Hệ thống kiểm tra trùng lặp nội dung

KẾT QUẢ KIỂM TRÙNG TÀI LIỆU

THÔNG TIN TÀI LIỆU

Tên tác giả:	Lê Thanh Hương
Tên file:	Do An DTH.pdf
Thời gian nộp:	02/03/2024 22:06:44
Thời gian trả kết quả:	02/03/2024 22:07:54
Chế độ kiểm tra:	Việt - Việt
Số trang:	47
Số câu:	469
Số câu tương đồng:	443
Mức độ cảnh báo:	CAO (cao: > 15%; trung bình: 2÷15%; thấp: < 2%)

KẾT QUẢ KIỂM TRA TRÙNG LẶP

Độ tương đồng:

94.46%	**94.46%**	**0.64%**	**0.64%**
Trên tất cả tài liệu	Trên tài liệu nội bộ của trường	Trên tài liệu nội bộ của trường khác	Từ nguồn Internet

Nguồn trùng lặp nhiều nhất: 94.243%

Tài liệu hệ thống - dangthihoa_20161600_2.1m.txt

Các loại trừ:
- Các nội dung trước lời nói đầu, lời mở đầu.
- Các câu ít hơn 7 từ.

Kết quả kiểm trùng với tài liệu: Tài liệu hệ thống - dangthihoa_20161600_2.1m.txt

Tỉ lệ sao chép: **94.243%**

Trang	Chỉ số	Tài liệu kiểm tra	Tài liệu gốc
10	1	TỔNG QUAN ĐỀ TÀI 1.1 Đặt vấn đề Hiện nay, chúng ta đang sống trong cuộc cách mạng công nghệ lần thứ 4, cuộc cách mạng của Công nghệ thông tin	TỔNG QUAN ĐỀ TÀI 1.1 Đặt vấn đề Hiện nay, chúng ta đang sống trong cuộc cách mạng công nghệ lần thứ 4, cuộc cách mạng của Công nghệ thông tin
10	2	Trong thời kỳ mà sự bùng nổ và phát triển vô cùng nhanh chóng của các lĩnh vực liên quan đến Internet đặc biệt là lượng thông tin, dữ liệu vô cùng lớn thì nhu cầu tiếp cận thông tin của mọi người được giải quyết một cách cực kỳ dễ dàng	Trong thời kỳ mà sự bùng nổ và phát triển vô cùng nhanh chóng của các lĩnh vực liên quan đến Internet đặc biệt là lượng thông tin, dữ liệu vô cùng lớn thì nhu cầu tiếp cận thông tin của mọi người được giải quyết một cách cực kỳ dễ dàng
10	3	Tuy nhiên, đi liền với lượng thông tin khổng lồ đó thì cũng phát sinh ra rất nhiều vấn đề liên quan đến chất lượng thông tin như tin giả, truyền bá những tư tưởng sai lệch hay đơn giản là sai lỗi chính tả	Tuy nhiên, đi liền với lượng thông tin khổng lồ đó thì cũng phát sinh ra rất nhiều vấn đề liên quan đến chất lượng thông tin như tin giả, truyền bá những tư tưởng sai lệch hay đơn giản là sai lỗi chính tả
10	4	Với sự bùng nổ của công nghệ thông tin hiện nay thì thay cho báo giấy, báo điện tử đã lên ngôi và vô cùng phát triển	Với sự bùng nổ của công nghệ thông tin hiện nay thì thay cho báo giấy, báo điện tử đã lên ngôi và vô cùng phát triển
10	5	Tuy nhiên, cùng với sự phát triển đó là một thực tế “ai cũng có thể thành nhà báo”, nên chất lượng của những bài báo điện tử rất khó có thể đảm bảo	Tuy nhiên, cùng với sự phát triển đó là một thực tế “ai cũng có thể thành nhà báo”, nên chất lượng của những bài báo điện tử rất khó có thể đảm bảo
10	6	Ta có thể dễ dàng bắt gặp những lỗi chính tả trên những bài báo, những tiêu đề, hay thậm chí là cả những bài luận văn, báo cáo khoa học và đồ án do lỗi của người viết viết nhầm và không để ý	Ta có thể dễ dàng bắt gặp những lỗi chính tả trên những bài báo, những tiêu đề, hay thậm chí là cả những bài luận văn, báo cáo khoa học và đồ án do lỗi của người viết viết nhầm và không để ý
10	7	Sai lỗi chính tả tưởng chừng như chỉ là một vấn đề nhỏ, không quá được chú trọng nhưng nó lại đem đến những ảnh hưởng vô cùng to lớn	Sai lỗi chính tả tưởng chừng như chỉ là một vấn đề nhỏ, không quá được chú trọng nhưng nó lại đem đến những ảnh hưởng vô cùng to lớn
10	8	Khi đọc một nội dung hay nhưng lại bị viết sai chính tả, nó giống như việc bạn đang chạy xe bon bon trên đường thì vấp phải ổ gà vậy	Khi đọc một nội dung hay nhưng lại bị viết sai chính tả, nó giống như việc bạn đang chạy xe bon bon trên đường thì vấp phải ổ gà vậy
10	9	Hơn nữa, ta thử nghĩ xem, nếu nội dung của một bài cáo khoa học hay đồ án nếu dính lỗi chính tả thì sẽ bị xem là thiếu chuyên nghiệp và độ đáng tin cũng giảm xuống	Hơn nữa, ta thử nghĩ xem, nếu nội dung của một bài cáo khoa học hay đồ án nếu dính lỗi chính tả thì sẽ bị xem là thiếu chuyên nghiệp và độ đáng tin cũng giảm xuống
10	10	Và đặc biệt là sai chính tả thì rất dễ bị lây, khi ta đọc phải một từ bị viết sai trong thời gian dài thì chính chúng ta có thể bị nhiễm và tưởng đó là từ đúng nên lại viết sai theo	Và đặc biệt là sai chính tả thì rất dễ bị lây, khi ta đọc phải một từ bị viết sai trong thời gian dài thì chính chúng ta có thể bị nhiễm và tưởng đó là từ đúng nên lại viết sai theo
10	11	Chính vì vậy, theo em, sửa lỗi chính tả là một vấn đề cấp bách, rất đáng được quan tâm trong thời kì bùng nổ thông tin như hiện nay	Chính vì vậy, theo em, sửa lỗi chính tả là một vấn đề cấp bách, rất đáng được quan tâm trong thời kì bùng nổ thông tin như hiện nay

