Đại học Quốc gia Hà Nội
Đại học Công nghệ

Lê Quang Hưng

Nhận diện chéo tên thực thể y sinh sử dụng học sâu đa tác vụ

Chuyên ngành: Công nghệ thông tin

Hà Nội - 2020

Đại học Quốc gia Hà Nội
Đại học Công nghệ

Lê Quang Hưng

Nhận diện chéo tên thực thể y sinh sử dụng học sâu đa tác vụ

Chuyên ngành: Công nghệ thông tin

Giảng viên hướng dẫn: TS Đặng Thanh Hải

Hà Nội - 2020

Tóm tắt

Các mô hình nhận diện tên thực thể Y Sinh thường yêu cầu trích xuất ra các đặc trưng một cách thủ công cho từng loại thực thể như gene, tên bệnh, chất hóa học, ... để đạt kết quả tốt nhất. Mặc dù đã có một vài nghiên cứu thử nghiệm trên mạng nơ-ron để giải bài toán giúp giảm thiểu việc thủ công, kết quả của các mô hình vẫn thường giới hạn cho từng loại thực thể. Khóa luận này đưa ra mô hình học sâu đa tác vụ nhằm giải quyết bài toán này. Thử nghiệm trên 5 tập dữ liệu cho thấy kết quả có sự cải thiện so với chỉ sử dụng 1 tập dữ liệu. Có được điều này là nhờ việc học được thông tin lẫn nhau giữa các tập dữ liệu - các thông tin về kí tự và ở mức tư.

Từ khóa: Nhận diện tên thực thể, học đa tác vụ, đặc trưng ngữ pháp, BiLSTM, CRF

Mục lục

Danh sách bảng

Danh sách hình vẽ

1 Giới thiệu

Hoàn cảnh nghiên cứu

Giới thiệu về toán nhận diện tên thực thể Y Sinh

Nhận diện tên thực thể là một trong những bài toán quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên với nhiệm vụ là nhận diện được tên các thực thể trong một đoạn văn. Các tên thực thể có thể kể đến là tên người, tên địa điểm, ngày tháng, thời gian khi tiếp cận với các văn bản trong các lĩnh vực tổng quan. Trong khi đó, tên của các loại gene, protein, tên bệnh, tế bào, tên thuốc, ... là các thực thể trong các văn bản Y Sinh, hay còn gọi là tên các thực thể Y Sinh.

Ví dụ	Loại thực thể
PuB1	Gene
CD28	Protein
Interferon	Thuốc
methylmercury, docosahexaenoic acid	Chất hóa học
Haemoglobin C disease	Tên bệnh

Bảng 1.1: Ví dụ một vài tên thực thể Y Sinh và phân loại thực thể

Bảng 1.1 chỉ ra một vài tên thực thể Y Sinh và loại thực thể tương ứng.

Nhiệm vụ của bài toán nhận diện tên thực thể Y Sinh là tìm ra được tất cả những thực thể như vậy trong một đoạn văn.

Tầm quan trọng của bài toán nhận diện tên thực thể Y Sinh nằm ở việc đó là một trong các bước tiền xử lý cho việc khai phá dữ liệu hướng lĩnh vực Y Sinh. Tầm quan trọng này càng được nâng lên khi số bài nghiên cứu trong lĩnh vực Y Sinh ngày càng tăng lên nhanh chóng. Lấy ví dụ một cơ sở dữ liệu nổi tiếng chưa các thông tin về đời sống khóa học cũng như các thông tin về Y Sinh như kháng sinh, dược, chăm sóc sức khỏe, ... là MEDLINE. MEDLINE được soạn thảo với thư viện quốc gia về Y tế của Mỹ. Cơ sở dữ liệu này là miễn phí và còn có thể truy cập bởi PubMed. Vào năm 2008, số lượng bài báo đã là hơn 17 triệu từ khoảng 5000 tạp chí về Y Sinh. Đến năm 2017, con số này đã là hơn 24 triệu trích dẫn. Tên của các thực thể có thể được sử dụng để tạo ra các chú thích về ngữ nghĩa của một văn bản, xây dựng ra từ khóa đặc trưng cho việc tra cứu. Hay việc nhận diện cũng giúp xây dựng bộ dữ liệu về mối quan hệ thực thể bệnh, thuốc hoặc các triệu chứng đi kèm. Ngoài ra, một ứng dụng khác của bài toán nhận diện tên thực thể là hệ thống trả lời câu hỏi. Các tên thực thể vừa được sử dụng để xử lý truy vấn cũng như tìm kiếm câu trả lời. Hoặc với bài toán trả lời câu hỏi, các kĩ thuật nhận diện tên thực thể được sử dụng như một công cụ giúp chọn ra các câu trả lời phù hợp. Tại hội nghị TREC-8 (Text REtrieval Conference) đã chỉ là 80% các truy vấn dùng để đánh giá các hệ thống này là các câu hỏi "who", "where", "when"[39] - "ai", "ở đâu", "khi nào vốn có thể trả lời với tên người, tổ chức, địa điểm hoặc ngày tháng.

Với sự phát triển nhanh như vậy, việc luôn tiếp cận mới những kết quả nghiên cứu mới là một thách thức. Chỉ có thể bằng sự trợ giúp của các kĩ thuật khai phá dữ liệu để tiếp thu tri thức và thông tin từ các nghiên cứu đó và góp phần bổ sung cho chính nghiên cứu đang thực hiện. Với các nhà nghiên cứu lĩnh vực này, tên của các thực thể là đơn vị cơ bản nhất nhưng chưa nhiều tri thức và thông tin. Để hiểu được bài viết thì bắt buộc phải hiểu tên của các thực thể và ý nghĩa của chúng. Thêm nữa, việc nhận diện còn giúp trích xuất

mối quan hệ hay các thông tin khác bằng việc xác định nội dung chính, biểu diễn những nội dung đó một cách nhất quán và theo định dạng chuẩn.

Tuy nhiên, bài toán nhận diện tên thực thể Y Sinh là không dễ. Điều đầu tiên làm nên độ khó của bài toán là ta không có từ điển nào hoàn chỉnh cho đa phần các loại tên thực thể Y Sinh. Vì thế việc chỉ sử dụng thuật toán so khớp xâu không thể đủ để giải quyết. Đã có một vài dữ liệu được gán nhãn một cách thủ công, tuy nhiên các dữ liệu đó lại quá tập trung vào một miền con và không thể bao phủ hết các tên thực thể ở những miền con đó chứ chưa nói đến những cái tên liên tục được tạo ra và chưa kịp đưa vào bộ từ điển ngay lập tức.

Thứ hai, tên của các thực thể Y Sinh thường gây mơ hồ, hay xuất hiện từ đồng nghĩa và các tên biển thể vì không có quy tắc cố định về việc đặt tên mặc dù đã có những hướng dẫn. Cùng một từ hay một cụm từ có thể nhắc đến các loại thực thể khác dựa trên ngữ cảnh. Ví dụ "ferritin"có thể vừa là chất sinh học hoặc vừa là tên một loại thí nghiệm. Rồi nhiều thực thể có chung tên như "PTEN"hay "MMAC1"đều nhắc đến cùng 1 gene. Thông thường các tên giống nhau thường có một ít biển đổi về chính tả (NF kappa B còn có thể viết là NF-kappa B hoặc NF kappa-B). Cho dù khi các nhà nghiên cứu chỉ sử dụng những cái tên được chuẩn hóa và chấp nhận thì vẫn còn một số lượng lớn các tài liệu trước chứa tên thực thể mà cần suy luận.

Một đặc điểm khác của tên các thực thể Y Sinh là thường bao gồm nhiều từ gộp lại (ví dụ CD28 surface receptor), và những tên thực thể ngắn hơn có thể gộp lại với nhau tạo thành một cái tên dài hơn. Đặc điểm này đặt ra bài toán con rằng làm sao để xác định biên của tên và xử lý tên các thực thể lồng nhau. Nhất là khi trước khi xử lý dữ liệu thường phải đưa qua bước tách từ thì tên những thực thể như "an alpha galactosyl-1,4-beta-galactosyl-specific adhension"bị phân mảnh quá nhiều khiến cho việc nhận diện tên gặp khó khăn. Hay tên của 2 tế bào là "NB4"và "APL"đôi khi còn xuất hiện cùng nhau tạo thành "NB4-APL". Thực tế thì NB4 là dòng tế bào đi ra từ tế bào acute promyelocytic leukemia (APL). Có đến 90% tên các thực thể chứa từ 2 đến 3

từ trong bộ dữ liệu GENIA, trong khi tên các thực thể gồm 6 từ trở lên là hiếm dù đúng là chúng có tồn lại.

Trong các văn bản Y Sinh còn thường gặp các tên viết tắt. Gần một nửa các tóm tắt trên MEDLINE chưa các từ viết tắt như đã đề cập trong [3]. Trong năm 2004, có 64262 tên viết tắt được đưa ra, và trung bình có từ một từ viết tắt mới trong khoảng từ 5 đến 10 tóm tắt. Nhiều từ viết tắt như vậy nhưng chỉ mới một phần nhỏ được đưa vào từ điển để nhận biết. Việc viết tắt cũng là nguyên nhân cho sự thiếu rành mạnh và rõ ràng. Ví dụ ACE có thể là angiotensin converting enzyme nhưng cũng có thể hiểu là affinity capillary electrophresis...

Tên các thực thể Y Sinh được kì vọng là sẽ rất lớn, và các tên mới được phát minh ra hàng ngày. Vì thế, bài toán phải đảm bảo có thể mở rộng hay nói cách khác là nhận diện được đa phần tên các thực thể trong thời gian chấp nhận được. Đồng thời, bài toán còn phải phát hiện được những tên mà chưa từng thấy.

Cuối cùng, các kĩ thuật dùng cho nhận diện tên thực thể trong văn bản thường ngày không thể áp dụng trực tiếp cho bài toán nhận diện tên thực thể Y Sinh vì tên các thực thể trong miền dữ liệu có những đặc điểm rất khác với tên các thực thể thường thấy. Ví dụ tên các thực thể trong văn bản thường ngày thường là danh từ riêng với các chữ cái được viết hóa, trong khi rất nhiều tên thực thể Y Sinh không phải danh từ riêng (hơn 60% từ trong các thực thể Y Sinh là chữ cái viết thường [49]).

Nhìn chung, tên của các thực thể có thể xuất hiện ở nhiều dạng khác nhau trong các văn bản về Y Sinh. Và bài toán nhận diện tên thực thể này phải xử lý được nhiều nhất các trường hợp đó, tạo tiền đề cho các bài toán phía sau cũng dựa trên kết quả của quá trình này.

Các nghiên cứu liên quan

Các phương pháp tiếp cận bài toán thường rơi vào một trong các nhóm: dựa trên từ điển, dựa trên luật hoặc dùng mạng nơ-ron. Việc dựa vào từ điển sẽ tìm ra các tên trong một đoạn văn dựa vào một từ điển đã định nghĩa sẵn một cách thủ công hoặc tự động. Phương pháp sử dụng luật sẽ tận dụng những luật hoặc thành phần được định nghĩa từ trước để so khớp với tên thực thể. Phương pháp sử dụng mạng nơ-ron sẽ sử dụng các kĩ thuật để tạo thành mô hình dự đoán tên các thực thể đó.

- Phương pháp dựa trên từ điển

  Phương pháp này cần trước một danh sách tên các thực thể Y Sinh đã biết, thường là lấy từ các cơ sở dữ liệu. Trong miền dữ liệu Y Sinh có một vài cơ sở dữ liệu như vậy cho gene (GeneBank), protein (UniProt) hay chất hóa học (ChemIDplus). Việc tìm kiếm tên sẽ dựa trên so khớp một phần hoặc so khớp toàn phần. Thế mạnh của phương pháp này là đơn giản và hiệu quả vì thuật toán so khớp xâu đã được nghiên cứu rất nhiều trong lĩnh vực khoa học máy tính. Vấn đề lớn nhất là độ chính xác không cao, lý giải là vì các cơ sở dữ liệu bao phủ không nhiều, thường là chuyên biệt về một vài loại thực thể và thường không cập nhật thường xuyên. Lý do khác có thể do sự biến đổi về mặt chính tả khi thay đổi thứ tự từ trong một thực thể hoặc các từ viết tắt. Một vài mô hình đã sử dụng phương pháp này và đạt kết quả bước đầu như [42], [41], [10]

- Tiếp cận dựa trên luật

  Phương pháp tiếp cận này sẽ phát hiện ra tên các thực thể dựa trên luật đã được định nghĩa từ trước. Các luật thường mô tả cấu trúc tên sử dụng hình thái từ các các đặc trưng khác ví dụ như việc kết hợp từ và số, sự xuất hiện của kí tự đặc biệt, các danh từ hay tính từ lạ. Có được những luật như vậy yêu cầu phải có kiến thức sâu về miền, về ngôn ngữ và cả ngôn ngữ lập trình. Điểm mạnh của phương pháp này là các thuật được

thiết kế cẩn thận để giải quyết được đặc trưng về ngôn ngữ.

