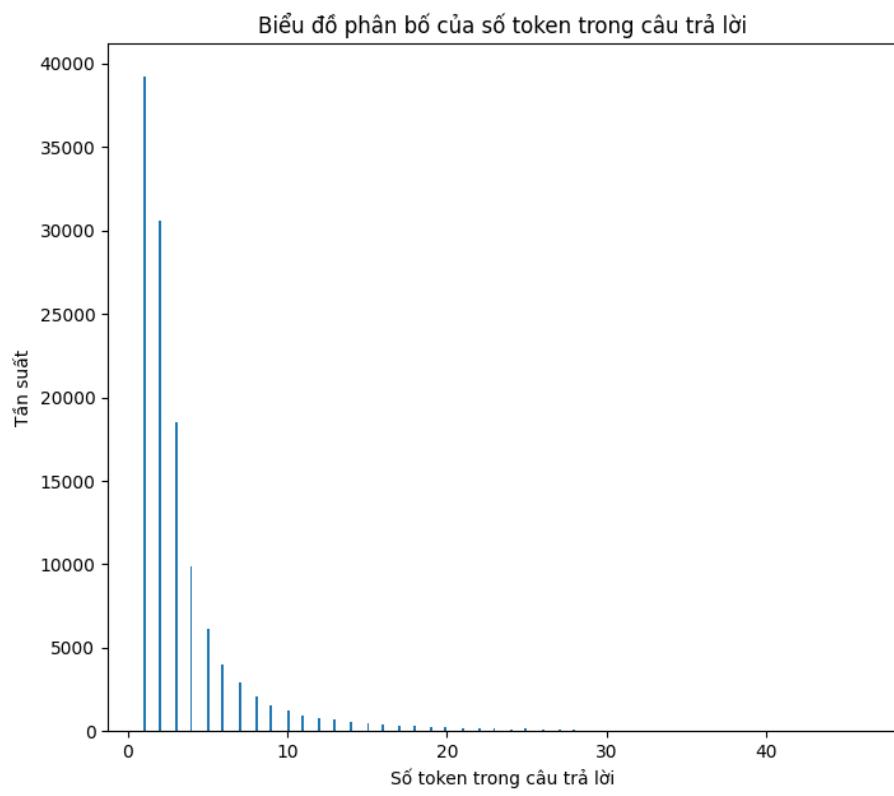


Hình 3.6: Biểu đồ về phân bố về số lượng token của câu hỏi trong bộ dữ liệu SQuAD 1.1.

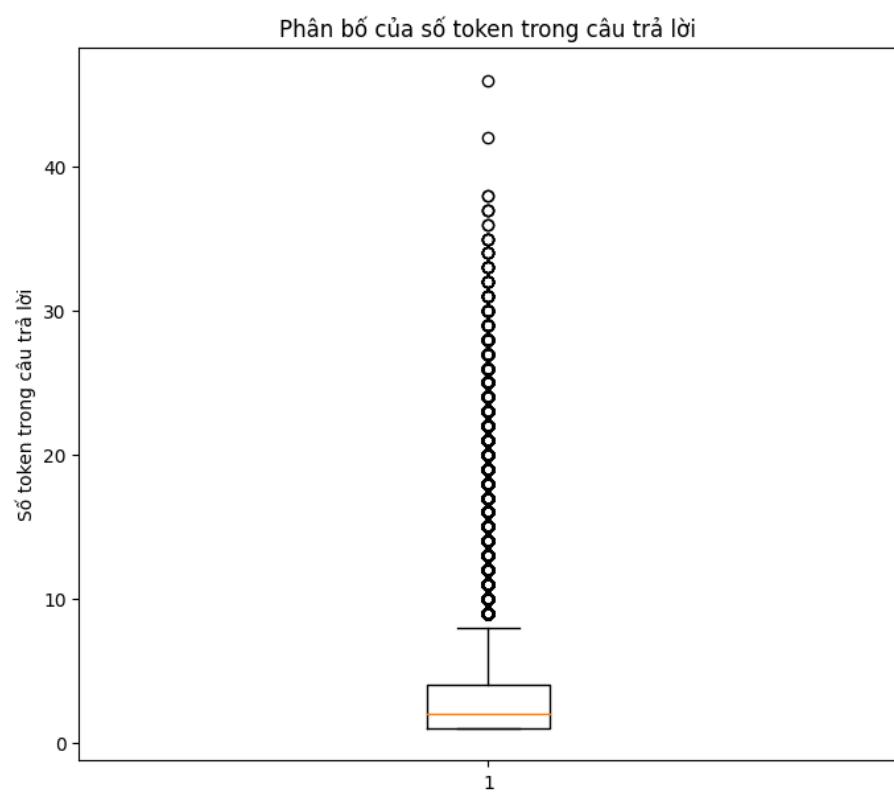


Hình 3.7: Biểu đồ boxplot về phân bố về số lượng token của câu hỏi trong bộ dữ liệu SQuAD 1.1.

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT



Hình 3.8: Biểu đồ về phân bố về số lượng token của câu trả lời trong bộ dữ liệu SQuAD 1.1.



Hình 3.9: Biểu đồ boxplot về phân bố về số lượng token của câu trả lời trong bộ dữ liệu SQuAD 1.1.

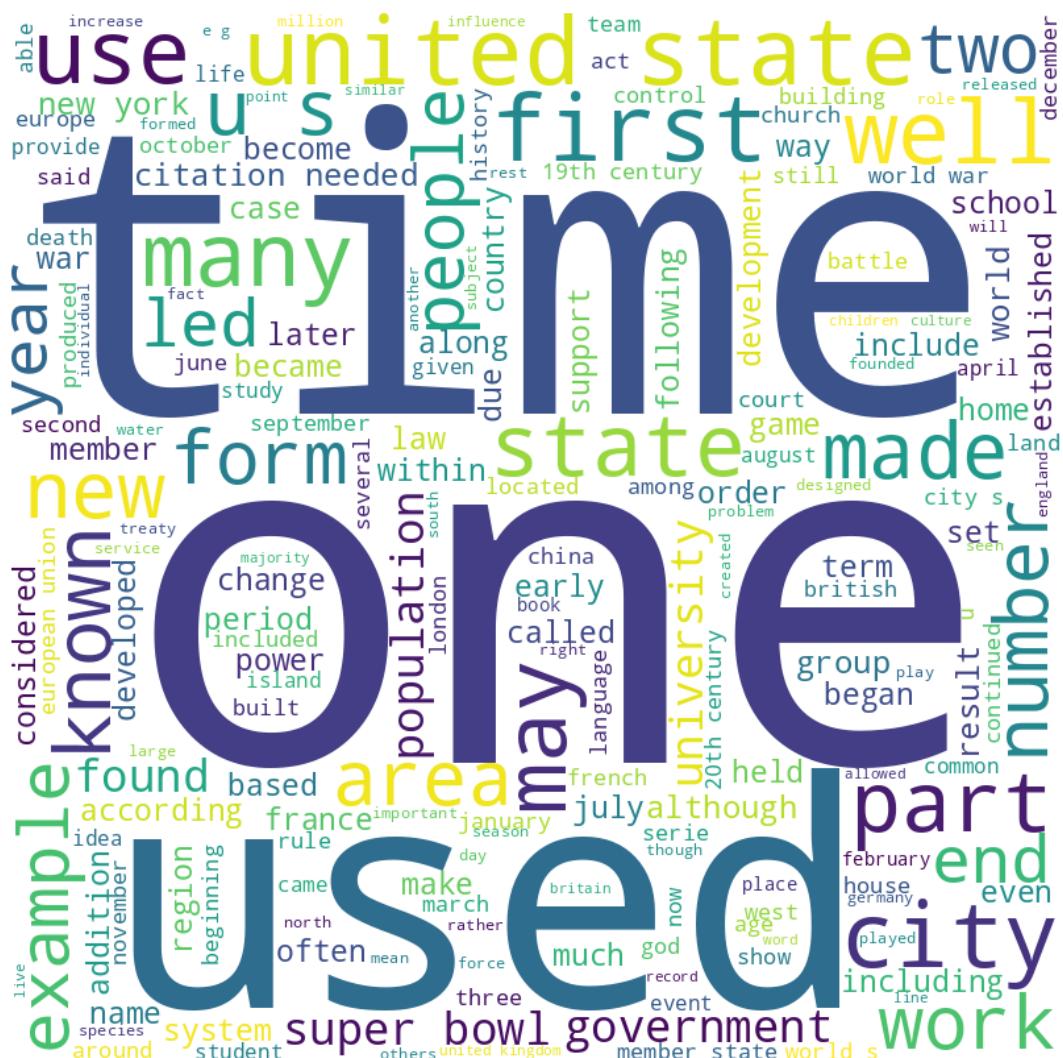
Thực thể có tên (Named Entity) là các từ hoặc cụm từ đại diện cho các tên riêng như tên người, địa điểm, tổ chức, thời gian, ... và nó đóng vai trò rất quan trọng trong bài toán sinh câu hỏi. Trong lúc tạo câu hỏi, một số các thực thể như địa danh, tên người hay các từ phiên âm không xuất hiện trong từ điển, khi đó việc giữ nguyên các từ hoặc cụm từ đó là cần thiết để đảm bảo câu hỏi sinh ra được cụ thể và có ý nghĩa. Bảng (3.1) thống kê số lượng của các nhãn NER xuất hiện trong từng tập train, dev và test.

Nhãn NER	Tập train	Tập dev	Tập test
GPE	31764	2120	899
CARDINAL	31698	2751	709
PERCENT	3810	260	53
ORG	14093	1907	357
DATE	39832	3362	905
PERSON	19483	1858	504
NORP	23498	1165	955
TIME	1061	124	14
EVENT	713	47	23
FAC	638	57	15
LOC	4025	280	182
QUANTITY	2571	287	47
ORDINAL	7176	660	158
PRODUCT	480	42	24
LANGUAGE	812	26	14
MONEY	1459	163	21
LAW	207	85	2
WORK_OF_ART	77	8	5