10	12	1.2 Tổng quan về chính tả tiếng Việt và các phương pháp đã có Nguồn gốc và đặc điểm của tiếng Việt Tiếng Việt có nguồn gốc rất cổ xưa và đã trải qua một quá trình phát triển lâu dài, đầy sức sống	1.2 Tổng quan về chính tả tiếng Việt và các phương pháp đã có Nguồn gốc và đặc điểm của tiếng Việt Tiếng Việt có nguồn gốc rất cổ xưa và đã trải qua một quá trình phát triển lâu dài, đầy sức sống
10	13	Sức sống đó thể hiện tinh thần dân tộc mạnh mẽ và sáng tạo của nhân dân Việt Nam trong công cuộc đấu tranh anh dũng vì tiền đồ của đất nước, trong sự phấn đấu bền bỉ để xây dựng và phát triển một nền quốc ngữ, quốc văn, quốc học Việt Nam	Sức sống đó thể hiện tinh thần dân tộc mạnh mẽ và sáng tạo của nhân dân Việt Nam trong công cuộc đấu tranh anh dũng vì tiền đồ của đất nước, trong sự phấn đấu bền bỉ để xây dựng và phát triển một nền quốc ngữ, quốc văn, quốc học Việt Nam
10	14	Tiếng Việt là ngôn ngữ của dân tộc Kinh – tộc người đa số ở Việt Nam, được dùng làm phương tiện giao tiếp chung từ lâu trong lịch sử dân tộc	Tiếng Việt là ngôn ngữ của dân tộc Kinh – tộc người đa số ở Việt Nam, được dùng làm phương tiện giao tiếp chung từ lâu trong lịch sử dân tộc
10	15	Tiếng Việt có một nguồn gốc rất đa dạng vì qua 4 ngàn năm, nó đã lai với rất nhiều tiếng Mon, tiếng Khmer, tiếng Thái, tiếng Lào, tiếng Chàm, tiếng Malay, và đã vay mượn rất nhiều từ tiếng Trung	Tiếng Việt có một nguồn gốc rất đa dạng vì qua 4 ngàn năm, nó đã lai với rất nhiều tiếng Mon, tiếng Khmer, tiếng Thái, tiếng Lào, tiếng Chàm, tiếng Malay, và đã vay mượn rất nhiều từ tiếng Trung
10	16	Trong trăm năm vừa qua, tiếng Việt đã mượn hàng trăm từ tiếng Pháp như cái gara, vải kaki, ..	Trong trăm năm vừa qua, tiếng Việt đã mượn hàng trăm từ tiếng Pháp như cái gara, vải kaki, ..
x	17	[1] Hiện nay thì tiếng Việt đã mượn rất nhiều và sử dụng rất tự nhiên, thoải mái hàng ngàn từ tiếng Anh, tiếng Mỹ như computer, battery… sau một thời gian chúng	[1] Hiện nay thì tiếng Việt đã mượn rất nhiều và sử dụng rất tự nhiên, thoải mái hàng ngàn từ tiếng Anh, tiếng Mỹ như computer, battery… sau một thời gian chúng
x	18	11 sẽ được Việt hóa hoàn toàn và trở thành một phần của tiếng Việt luôn	11 sẽ được Việt hóa hoàn toàn và trở thành một phần của tiếng Việt luôn
11	19	Đây là một điều rất hay, nó giúp tiếng Việt trở nên dồi dào hơn, có thêm nhiều từ vựng, nhiều cách nói	Đây là một điều rất hay, nó giúp tiếng Việt trở nên dồi dào hơn, có thêm nhiều từ vựng, nhiều cách nói
11	20	Các đặc điểm của tiếng Việt: - Thường xuyên tiếp xúc, tiếp thu các yếu tố của ngôn ngữ các dân tộc Việt Nam và tiếng Hán, tiếng Pháp, Nga, Anh để làm giàu thêm tiếng Việt; làm cho sự giao lưu giữa người Việt với các tộc người khác trên lãnh thổ ngày càng gắn bó chặt chẽ với nhau	Các đặc điểm của tiếng Việt: - Thường xuyên tiếp xúc, tiếp thu các yếu tố của ngôn ngữ các dân tộc Việt Nam và tiếng Hán, tiếng Pháp, Nga, Anh để làm giàu thêm tiếng Việt; làm cho sự giao lưu giữa người Việt với các tộc người khác trên lãnh thổ ngày càng gắn bó chặt chẽ với nhau
11	21	Chính vì sự giao tiếp thường nhật nên tiếng Việt đã sớm trở thành ngôn ngữ chung của cả nước	Chính vì sự giao tiếp thường nhật nên tiếng Việt đã sớm trở thành ngôn ngữ chung của cả nước
11	22	- Tiếng Việt luôn phát triển, bổ sung vốn từ để đáp ứng mọi phương tiện đời sống chính trị, văn hóa, kinh tế, khoa học, xã hội, kỹ thuật và giáo dục	- Tiếng Việt luôn phát triển, bổ sung vốn từ để đáp ứng mọi phương tiện đời sống chính trị, văn hóa, kinh tế, khoa học, xã hội, kỹ thuật và giáo dục
11	23	- Tiếng Việt có tính thống nhất cao, không kỳ thị phương ngữ	- Tiếng Việt có tính thống nhất cao, không kỳ thị phương ngữ

11	24	Bên cạnh các quy tắc chuẩn sử dụng tiếng nói và chữ viết tiếng Việt, ở các địa phương vẫn tồn tại các yếu tố riêng có tính địa phương về âm điệu, một số từ vị… - Tiếng Việt rất giàu sức sống, dù bị tiếng Hán, tiếng Pháp chèn ép trong hàng trăm năm nhưng không mất đi, trái lại đã tiếp thu và làm phong phú thêm để dần dần đấu tranh giành lại vị trí quốc ngữ của mình	Bên cạnh các quy tắc chuẩn sử dụng tiếng nói và chữ viết tiếng Việt, ở các địa phương vẫn tồn tại các yếu tố riêng có tính địa phương về âm điệu, một số từ vị… - Tiếng Việt rất giàu sức sống, dù bị tiếng Hán, tiếng Pháp chèn ép trong hàng trăm năm nhưng không mất đi, trái lại đã tiếp thu và làm phong phú thêm để dần dần đấu tranh giành lại vị trí quốc ngữ của mình
11	25	Tiếng Việt là loại hình đơn lập	Tiếng Việt là loại hình đơn lập
11	26	Tức là loại ngôn ngữ không có hình thái, từ ngữ không bị biến hình, không bị thay đổi dù ở bất kỳ trạng thái nào	Tức là loại ngôn ngữ không có hình thái, từ ngữ không bị biến hình, không bị thay đổi dù ở bất kỳ trạng thái nào
11	27	Nó mang những đặc trưng sau: - Tiếng là đơn vị cơ sở của ngữ pháp - Từ không bị biến đổi hình thái - Biện pháp biểu thị ý nghĩa ngữ pháp là đảo lộn trật tự sắp xếp của từ hoặc sử dụng hư từ Cấu tạo tiếng, cấu tạo vần, cấu tạo từ trong tiếng Việt • Tiếng: gồm 3 bộ phận: phụ âm đầu, vần và thanh điệu - Tiếng nào cũng có vần và thanh	Nó mang những đặc trưng sau: - Tiếng là đơn vị cơ sở của ngữ pháp - Từ không bị biến đổi hình thái - Biện pháp biểu thị ý nghĩa ngữ pháp là đảo lộn trật tự sắp xếp của từ hoặc sử dụng hư từ Cấu tạo tiếng, cấu tạo vần, cấu tạo từ trong tiếng Việt • Tiếng: gồm 3 bộ phận: phụ âm đầu, vần và thanh điệu - Tiếng nào cũng có vần và thanh
11	28	Có tiếng không có phụ âm đầu	Có tiếng không có phụ âm đầu
11	29	- Tiếng Việt có 6 thanh: thanh ngang (còn gọi là thanh không), thanh huyền, thanh sắc, thanh hỏi, thanh ngã, thanh nặng	- Tiếng Việt có 6 thanh: thanh ngang (còn gọi là thanh không), thanh huyền, thanh sắc, thanh hỏi, thanh ngã, thanh nặng
11	30	- 22 phụ âm: b, c (k, q), ch, d, đ, g (gh), h, kh, l, m, n, nh, ng (ngh), p, ph, r, s, t, tr, th, v, x	- 22 phụ âm: b, c (k, q), ch, d, đ, g (gh), h, kh, l, m, n, nh, ng (ngh), p, ph, r, s, t, tr, th, v, x
11	31	- 11 nguyên âm: i, e, ê, ư, u, o, ô, ơ, a, ă, â	- 11 nguyên âm: i, e, ê, ư, u, o, ô, ơ, a, ă, â
11	32	• Vần: gôm có 3 phần: âm đệm, âm chính, âm cuối	• Vần: gôm có 3 phần: âm đệm, âm chính, âm cuối
11	33	- Âm đệm: + Âm đệm được ghi bằng chữ u và o	- Âm đệm: + Âm đệm được ghi bằng chữ u và o
11	34	• Ghi bằng chữ o khi đứng trước các nguyên âm: a, ă, e • Ghi bằng chữ u khi đứng trước các nguyên âm y, ê, ơ, â	• Ghi bằng chữ o khi đứng trước các nguyên âm: a, ă, e • Ghi bằng chữ u khi đứng trước các nguyên âm y, ê, ơ, â
11	35	+ Âm đệm không xuất hiện sau các phụ âm b, m, v, ph, n, r, g	+ Âm đệm không xuất hiện sau các phụ âm b, m, v, ph, n, r, g
x	36	12 • Sau ph, b: thùng phuy, voan, ô tô, buýt (là từ nước ngoài) • Sau n: thê noa, noãn sào (từ Hán Việt) • Sau r: roàn roạt • Sau g: góa - Âm chính: trong tiếng Việt, nguyên âm nào cũng có thể làm âm chính của tiếng	12 • Sau ph, b: thùng phuy, voan, ô tô, buýt (là từ nước ngoài) • Sau n: thê noa, noãn sào (từ Hán Việt) • Sau r: roàn roạt • Sau g: góa - Âm chính: trong tiếng Việt, nguyên âm nào cũng có thể làm âm chính của tiếng