- Phương pháp tiếp cận học máy

  Việc nhận diện tên thực thể Y Sinh thường được đưa về bài toán gán nhãn chuỗi với mục tiêu là gán một nhãn cho mỗi từ trong một chuỗi. Các hệ thống nhận diện tên thực thể Y Sinh đạt kết quả tốt nhất thường yêu cầu các đặc trưng định nghĩa thủ công như đặc trưng về chữ hoa, tiền tố hay hậu tố ... Các đặc trưng này sẽ được thiết kế riêng cho từng loại thực thể. Một vài mô hình như vậy đã được đề cập trong [21], [15] hay [48]. Quá trình tạo ra các đặc trưng này chiếm phần lớn và thời gian và chi phí khi phát triển một hệ thống nhận diện tên thực thể [23], đồng thời hướng đến các hệ thống đặc thù mà không thể trực tiếp sử dụng để nhận diện các loại thực thể khác.

  Một vài nghiên cứu gần đây sử dụng mạng nơ-ron để tự đồng tạo ra các đặc trưng có giá trị như [4], [28], [20], [24]. Mô hình của Crichton lấy mỗi từ và các từ xung quanh làm ngữ cảnh thành đầu vào cho mạng nơ-ron tích chập (CNN). Habibi sử dụng mô hình giống với Lample, đồng thời thêm vào các word embedding làm đầu vào cho mạng BiLSTM-CRF. Các mô hình này giúp cho ta không phải tạo ra các đặc trưng một cách thủ công. Tuy nhiên, cũng các mô hình này yêu cầu hàng triệu tham số và cần dữ liệu rất lớn để ước lượng được các tham số đó. Điều này là thách thức với lĩnh vực Y Sinh khi những dữ liệu thô thì có nhiều nhưng những dữ liệu đã được gán nhãn thì lại rất mất công để thực hiện, đi kèm đó là tốn chi phí. Thế nên mặc dù các mạng nơ-ron cho thấy kết quả vượt trội với các phương pháp gán nhãn chuỗi truyền thống như mô hình CRF [19], các kết quả vẫn không vượt qua được những mô hình dựa trên đặc trưng thủ công.

Cấu trúc khóa luận

- Đầu tiên, khóa luận đề cập đến bài toán nhận diện tên thực thể Y sinh, bối cảnh cùng các đặc điểm và thách thức của nghiên cứu.
- Chương thứ 2 nêu ra các kiến thức nền tảng. Chương này tập trung vào các lý thuyết chung, các thành phần tạo nên mô hình để giải quyết bài toán nhận diện tên thực thể trong khóa luận bao gồm word embedding, các mạng nơ-ron đặc biệt, mô hình CRF và phương pháp học đa tác vụ.
- Chương tiếp theo mô tả phương pháp tiếp cận để giải bài toán bằng cách áp dụng việc học đa tác vụ dựa trên một mô hình đơn lẻ ban đầu nhằm cải thiện độ chính xác của bài toán.
- Chương thứ 4 ghi lại môi trường, các tham số và đặc biệt là kết quả thực nghiệm sử dụng mô hình trong khóa luận, phân tích nhận xét về kết quả.
- Kết luận đánh giá những kết quả mà qua khóa luận đạt được trong việc nghiên cứu áp dụng học sâu đa tác vụ vào bài toán chính, đồng thời đưa ra phương hướng để cải tiến trong tương lai

New sec

2 Cơ sở lý thuyết

Bài toán nhận diện tên thực thể Y-Sinh

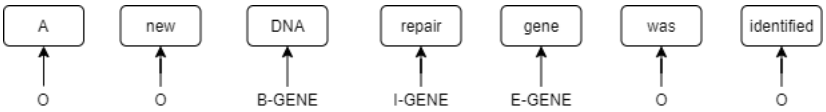
Chương này sẽ định nghĩa một cách rõ ràng hơn bài toán nhận diện tên thực thể Y Sinh. Giả sử được cho một văn bản về Y Sinh, ta giả định rằng văn bản này đã được tách từ một cách hoàn chỉnh và biên giữa các câu cũng được định nghĩa đầy đủ. Nhận thấy rằng tên thực thể thường không mở rộng sang 2 câu, ta xem bài toán nhận diện tên thực thể trong văn bản thành nhận diện tên thực thể trong câu. Bài toán chỉ tập trung những thực thể được thể hiện rõ ràng trong câu thay vì tập trung vào cả những thực thể đã được nhắc đến và xuất hiện trong câu hiện tại dưới dạng đại từ. Khi gặp những từ như thế, bài toán sẽ bỏ qua.

Với những giả sử như vậy, mỗi câu S có n từ. Chuỗi từ này được định nghĩa là $S = x_1, x_2, ..., x_n$. Về căn bản, bài toán có thể chia ra làm hai bước: Phát hiện biên của tên thực thể và phân loại tên các thực thể đó. Bước phát hiện biên của thực thể giúp xác định rằng tên thực thể này chỉ bao gồm 1 từ hay gồm nhiều từ. Nói cách khác, việc này giúp xác định tên thực thể với những từ không phải tên thực thể. Bước tiếp theo sẽ gán loại thực thể tương ứng với tên thực thể đã xác định ở bước đầu tiên. Thực thế 2 bước này có thể gộp vào 1 trong mô hình học máy.

Để giải quyết những vấn đề trên, bài toán nhận diện tên thực thể thường được đưa về bài toán gán nhãn chuỗi. Khi đó, mỗi từ trong một chuỗi sẽ được gán tương ứng với một nhãn thể hiện từ đó có là một phần trong tên của một thực thể nào không. Có tập nhãn thường được dùng là BIO hoặc BIOES. Mỗi

kí tự là một nhãn thể hiện sự:

- B (Begin): bắt đầu xuất hiện tên thực thể
- I (Inside): từ nằm trong tên của một thực thể khi đi sau B
- O (Outside): từ không nằm trong tên của một thực thể
- E (End): đánh dấu tên thực thể kết thúc tại đó
- S (Single): từ đó chính là tên đầy đủ của một thực thể



Hình 2.1: Ví dụ của việc gán nhãn chuỗi với bộ nhãn BIOES

Hình 2.1 thể hiện ví dụ của việc gán nhãn chuỗi giúp từ đó xác định tên thực thể. Việc gán nhãn như vậy sẽ vừa giúp xác định được tên của thực thể gồm những từ nào tạo nên, ví trí bắt đầu và kết thúc, cũng như loại thực thể được xác định.

Về tổng quát, có thể mô tả bài toán nhận diện tên thực thể như sau:

Cho L là tập các nhãn thể hiện một từ có thuộc về tên một thực thể hay không. Cho chuỗi các từ $w = \{w_1, w_2, ..., w_n\}$. Đầu ra của bài toán là chuỗi nhãn $y = \{y_1, y_2, ..., y_n\}$ với $y_i \in L$

Word Embedding

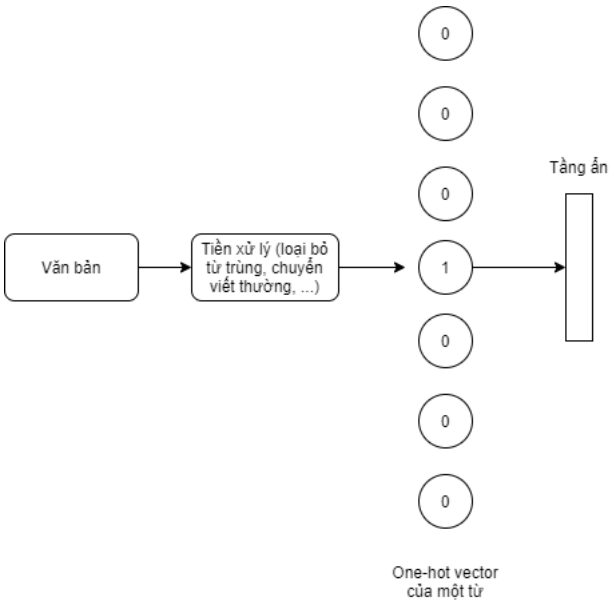
Các mô hình tập trung vào việc tính toán hay chính các mạng noron đều xử lý đầu vào là các con số. Vì vậy đặt ra yêu cầu làm sao để biểu diễn ngôn ngữ tự nhiên của con người sao cho thuận lợi nhất cho việc tính toán. Với mong muốn như vậy, word embedding (phép nhúng từ) ra đời, là phương pháp để ánh xạ mỗi từ dưới dạng toán học. Cụ thể, phép nhúng sẽ ánh xạ mỗi từ mà

ta gặp khi xử lý dữ liệu vào một không gian số thực nhiều chiều. Tuy nhiên, số chiều có kích thước nhỏ hơn nhiều so với kích thước từ điển.

Phương pháp tổng quan để xử lý các từ được biết đến là mã hóa one-hot. Cho N là số từ khác nhau xuất hiện trong văn bản. Phương pháp này sẽ gán cho mỗi từ một vector 0 có chiều là N ngoại trừ phần tử tại chỉ mục mô tả từ tương ứng trong văn bản sẽ có giá trị là 1.

Ví dụ: Xét đoạn văn bản gồm 2 câu: "Tôi thích đá bóng. Tôi thích xem phim" Văn bản trên có 6 từ riêng biệt bao gồm "Tôi", "thích", "đá", "bóng", "xem", "phim". Mã hóa one-hot của mỗi từ cho ra kết quả như sau:

"Tôi" = $[1, 0, 0, 0, 0, 0]$, "thích" = $[0, 1, 0, 0, 0, 0]$, "đá" = $[0, 0, 1, 0, 0, 0]$, "bóng" = $[0, 0, 0, 1, 0, 0]$, "xem" = $[0, 0, 0, 0, 1, 0]$, "phim" = $[0, 0, 0, 0, 0, 1]$



Hình 2.2: Ví dụ xử lý vector one-hot trong một mô hình

Tuy nhiên, việc mã hóa này không quan tâm đến ngữ nghĩa cũng như không đem lại hiệu quả tính toán, nhất là khi văn bản có số lượng lớn từ bởi nguyên nhân sẽ sinh ra một ma trận lớn nhưng thưa với đa số các phần tử

bằng 0, và chỉ số ít các phần tử có giá trị là 1. Việc tính toán khi đưa qua các tầng cũng sẽ cho ra các vector có đa số phần tử là 0. Để giải quyết vấn đề này, có những cách tiếp cận sau:

Word2Vec

Word2Vec là mô hình được đề xuất bởi Mikolov [30]. Với việc lấy các one-hot vector làm đầu vào, mô hình Word2Vec tạo ra các vector có số chiều là cố định. Đồng thời các vector biểu diễn từ mang nhiều ý nghĩa hơn khi những từ mà có nghĩa gần nhau sẽ có vector gần nhau hoặc *cos* của 2 vector lớn. Ý tưởng của mô hình này bắt nguồn từ việc con người nhận biết một từ thông qua các từ xung quanh. Ví dụ với văn bản:

"Hôm nay tôi đi học"

Khi bỏ động từ "đi học" để câu trở này "Hôm nay tôi ...", con người sẽ có khả năng đoán sau đó là một động từ. Hoặc khi giữ lại từ "đi học" và bỏ phần còn lại để câu trở thành "... đi học", người đọc cũng sẽ có xu hướng đoán trước đó phải là một danh từ "tôi", "tớ" và có thể đi kèm trước đó là trạng từ "hôm nay" hoặc "hôm qua". Từ đó, các mô hình nhúng từ về sau dựa trên ý tưởng này sẽ xem xét ngữ cảnh trong một cửa số có kích thước là W để xây dựng các vector biểu diễn.

Có 2 mô hình tương ứng với cách dự đoán cho phần được bỏ đi như trên ví dụ đã phân tích là $CBOW$ ($Continuous\ Bag\ Of\ Words$) và $Skip-gram$. Cụ thể, $Skip-gram$ sẽ dự đoán các từ xung quanh một từ cho trước. Đầu vào của mô hình là một vector one-hot của từ đích và output là W vector với W là kích cỡ cửa sổ nội dung được định nghĩa với người tạo ra mô hình. Ngược lại, $CBOW$ nhận đầu vào là các W từ xung quanh và cho đầu ra là vector của từ được dự đoán.