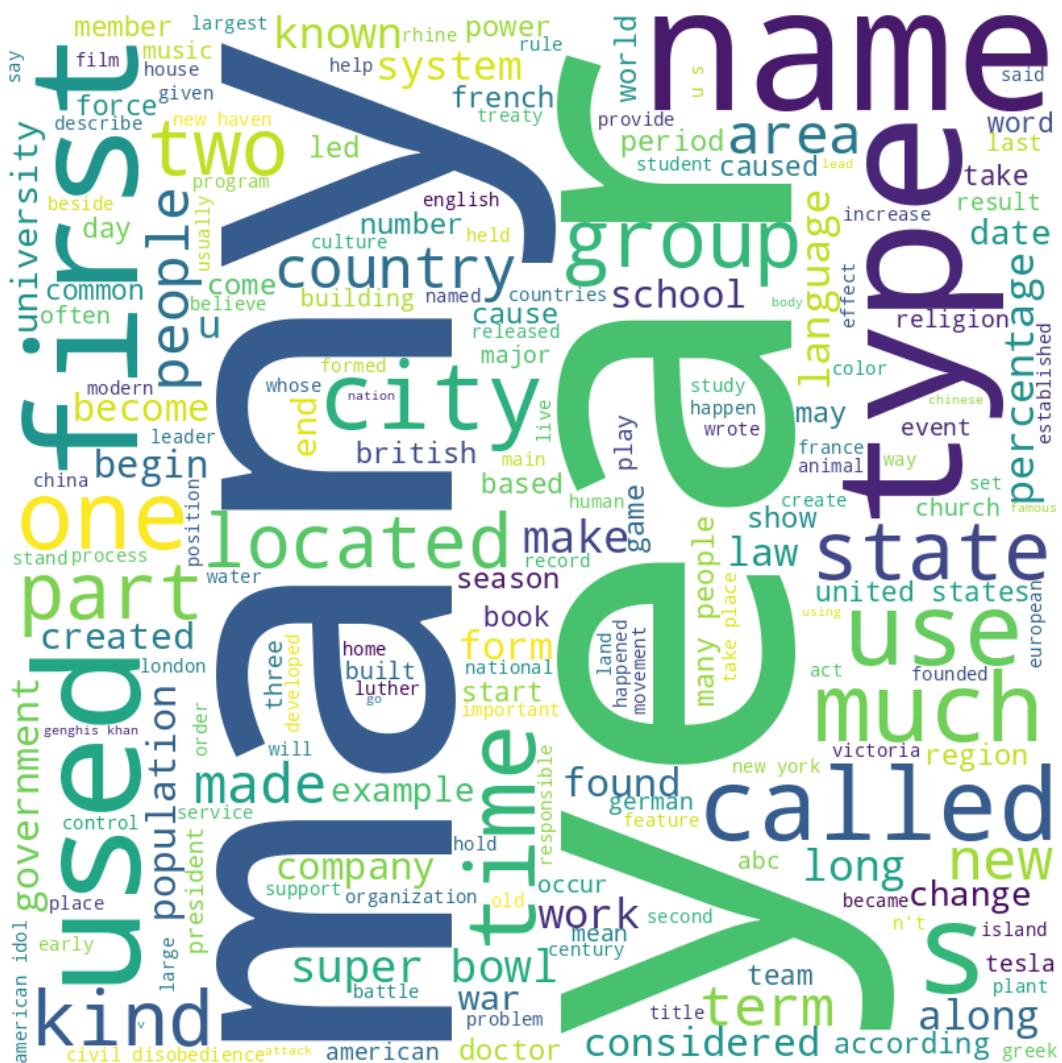
Bảng 3.1: Số lượng các nhãn NER xuất hiện trong các tập train, dev và test.

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

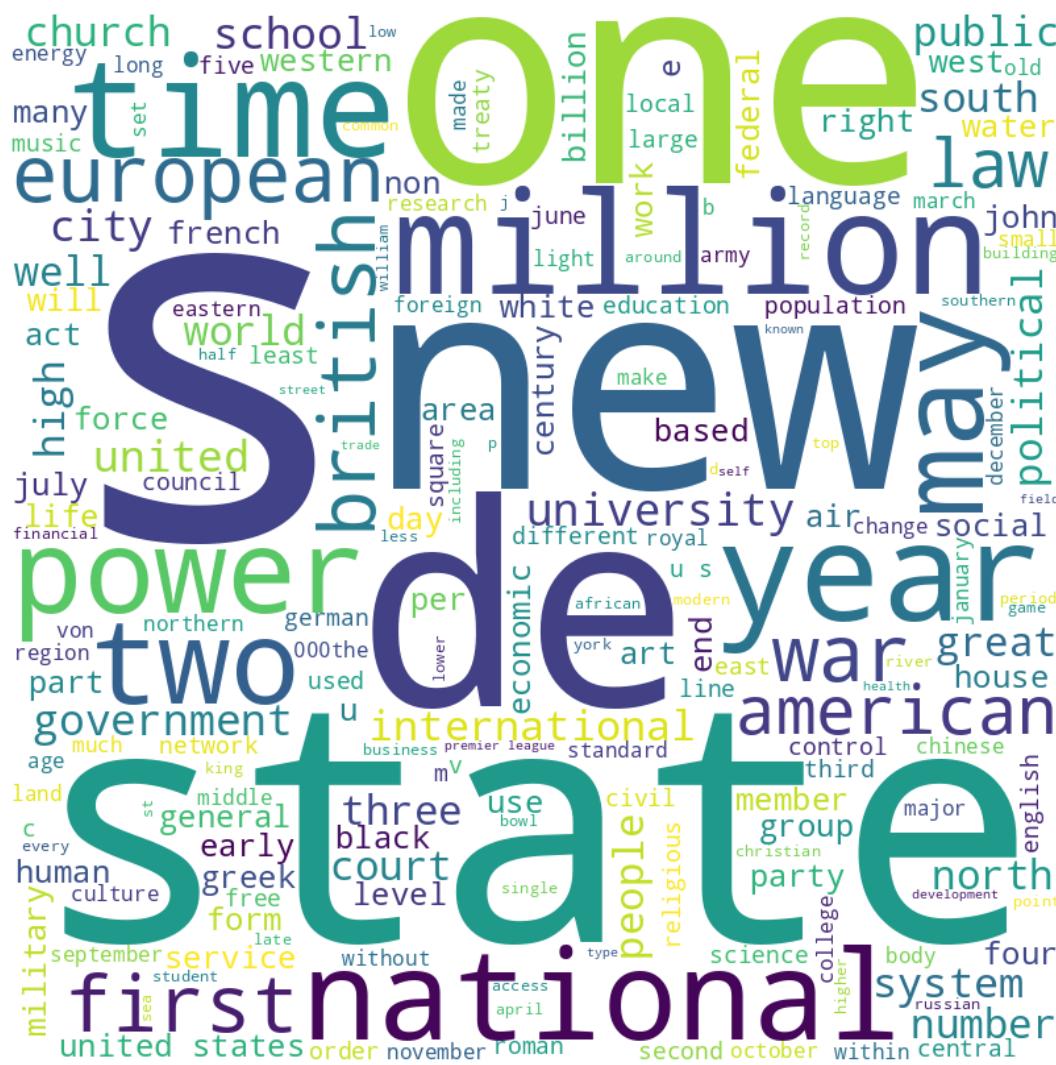
Thống kê về tần suất xuất hiện của các từ vựng trong bộ dữ liệu SQuAD 1.1 thể hiện qua WordCloud. Đối với đoạn văn bản (Hình 3.10), vì được tổng hợp từ 536 bài viết tiếng Anh trên Wikipedia nên về chủ đề của các bộ dữ liệu trải đều trên nhiều lĩnh vực như âm nhạc, lễ hội, nội dung trừu tượng, ... Tần suất xuất hiện của các từ vựng trong câu trả lời như hình (3.12) đã cho thấy các thể loại trong câu trả lời rất đa dạng, và đa phần chỉ thời gian, địa điểm và dạng số học. Còn đối với tần suất từ vựng trong câu hỏi, dựa vào những từ khóa xuất hiện nhiều như "located", "called", "time", "area", "name" (Hình 3.11), ta có thể thấy các câu hỏi thường tập trung vào 3 dạng câu "what", "where", "when". Ta sẽ thống kê rõ ràng hơn về các kiểu câu hỏi trong tập dữ liệu.



Hình 3.10: Các từ vựng xuất hiện trong đoạn văn trong bộ dữ liệu SQuAD 1.1.

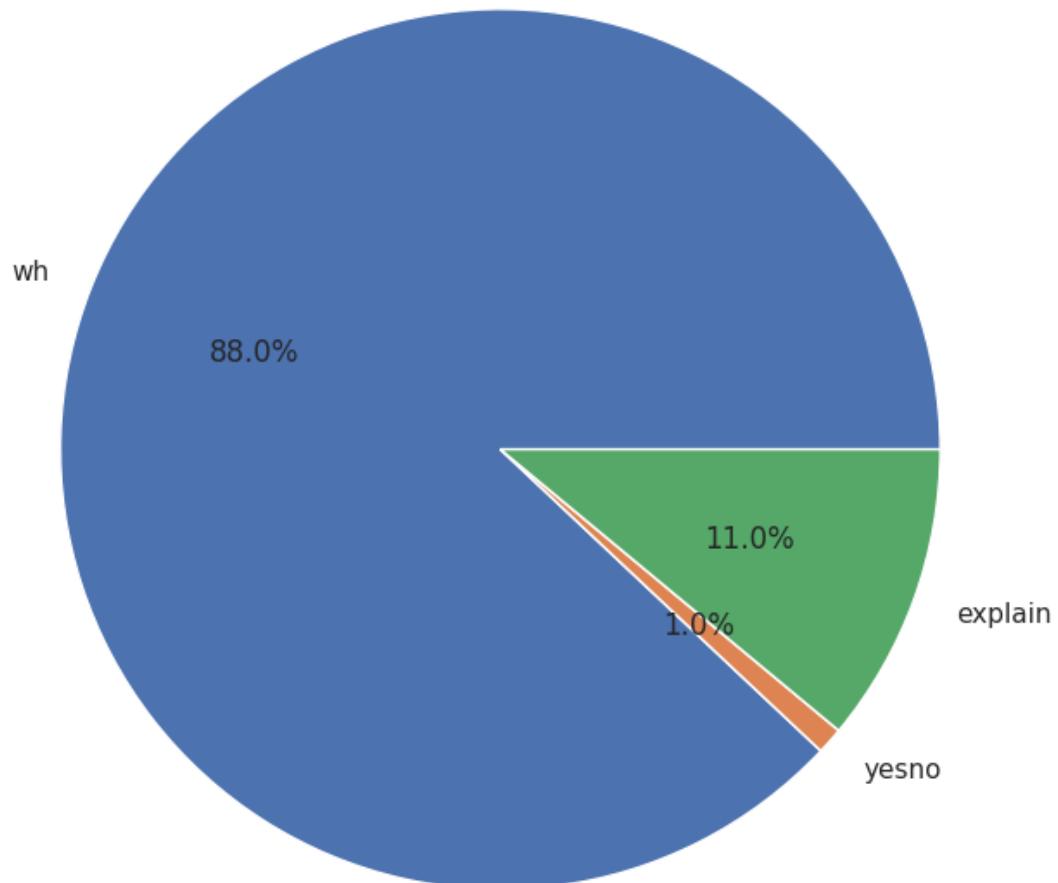


Hình 3.11: Các từ vựng xuất hiện trong câu hỏi trong bộ dữ liệu SQuAD 1.1.

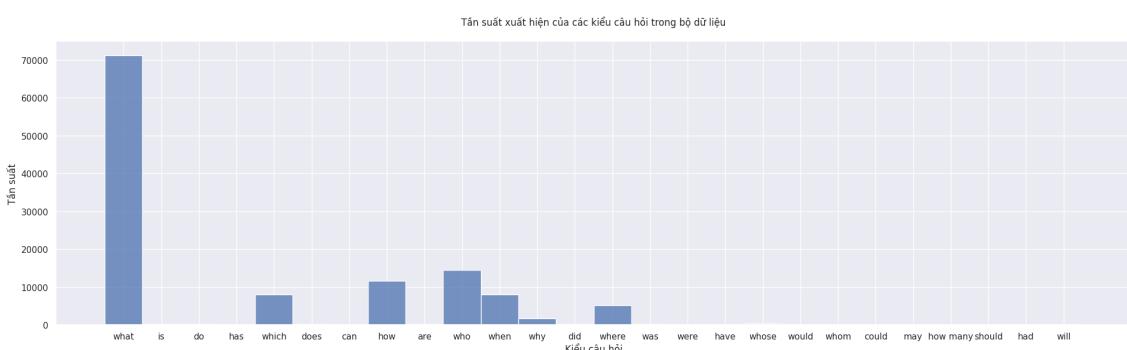


Hình 3.12: Các từ vựng xuất hiện trong câu trả lời trong bộ dữ liệu SQuAD 1.1.