12	37	+ Các nguyên âm đơn: (11 nguyên âm đã ghi ở trên) + Các nguyên âm đôi: có 3 nguyên âm đôi và được tách thành 8 nguyên âm sau: • iê: - Ghi bằng ia khi phía trước không có âm đệm và phía sau không có âm cuối (Ví dụ: mía, kia,…) - Ghi bằng yê khi phía trước có âm đệm hoặc không có âm nào, phía sau có âm cuối (Ví dụ: yêu, nguyên, chuyên,…) - Ghi bằng ya khi phía trước có âm đệm và phía sau không có âm cuối (Ví dụ: khuya,…) - ghi bằng iê khi phía trước có phụ âm đầu và phía sau có âm cuối (Ví dụ: tiến, kiến,…) • ươ: - Ghi bằng ươ khi sau nó có âm cuối (Ví dụ: mượn,…) - ghi bằng ưa khi phía sau nó không có âm cuối (Ví dụ: mưa,…) • uô: - Ghi bằng uô khi sau nó có âm cuối (Ví dụ: muốn,…) - Ghi bằng ua khi sau nó không có âm cuối (Ví dụ: mua, …) - Âm cuối: + Các phụ âm cuối vần: p, t, c (ch), m, n, ng (nh) + 2 bán âm cuối vần: i (y), u (o) • Các kiểu cấu tạo từ tiếng Việt - Từ đơn: là những từ được cấu tạo bằng một tiếng độc lập	+ Các nguyên âm đơn: (11 nguyên âm đã ghi ở trên) + Các nguyên âm đôi: có 3 nguyên âm đôi và được tách thành 8 nguyên âm sau: • iê: - Ghi bằng ia khi phía trước không có âm đệm và phía sau không có âm cuối (Ví dụ: mía, kia,…) - Ghi bằng yê khi phía trước có âm đệm hoặc không có âm nào, phía sau có âm cuối (Ví dụ: yêu, nguyên, chuyên,…) - Ghi bằng ya khi phía trước có âm đệm và phía sau không có âm cuối (Ví dụ: khuya,…) - ghi bằng iê khi phía trước có phụ âm đầu và phía sau có âm cuối (Ví dụ: tiến, kiến,…) • ươ: - Ghi bằng ươ khi sau nó có âm cuối (Ví dụ: mượn,…) - ghi bằng ưa khi phía sau nó không có âm cuối (Ví dụ: mưa,…) • uô: - Ghi bằng uô khi sau nó có âm cuối (Ví dụ: muốn,…) - Ghi bằng ua khi sau nó không có âm cuối (Ví dụ: mua, …) - Âm cuối: + Các phụ âm cuối vần: p, t, c (ch), m, n, ng (nh) + 2 bán âm cuối vần: i (y), u (o) • Các kiểu cấu tạo từ tiếng Việt - Từ đơn: là những từ được cấu tạo bằng một tiếng độc lập
12	38	Ví dụ: nhà, ghế, bàn, xe, … + Xét về mặt lịch sử: hầu hết từ đơn là những từ đã có từ lâu đời	Ví dụ: nhà, ghế, bàn, xe, … + Xét về mặt lịch sử: hầu hết từ đơn là những từ đã có từ lâu đời
12	39	Một số từ có nguồn gốc thuần Việt, một số từ vay mượn từ các ngôn ngữ nước ngoài như tiếng Hán, tiếng Pháp, tiếng Anh, Nga, … + Xét về mặt ý nghĩa: từ đơn biểu thị những khái niệm cơ bản trong sinh hoạt của đời sống hàng ngày của người Việt, biểu thị các hiện tượng thiên nhiên, các quan hệ gia đình, xã hội, các số đếm,… + Xét về mặt số lượng: tuy không nhiều bằng từ láy và từ ghép nhưng là những từ cơ bản nhất, giữ vai trò quan trọng nhất	Một số từ có nguồn gốc thuần Việt, một số từ vay mượn từ các ngôn ngữ nước ngoài như tiếng Hán, tiếng Pháp, tiếng Anh, Nga, … + Xét về mặt ý nghĩa: từ đơn biểu thị những khái niệm cơ bản trong sinh hoạt của đời sống hàng ngày của người Việt, biểu thị các hiện tượng thiên nhiên, các quan hệ gia đình, xã hội, các số đếm,… + Xét về mặt số lượng: tuy không nhiều bằng từ láy và từ ghép nhưng là những từ cơ bản nhất, giữ vai trò quan trọng nhất
x	40	13 trong việc biểu thị các khái niệm có liên quan đến đời sống và cấu tạo từ mới cho tiếng Việt	13 trong việc biểu thị các khái niệm có liên quan đến đời sống và cấu tạo từ mới cho tiếng Việt
13	41	- Từ ghép: + Từ ghép đẳng lập: từ ghép đẳng lập có những đặc trưng chung: • Quan hệ ngữ pháp giữa các thành tố trong từ là quan hệ bình đẳng	- Từ ghép: + Từ ghép đẳng lập: từ ghép đẳng lập có những đặc trưng chung: • Quan hệ ngữ pháp giữa các thành tố trong từ là quan hệ bình đẳng
13	42	• Xét về mặt quan hệ ý nghĩa giữa các thành tố: hoặc các thành tố đồng nghĩa nhau, hoặc các thành tố gần nghĩa nhau, hoặc các thành tố trái nghĩa nhau	• Xét về mặt quan hệ ý nghĩa giữa các thành tố: hoặc các thành tố đồng nghĩa nhau, hoặc các thành tố gần nghĩa nhau, hoặc các thành tố trái nghĩa nhau
13	43	• Xét về mặt nội dung, từ ghép đẳng lập thường gợi lên nhưng phạm vi sự vật mang ý nghĩa phí các thể hay tổng hợp	• Xét về mặt nội dung, từ ghép đẳng lập thường gợi lên nhưng phạm vi sự vật mang ý nghĩa phí các thể hay tổng hợp
13	44	• Tuy có quan hệ bình đẳng về mặt ngữ pháp, nhưng không đưa đến hệ quả là ý nghĩa từ vựng của các thành tố trong từ đều có giá trị ngang nhau trong mọi trường hợp	• Tuy có quan hệ bình đẳng về mặt ngữ pháp, nhưng không đưa đến hệ quả là ý nghĩa từ vựng của các thành tố trong từ đều có giá trị ngang nhau trong mọi trường hợp
13	45	Những trường hợp một trong hai thành tố phai mờ nghĩa xảy ra phổ biến trong từ ghép đẳng lập	Những trường hợp một trong hai thành tố phai mờ nghĩa xảy ra phổ biến trong từ ghép đẳng lập

