TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Tìm hiểu về mô hình đồng tham chiếu trong xử lý ngôn ngữ tự nhiên

Trần Hữu Hiếu
hieu.th180078@sis.hust.edu.vn

Ngành: Khoa học máy tính

Giảng viên hướng dẫn: PGS. Nguyễn Thị Kim Anh _____

Chữ kí GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 1/2024

LỜI CAM KẾT

Họ và tên sinh viên: Trần Hữu Hiếu
Điện thoại liên lạc: 0868984191
Email: Trần Hữu Hiếu
Lớp: KSTN-CNTT-K63
Hệ đào tạo: Chương trình tài năng CNTT

Tôi – *Trần Hữu Hiếu* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *PGS.TS. Nguyễn Thị Kim Anh*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 9 tháng 1 năm 2024

Tác giả ĐATN

Trần Hữu Hiếu

LỜI CẢM ƠN

Lời đầu tiên, em xin cảm ơn đến toàn thể thầy cô trường công nghệ thông tin, những người đã cung cấp cho em những kiến thức quan trọng và cần thiết, tạo môi trường để em có thể chuẩn bị những hành trang cần thiết cho cuộc đời sau này.

Đặc biệt, em xin chân thành cảm ơn cô giáo PSG.TS Nguyễn Thị Kim Anh, người đã quan tâm, hướng dẫn em hết lòng để em có thể hoàn thành đồ án tốt nghiệp một cách tốt nhất.

Bên cạnh đó, em rất cảm ơn các bạn, các anh chị đã hỗ trợ và động viên em rất nhiều trong quá trình làm đồ án tốt nghiệp.

Cuối cùng, con xin bày tỏ lòng biết ơn sâu sắc đến bố mẹ, ông bà, những người đã kiên nhẫn hỗ trợ con nhiều mặt, tạo mọi điều kiện gián tiếp hay trực tiếp để con có thể hoàn thành được đồ án cuối cùng trong quãng đời sinh viên.

Tuy rằng đã cố gắng rất nhiều, nhưng với những hạn chế về kinh nghiệm và kĩ năng của một sinh viên, đồ án này không tránh khỏi những thiếu sót. Em rất mong thầy cô có thể cho em nhiều góp ý để em có thể hoàn thiện hơn.

Em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Cú pháp và ngữ nghĩa đã được chứng minh là đem lại lợi ích cho phân giải đồng tham chiếu trong các mô hình học máy thống kế truyền thống hoặc các mô hình dựa trên luật. Tuy nhiên, ứng dụng các thông tin này vào mô hình nơ ron đồng tham chiếu có lẽ vẫn còn ít được khám phá. Do đó trong đồ án này, em sẽ tìm hiểu và thử nghiệm các phương pháp tích hợp thông tin cú pháp và ngữ nghĩa vào mô hình nơ ron đồng tham chiếu. Cụ thể, có hai phương pháp tích hợp dựa trên mạng đồ thị chú ý GAT - một phiên bản của GNN - được đề xuất: (i) phương pháp đầu tiên tích hợp thông tin cú pháp phụ thuộc và nhãn vai trò ngữ nghĩa dựa trên đồ thị hỗn hợp, (ii) phương pháp thứ hai sẽ tích hợp thông tin từ cây cú pháp thành phần bằng mạng GAT hai chiều và cơ chế lan truyền bậc cao hơn. Cuối cùng, những biểu diễn token được tăng cường bởi các thông tin này sẽ được sử dụng để phát hiện đề cập và dự đoán đồng tham chiếu. Kết quả thực nghiệm trên bộ dữ liệu Ontonotes cho thấy hiệu quả của các phương pháp đề xuất khi làm tăng hiệu năng của mô hình baseline đầu tiên.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	**Ý nghĩa**
BERT	Bidirectional Encoder Representations from Transformers
BFS	Breath first search
c2f	Mô hình e2e với bộ mã hoá SpanBERT [1] và kĩ thuật course to fine pruning
Coref-cons	Mô hình đồng tham chiếu tích hợp cây cú pháp thành phần (consituent syntax)
Coref-HGAT	Mô hình phân giải đồng tham chiếu tích hợp cú pháp phụ thuộc và ngữ phĩa
CR	Phân giải đồng tham chiếu (coreference resolution)
DEP	Cây cú pháp phụ thuộc (Dependency tree)
e2e	Mô hình phân giải đồng tham chiếu đầu cuối (End-to-End Coreference) [2]
ELMo	Embeddings from Language Models
GAT	Graph Attention Network
GNN	Graph Neural Network
HOI	High order inference for coreference resolution [3]
NLP	Xử lý ngôn ngữ tự nhiên (Natural Language Processing)
NSP	Dự đoán câu tiếp thep (Next Sentence Prediction)
RGCN	Relational Graph Convolutional Network
s2e	Mô hình phân giải đồng tham chiếu đầu đến cuối (start-to-end coreference resolution [4])
SBO	Hàm mục tiêu dựa trên ranh giới của span (Span Boundary Objective)
SpanBERT	Span-based BERT
SRL	Semantic Role Labeling

Thuật ngữ	Ý nghĩa

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Để hiểu toàn bộ văn bản bất kì, máy tính cần xác định chính xác ý nghĩa của các từ hay cụm từ trong ngữ cảnh tương ứng với văn bản đó. Để làm được điều này, chúng ta cần phải phân giải đồng tham chiếu - một nhiệm vụ liên quan đến việc xác định và liên kết các đề cập (mention) trong một văn bản mà tham chiếu đến cùng một thực thể. Ứng dụng của phân giải đồng tham chiếu là rất đa dạng và quan trọng đối với các tác vụ xử lý ngôn ngữ tự nhiên khác. Chúng có thể là liên kết thực thể (Kundu và cộng sự 2018[5]), nhận dạng thực thể có tên (Dai và cộng sự 2019[6]), trả lời câu hỏi (Young và cộng sự 2018[7]), phân tích cảm xúc (Valdivia và cộng sự 2018[8]) và chatbot (Zhu và các cộng sự 2018[9]).

Tuy nhiên, việc xác định và phân cụm các tham chiếu (đề cập) không đơn giản nhất là khi gặp các đề cập mơ hồ, đề cập chung chung (generic mention), đề cập ẩn và nhiễu từ những cụm từ không có tham chiếu. Những điều này tạo ra thách thức cho các hệ thống để có thể phân giải hiệu quả và đạt được độ chính xác cao.

Trong vài thập kỉ vừa qua, quá trình nghiên cứu phân giải đồng tham chiếu đã trải qua ba giai đoạn chính, đó là phương pháp tiếp cận dựa trên luật, các phương pháp dựa trên học máy và các phương pháp sử dụng mạng nơ-ron nhân tạo. Với hướng tiếp cận học máy, có ba kiến trúc mô hình đã được phát triển, đó là mô hình các cặp đề cập (mention-pair model) (Ng và Cardie, 2002[10]; Bengtson và Roth, 2008[11]), mô hình xếp hạng đề cập (mention-ranking model) (Durrett và Klein, 2013[12]) và các mô hình cấp thực thể (entity-based model) (LHaghighi và Klein, 2010[13]; Wiseman và cộng sự, 2016[14]).

Mô hình cặp đề cập (mention-pairs) là bộ phân loại quyết định xem hai đề cập có đồng tham chiếu hay không mà không phụ thuộc vào cặp khác. Cụ thể, kiến trúc này gồm 3 pha: Tạo các training instances, huấn luyện mô hình, và sinh các cụm thực thể [15]. Mỗi pha đều có nhiều công trình nghiên cứu và việc cải thiện trên pha này chưa chắc đã giúp cải tiến trên những pha sau nó. Mặc dù mô hình cặp đề cập có ưu điểm là đơn giản nhưng nó có hai vấn đề chính[16]. Đầu tiên, bộ phân loại không so sánh trực tiếp các ứng cử viên antecedents (đề cập trước đó có khả năng đồng tham chiếu) với nhau, vì vậy nó không được đào tạo để biết giữa hai antecedents cái nào có khả năng liên kết cao hơn. Thứ hai, mô hình bỏ qua cấu trúc diễn ngôn (discourse model), chỉ để tâm đến đề cập (mention) chứ không để tâm vào thực thể (entities). Mỗi quyết định phân loại được thực hiện hoàn toàn cục bộ trên các cặp mà không xem xét đến các đề cập khác của cùng một thực thể. Hai

mô hình tiếp theo đều giải quyết một trong hai nhược điểm này.

Với mô hình xếp hạng đề cập, nó so sánh trực tiếp các ứng cử viên antecedent với nhau, chọn ra antecedent có điểm cao nhất cho mỗi đề cập [17]. Mô hình này được cải tiến bởi (Denis và Baldridge, (2008) [18]) khi sử dụng hàm softmax để xếp hạng các ứng cử viên, mà hầu hết các mô hình đồng tham chiếu hiện đại đều áp dụng[17]. Daum´e III and Marcu (2005)[19] đề xuất hệ thống phi đường ống (non-pipeline) đầu tiên chung cho hai nhiệm vụ phát hiện đề cập và dự đoán đồng tham chiếu.

Cả hai mô hình cặp đề cập và xếp hạng đề cập đều dự đoán dựa trên đề cập. Tuy nhiên, các mô hình cấp thực thể (entity-based models) liên kết mỗi đề cập không phải với các đề cập trước đó mà với một thực thể trước đó (cụm đề cập). Các mô hình cấp thực thể có gây ấn tượng ([16]) nhưng trong thực tế việc sử dụng thông tin cấp độ cụm không làm tăng nhiều hiệu năng (Xu và Choi 2020[20]). Vì vậy nên các mô hình xếp hạng đề cập vẫn là xu hướng chính.

Ngoài ra, các phương pháp học máy truyền thống phụ thuộc nhiều vào những đặc trưng được thiết kế thủ công (hand-engineer features) từ cấu trúc cú pháp và ngữ nghĩa[17]. Hướng tiếp cận mạng nơ ron tuy có kiến trúc tương tự như các mô hình học máy này, nhưng lại có thể huấn luyện toàn bộ (end-to-end) mà không cần sự can thiệp của con người. Điều này dẫn đến giảm bớt phụ thuộc vào các bộ phân tích cú pháp hay các đặc trưng được thiết kế thủ công. Tiêu biểu là các mô hình xếp hạng spans kết hợp với bộ mã hoá transformer: Kantor et al, (2019) [21], Joshi et al (2019), (2020) [22][1], Kirstain et al.(2021)[4]..v..

Tuy nhiên, nếu như các thông tin cú pháp và ngữ nghĩa cung cấp nhiều ích lợi cho các mô hình học máy truyền thống, vậy liệu có thể tận dụng những thông tin này cho các mô hình mạng nơ ron hay không?. Đồ án sẽ tập trung vào tìm hiểu vấn đề này.

1.2 Các giải pháp hiện tại và hạn chế

Việc tận dụng thông tin cú pháp trong mô hình nơ ron phân giải đồng tham chiếu vẫn còn ít được khai phá. Xu và Yang (2019)[23] đã sử dụng GCN tích hợp cây cú pháp phụ thuộc vào mạng nơ ron tuần tự. Điều này cải thiện mức độ phân giải đại từ theo giới tính. Tuy nhiên, mô hình đã không đánh giá trên các bộ dữ liệu lớn hơn như Ontonotes hay kiểm tra xem liệu cây cú pháp có đóng góp như thế nào cho phân giải đồng tham chiếu hay không. Trieu et al. (2019) [24], Kong và Jian (2019) [25] đã sử dụng các cây cú pháp thành phần (constituent tree) như những ràng buộc cứng (hard constraints) để loại bỏ các đề cập không hợp lệ, giúp có được mô đun phát hiện đề cập (mention detectors) và các mô đun đồng tham chiếu tổng

thể (overall coreference resolvers) tốt hơn[26]. Tuy nhiên, những phương pháp này không bảo tồn toàn bộ cấu trúc của cây ban đầu, do sự ánh xạ cây dưới dạng một đường đi gồm dãy các nút với một tập hợp các đặc trưng. Điều này làm mất đi thông tin cấu trúc phân cấp. Ngoài ra, mô hình của Kong và Jian vẫn sử dụng những đặc trưng phức tạp được thiết kế thủ công[27].

Tuy nhiên đối với tác vụ khác, từ lâu cú pháp đã được sử dụng để tăng cường các mô hình mạng nơ ron [26]. Xu hướng tiếp cần thường cố gắng để nắm bắt thông tin cấu trúc được mã hóa trong cây cú pháp phụ thuộc bằng mạng nơ ron đồ thị. Marcheggiani và Titov [26]. (2017)[28] và Bastings et al. (2017)[29] đã áp dụng Mạng tích chập đồ thị (GCN) (Schlichtkrull et al. (2017)[30]) để kết hợp các cây phụ thuộc nhằm nắm bắt (capture) các mối quan hệ phi cục bộ giữa các từ. Wang et al. (2020)[31] dùng cây phụ thuộc được tạo lại (reshaped) bằng mạng đồ thị quan hệ dựa trên cơ chế chú ý (RGAT) để nắm bắt sự phụ thuộc dài (long-range dependencies) một cách hiệu quả trong khi bỏ qua các mối quan hệ gây nhiễu.

Còn đối với thông tin ngữ nghĩa, các hệ thống đồng tham chiếu dựa trên thống kê trước đây đã thành công tích hợp thông tin như nhãn vai trò ngữ nghĩa (SRL) (Ponzetto và Strube, 2006 [32], Kong và cộng sự, 2009 [33]). Trong đó, các tác giả Ponzetto và Strube đã chỉ ra rằng đặc trưng ngữ nghĩa này giúp tăng cường phân giải đại từ và phân giải danh từ chung. Tuy vậy theo như em biết, trong các mô hình nơ-ron phân giải đồng tham chiếu thì hiệu quả của SRL có lẽ mới chỉ được kiểm chứng trong nghiên cứu của Jiang and Cohn (2021) [27] .

1.3 Mục tiêu và định hướng giải pháp

Trong đồ án này, em sẽ trình bày hai mô hình của Jiang and Cohn (2021)[27], (2022)[17] dựa trên mô hình phân giải đồng tham chiếu của c2f[1] của Lee et al.(2018)[3], nhưng mở rộng thêm bằng cách kết hợp các thông tin cú pháp và ngữ nghĩa.

Cụ thể với mô hình đầu tiên, thông tin cú pháp và ngữ nghĩa sẽ được mã hoá trong một đồ thị hỗn hợp. Các nhúng biểu diễn node được cập nhật nhiều lần thông qua cơ chế lan truyền thông điệp và được tích hợp vào các nhúng gốc theo ngữ cảnh ban đầu bằng cách sử dụng mô-đun tích hợp dựa trên cơ chế chú ý và cơ chế cổng (gating mechanism).

Tương tự, mô hình thứ hai sẽ kết hợp thông tin từ cây cú pháp thành phần nhưng với cơ chế lan truyền thông tin lân cận bậc cao hơn. Cụ thể, theo Jiang và Cohn, để nắm bắt thông tin vùng lân cận bậc hai và giảm vấn đề làm mịn quá mức (over-smoothing problem) (Chen et al. 2020[34]) của mạng đồ thị dựa trên cơ chế chú

[1] c2f là mô hình đông tham chiếu e2e kết hợp kĩ thuật course to fine pruning. Mô hình này sẽ được trình bày chi tiết trong chương 2.

ý bằng cách xếp chồng quá nhiều lớp, mô hình sẽ thiết kế thêm các cạnh hai bước nhảy dựa trên cây cú pháp thành phần . Ngoài ra, các nút constituent được cập nhật nhiều lần bằng cách sử dụng mạng GAT hai chiều với thông tin nhãn và biên của các constituent được lan truyền để tăng cường các nhúng gốc theo ngữ cảnh ban đầu.

Hơn nữa, để kiểm tra những thông tin này có thể đóng góp vào mô hình phân giải đồng tham chiếu khác hay không, em sẽ kết hợp thông tin cú pháp phụ thuộc và ngữ nghĩa vào mô hình s2e của Kirtain at al. (2021)[4]. Mô hình s2e khác c2f ở điểm nó không cần tính biểu diễn span nhưng vẫn có thể tính điểm số đề cập và điểm số đồng tham chiếu, do đó nó giúp tiết kiệm nhiều bộ nhớ hơn.

1.4 Đóng góp của đồ án

Đồ án này có đóng góp chính như sau:

1. Đồ án thử nghiệm hai phương án tích hợp thông tin cú pháp và ngữ nghĩa để cải thiện các mô hình đồng tham chiếu khác nhau.