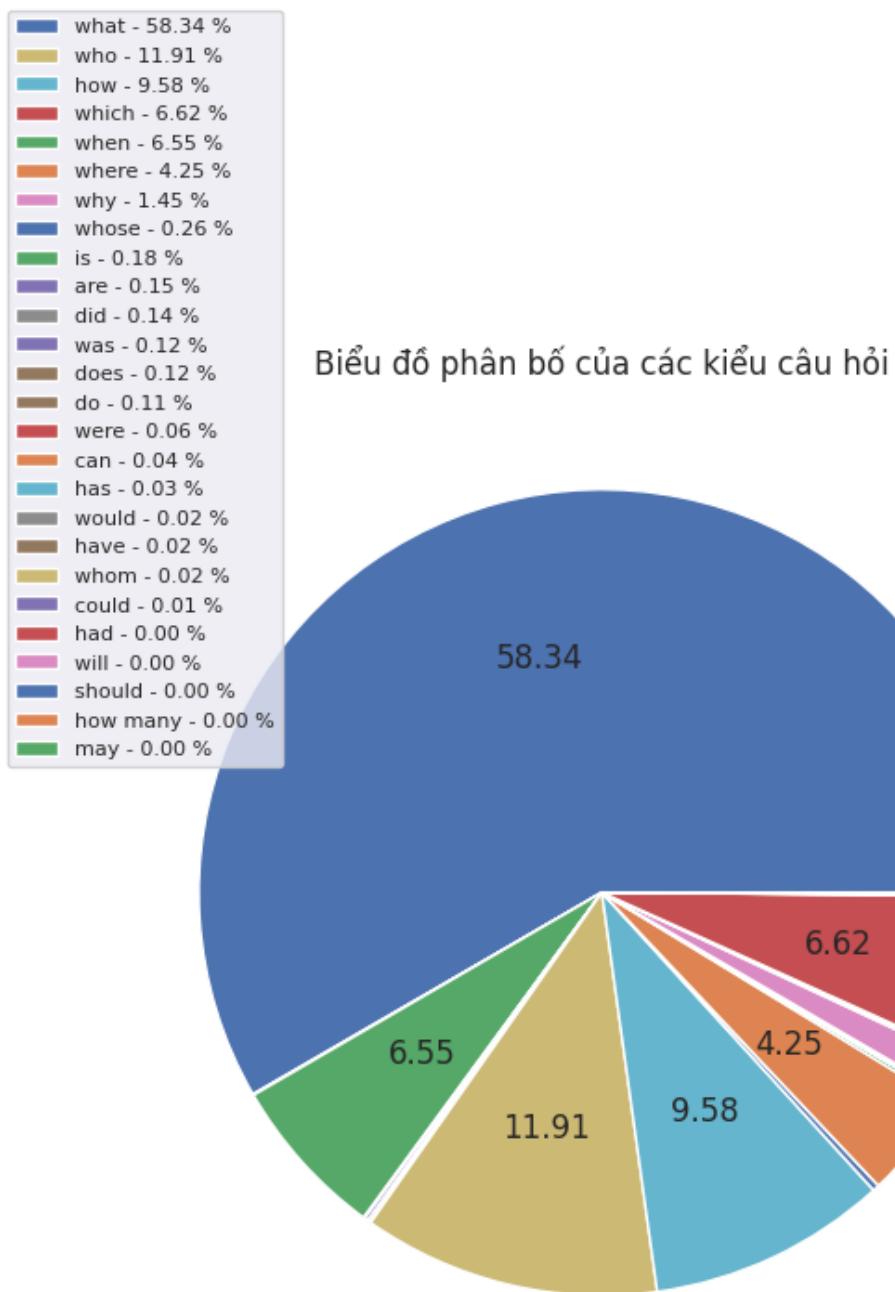
Ta có danh sách các từ để hỏi xuất hiện trong câu hỏi của bộ dữ liệu SQuAD 1.1 gồm {'what', 'can', 'when', 'were', 'are', 'does', 'will', 'would', 'who', 'how', 'whose', 'where', 'is', 'had', 'was', 'have', 'why', 'did', 'whom', 'has', 'should', 'how many', 'may', 'which', 'could', 'do'}, trong đó một số câu hỏi không xuất hiện chính xác các từ để hỏi trên nhưng có những từ khóa hay cụm từ để xác định được từ để hỏi liên quan. Ví dụ ("known as", "called") ở vị trí cuối câu, ("list", "define", "identify", ...) ở vị trí đầu câu xuất hiện trong câu hỏi sẽ có chức năng tương đương với câu hỏi "what", từ khóa "explain" ở đầu câu có chức năng như một câu hỏi "why". Dựa theo biểu đồ tròn hình (3.13) ta thấy kiểu câu hỏi đa phần là dạng câu hỏi về thông tin cụ thể (wh - factoid question) với 88%, còn lại là kiểu hỏi giải thích và kiểu đúng sai với tỷ lệ lần lượt là 11% và 1%. Cụ thể số lượng câu hỏi tương ứng với mỗi từ để hỏi được thống kê ở biểu đồ (3.14) và (3.15) với số lượng câu hỏi "what" nhiều nhất, chiếm 58.34%, sau đó là những câu hỏi dạng "wh" và hỏi về cách thức "how" chiếm hơn 40%.



Hình 3.13: Biểu đồ thống kê phần trăm số lượng câu hỏi theo mục đích hỏi trong bộ dữ liệu SQuAD 1.1.



Hình 3.15: Biểu đồ thống kê số lượng câu hỏi tương ứng với từ để hỏi trong bộ dữ liệu SQuAD 1.1.



Hình 3.14: Biểu đồ thống kê phân trăm số lượng câu hỏi tương ứng với từ để hỏi trong bộ dữ liệu SQuAD 1.1.

3.3 Phương pháp xây dựng đồ thị từ đoạn văn

Để xây dựng đồ thị từ văn bản ta có thể sử dụng các kỹ thuật như nhận diện thực thể có tên, phân giải đồng tham chiếu, cây phân tích cú pháp. Đối với đầu vào là một đoạn văn có độ dài tương đối như bộ dữ liệu SQuAD 1.1, cây phân tích cú pháp là lựa chọn tốt hơn để dựng đồ thị cho văn bản, vì nó có thể bao quát được ngữ nghĩa và liên kết về mặt cú pháp.

Để xây dựng đồ thị phụ thuộc đoạn văn bản, ta sẽ thực hiện dựng cây cú pháp cho từng câu và sau cùng kết nối các cây đó lại với nhau. Từ kết quả cây phân tích cú pháp cho câu, ta có thể dựng được đồ thị phụ thuộc tương ứng bằng cách coi những nút trong cây như nút trong đồ thị và cạnh nối giữa các nút trong đồ thị chính là thể hiện của mối quan hệ giữa các nút trong cây cú pháp. Ngoài ra, trên các cạnh của đồ thị, ta có thể thêm các nhãn tương ứng với kiểu phụ thuộc hay mối quan hệ ngữ pháp như chủ ngữ (Subject), tân ngữ (Object), hay phó từ (Adverbial).

Đồ án thực hiện nghiên cứu về cách nối các cây cú pháp phụ thuộc của từng câu trong đoạn văn.

3.3.1 Phương pháp xây dựng đồ thị dựa trên biên của câu

Đầu tiên, ta sẽ thực hiện xây dựng cây cú pháp phụ thuộc cho từng câu. Thuật toán sử dụng trong việc xây dựng là thuật toán dựa trên các chuyển đổi (transition-based), cụ thể là hệ thống cải tiến Arc-eager không tuần tự (Improved non-monotonic transition system) ([25] Honnibal và Johnson., 2014).

Hệ thống cải tiến Arc-eager không tuần tự được kết hợp từ hệ thống Arc-eager không tuần tự ([26] Honnibal et al., 2013) và hệ thống Arc-eager với ràng buộc về cây (Tree Constraint) ([27] Nivre và Fernandez-Gonzalez., 2014).

Hệ thống Arc-eager với ràng buộc về cây (Tree Constraint) ([27] Nivre và Fernandez-Gonzalez., 2014) có một số thay đổi về cấu hình thuật toán và các toán tử chuyển đổi so với hệ thống Arc-eager gốc ([28] Nivre., 2008) để giải quyết vấn đề khi bộ đệm rỗng mà ngăn xếp vẫn còn các nút không có head.

Cấu hình của hệ thống Arc-eager với ràng buộc về cây gồm bốn thành phần: (i) một bộ đệm (buffer) β chứa các từ chưa được xét trong câu, (ii) một ngăn xếp (stack) σ chứa các từ đã được xét và có thể được xét tiếp trong tương lai, (iii) một tập các cung (arc set) A chứa mối quan hệ đã tìm ra, và (iv) biến nhị phân e để kiểm tra điều kiện khi gấp cấu hình kết thúc thì bộ đệm β rỗng không? e bằng *True* nếu bộ đệm rỗng, *False* ngược lại. Cung phụ thuộc được kí hiệu (i, lb, j) với i là head, j là dependent, lb là nhãn phụ thuộc.

Các toán tử chuyển đổi áp dụng trong hệ thống Arc-eager với ràng buộc về cây

bao gồm:

1. SHIFT: loại bỏ nút đầu tiên của bộ đệm β và đẩy vào đỉnh của ngăn xếp σ , không thêm mối quan hệ nào. Toán tử chỉ được thực hiện khi biến nhị phân e có giá trị False.
2. REDUCE: loại bỏ từ ở đỉnh ngăn xếp σ với điều kiện từ ở đỉnh ngăn xếp sau đó phải có một head.
3. LEFT-ARC: tạo một cung phụ thuộc (b, lb, s) thêm vào tập A với s là nút ở đỉnh ngăn xếp σ , b là nút đầu tiên của bộ đệm β , lb là nhãn phụ thuộc bất kỳ, thêm vào đó loại bỏ từ ở đỉnh ngăn xếp và có một điều kiện là nút s đó không phải nút gốc (được chọn từ trước) và chưa có head.
4. RIGHT-ARC: tạo một cung phụ thuộc (s, lb, b) thêm vào tập A với s là nút ở đỉnh ngăn xếp σ , b là nút đầu tiên của bộ đệm β , lb là nhãn phụ thuộc bất kỳ, thêm vào đó lấy ra nút b ở bộ đệm đẩy vào đỉnh của ngăn xếp σ và có một điều kiện là nút b đó chưa có head.
5. UNSHIFT: chuyển nút ở đỉnh ngăn xếp về bộ đệm nếu bộ đệm rỗng và nút đó không có cung phụ thuộc nào trỏ đến.

Trong đó, toán tử UNSHIFT cho phép bộ phân tích cú pháp hoàn tác toán tử chuyển đổi trước đó, điều này sẽ giúp ích trong việc tự sửa lỗi của bộ phân tích. Ví dụ nếu bộ phân tích gắn sai nút với head thì toán tử UNSHIFT cho phép ta loại bỏ cung phụ thuộc đó khỏi cây cú pháp và thực hiện lại quá trình chọn toán tử chuyển đổi.