Bằng ý tưởng trên, cùng với bộ dữ liệu 3 triệu từ cung cấp bởi Google News đã tạo ra bộ vector biểu diễn từ Word2Vec. Mỗi từ sẽ được ánh xạ đến

một vector có số chiều là 300. Các vector này đã mô tả được phần nào thông tin về ngữ nghĩa.

fastText

Bộ embedding của Word2Vec dù đã nắm bắt được ngữ nghĩa của một số lượng lớn các từ, tuy nhiên vẫn tồn tại những từ không nằm trong tập từ điển. fastText ra đời và trở thành giải pháp cho vấn đề này. Đây là bộ embedding tạo ra bởi Facebook. Các vector biểu diễn từ tạo ra bởi mô hình này có thể trải qua quá trình học có giám sát hoặc không có giám sát.

Cũng giống Word2Vec, fastText sử dụng 2 mô hình là $skip-gram$ và $CBOW$. Điểm khác biệt khiến fastText có thể học được ra biểu diễn của những từ lần đầu thấy là việc mô hình này chia một từ ra thành n kí tự liên tiếp nhau, hay còn gọi là n-gram mức kí tự. Một từ sẽ được thêm dấu $<$ vào đầu và $>$ vào cuối, rồi lấy ra cụm n kí tự liên tiếp. Ví dụ từ “apple” sẽ chuyển thành “<apple>”, lấy $n = 3$ thì được các từ con “<ap”, “app”, “ppl”, “ple”, “le>”. Từ “apple” sẽ được biểu diễn dựa trên các từ con này. Việc thêm các dấu $>$ hay $<$ giúp phân biệt được từ đang xét với một từ khác. Ví dụ như biểu diễn của từ "app"là "<app>", khác với từ con "app"liệt kê phía trên được lấy ra trong từ "apple".

BioWordVec

Các bộ embedding kể trên đều đã giúp biểu diễn các từ trong không gian. Tuy nhiên, để sử dụng cho từng miền dữ liệu đặc thù hơn như miền dữ liệu về Y Sinh, cần phải huấn luyện thêm bằng các tập dữ liệu thuộc miền đó. BioWordVec đã thừa kế mô hình của fastText, kết hợp với các tập dữ liệu văn bản về Y Sinh để cho ra mô hình này vào năm 2019. Cụ thể, BioWordVec cũng sử dụng tư tưởng chia một từ thành các từ con như fastText, kết hợp với tri thức miền lấy từ nguồn dữ liệu MeSH (Medical Subject Heading) hoặc hệ thống

ngôn ngữ y học thống nhất (UMLS) - vốn chứa nhiều dữ liệu về Y Sinh, đã giúp nâng cao chất lượng của việc biểu diễn vector của những từ mang nhiều tính chuyên môn.

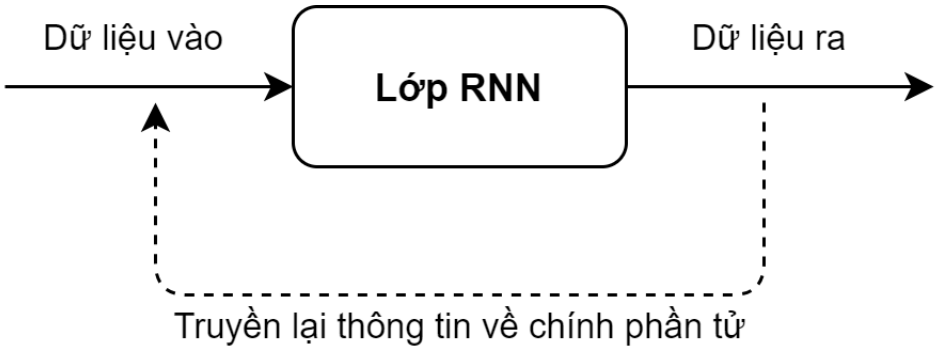
BioWordVec cũng thừa kề ưu điểm của fastText rằng có thể tạo ra biểu diễn vector của những từ gặp lần đầu dựa trên các từ con. Điều này lại đặc biệt cần thiết trong các văn bản Y Sinh vì thường xuất hiện những từ chuyên môn như “eltaproteobacteria” - vốn rất hiếm gặp, và xuất hiện cũng rất thưa trong 1 văn bản. Việc chia ra thành các cụm từ quen thuộc hơn như “bacteria”, “delta”, “proteo” đã giúp tạo ra được embedding cuối cùng cho một từ hiếm, trong khi Word2Vec thì không chắc có thể cho ra biểu diễn của từ.

Mạng Long Short-Term Memory

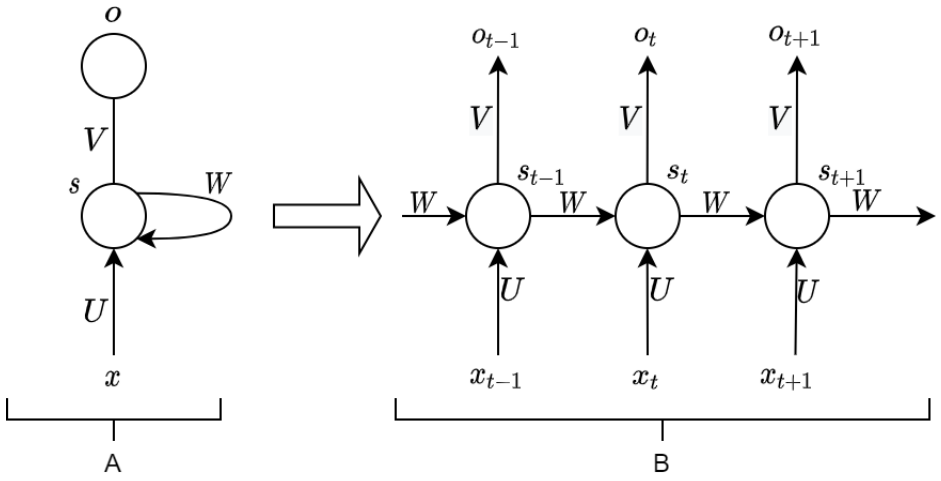
RNN

Mạng nơ-ron hồi quy là một lớp trong các mạng nơ-ron nhân tạo. Với các mạng nơ-ron thông truyền, có một giả thuyết được đặt ra là tất cả các đầu vào và đầu ra là độc lập với nhau. Tuy nhiên điều này lại không đúng với rất nhiều những bài toán. Ví dụ để dự đoán một từ ở trong câu, việc nhận biết, nắm bắt thông tin của các từ trước đó là cần thiết. Khi đó mạng nơ-ron hồi quy sẽ hoạt động theo cơ chế trên. Theo nghiên cứu [25], lý do chính để sử dụng mạng nơ-ron hồi quy là để xử lý thông tin dạng chuỗi. Và phần "hồi quy"trong mạng chính là kết quả khi thực hiện cùng một biến đổi cho mọi phần tử của chuỗi đầu vào và đầu ra phụ thuộc vào các tính toán trước đó. Về lý thuyết, cách mà mạng nơ-ron hồi quy có thể biểu diễn như thể mạng này có bộ nhớ để lưu trữ lịch sử về các phần tử đã xử lý trước đó. Vì thế tại mỗi thời điểm, các tính toán trước đó đều được sử dụng để dự đoán đầu ra tiếp theo từ quá trình.

Trong hình 2.4 có mục A mô tả trạng thái chưa phân rã của mạng nơ-ron hồi quy và hình B mô tả mạng nơ-ron hồi quy khi phân rã thành các mạng con. Từ mục B có thể quan sát rằng giả sử có 3 tầng mạng nơ-ron. Trong hình



Hình 2.3: Mô hình mạng nơ-ron hồi quy



Hình 2.4: Mô hình mạng nơ-ron hồi quy chưa phân rã và đã phân rã

cũng có đề cập đến 3 tham số U, V, W dùng cho việc tính toán mạng RNN. Đó là 3 tham số biểu diễn trọng số của các nơ-ron. Ví dụ, W biểu diễn trọng số của các nốt nằm trong trạng thái ẩn S. V biểu diễn trọng số của các nốt giữa các trạng thái ẩn S và đầu ra O. U biểu diễn trọng số của các nốt giữa đầu vào X và trạng thái ẩn s. Điểm khác biệt giữa RNN và mạng nơ-ron truyền thống là 3 tham số này không luôn có cùng một giá trị, nhưng trong mạng nơ-ron truyền thống thì đây là các tham số thay đổi. Có được điều này là bởi vì cùng một nhiệm vụ sẽ được thực hiện với mỗi tham số đầu vào.

- x_t là đầu vào tại thời gian t. Ví dụ x_0 là một vector one-hot tương ứng với từ đầu tiên của một câu.

- s_t là trạng thái ẩn tại thời gian t. Đây chính là bộ nhớ của mạng. s_t được tính dựa vào trạng thái ẩn trước đó s_{t-1} và đầu vào tại trạng thái hiện tại.

- $s_t = f(Ux_t + Ws_{t-1})$

Hàm f thường là hàm phi tuyến tính như $tanh$ hoặc $ReLU$. s_0 được gán là 0.

- o_t là đầu ra tại thời gian t. $o_t = softmax(Vs_t)$

Khi huấn luyện mô hình sẽ sử dụng chung tham số về các nốt tại mọi thời điểm, vì thế đạo hàm tại mỗi đầu ra phụ thuộc vào các bước tính toán hiện tại và trước đó. Ví dụ, để tính đạo hàm tại $t = 4$, mô hình sẽ phải truyền ngược về $t = 3$ và lấy tổng các đạo hàm trước đó. Quá trình này gọi là truyền ngược theo thời gian (Back Propagatiton Through Time - BPTT).

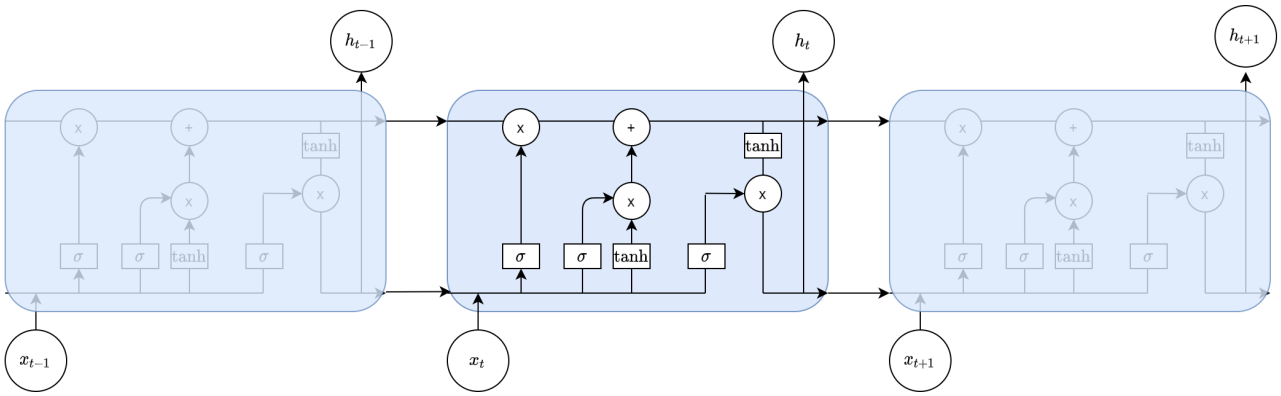
Tuy nhiên xuất hiện những người hợp mô hình cần nhiều ngữ cảnh hơn để đưa ra được đầu ra hợp lý. Điều này đặt ra vấn đề về phụ thuộc xa. Về lý thuyết, mạng nơ-ron hồi quy hoàn toàn có thể xử lý vấn đề này. Với sự can thiệp của con người để chọn ra tham số thích hợp thì vấn đề có thể được giải quyết. Thế nhưng trong thực tiễn, mô hình RNN không cho thấy khả năng đó. Điều này được chỉ ra bởi[2]

LSTM

LSTM (Long Short-Term Memory) là một mạng nơ-ron được sửa đổi dựa trên mạng nơ-ron hồi quy đã nêu ở phần trước cũng với vai trò nắm bắt thông tin phụ thuộc vào các thông tin trước đó. LSTM có khả giải quyết vấn đề phụ thuộc xa bằng cách thêm các cổng cho các ô. Mỗi ô có 2 trạng thái là mở và đóng, cho phép LSTM có thể hoạt động như bộ nhớ máy tính khi đưa ra quyết định dữ liệu nào được phép ghi vào, dữ liệu nào được phép đọc và lưu trữ lại. Thêm nữa, LSTM có 4 mạng nơ-ron tương tác với nhau.

Hình trên mô tả kiến trúc mạng LSTM. Trong đó các kí hiệu thể hiện:

+ : phép cộng vector



Hình 2.5: Mô hình LSTM

$\times$: thể hiện phép nhân vector

tanh và σ : các hàm phi tuyến

Mô hình gồm 4 bước:

Bước 1

Mạng LSTM nhìn vào các thông tin từ trạng thái h_{t-1} và đầu vào x_t để đưa ra quyết định giữ hoặc bỏ đi bằng hàm sigmoid. Đây chính là cổng giúp loại bỏ đi bớt thông tin từ trước đó.