1.5 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau:

Chương 2 trình bày về cơ sở lý thuyết của đồng tham chiếu, bao gồm các dạng đồng tham chiếu cùng với các nghiên cứu liên quan. Ngoài ra, chương cũng sẽ trình bày sâu về kiến trúc xếp hạng đề cập, bộ mã hoá SpanBERT để biểu diễn thông tin đầu vào và mạng đồ thị dựa trên cơ chế chú ý để lan truyền thông tin.

Chương 3 trình bày về hai phương pháp tích hợp cú pháp và ngữ nghĩa: Phương pháp đầu tiên sẽ tích hợp thông tin cú pháp phụ thuộc và nhãn vai trò ngữ nghĩa dựa trên đồ thị hỗn hợp. Phương pháp thứ hai sẽ tích hợp thông tin cú pháp thành phần dưa trên cơ chế lan truyền bậc cao hơn và bộ mã hoá GAT hai chiều.

Chương 4 sẽ trình bày về bộ dữ liệu Ontonotes, các tham số đánh giá, đồng thời báo cáo kết quả kiểm thử trên bộ dữ liệu này. Cụ thể, chương sẽ kiểm thử mức độ đóng góp của các tầng tích hợp cú pháp và ngữ nghĩa trong việc nâng cao hiệu năng mô hình đồng tham chiếu. Cuối cùng, chương sẽ so sánh và nhận xét các kết quả đã kiểm thử được.

Chương 5 sẽ đưa ra một số kết luận và hướng pháp triển trong tương lai.

CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT

2.1 Tổng quan chương

Ở chương trước, em đã giới thiệu sơ qua về phân giải đồng tham chiếu cũng như tổng quan các phương pháp. Trong chương này em sẽ trình bày chi tiết hơn về lý thuyết đồng tham chiếu, thể loại, các nghiên cứu liên quan cũng như các mô hình xếp hạng đề cập (mention-ranking model). Ngoài ra, chương còn giới thiệu kiến trúc mạng đồ thị GAT cùng với cây cú pháp và tác vụ gán nhãn vai trò ngữ nghĩa.

2.2 Các nghiên cứu liên quan

Cây cú pháp đã được sử dụng trong những nghiên cứu sơ khai về phân giải đồng tham chiếu. Thuật toán phân giải đại từ Hobbs(1978) để tìm ra các antecedent, đã duyệt cây cú pháp thành phần (constituent tree[1]) bằng BFS. Sau đó Ge, Hale và Charniak (1998)[35] tận dụng thuật toán này để mã hoá thứ hạng các antecedents. Bergsma và Lin (2006)[36] đã khai thác các đặc trưng liên quan đến đường đi dựa trên cây phân tích cú pháp phụ thuộc để đo khả năng phân giải đồng tham chiếu, mà cụ thể là chuỗi các từ và các nhãn phụ thuộc (dependency label) giữa một đại từ và ứng cử viên antecedent của nó. Thông tin cú pháp cũng đã được áp dụng thành công trong mô đun xác định anaphora (anaphora determination). Trong đó Kong at al.2010[37]) sử dụng các đặc trưng liên quan đến đường đi gốc giữa nút gốc và đề cập. Cụ thể nó sẽ giữ lại các đỉnh và đường đi liên quan đến đề cập dựa trên thông tin cú pháp phụ thuộc và SRL, trong khi bỏ qua những thông tin nhiễu, từ đó tạo ra cây phân tích cú pháp động mới (dependency-driven dynamic syntactic parse tree). Cây cú pháp này sẽ được dùng trong nhiệm vụ xác định anaphora.

Ngược lại, có rất ít nỗ lực đã được thực hiện để đánh giá ích lợi của cây cú pháp cho các mô hình mạng nơ ron đồng tham chiếu. Trieu et al. (2019) [24] và Kong và Jian (2019) [25] đã sử dụng cây cú pháp (constituent trees) như các dấu hiệu để lọc những ứng cử viên đề cập không hợp lệ. Tuy nhiên, Trieu đã bỏ qua cấu trúc phân cấp còn Kong và Jian đã ánh xạ cây sang dãy các nút dựa trên thuật toán duyệt post-order, do đó cũng làm mất cấu trúc của câu.

Để lấp đầy các khoảng trống này, Jiang và Cohn đã đề xuất hai phương pháp tích hợp cú pháp phụ thuộc và cú pháp thành phần mà trong đó, phương pháp thứ hai có thể bảo toàn cấu trúc phân cấp của cây. Phương pháp này giống với phương pháp Marcheggiani và Titov (2020) [38] khi đều sử dụng cơ chế lan truyền thông điệp. Tuy nhiên nó khác ở 3 điểm: (i) Thứ nhất, mô hình mở rộng các cây thành

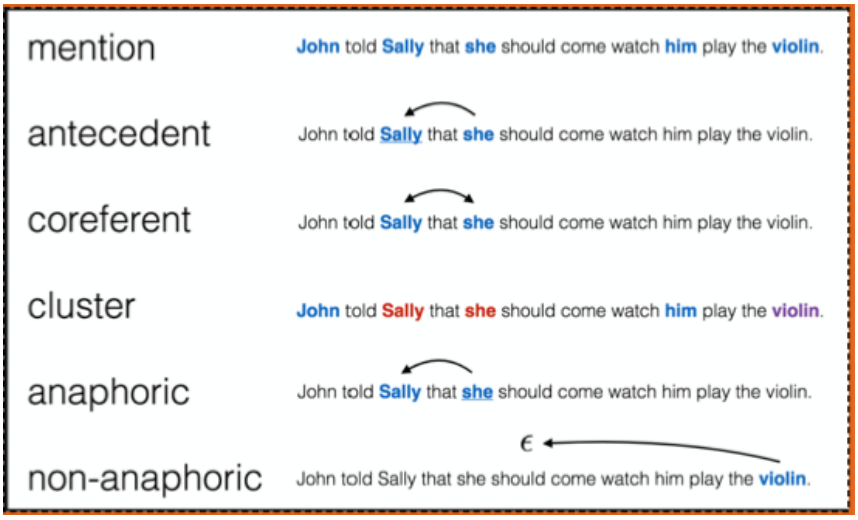
[1]Vì để tránh nhầm lẫn với cây cú pháp phụ thuộc, nên em dùng thuật ngữ là "cây cú pháp thành phần " cho constituent tree.

phần ban đầu với các đồ thị hai chiều và các cạnh bậc hai để thu được thông tin các nút lân cận ở phạm vi dài hơn. (ii) Thứ hai, các nút constituent được khởi tạo bởi nhãn thể loại và hai token biên thay vì là các vector 0. Điều này tương tự như cách tính biểu diễn của đề cập trong mô hình đồng tham chiếu nơ ron cơ bản. Hơn nữa, (iii) mục đích của hai mô hình khác nhau khi mô hình của Marcheggiani và Titov là để dự đoán predicates và argument spans trong SRL - một nhiệm vụ chỉ ở cấp độ câu thay vì ở cấp độ tài liệu.

Phần tiếp theo, em xin trình bày về lý thuyết cơ bản đồng tham chiếu, các thể loại, cũng như kiến trúc điển hình.

2.3 Khái niệm và thể loại đồng tham chiếu

Đồng tham chiếu (coreference) được định nghĩa là khi một hoặc nhiều đề cập (mention) trong văn bản tham chiếu trở lại thực thể xuất hiện trước nó. Khi đó, phân giải đồng tham chiếu là tác vụ tìm kiếm tất cả các đề cập mà tham chiếu tới bất kỳ thực thể nào và phân cụm chúng. Ta xét ví dụ sau:



Hình 2.1: Các khái niệm cơ bản và ví dụ trong đồng tham chiếu[39]

Trong ví dụ này, một chuỗi từ liên tiếp bất kì được gọi là span. Các cụm từ được tô màu tùy theo chúng có phải là thực thể hay không. Ta gọi những cụm từ này là đề cập/tham chiếu (mention). Các từ cùng màu sẽ là thành viên của cùng một cụm tham chiếu (cluster). Antecedent là đề cập được đề cập sau đó tham chiếu đến. Ngoài ra, anaphora là đề cập mà ý nghĩa phụ thuộc vào một thực thể đã được giới thiệu trước đó.

2.3.1 Một số thể loại đồng tham chiếu

Pronominal anaphora Đồng tham chiếu đại từ xuất hiện khi mà một đại từ tham chiếu tới một đối tương đã biết trước đó. Đây là loại phổ biến nhất xảy ra

trong lời nói hàng ngày và chiếm một phần đáng kể trong số các anaphors mà chúng ta thường thấy trong dữ liệu web như bài đánh giá, bài đăng trên blog[40]. Thông thường, đại từ sẽ có các thể loại như đại từ nhân xưng, đại từ chỉ định hoặc không xác định. Xét ví dụ sau:

"Although [Mr. Clinton] is out of office, but dont' worry, [he] says [he]'ll still be in this office. We need [his] wife because if there is [anyone] who can organize a meeting, it is [her]."

Trong đó cụm đồng tham chiếu thứ nhất *[his, he, he, Mr.Clinton]* gồm các đại từ nhân xưng và đại từ sở hữu, cụm thứ hai *[office, this office]* gồm đại từ chỉ định *this*, và cụm thứ ba *[her, anyone]* gồm đại từ không xác định *anyone*.

Verb mention xuất hiện khi một đại từ hoặc một cụm danh từ tham chiếu tới một hành động hay một trạng thái trước đó được mô tả bởi động từ:

"She [studied] hard for the exam, and the effort paid off. [It] significantly improved her grades."

Trong ví dụ này, đại từ *it* tham chiếu tới động từ *studied* chỉ hành động học tập. Số lượng tham chiếu kiểu này chỉ chiếm hơn 1,5% trong bộ dữ liệu OntoNotes tiếng anh [41].

Còn có nhiều dạng tham chiếu khó khác như:

Split anaphora Trong đó, một đề cập mà tham chiếu tới nhiều hơn một antecedent ở trước nó. Những trường hợp này không xuất hiện trong bộ Ontonotes[17]. Dưới đây là một ví dụ mà đại từ *they* tham chiếu tới 2 antecedent là *Katherine* và *Maggie*:

"[Katherine] and [Maggie] love reading. [They] are also the members of the reader's club."

Cataphora ngược lại với anaphora, khi ý nghĩa của một đề cập trong văn bản xác định bởi một đề cập được giới thiệu sau nó. Trong vi dụ ở dưới, đại từ *she* tham chiếu tới đề cập sau nó là *Marry*:

"If [she] was hungry, [Mary] always had a snack.

Non-anaphoric pronominal Ngoài ra, còn một số đại từ là tham chiếu rỗng hoặc không tham chiếu tới bất kì antecedent nào ở trước đó. Các hệ thống phân giải đồng tham chiếu đều gặp khó khăn khi xác định (identify) và loại bỏ các *tham*

chiếu giả gây nhiễu này[15]. Một trường hợp điển hình là đại từ *it* (pleonastic) với ví dụ: *"it was raining heavily"*.

2.4 Mô hình đồng tham chiếu đầu cuối (e2e)

Phần này sẽ mô tả thuật toán neural e2e của (Lee et al.2017 [2]) (được mở rộng dựa trên Joshi et al. (2019)[22] và những tác giả khác) cùng với một số thảo luận. Cụ thể, thuật toán hoạt động như sau: Đầu tiên thuật toán sẽ xem xét tất cả các spans, tính toán mention scores cho mỗi span sau đó xếp hạng và cắt bớt những spans dựa trên scores vừa tính được. Sau đó, với mỗi đề cập sẽ tính điểm số đồng tham chiếu cho từng antecedent, và chọn ra antecedents có điểm số cao nhất. Cuối cùng sẽ liên kết và phân cụm các đề cập này.

Ta có thể mô hình hoá bài toán như sau: Cho một tài liệu D với các từ T, mô hình xem xét tất cả $\frac{T(T+1)}{2}$ spans trong D (chú ý số lượng spans <=30). Nhiệm vụ là gán cho mỗi span i một antecedent y_i, là một biến ngẫu nhiên nằm trong tập $Y(i) = \{\sigma, 1, 2 \ldots, i-1\}$; trong đó σ là một dummy antecedent đặc biệt. Nếu $y_i = \sigma$ thì có hai trường hợp: span i không phải là đề cập hoặc là đề cập đầu tiên trong chuỗi đồng tham chiếu, hoặc là non-anaphoric (singleton mention).

Đối với mỗi cặp spans i và j, hệ thống sẽ tính điểm số $s(i, j)$ (coreference score) cho liên kết đồng tham chiếu giữa span i và span j và dự đoán antecedent dựa trên điểm số này. Ngoài ra, với mỗi span i, hệ thống tính thêm softmax $P(y_i)$ trên các antecedents để huấn luyện mô hình:

$$P(y_i) = \frac{exp(s(i, y_i))}{\Sigma_{y' \in Y(i)} exp(s(i, y'))}$$

Cụ thể, điểm số $s(i, j)$ bao gồm ba thành phần:

$$s(i, j) = s_m(i) + s_m(j) + s_c(i, j)$$

Trong đó $m(i)$; $m(j)$ tương ứng cho biết khả năng liệu span i, j có phải là một đề cập (mention) hay không; và $s_c(i, j)$ cho biết khả năng j có phải là antecedent của i.

Đối với dummy antecedent ϵ , điểm $s(i, \epsilon)$ được cố định bằng 0 với mọi span i. Bằng cách này, nếu tồn tại non dummy antecedents có score lớn hơn không, model sẽ chọn ra antecedent có score cao nhất, nhưng ngược lại nếu tất cả antecedent score của span i đều âm thì model sẽ chọn dummy antecedent, tức span i không có đồng tham chiếu. Trong trường hợp này, model sẽ quyết định xem span i có phải là singleton mention phụ thuộc vào $s_m(i) \geq 0$ hay không.

2.4.1 Biểu diễn span

Để tính toán hai hàm $s_m(i)$ và $s_c(i,j)$ tương ứng với việc ghi điểm cho span i hoặc một cặp spans (i, j), chúng ta sẽ cần một cách để biểu diễn một span. Dựa hai ý tưởng của (Lee et al.2016)[42] và (Durrett and Klein. 2013[12]), thuật toán e2e biểu diễn mỗi span bằng cách cố gắng kết hợp 3 token: token đầu tiên, token cuối cùng và token quan trọng nhất (head word). Đầu tiên, thuật toán chạy từng phân đoạn văn bản thông qua một bộ mã hoá (như SpanBERT) để tạo các nhúng từ x_i cho mỗi token i. Span i sau đó được biểu thị bằng một vectơ g_i là sự kết hợp của nhúng token đầu tiên và token cuối cùng, nhúng head word của span được biểu diễn dựa trên cơ chế chú ý và nhúng mã hoá đặc trưng về độ dài của span:

$$g_i = [x_{START(i)}, x_{END(i)}, \hat{x_i}, \phi(i)]$$

Trong đó, vector $\hat{x_i}$ giúp biểu diễn xem một word hay token có khả năng là syntactic head word của span không; vì head word sẽ giúp đóng góp tốt vào việc dự đoán đồng tham chiếu giữa hai đề cập (Lee et al.2017[2]). Để biểu diễn vector này, mô hình sẽ cho các nhúng token x_t đi qua mạng feedforward, sau đó nhân tích vô hướng với một vector trọng số w_a, ta được các trọng số chú ý α_t:

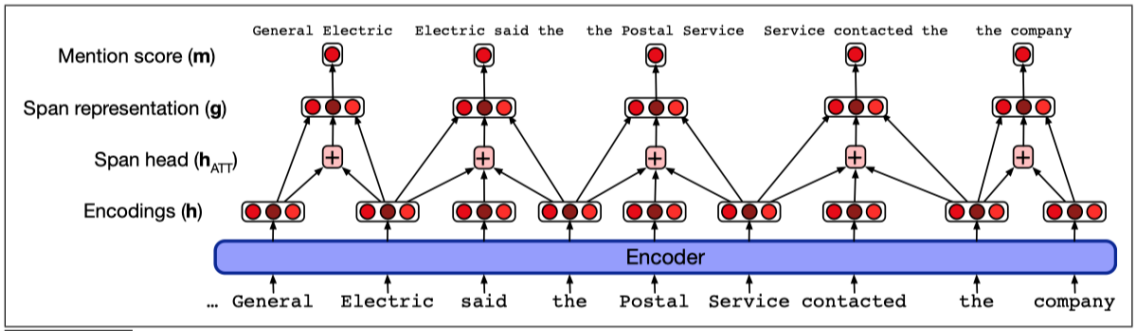
$$\alpha_t = w_\alpha . FFNN_\alpha(x_t)$$

Trọng số này thể hiện mức độ chú ý của span i đối với từng token và được chuẩn hoá thành phân phối thông qua hàm softmax:

$$\alpha_t = \frac{exp(\alpha_t)}{\Sigma_{k=START(i)}^{END(i)} exp(\alpha_k)}$$

Sau đó, vectơ $\hat{x_i}$ sẽ được tính bằng tổng trung bình của các nhúng từ x_t trong span i dựa trên phân phối chú ý:

$$\hat{x_i} = \Sigma_{t=START(i)}^{END(i)} a_{i,t} . x_t$$



Hình 2.2: Hình vẽ mô tả quá trình tính vector biểu diễn của span g đối với tập con của các spans bigram và triagram[16].