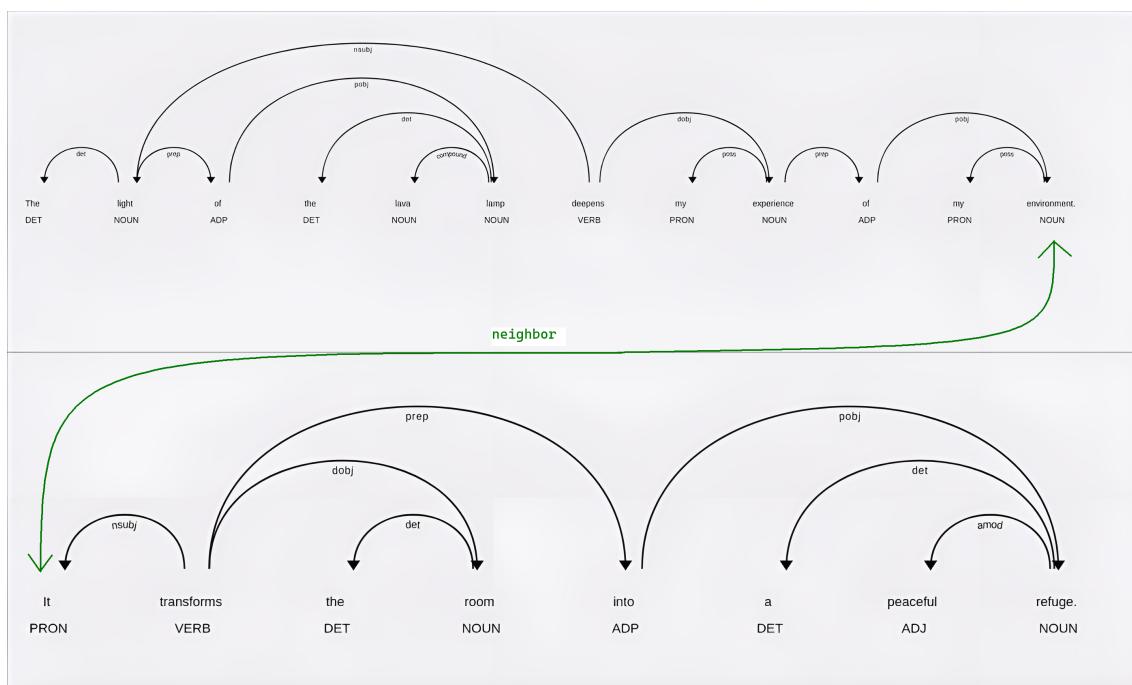
Vì toán tử UNSHIFT chỉ được sử dụng khi gặp điều kiện bộ đệm rỗng và nút ở đỉnh ngăn xếp không có cung phụ thuộc nào trỏ đến nên trong quá trình huấn luyện không cần phải có các mẫu học liên quan đến toán tử UNSHIFT. Các tác giả ([27] Nivre và Fernandez-Gonzalez, 2014) đã thực hiện huấn luyện hệ thống với chiến lược "static oracle" ([29] Goldberg and Nivre., 2012) với các cấu hình mẫu đều chính xác, chuẩn chỉnh. Điều này sẽ dẫn đến mô hình không học được cách sửa lại lỗi sai ở cây cú pháp. Vì vậy, một chiến lược huấn luyện khác được ([26] Honnibal et al., 2013) áp dụng là "dynamic oracle" với mô hình phân tích cú pháp không tuần tự (non-monotonic parsing model) với các mẫu học bao gồm cả những phép chuyển đổi đúng và sai và mô hình này có thể dự đoán ra toán tử chuyển đổi tiếp theo cho bộ phân tích cú pháp với khả năng tự sửa lỗi cho cây cú pháp tốt hơn.

Hệ thống cải tiến Arc-eager không tuần tự với toán tử UNSHIFT và huấn luyện trên mô hình phân tích cú pháp không tuần tự theo chiến lược "dynamic oracle" được áp dụng vào trong thư viện xử lý ngôn ngữ tự nhiên SpaCy. Trong đồ án này,

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

việc xây dựng cây phân tích cú pháp cho câu được thực hiện trên thư viện SpaCy.

Sau khi dựng cây phân tích cú pháp cho mỗi câu trong đoạn văn, ta thực hiện nối các cây đó qua biên của câu với nhãn phụ thuộc là "neighbor". Ví dụ, ta có hai câu liên tiếp nhau $X = \{x_1, x_2, \dots, x_n\}$ và $Y = \{y_1, y_2, \dots, y_n\}$, ta lấy biên cuối của câu X là từ cuối cùng của câu x_n để tạo cạnh với biên đầu của câu Y là từ đầu tiên của câu y_1 . Với nhãn là "neighbor" ta có hai cạnh mới là $(x_n, \text{"neighbor"}, y_1)$ và $(y_1, \text{"neighbor"}, x_n)$. Thực hiện nối lần lượt các biên của các cặp câu liên tiếp còn lại ta thu được đồ thị cho đoạn văn.



Hình 3.16: Ví dụ về kết nối các cây phân tích cú pháp qua biên của câu. Đoạn văn "The light of the lava lamp deepens my experience of my **environment**. **It** transforms the room into a peaceful refuge." gồm hai câu, trong đó biên cuối của câu thứ nhất là "environment", biên đầu của câu thứ hai là "It".

3.3.2 Phương pháp xây dựng đồ thị phụ thuộc dựa trên HEAD của câu

Với một câu đầu vào, qua bộ phân tích cú pháp ở phần 3.3.1 ta có được cây phân tích cú pháp tương ứng. Trong cây phân tích cú pháp phụ thuộc, mỗi từ trong câu được kết nối với head của nó thông qua một cung phụ thuộc có hướng, thể hiện mối quan hệ ngữ pháp giữa chúng. Do đó, HEAD của một câu thường là một từ không có cung phụ thuộc trỏ đến và đóng vai trò là điểm "neo" cho toàn bộ cấu trúc của câu.

Với câu đầu vào X gồm n tokens, $X = \{x_1, x_2, \dots, x_n\}$, tập các cung $A = \{(x_i, label, x_j) | x_i, x_j \in X\}$. Với một cung $arc = (x_i, label, x_j)$, để lấy ra head và dependent của cung ta lần lượt sử dụng $arc.head$ và $arc.dependent$.

Gọi $S_{head} = \{arc.head | arc \in A\}$ là tập hợp các nút đã xuất hiện trong tập A với vị trí là head, $S_{dependent} = \{arc.dependent | arc \in A\}$ là tập hợp các nút xuất hiện trong tập A với vị trí là dependent. Khi đó HEAD của câu là nút xuất hiện ở tập S_{head} mà không xuất hiện ở tập $S_{dependent}$.

$$HEAD \in S_{head} - S_{dependent} \quad (3.1)$$

Đối với một đoạn văn cho trước gồm nhiều câu, sau khi lấy được cây phân tích cú pháp của từng câu, ta kết nối các cây phân tích cú pháp qua HEAD của mỗi câu - HEAD của câu trước nối với HEAD của câu sau và ngược lại với nhãn quan hệ là hàng xóm "neighbor" (Hình 3.17). Việc sử dụng HEAD làm điểm kết nối giữa các câu sẽ có ưu điểm so với cách sử dụng biên của câu làm điểm kết nối khi thực hiện lan truyền tính toán nhúng nút đồ thị trong mạng bi-GGNN. Từ công thức (2.9) đến công thức (2.12) là công thức tổng hợp thông tin các nút hàng xóm theo hai hướng vào, ra của cung phụ thuộc. Do HEAD chứa thông tin cấu trúc ngữ pháp của câu nên khi kết nối các câu qua HEAD sẽ giúp liên kết thông tin cú pháp tại các câu với nhau tốt hơn.



Hình 3.17: Ví dụ về kết nối các cây phân tích cú pháp qua HEAD của câu. Đoạn văn "The light of the lava lamp **deepens** my experience of my environment. It **transforms** the room into a peaceful refuge." gồm hai câu, trong đó HEAD của mỗi câu được chọn là các đỉnh không có cung phụ thuộc nào trỏ đến, ở đây là hai nút "deepens" và "transforms".

3.3.3 Phương pháp nối các cây cú pháp phụ thuộc sử dụng phân giải đồng tham chiếu

Đầu tiên, ta cũng thực hiện xây dựng các cây phân tích cú pháp phụ thuộc cho từng câu sử dụng thuật toán dựa trên sự chuyển đổi. Nhận thấy trong một câu văn dài, các thực thể, đối tượng hoặc sự kiện thường được đề cập đến bằng nhiều cách khác nhau, sử dụng các từ hoặc cụm từ thay thế. Điều này có thể gây hiểu nhầm về việc đề cập đến cùng một thực thể. Vì vậy, ta sử dụng phân giải đồng tham chiếu để giúp kết hợp thông tin từ các mô tả khác nhau của cùng một thực thể, tạo ra một đồ thị toàn diện hơn về thực thể đó và giúp nắm bắt thông tin tổng quan của đoạn văn.

Về giải pháp, ta sử dụng mô hình huấn luyện trước SpanBERT ([30] Joshi et al., 2020) và sử dụng thư viện AllenNLP ([31] Gardner et al., 2017) để cài đặt tác vụ phân giải đồng tham chiếu. SpanBERT có ba sự thay đổi so với mô hình BERT truyền thống:

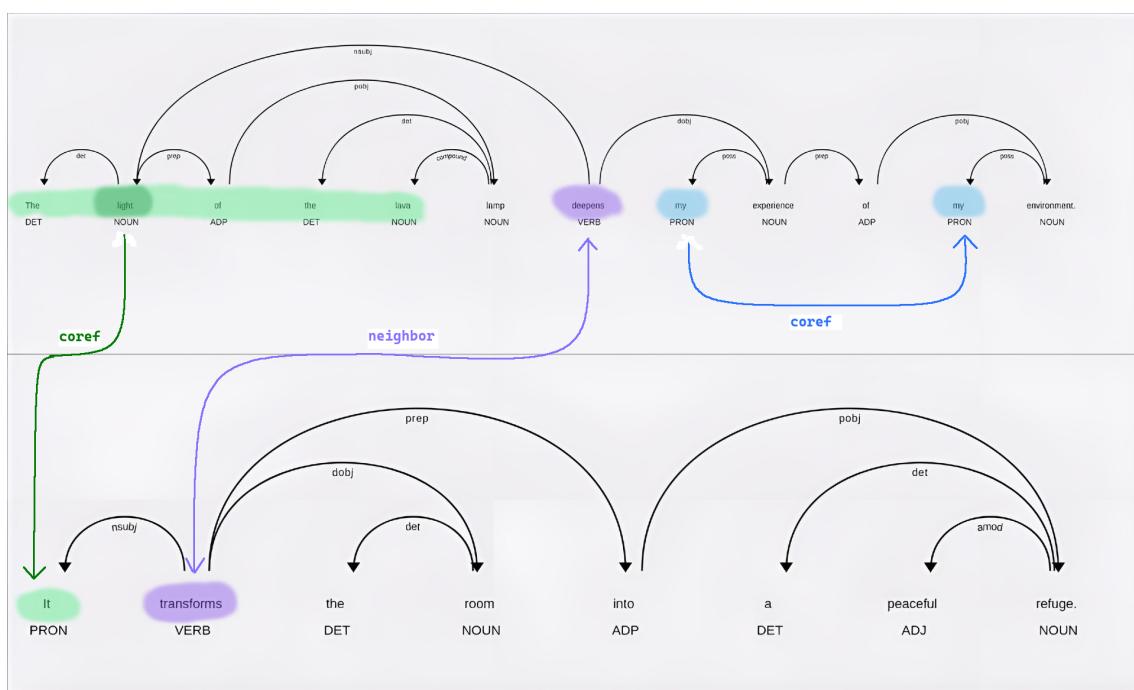
1. Kỹ thuật che dấu spans (Span Masking): ta sẽ đi tìm tập con của tập các tokens của câu đầu vào bằng cách lặp đi lặp lại quá trình lấy mẫu cho đến khi đạt ngưỡng số lượng nhãn [MASK]. Trong mỗi vòng lặp thực hiện lấy mẫu các spans với độ dài xác định theo phân phối hình học, khi đó, các spans có độ dài ngắn sẽ được lựa chọn nhiều hơn. Vị trí bắt đầu của spans được lựa chọn ngẫu nhiên. Việc che dấu bằng nhãn [MASK] thực hiện ở mức span, tức tất cả các tokens trong span đó đều thay thế bằng nhãn [MASK].
2. Mục tiêu biên Span (Span Boundary Objective): mô hình cố gắng dự đoán span bị che dấu bằng cách sử dụng chỉ các biểu diễn của các token ở ranh giới span. Mô hình tính toán ra biểu diễn của các token trong span bằng cách kết hợp các thông tin về vị trí bắt đầu span, vị trí kết thúc span, vị trí của token trong span. Điều này giúp cải thiện khả năng của mô hình trong việc biểu diễn và hiểu các mối quan hệ phụ thuộc dài trong văn bản.
3. Huấn luyện trên từng chuỗi (Single-Sequence Training): thay vì sử dụng mẫu học gồm hai chuỗi và hàm mục tiêu dự đoán hai chuỗi đó có liên tiếp nhau không như ở BERT, SpanBERT sử dụng mẫu học chỉ gồm một chuỗi duy nhất. Lý do được đưa ra là để mô hình có thể tập trung vào việc biểu diễn các thực thể có trong spans, hơn nữa, việc sử dụng mẫu học gồm hai chuỗi có thể gây ra nhiễu trong biểu diễn spans vì có thể hai chuỗi đó không liên quan gì đến nhau.