$f_t = \sigma(w_f \times [h_{t-1}, x_t] + b_f) \Leftarrow (1)$

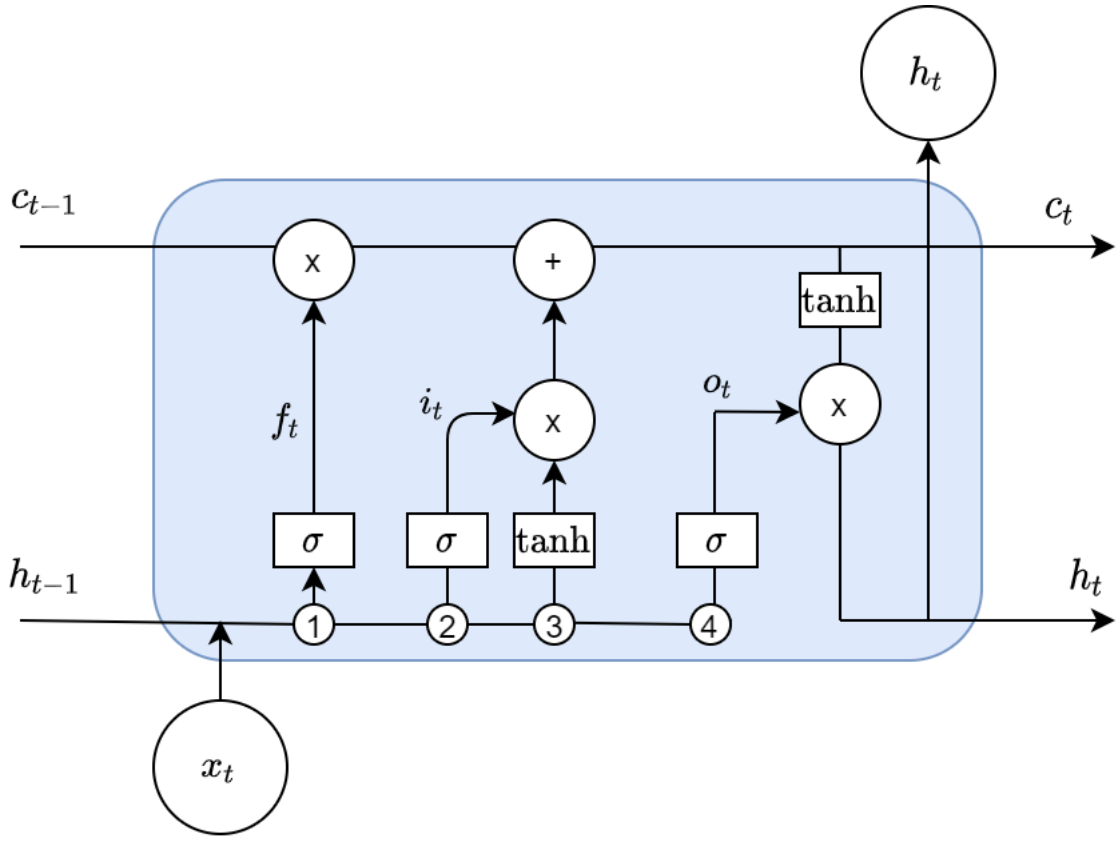
Bước 2

Bước này giúp ô trạng thái nhập nhật thông tin mới bằng biểu thức:

$i_t = \sigma(w_f \times [h_{t-1}, x_t] + b_i) \Leftarrow (2)$

$g_t = \tanh(w_c \times [h_{t-1}, x_t] + b_c) \Leftarrow (3)$

Phương trình 2 sử dụng hàm sigmoid để quyết định thông tin nào sẽ được thêm vào ô trạng thái.



Hình 2.6: Kiến trúc mô hình LSTM tại mỗi điểm thời gian t

Phương trình 3 chuyển thông tin mới về dạng vector để có thể cộng vào ô trạng thái.

Bước 3

Tại bước này, ô trạng thái trước đó là c_{t-1} được cập nhật vào ô trạng thái c_t bằng cách nhân f_t với c_{t-1}. Sau đó cộng với $i_t \times g_t$ để xác định nên cập nhật bao nhiêu vào trạng thái hiện tại.

$$c_t = f_t \times c_{t-1} + i_t \times g_t \Leftarrow (4)$$

Bước 4

Bước này có trách nhiệm tính toán đầu ra của LSTM - vốn sẽ là đầu vào cho tầng tiếp theo trong mạng nơ-ron. Trước khi xác định đầu ra, mô hình truyền thông tin qua một hàm sigmoid để xác định thông tin gì là quan trọng.

$o_t = \sigma(w_o \times [h_{t-1}, x_t] + b_o) \Leftarrow (5)$

$h_t = o_t \times \tanh(c_t) \Leftarrow (6)$

Phương trình số (6) truyền ô kết quả của trạng thái qua một hàm tanh rồi nhân kết quả với đầu ra sau khi đi qua hàm sigmoid.

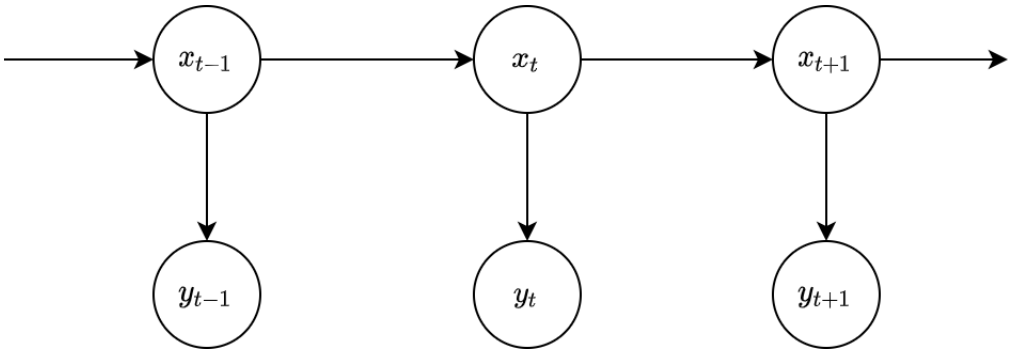
Tuy nhiên, kiến trúc LSTM trên chỉ giúp xử lý thông tin một chiều. Để cải thiện mô hình này, các đầu vào sẽ được truyền thêm vào mô hình một lần nữa nhưng theo chiều ngược lại. Kết quả từ việc truyền đầu vào theo 2 chiều rồi sẽ được ghép với nhau để biểu diễn kết quả cuối cùng. Việc kết hợp này sẽ giúp phát hiện, nắm bắt được cả sự phụ thuộc từ các từ trước hoặc sau một từ trong văn bản. Kiến trúc mạng được mô tả như dưới đây.

Hình 2.7: Kiến trúc mạng BiLSTM, các w_1, w_2, ... là các đầu vào

HMM

Mô hình Markov ẩn (Hidden Markov Model - HMM)

HMM là một mô hình thống kê được sử đụng dể mô tả quy trình Markov với các tham số không biết được ẩn đi. Mô hình xác định các tham số ẩn cho quy trình thông qua chuỗi các quan sát rồi sử dụng các tham số đó cho việc phân tích về sau [34]



Hình 2.8: Cấu trúc của HMM

Hình 2.7 mô tả cấu trúc của HMM. X là biến ẩn mà bên quan sát không phát hiện ra. x_t biểu diễn trạng thái tại thời gian t. Mỗi biến quan sát được y chỉ phụ thuộc vào x_t, và x_t liên quan đến trạng thái trước đó là x_{t-1}

Nếu các trạng thái ẩn có N giá trị, thì tại thời điểm t sẽ có thể nhận N giá trị . Vì vậy, có tối đa N^2 khả năng từ một trạng thái ẩn chuyển sang trạng thái ẩn tiếp theo. Với biến quan sát y có M có thể nhận, mỗi giá trị trong trạng thái ẩn đều có xác suất chuyển tới tới mọi biến quan sát khác. Vì vậy, nếu chuỗi quan sát gọi là Y, chuỗi trạng thái ẩn là X, trong đó:

$$X = (x_0, x_1, ..., x_n)$$

$$Y = (y_0, y_1, ..., y_n)$$

Xác suất của chuỗi Y qua mô hình HMM có thể được biểu diễn bằng biểu thức sau:

$$P(Y) = \sum_X P(Y|X)P(X)$$

Với dữ liệu chưa được gán nhãn, phương pháp thống kê của các tham số không thể tính toán được trực tiếp do sự tồn tại của các biến ẩn. Khi đó thuật toán Cực đại hóa kì vọng (Expectation Maximization - EM) được sử dụng để

lặp cho đến khi hội tụ để ra được tham số cho mô hình. Thuật toán EM được chia làm 2 phần là E và M. Ở bước E, thuật toán sử dụng tham số đã biết để tính ra hậu phân phối của biến ẩn $P(T|S,\theta^{old})$. Ở bước M, thuật toán tính giá trị kì vọng cực đại của log-likelihood dựa vào hậu phân phối. Kì vọng ở đây là một hàm có tham số θ, và hàm sẽ cực đại hóa kì vọng của hàm $Q = (\theta, \theta^{old})$ để có thể có được nghiệm của θ rồi dùng nghiệm này đưa vào θ^{old} trong bước E. Sau đó việc này được lặp lại cho đến khi hội tụ. (zhang2001segmentation)

Sau khi quá trình huấn luyện kết thúc, sử dụng mô hình HMM có thể dự đoán chuỗi câu mới. Khi mô hình nhận vào một tập các chuỗi có được từ việc quan sát, mô hình sẽ tìm ra chuỗi ẩn phù hợp nhất sử dụng thuật toán Viterbi ([11])

Mô hình HMM đã được sử dụng rộng rãi để giải quyết nhiều bài toán về xử lý ngôn ngữ tự nhiên như nhận diện giọng nói, dịch máy, gán nhãn, nhận diện tên thực thể ... [33]

CRF

Trường điều kiện ngẫu nhiên - Conditional Random Field (CRF)

CRF là mô hình giúp xử lý bài toán về chuỗi tương đối giống với mô hình HMM. Mô hình CRF có tất cả ưu điểm của HMM và tránh được vấn đề thiên vị cho nhãn của mô hình cực đại hóa entropy Markov (MEMM) ([29]).

Mục đích của mô hình là học hàm ánh xạ $x_s \to y_s$. Tuy nhiên mỗi đầu ra y_s không độc lập với nhau. Mô hình CRF có khả năng dự đoán vector đầu ra $y = y_0, y_1, ..., y_t$ thông qua tính toán xác suất có điều kiện của các biến ngẫu nhiên đưa ra trong vector quan sát là $x = x_0, x_1, ..., x_t$ ([40]) Mô hình CRF kết hợp phân loại phân biệt và mô hình xóa bằng đồ thị. Nhờ thế mà có thể mô hình hóa các đặc trưng một cách gọn gàng về sử dụng lượng lớn các thông tin đầu vào cho việc dự đoán. ([40])

Mô hình CRF cũng được áp dụng cho nhiều lĩnh vực. Ví dụ như việc xử

lý ngôn ngữ hay tiếng nói như bài toán gán nhãn từ ([12]), NER ([37]), trích xuất thông tin ([9] hay giải quyết mập mờ về ngữ pháp ([35]). Trong Tin sinh học, các ứng dụng CRF có thể kể đến là tìm ra liên kết protein ([31]), khám phá và dự đoán cấu trúc RNA [16].

Để mô tả mô hình CRF bằng đồ thị, gọi $G = (V, E)$ là một đồ thị. V biểu diễn tập đỉnh, E biểu diễn tập cạnh nối trực tiếp giữa 2 đỉnh. Mỗi đỉnh ν được gán một biến Y_v, thế nên $Y = (Y_v)\nu \in V$, với mọi Y_v tuân theo tính chất của Markov. Nhờ vậy mà cả mô hình đồ thị có thể mô tả bằng biểu thức sau: ([19])

$$P(Y_v|X, Y_w, w \neq \nu) = P(Y_v|X, Y_w, w \sim v)$$

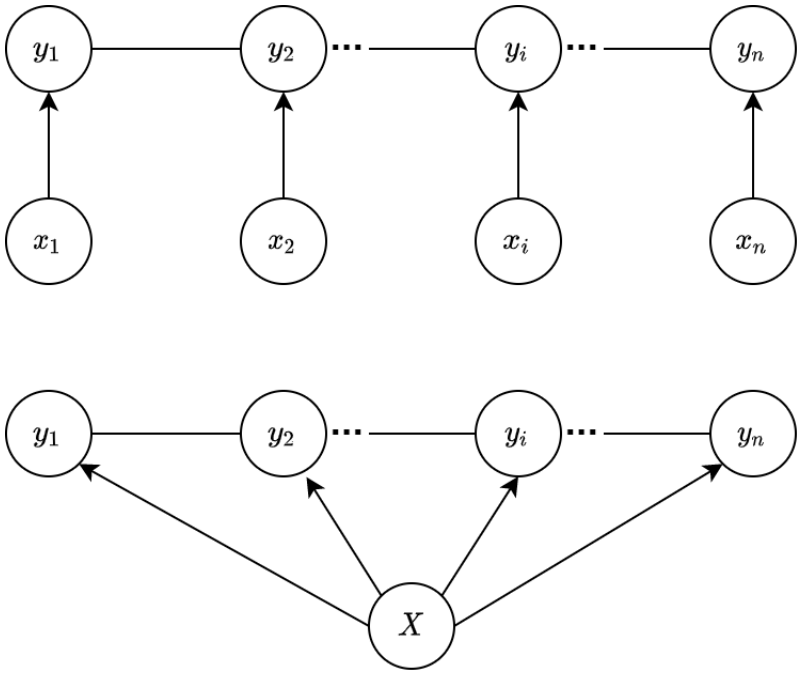
Trong đó, $w \sim v$ có nghĩa rằng w và v kết nối với nhau trong G.