2.4.2 Tính điểm số đề cập $s_m(i)$ và điểm số antecedent $s_a(i,j)$

Chúng ta đã có biểu diễn của từng span, cho chúng đi qua mạng FeedForward ta được mention scores:

$$s_m(i) = w_m.FFNN_m(g_i)$$

Với antecedent scores, ta sẽ nối vector span i, span j, tích element wise giữa 2 span này, nhúng của các đặc trưng về thể loại, người nói (nếu có), và khoảng cách giữa 2 span. Sau đó đưa vector hợp nhất này đi qua mạng feed forward ta được $s_a(i,j)$:

$$s_a(i,j) = w_a.FFNN_a([g_i, g_j, g_i \circ g_j, \phi(i,j])$$

Trong đó, thông tin speaker sẽ được mã hoá thành đặc trưng nhị phân để cho biết hai span có cùng người nói hay không. Đặc trưng khoảng cách sẽ được xếp vào 10 bucket (thùng) tương ứng, sao cho các khoảng cách gần nhau sẽ nằm trong cùng 1 bucket và mỗi bucket sẽ học một nhúng tương ứng. Cụ thể ta có phân hoạch sau: [0, 1, 2, 3, 4, 5-7, 8-15, 16-31, 32-63, 64+]. Ngoài ra, đặc trưng thể loại cũng được mã hoá thành nhúng dựa trên từ điển.

a, Thảo luận và hướng phát triển

Wu et al. 2020 [43] thiết kế lại khi chọn mã hoá speaker đầu vào cùng với văn bản mỗi khi speaker thay đổi thay vì như đặc trưng nhị phân ở trên. Phương án này giúp mô hình hoạt động tốt hơn trên văn bản có một số lượng lớn speakers [43]. Cụ thể, các biểu diễn được học có tiềm năng khái quát hoá thông tin toàn cầu của người nói trong tình huống đối thoại nhiều bên, dẫn đến mô hình hoá bối cảnh tốt hơn. Hơn nữa, nó còn giúp siêu dữ liệu này phù hợp với mọi mô hình[4].

Ngoài ra, vì chi phí biểu diễn span là tốn kém, Kirstain et al. 2021[4] đề xuất mô

hình có thể tính điểm số đề cập và điểm số antecedent mà không cần tính vector biểu diễn spans. Mô hình này sẽ được giới thiệu trong phần 2.6.

2.4.3 Cắt tỉa span và antecedent

Cắt tỉa bớt spans Vì khi tính $s_a(i, j)$ thuật toán naive cần biểu diễn $o(T^4)$ cặp spans khiến chi phí bộ nhớ quá lớn nên chúng ta sẽ chỉ xem xét span có độ dài không vượt quá L và nằm trong 1 câu. Sau đó, chúng ta sẽ cắt tỉa thêm bằng cách lọc ra $\lambda * T$ span có mention scores cao nhất. Ngoài ra, các span không được phép giao nhau.

Cắt tỉa bớt antecedents Cuối cùng, mỗi span sẽ chỉ xem xét K antecedent gần nó nhất. Điều này giúp giảm chi phí bộ nhớ tensor xuống còn kích thước $\leq \lambda * T * K * (3|g| + o)$.

a, Thảo luận và hướng phát triển

Việc cắt tỉa spans như trên vẫn để lại nhiều span gây nhiễu cho pha liên kết đồng tham chiếu. Do đó, Trieu et al.2019 [24] đã đề ra phương án sử dụng cây cú pháp để loại bỏ các span nhiễu trước khi tính vector biểu diến span. Cụ thể, Trieu và các cộng sự đã tạo ra các mẫu cú pháp (tập các nhãn) của đề cập chân lý dựa trên các cây cú pháp được sinh ra từ bộ constituent parser. Sau đó, chỉ những span mà thoả mãn các mẫu này mới đưa vào mô hình.

Ngoài ra, việc chỉ chọn K antecedents gần nhất làm giới hạn độ dài liên kết đồng tham chiếu. Tuy nhiện, trong những văn bản ở lĩnh vực y sinh, khoảng cách giữa hai đề cập có thể xa hơn nhiều [17]. Để giải quyết vấn đề này, Lee et al.2018 [3] đã giới thiệu kĩ thuật cắt tỉa từ thô đến tinh (course to fine prunning) sẽ được trình bày ở phần tiếp theo.

2.4.4 Huấn luyện mô hình

Đối với việc huấn luyện, ta chỉ có nhãn chân lý trên toàn cụm, nên mỗi đề cập (mention) có thể có nhiều antecedent chân lý ở trước nó. Do đó, cần tối đa hóa tổng các phân phối softmax của các antecedent chân lý này. Đối với một đề cập i với tập các ứng cử viên antecedent Y(i), đặt GOLD(i) là tập hợp các đề cập trong cụm chân lý chứa i. Vì tập hợp các antecedents xảy ra trước i là Y(i), nên tập hợp các mentions trong cụm chân ý đó xuất hiện trước i là Y(i) ∩ GOLD(i). Do đó, với mỗi đề cập i, cần cực đại hóa tổng các phân phối softmax:

$$\Sigma_{\hat{y} \in Y(i) \cap GOLD(i)} P(\hat{y})$$

Nếu đề cập i không nằm trong cụm chân lý thì $GOLD(i)$ được gán bằng dummy

antecedent ϵ.

Để huấn luyện mô hình, ta cần biến đổi tổng phân phối này thành hàm mất mát. Với mỗi đề cập, ta dùng hàm marginal log-likelihood bằng cách lấy -log của tổng phân phối. Cuối cùng, ta tính tổng các hàm mất mát cho tất cả các đề cập:

$$L_{cluster} = \Sigma_{i=2}^{N} - log\Sigma_{\hat{y} \in Y(i) \cap GOLD(i)} P(\hat{y}) \quad (1)$$

a, Thảo luận và hướng phát triển

Tuy nhiên, việc huấn luyện như trên có thể làm việc học trở nên chậm và không hiểu quả nhất là đối với việc phát hiện đề cập (mention detection) do mô đun này chỉ được giám sát gián tiếp thông qua hàm mục tiêu (1)[44]. Do đó, Zhang et al. 2018 [44] đề xuất huấn luyện trực tiếp mô đun lọc đề cập này:

$$L = L_{cluster} + \beta * L_{mention}$$

trong đó β là trọng số và $L_{mention}$ là hàm mất mát nhị phân cross-entropy:

$$L_{mention} = \Sigma_{i=1}^{N} y_i * log(\hat{y}_i + (1 - y_i) * log(1 - \hat{y}_i) \quad (2)$$

với $\hat{y}_i = sigmoid(s_m(i))$, $y_i = 1$ nếu span i là đề cập chân lý, ngược lại $y_i = 0$.

Đây cũng chính là hàm mục tiêu được chọn cho các mô hình đề xuất ở chương 3.

[2] Lai et al. 2022 [45] cũng đề xuất phương án huấn luyện khác khi chia ra hai bước: (i) Bước đầu sẽ tiền huấn luyện mô đun phát hiện đề cập bởi hàm mục tiêu (2) và giảm λ xuống 0.25. (ii) Sau đó, sẽ huấn luyện đồng thời hai mô đun phát hiện đề cập và liên kết đồng tham chiếu bằng cách tối ưu hàm mục tiêu (1).

Cụ thể, theo Lai và cộng sự, việc thiết lập $\lambda = 0.25$ giúp giảm bớt số lượng ứng cử viên span phải lọc, trong khi tiền huấn luyện giúp giữ lại được số lượng đề cập chân lý trong top N span cao hơn (tức mention recall cao hơn) so với chỉ tối ưu hàm mục tiêu (1).

2.5 Kĩ thuật course to fine pruning

Để giải quyết vấn đề bộ nhớ và giới hạn liên kết đồng tham chiếu ở phần 2.4.3. Lee et al. 2018 [3] đã đề xuất kĩ thuật cắt tỉa từ thô đến tinh bằng cách thêm thành

[2]Lai và cộng sự không cung cấp mã nguồn cho việc tiền huấn luyện và giới hạn về chi phí nên em không tiếp tục thử nghiệm.

phần $s_c(i,j)$ vào $s(i,j)$:

$$s(i,j) = s_m(i) + s_m(j) + s_c(i,j) + s_a(i,j)$$

Trong đó $s_c(i,j)$ là hàm một song tuyên tính:

$$s_c(i,j) = g_i^T W_c g_i$$

Điều này giúp tính antecedent score thô $s_c(i,j)$ trên toàn bộ top span mà không cần phải giới hạn k antecedent gần nhất vì $s_c(i,j)$ chỉ yêu cầu ma trận có kích thước $M * |g|$ và $M * M$ thay vì tensor $M * K * (3|g| + o)$ như $s_a(i,j)$. Sau đó ta có thể tính $s_a(i,j)$ trên một tập nhỏ antecedents còn lại.

Cụ thể ta có quy trình cắt tỉa gồm 3 bước như sau: (i) Bước đầu tiên sẽ lọc ra top M spans dưa trên mention scores $s_m(i)$ của từng span. (ii) Tiếp theo với mỗi span sẽ lọc ra top K antecedents dựa trên 3 thành phần đầu tiên $s_m(i) + s_m(j) + s_c(i,j)$. (iii) Và bước cuối cùng sẽ tình điểm số đồng tham chiếu $s(i,j)$ dựa trên các cặp span còn lại.

Ngoài ra, e2e còn gặp một vấn đề chung của các mô hình xếp hạng span là phụ thuộc vào các quyết định cục bộ giữa các đề cập. Từ đó, khi phân nhóm các đề cập có thể tạo ra các cụm không nhất quán (inconsistent cluster[3]). Lee et al. 2018[3], Xu và Choi. 2020[20] đã đề xuất các kĩ thuật suy diễn bậc cao (high-order inference) với hi vọng khắc phục được vấn đề này. Tuy nhiên theo Xu và Choi, những kĩ thuật này vẫn tốn nhiều bộ nhớ và thời gian do vòng lặp mà không làm tăng hiệu năng của mô hình lên nhiều.

2.6 Mô hình start to end coreference resolution (s2e)

$$s(c,q) = s_m(q) + s_m(c) + s_a(c,q)$$

Việc biểu diễn span và các cặp spans rất tốn kém, vậy có cách nào vẫn tính được điểm số đề cập và điểm số antecedents mà không cần biểu diễn span hay cặp span hay không?. Kirtain et al. 2021 đã sử dụng sự kết hợp của các hàm song tuyến tính (Dozat và Manning (2016)[46]) để làm điều này.

Cụ thể, Kirtain và cộng sự sử dụng hai biên của một span (chứ không phải tất cả các token) để tính toán điểm số đề cập $s_m(i)$ và antecedent scores $s_a(i,j)$.

Đầu tiên, mô hình tính các vector biểu diễn *bắt đầu* và *kết thúc* cho mỗi nhúng

[3]Một ví dụ với 2 liên kết động tham chiếu *(I, you)* và *(you, both of you)* là cụm *(I, you, both of you)*. Trong đó đề cập *I* và *both of you* không cùng số lượng.

token từ bộ mã hoá:

$$m_x^s = GeLU(W_s x)$$

$$m_x^e = GeLU(W_e x)$$

Sau đó, với mỗi span độ dài không vượt quá L, ta sử dụng hàm biaffine để tính toán mention scores qua biểu diễn bắt đầu của token đầu tiên và biểu diễn kết thúc của token cuối cùng:

$$s_m(q) = V_s.m_{q_s}^s + V_e.m_{q_e}^e + m_{q_s}^s.B_m.m_{q_e}^e$$

Trong đó, các vec-tơ V_s, V_e và ma trận B_m là các tham số huấn luyện của hàm tính điểm đề cập s_m. Theo Kirstain et al. 2021[4], khả năng (likelihood) token bắt đầu/cuối cùng q_s, q_e của span là điểm đầu tiên/ cuối cùng của một đề cập bất kì được đo bởi hai thành phần đầu tiên. Thành phần thứ ba thể hiện sự tương tác và mối quan hệ giữa hai token. Cụ thể nó đo khả năng hai token đó có là hai biên của chung một đề cập hay không.

Tương tự như mô hình e2e, sau khi tính các điểm số đề cập (mention scores), mô hình chỉ giữ lại $\lambda.T$ các ứng cử viên span có điểm số cao nhất để tránh độ phức tạp $O(n^4)$ khi tính toán điểm số antecedents.

Tiếp theo, mô hình tính các vector biểu diễn bắt đầu và kết thúc của mỗi nhúng token từ bộ mã hoá cho hàm tính điểm antecedent s_a:

$$a^s = GeLU(W_a^s.x)$$

$$a^e = GeLU(W_a^e.x)$$

Sau đó, chúng ta tính điểm số antecedent được định nghĩa bằng tổng bốn hàm song tuyến tính:

$$s_a(c, q) = a_{c_s}^s.B_a^{ss}.a_{q_s}^s + a_{c_s}^s.B_a^{se}.a_{q_e}^e + a_{c_e}^e.B_a^{es}.a_{q_s}^s + a_{c_e}^e.B_a^{ee}.a_{q_e}^e$$

Theo Kirstain et al.2021[4], mỗi thành phần đo lường khả năng tương thích (compatibility) của các spans c và q bằng sự tương tác giữa các token đầu mút của hai span này với nhau. Thành phần đầu tiên so sánh các biểu diễn đầu của c và q, trong khi thành phần thứ tư so sánh các biểu diễn cuối. Thành phần thứ hai và thứ ba so sánh chéo token bắt đầu của span c với token kết thúc của span q và ngược

lại.

So với mô hình e2e, s2e có nhiều điểm tốt hơn. Đầu tiên, s2e không cần biểu diễn span hay cặp span. Hơn nữa, mô hình cũng không sử dụng bất kì đặc trưng thủ công hay siêu dữ liệu (metadata) nào, không cắt bỏ các antecedents và chỉ cắt bỏ đề cập dựa trên điểm số đề cập $s_m(q)$.

2.7 Mạng đồ thị dựa trên cơ chế chú ý

Ở phần này, em sẽ trình bày lại kiến trúc mạng đồ thị dựa trên cơ chế chú ý (GAT[47]) để lan truyền thông điệp.

Cụ thể, đầu vào cho lớp GAT là một tập hợp các vector nhúng[4] của mọi node trong đồ thị, $h = \{\vec{h_1}, \vec{h_2}, ..., \vec{h_N}\}, \vec{h_i} \in R^F$, trong đó N là số node và F là chiều dài của mỗi vector nhúng. Sau khi lan truyền và tổng hợp thông điệp, lớp GAT sẽ sinh ra một tập hợp các vector nhúng mới là $h' = \{\vec{h_1'}, \vec{h_2'}, ..., \vec{h_n'}\}, \vec{h_i'} \in R^{F'}$.

Để làm được điều này, đầu tiên ta sẽ áp dụng cơ chế self-attention cho mỗi node trong đồ thị bằng cách sử dụng các phép biến đổi tuyến tính và hàm LeakyReLU σ:

$$e_{ij} = \sigma(\vec{a}^T(W\vec{h_i}, W\vec{h_j}))$$

Hệ số chú ý e_{ij} cho biết tầm quan trọng của đặc trưng của node j đối với node i. Ngoài ra, để tránh mất đi thông tin cấu trúc của đồ thị, các node sẽ chỉ chú ý đến các node lân cận N_i bằng cách chỉ tính e_{ij} trên tập lân cận này. Để so sánh các hệ số giữa các node lân cận, ta chuẩn hóa chúng bằng cách sử dụng hàm softmax:

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\Sigma_{k \in N_i} exp(e_{ik})}$$

Sau đó, ta sẽ tổng hợp thông điệp của mọi node từ các node lân cận bằng cách lấy trung bình các nhúng lân cận dựa trên hệ số chú ý đã được chuẩn hoá này, với σ có thể là hàm phi tuyến tính bất kì, và $\vec{h_i'}$ là nhúng đầu ra của mạng:

$$\vec{h_i'} = \sigma(\Sigma_{j \in N_i} \alpha_{ij} W\vec{h_j})(1)$$

Theo Veličković et al.[47], để ổn định quá trình học của self-attention, việc mở rộng cơ chế thành multi-head là có lợi. Cụ thể, K cơ chế chú ý sẽ độc lập thực hiện phép biến đổi ở trên, sau đó các nhúng đầu ra của chúng được nối lại với nhau thành vector cuối cùng có độ dài là KF:

[4]Trong paper gọi là feature, em xin dùng thuật ngữ thay thế là vector nhúng (embedding) để phù hợp với chương 3.

$$\vec{h_i'} = ||_{k=1}^{K}\sigma(\Sigma_{j\in N_i}\alpha_{ij}^k W^k \vec{h_j})$$

trong đó || đại diện cho phép nối, α_{ij}^k là hệ số chú ý đã được chuẩn hóa được tính bởi cơ chế chú ý thứ k và W^k là ma trận trọng số của phép biến đổi tuyến tính đầu vào tương ứng. Ta có thể thu gọn toàn bộ công thức lại như sau:

$$h_i = GAT(h_i, h_j | h_j \in N_i)$$

Kiến trúc này sẽ được sử dụng để tích hợp thông tin cú pháp và ngữ nghĩa cho mô hình ở chương ba. Phần tiếp theo em sẽ giới thiệu về bộ mã hoá SpanBERT và các chiến lược phân đoạn của nó.