13	46	• Căn cứ vào vai trò của các thành tố trong việc tạo nghĩa và phạm vi biểu đạt của từ ghép, có thể phân từ ghép đẳng lập thành ba loại nhỏ là từ ghép đẳng lập gộp nghĩa, từ ghép đẳng lập đơn nghĩa và từ ghép đẳng lập hợp nghĩa	• Căn cứ vào vai trò của các thành tố trong việc tạo nghĩa và phạm vi biểu đạt của từ ghép, có thể phân từ ghép đẳng lập thành ba loại nhỏ là từ ghép đẳng lập gộp nghĩa, từ ghép đẳng lập đơn nghĩa và từ ghép đẳng lập hợp nghĩa
13	47	+ Từ ghép chính phụ: là những từ ghép mà ở đó có ít nhất một thành tố cấu tạo nằm ở vị trí phụ thuộc vào một thành tố cấu tạo khác, tức trong kiểu từ ghép này thường có một yếu tố chính và một yếu tố phụ về mặt ngữ pháp	+ Từ ghép chính phụ: là những từ ghép mà ở đó có ít nhất một thành tố cấu tạo nằm ở vị trí phụ thuộc vào một thành tố cấu tạo khác, tức trong kiểu từ ghép này thường có một yếu tố chính và một yếu tố phụ về mặt ngữ pháp
13	48	Loại này có những đặc điểm sau: • Xét về mặt ý nghĩa, nếu từ ghép đẳng lập có khuynh hướng gợi lên các sự vật, tính chất có ý nghĩa khái quát, tổng hợp, thì kiểu cấu tạo từ này có khuynh hướng nêu lên các sự vật theo mang ý nghĩa cụ thể	Loại này có những đặc điểm sau: • Xét về mặt ý nghĩa, nếu từ ghép đẳng lập có khuynh hướng gợi lên các sự vật, tính chất có ý nghĩa khái quát, tổng hợp, thì kiểu cấu tạo từ này có khuynh hướng nêu lên các sự vật theo mang ý nghĩa cụ thể
13	49	• Trong từ ghép chính phụ, yếu tố chính thường giữ vai trò chỉ loại sự vật, đặc trưng hoặc hoạt động lớn, yếu tố phụ tường được dùng để cụ thể hóa loại sự vật, hoạt động hoặc đặc trưng đó	• Trong từ ghép chính phụ, yếu tố chính thường giữ vai trò chỉ loại sự vật, đặc trưng hoặc hoạt động lớn, yếu tố phụ tường được dùng để cụ thể hóa loại sự vật, hoạt động hoặc đặc trưng đó
13	50	• Căn cứ vào vai trò của các thành tố trong việc tạo nghĩa, có thể chia từ ghép chính phụ thành hai tiểu loại: từ ghép chính phụ dị biệt và từ ghép chính phụ sắc thái hóa	• Căn cứ vào vai trò của các thành tố trong việc tạo nghĩa, có thể chia từ ghép chính phụ thành hai tiểu loại: từ ghép chính phụ dị biệt và từ ghép chính phụ sắc thái hóa
13	51	- Từ láy: giữa các tiếng trong từ láy có quan hệ với nhau về mặt ngữ âm, biểu hiện ở một trong các dạng sau : + Hoặc giống nhau ở phần phụ âm đầu	- Từ láy: giữa các tiếng trong từ láy có quan hệ với nhau về mặt ngữ âm, biểu hiện ở một trong các dạng sau : + Hoặc giống nhau ở phần phụ âm đầu
x	52	14 + Hoặc giống nhai ở phần vần	14 + Hoặc giống nhai ở phần vần
14	53	+ Hoặc giống nhau ở cả phần phụ âm đầu lẫn phần vần	+ Hoặc giống nhau ở cả phần phụ âm đầu lẫn phần vần
14	54	Ví dụ: hao hao, tim tím, ..	Ví dụ: hao hao, tim tím, ..
14	55	- Từ ngẫu hợp: Ngoại trừ các trường hợp trên, còn lại là các từ ngẫu hợp	- Từ ngẫu hợp: Ngoại trừ các trường hợp trên, còn lại là các từ ngẫu hợp
14	56	Đấy là trường hợp mà giữa các tiếng không có quan hệ ngữ âm hay ngữ nghĩa	Đấy là trường hợp mà giữa các tiếng không có quan hệ ngữ âm hay ngữ nghĩa
14	57	Ví dụ: cà phê, a xít, mè nheo, ba hoa,..	Ví dụ: cà phê, a xít, mè nheo, ba hoa,..
14	58	Chính tả tiếng Việt Chính tả là sự chuẩn hóa hình thức chữ viết của ngôn ngữ	Chính tả tiếng Việt Chính tả là sự chuẩn hóa hình thức chữ viết của ngôn ngữ
14	59	Đó là một hệ thống các quy tắc về cách viết các âm vị, âm tiết, từ, cách dùng các dấu câu và lỗi viết hoa… Chuẩn chính tả có những đặc điểm sau đây: - Đặc điểm đầu tiên của chuẩn chính tả là tích chất bắt buộc gần như tuyệt đối của nó	Đó là một hệ thống các quy tắc về cách viết các âm vị, âm tiết, từ, cách dùng các dấu câu và lỗi viết hoa… Chuẩn chính tả có những đặc điểm sau đây: - Đặc điểm đầu tiên của chuẩn chính tả là tích chất bắt buộc gần như tuyệt đối của nó
14	60	Đặc điểm này đòi hỏi người viết bao giờ cũng phải viết đúng chính tả	Đặc điểm này đòi hỏi người viết bao giờ cũng phải viết đúng chính tả
14	61	Chữ viết có thể chưa hợp lí nhưng khi đã được thừa nhận là chuẩn chính tả thì người cầm bút không được tự ý viết khác đi	Chữ viết có thể chưa hợp lí nhưng khi đã được thừa nhận là chuẩn chính tả thì người cầm bút không được tự ý viết khác đi
14	62	Trong chính tả không có sự phân biệt hợp lí – không hợp lí, hay – dở mà chỉ có sự phân biệt đúng – sai, không lỗi – lỗi	Trong chính tả không có sự phân biệt hợp lí – không hợp lí, hay – dở mà chỉ có sự phân biệt đúng – sai, không lỗi – lỗi