Sau đó, với mỗi đề cập (mention), ta tính toán phân phối trên những spans đã được diễn giải trước đó để tìm ra thực thể có khả năng nhất tham chiếu đến.

Sau khi dựng câu phân tích cú pháp cho từng câu và phân giải đồng tham chiếu cho đoạn văn, ta thấy việc thay thế những đề cập (mentions) với span tham chiếu sẽ giúp liên kết được các thông tin ở xa trong đoạn văn với thực thể tương ứng. Từ đó, khi thực hiện tính toán nhúng nút trong đồ thị, ta có thể biểu diễn đầy đủ thông tin về thực thể hơn.

Để thực hiện kết nối các cây phân tích cú pháp của từng câu, ta sẽ kết nối qua các đồng tham chiếu đã tìm ra trong đoạn văn và sử dụng thêm HEAD của câu (công thức 3.1) như sau:

1. Với mỗi cặp đồng tham chiếu, ta sẽ nối các đề cập (mention) đến span tham chiếu gốc với nhãn là "coref". Trong trường hợp các đề cập (mentions) hoặc span tham chiếu gốc gồm nhiều tokens, ta sẽ lấy HEAD của cụm từ đó để tạo cạnh phụ thuộc (phương pháp tìm ra HEAD như ở phần 3.3.2).
2. Kết nối HEAD của các câu lại với nhau theo dạng câu trước nối với câu sau, vì có thể có trường hợp giữa hai câu không có đồng tham chiếu nào đến nhau.



Hình 3.18: Ví dụ về kết nối các cây phân tích cú pháp sử dụng phân giải đồng tham chiếu và HEAD của câu.

Ví dụ ở hình 3.18 ta có đoạn văn gồm hai câu "The light of the lava lamp deepens my experience of my environment. It transforms the room into a peaceful refuge.". Thực hiện phân giải đồng tham chiếu, ta có hai cặp đồng tham chiếu { "my" (từ ở vị trí thứ 7), "my" (từ ở vị trí thứ 10)}, { "The light of the lava lamp", "It" }. Với cặp đồng tham chiếu { "my", "my" }, ta thêm hai cạnh với nhãn là "coref" là ("my" (7th token), "coref", "my" (10th token)) và ("my" (10th token), "coref", "my" (7th

token)). Với cặp đồng tham chiếu {"The light of the lava lamp", "It"}, vì "The light of the lava lamp" gồm nhiều tokens ghép lại nên ta sẽ lấy HEAD của span tham chiếu đó là "light" để tạo cạnh nối với "It", khi đó ta thêm được hai cạnh là ("light", "coref", "It"), ("It", "coref", "light"). Ngoài ra ta vẫn sử dụng HEAD của hai câu để tạo cạnh nối, ta có thêm hai cạnh nữa là ("transforms", "neighbor", "deepens"), ("deepens", "neighbor", "transforms").

3.4 Kết chương

Trong chương 3, em đã trình bày về việc phân tích và lý do sử dụng bộ dữ liệu SQuAD phiên bản 1.1 và ba phương pháp xây dựng đồ thị từ đoạn văn bản: phương pháp sử dụng biên của câu, phương pháp sử dụng HEAD của câu và phương pháp sử dụng phân giải đồng tham chiếu. Trong chương 4, em sẽ trình bày về các kết quả thực nghiệm với ba phương pháp kể trên.

CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM

4.1 Các tham số đánh giá

Trong các tác vụ xử lý ngôn ngữ tự nhiên có liên quan đến quá trình sinh từ, mặc dù đánh giá của con người đáng tin cậy hơn để suy ra chất lượng của một văn bản sinh ra về mặt cấu trúc ngữ pháp cũng như ngữ nghĩa, nhưng các tham số đánh giá tự động lại có ưu điểm là dễ tính toán và có mức độ khả dụng cao hơn. Các tham số đánh giá được sử dụng phổ biến hầu hết đều dựa trên sự trùng lặp từ và ban đầu được dùng để đánh giá hiệu suất của các hệ thống dịch máy.

Trong đồ án này, các tham số đánh giá được sử dụng là Bi-Lingual Evaluation Understudy (BLEU) ([32] Papineni et al., 2002) và Recall-Oriented Understudy for Gisting Evaluation (ROUGE) ([33] Lin, 2004).

4.1.1 BLEU

Tham số đánh giá BLEU ([32] Papineni et al., 2002) được phát triển dựa trên cơ sở "câu được sinh ra càng giống câu mà con người đưa ra" thì càng tốt. Để đo được sự tương đồng giữa câu mà con người đưa ra và câu do mô hình sinh, ta sẽ so sánh dựa trên n-gram. Cụ thể, ta sẽ tìm n-gram của hai câu cần so sánh, sau đó đếm số lượng các cặp trùng nhau và bỏ qua thông tin về vị trí của các cặp đó trong câu. Số lượng các cặp trùng nhau càng lớn thì càng tốt. Để tính giá trị độ đo BLEU, ta cần tính độ chính xác của câu qua n-gram. n-gram là tần suất xuất hiện của n token liên tiếp nhau trong dữ liệu, còn độ chính xác đo số lượng token ở câu được mô hình sinh ra trùng với token trong câu đích. Xét trường hợp *unigram*, ta có công thức tính độ chính xác *Precision* là:

$$Precision(unigram) = \frac{Count(matches)}{Count(target)}$$

với *Count(matches)* là số lượng các token trùng nhau giữa câu đích và câu do mô hình sinh ra, *Count(target)* là số lượng token trong câu đích.

Nhưng khi dùng độ đo precision sẽ gặp tình huống câu bị lặp một từ duy nhất và từ đó cũng xuất hiện trong câu đích (Repetition), ví dụ câu đích là "*default or booktabs table style?*", câu mô hình sinh ra là "*default default default default?*" thì độ chính xác với trường hợp unigram tính được là 1. Để giải quyết vấn đề này, ([32] Papineni et al., 2002) đã đưa ra phương án là giới hạn số lượng token giống nhau ở câu sinh ra tham chiếu đến token trong câu đích, ở ví dụ trên, số lượng từ trong câu đích là 4, số lượng cặp token trùng nhau thay vì là 4 là bị cắt chỉ còn 1 (token "*default*" chỉ xuất hiện 1 lần trong câu đích), nên độ chính xác sau khi bị cắt (modified unigram precision) là 0.25. Ta có công thức tổng quát khi tính BLEU với

n-gram cho hai câu Q và Q' :

$$p_i = Precision_{clip}(n - gram) = \frac{\sum_{\substack{n_gram \in Q \\ n_gram' \in Q'}} Count_{clip}(n_gram)}{\sum_{n_gram' \in Q'} Count(n_gram')} \quad (4.1)$$

với $Count_{clip}(n_gram)$ là số lượng cặp n-gram trùng nhau sau khi đã sử dụng phương pháp cắt, $Count_{clip} = \min(Count, < Số n-gram tham chiếu đến tối đa >)$

Tiếp đó, ta sẽ kết hợp các độ chính xác của unigram, bigram, ..., n-gram lại để tính ra độ đo chính xác trung bình hình học (Geometric Average Precision Scores):

$$GeometricAveragePrecision(n - gram) = \exp\left(\sum_{i=1}^n w_i * \log(p_i)\right) = \prod_{i=1}^n p_i^{w_i} \quad (4.2)$$

với w_i là trọng số chuẩn hóa được chọn bằng $1/N$.

Để tránh trường hợp câu do mô hình sinh ra quá ngắn nhưng độ chính xác theo tính toán vẫn cao, tác giả ([32] Papineni et al., 2002) đề xuất một hệ số phạt (Brevity Penalty) để "phạt" khi mô hình sinh ra câu ngắn nhưng điểm cao, nếu câu sinh ra càng ngắn, giá trị phạt sẽ càng nhỏ.

$$BrevityPenalty = \begin{cases} 1 & c > r \\ e^{1-\frac{r}{c}} & c \leqslant r \end{cases} \quad (4.3)$$

với r là số lượng từ ở câu do mô hình sinh ra, c là số lượng từ ở câu đích.

Tổng hợp giá trị độ chính xác với n-gram và hệ số phạt (Brevity Penalty), ta có độ đo BLEU được tính như sau:

$$BLEU(n - gram) = GeometricAveragePrecision(n - gram) * BrevityPenalty \quad (4.4)$$

4.1.2 ROUGE-L

ROUGE-L ([33] Lin, 2004) được phát triển dựa trên bài toán tìm dãy con chung dài nhất (LCS - Longest common subsequence) của câu đích và câu do mô hình sinh ra (các dãy con chung không nhất thiết phải là các từ liên tục, nhưng vẫn phải theo thứ tự trước sau). Để tính ra được giá trị ROUGE-L ta tính các giá trị "precision", "recall" và điểm F1 dựa trên dãy con chung dài nhất (LCS) giữa câu đích Q và câu được sinh ra Q' .

$$Recall_{lcs} = \frac{LCS(Q, Q')}{Q_{length}} \quad (4.5)$$

$$Precision_{lcs} = \frac{LCS(Q, Q')}{Q'_{length}} \quad (4.6)$$

$$F1_{LCS} = \frac{(1 + \beta^2) Recall_{lcs} Precision_{lcs}}{Recall_{lcs} + \beta^2 Precision_{lcs}} \quad (4.7)$$

trong đó β là hệ số điều hòa, cho biết độ ưu tiên "Precision" hơn hay "Recall" hơn. Nếu $\beta > 1$, giá trị "Recall" quan trọng hơn vì nó thể hiện số lượng dây con ở câu đích xuất hiện ở câu do mô hình sinh ra. Trong đồ án này, giá trị β được chọn là 1,2.

4.2 Phương pháp thí nghiệm

4.2.1 Dữ liệu

Dữ liệu sử dụng trong thực nghiệm là bộ dữ liệu SQuAD 1.1 ([14] Rajpurkar et al., 2016). Dữ liệu được lấy nguồn từ 536 bài viết trên Wikipedia và có hơn 100 nghìn cặp câu hỏi, câu trả lời và được chia thành 3 tập "train", "dev", "test" theo tỉ lệ 8:1:1.