2.8 Bộ mã hoá SpanBERT và các chiến lược phân đoạn

SpanBERT (Joshi et al.2020[1]) là một mô hình tiền huấn luyện tự giám sát được đào tạo bằng cách dự đoán các spans. Mô hình này được lấy cảm hứng từ BERT nhưng khác với kiến trúc của BERT theo nhiều cách. Trong phần này, em sẽ tập trung đến vấn đề phân đoạn với bộ mã hoá SpanBERT trong các mô hình đồng tham chiếu.

2.8.1 Các chiến lược phân đoạn

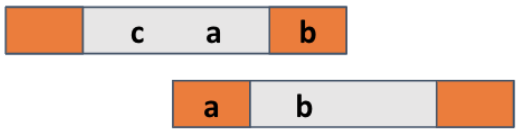
Bởi vì SpanBERT chỉ giới hạn tối đa đầu vào là 384 token hoặc 512 token nên ta cần phải chia văn bản ra thành các phân đoạn để có thể nạp vào mô hình. Có hai cách chia là chia thành các phân đoạn độc lập và chia thành các phân đoạn giao nhau. Cách chia đầu tiên có nhược điểm là có khả năng modeling hạn chế vì token chỉ có thể chú ý đến các token khác trong cùng phân đoạn và khiến các token ở hai biên của từng phân đoạn [22] mất đi nhiều ngữ cảnh quan trọng. Đối với cách chia thứ hai, Joshi và các cộng sự [22] đề xuất cơ chế sliding window với phân đoạn kích thước T và bước nhảy T/2. Cụ thể mỗi token sẽ học được 2 nhúng $r1, r2$ từ 2 phân đoạn, sau đó dùng cơ chế cổng (gating mechanism) để học cách tích hợp 2 nhúng này với nhau một cách linh hoạt:

$$f = \sigma(w^T(r1; r2)) \quad r = f.r1 + (1 - f).r2$$

Từ đó mỗi nhúng token sẽ có ngữ cảnh dài hơn. Tuy nhiên, theo kết quả thử nghiệm mà Joshi công bố, cách này không có kết quả tốt hơn so với cách chia đầu tiên trong mô hình đồng tham chiếu.

Wu et al. 2020[43], Jiang and Cohn 2022[26] dùng cách chia thứ hai nhưng masking xấp xỉ $1/4$ phần ở đầu và ở cuối mỗi phân đoạn. Điều này giúp mỗi token

chỉ học một nhúng và nhúng này sẽ có nhiều ngữ cảnh nhất.

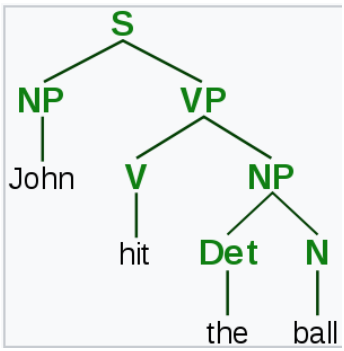


Hình 2.3: Ví dụ về chiến lược sliding window che đi $1/4$ phần ở đầu và ở cuối mỗi phân đoạn. Từ đó, token b sẽ được mã hoá thành nhúng có nhiều ngữ cảnh hơn.

2.9 Cây cú pháp

2.9.1 Cây cú pháp thành phần

Cây cú pháp thành phần (constituent tree) bao gồm nhiều thành phần (constituents), trong đó, thành phần ở nút cha sẽ chứa thành phần ở nút con, và mỗi thành phần sẽ có một nhãn tương ứng. Trong đó các node không phải node lá (non-terminal node) có thể chứa từ loại là cụm danh từ (NP), cụm động từ (VP), cụm giới từ (PP). Các node lá (terminal node) sẽ là chứa các từ loại của từng từ như danh từ (N), động từ (V), tính từ (ADJ),..v.. Dưới đây là một ví dụ:

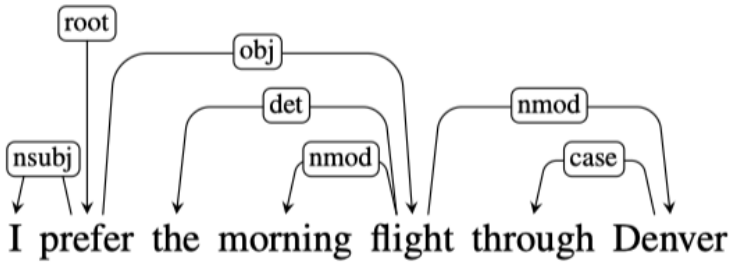


Hình 2.4: Example of constituent tree from Wikipedia[48]

trong đó *John* là danh từ (NP), *hit the ball* là cụm động từ (NP). Cây cú pháp thành phần có thể ứng dụng trong việc phân tích cấu trúc ngữ pháp của câu[49] hay trích xuất các cụm từ trong câu[50].

2.9.2 Cây cú pháp phụ thuộc

Cây cú pháp phụ thuộc (dependency tree) biểu diễn mối quan hệ giữa các từ trong một câu với nhau. Trong đó: (i) head - đầu không có mũi tên - là từ được bổ nghĩa, (ii) dependent - đầu có mũi tên - là từ bổ nghĩa, và (iii) label: Là nhãn quan hệ phụ thuộc giữa hai từ này. Dưới đây là một ví dụ:



Hình 2.5: Dependency tree example (Speech and language processing book [51])

Cây cú pháp phụ thuộc và thành phần giúp chúng ta hiểu cấu trúc ngữ pháp của câu, nhưng để hiểu ý nghĩa hay vai trò của các cụm từ trong câu, ta cần quan tâm đến một tác vụ khác.

2.10 Semantic role labeling

Semantic role labeling (SRL) là quá trình mà gán nhãn cho từ hay cụm từ trong câu mà thể hiện một vai trò ngữ nghĩa nào đó, như là tác nhân (AGENT), bệnh nhân (PATIENT) hay mục tiêu (RESULT). Để thực hiện quá trình gán nhãn, đầu tiên SRL sẽ xác định (identify) predicate của câu, sau đó sẽ xác định và phân loại những argument xoay xung quanh predicate đó[52]. Predicate thường sẽ là động từ chính trong câu. Dưới đây là một ví dụ:

"John broke the window with a rock"

trong đó động từ chính *broke* sẽ là predicate. Các argument *John*, *the window*, *a rock* sẽ lần lượt có vai trò là AGENT, THEME, INSTRUMENT.

SRL giúp máy tính hiểu rõ hơn về ý nghĩa và vai trò của các thực thể khác nhau chẳng hạn như tác nhân, bệnh nhân và địa điểm, trong việc diễn đạt một hành động hoặc sự kiện thông qua một câu [52]. Từ đó đóng một vai trò quan trọng giúp máy tính phân tích và hiểu sâu văn bản hơn.

2.11 Kết chương

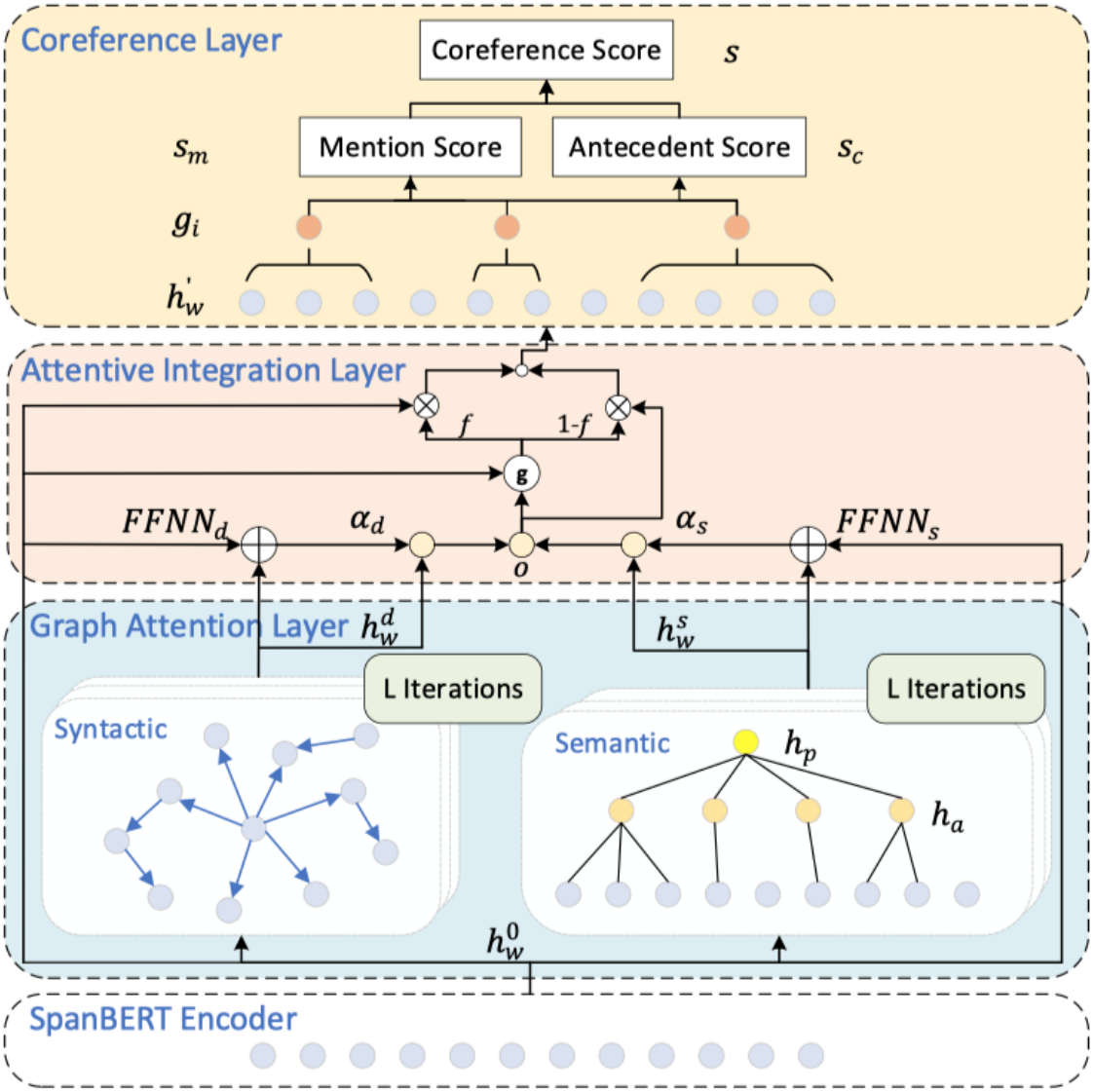
Trong chương này em đã trình bày các khái niệm và các thể loại đồng tham chiếu. Ngoài ra, chương cũng đề cập đến các nghiên cứu liên quan, cách các nghiên cứu này ứng dụng cây cú pháp và thông tin ngữ nghĩa để giải quyết các vấn đề khác nhau của bài toán đồng tham chiếu. Cuối cùng, chương trình bày các kiến trúc đồng tham chiếu, mạng đồ thị chú ý cũng như bộ mã hoá SpanBERT. Chi tiết cách áp dụng những lý thuyết này sẽ được trình bày trong chương tiếp theo.

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

3.1 Tổng quan các giải pháp

Trong chương này, em sẽ giới thiệu hai phương pháp tích hợp thông tin vào biểu diễn token với mạng đồ thị chú ý (GAT[47]) để nâng cao hiệu năng các mô hình baseline. Phương pháp đầu tiên sẽ tích hợp thông tin cú pháp phụ thuộc và nhãn vai trò ngữ nghĩa dựa trên đồ thị hỗn hợp. Phương pháp thứ hai sẽ tích hợp thông tin từ cây cú pháp thành phần (constituent syntax) với mạng GAT hai chiều và cơ chế lan truyền thông tin lân cận bậc cao hơn.

3.2 Tích hợp thông tin cú pháp phụ thuộc và ngữ nghĩa để tăng cường nhúng token (Jiang và Cohn 2021 [27])



Hình 3.1: Kiến trúc của mô hình đồng tham chiếu tích hợp thông tin cú pháp phụ thuộc và ngữ nghĩa [27]

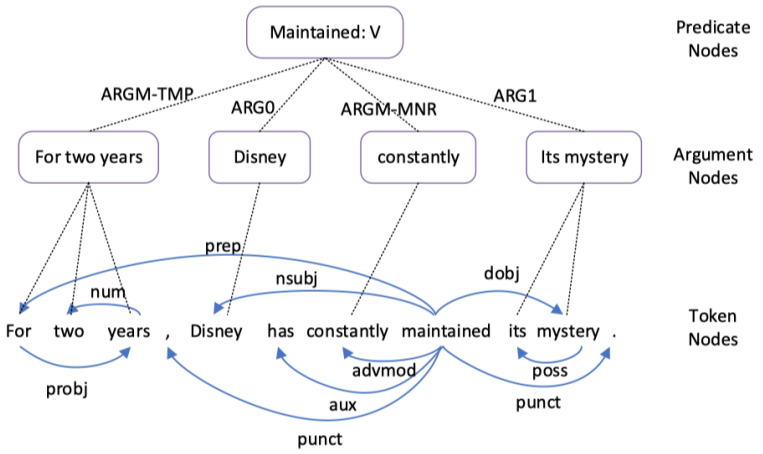
Hình 1 cho ta kiến trúc của mô hình gồm bốn mô-đun. Trong đó các mô-đun chính của phương án được biểu thị bằng nền màu xanh da trời và màu da cam. Mô-đun 4 sẽ giống với baseline c2f của Lee et al. (2018) [3] (2.5) nhưng không dùng kĩ thuật HOI - tinh lọc lại các Span - như đề xuất của Xu và Choi (2020) [20]. Ngoài ra, mô hình sử dụng SpanBERT với các phân đoạn (segment) độc lập làm bộ mã hoá như Joshi et al (2020) [1]. Tiếp theo, em sẽ trình bày cách đồ thị được xây dựng và cách lan truyền thông điệp trong đồ thị.

3.2.1 Tầng đồ thị dựa trên cơ chế chú ý

a, Xây dựng các đỉnh

Có ba loại node trong đồ thị hỗn hợp: token node (T), argument node (A) và predicate node (P). Véc tơ biểu diễn các token nodes và các predicate nodes là các nhúng được ngữ cảnh hóa từ bộ mã hoá SpanBERT, được ký hiệu lần lượt là h_w và h_p. Véc tơ biểu diễn của argument node được tính bằng cách lấy trung bình các véc tơ nhúng của những token mà nó chứa, được ký hiệu là h_a .

b, Xây dựng cạnh



Hình 3.2: Đồ thị hỗn hợp dựa trên cấu trúc cú pháp và ngữ nghĩa [27]

Mô hình sử dụng các nhãn phụ thuộc (dep) và nhãn vai trò ngữ nghĩa (SRL) để xây dựng các cạnh. Điều này được mô tả trong ví dụ ở hình 2.

Token-Token

Các cạnh được xây dựng theo cấu trúc cây phụ thuộc. Cụ thể, giữa hai token node bắt đầu từ head token đến dependent token sẽ có một cạnh có nhãn và có hướng. Một cạnh self-loop với nhãn cyclic cũng được thêm vào mỗi node trong đồ thị để cho phép các token trong một từ có thể lan truyền thông điệp với nhau. Bên cạnh đó, mô hình cũng liên kết các node gốc (root) của hai câu liền kề để cho phép thông tin lan truyền từ một câu đến các câu kề nó.

Token-argument

Các argument node được liên kết với các token node mà chúng chứa bằng một cạnh hai chiều nhưng không được gãn nhãn. Điều này cho phép thông tin từ các token nodes tăng cường các argument node và từ argument node sẽ truyền thông tin ngữ nghĩa trở lại các token.