Điều cần chú ý trong CRF là việc mô phỏng CRF thì lại dùng fđồ thị vô hướng. Vì thế, trong xử lý ngôn ngữ tự nhiên, các nhà nghiên cứu đã sử dụng chuỗi tuyến tính CRF [46].

Với bài toán nhận diện tên thực thể cũng thường sử dụng chuỗi tuyến tính CRF. Cấu trúc của chuỗi được nêu ra ở trên. Các đỉnh được gộp vào cấu trúc chuỗi tuyến tính, và chuỗi các đỉnh này tạo ra tương ứng với chuỗi Y. Khi cho một biến ngẫu nhiên $X = x_1, x_2, ..., x_n$ cùng $Y = y_1, y_2, ..., y_n$ với cấu trúc chuỗi tuyến tính thì phân phối xác suất có điều kiện của biến ngẫu nhiên Y là $P(Y|X)$ cũng tuân theo tính chất Markov.

$$P(y_i|x, y_1, y_2, ..., y_n) = P(y_i|x, y_{i-1}, y_{i+1})$$

Vì thế, với chuỗi quan sát X, ta có thể sử dụng phân phối $P(Y|X)$ của biến ngẫu nhiên Y để dự đoán chuỗi Y [44]



Hình 2.9: Cấu trúc của mô hình chuỗi tuyến tính CRF

Học đa tác vụ

Trong các mô hình học máy, chúng ta thường chỉ quan tâm đến một đơn vị cụ thể và huấn luyện riêng lẻ một mô hình để thực hiện tác vụ mà ta cần giải quyết. Sau bước này sẽ là tiếp tục tối ưu mô hinh cho đến khi kết quả của mô hình không còn tăng nữa. Hướng tiếp cận này đã giúp ta đạt được kết quả một mức tương đối tốt bằng việc chỉ tập trung vào một mục tiêu duy nhất. Thế nhưng điều này làm ta vô tình quên đi có những thông tin chất lượng, hữu ích từ những bài toán tương tự hoặc liên quan có thể giúp ta tăng hiệu suất của mô hình. Bằng việc chia sẻ những biểu diễn, thông tin giữa các tác vụ hay bài toán, ta có thể giúp mô hình tổng quan hóa tốt hơn trên tác vụ gốc. Đây chính là việc học đa tác vụ.

Việc học kết hợp như vậy đã được áp dụng thành công vào một số bài toán như xử lý ngôn ngữ tự nhiên [6], nhận diện giọng nói [8], thị giác máy [13]. Mô hình học này còn gọi là học kết hợp, học với những tác vụ liên quan. Tổng

quan, khi thấy rằng mô hình bài toán ta đang phải tối ưu nhiều hơn 1 hàm mất mát thì có nghĩa là ta đang áp dụng học đa tác vụ, vì mỗi tác vụ tương ứng với một hàm mất mát. Hoặc kể cả khi chỉ đang tối ưu một hàm mất mát thì vẫn có thể có một tác vụ liên quan giúp cho kết quả bài toán chính tốt hơn.

Điều này giống như khi con người muốn giải quyết một bài toán sẽ dùng đến kiến thức của những tác vụ liên quan. Ví dụ như một vận động viên điền kinh, ngoài các bài tập chạy thì họ còn phải trải qua những bài tập như giãn cơ, khởi động, xoay các khớp để giúp cơ thể linh hoạt, từ đó giúp cho việc tránh chấn thương và đạt được thể lực tốt nhất.

Để làm rõ tư tưởng về việc kết hợp học các bài toán, ta đi vào 2 phương pháp thường được sử dụng để học đa tác vụ trong các mạng nơ-ron. Trong ngữ cảnh học sâu, việc kết hợp này thường được thực hiện qua qua việc chia sẻ cố định hoặc tùy ý các tầng ẩn.

- Chia sẻ cố định các tham số

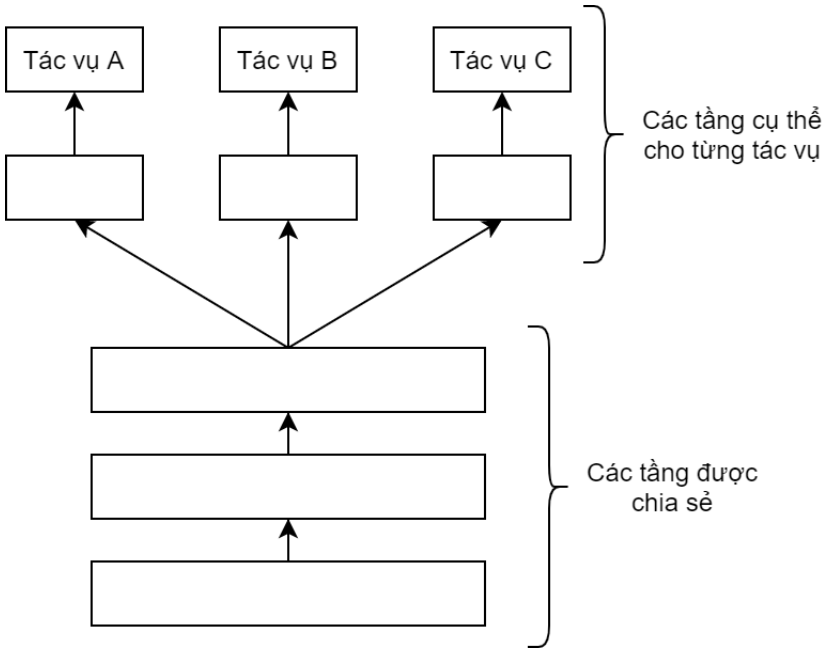
  Phương pháp này thường được sử dụng trong các mạng nơ-ron. Một vài tầng ẩn sẽ được chia sẻ giữa các tác vụ, trong khi đó giữ lại những tầng cụ thể phục vụ cho từng tác vụ riêng.

  Việc chia sẻ cố định như vậy giảm đáng kể việc mô hình bị quá khớp. Điều này là rõ ràng vì càng nhiều tác vụ ta huấn luyện cùng lúc, mô hình sẽ càng phải tìm ra một biểu diễn mà tổng quát được tất cả thông tin của các tác vụ đó, vì thế nên khả năng bị quá khớp giảm đi.

- Chia sẻ linh hoạt giữa các tham số

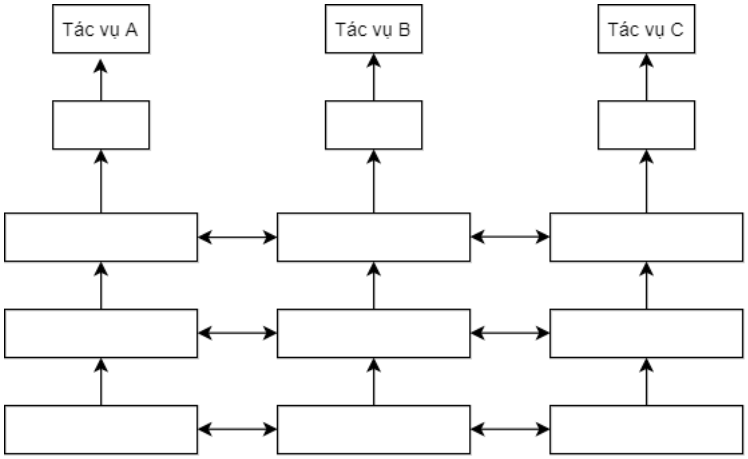
  Với phương pháp này, mỗi tác vụ sẽ được giải quyết với một mô hình riêng, bộ tham số riêng. Khoảng cách giữa các tham số sau đó được chuẩn hóa để làm sao cho các tham số là gần giống nhau.

Mô hình học đa tác vụ đã xuất hiện từ trước đó, tuy nhiên trong bài toán nhận diện tên thực thể Y Sinh thì chưa có nhiều thử nghiệm. [7] đã thử nghiệm mô hình này với mạng nơ-ron sử dụng chính là CNN. Mô hình của Crichton



Hình 2.10: Mô tả cấu trúc việc chia sẻ cố định các tham số

chỉ quan tâm các đặc trưng về từ ngữ mà bỏ qua ngữ nghĩa phản ánh bởi kí tự - một điều rất cần quan tâm trong các bài toán về tên thực thể Y Sinh (ví dụ "-ase" thường là một từ con quan trọng cần phải cân nhắc khi nhận diện tên gene/protein).



Hình 2.11: Mô tả cấu trúc việc chia sẻ linh hoạt các tham số

3 Phương pháp giải quyết vấn đề

Các chương trước đó đã cung cấp cái nhìn và kiến thức cơ bản về bài toán nhận diện tên thực thể, hay cụ thể hơn là nhận diện tên thực thể Y Sinh. Trong chương này sẽ mô tả phương pháp được sử dụng để giải quyết bài toán này.

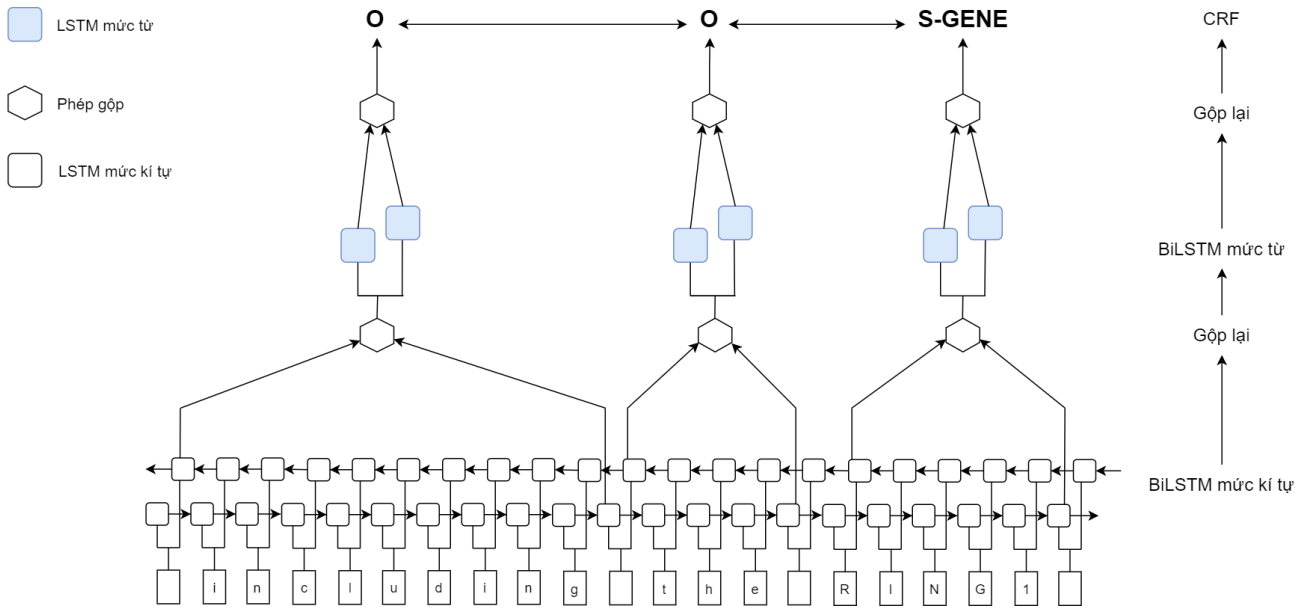
Với mô hình được đề xuất trong bài, khóa luận sẽ đi qua từng tầng, giải thích vì sao lại chọn thuật toán này cho từng tầng và việc kết hợp nhiều tác vụ cùng lúc sẽ đóng góp như thế nào vào kết quả chung của bài toán.