Dữ liệu ban đầu gồm đoạn văn, câu hỏi, câu trả lời và vị trí bắt đầu và kết thúc của câu trả lời trong đoạn văn. Để thực hiện tiền xử lý cho bộ dữ liệu SQuAD phiên bản 1.1 trên, ta sử dụng thư viện Stanza ([34] Qi et al., 2020) để thực hiện trích xuất thông tin về thực thể có tên (Name entity recognition), từ loại trong câu (Part-of-speech) và sử dụng thư viện SpaCy để thực hiện tách tokens. Riêng đối với phương pháp xây dựng đồ thị từ đoạn văn sử dụng phân giải đồng tham chiếu, ta sử dụng thư viện AllenNLP ([31] Gardner et al., 2017) để tách tokens và thực hiện phân giải đồng tham chiếu. Sau đó, em sử dụng thư viện SpaCy để xây dựng cây phân tích cú pháp phụ thuộc cho từng câu.

4.2.2 Phương pháp baseline

Phần thực nghiệm trong đồ án sẽ so sánh kết quả các phương pháp xây dựng đồ thị từ đoạn văn bản khi kết hợp với mô hình Graph2Seq với bộ giải mã IGND ([8] Fei et al., 2021). Phương pháp baseline được sử dụng là phương pháp sử dụng cây phân tích cú pháp và nối các cây đó tại nút tương ứng với biên của câu, đây chính là phương pháp xây dựng đồ thị từ đoạn văn trong nghiên cứu về IGND của tác giả Fei et al., 2021.

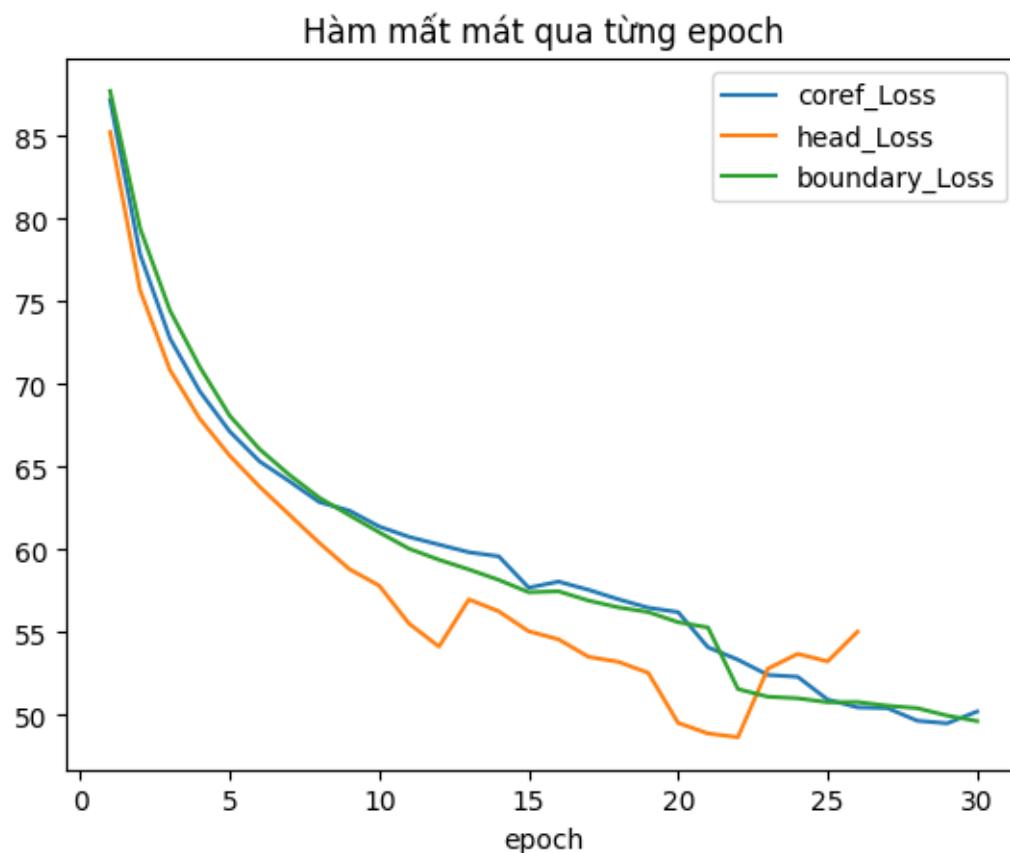
4.2.3 Tham số huấn luyện

Trong bộ mã hóa, trong phần xử lý làm giàu thông tin đầu vào, ta sử dụng cấu hình nhúng GloVe 300 chiều được huấn luyện trước, nhúng BERT đã huấn luyện

sẵn 1024 chiều, và nhúng của các thông tin "word case", nhãn BIO, POS, NER lần lượt là 3, 3, 12 và 8 chiều. Số chiều của vector ẩn trong mạng bi-LSTM sau khi đã nối hai vector trạng thái ẩn theo chiều xuôi và ngược là 300 và kích thước các trạng thái ẩn trong tất cả các lớp ẩn khác cũng là 300. Ngoài ra, tại mỗi lớp nhúng từ ta áp dụng "dropout" tỷ lệ 0.4, còn những lớp mạng LSTM ở cả bộ mã hóa và giải mã ta áp dụng "dropout" tỷ lệ 0.3. Số lượng vòng lặp tính toán trong mạng bi-GGNN (graph hop) là 4. Phương pháp tối ưu sử dụng Adam với tốc độ học (learning rate) là 0.0001. Trong đó, nếu giá trị BLEU-4 không tăng trong 3 epochs sẽ thực hiện chia đôi tốc độ học và sẽ thực hiện dừng huấn luyện sớm nếu BLEU-4 không tăng trong 5 vòng lặp. Số epoch huấn luyện là 30. Kích thước lô dữ liệu (batch size) là 30. Số lượng ứng viên lựa chọn trong thuật toán tìm kiếm chùm (beam search) là 5.

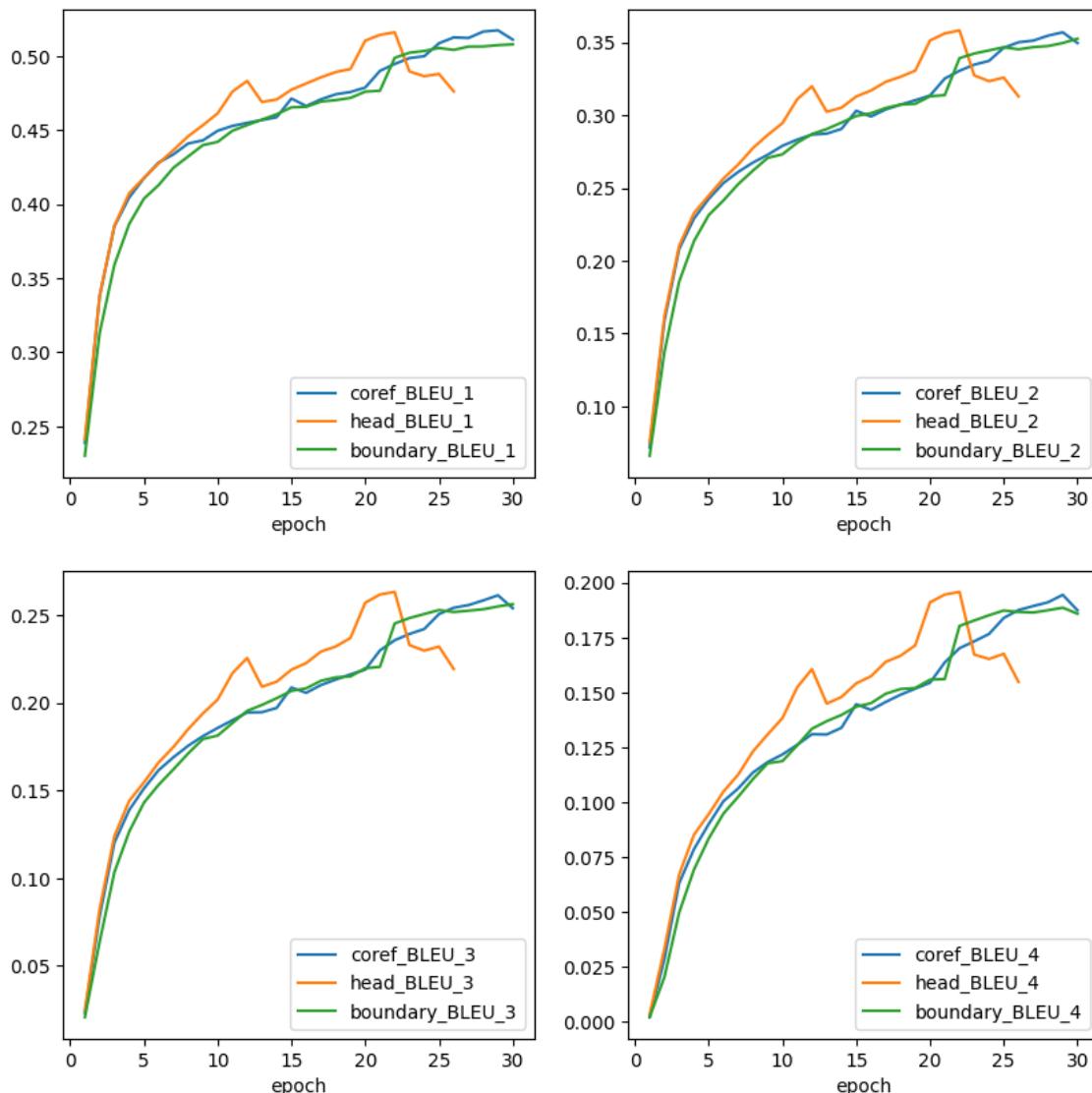
4.3 Kết quả thực nghiệm

Thực hiện huấn luyện mô hình IGND với ba phương pháp xây dựng đồ thị từ văn bản: (i) phương pháp baseline là phương pháp xây dựng đồ thị dựa trên biên của câu, hai phương pháp nghiên cứu thêm là (ii) phương pháp xây dựng đồ thị dựa trên HEAD của câu và (iii) phương pháp xây dựng đồ thị sử dụng phân giải đồng tham chiều. Dưới đây là các biểu đồ về giá trị hàm mất mát và hai độ đo BLEU và ROUGE-L trong quá trình huấn luyện.