Predicate-argument

Các cạnh là các nhãn SRL tương ứng sẽ kết nối các argument node với các predicate node. Các cạnh cho phép truyền thông tin hai chiều. Do đó, mỗi argument node có thể tổng hợp thông tin từ các argument node khác mà có chung predicate node.

Từ đây ta thấy đồ thị được tạo sẽ gồm 2 đồ thị con: Đồ thị con cú pháp (syntatic subgraph) gồm các cạnh nối token-token và đồ thị con ngữ nghĩa (semantic subgraph) gồm các loại cạnh còn còn lại.

c, Mạng đồ thị dựa trên cơ chế chú ý

Mô hình sử dụng mạng đồ thị chú ý GAT để truyền thông tin cú pháp và ngữ nghĩa đến các token nodes cơ bản. Đối với một node i, cơ chế chú ý cho phép nó kết hợp có chọn lọc thông tin từ các node lân cận của nó:

$$\alpha_{ij} = softmax(\sigma(a^T[Wh_i; Wh_j; e_{ij}]))$$
$$h'_i = ||_{k=1}^{K} ReLU(\Sigma_j \alpha_{ij}^k W^k h_j)$$

Trong đó h_i và h_j là các nhúng vector của node i và j, a^T, W và W^K là các tham số huấn luyện. e_{ij} là nhúng vector của các loại nhãn cạnh giữa node i và j dựa trên cấu trúc đồ thị, σ là hàm kích hoạt LeakyReLU. $||$ và $[;]$ thể hiện cho phép nối. Hai công thức trên được gộp lại trong 1 hàm $h_i = GAT(h_i, h_j)$, trong đó h_i và h_j là embeddings của node đích và node lân cận và h'_i là embedding được cập nhật của node đích.

Phiên bản GAT ở trên có sự khác biệt với phiên bản GAT cơ bản đã được trình bày trong chương hai khi có thêm thông tin nhãn cạnh DEP và SRL đóng góp vào quá trình lan truyền. Phần tiếp theo sẽ dùng cả 2 phiên bản GAT để cập nhật các node: Trong đó phiên bản trên cho các cạnh có nhãn, và phiên bản gốc cho các cạnh không nhãn.

d, Lan truyền thông điệp

Để tăng cường thông tin cho mỗi node embedding trong đồ thị cú pháp phụ thuộc, mô hình cập nhật tất cả các node trong đồ thị nhiều vòng thông qua cạnh truyền thông tin. Đầu tiên, mô hình sẽ cập nhật các token node từ các token node

hàng xóm được kết nối thông qua cạnh cú pháp phụ thuộc:

$$h_w^l = GAT(h_w^{l-1}, h_w^{l-1})$$

trong đó h_w^{l-1} là biểu diễn véc tơ của token trong vòng lặp trước $l-1$, h_w^l là biểu diễn véctơ được cập nhật trong vòng lặp hiện tại l và h_w^0 là nhúng token từ tầng SpanBERT.

Đối với đồ thị con ngữ nghĩa, mô hình cập nhật các argument node bằng cách sử dụng véc tơ biểu diễn của token node; rồi các argument nodes được sử dụng để cập nhật các predicates; sau đó, các predicate node được cập nhật sẽ truyền thông tin trở lại các argument nodes được kết nối của chúng; cuối cùng, các argument nodes được cập nhật sẽ phân phối thông điệp lại cho tất cả các token nodes cơ bản mà nó chứa:

$$h_a^l = GAT(h_a^{l-1}, h_w^{l-1}) \tag{3.1}$$
$$h_p^l = GAT(h_p^{l-1}, h_a^l) \tag{3.2}$$
$$h_a^l = GAT(h_a^l, h_p^l) \tag{3.3}$$
$$h_w^l = GAT(h_w^{l-1}, h_a^l) \tag{3.4}$$

Sau L vòng lặp, với mỗi token, ta có được véc tơ biểu diễn cuối cùng từ đồ thị con cú pháp và véc tơ biểu diễn cuối cùng từ đồ thị con ngữ nghĩa. Cả hai được ký hiệu lần lượt là h_w^d và h_w^s. Tầng tiếp theo sẽ học cách tích hợp 2 thông tin này vào vector biểu diễn ban đầu của nó.

3.2.2 Tầng tích hợp dựa trên cơ chế chú ý

Vì các cơ chế chú ý có hiệu quả trong việc lựa chọn thông tin phù hợp nhất (Nie et al. 2020a,b [53], Lee et al. 2017 [2]), mô hình sử dụng lớp tích hợp chú ý để kết hợp có chọn lọc thông tin cú pháp và ngữ nghĩa. Đối với mỗi loại thông tin $h_w^c \in \{h_d^w, h_s^w\}$, mô hình nối nó với biểu diễn token ban đầu h_w^0, truyền qua tầng FFNN, chuẩn hoá nó bằng softmax ta được hệ số chú ý (attention weight) α_c:

$$\alpha_c = softmax(FFNN_c([h_w^0; h_w^c])$$

trong đó với mỗi loại thông tin c (Dep hoặc SRL) sẽ có một lớp mạng chuyển tiếp riêng $FFNN_c$ có hàm kích hoạt là sigmoid. Sau khi có được thông tin hệ số chú ý hợp lệ bằng hàm softmax, chúng ta có thể tính tổng trung bình có trọng số

của cả thông tin cú pháp và ngữ nghĩa:

$$o = \Sigma_{c \in \{d,s\}} \alpha_c h_w^c$$

Vì thông tin cú pháp và ngữ nghĩa tăng cường cho token chưa chắc có ích cho phân giải đồng tham chiếu, nên cơ chế gate được sử dụng để học cách tăng cường thông tin đó một cách linh hoạt:

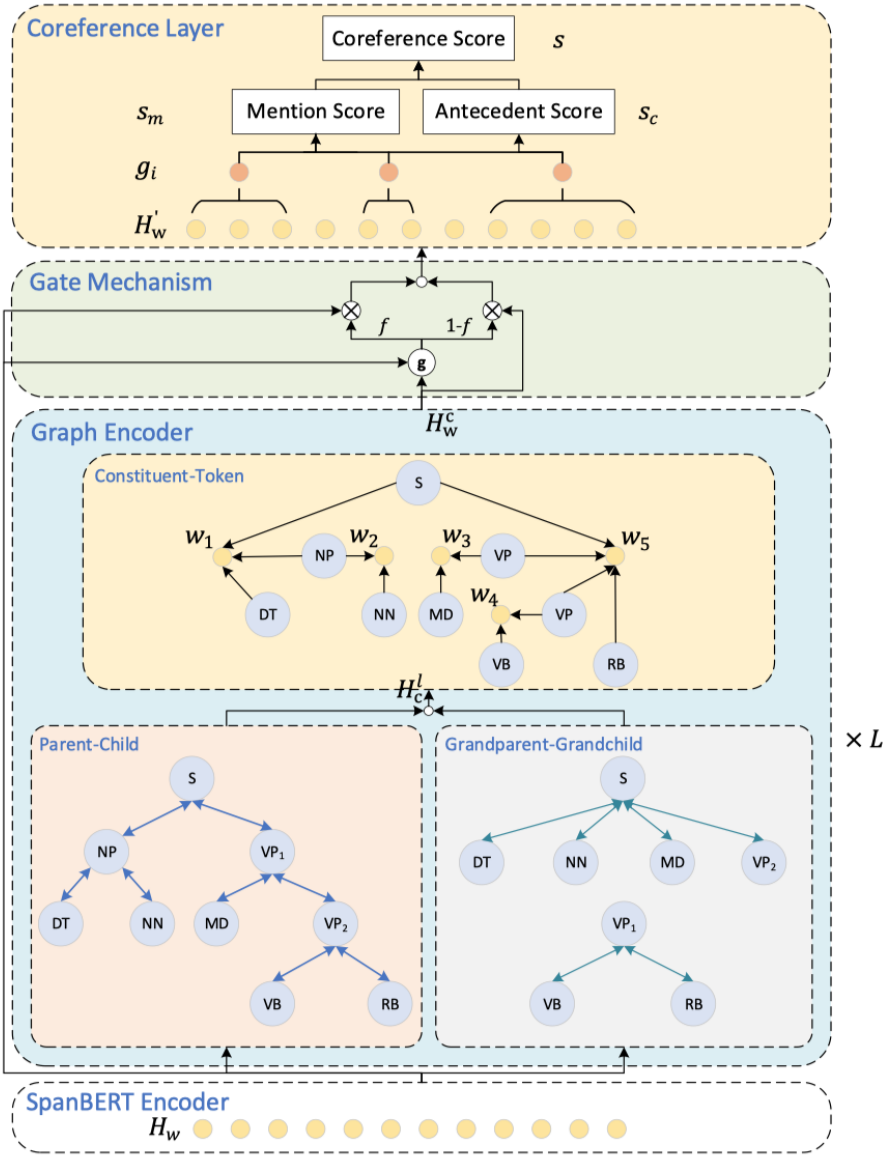
$$f = \sigma(W_g.[h_w^0; o] + b_g)$$
$$h'_w = f \odot h_w^0 + (1 - f) \odot o$$

trong đó W_g và b_g là các tham số có thể huấn luyện, $\odot$ biểu thị phép nhân theo từng phần tử và σ là hàm sigmoid logistic.

Cuối cùng, các véc tơ biểu diễn tăng cường h'_w của từng token sẽ được truyền vào cho tầng số 4 - tầng phân giải đồng tham chiếu c2f (2.5).

Phần tiếp theo sẽ trình bày một giải pháp tăng cường biểu diễn token trên cây cú pháp thành phần với mạng GAT hai chiều và cơ chế lan truyền thông tin lân cận bậc cao hơn (higher-order neighborhood information).

3.3 Tích hợp thông tin cú pháp thành phần để tăng cường nhúng token (Jiang and Cohn 2022[26])



Hình 3.3: Kiến trúc của mô hình đồng tham chiếu tích hợp thông tin cú pháp từ cây thành phần[26])

.

3.3.1 Bộ mã hóa tài liệu

Khác với cách chia tài liệu thành các phân đoạn độc lập như (joshi et al. 2020[1]), mô hình chọn chia tài liệu thành các phân đoạn (segment) giao nhau (như phần 2.8) và xử lý thông tin người nói (speaker) như một phần của của đầu vào (như phần a). Cụ thể thông tin người nói sẽ được chèn vào đầu lời phát ngôn mỗi khi thay đổi. Điều này giúp thông tin speaker phù hợp với mọi mô hình[1]. Sau đó phân đoạn đính kèm thông tin người nói sẽ được mã hoá bởi SpanBERT để có được các biểu diễn

[1]Eg. s2e model2.6

theo ngữ cảnh, được biểu thị là $H_w = (h_1, h_2, ..., h_n)$, trong đó h_i thuộc R^d và n là chiều dài của tài liệu.

3.3.2 Xây dựng đồ thị

Đối với mỗi câu trong tài liệu, chúng ta có một cây cú pháp thành phần tương ứng bao gồm các từ (terminals) và các thành phần (non-terminals). Vì vậy, đồ thị thiết kế ra hai loại nút: các nút token ($W = \{w_1, w_2, ..., w_n\}$) và các nút thành phần (constituent nodes)($C = \{c_1, c_2, ..., c_m\}$), trong đó n và m lần lượt là số lượng các token và nút thành phần. Đối với một nút thành phần nhất định c_i, $START(c_i)$ và $END(c_i)$ là các kí hiệu để biểu thị các chỉ số token bắt đầu và kết thúc của nó.

a) Khởi tạo các nút

Các biểu diễn nút token được khởi tạo bằng cách sử dụng các biểu diễn token theo ngữ cảnh H_w từ bộ mã hóa tài liệu. Thông tin hai đầu mút của span đã được chứng minh là cực kỳ quan trọng đối với các tác vụ dựa trên span (span-based tasks)[26]. Do đó, các node thành phần $c_i \in C$ có thể được khởi tạo như sau :

$$h_{c_i} = [h_{START(c_i)}; h_{END(c_i)}; e_{type}(c_i)] \quad (5)$$

trong đó $h_{START(c_i)}$ và $h_{END(c_i)}$ là véc tơ nhúng của token bắt đầu và kết thúc của thành phần c_i. $e_{type(c_i)}$ là các nhúng biểu thị kiểu của consituent được khởi tạo ngẫu nhiên từ bảng tra cứu (look-up table), bảng này sẽ được tối ưu hóa trong quá trình huấn luyện. Vì vậy, chúng ta có thể thu được một tập hợp các biểu diễn nút thành phần được khởi tạo: $H_c = \{h_{c_1}, h_{c_2}, ..., h_{c_m}\}$.

b) Xây dựng các cạnh

Constituent-Constituent

Trong đồ thị, có hai loại cạnh được thiết kế cho các đỉnh constituent - là loại cạnh *cha-con* và *ông-cháu* - với mục đích nắm bắt (capture) các phụ thuộc ở phạm vi dài hơn (longer-range dependencies). Cụ thể, với mỗi cạnh sẽ có các cạnh đối ứng (reciprocal edges) và chúng được gán nhãn với các kiểu *tiến (forward)* và *lùi (backward)* tương ứng. Ngoài ra, mỗi nút sẽ có thêm cạnh vòng và hai gốc của hai câu kề nhau cũng được nối với nhau. Do đó, thông tin có thể lan truyền giữa các câu thông qua gốc của chúng.

Tổng kết lại, các cạnh được xây dựng như dựa trên các quy tắc sau:

1. Một cặp cạnh *cha-con* và cạnh *con-cha* giữa nút c_i và c_j được xây dựng nếu chúng được kết nối trực tiếp trong cây cú pháp thành phần.
2. Một cặp cạnh *ông-cháu* và *cháu-ông* giữa nút c_i và c_j được xây dựng nếu nút

c_i có thể đến nút c_j bằng hai bước nhảy và ngược lại.

3. Một cặp cạnh $root_i - root_{i-1}$ và $root_{i-1} - root_i$ được xây dựng với mọi câu i, trong đó $root_i$ là gốc của cây cú pháp của câu thứ i

Constituent-Token

Một nút token w_i được liên kết với c_j nếu nó là token ngoài cùng bên trái hoặc ngoài cùng bên phải trong các token của c_j (yield of c_j). Các cạnh này được tạo một chiều để đảm bảo rằng thông tin chỉ có thể được truyền từ các nút constituent đến các nút token, nhằm mục đích tăng cường các nhúng token cơ bản với thông tin hai đầu mút của span (span boundary information) và cấu trúc cú pháp thành phần.

Cạnh (nhãn cạnh)	Số chiều
cha – con (forward), con – cha (backward)	Hai chiều
ông – cháu (forward), cháu – ông (backward)	Hai chiều
constituent–>token	Một chiều
$root_{i-1} - root_i$ (forward), $root_i - root_{i-1}$ (backward)	Hai chiều
cạnh vòng	

Bảng 3.1: Bảng phân loại các cạnh theo số chiều

3.3.3 Bộ mã hóa đồ thị với lớp GAT hai chiều

Mạng đồ thị chú ý sẽ được sử dụng để cập nhật biểu diễn các nút constituent và lan truyền thông tin cú pháp đến các nút token cơ bản. Phiên bản được sử dụng là phiên bản GAT cơ bản đã được giới thiệu trong chương 2 phần 2.7. Khác với GAT ở phần trước, GAT chỉ dùng các nhãn cạnh để tổ chức và phân loại chứ không dùng nó như là một thông tin đóng góp vào quá trình lan truyền. Cụ thể, ta có công thức sau:

$$\alpha_{ij} = softmax(\sigma(a^T[Wh_i; Wh_j;]))(6)$$
$$h'_i = ||_{k=1}^{K} ReLU(\Sigma_j \alpha_{ij}^k W^k h_j)(7)$$

Lớp GAT hai chiều

Một lớp GAT hai chiều được thiết kế để mô hình hóa các cạnh constituent-constituent. Cụ thể, đối với một nút thành phần nhất định c_i, ta thu được tập các nút lân cận của nó với loại cạnh t theo *hướng tiến (forward)* và *lùi (backward)*: $N_{c_i}^{tf}$ và $N_{c_i}^{tb}$ tương ứng. Sau đó, hai bộ mã hoá GAT riêng biệt được sử dụng để thu được biểu diễn cập nhật của nút c_i theo các hướng khác nhau:

$$h_{c_i}^{tf} = GAT(h_{c_i}, h_{c_j} | c_j \in N_{c_i}^{tf})(8)$$

$$h_{c_i}^{tb} = GAT(h_{c_i}, h_{c_j} | c_j \in N_{c_i}^{tb})(9)$$

Cuối cùng, biểu diễn cập nhật của nút thành phần c_i thu được bằng cách tính tổng các biểu diễn của hai hướng: $h_{c_i}^{t} = h_{c_i}^{tf} + h_{c_i}^{tb}(10)$

Lớp kết hợp nhiều loại cạnh dựa trên cơ chế chú ý

Để tổng hợp các biểu diễn nút đã cập nhật với các loại cạnh khác nhau, mô hình sử dụng cơ chế tự chú ý tương tự như Lee et al. 2017 [2] (dùng để tính biểu diễn head của span) đã được giới thiệu ở 2.4.1:

$$\alpha_{c_i,t} = softmax(FFNN(h_{c_i}^{t})$$
$$h'_{c_i} = \Sigma_{t=1}^{T} \alpha_{c_i,t} h_{c_i}^{t}(11)$$

Trong đó h'_{c_i} là là biểu diễn được cập nhật của node c_i. T là số loại cạnh và FFNN là mạng nơ-ron feed forward hai lớp với hàm kích hoạt ReLU.