14	63	Đối với chính tả, yêu cầu cao nhất là cách viết thống nhất, thống nhất trong mọi văn bản, mọi người, mọi địa phương	Đối với chính tả, yêu cầu cao nhất là cách viết thống nhất, thống nhất trong mọi văn bản, mọi người, mọi địa phương
14	64	- Do chuẩn chính tả có tính chất bắt buộc gần như tuyệt đối cho nên nó ít bị thay đổi như các chuẩn mực khác của ngôn ngữ (như chuẩn ngữ âm, chuẩn từ vựng, chuẩn ngữ pháp)	- Do chuẩn chính tả có tính chất bắt buộc gần như tuyệt đối cho nên nó ít bị thay đổi như các chuẩn mực khác của ngôn ngữ (như chuẩn ngữ âm, chuẩn từ vựng, chuẩn ngữ pháp)
14	65	Nói cách khác, chuẩn chính tả có tính chất ổn định, tính chất cố hữu khá rõ	Nói cách khác, chuẩn chính tả có tính chất ổn định, tính chất cố hữu khá rõ
14	66	Sự tồn tại nhất nhất hàng thế kỉ của nó đã tạo nên ấn tượng về một cái gì đó “bất di bất dịch”, một tâm lí rất bảo thủ	Sự tồn tại nhất nhất hàng thế kỉ của nó đã tạo nên ấn tượng về một cái gì đó “bất di bất dịch”, một tâm lí rất bảo thủ
14	67	- Ngữ âm phát triển, chính tả không thể giữ mãi tính chất cố hữu của mình mà dần dần cũng có một sự biết động nhất định	- Ngữ âm phát triển, chính tả không thể giữ mãi tính chất cố hữu của mình mà dần dần cũng có một sự biết động nhất định
14	68	Do đó, bên cạnh chuẩn mực chính tả hiện có lại có thể xuất hiện một cách viết mới tồn tại song song với nó	Do đó, bên cạnh chuẩn mực chính tả hiện có lại có thể xuất hiện một cách viết mới tồn tại song song với nó
14	69	Tình trạng có nhiều cách viết như vậy đòi hỏi phải tiến hành chuẩn hóa chính tả	Tình trạng có nhiều cách viết như vậy đòi hỏi phải tiến hành chuẩn hóa chính tả
14	70	Các nguyên nhân gây ra lỗi chính tả Có nhiều nguyên nhân khác nhau gây ra lỗi chính tả, nhưng ta có thể tổng hợp thành một số nguyên nhân như sau: - Nguyên nhân do gõ từ sai: lỗi này xảy ra có thể do gõ sai/ thiếu/ thừa phím, hay do cách cài đặt bàn phím, loại bàn phím, do quy tắc gõ tiếng Việt của các kiểu gõ khác nhau (Telex, VNI, Unicode,...)	Các nguyên nhân gây ra lỗi chính tả Có nhiều nguyên nhân khác nhau gây ra lỗi chính tả, nhưng ta có thể tổng hợp thành một số nguyên nhân như sau: - Nguyên nhân do gõ từ sai: lỗi này xảy ra có thể do gõ sai/ thiếu/ thừa phím, hay do cách cài đặt bàn phím, loại bàn phím, do quy tắc gõ tiếng Việt của các kiểu gõ khác nhau (Telex, VNI, Unicode,...)
14	71	- Nguyên nhân do phát âm sai: lỗi này do sự nhầm lẫn giữa cách đọc và cách viết của những từ đồng âm hoặc có âm gần giống nhau dân đến viết sai (như lỗi sai âm đầu tr/ch, d/r/gi, s/x,lỗi dấu hỏi/ ngã,...)	- Nguyên nhân do phát âm sai: lỗi này do sự nhầm lẫn giữa cách đọc và cách viết của những từ đồng âm hoặc có âm gần giống nhau dân đến viết sai (như lỗi sai âm đầu tr/ch, d/r/gi, s/x,lỗi dấu hỏi/ ngã,...)
14	72	Ở Việt Nam, do có nhiều khác biệt về cách phát âm giữa các vùng miền trong khi hệ thống chữ viết lại dựa trên đặc trưng phát âm của Hà Nội nên dễ dẫn đến lỗi sai này	Ở Việt Nam, do có nhiều khác biệt về cách phát âm giữa các vùng miền trong khi hệ thống chữ viết lại dựa trên đặc trưng phát âm của Hà Nội nên dễ dẫn đến lỗi sai này
x	73	15 - Nguyên nhân do sử dụng từ vựng sai: lỗi này do sử dụng từ sai với ý nghĩa thực của nó	15 - Nguyên nhân do sử dụng từ vựng sai: lỗi này do sử dụng từ sai với ý nghĩa thực của nó
15	74	Đây là lỗi do vốn từ vựng của người viết, nhưng nhiều khi vẫn đòi hỏi trình bắt lỗi chính tả phải tìm ra được những lỗi này	Đây là lỗi do vốn từ vựng của người viết, nhưng nhiều khi vẫn đòi hỏi trình bắt lỗi chính tả phải tìm ra được những lỗi này
15	75	- Các nguyên nhân khác: ngoài ra còn các loại lỗi chính tả khác như viết hoa, viết tên riêng, thuật ngữ không đúng quy cách, ..	- Các nguyên nhân khác: ngoài ra còn các loại lỗi chính tả khác như viết hoa, viết tên riêng, thuật ngữ không đúng quy cách, ..
15	76	Các loại lỗi chính tả tiếng Việt Có nhiều cách phân loại lỗi chính tả khác nhau	Các loại lỗi chính tả tiếng Việt Có nhiều cách phân loại lỗi chính tả khác nhau
15	77	Tuy nhiên, trong đồ án này em sẽ phân lỗi chính tả thành hai loại: - Lỗi tạo từ sai, hoàn toàn không có từ đó trong từ điển tiếng Việt	Tuy nhiên, trong đồ án này em sẽ phân lỗi chính tả thành hai loại: - Lỗi tạo từ sai, hoàn toàn không có từ đó trong từ điển tiếng Việt
15	78	Đây là loại lỗi dễ phát hiện hơn	Đây là loại lỗi dễ phát hiện hơn