Với bài toán nhận diện tên thực thể Y Sinh, mô hình học đa tác vụ, cùng với kiến trúc BiLSTM-CRF là mô hình học sâu được chọn để giải quyết. Đầu tiên, mô hình sẽ sử dụng BiLSTM trên mức kí tự. Bên cạnh đó, embedding của các từ cũng sẽ được cho qua BiLSTM để nắm bắt thông tin ở mức từ ngữ. Kết quả của việc đưa từ và kí tự qua LSTM sẽ được ghép với nhau và đưa đến tầng cuối cùng là CRF. Tầng CRF này sẽ giúp gán nhãn cho chuỗi đầu vào.

Mô hình xử lý một tác vụ:

- Word Embedding:

  Với embedding ở mức từ, khóa luận này sử dụng vector cung cấp bởi mô hình BioWordVec - dựa trên tư tưởng của fastText rồi huấn luyện lại trên dữ liệu có miền đặc thù là Y Sinh: *PubMed* và *MIMIC III Clinical notes*. Các vector này có 200 chiều, giúp ánh xạ một từ vào không gian vector các số thực để có thể tính toán được. Ưu điểm của mô hình là giúp sinh ra được biểu diễn của những từ mà chưa từng gặp dựa trên các từ con



Hình 3.1: Kiến trúc mạng nơ-ron. Câu đầu vào là từ một văn bản Y Sinh. Các hình chữ nhật mô tả embedding của từ và các kí tự. Hình chữ nhật có viền tròn thể hiện kết quả việc sử dụng BiLSTM trên mức kí tự. Hình chữ nhật viền tròn có màu bên trong thể hiện kết quả việc sử dụng BiLSTM trên mức từ. Hình lục giác thể hiện phép gộp các kết quả lại với nhau. Các nhãn trên cùng là 'O' hay 'S-GENE' thể hiện đầu ra của tầng CRF - là các nhãn của thực thể cho mỗi từ trong một câu

hình thành bởi n kí tự liên tiếp. Điều này giúp nắm bắt thông tin tối đa về ngữ nghĩa của các từ, trong đó có cả các từ thuộc về tên của một thực thể Y Sinh.

- BiLSTM:

  Mô hình BiLSTM-CRF có thể học được những biểu diễn có chất lượng tốt cho các từ xuất hiện trong tập huấn luyện. Tuy nhiên, mô hình lại thường thất bại trong việc tổng quát hóa các từ không nằm trong tập từ vựng (out of vocabulary - OOV) vì không có embedding được huấn luyện sẵn. Nhờ thế mà mô hình trong khóa luận này sẽ thêm một mạng

BiLSTM để mô phỏng chuỗi kí tự trong câu đầu vào. Đầu vào của mạng là embedding của các kí tự. Sau khi cho qua mạng BiLSTM sẽ tạo ra các vector trạng thái ẩn. Các vector trạng thái ẩn đó tương ứng ở các vị trí biên của mỗi từ sẽ được ghép vào với nhau, sau đó tiếp tục ghép vào với các vector embedding của các từ đã có được ở tầng embedding của từ để tạo thành biểu diễn từ cuối cùng. Biểu diễn từ này cũng lại được đưa qua mạng BiLSTM (Mạng ở phía trên trong hình).

Đầu ra của mạng BiLSTM là các một chuỗi các vector trạng thái ẩn $h_1, h_2, ..., h_n$. Trọng số và bias của các tham số trong mạng BiLSTM bao gồm W^j, U^j, b^j với $j \in \{i, f, o, g\}$ được khởi tạo trong khoảng $\mu(-\sqrt{k}, \sqrt{k})$ với:

$$k = \frac{1}{h}$$

với h là số feature trong trạng thái ẩn.

Để tránh tình trạng quá khớp, một tầng drop-out được thêm vào mạng BiLSTM và sử dụng hàm tanh để kích hoạt tầng ẩn của BiLSTM. Đầu ra của mạng BiLSTM theo chiều xuôi $\{\overrightarrow{h_1}, \overrightarrow{h_2}, ..., \overrightarrow{h_n}\}$ và chiều ngược $\{\overleftarrow{h_1}, \overleftarrow{h_2}, ..., \overleftarrow{h_n}\}$ được kết hợp với nhau tạo thành tập $\{h_1, h_2, ..., h_n\} \in R^{n \times m}$

- Tầng tuyến tính

  Trước khi đưa các vector trạng thái ẩn vào tầng CRF, các vector này phải được biến đổi từ m chiều về k chiều với k là số nhãn mà ta sẽ gán cho các từ. Khi đó các trạng thái ẩn sau khi qua tầng này sẽ cho ra các ma trận $P_i \in R^k$, trong đó p_{ij} là số điểm cho việc gán nhãn thứ j cho từ x_i

- Tầng CRF Có thể chỉ dùng mạng BiLSTM để tiếp cận bài toán gán nhãn chuỗi là sử dụng các vector trạng thái ẩn để đưa ra các nhãn một cách độc lập. Tuy nhiên trong nhiều bài toán gán nhãn chuỗi như BioNER, việc quan tâm đến sự phụ thuộc giữa các nhãn đem lại kết quả chính xác hơn.

Tầng CRF thực hiện việc gán nhãn chuỗi ở mức từng câu. Tham số của CRF là một ma trận A có kích thước $(k+2) \times (k+2)$. A_{ij} thể hiện khả năng nhãn thứ i chuyển sang nhãn thứ j. Kích cỡ ma trận như vậy do mô hình sẽ thêm 2 nhãn đặc biệt là 'START' và 'END'. Nhãn 'START' đánh dấu bắt đầu câu và cũng là trạng thái đầu tiên, trong khi nhãn 'END' đánh dấu trạng thái kết thúc câu.

Với chuỗi nhãn $y = y_1, y_2, ..., y_n$, số điểm được tính toán bởi tầng CRF có công thức là:

$$score(x, y) = \sum_{i=1}^{n} P_{i,y_i} + \sum_{i=1}^{n+1} A_{y_{i-1}y_i}$$

Số điểm cho tại mỗi vị trí i gồm 2 thành phần: số điểm cho việc gán nhãn y_i cho từ x_i và số điểm cho việc chuyển từ trạng thái lấy từ mà trận chuyển A của mô hình CRF.

Gọi Y_X là tất scar khả năng gán nhãn chuỗi với chuỗi đầu vào là X. Sau đó số điểm này sẽ được chuẩn hóa thông qua hàm Softmax để đưa về xác suất:

$$P(y|X) = \frac{\exp(score(x, y))}{\sum_{y' \in Y_X}(score(x, y'))}$$

Thực hiện lấy log, công thức trên chuyển thành:

$$\log(P(y|X)) = \log(\frac{\exp(score(x, y))}{\sum_{y' \in Y_X}(score(x, y'))})$$

$$\Leftrightarrow \log(P(y|X)) = score(x, y) - log(\sum_{y' \in Y_X}(score(x, y'))) \quad (1)$$

Việc huấn luyện mô hình sẽ cực đại hóa xác suất log của chuỗi nhãn y.

Kiến trúc 3 lớp BiLSTM-CRF được sử dụng bởi [20] để kết hợp biểu diễn

chuỗi từ và chuỗi kí tự trong câu đầu vào. Ở kiến trúc trong khóa luận này, tầng BiLSTM đầu tiên lấy embedding trên chuỗi kí tự của mỗi từ làm đầu vào rồi cho ra vector biểu diễn ở mức kí tự của mỗi từ. Các vector này seau đó được kết hợp với vector biểu diễn từ và đưa vào tầng BiLSTM thứ 2. Cuối cùng, tầng CRF đưa ra nhãn từ hợp nhất bằng cách cực đại hóa xác suất log trong công thức trên.

Trong thực tế, các vector embedding cho các kí tự đầu tiên được khởi tạo ngẫu nhiên và cùng được huấn luyện trong quá trình học. Ở tầng cuối cùng, thuật toán Viterbi được sử dụng để suy ra chuỗi nhãn cuối cùng cho mô hình CRF. So sánh với mô hình BiLSTM-CRF thường thấy, ưu điểm của mô hình này là có thể suy luận ngữ nghĩa của những từ nằm ngoài từ điểm dựa vào chuỗi kí tự và các kí tự xung quanh. Đặc biệt với dữ liệu Y Sinh vốn có đặc điểm về hình thái từ như tiền tố, hậu tố, ... Ví dụ mô hình có thể suy luận rằng "RING2"có thể là tên của một loại gene, mặc dù mô hình chỉ thấy thực thể có tên "RING1"là một loại gene trong lúc huấn luyện.

Mô hình học đa tác vụ:

Định nghĩa cho mô hình học đa tác vụ như sau:

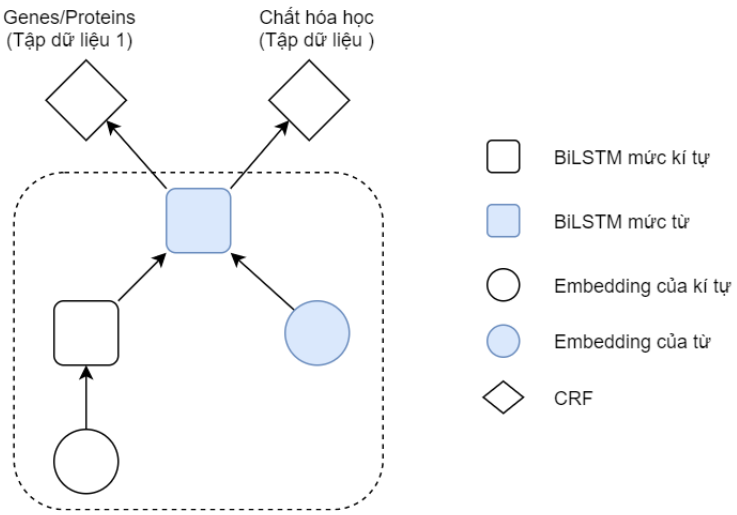
Cho m tập dữ liệu, với mỗi $i \in 1, 2, .., m$, mỗi tập dữ liệu D_i có n_i mẫu huấn luyện. Hay nói cách khác $D_i = \{w_j^i, y_j^i\}_{j=1}^{n_i}$. Gọi ma trận huấn luyện của mỗi tập dữ liệu là $X^i = \{x_1^i, x_2^i, ..., x_n^i\}$ (X^i là biểu diễn của từ w_j^i) và các các nhãn cho mỗi tập dữ liệu là $y^i = \{y_1^i, y_2^i, ..., y_n^i\}$. Các tham số của mô hình bao gồm tham số của mạng BiLSTM ở mức từ (θ_i^w), tham số của BiLSTM ở mức kí tự (θ_i^c), tham số của CRF (θ_i^o). Một mô hình học đa tác vụ bao gồm m mô hình khác, mỗi mô hình được huấn luyện trên một bộ dữ liệu trong khi chia sẻ một phần các tham số trong mô hình giữa các tập dữ liệu. Hàm mất mát L của mô hình đa tác vụ là:

$$L = \sum_{i=1}^{m} \lambda_i L_i = \sum_{i=1}^{m} log(P_{\theta_i^w, \theta_i^c, \theta_i^o}(y^i|X^i))$$

Với hàm log đã được định nghĩa trong công thức (1). Trong khóa luận này, các tham số được chia sẻ giữa các mô hình là θ_i^c và θ_i^w là các tham số của tầng BiLSTM. Các tham số của tầng CRF dùng cho việc dự đoán nhãn là θ_i^o được sử dụng riêng cho từng tập dữ liệu. Mô hình này sẽ chia sẻ tối đa thông tin bao gồm thông tin về kí tự và từ về tên các thực thể Y Sinh giữa các tác vụ.

λ_i là một siêu tham số không âm cho phép tinh chỉnh mức độ đóng góp của tập dữ liệu thứ i. Trong khóa luận này, tất cả các λ_i được đặt bằng 1, đồng nghĩa với việc tất cả các tập dữ liệu đều có đóng góp như nhau vào kết quả chung.

Khóa luận sử dụng mô hình học đa tác vụ thay vì việc gộp chung tất cả các bộ dữ liệu và huấn luyện trên một mô hình do omoix tập dữ liệu chỉ tập trung đến một hoặc một vài loại thực thể. Việc kết hợp tất cả tập dữ liệu sẽ dễ gây ra những lỗi False Negative (đoán sai hoàn toàn tên thực thể). Ví dụ tập dữ liệu A chứa thông tin về gene, trong khi tập dữ liệu B chứa toàn thông tin về thuốc. Việc gộp A và B để huấn luyện một mô hình đơn tác vụ sẽ dẫn đến hậu quả không nhận diện được thực thể về thuốc trong A và thực thể về gene trong B.