Ta có thể tóm tắt công thức 8 đến công thức 11 dưới hàm sau:

$$h'_{c_i} = Multi - BiGAT(h_{c_i}, h_{c_j} | c_j \in N_{c_i})(12)$$

Phần tiếp theo em sẽ trình bày chi tiết về cơ chế lan truyền thông điệp được sử dụng trong đồ thị.

3.3.4 Lan truyền thông điệp

Để truyền thông tin từ các nút thành phần đến các nút token cơ bản, ta sẽ sử dụng cơ chế lan truyền thông điệp với nhiều vòng. Đầu tiên, biểu diễn nút constituent được cập nhật bằng cách sử dụng công thức (12):

$$h_{c_i}^{l+1} = Multi - BiGAT(h_{c_i}^{l}, h_{c_j}^{l} | c_j \in N_{c_i})(13)$$

trong đó $h_{c_i}^{l}$ là biểu diễn nút thành phần từ vòng lặp trước l và $h_{c_i}^{0}$ là nhúng của nút thành phần khởi tạo ban đầu.

Sau đó, nút thành phần (constituent node) được cập nhật lan truyền thông tin đến các nút token thông qua cạnh constituent–token:

$$h_{i}^{l+1} = GAT(h_{i}^{l}, h_{c_j}^{l+1} | c_j \in N_i)(14)$$

Các thời điểm gọi hàm GAT	Các cạnh hoạt động truyền thông điệp
Lần gọi 1: Với nhãn forward	Cha-con, cạnh vòng, root(câu i-1) - root(câu i)
Lần gọi 2: Với nhãn backward	Cạnh con-cha, cạnh vòng, và root(câu i-1) - root (câu i)
Lần gọi 3: Với nhãn forward	Cạnh ông-cháu, root(câu i-1) - root(câu i)
Lần gọi 4: Với nhãn backward	Cạnh cháu-ông, root(câu i-1) - root(câu i)
Lần gọi 5:	Cạnh token-constituent

Bảng 3.2: Các lần gọi đến hàm GAT trong một vòng lặp

trong đó h_i^l là biểu diễn token từ lớp l và h_i^0 là nhúng token học được từ bộ mã hóa SpanBERT.

Cuối cùng, biểu diễn token cập nhật được sử dụng để xây dựng lại các biểu diễn của nút thành phần bằng việc sử dụng công thức (5), và các nhúng này sẽ được sử dụng trong vòng lặp tiếp theo. Sau L lần lặp, chúng ta có thể thu được các biểu diễn token được tăng cường cú pháp thành phần (final constituent syntax enhanced token representations), được ký hiệu là H_w^c.

Ngoài ra, cơ chế cổng (gating mechanism) được sử dụng để học tích hợp (infuse) biểu diễn token tăng cường cú pháp ở trên với biểu diễn token ban đầu một cách linh hoạt và tự động:

$$f = \sigma(W_g.[H_w; H_w^c] + b_g)(14)$$
$$H'_w = f \circ H_w + (1 - f) \circ H_w^c(15)$$

trong đó W_g và b_g là các tham số có thể huấn luyện được, $\circ$ và σ lần lượt là phép nhân phần tử và hàm sigmoid .

Các biểu diễn token tăng cường H'_w này sẽ được truyền vào tầng 4 để hình thành các nhúng của span và tính điểm đồng tham chiếu với mô hình c2f 2.4

3.4 Kết chương

Trong chương này, em đã trình bày hai phương pháp để tăng cường thông tin cho biểu diễn token dựa trên bộ mã hoá GAT: phương pháp tích hợp thông tin cú pháp phụ thuộc với ngữ nghĩa và phương pháp tích hợp thông tin cú pháp thành phần. Hai mô hình này sẽ được huấn luyện dựa trên hàm mục tiêu:

$$L = L_{cluster} + \lambda * L_{mention}$$

mà đã được trình bày chi tiết trong chương 2 phần a. Trong chương tiếp theo, em sẽ trình bày kết quả thực nghiệm của hai phương pháp này.

CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM

4.1 Tập dữ liệu OntoNotes phiên bản 5.0

Phiên bản OntoNotes 5.0 là bản phát hành cuối cùng của dự án OntoNotes (LDC[54]). Mục tiêu của dự án là chú thích một kho văn bản lớn bao gồm nhiều thể loại văn bản khác nhau (tin tức, bài báo trên tạp chí, bài phát biểu qua điện thoại, blog, nhóm tin usenet, chương trình phát sóng, chương trình trò chuyện) bằng ba ngôn ngữ (tiếng Anh, tiếng Trung và tiếng Ả Rập) với nhiều lớp ghi chú (annotation layers) gồm các thông tin cấu trúc cú pháp (syntax), thông tin ngữ nghĩa (SRL), nghĩa của từ (word sense), thực thể có tên (named entities) và cuối cùng là đồng tham chiếu (coreference).

OntoNotes phiên bản 5.0 chứa nội dung của các bản phát hành trước đó – 1.0, 2.0, 3.0, 4.0 – và thêm dữ liệu và ghi chú bổ sung từ các nguồn khác nhau. Ấn phẩm tích lũy tới 2,9 triệu từ với số lượng được hiển thị trong bảng bên dưới.

	Arabic	English	Chinese
News	300k	625k	250k
BN	n/a	200k	250k
BC	n/a	200k	250k
Web	n/a	200k	150k
Tele	n/a	120k	100k
Pivot	n/a	n/a	300

Bảng 4.1: Số lượng từ theo từng thể loại của ba ngôn ngữ tiếng anh, tiếng ả rập và tiếng trung [54]

Trong đó, bộ dữ liệu Ontnotes tiếng anh được chia thành 3 tập với tập huấn luyện (2802 tài liệu), tập tối ưu (343 tài liệu), và tập kiểm thử (700 tài liệu)

Bộ dữ liệu Ontonotes có format dạng .conll chứa biểu diễn hợp nhất của tất cả các lớp chú thích (annotation layers) trong OntoNotes dưới dạng bảng với mỗi dòng tương ứng với mỗi token, và các cột sẽ là các lớp chú thích của token đó (CoNLL-2012 Shared Task[41]). Cột cuối cùng sẽ là lớp chú thích về đồng tham chiếu. Dưới đây là một ví dụ:

> "Vandenberg (8|(0) and Rayburn (23)|8) are heroes of mind (15),"". Mr.(15Boen15) says, refereing as well to Sam (23 RayBerb, the Democratic House speaker who cooperated with President Eisenhower 23)"

```
#begin document (nw/wsj/07/wsj_0771); part 000
...
...
nw/wsj/07/wsj_0771 0   0         ``   ``  (TOP(S(S*          -    -  -  -         *       *     (ARG1*          *          *      -
nw/wsj/07/wsj_0771 0   1 Vandenberg  NNP       (NP*          -    -  -  -  (PERSON)  (ARG1*          *          *  (8|(0)
nw/wsj/07/wsj_0771 0   2        and   CC          *          -    -  -  -         *       *          *          *      -
nw/wsj/07/wsj_0771 0   3    Rayburn  NNP          *)         -    -  -  -  (PERSON)       *)         *          *  *(23)|8)
nw/wsj/07/wsj_0771 0   4        are  VBP       (VP*       be   01  1  -         *    (V*)          *          *      -
nw/wsj/07/wsj_0771 0   5     heroes  NNS   (NP(NP*)          -    -  -  -         *  (ARG2*          *          *      -
nw/wsj/07/wsj_0771 0   6         of   IN       (PP*          -    -  -  -         *       *          *          *      -
nw/wsj/07/wsj_0771 0   7       mine   NN   (NP*))))         -    -  5  -         *       *)         *          *   (15)
nw/wsj/07/wsj_0771 0   8          ,;   ,;          *          -    -  -  -         *       *          *          *      -
nw/wsj/07/wsj_0771 0   9          ,;   ,;          *)         -    -  -  -         *       *)         *          *      -
nw/wsj/07/wsj_0771 0  10        Mr.  NNP       (NP*          -    -  -  -         *       *     (ARG0*     (ARG0*     *    (15
nw/wsj/07/wsj_0771 0  11      Boren  NNP          *)         -    -  -  -  (PERSON)       *        *)         *)     *    15)
nw/wsj/07/wsj_0771 0  12       says  VBZ       (VP*       say   01  1  -         *       *      (V*)          *          *      -
nw/wsj/07/wsj_0771 0  13          ,   ,          *          -    -  -  -         *       *        *           *          *      -
nw/wsj/07/wsj_0771 0  14  referring  VBG     (S(VP*      refer   01  2  -         *       *  (ARGM-ADV*       (V*)          *      -
nw/wsj/07/wsj_0771 0  15         as   RB     (ADVP*          -    -  -  -         *       *        *  (ARGM-DIS*          *      -
nw/wsj/07/wsj_0771 0  16       well   RB          *)         -    -  -  -         *       *        *)         *          *      -
nw/wsj/07/wsj_0771 0  17         to   IN       (PP*          -    -  -  -         *       *        *     (ARG1*          *      -
nw/wsj/07/wsj_0771 0  18        Sam  NNP    (NP(NP*          -    -  -  -  (PERSON*       *        *          *          *    (23
nw/wsj/07/wsj_0771 0  19    Rayburn  NNP          *)         -    -  -  -        *)       *        *          *          *      -
nw/wsj/07/wsj_0771 0  20          ,   ,          *          -    -  -  -         *       *        *          *          *      -
nw/wsj/07/wsj_0771 0  21        the   DT    (NP(NP*          -    -  -  -         *       *        *          *     (ARG0*      -
nw/wsj/07/wsj_0771 0  22 Democratic   JJ          *          -    -  -  -    (NORP)       *        *          *          *      -
nw/wsj/07/wsj_0771 0  23      House  NNP          *          -    -  -  -     (ORG)       *        *          *          *      -
nw/wsj/07/wsj_0771 0  24    speaker   NN          *)         -    -  -  -         *       *        *          *)         *      -
nw/wsj/07/wsj_0771 0  25        who   WP (SBAR(WHNP*)         -    -  -  -         *       *        *          * (R-ARG0*)      -
nw/wsj/07/wsj_0771 0  26 cooperated  VBD     (S(VP*  cooperate   01  1  -         *       *        *          *       (V*)      -
nw/wsj/07/wsj_0771 0  27       with   IN       (PP*          -    -  -  -         *       *        *          *     (ARG1*      -
nw/wsj/07/wsj_0771 0  28  President  NNP       (NP*          -    -  -  -         *       *        *          *          *      -
nw/wsj/07/wsj_0771 0  29 Eisenhower  NNP *))))))))))         -    -  -  -  (PERSON)       *        *)         *)         *)    23)
nw/wsj/07/wsj_0771 0  30          .    .          *))        -    -  -  -         *       *        *          *          *      -

nw/wsj/07/wsj_0771 0   0         ``   ``   (TOP(S*          -    -  -  -         *       *        *          -
nw/wsj/07/wsj_0771 0   1       They  PRP      (NP*)         -    -  -  -         *  (ARG0*)        *        (8)
nw/wsj/07/wsj_0771 0   2    allowed  VBD      (VP*      allow   01  1  -         *    (V*)        *          -
nw/wsj/07/wsj_0771 0   3       this   DT     (S(NP*          -    -  -  -         *  (ARG1*     (ARG1*        (6
nw/wsj/07/wsj_0771 0   4    country   NN          *)         -    -  3  -         *       *        *)         6)
nw/wsj/07/wsj_0771 0   5         to   TO       (VP*          -    -  -  -         *       *        *          -
nw/wsj/07/wsj_0771 0   6         be   VB       (VP*       be   01  1  -         *       *      (V*)       (16)
nw/wsj/07/wsj_0771 0   7   credible   JJ (ADJP*)))))         -    -  -  -         *       *)     (ARG2*)       -
nw/wsj/07/wsj_0771 0   8          .    .          *))        -    -  -  -         *       *        *          -
#end document
```

Hình 4.1: Một điểm dữ liệu format dạng .conll trong Ontonotes 5.0[41]

4.1.1 Đồng tham chiếu trong Ontonnotes

Theo coreference guidelines [55], bộ dữ liệu OntoNotes được phân chia thành hai loại đồng tham chiếu: Identiy (IDENT) và Appositve (APPOS). Trong đó, đồng tham chiếu Identity (IDENT) là các liên kết thông thường giữa các đề cập đại từ, cụm danh từ, danh từ riêng,.v.. Còn đồng tham chiếu bổ ngữ (appositive), bao gồm cụm danh từ (noun phrases) gọi là head được sửa đổi (modified), bổ nghĩa bởi nhiều cụm danh từ liền kề gọi là thuộc tính (atrributes), được phân tách bằng dấu phẩy, dấu hai chấm hoặc dấu ngoặc đơn. Và chỉ toàn bộ cấu trúc APPOS mới được kết nối trong một chuỗi IDENT. Dưới dây là một ví dụ trong coreference guidelines [55] :

> "Richard Godown, president of the company, gave a speech today. He said..."

trong đó chuỗi APPOS bao gồm [Richard Godown; president of the company] và chuỗi IDENT bao gồm [Richard Godown, president of the company; He]

Bảng 2 và bảng 3 cho ta một số thống kê về đồng tham chiếu của bộ dữ liệu này[1]:

4.2 Tiền xử lý dữ liệu

Từ file dữ liệu .conll, mô hình sẽ trích xuất các lớp ghi chú từ cột 5 (POS), cột 6 (Parse bit), cột 10 (speaker), các cột 12:N (SRL), cột N (coreference) và tiền xử lý thành năm file json. Trong đó, ba file json đầu tiên tương ứng với các tập huấn

[1]Theo ConNLL-2012 Shared Task[41], Ontonotes không ghi chú singleton mention.

Type	Train	Development	Test	All
English/Chains	35,143	4,546	4,532	44,221
Links	120,417	14,610	15,232	150,259
Mentions	155,560	19,156	19,764	194,480

Bảng 4.2: Số lượng thực thể, liên kết và đề cập trong bộ dữ liệu OntoNotes 5.0 ngôn ngữ tiếng anh trong CoNLL-2012 Shared Task[41]

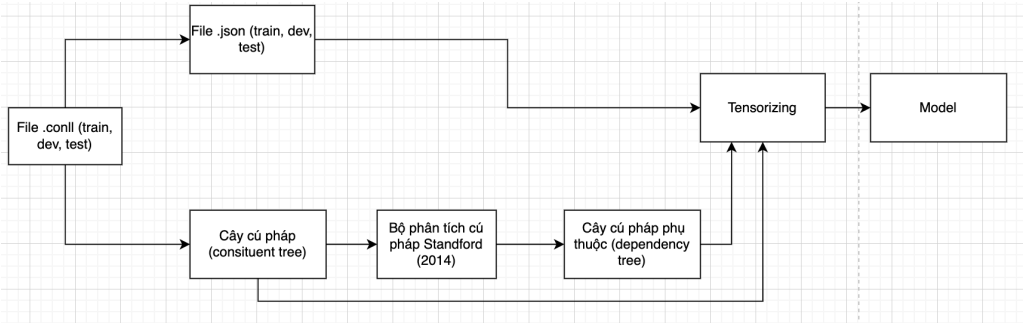
Language	Syntatic category	Train	Dev	Test
English	Noun phrase	39,46 %	45,57%	42,97%
	Pronoun	42,61%	36,66%	38,69%
	Proper Noun	11,60%	10,66%	10,96%
	Dropped Pro.	-	-	-
	Other Noun	1,68%	2,55%	2,33%
	Verb	1,61%	1,40%	1,60%
	Other	3,04%	3,16%	3,45%

Bảng 4.3: Phân bố của các loại đề cập theo syntactic trong CoNLL-2012 Shared Task[41]. Có thể thấy, bộ dữ liệu không tập trung vào dự đoán verb coreference.

luyện, tập tối ưu, và tập kiểm thử; 2 file còn lại chứa thông tin cây cú pháp thành phần và cây cú pháp phụ thuộc.