15	79	(Ví dụ: : “ko”, “đượk”) - Lỗi chính tả mà từ/ tiếng đó có trong từ điển	(Ví dụ: : “ko”, “đượk”) - Lỗi chính tả mà từ/ tiếng đó có trong từ điển
15	80	Nếu không dựa vào ngữ cảnh xung quanh thì không thể xác định đó có phải là lỗi chính tả hay không	Nếu không dựa vào ngữ cảnh xung quanh thì không thể xác định đó có phải là lỗi chính tả hay không
15	81	(Ví dụ: “Chúng ta không lên đi đường này” – từ “lên” có trong từ điển nhưng trong câu này nó không đúng mà ta nên sử dụng từ “nên”) Ngoài ra còn có thể phân loại lỗi theo nguồn gốc phát sinh lỗi	(Ví dụ: “Chúng ta không lên đi đường này” – từ “lên” có trong từ điển nhưng trong câu này nó không đúng mà ta nên sử dụng từ “nên”) Ngoài ra còn có thể phân loại lỗi theo nguồn gốc phát sinh lỗi
15	82	Theo cách phân loại này, có hai loại lỗi đó là lỗi nhập sai và lỗi phát âm sai	Theo cách phân loại này, có hai loại lỗi đó là lỗi nhập sai và lỗi phát âm sai
15	83	- Lỗi nhập sai là lỗi gây ra do gõ sai phím, gõ sót hoặc dư phím	- Lỗi nhập sai là lỗi gây ra do gõ sai phím, gõ sót hoặc dư phím
15	84	- Lỗi phát âm sai là do sự nhầm lẫn giữa cách đọc và cách viết giữa những từ đồng âm hoặc gần với nhau	- Lỗi phát âm sai là do sự nhầm lẫn giữa cách đọc và cách viết giữa những từ đồng âm hoặc gần với nhau
15	85	1.3 Một số nghiên cứu liên quan Phương pháp phát hiện và sửa lỗi chính tả tiếng Việt sử dụng mô hình từ điển Đây là phương pháp dựa trên tư tưởng vét cạn	15 Phương pháp phát hiện và sửa lỗi chính tả tiếng Việt sử dụng mô hình từ điển ....
15	85	1.3 Một số nghiên cứu liên quan Phương pháp phát hiện và sửa lỗi chính tả tiếng Việt sử dụng mô hình từ điển Đây là phương pháp dựa trên tư tưởng vét cạn	1.3 Một số nghiên cứu liên quan Phương pháp phát hiện và sửa lỗi chính tả tiếng Việt sử dụng mô hình từ điển Đây là phương pháp dựa trên tư tưởng vét cạn
15	86	Tức là với một từ được xem là đúng chính tả, tác giả sẽ phát sinh và lưu trữ lại tất cả các trường hợp bị sai chính tả có thể có của từ đó và tất cả được lưu trữ trong một bộ từ điển	Tức là với một từ được xem là đúng chính tả, tác giả sẽ phát sinh và lưu trữ lại tất cả các trường hợp bị sai chính tả có thể có của từ đó và tất cả được lưu trữ trong một bộ từ điển
15	87	Sau này khi mà gặp một từ có dạng sai tương tự như trường hợp đã lưu trữ, chương trình sẽ báo là có lỗi và đưa ra gợi ý từ gốc của từ đó	Sau này khi mà gặp một từ có dạng sai tương tự như trường hợp đã lưu trữ, chương trình sẽ báo là có lỗi và đưa ra gợi ý từ gốc của từ đó
15	88	- Nhược điểm: - Rất tốn công và thời gian để có thể tạo lên bộ từ điển	- Nhược điểm: - Rất tốn công và thời gian để có thể tạo lên bộ từ điển
15	89	- Do bộ từ điển được tạo thủ công nên khó có thể bao quát hết được tất cả các lỗi	- Do bộ từ điển được tạo thủ công nên khó có thể bao quát hết được tất cả các lỗi
15	90	- Hiệu quả với loại lỗi tạo từ sai, hoàn toàn không có từ đó trong từ điển tiếng Việt và không hiệu quả với loại lỗi chính tả mà từ/ tiếng đó có trong từ điển do không nắm được nội dung câu	- Hiệu quả với loại lỗi tạo từ sai, hoàn toàn không có từ đó trong từ điển tiếng Việt và không hiệu quả với loại lỗi chính tả mà từ/ tiếng đó có trong từ điển do không nắm được nội dung câu
15	91	Kiểm tra lỗi chính tả tiếng Việt dựa trên luật cấu tạo âm tiết [2] • Cấu trúc âm tiết 3 thành phần: - Chúng ta có thể phân tích một âm tiết thành ba thành phần sau: + Âm đầu + Tổ hợp âm giữa + Âm cuối	Kiểm tra lỗi chính tả tiếng Việt dựa trên luật cấu tạo âm tiết [2] • Cấu trúc âm tiết 3 thành phần: - Chúng ta có thể phân tích một âm tiết thành ba thành phần sau: + Âm đầu + Tổ hợp âm giữa + Âm cuối
x	92	16 - Cấu trúc của một âm tiết theo cách tiếp cận 3 thành phần sẽ được viết lại như sau: Âm tiết = [Âm đầu][Âm cuối] Trong đó những thành phần nằm trong cặp dấu <> là bắt buộc phải có, những thành phần nằm trong cặp dấu [ ] thì có thể có hoặc không	16 - Cấu trúc của một âm tiết theo cách tiếp cận 3 thành phần sẽ được viết lại như sau: Âm tiết = [Âm đầu][Âm cuối] Trong đó những thành phần nằm trong cặp dấu <> là bắt buộc phải có, những thành phần nằm trong cặp dấu [ ] thì có thể có hoặc không