Hình 3.2: Mô hình học đa tác vụ. Hình tròn màu trắng thể hiện embedding của các kí tự. Hình chữ nhật có viền tròn thể hiện thông tin về kí tự khi đi qua BiLSTM. Hình tròn có màu xanh thể hiện embedding ở mức từ. Hình chữ nhật viền tròn có màu xanh thể hiện thông tin ở mức từ khi đi qua BiLSTM. Hình vuông thể hiện lớp CRF. Tham số về từ và kí tự được sử dụng chung cho tất cả các tác vụ

4 Thực nghiệm, kết quả, so sánh

Tập dữ liệu

Các thực nghiệm của bài toán nhận diện tên thực thể Y Sinh trong khóa luận này sẽ sử dụng 5 bộ dữ liệu có tên BC2GM, BC4CHEMD, BC5CDR, NCBI-disease và JNLPBA theo gợi ý của [45]. Mỗi bộ dữ liệu đều chia ra thành tập huấn luyện, tập phát triển và tập kiểm tra. Tập huấn luyện và tập phát triển được gộp chung để huấn luyện mô hình như trong các nghiên cứu [27], [26] hay [21]. Tất cả các tập dữ liệu đều được công khai. Bộ nhãn được dùng trong bài toán là IOBES. Ví dụ một từ mô tả tên một bực thể được gán nhãn là "B-Gene"khi và chỉ khi từ đó đánh dấu bắt đầu tên một thực thể, gán nhãn là "I-Gene"khi từ đó ở giữa tên một thực thể và "E-Gene"khi từ đó là từ cuối cùng xuất hiện trong tên. Bên cạnh đó, các tên thực thể có 1 từ thì được gán nhãn là S- và đi kèm sau đó là loại thực thể, ví dụ như "S-Gene". Tất cả những từ khác không giúp mô tả tên thực thể ta cần quan tâm sẽ gán nhãn là O. Các tập dữ liệu tuân theo định dạng giống như tập dữ liệu CoNLL 2003 NER. Cụ thể hơn, các dòng trống thể hiện ngăn cách giữa các câu, và ngăn cách giữa các tài liệu có dạng là:

```
-DOCSTART- -X- -X- -X- O
```

Các dòng khác chứa từ và nhãn của mỗi từ. Trường đầu tiên của mỗi dòng phải là từ, và trường cuối cùng trong dòng là nhãn của từ.

Bảng 4.1 mô tả kích cỡ cùng loại dữ liệu của mỗi tập dữ liệu.

Bảng 4.1: Các tập dữ liệu Y Sinh sử dụng cho thí nghiệm

Tập dữ liệu	Kích cỡ	Loại thực thể và số lượng
BC2GM	20000 câu	Gene/Protein (24583)
BC4CHEMD	10000 tóm tắt	Chất hóa học (84310)
BC5CDR	1500 bài báo	Chất hóa học (15935), Tên bệnh (12852)
NCBI-Disease	793 tóm tắt	Tên bệnh (6881), Gene/Protein (35336)
JNLPBA	2404 tóm tắt	Dòng tế bào gốc (4330), DNA (10589) Loại tế bào (8649), RNA (1069)

Cách đánh giá

Khóa luận sẽ báo cáo trên tập kiểm thử. Mỗi tên thực thể do mô hình dự đoán được xem là đúng khi và chỉ khi loại thực thể và tên thực thể khớp hoàn toàn với dữ liệu thực tế. Đây gọi là đánh giá so khớp trên mức tên thực thể.

Ma trận lỗi của bài toán nhận diện tên thực thể bao gồm:

		Nhãn thật sự	
		True	False
Nhãn dự đoán	Positive	True Positive	False Positive
	Negative	True Negative	False Negative

Bảng 4.2: Định nghĩa ma trận lỗi của bài toán nhận diện tên thực thể

Xét trong một thực thể. Khi các từ trong tên của thực thể gán nhãn sai tất cả sẽ tính số từ gán nhãn sai đó vào False Negative. Ngược lại, khi tất cả các từ trong tên một thực thể được gán nhãn đúng, số lượng nhãn đúng tính vào mục True Positive. Thêm vào đó, trong trường hợp mô hình đoán sai tên biên của thực thể, số nhãn sai trong tên đó được tính vào False Negative. Không kể các nhãn được gán đúng, các nhãn gán sai khi gặp trường hợp xác định không

đúng biên được tính vào False Positive.

Sau đó báo cáo kết quả sử dụng độ đo F1 trên toàn bộ các tập dữ liệu.

Môi trường và các tham số

Để huấn luyện mô hình, tham số tốc độ học đặt là 0.01. Chiều embedding của từ là 200, chiều embedding của kí tự là 30. Nhằm giúp mô hình tránh gặp tình trạng quá khớp, hệ số dropout đặt là 0.5, đồng nghĩa sẽ lược bỏ 50% đỉnh trong mạng khi huấn luyện BiLSTM. Bên cạnh đó, mô hình sẽ được chỉ định dừng lại khi số epoch tối đa không cho ra kết quả f1 cải thiện là 30. Kích cỡ của mỗi lô là 10, đồng thời mô hình phải chạy ít nhất 50 epoch cho mỗi lần huấn luyện.

Mô hình cài đặt sử dụng Python và Pytorch. Pytorch là thư viện mã nguồn mở về các thuật toán học máy, thường sử dụng cho các ứng dụng như xử lý ngôn ngữ tự nhiên hoặc thị giác máy bên cạnh các ưu điểm như nhiều tài liệu, cộng đồng hỗ trợ đông (hơn 30000 người).

Các thực nghiệm

- So sánh giữa mô hình học đơn tác vụ và đa tác vụ

Tập dữ liệu	Đơn tác vụ	Đa tác vụ
BC2GM	79.64	79.98
BC4CHEMD	87.51	89.02
BC5CDR	86.95	88.75
NCBI-disease	83.74	86.61
JNLPBA	72.02	73.45

Bảng 4.3: Kết quả khi chạy mô hình học đơn tác vụ và đa tác vụ

Bảng 4.3 ghi lại số đo F1 khi huấn luyện mô hình chỉ sử dụng một bộ dữ liệu duy nhất và sử dụng kết hợp nhiều bộ dữ liệu với nhau. Số liệu cho thấy kết quả đã có sự tiến bộ rõ rệt khi sử dụng mô hình học kết hợp nhiều tác vụ với nhau. Kết quả của tập dữ liệu BC2GM tăng ít nhất với mức tăng 0.34% trong khi kết quả cải tiến được nhiều nhất ở tập NCBI-disease với mức tăng 2.87%. Tổng mức tăng trên cả bộ dữ liệu là 7.95% khi áp dụng việc học đa tác vụ. Từ đó thấy được mô hình học đa tác vụ có thể tận dụng được thông tin trên các bộ dữ liệu với nhau và đồng thời cùng nhau tăng độ chính xác của các trên các tác vụ.

- Mức độ tương quan giữa các tập dữ liệu

  Để xác định tác động của việc học kết hợp các bộ dữ liệu theo mô hình đa tác vụ, khóa luận sẽ chọn một ra n bộ dữ liệu (với $n \geq 2$ và $n \leq 4$ rồi huấn luyện mô hình trên n tập dữ liệu đó. Mỗi tập dữ liệu tương ứng với một tác vụ. Dấu "thể hiện tập dữ liệu đó không tham gia vào quá trình huấn luyện. Vì giới hạn tránh quá dài dòng, các kết quả quan trọng được thể hiện bằng bảng dưới đây:

Các tập dữ liệu	BC2GM	BC4CHEMD	BC5CDR	NCBI-disease	JNLPBA
BC2GM + BC4CHEMD + BC5CDR	79.75	88.84	88.32	-	-
BC2GM + BC5CDR + NCBI-disease	79.87	-	88.71	86.43	-
BC4CHEMD + BC5CDR	-	88.86	87.93	-	-
BC2GM + NCBI-disease + JNLPBA	79.85	-	-	86.33	73.31
BC4CHEMD + BC2GM	79.63	88.82	-	-	-
BC2GM + BC5CDR + NCBI-disease + JNLPBA	79.89	-	88.63	88.61	73.32
BC4CHEMD + JNLPBA	-	88.84	-	-	72.04

Bảng 4.4: Kết quả của từng bộ dữ liệu khi kết hợp huấn luyện với nhau

Số liệu từ bảng 4.4 cho thấy việc kết hợp học giữa các tập dữ liệu cũng sẽ ảnh hưởng đến kết quả. Nhìn chung, kết quả sẽ có sự cải thiện khi tăng số tác vụ lên theo số tập dữ liệu, đặc biệt với những bộ dữ liệu có đặc điểm

chung về tên của thực thể. Cụ thể, việc sử dụng các tập dữ liệu có chung loại thực thể như BC2GM, BC5CDR và NCBI-disease cùng có nhiều thực thể về gene/protein (24583 và 35336 thực thể) sẽ giúp cho mô hình dự đoán được chính xác hơn (thể hiện qua f1 trên tập BC2GM tăng từ 79.80% lên 79.85% khi được huấn luyện cùng NCBI-disease và BC5CDR) thay vì việc kết hợp giữa các bộ dữ liệu không có chung loại thực thể rồi khiến kết quả giảm đi. Điều này có thể lý giải bằng việc khi chọn các tập dữ liệu không có đặc điểm chung gần như có nghĩa rằng ta đang vừa huấn luyện các mô hình đơn tác vụ, đồng thời vô tình gây nhiễu giữa các tác vụ nhận diện với nhau.

Tuy nhiên với tập dữ liệu BC4CHEMD, kết quả thay đổi rất ít (chênh lệch khoảng 0.02% đến 0.04% dù đã sử dụng với bộ dữ liệu BC5CDR cũng có tên về các chất hóa học. Điều này là do tương quan về dữ liệu giữa 2 tập này không cao khi BC4CHEMD có đến 84310 tên chất hóa học, nhiều hơn gấp 5 lần so với số thực thể trong tập BC5CDR - chỉ có 15935 thực thể. Thêm nữa, việc kết hợp thêm tập BC4CHEMD còn khiến thời gian huấn luyện tăng lên trong khi như đã thực nghiệm ở trên, đôi khi còn gây khó khăn cho việc huấn luyện do kích cỡ của tập này không nhỏ (gồm 10000 tóm tắt chứa 84310 tên chất hóa học).

Kết quả trên tập dữ liệu JNLPBA đạt 73.31% và 73.32% khi huấn luyện cùng tập BC2GM và NCBI-disease. Kết quả này cho thấy cũng không có sự thay đổi nhiều khi được sử dụng để huấn luyện chung với các tập dữ liệu khác, do đặc điểm đây là bộ dữ liệu có nhiều loại thực thể, đồng thời cũng có độ nhiễu nhất định. Ví dụ như cụm từ "truncated RARalpha"thì "trucated"(dịch: đã cắt bớt) không mang lại bất kì định danh nào có ý nghĩa cho từ "RARalpha"phía sau. Tuy nhiên khi đặt vào huấn luyện chung với tập BC4CHEMD, F1 trên tập JNLPBA chỉ đạt 72.04%, cho thấy để đạt kết quả trên 73% trên JNLPBA cũng cần thiết có sự kết hợp việc học được các tên thực thể về gene/protein trên tập BC2GM và NBCI-disease.

- Ảnh hưởng của word embedding

  Để có được biểu diễn vector của các từ, mô hình trong khóa luận thử nghiệm vector lấy từ 2 mô hình là Word2Vec và BioWordVec. Mô hình Word2Vec đã được huấn luyện sử dụng skip-gram trên bộ dữ liệu Pubmed và Wikipedia, trong khi BioWordVec là mô hình fastText và được huấn luyện trên Pubmed và MeSH. Kết quả từ bảng 4.5 cho thấy việc sử dụng vector biểu diễn từ sinh ra bởi mô hình BioWordVec giúp cho kết quả tốt hơn từ 0.03% cho đến 0.36%. Kết quả có được nhờ việc huấn luyện riêng từng tập dữ liệu. Lý giải cho cải tiến này là vì fastText có thể sinh ra được biểu diễn của những từ không nằm trong bộ từ vựng đã có dựa trên biểu diễn của những kí tự hoặc cụm kí tự cấu tạo nên từ, trong khi Word2Vec vẫn tồn tại những từ chưa có biểu diễn vector. Các từ mà chưa biết đó sẽ được thay bằng từ "UNK"và vector biểu diễn cho từ "UNK"sẽ được khởi tạo ngẫu nhiên. Kết quả trong bảng là số đo f1 khi chạy riêng từng tác vụ, chỉ thay đổi word embedding sử dụng cho từ là Word2Vec hay BioWordVec.