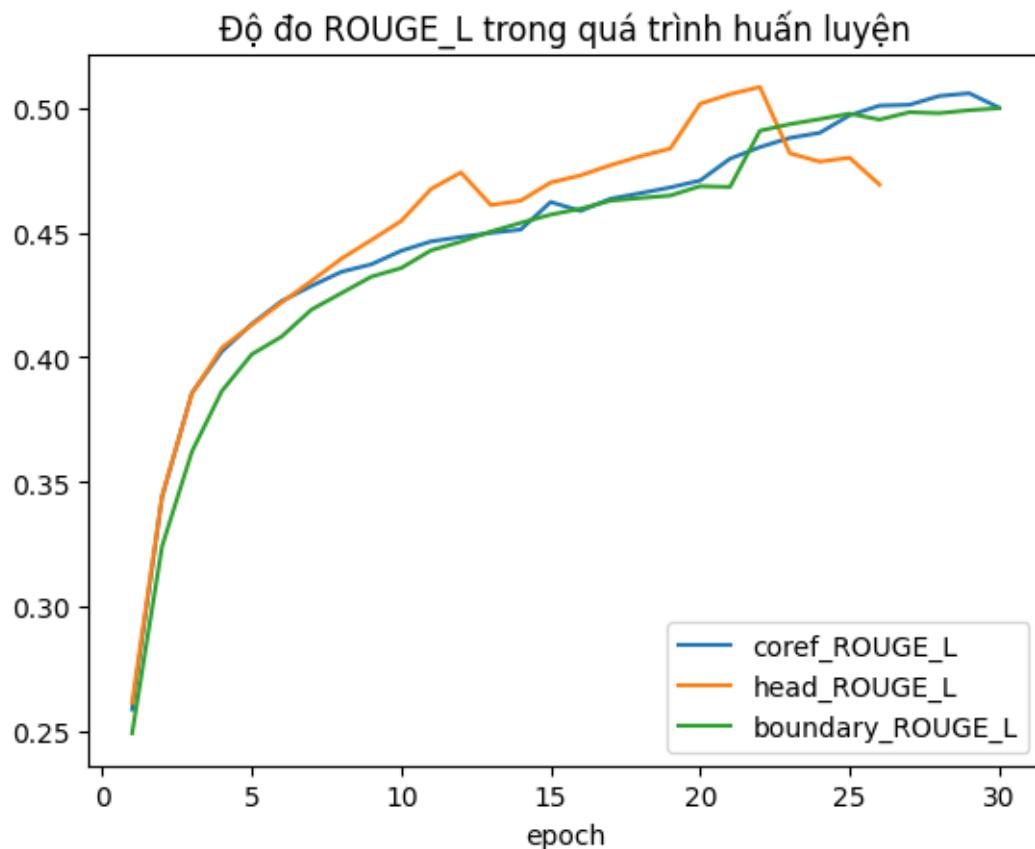


Hình 4.1: Biểu đồ giá trị của hàm mất mát qua từng epoch.

Độ đo BLEU trong quá trình huấn luyện



Hình 4.2: Biểu đồ giá trị độ đo BLEU-1, BLEU-2, BLEU-3, BLEU-4 qua từng epoch.

**Hình 4.3:** Biểu đồ giá trị độ đo ROUGE-L F1 qua từng epoch.

Chú thích: tiền tố trong các chú thích trong các hình (4.1), (4.2), (4.3) tương ứng với các phương pháp: (i) "coref_": phương pháp xây dựng đồ thị sử dụng phân giải đồng tham chiếu, (ii) "head_": phương pháp xây dựng đồ thị dựa trên HEAD của câu và (iii) "boundary_": phương pháp xây dựng đồ thị dựa trên biên của câu.

Phương pháp	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
Bound	0.4642	0.3050	0.2203	0.1635	0.4549
Bound w/o BERT	0.4617	0.3008	0.2167	0.1619	0.4475
HEAD	0.4804	0.3230	0.2378	0.1806	0.4666
HEAD w/o BERT	0.4673	0.3058	0.2208	0.1655	0.4500
coref	0.4888	0.3310	0.2439	0.1850	0.4756
coref w/o BERT	0.4591	0.3001	0.2158	0.1612	0.4512

Bảng 4.1: Kết quả độ đo BLEU và ROUGE-L thực hiện trên tập test.

Phương pháp	Fluency	Relevancy	Answerability
Bound	2.21	1.98	1.92
HEAD	2.22	2.01	1.88
coref	2.40	2.21	2.01

Bảng 4.2: Kết quả "Human Evaluation" thực hiện trên 100 mẫu ngẫu nhiên.

Kết quả thực nghiệm trên tập test của mô hình Graph2Seq với ba phương pháp xây dựng đồ thị được thể hiện trong bảng (4.1). Từ kết quả cho ta thấy phương pháp sử dụng phân giải đồng tham chiếu có kết quả tốt nhất với giá trị BLEU-4 đạt được 0.1850, hơn 0.02 so với phương pháp sử dụng biên của câu. Phương pháp sử dụng HEAD của câu có kết quả khá gần với phương pháp sử dụng phân giải đồng tham chiếu với kết quả các độ đo BLEU, ROUGE-L chỉ kém hơn khoảng 0.01.

Đánh giá của con người rất quan trọng trong việc đánh giá chất lượng của các câu hỏi sinh ra vì mô hình có thể tạo ra các câu hỏi khá hợp lý nhưng không thể so sánh tốt với các câu hỏi có cơ sở được cung cấp sẵn. Em chọn ngẫu nhiên 20 đoạn văn bản và lấy được tất cả 100 câu hỏi do mỗi phương pháp sinh ra. Em đã nhờ được người thân ở nước ngoài biết tiếng Anh để đánh giá theo ba tiêu chí "Fluency", "Relevancy" và "Answerability" với thang điểm từ 1 đến 3, 3 là điểm tốt nhất.

Fluency: Sự lưu loát, đánh giá xem một câu hỏi có đúng ngữ pháp và lưu loát hay không.

Relevancy: Mức độ liên quan, đánh giá xem câu hỏi có liên quan đến đoạn văn đầu vào hay không.

Answerability: Khả năng trả lời, cho biết liệu câu hỏi có thể được trả lời dựa trên đoạn văn đã cho hay không.

Ví dụ 1

Đoạn văn: France took control of Algeria in 1830 but began in earnest to rebuild its worldwide empire after 1850, concentrating chiefly in *North and West Africa*, as well as South-East Asia, with other conquests in Central and East Africa, as well as the South Pacific. . . .

Câu trả lời: North and West Africa.

Câu hỏi mẫu: Where did France focus its efforts to rebuild its empire?

Phương pháp sử dụng biên của câu: where did france take place ?

Phương pháp sử dụng HEAD của câu: along with north and west , what other country did france join in 1850 ?

Phương pháp sử dụng phân giải đồng tham chiếu: which country did france rebuild its empire ?

Ví dụ 2

Đoạn văn: . . . Presented to the Scottish Parliament by the Queen upon its official opening in July 1999, the mace is displayed in *a glass case suspended from the lid*. At the beginning of each sitting in the chamber, the lid of the case is rotated so that the mace is above the glass, to symbolise that a full meeting of the Parliament is taking place.

Câu trả lời: a glass case suspended from the lid

Câu hỏi mẫu: What is the mace displayed in?

Phương pháp sử dụng biên của câu: how is the mace of the scottish displayed in 1999 ?

Phương pháp sử dụng HEAD của câu: where is the mace displayed in july 1999?

Phương pháp sử dụng phân giải đồng tham chiếu: where is the mace displayed?

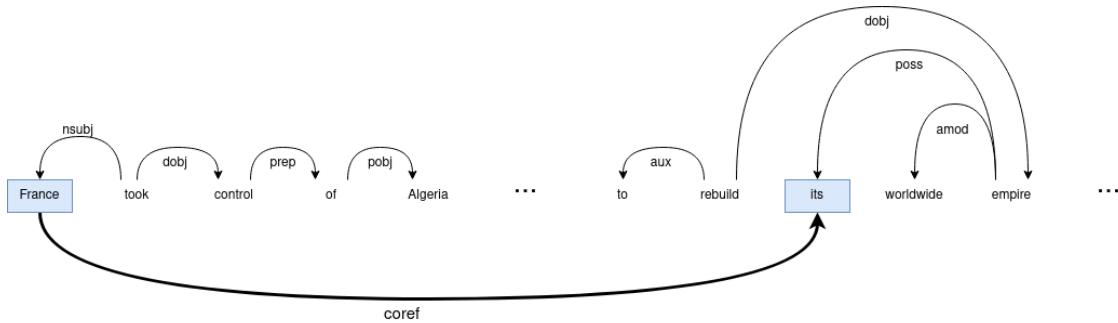
Bảng 4.3: Một số ví dụ về kết quả câu hỏi sinh ra của mô hình Graph2Seq với ba phương pháp xây dựng đồ thị từ văn bản.

Một số ví dụ về câu hỏi sinh ra khi sử dụng các phương pháp xây dựng đồ thị từ văn bản khác nhau được thể hiện ở bảng (4.3).

Trong ví dụ đầu tiên, ta thấy rằng phương pháp xây dựng đồ thị sử dụng phân giải đồng tham chiếu đã liên kết các thông tin đến thực thể một cách đầy đủ hơn so với hai phương pháp còn lại. Trong hình (4.4) là đồ thị với cạnh nối các đồng tham chiếu đã giúp có thêm thông tin liên quan đến thực thể "France". Vì vậy câu hỏi sinh ra có ngữ nghĩa giống với câu hỏi mẫu (cùng hỏi về địa điểm mà nước Pháp tái thiết đế chế), trong khi đó câu hỏi do phương pháp sử dụng biên của câu không liên quan gì đến câu trả lời, còn phương pháp sử dụng HEAD của câu sinh ra câu hỏi có phần không đúng với câu trả lời, khi hỏi đến quốc gia khác mà không phải là "North and West Africa".

Ở trong ví dụ thứ 2, ta thấy câu hỏi sinh ra với phương pháp sử dụng phân giải đồng tham chiếu khá sát với câu hỏi mẫu với giá trị BLEU-4 là 0.45, tuy vậy, hai phương pháp còn lại lại sinh ra câu hỏi với nhiều thông tin hơn. Phương pháp sử

dụng HEAD của câu sinh ra câu hỏi có thêm thông tin về thời gian "july 1999", phương pháp sử dụng biên của câu ngoài thông tin về thời gian còn có thông tin về tổ chức "Scottish" nhưng về mặt ngữ nghĩa lại khá "tối" khi câu hỏi hỏi về cách thức, không phải vị trí cụ thể.



Hình 4.4: Trong ví dụ đầu tiên, từ "its" tham chiếu đến từ "France", từ đó câu hỏi liên quan đến "France" sẽ có được thông tin quanh từ "its" là "worldwide empire", "rebuild".

4.4 Kết chương

Trong chương 4, em đã trình bày về phần tiền xử lý dữ liệu SQuAD phiên bản 1.1, các tham số huấn luyện, tham số đánh giá và kết quả thực nghiệm ba phương pháp xây dựng đồ thị từ văn bản kết hợp mô hình học sâu Graph2Seq với bộ giải mã IGND. Do em còn hạn chế về các kỹ năng nghiên cứu nên phần trình bày các kết quả thực nghiệm có thể chưa tốt, các đóng góp cũng chưa thực sự nổi bật nhưng trong quá trình làm đồ án em đã cải thiện thêm được nhiều kĩ năng. Trong chương tiếp theo em xin tổng hợp lại những gì đã làm được và hướng triển cho hệ thống trong tương lai.

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

Trong đồ án nghiên cứu, em đã tìm hiểu về bài toán sinh câu hỏi và mô hình học sâu IGND để giải quyết bài toán, sau đó thực hiện thử nghiệm với ba phương pháp xây dựng đồ thị từ văn bản gồm: (i) phương pháp xây dựng đồ thị dựa trên biên của câu; (ii) phương pháp xây dựng đồ thị dựa trên HEAD của câu và (iii) phương pháp xây dựng đồ thị sử dụng phân giải đồng tham chiếu với mô hình IGND trên bộ dữ liệu SQuAD phiên bản 1.1 và qua kết quả thực nghiệm đã cho thấy việc xây dựng đồ thị cho đoạn văn bản có ảnh hưởng đến chất lượng câu hỏi sinh ra.