16	93	• Tổ chức lưu trữ luật âm tiết Dựa trên những phân tích về âm tiết 3 thành phần, chúng ta có thể tổ chức lưu trữ từ điển luật theo Tổ hợp âm giữa trên file dữ liệu như sau: Structure CT_AM Tong_Am_Dau : LongInt To_Hop_Am_Giua : String(3) Tong_Am_Cuoi : LongInt End Structure Trong đó: Tong_Am_Dau là giá trị tổng của các Âm đầu có thể đi với tổ hợp âm giữa Tong_Am_Cuoi là giá trị tổng của các Âm cuối có thể đi với tổ hợp âm giữa Lưu cấu trúc âm này (có sắp xếp) thành một từ điển các cấu trúc âm để sau này chúng ta kiểm tra các âm tiết ở trong từ điển	• Tổ chức lưu trữ luật âm tiết Dựa trên những phân tích về âm tiết 3 thành phần, chúng ta có thể tổ chức lưu trữ từ điển luật theo Tổ hợp âm giữa trên file dữ liệu như sau: Structure CT_AM Tong_Am_Dau : LongInt To_Hop_Am_Giua : String(3) Tong_Am_Cuoi : LongInt End Structure Trong đó: Tong_Am_Dau là giá trị tổng của các Âm đầu có thể đi với tổ hợp âm giữa Tong_Am_Cuoi là giá trị tổng của các Âm cuối có thể đi với tổ hợp âm giữa Lưu cấu trúc âm này (có sắp xếp) thành một từ điển các cấu trúc âm để sau này chúng ta kiểm tra các âm tiết ở trong từ điển
16	94	• Thuật toán kiểm tra một âm tiết có đúng hay không - Đầu vào: một âm tiết - Đầu ra: Âm tiết đúng chính tả hay sai	• Thuật toán kiểm tra một âm tiết có đúng hay không - Đầu vào: một âm tiết - Đầu ra: Âm tiết đúng chính tả hay sai
16	95	- Phương pháp: + Bước 1: Tách âm tiết ra làm 3 phần: âm đầu, tổ hợp âm giữa, âm cuối và chuyển thành một cấu trúc âm tiết X, tương ứng theo âm đầu, tổ hợp âm giữa và âm cuối	- Phương pháp: + Bước 1: Tách âm tiết ra làm 3 phần: âm đầu, tổ hợp âm giữa, âm cuối và chuyển thành một cấu trúc âm tiết X, tương ứng theo âm đầu, tổ hợp âm giữa và âm cuối
16	96	+ Bước 2: Tìm tổ hợp âm giữa trong từ điển theo phương pháp tìm kiếm nhị phân	+ Bước 2: Tìm tổ hợp âm giữa trong từ điển theo phương pháp tìm kiếm nhị phân
16	97	+ Bước 3: Nếu tìm thấy thì tiếp tục bước 4, nếu không thì nhảy đến bước 6	+ Bước 3: Nếu tìm thấy thì tiếp tục bước 4, nếu không thì nhảy đến bước 6
16	98	+ Bước 4: Ta lấy được một cấu trúc âm tiết CTAM tương ứng trong từ điển	+ Bước 4: Ta lấy được một cấu trúc âm tiết CTAM tương ứng trong từ điển
16	99	+ Bước 5: Kiểm tra xem âm đầu, âm cuối của X có trong trong cấu trúc âm tiết CTAM đó hay không	+ Bước 5: Kiểm tra xem âm đầu, âm cuối của X có trong trong cấu trúc âm tiết CTAM đó hay không
16	100	Nếu có thì kết luận là âm tiết đúng, nhảy đến bước 7	Nếu có thì kết luận là âm tiết đúng, nhảy đến bước 7
16	101	+ Bước 6: Kết luận âm tiết sai	+ Bước 6: Kết luận âm tiết sai
16	102	+ Bước 7: Kết thúc Việc kiểm tra toàn bộ các âm tiết của văn bản là việc kiểm tra tất cả các âm tiết có trong từ điển hay không	+ Bước 7: Kết thúc Việc kiểm tra toàn bộ các âm tiết của văn bản là việc kiểm tra tất cả các âm tiết có trong từ điển hay không
16	103	Với phương pháp này chúng ta kiểm tra được tất cả các âm tiết trong văn bản có đúng chính tả hay không	Với phương pháp này chúng ta kiểm tra được tất cả các âm tiết trong văn bản có đúng chính tả hay không
x	104	17 • Ưu điểm: - Phương pháp này tiết kiệm được không gian lưu trữ từ điển, số cấu trúc lưu trữ bằng số tổ hợp âm giữa của tiếng Việt, số lượng này không nhiều (khoảng 700 cấu trúc)	17 • Ưu điểm: - Phương pháp này tiết kiệm được không gian lưu trữ từ điển, số cấu trúc lưu trữ bằng số tổ hợp âm giữa của tiếng Việt, số lượng này không nhiều (khoảng 700 cấu trúc)
17	105	- Do số lượng cấu trúc âm tiết nhỏ nên việc tìm kiếm rất nhanh, với phương pháp tìm kiếm nhị phân thi tốc độ tìm kiếm là log2(n) (n là số cấu trúc âm tiết)	- Do số lượng cấu trúc âm tiết nhỏ nên việc tìm kiếm rất nhanh, với phương pháp tìm kiếm nhị phân thi tốc độ tìm kiếm là log2(n) (n là số cấu trúc âm tiết)
17	106	• Nhược điểm: - Chưa đưa được yếu tố ngữ cảnh trong câu vào để giải quyết nên chỉ có tác dụng cho bài toán lỗi tạo từ sai, hoàn toàn không có từ đó trong từ điển tiếng Việt	• Nhược điểm: - Chưa đưa được yếu tố ngữ cảnh trong câu vào để giải quyết nên chỉ có tác dụng cho bài toán lỗi tạo từ sai, hoàn toàn không có từ đó trong từ điển tiếng Việt