Tập dữ liệu	Word2Vec	BioWordVec
BC2GM	79.64	79.72
BC4CHEMD	87.48	87.51
BC5CDR	83.38	83.74
NCBI-disease	83.51	83.74
JNLPBA	71.89	72.02

Bảng 4.5: Kết quả mô hình đơn tác vụ khi sử dụng Word2Vec và BioWordVec

- Ảnh hưởng của việc lựa chọn tham số để chia sẻ giữa các tác vụ

  Như trong mô hình đề xuất, giữa các tác vụ sẽ sử dụng chung các tham số cho mô hình BiLSTM trên kí tự và trên các từ. Thử nghiệm với việc

lần lượt chỉ cho các tác vụ chia sẻ tham số về BiLSTM ở mức kí tự và BiLSTM ở mức từ và để mô hình tự học ra các tham số còn lại cho từng tác vụ. Kết quả thu được như sau:

Tập dữ liệu	MTL-C	MTL-W	MTL-CW
BC2GM	77.88	78.73	79.98
BC4CHEMD	88.46	88.65	89.02
BC5CDR	86.57	88.16	88.75
NCBI-disease	82.37	84.52	88.61
JNLPBA	71.31	71.69	73.45

Bảng 4.6: Kết quả khi chạy mô hình với các tham số dùng chia sẻ giữa các tụ được thay đổi

MTL-C thể hiện việc huấn luyện 5 tập dữ liệu và chỉ chia sẻ tham số về BiLSTM mức kí tự. MTL-W thể hiện việc huấn luyện 5 tập dữ liệu và chỉ chia sẻ tham số về BiLSTM mức từ ngữ. MTC-WC là kết quả mô hình khi đề xuất trong khóa luận. Kết quả cho thấy mô hình MTL-CW cho kết quả tốt nhất, thường cao hơn cách chọn tham số là từ hay kí tự từ khoảng hơn 1% đến gần 6%. Điều này giúp củng cố rằng việc học được thông tin về hình thái từ bởi BiLSTM mức kí tự và thông tin về từ vựng cũng như ngữ cảnh ở mức từ là cần thiết để cải thiện kết quả việc nhận diện tên thực thể.

5 Kết luận

Kết quả đạt được từ khóa luận

Khóa luận này đã sử dụng hướng tiệp cận bài toán nhận diện tên các thực thể Y Sinh trong một văn bản bằng cách học nhiều tác vụ liên quan, làm giàu thông tin từ các tác vụ đó. Quá trình nghiên cứu, tìm hiểu, xây dựng và thử nghiệm mô hình theo hướng tiếp cận này đã đem đến những kết quả như:

- Đem đến cái nhìn khác cho bài toán nhận diện tên thực thể. Bài toán này giờ đây được tiếp cận bằng cách gán nhãn chuỗi bằng các định danh chỉ một từ nằm trong, bắt đầu, kết thúc, chính từ đó là tên thực thể hoặc không thuộc về bất kì tên thực thể nào.
- Mô hình tương đối đơn giản chỉ với 2 tầng BiLSTM và 1 tầng CRF đặc trưng cho mỗi tác vụ. Tuy mô hình không phức tạp nhưng lại đem lại những cải tiến cho bài toán khi so sánh với việc chỉ sử dụng một tác vụ. Điều này có được là nhờ việc chia sẻ thông tin về từ cũng như về kí tự giữa các loại thực thể Y Sinh.
- Nghiên cứu về mức độ tương quan giữa các tập dữ liệu về Y Sinh. Việc kết hợp các tập dữ liệu phù hợp sẽ giúp cho việc nhận diện tên thực thể hiệu quả hơn. Ngược lại, việc chọn không đúng thậm chí có thể gây nhiễu cho tác vụ mà mô hình muốn cải tiến.
- Tìm hiểu về mô hình tạo ra biểu diễn vector của các từ sử dụng công cụ Word2Vec và fastText (BioWordVec) và thực nghiệm áp dụng 2 công cụ vào mô hình của khóa luận để quyết định bộ embedding nào là phù hợp hơn. Qua thực nghiệm đã thấy được BioWordVec đem lại hiệu quả tốt

hơn nhờ việc biểu diễn được cả những từ không nằm trong từ điển so với Word2Vec đã huấn luyện trên bộ dữ liệu Y Sinh.

Phương hướng tiếp cận trong tương lai

Tuy nhiên mô hình này vẫn có thể cải tiến thêm theo những hướng đi như:

- Chỉnh sửa tham số về mức độ tham gia của mỗi tập dữ liệu thay vì để mặc định mỗi bộ dữ liệu đều có vai trò tương đương nhau tham gia vào kết quả chung.
- Sử dụng những phép biểu diễn từ khác như BERT, Elmo, ... là những phép biểu diễn từ dựa trên cả ngữ nghĩa của những từ xung quanh. Việc sử dụng được các biểu diễn mang nhiều thông tin có thể sẽ giúp việc nắm bắt thông tin về ngữ nghĩa được tốt hơn, làm cơ sở cho việc kết hợp thông tin về từ và về kí tự một từ một cách hiệu quả.
- Kết hợp học đơn tác vụ và đa tác vụ cũng có thể là cách để giúp mô hình nhẹ hơn, đồng thời vẫn cải tiến được độ chính xác của mô hình.
- Tìm hiểu và thử nghiệm thêm các bộ dữ liệu đã được gán nhãn khác dựa trên sự tương quan về loại thực thể cũng là một hướng cần nghiên cứu thêm.

6 Tham khảo

Bibliography

[1] Jonathan Baxter. "A Bayesian/information theoretic model of learning to learn via multiple task sampling". In: *Machine learning* 28.1 (1997), pp. 7–39.

[2] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166.

[3] Jeffrey T Chang, Hinrich Schütze, and Russ B Altman. "GAPSCORE: finding gene and protein names one word at a time". In: *Bioinformatics* 20.2 (2004), pp. 216–225.

[4] Jason PC Chiu and Eric Nichols. "Named entity recognition with bidirectional LSTM-CNNs". In: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 357–370.

[5] Murat Cokol, Ivan Iossifov, Chani Weinreb, and Andrey Rzhetsky. "Emergent behavior of growing knowledge about molecular interactions". In: *Nature biotechnology* 23.10 (2005), pp. 1243–1247.

[6] Ronan Collobert and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 160–167.

[7] Gamal Crichton, Sampo Pyysalo, Billy Chiu, and Anna Korhonen. "A neural network multi-task learning approach to biomedical named entity recognition". In: *BMC bioinformatics* 18.1 (2017), p. 368.

[8] Li Deng, Geoffrey Hinton, and Brian Kingsbury. "New types of deep neural network learning for speech recognition and related applications: An overview". In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE. 2013, pp. 8599–8603.

[9] Mike Ebersbach, Robert Herms, Christina Lohr, and Maximilian Eibl. "Wrappers for Feature Subset Selection in CRF-based Clinical Information Extraction." In: *CLEF (Working Notes)*. 2016, pp. 69–80.

[10] Sergei Egorov, Anton Yuryev, and Nikolai Daraselia. "A simple and practical dictionary-based approach for identification of proteins in MEDLINE abstracts". In: *Journal of the American Medical Informatics Association* 11.3 (2004), pp. 174–178.

[11] Zoubin Ghahramani. "An introduction to hidden Markov models and Bayesian networks". In: *Hidden Markov models: applications in computer vision*. World Scientific, 2001, pp. 9–41.

[12] Souvick Ghosh, Satanu Ghosh, and Dipankar Das. "Part-of-speech tagging of code-mixed social media text". In: *Proceedings of the Second Workshop on Computational Approaches to Code Switching*. 2016, pp. 90–97.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Proceedings of the IEEE international conference on computer vision*. 2015.

[14] Lynette Hirschman, Alexander A Morgan, and Alexander S Yeh. "Rutabaga by any other name: extracting biological names". In: *Journal of Biomedical Informatics* 35.4 (2002), pp. 247–259.

[15] Chung-Chi Huang and Zhiyong Lu. "Community challenges in biomedical text mining over 10 years: success, failure and the future". In: *Briefings in bioinformatics* 17.1 (2016), pp. 132–144.

[16] Alexander Rosenberg Johansen, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. "Deep recurrent conditional random field network for protein secondary prediction". In: *Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics*. 2017, pp. 73–78.

[17] Ross Kindermann. "Markov random fields and their applications". In: *American mathematical society* (1980).

[18] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[19] John Lafferty, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In: (2001).

[20] Guillaume Lample et al. "Neural architectures for named entity recognition". In: *arXiv preprint arXiv:1603.01360* (2016).

[21] Robert Leaman and Zhiyong Lu. "TaggerOne: joint named entity recognition and normalization with semi-Markov Models". In: *Bioinformatics* 32.18 (2016), pp. 2839–2846.

[22] Digits Using Backpropagation Learning. "7] CK Chow and CN Liu, Approximating Discrete Probability Distribu-tions with Dependence Trees, IEEE Trans. Inform. Theory, IT-14, pages 462-467, 1968. 8] Kenneth W. Church and William A. Gale, A Comparison of the Enhanced". In: ().

[23] Ulf Leser and Jörg Hakenberg. "What makes a gene name? Named entity recognition in the biomedical literature". In: *Briefings in bioinformatics* 6.4 (2005), pp. 357–369.

[24] Liyuan Liu et al. "Empower sequence labeling with task-aware neural language model". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[25] Shujie Liu, Nan Yang, Mu Li, and Ming Zhou. "A recursive recurrent neural network for statistical machine translation". In: (2014).

[26] Yanan Lu et al. "CHEMDNER system with mixed conditional random fields and multi-scale word clustering". In: *Journal of cheminformatics* 7.S1 (2015), S4.

[27] Ling Luo et al. "An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition". In: *Bioinformatics* 34.8 (2018), pp. 1381–1388.

[28] Xuezhe Ma and Eduard Hovy. "End-to-end sequence labeling via bi-directional lstm-cnns-crf". In: *arXiv preprint arXiv:1603.01354* (2016).

[29] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. "Maximum Entropy Markov Models for Information Extraction and Segmentation." In: *Icml*. Vol. 17. 2000. 2000, pp. 591–598.

[30] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[31] Juan A Morales-Cordovilla, Victoria Sanchez, and Martin Ratajczak. "Protein alignment based on higher order conditional random fields for template-based modeling". In: *PloS one* 13.6 (2018).

[32] Eric Neufeld. "Judea Pearl. Probabilistic reasoning in intelligent systems: networks of plausible inference. Series in representation and reasoning. Morgan Kaufmann, San Mateo1988, xix+ 552 pp." In: *The Journal of Symbolic Logic* 58.2 (1993), pp. 721–721.

[33] Natalia Ponomareva, Ferran Pla, Antonio Molina, and Paolo Rosso. "Biomedical named entity recognition: a poor knowledge HMM-based approach". In: *International Conference on Application of Natural Language to Information Systems*. Springer. 2007, pp. 382–387.

[34] Lawrence R Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[35] Adam Radziszewski. "A tiered CRF tagger for Polish". In: *Intelligent tools for building a scientific information platform*. Springer, 2013, pp. 215–230.

[36] Mark Schmidt, Kevin Murphy, Glenn Fung, and Rómer Rosales. "Structure learning in random fields for heart motion abnormality detection". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8.

[37] Gökhan Akin Seker and Gülsen Eryigit. "Extending a CRF-based named entity recognition model for Turkish well formed text and user generated content." In: *Semantic Web* 8.5 (2017), pp. 625–642.

[38] Larry Smith et al. "Overview of BioCreative II gene mention recognition". In: *Genome biology* 9.S2 (2008), S2.

[39] Rohini Srihari and Wei Li. "A question answering system supported by information extraction". In: 2000, pp. 166–172. DOI: 10.3115/974147.974170.

[40] Charles Sutton, Andrew McCallum, et al. "An introduction to conditional random fields". In: *Foundations and Trends® in Machine Learning* 4.4 (2012), pp. 267–373.

[41] Lorraine Tanabe and W John Wilbur. "Tagging gene and protein names in biomedical text". In: *Bioinformatics* 18.8 (2002), pp. 1124–1132.

[42] Yoshimasa Tsuruoka and Jun'ichi Tsujii. "Probabilistic term variant generator for biomedical terms". In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. 2003, pp. 167–173.

[43] Douglas L Vail, Manuela M Veloso, and John D Lafferty. "Conditional random fields for activity recognition". In: *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. 2007, pp. 1–8.

[44] Hanna M Wallach. "Conditional random fields: An introduction". In: *Technical Reports (CIS)* (2004), p. 22.

[45] Xuan Wang et al. "Cross-type biomedical named entity recognition with deep multi-task learning". In: *Bioinformatics* 35.10 (2019), pp. 1745–1752.

[46] Haobin Yu. "Named Entity Recognition With Deep Learning". PhD thesis. Auckland University of Technology, 2019.

[47] Yongyue Zhang, Michael Brady, and Stephen Smith. "Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm". In: *IEEE transactions on medical imaging* 20.1 (2001), pp. 45–57.

[48] GuoDong Zhou and Jian Su. "Exploring deep knowledge resources in biomedical name recognition". In: *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*. 2004, pp. 99–102.

[49] Guodong Zhou et al. "Recognizing names in biomedical texts: a machine learning approach". In: *Bioinformatics* 20.7 (2004), pp. 1178–1190.