Trong quá trình thực hiện đồ án nghiên cứu này, em đã tích lũy thêm được các kỹ năng chuyên môn như khai thác mã nguồn mở, kinh nghiệm khi tiền xử lý với khối lượng dữ liệu lớn khi không đáp ứng được tài nguyên phần cứng, tăng cường kỹ năng tìm và đọc hiểu các tài liệu, bài báo khoa học. Với sự hướng dẫn tận tình của PGS.TS. Nguyễn Thị Kim Anh đã giúp em không những cải thiện về mặt kỹ thuật chuyên môn như cách tiếp cận bài toán, các phương pháp xử lý bài toán mà các kỹ năng như giao tiếp, làm việc nhóm và trao đổi thông tin một cách hiệu quả. Đó là những kỹ năng rất quan trọng và chắc chắn sẽ giúp ích rất nhiều cho công việc trong tương lai.

Tuy nhiên, do vấn đề liên quan đến tài nguyên phần cứng và thời gian có hạn nên việc huấn luyện dừng lại ở epoch 30, cũng như số lượng dữ liệu huấn luyện bị giới hạn lại và việc huấn luyện với dữ liệu gốc SQuAD rất chậm, mất nhiều thời gian vì kích thước đoạn văn bản lớn.

5.2 Hướng phát triển trong tương lai

Với những ứng dụng hữu ích của bài toán sinh câu hỏi, trong tương lai có thể phát triển hơn nữa. Phương pháp xây dựng đồ thị có thể phát triển nghiên cứu thêm về việc cắt tỉa đồ thị và thêm các cạnh hướng đến câu trả lời nhằm giảm tải tính toán và giúp tập trung thông tin tốt hơn vào câu trả lời đó. Ngoài ra còn những vấn đề cần giải quyết liên quan đến dữ liệu khi xây dựng đồ thị như sự mơ hồ về ngữ nghĩa, đoạn văn gấp những vấn đề liên quan đến cấu trúc ngữ pháp, ...

Ngoài bộ dữ liệu tiếng Anh, mô hình sử dụng trong đồ án hoàn toàn có thể ứng dụng sang tiếng Việt và phần tiền xử lý, xây dựng đồ thị sẽ phải xử lý theo hướng phù hợp với cấu trúc ngữ pháp tiếng Việt hơn.

TÀI LIỆU THAM KHẢO

- [1] H. Elsahar, C. Gravier **and** F. Laforest, “Zero-shot question generation from knowledge graphs for unseen predicates and entity types,” *inProceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* New Orleans, Louisiana: Association for Computational Linguistics, **june** 2018, **pages** 218–228. DOI: 10.18653/v1/N18-1020. **url:** <https://aclanthology.org/N18-1020>.
- [2] S. Reddy, D. Raghu, M. M. Khapra **and** S. Joshi, “Generating natural language question-answer pairs from a knowledge graph using a RNN based question generation model,” *inProceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* Valencia, Spain: Association for Computational Linguistics, **april** 2017, **pages** 376–385. **url:** <https://aclanthology.org/E17-1036>.
- [3] V. Kumar, Y. Hua, G. Ramakrishnan, G. Qi, L. Gao **and** Y.-F. Li, “Difficulty-controllable multi-hop question generation from knowledge graphs,” *inInternational Workshop on the Semantic Web* 2019.
- [4] Z. Fan, Z. Wei, P. Li, Y. Lan **and** X. Huang, “A question type driven framework to diversify visual question generation,” *inProceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18* International Joint Conferences on Artificial Intelligence Organization, **july** 2018, **pages** 4048–4054. DOI: 10.24963/ijcai.2018/563. **url:** <https://doi.org/10.24963/ijcai.2018/563>.
- [5] Z. Fan, Z. Wei, S. Wang, Y. Liu **and** X. Huang, “A reinforcement learning framework for natural question generation using bi-discriminators,” *inProceedings of the 27th International Conference on Computational Linguistics* Santa Fe, New Mexico, USA: Association for Computational Linguistics, **august** 2018, **pages** 1763–1774. **url:** <https://aclanthology.org/C18-1150>.
- [6] Y. Li, N. Duan, B. Zhou, X. Chu, W. Ouyang **and** X. Wang, *Visual question generation as dual task of visual question answering*, 2017. arXiv: 1709.07192 [cs.CV].
- [7] R. Zhang, J. Guo, L. Chen, Y. Fan **and** X. Cheng, “A review on question generation from natural language text,” *ACM Trans. Inf. Syst., jourvol* 40,

- number** 1, 2021, ISSN: 1046-8188. DOI: 10.1145/3468889. **url:** <https://doi.org/10.1145/3468889>.
- [8] Z. Fei, Q. Zhang **and** Y. Zhou, “Iterative GNN-based decoder for question generation,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* Online **and** Punta Cana, Dominican Republic: Association for Computational Linguistics, **november** 2021, **pages** 2573–2582. DOI: 10.18653/v1/2021.emnlp-main.201. **url:** <https://aclanthology.org/2021.emnlp-main.201>.
- [9] K. Xu, L. Wu, Z. Wang, Y. Feng, M. Witbrock **and** V. Sheinin, *Graph2seq: Graph to sequence learning with attention-based neural networks*, 2018. arXiv: 1804.00823 [cs.AI].
- [10] K. Xu, L. Wu, Z. Wang, M. Yu, L. Chen **and** V. Sheinin, *Exploiting rich syntactic information for semantic parsing with graph-to-sequence model*, 2018. arXiv: 1808.07624 [cs.CL].
- [11] L. Song, Y. Zhang, Z. Wang **and** D. Gildea, “A graph-to-sequence model for AMR-to-text generation,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* Melbourne, Australia: Association for Computational Linguistics, **july** 2018, **pages** 1616–1626. DOI: 10.18653/v1/P18-1150. **url:** <https://aclanthology.org/P18-1150>.
- [12] Y. Chen, L. Wu **and** M. J. Zaki, *Reinforcement learning based graph-to-sequence model for natural question generation*, 2020. arXiv: 1908.04942 [cs.CL].
- [13] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao **and** M. Zhou, *Neural question generation from text: A preliminary study*, 2017. arXiv: 1704.01792 [cs.CL].
- [14] P. Rajpurkar, J. Zhang, K. Lopyrev **and** P. Liang, *Squad: 100,000+ questions for machine comprehension of text*, 2016. arXiv: 1606.05250 [cs.CL].
- [15] X. Du, J. Shao **and** C. Cardie, *Learning to ask: Neural question generation for reading comprehension*, 2017. arXiv: 1705.00106 [cs.CL].
- [16] J. Mostow **and** W. Chen, “Generating instruction automatically for the reading strategy of self-questioning,” in *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling* NLD: IOS Press, 2009, 465–472, ISBN: 9781607500285.
- [17] M. Heilman **and** N. A. Smith, “Good question! statistical ranking for question generation,” in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*

- Los Angeles, California: Association for Computational Linguistics, **june** 2010, **pages** 609–617. **url:** <https://aclanthology.org/N10-1086>.
- [18] X. Sun, J. Liu, Y. Lyu, W. He, Y. Ma **and** S. Wang, “Answer-focused and position-aware neural question generation,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* Brussels, Belgium: Association for Computational Linguistics, 2018, **pages** 3930–3939. DOI: 10.18653/v1/D18-1427. **url:** <https://aclanthology.org/D18-1427>.
- [19] Y. Kim, H. Lee, J. Shin **and** K. Jung, *Improving neural question generation using answer separation*, 2018. arXiv: 1809.02393 [cs.CL].
- [20] L. Song, Z. Wang, W. Hamza, Y. Zhang **and** D. Gildea, “Leveraging context information for natural question generation,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* New Orleans, Louisiana: Association for Computational Linguistics, **june** 2018, **pages** 569–574. DOI: 10.18653/v1/N18-2090. **url:** <https://aclanthology.org/N18-2090>.
- [21] B. Liu, M. Zhao, D. Niu **and others**, “Learning to generate questions by learning what not to generate,” in *The World Wide Web Conference jourser WWW ’19*, San Francisco, CA, USA: Association for Computing Machinery, 2019, 1106–1118, ISBN: 9781450366748. DOI: 10.1145/3308558.3313737. **url:** <https://doi.org/10.1145/3308558.3313737>.
- [22] Y.-H. Chan **and** Y.-C. Fan, “A recurrent BERT-based model for question generation,” in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering* Hong Kong, China: Association for Computational Linguistics, **november** 2019, **pages** 154–162. DOI: 10.18653/v1/D19-5821. **url:** <https://aclanthology.org/D19-5821>.
- [23] W. L. Hamilton, R. Ying **and** J. Leskovec, *Inductive representation learning on large graphs*, 2018. arXiv: 1706.02216 [cs.SI].
- [24] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard **and** D. McClosky, “The Stanford CoreNLP natural language processing toolkit,” in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* Baltimore, Maryland: Association for Computational Linguistics, **june** 2014, **pages** 55–60. DOI: 10.3115/v1/P14-5010. **url:** <https://aclanthology.org/P14-5010>.

- [25] M. Honnibal **and** M. Johnson, “An improved non-monotonic transition system for dependency parsing,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* Lisbon, Portugal: Association for Computational Linguistics, **september** 2015, **pages** 1373–1378. DOI: 10.18653/v1/D15-1162. **url:** <https://aclanthology.org/D15-1162>.
- [26] M. Honnibal, Y. Goldberg **and** M. Johnson, “A non-monotonic arc-eager transition system for dependency parsing,” in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning* Sofia, Bulgaria: Association for Computational Linguistics, **august** 2013, **pages** 163–172. **url:** <https://aclanthology.org/W13-3518>.
- [27] J. Nivre **and** D. Fernández-González, “Arc-Eager Parsing with the Tree Constraint,” *Computational Linguistics*, **jourvol** 40, **number** 2, **pages** 259–267, **june** 2014, ISSN: 0891-2017. DOI: 10.1162/COLI_a_00185. **eprint:** https://direct.mit.edu/coli/article-pdf/40/2/259/1803243/coli_a_00185.pdf. **url:** https://doi.org/10.1162/COLI_a_00185.
- [28] J. Nivre, “Algorithms for deterministic incremental dependency parsing,” *Computational Linguistics*, **jourvol** 34, **number** 4, **pages** 513–553, 2008. DOI: 10.1162/coli.07-056-R1-07-027. **url:** <https://aclanthology.org/J08-4003>.
- [29] Y. Goldberg **and** J. Nivre, “A dynamic oracle for arc-eager dependency parsing,” in *Proceedings of COLING 2012* Mumbai, India: The COLING 2012 Organizing Committee, **december** 2012, **pages** 959–976. **url:** <https://aclanthology.org/C12-1059>.
- [30] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer **and** O. Levy, “SpanBERT: Improving pre-training by representing and predicting spans,” *Transactions of the Association for Computational Linguistics*, **jourvol** 8, **pages** 64–77, 2020. DOI: 10.1162/tacl_a_00300. **url:** <https://aclanthology.org/2020.tacl-1.5>.
- [31] M. Gardner, J. Grus, M. Neumann **and others**, “AllenNLP: A deep semantic natural language processing platform,” 2017. **eprint:** arXiv:1803.07640.
- [32] K. Papineni, S. Roukos, T. Ward **and** W.-J. Zhu, “Bleu: A method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, **july** 2002, **pages** 311–318.

- DOI: 10.3115/1073083.1073135. url: <https://aclanthology.org/P02-1040>.
- [33] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” *inText Summarization Branches Out* Barcelona, Spain: Association for Computational Linguistics, **july** 2004, **pages** 74–81. url: <https://aclanthology.org/W04-1013>.
- [34] P. Qi, Y. Zhang, Y. Zhang, J. Bolton **and** C. D. Manning, “Stanza: A Python natural language processing toolkit for many human languages,” *inProceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations* 2020.