Cụ thể, mô hình sử dụng POS và Parse Bit để trích xuất cây cú pháp thành phần (constituent tree) và sau đó sử dụng bộ phân tích cú pháp của Stand Ford để sinh cây cú pháp phụ thuộc (dependency tree) từ cây cú pháp thành phần này. Đối với thông tin speaker, mô hình sẽ chèn vào đầu văn bản mỗi khi speaker thay đổi, hoặc dùng như một đặc trưng nhị phân để xem hai đề cập có cùng một speaker hay không. Thông tin SRL sẽ được tiền xử lý thành cấu trúc (p, a, l) và được dùng để tạo đồ thị ngữ nghĩa (SRL). Thông tin chú thích cột đồng tham chiếu sẽ được trích xuất ra các đề cập và nhóm lại thành các cụm khác nhau.

Ngoài ra, mỗi văn bản sẽ được chia ra thành các segment độc lập hoặc các segment giao nhau với độ dài mỗi segment là 384 như trong phần 2.8.



Hình 4.2: Quy trình tiền xử lý bộ dữ liệu OntoNotes

4.3 Các tham số đánh giá

Để đánh giá các mô hình đồng tham chiếu về mặt lý thuyết, theo Jurafsky and Martin[16] là so sánh một tập hợp các (hypothesis/predicted chains) chuỗi hoặc cụm H do hệ thống dự đoán với một tập hợp chuỗi hoặc cụm tham chiếu chân lý R, đồng thời báo cáo lại các độ đo precision và recall. Tuy nhiên, có rất nhiều phương pháp để thực hiện việc so sánh này. Cụ thể là, có năm thước đo (metrics) phổ biến được sử dụng để đánh giá các thuật toán đồng tham chiếu [16]: Thước đo MUC và BLANC dựa trên liên kết (link based), thước đo B3 dựa trên đề cập (mention based), thước đo CEAF dựa trên thực thể (entity based) và thước đo LEA nhận biết thực thể dựa trên liên kết.

Vì mỗi thước đo thể hiện các khía cạnh quan trọng khác nhau nên Denis và Baldridge (2009[56]) đề xuất thước đo MELA tính điểm trung bình 3 thước đo MUC, CEAF và B-CUBED. Do đó, các thí nghiệm sau sẽ được đo dựa trên trung bình các điểm Precision Recall F1 của ba thước đo này.

4.4 Phương pháp thí nghiệm

4.4.1 Phương pháp baseline

Đồ án sử dụng hai mô hình baseline. Mô hình baseline thứ nhất được sử dụng là mô hình c2f (phần 2.5) của Joshi et al 2020 [1] đã được cài đặt lại trên pytorch bởi Jiang và Cohn 2021 [27], 2022 [26]. Mô hình là một phiên bản dựa trên kiến trúc của c2f (Lee et al. 2018) [3], trong đó bộ mã hoá token được nâng cấp thành SpanBERT và không sử dụng kĩ thuật HOI (2.5).

Mô hình thứ baseline thứ hai là mô hình s2e của Kirstain at al [4] đã được trình bày ở phần 2.6. Mô hình được cài đăt lại và thay thế bộ mã hoá Longformer thành bộ mã hoá SpanBERT để giảm chi phí bộ nhớ.

Phần thực nghiệm trong đồ án sẽ so sánh kết quả các phương pháp tích hợp cú pháp và ngữ nghĩa với phương pháp trong nghiên cứu Jiang và Cohn 2021 [27] 2022 [26] và Kirstain et al [4]. Trong đó mỗi kịch bản được em tiến hành một lần.

4.4.2 Lựa chọn siêu tham số

Mô hình sử dụng bộ mã hoá $SpanBERT_{based}$ để mã hoá tài liệu. Mô hình được huấn luyện với giải thuật tối ưu Adam và gradient clippling l = 0,1. Do việc biểu diễn span là tốn bộ nhớ nên mỗi batch chỉ chứa 1 văn bản ([3]) và với mỗi văn bản, mô hình chỉ trích ra ngẫu nhiên 5 phân đoạn liên tiếp để huấn luyện. Bô mã hoá $SpanBERT_{based}$ được fine-tuning sử dụng tốc độ học là $2 * 10^{-5}$ và $1 * 10^{-5}$ với bộ lập lịch lịch khởi động cho 10% bước đầu tiên. Đối với các tham số liên quan đến tác vụ phân giải đồng tham chiếu, mô hình sử dụng tốc độ học $3 * 10^{-4}$ và $5 * 10^{-4}$

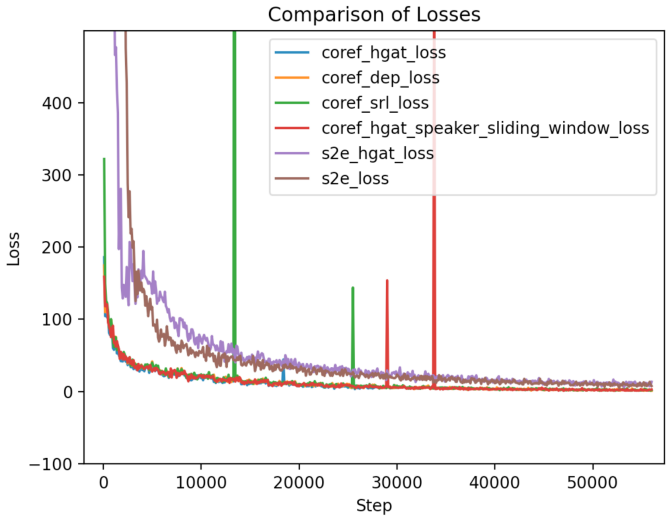
với linear decay giảm về 0. Những tham số này đã được cho là tốt nhất trong survey của Lu và Ng (2020)[57]. Ngoài ra, mô hình được huấn luyện trên T4 GPU với số epoch được mặc định là 20.

Đối với kịch bản tích hợp cú pháp và ngữ nghĩa (coref-HGAT), mô hình sử dụng bộ công cụ Stanford CoreNLP (Manning và cộng sự, 2014 [58]) để chuyển đổi các cây cú pháp thành phần (consituent tree) được chú thích thành cây phụ thuộc Stanford (de Marneffe và Manning, 2008). Các nhãn SRL được tổ chức dưới dạng bộ ba: (p, a, l), tương ứng là predicate, argument và labels. Số lượng heads và số lớp GAT của các đồ thị con cú pháp và đồ thị con SRL lần lượt tương ứng là 4 và 2. Ngoài ra kích thước nhúng các nhãn cạnh SRL và DEP là 300.

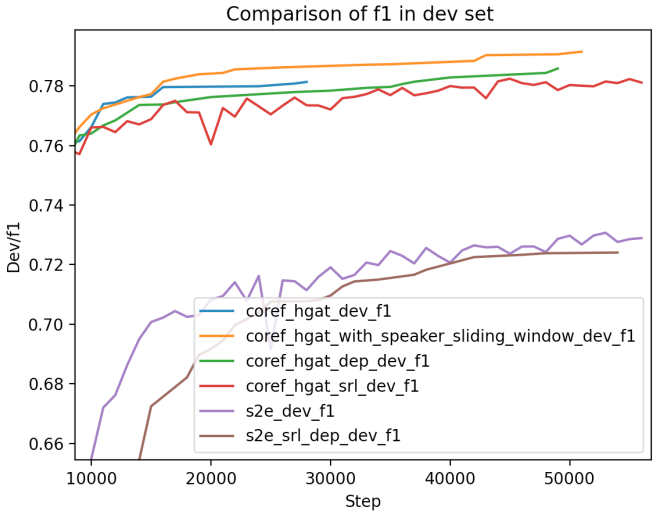
Đối với kịch bản tích hợp cây cú pháp thành phần (constituent syntax) các tham số cũng được thiết lập tương tự như kịch bản tích hợp cú pháp và ngữ nghĩa, nhưng bộ mã hoá SpanBERT sẽ dùng cơ chế sliding window (2.8) và tích hợp thông tin speaker cùng với văn bản mỗi khi speaker thay đổi (a).

4.5 Kết quả thực nghiệm trên mô hình tích hợp cú pháp và ngữ nghĩa

Dưới đây là các biểu đồ về giá trị hàm mất mát và độ đo F1 trên tập tối ưu trong quá trình huấn luyện:



Hình 4.3: Biểu đồ giá trị hàm mất mát trên tập huấn luyện trong 20 epoch



Hình 4.4: Biểu đồ giá trị điểm F1 trên tập tối ưu qua 20 epoch

Chú thích: Kí hiệu trong các hình 5.3 và hình 5.4 có ý nghĩa như sau: (i) tiền tố "coref_" và "s2e_" tương ứng với baseline thứ nhất và baseline thứ hai, (ii) "`hgat`" tương ứng với mô hình sử dụng 2 đồ thị con cú pháp và ngữ nghĩa, (ii) "dep", "srl" tướng ứng với các mô hình chỉ sử dụng một trong 2 trong hai đồ thị con, và (iii) speaker_sliding_window" tương ứng với mô hình tích hợp speaker cùng với văn bản làm đầu vào cho bộ mã hoá (a), và sử dụng sliding window cho SpanBERT (2.8)

Kịch bản thử nghiệm	Avg Precision	Avg Recall	Avg F1
Coref-HGAT + sliding window + speaker as input	79.90	78.99	79.44
Coref-HGAT(10 epoch)	78.50	78.43	78.46
Coref-HGAT(only dep)	78.94	77.86	78.39
Coref-HGAT(only SRL)	79.77	76.29	77.99
Baseline c2f (Jing and Cohn 2021 - 40 epoch)			77.5
s2e	73.70	72.45	73.06
s2e + HGAT	74.18	71.05	72.57
Baseline s2e + LongFormer (Kirstain 2021)			80.3

Bảng 4.4: Kết quả trung bình các độ đo Precision Recall F1 thực hiện trên tập test.

Kết quả thực nghiệm trên tập test của mô hình được thể hiện trong bảng 4[2]. Từ kết quả ta thấy đối với baseline thứ nhất c2f, phương pháp tích hợp speaker với sliding window vẫn tốt nhất. Ngoài ra việc tích hợp thông tin cú pháp phụ thuộc giúp tăng hiệu năng mô hình gần 1% và tích hợp thông tin SRL giúp tăng 0.4%. Trường hợp baseline thứ hai s2e, việc tích hợp thông tin cú pháp và ngữ nghĩa làm giảm hiệu năng mô hình 0.5% cho ta nhiều giả thuyết. Ngoài ra việc thay bộ mã

[2]Số epoch sẽ được mặc định là 20 nếu không chú thích thêm trong dấu ngoặc.

hoá LongFormer với SpanBERT làm hiệu năng mô hình s2e giảm đi hơn 7% cho thấy tầm quan trọng của bộ mã hoá khi $Longformer_{base}$ mã hoá tối đa 4096 token và không cần cắt tỉa hay dùng sliding window như SpanBERT.

Với mô hình tốt nhất ta có các bảng đánh giá theo thể loại văn bản và theo số lượng tokens:

Thể loại	broadcast conversation	broadcast news	magazine articles	newswire
Điểm F1	75.47	81.37	83.70	75.44
Thể loại	pivot text	telephone conversation	weblogs	
Điểm F1	87.82	76.95	75.35	

Bảng 4.5: Kết quả trung bình điểm F1 trên tập tối ưu theo thể loại văn bản

Độ dài văn bản	0-128	128-256	257-512	513-768	768-1152	>1152
Điểm F1	84.77	83.07	82.1	79.27	80.03	72.82
Số lượng văn bản	49	67	75	70	53	29

Bảng 4.6: Kết quả trung bình điểm F1 trên tập tối ưu tiếng anh được chia theo độ dài văn bản

Từ bảng 5 có thể so sánh điểm F1 trên từng thể loại văn bản khác nhau. Cụ thể, điểm F1 đạt cao nhất trên thể loại pivot text, magazine và thấp nhất trên văn bản weblogs. Còn Bảng 6 cho thấy khi tăng dần độ dài văn bản thì kết quả điểm F1 giảm dần. Điều đó chứng tỏ thông tin cú pháp và ngữ nghía của mỗi câu không đủ làm mô hình ổn định trong việc nắm bắt những phụ thuộc tầm xa (long-range dependency). Có lẽ mô hình cần thông tin ở cấp độ văn bản (document level) để nắm bắt những phụ thuộc này.

Tiếp theo là kết quả thực nghiệm với mô hình tích hợp cú pháp thành phần.

4.6 Kết quả thực nghiệm trên mô hình tích hợp cú pháp thành phần

Kịch bản thử nghiệm	Precision	Recall	F1
Coref-cons (10 epochs~24h)	79.69	77.95	78.81
Coref-HGAT (10 epochs~12h)	78.50	78.43	78.46
Baseline Coref-HGAT (Jing and Cohn 2021 - 40 epoch)			78.8
Baseline c2f (Jing and Cohn 2022 - 40 epoch)			78.1

Bảng 4.7: Kết quả trên tập test với 10 epoch

Từ bảng 7 cho ta thấy việc tích hợp cú pháp thành phần (constituent syntax) làm hiệu năng tăng thêm 0.7 so với baseline mặc dù chỉ được huấn luyện trong số lượng epoch ít hơn 4 lần[3]. Điều này chứng tỏ cây cú pháp thành phần cung cấp nhiều

[3]Vì giới hạn chi phí tính toán nên thời gian huấn luyện Coref-cons chỉ giới hạn trong 10 epochs(~1 ngày).

thông tin quan trọng cho phân giải đồng tham chiếu.

4.7 Kết chương

Chương đã trình bày về bộ dữ liệu Ontonotes phiên bản 5.0 được sử dụng để huấn luyện và đánh giá mô hình. Ngoài ra, chương cũng báo cáo về cách tiền xử lý dữ liệu, các tham số huấn luyện, tham số đánh giá và kết quả thực nghiệm của hai phương pháp tích hợp cú pháp và ngữ nghĩa. Trong chương tiếp theo em xin tổng hợp lại nội dung đã đóng góp và hướng phát triển cho đồ án trong tương lai.

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

Trong đồ án này, em đã thử nghiệm hai phương pháp tích hợp thông tin cú pháp và ngữ nghĩa dựa trên mạng đồ thị chú ý để tăng cường mô hình đồng tham chiếu. Phương pháp đầu tiên tích hợp cây cú pháp phụ thuộc và nhãn vai trò ngữ nghĩa dựa trên đồ thị hỗn hợp. Cụ thể, các node với các thể loại khác nhau sẽ lan truyền và tổng hợp thông tin đến và đi từ các hàng xóm để có những biểu diễn token được tăng cường. Phương pháp thứ hai tích hợp cây cú pháp thành phần với cơ chế lan truyền bậc cao hơn và bộ mã hoá GAT hai chiều. Kết quả so sánh trên bộ dữ liệu OntoNotes cho thấy hiệu quả của các phương pháp. Tuy nhiên, đối với mô hình baseline s2e thì hiệu năng tích hợp lại giảm, cho thấy em cần phải có nhiều thử nghiệm và cách thiết kế tích hợp hiệu quả hơn.

Hơn nữa, vì giới hạn về thời gian cũng như kinh nghiệm nên em vẫn chưa thể thử nghiệm mô hình trên nhiều bộ dữ liệu hay các thể loại đồng tham chiếu khác nhau như kì vọng trước đó. Tuy nhiên, quá trình thực hiện đồ án đã giúp em tích luỹ được nhiều kĩ năng chuyên môn như kĩ năng đọc hiểu bài báo khoa học, tra cứu thông tin và khai thác mã nguồn mở. Những kĩ năng này rất quan trọng và chắc sẽ hỗ trợ nhiều cho công việc của em trong tương lai.

5.2 Hướng phát triển trong tương lai

Trong tương lai, em sẽ cố gắng huấn luyện và kiểm thử mô hình trên các bộ dữ liệu khác với văn bản có cấu trúc phức tạp hơn. Ngoài ra, việc xây dựng mô hình đồng tham chiếu nơ-ron tương tự cho ngôn ngữ tiếng việt và tích hợp thông tin cú pháp và ngữ nghĩa vào cũng là một hướng tiếp cận có thể khai phá.

TÀI LIỆU THAM KHẢO

[1] 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*, **journvol** 8, M. Johnson, B. Roark **and** A. Nenkova, **editors**, **pages** 64–77, 2020. DOI: 10.1162/tacl_a_00300. **url**: https://aclanthology.org/2020.tacl-1.5.

[2] K. Lee, L. He, M. Lewis **and** L. Zettlemoyer, "End-to-end neural coreference resolution," **in***Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* M. Palmer, R. Hwa **and** S. Riedel, **editors**, Copenhagen, Denmark: Association for Computational Linguistics, **september** 2017, **pages** 188–197. DOI: 10.18653/v1/D17-1018. **url**: https://aclanthology.org/D17-1018.

[3] K. Lee, L. He **and** L. Zettlemoyer, "Higher-order coreference resolution with coarse-to-fine inference," **in***Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* M. Walker, H. Ji **and** A. Stent, **editors**, New Orleans, Louisiana: Association for Computational Linguistics, **june** 2018, **pages** 687–692. DOI: 10.18653/v1/N18-2108. **url**: https://aclanthology.org/N18-2108.

[4] Y. Kirstain, O. Ram **and** O. Levy, "Coreference resolution without span representations," **in***Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* C. Zong, F. Xia, W. Li **and** R. Navigli, **editors**, Online: Association for Computational Linguistics, **august** 2021, **pages** 14–19. DOI: 10.18653/v1/2021.acl-short.3. **url**: https://aclanthology.org/2021.acl-short.3.

[5] G. Kundu, A. Sil, R. Florian **and** W. Hamza, "Neural cross-lingual coreference resolution and its application to entity linking," **in***Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* I. Gurevych **and** Y. Miyao, **editors**, Melbourne, Australia: Association for Computational Linguistics, **july** 2018, **pages** 395–400. DOI: 10.18653/v1/P18-2063. **url**: https://aclanthology.org/P18-2063.

[6] Z. Dai, H. Fei **and** P. Li, "Coreference aware representation learning for neural named entity recognition," **in***Proceedings of the Twenty-Eighth International*

Joint Conference on Artificial Intelligence, IJCAI-19 International Joint Conferences on Artificial Intelligence Organization, **july** 2019, **pages** 4946–4953. DOI: 10.24963/ijcai.2019/687. **url**: https://doi.org/10.24963/ijcai.2019/687.

[7] T. Young, E. Cambria, I. Chaturvedi, M. Huang, H. Zhou **and** S. Biswas, *Augmenting end-to-end dialog systems with commonsense knowledge*, 2018. arXiv: 1709.05453 [cs.AI].

[8] A. Valdivia, M. V. Luzón, E. Cambria **and** F. Herrera, "Consensus vote models for detecting and filtering neutrality in sentiment analysis," *Information Fusion*, **jourvol** 44, **pages** 126–135, 2018, ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2018.03.007. **url**: https://www.sciencedirect.com/science/article/pii/S1566253517306590.

[9] P. Zhu, Z. Zhang, J. Li, Y. Huang **and** H. Zhao, "Lingke: A fine-grained multi-turn chatbot for customer service," **in***Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations* D. Zhao, **editor**, Santa Fe, New Mexico: Association for Computational Linguistics, **august** 2018, **pages** 108–112. **url**: https://aclanthology.org/C18-2024.

[10] V. Ng **and** C. Cardie, "Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution," **in***COLING 2002: The 19th International Conference on Computational Linguistics* 2002. **url**: https://aclanthology.org/C02-1139.

[11] E. Bengtson **and** D. Roth, "Understanding the value of features for coreference resolution," **in***Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* M. Lapata **and** H. T. Ng, **editors**, Honolulu, Hawaii: Association for Computational Linguistics, **october** 2008, **pages** 294–303. **url**: https://aclanthology.org/D08-1031.

[12] G. Durrett **and** D. Klein, "Easy victories and uphill battles in coreference resolution," **in***Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu **and** S. Bethard, **editors**, Seattle, Washington, USA: Association for Computational Linguistics, **october** 2013, **pages** 1971–1982. **url**: https://aclanthology.org/D13-1203.

[13] A. Haghighi **and** D. Klein, "Coreference resolution in a modular, entity-centered model," **in***Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* R. Kaplan, J. Burstein, M. Harper **and** G. Penn, **editors**, Los Angeles, California:

Association for Computational Linguistics, **june** 2010, **pages** 385–393. **url**: https://aclanthology.org/N10-1061.

[14] K. Clark **and** C. D. Manning, "Improving coreference resolution by learning entity-level distributed representations," **in***Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* K. Erk **and** N. A. Smith, **editors**, Berlin, Germany: Association for Computational Linguistics, **august** 2016, **pages** 643–653. DOI: 10.18653/v1/P16-1061. **url**: https://aclanthology.org/P16-1061.

[15] R. Sukthanker, S. Poria, E. Cambria **and** R. Thirunavukarasu, *Anaphora and coreference resolution: A review*, 2018. arXiv: 1805.11824 [cs.CL].

[16] D. Jurafsky **and** J. H. Martin, *Coreference Resolution*. 2023, **chapter** 26, **pages** 516–542. **url**: https://web.stanford.edu/~jurafsky/slp3/26.pdf.

[17] F. Jiang, *Towards syntax and semantics-driven neural coreference resolution*, 2022.

[18] P. Denis **and** J. Baldridge, "Specialized models and ranking for coreference resolution," **in***Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* M. Lapata **and** H. T. Ng, **editors**, Honolulu, Hawaii: Association for Computational Linguistics, **october** 2008, **pages** 660–669. **url**: https://aclanthology.org/D08-1069.

[19] H. Daumé III **and** D. Marcu, "A large-scale exploration of effective global features for a joint entity detection and tracking model," **in***Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* R. Mooney, C. Brew, L.-F. Chien **and** K. Kirchhoff, **editors**, Vancouver, British Columbia, Canada: Association for Computational Linguistics, **october** 2005, **pages** 97–104. **url**: https://aclanthology.org/H05-1013.

[20] L. Xu **and** J. D. Choi, "Revealing the myth of higher-order inference in coreference resolution," **in***Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* B. Webber, T. Cohn, Y. He **and** Y. Liu, **editors**, Online: Association for Computational Linguistics, **november** 2020, **pages** 8527–8533. DOI: 10.18653/v1/2020.emnlp-main.686. **url**: https://aclanthology.org/2020.emnlp-main.686.

[21] B. Kantor **and** A. Globerson, "Coreference resolution with entity equalization," **in***Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* A. Korhonen, D. Traum **and** L. Màrquez, **editors**, Florence, Italy:

Association for Computational Linguistics, **july** 2019, **pages** 673–677. DOI: 10.18653/v1/P19-1066. **url**: https://aclanthology.org/P19-1066.

[22] M. Joshi, O. Levy, L. Zettlemoyer **and** D. Weld, "BERT for coreference resolution: Baselines and analysis," **in***Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* K. Inui, J. Jiang, V. Ng **and** X. Wan, **editors**, Hong Kong, China: Association for Computational Linguistics, **november** 2019, **pages** 5803–5808. DOI: 10.18653/v1/D19-1588. **url**: https://aclanthology.org/D19-1588.

[23] Y. Xu **and** J. Yang, "Look again at the syntax: Relational graph convolutional network for gendered ambiguous pronoun resolution," **in***Proceedings of the First Workshop on Gender Bias in Natural Language Processing* M. R. Costa-jussà, C. Hardmeier, W. Radford **and** K. Webster, **editors**, Florence, Italy: Association for Computational Linguistics, **august** 2019, **pages** 96–101. DOI: 10.18653/v1/W19-3814. **url**: https://aclanthology.org/W19-3814.

[24] H.-L. Trieu, A.-K. Duong Nguyen, N. Nguyen, M. Miwa, H. Takamura **and** S. Ananiadou, "Coreference resolution in full text articles with BERT and syntax-based mention filtering," **in***Proceedings of the 5th Workshop on BioNLP Open Shared Tasks* K. Jin-Dong, N. Claire, B. Robert **and** D. Louise, **editors**, Hong Kong, China: Association for Computational Linguistics, **november** 2019, **pages** 196–205. DOI: 10.18653/v1/D19-5727. **url**: https://aclanthology.org/D19-5727.

[25] F. Kong **and** F. Jian, "Incorporating structural information for better coreference resolution," **in***Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19* International Joint Conferences on Artificial Intelligence Organization, **july** 2019, **pages** 5039–5045. DOI: 10.24963/ijcai.2019/700. **url**: https://doi.org/10.24963/ijcai.2019/700.

[26] F. Jiang **and** T. Cohn, *Incorporating constituent syntax for coreference resolution*, 2022. arXiv: 2202.10710 [cs.CL].

[27] F. Jiang **and** T. Cohn, "Incorporating syntax and semantics in coreference resolution with heterogeneous graph attention network," **in***Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* K. Toutanova,

A. Rumshisky, L. Zettlemoyer **andothers**, **editors**, Online: Association for Computational Linguistics, **june** 2021, **pages** 1584–1591. DOI: 10.18653/v1/2021.naacl-main.125. **url**: https://aclanthology.org/2021.naacl-main.125.

[28] D. Marcheggiani **and** I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," **in***Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* M. Palmer, R. Hwa **and** S. Riedel, **editors**, Copenhagen, Denmark: Association for Computational Linguistics, **september** 2017, **pages** 1506–1515. DOI: 10.18653/v1/D17-1159. **url**: https://aclanthology.org/D17-1159.

[29] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani **and** K. Sima'an, "Graph convolutional encoders for syntax-aware neural machine translation," **in***Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* M. Palmer, R. Hwa **and** S. Riedel, **editors**, Copenhagen, Denmark: Association for Computational Linguistics, **september** 2017, **pages** 1957–1967. DOI: 10.18653/v1/D17-1209. **url**: https://aclanthology.org/D17-1209.

[30] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov **and** M. Welling, *Modeling relational data with graph convolutional networks*, 2017. arXiv: 1703.06103 [stat.ML].

[31] K. Wang, W. Shen, Y. Yang, X. Quan **and** R. Wang, "Relational graph attention network for aspect-based sentiment analysis," **in***Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* D. Jurafsky, J. Chai, N. Schluter **and** J. Tetreault, **editors**, Online: Association for Computational Linguistics, **july** 2020, **pages** 3229–3238. DOI: 10.18653/v1/2020.acl-main.295. **url**: https://aclanthology.org/2020.acl-main.295.

[32] S. P. Ponzetto **and** M. Strube, "Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution," **in***Proceedings of the Human Language Technology Conference of the NAACL, Main Conference* R. C. Moore, J. Bilmes, J. Chu-Carroll **and** M. Sanderson, **editors**, New York City, USA: Association for Computational Linguistics, **june** 2006, **pages** 192–199. **url**: https://aclanthology.org/N06-1025.

[33] F. Kong, G. Zhou **and** Q. Zhu, "Employing the centering theory in pronoun resolution from the semantic perspective," **in***Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* P. Koehn **and** R. Mihalcea, **editors**, Singapore: Association for Computational Linguistics,

august 2009, **pages** 987–996. **url**: https://aclanthology.org/D09-1103.

[34] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou **and** X. Sun, *Measuring and relieving the over-smoothing problem for graph neural networks from the topological view*, 2019. arXiv: 1909.03211 [cs.LG].

[35] N. Ge, J. Hale **and** E. Charniak, "A statistical approach to anaphora resolution," **in***Sixth Workshop on Very Large Corpora* 1998. **url**: https://aclanthology.org/W98-1119.

[36] S. Bergsma **and** D. Lin, "Bootstrapping path-based pronoun resolution," **in***Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* N. Calzolari, C. Cardie **and** P. Isabelle, **editors**, Sydney, Australia: Association for Computational Linguistics, **july** 2006, **pages** 33–40. DOI: 10.3115/1220175.1220180. **url**: https://aclanthology.org/P06-1005.

[37] F. Kong, G. Zhou, L. Qian **and** Q. Zhu, "Dependency-driven anaphoricity determination for coreference resolution," **in***Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)* C.-R. Huang **and** D. Jurafsky, **editors**, Beijing, China: Coling 2010 Organizing Committee, **august** 2010, **pages** 599–607. **url**: https://aclanthology.org/C10-1068.

[38] D. Marcheggiani **and** I. Titov, "Graph convolutions over constituent trees for syntax-aware semantic role labeling," **in***Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* B. Webber, T. Cohn, Y. He **and** Y. Liu, **editors**, Online: Association for Computational Linguistics, **november** 2020, **pages** 3915–3928. DOI: 10.18653/v1/2020.emnlp-main.322. **url**: https://aclanthology.org/2020.emnlp-main.322.

[39] (), **url**: https://github.com/shayneobrien/coreference-resolution.

[40] R. Sukthanker, S. Poria, E. Cambria **and** R. Thirunavukarasu, *Anaphora and coreference resolution: A review*, 2018. arXiv: 1805.11824 [cs.CL].

[41] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina **and** Y. Zhang, "CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes," **in***Joint Conference on EMNLP and CoNLL - Shared Task* S. Pradhan, A. Moschitti **and** N. Xue, **editors**, Jeju Island, Korea: Association for Computational

Linguistics, **july** 2012, **pages** 1–40. **url**: https://aclanthology.org/W12-4501.

[42] K. Lee, S. Salant, T. Kwiatkowski, A. Parikh, D. Das **and** J. Berant, *Learning recurrent span representations for extractive question answering*, 2017. arXiv: 1611.01436 [cs.CL].

[43] W. Wu, F. Wang, A. Yuan, F. Wu **and** J. Li, "CorefQA: Coreference resolution as query-based span prediction," **in***Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* D. Jurafsky, J. Chai, N. Schluter **and** J. Tetreault, **editors**, Online: Association for Computational Linguistics, **july** 2020, **pages** 6953–6963. DOI: 10.18653/v1/2020.acl-main.622. **url**: https://aclanthology.org/2020.acl-main.622.

[44] R. Zhang, C. Nogueira dos Santos, M. Yasunaga, B. Xiang **and** D. Radev, "Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering," **in***Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* I. Gurevych **and** Y. Miyao, **editors**, Melbourne, Australia: Association for Computational Linguistics, **july** 2018, **pages** 102–107. DOI: 10.18653/v1/P18-2017. **url**: https://aclanthology.org/P18-2017.

[45] T. M. Lai, T. Bui **and** D. S. Kim, *End-to-end neural coreference resolution revisited: A simple yet effective baseline*, 2022. arXiv: 2107.01700 [cs.CL].

[46] T. Dozat **and** C. D. Manning, "Deep biaffine attention for neural dependency parsing," *ArXiv*, **jourvol** abs/1611.01734, 2016. **url**: https://api.semanticscholar.org/CorpusID:7942973.

[47] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò **and** Y. Bengio, *Graph attention networks*, 2018. arXiv: 1710.10903 [stat.ML].

[48] Wikipedia contributors. "Parse tree." Accessed on January 5, 2024. (2024), **url**: https://en.wikipedia.org/wiki/Parse_tree.

[49] wisdomml. "What is constituency parsing in nlp." Accessed on January 5, 2024. (2023), **url**: https://wisdomml.in/what-is-constituencyparsing-in-nlp/.

[50] F. Elia. "Constituency vs dependency parsing." Accessed on January 5, 2024. (2022), **url**: https://www.baeldung.com/cs/constituencyvs-dependency-parsing.

[51] D. Jurafsky **and** J. H. Martin, *Speech and Language Processing*. 2023. **url**: https://web.stanford.edu/~jurafsky/slp3/.

[52] H. Bhaur. "What is semantic role labeling (srl)?" ().

[53] Y. Nie, Y. Tian, Y. Song, X. Ao **and** X. Wan, "Improving named entity recognition with attentive ensemble of syntactic information," **in***Findings of the Association for Computational Linguistics: EMNLP 2020* T. Cohn, Y. He **and** Y. Liu, **editors**, Online: Association for Computational Linguistics, **november** 2020, **pages** 4231–4245. DOI: 10.18653/v1/2020.findings-emnlp.378. **url**: https://aclanthology.org/2020.findings-emnlp.378.

[54] "Ontonotes 5.0 dataset." Accessed on Date. (2012), **url**: https://catalog.ldc.upenn.edu/LDC2013T19.

[55] BBN Technologies, *Ontonotes english co-reference guidelines*, Version 7.0, BBN Technologies, 2007. **url**: https://ufal.mff.cuni.cz/pcedt3.0/pubs/english-coreference-guidelines.pdf.

[56] P. Denis **and** J. Baldridge, "Global joint models for coreference resolution and named entity classification," *Procesamiento del lenguaje natural, ISSN 1135-5948, Nº. 42, 2009, pags. 87-96*, **jourvol** 42, **january** 2009.

[57] J. Lu **and** V. Ng, "Conundrums in entity coreference resolution: Making sense of the state of the art," **in***Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* B. Webber, T. Cohn, Y. He **and** Y. Liu, **editors**, Online: Association for Computational Linguistics, **november** 2020, **pages** 6620–6631. DOI: 10.18653/v1/2020.emnlp-main.536. **url**: https://aclanthology.org/2020.emnlp-main.536.

[58] 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* K. Bontcheva **and** J. Zhu, **editors**, Baltimore, Maryland: Association for Computational Linguistics, **june** 2014, **pages** 55–60. DOI: 10.3115/v1/P14-5010. **url**: https://aclanthology.org/P14-5010.