17	107	Phương pháp kiểm lỗi chính tả tiếng Việt dựa trên n-gram [3] Thuật toán kiểm lỗi bao gồm 2 bước: - Sinh tập nhầm lẫn âm tiết: tạo tập nhầm lẫn âm tiết dựa trên lỗi đánh máy cho từng phần của âm tiết, sau đó tạo tập nhầm lẫn âm tiết do lỗi phát âm	Phương pháp kiểm lỗi chính tả tiếng Việt dựa trên n-gram [3] Thuật toán kiểm lỗi bao gồm 2 bước: - Sinh tập nhầm lẫn âm tiết: tạo tập nhầm lẫn âm tiết dựa trên lỗi đánh máy cho từng phần của âm tiết, sau đó tạo tập nhầm lẫn âm tiết do lỗi phát âm
17	108	Kết hợp chúng ta được tập nhầm lẫn âm tiết	Kết hợp chúng ta được tập nhầm lẫn âm tiết
17	109	- Sử dụng n-gram lựa chọn các âm tiết tốt nhất trong tập nhầm lẫn âm tiết + dựa vào tập nhầm lẫn âm tiết của 3 âm tiết đầu tiên, tính xác suất tốt nhất trong tất cả các tổ hợp 3 âm tiết có thể từ 3 tập nhầm lẫn âm tiết của chúng	- Sử dụng n-gram lựa chọn các âm tiết tốt nhất trong tập nhầm lẫn âm tiết + dựa vào tập nhầm lẫn âm tiết của 3 âm tiết đầu tiên, tính xác suất tốt nhất trong tất cả các tổ hợp 3 âm tiết có thể từ 3 tập nhầm lẫn âm tiết của chúng
17	110	Công thức tính xác suất: P(s1s2s3) = P(s1).P(s2|s1).P(s3|s1s2) + Từ 3 âm tiết đã được lựa chọn và được cho là đúng này, ta lần lượt tính xác suất cho từng âm tiết tiếp theo: s4, s5,… P(s2s3s4) = P(s1).P(s3|s2).P(s4|s2s3) • Ưu điểm: - Cài đặt đơn giản và hoàn toàn áp dụng tư tưởng của mô hình n-gram - Đã đưa được yếu tố ngữ cảnh vào khi thực hiện	Công thức tính xác suất: P(s1s2s3) = P(s1).P(s2|s1).P(s3|s1s2) + Từ 3 âm tiết đã được lựa chọn và được cho là đúng này, ta lần lượt tính xác suất cho từng âm tiết tiếp theo: s4, s5,… P(s2s3s4) = P(s1).P(s3|s2).P(s4|s2s3) • Ưu điểm: - Cài đặt đơn giản và hoàn toàn áp dụng tư tưởng của mô hình n-gram - Đã đưa được yếu tố ngữ cảnh vào khi thực hiện
17	111	• Nhược điểm: - Tốc độ thực hiện chậm do phải tính xác suất của tất cả các tổ hợp của âm tiết đầu - Độ chính xác chưa cao	• Nhược điểm: - Tốc độ thực hiện chậm do phải tính xác suất của tất cả các tổ hợp của âm tiết đầu - Độ chính xác chưa cao
17	112	1.4 Mục tiêu đồ án Nhược điểm của phương pháp kiểm lỗi chính tả tiếng Việt sử dụng mô hình từ điển là tốn công và thời gian và trong các cách tiếp cận trên hầu hết việc sử dụng thông tin ngữ cảnh vào việc sửa lỗi chính tả còn rất ít hoặc không đạt kết quả như mong đợi	1.4 Mục tiêu đồ án Nhược điểm của phương pháp kiểm lỗi chính tả tiếng Việt sử dụng mô hình từ điển là tốn công và thời gian và trong các cách tiếp cận trên hầu hết việc sử dụng thông tin ngữ cảnh vào việc sửa lỗi chính tả còn rất ít hoặc không đạt kết quả như mong đợi
17	113	Do đó, nghiên cứu và phát triển một ứng dụng phát hiện và sửa lỗi chính tả tiếng Việt sử dụng thông tin ngữ cảnh sẽ giúp cho việc sửa lỗi chính tả đạt hiệu quả cao hơn	Do đó, nghiên cứu và phát triển một ứng dụng phát hiện và sửa lỗi chính tả tiếng Việt sử dụng thông tin ngữ cảnh sẽ giúp cho việc sửa lỗi chính tả đạt hiệu quả cao hơn
17	114	Đồ án này hướng tới việc tìm hiểu và ứng dụng kiểm lỗi chính tả tiếng Việt mức độ tiếng dựa vào thông tin ngữ cảnh, sử dụng phương pháp học máy trên mô hình mạng nơron	Đồ án này hướng tới việc tìm hiểu và ứng dụng kiểm lỗi chính tả tiếng Việt mức độ tiếng dựa vào thông tin ngữ cảnh, sử dụng phương pháp học máy trên mô hình mạng nơron
17	115	Nhờ khả năng học, chương trình có thể thích ứng được với sự	Nhờ khả năng học, chương trình có thể thích ứng được với sự
x	116	18 thay đổi không ngừng của ngôn ngữ mà không tốn quá nhiều công sức của con người	18 thay đổi không ngừng của ngôn ngữ mà không tốn quá nhiều công sức của con người
18	117	Bài toán có dữ liệu đầu vào là một câu sai chính tả và đầu ra là một câu đúng chuẩn chính tả nên ta thấy mô hình dạng seq2seq sẽ thích hợp cho bài toán	Bài toán có dữ liệu đầu vào là một câu sai chính tả và đầu ra là một câu đúng chuẩn chính tả nên ta thấy mô hình dạng seq2seq sẽ thích hợp cho bài toán
18	118	Với các mô hình thường được sử dụng nhiều trong các bài toán xử lí ngôn ngữ tự nhiên (NLP) như RNN, LSTM, em thấy chúng có nhược điểm là: - RNN: - Vanishing gradient: hiện tượng gradient sẽ bị nhỏ lại tới mức gần như biến mất ở những hidden state cuối khi input là một chuỗi dài	Với các mô hình thường được sử dụng nhiều trong các bài toán xử lí ngôn ngữ tự nhiên (NLP) như RNN, LSTM, em thấy chúng có nhược điểm là: - RNN: - Vanishing gradient: hiện tượng gradient sẽ bị nhỏ lại tới mức gần như biến mất ở những hidden state cuối khi input là một chuỗi dài

18	119	- Exploding gradient: hiện tượng gradient quá lớn do tích tụ gradient ở những lớp cuối đặc biệt hay xảy ra đối với câu dài	- Exploding gradient: hiện tượng gradient quá lớn do tích tụ gradient ở những lớp cuối đặc biệt hay xảy ra đối với câu dài
18	120	- LSTM: là bước cải tiến lớn so với RNN, đã cải thiện được những nhược điểm của RNN, nhưng với những câu quá dài vẫn sẽ gặp phải hiện tượng vanishing gradient	- LSTM: là bước cải tiến lớn so với RNN, đã cải thiện được những nhược điểm của RNN, nhưng với những câu quá dài vẫn sẽ gặp phải hiện tượng vanishing gradient
18	121	- RNN và LSTM đều xử lý các từ tuần tự nên tốc độ tính toán chưa được cải thiện	- RNN và LSTM đều xử lý các từ tuần tự nên tốc độ tính toán chưa được cải thiện
18	122	Do đó, trong đồ án này, em chọn mô hình Transformer vì nó kết hợp điểm mạnh của CNN (Convolutional Neural Network – một giải pháp cho tính toán song song) và Attention nên vừa có thể cải thiện tốc độ tính toán, vừa không lo hiện tượng vanishing gradient do cơ chế tính toán song song	Do đó, trong đồ án này, em chọn mô hình Transformer vì nó kết hợp điểm mạnh của CNN (Convolutional Neural Network – một giải pháp cho tính toán song song) và Attention nên vừa có thể cải thiện tốc độ tính toán, vừa không lo hiện tượng vanishing gradient do cơ chế tính toán song song
18	123	Trong khuôn khổ đồ án này, em đã chọn sử dụng bộ công cụ mã nguồn mở OpenNMT-py để cài đặt mô hình Transformer cho bài toán vì những lý do sau: - Thân thiện với người sử dụng và đã chế độ (multimodal) - Thừa hưởng tính dễ sử dụng của Pytorch	Trong khuôn khổ đồ án này, em đã chọn sử dụng bộ công cụ mã nguồn mở OpenNMT-py để cài đặt mô hình Transformer cho bài toán vì những lý do sau: - Thân thiện với người sử dụng và đã chế độ (multimodal) - Thừa hưởng tính dễ sử dụng của Pytorch
19	124	CƠ SỞ LÝ THUYẾT 2.1 Dịch máy mạng nơron (NMT) NMT (Neural Machine Translation) là sự kết hợp của dịch máy (Machine Translation - MT) và mạng nơron nhân tạo (Artificial Neural Network - NN)	CƠ SỞ LÝ THUYẾT 2.1 Dịch máy mạng nơron (NMT) NMT (Neural Machine Translation) là sự kết hợp của dịch máy (Machine Translation - MT) và mạng nơron nhân tạo (Artificial Neural Network - NN)
19	125	Khởi nguồn của MT hoạt động theo cách chia nhỏ câu thành các cụm từ và tiến hành dịch trên từng cụm từ một	Khởi nguồn của MT hoạt động theo cách chia nhỏ câu thành các cụm từ và tiến hành dịch trên từng cụm từ một
19	126	Kết quả cuối cùng sẽ là một câu ghép lại từ các cụm từ đã được dịch	Kết quả cuối cùng sẽ là một câu ghép lại từ các cụm từ đã được dịch
19	127	Cách tiếp cận này được gọi là dịch theo cụm (pharase-based), và kết quả thì không được ấn tượng lắm vì cách tiếp cận của phương pháp này không thực sự giống với cách mà con người sử dụng trong dịch thuật là đọc toàn bộ câu, nắm ý nghĩa của câu và đưa ra câu dịch tương ứng	Cách tiếp cận này được gọi là dịch theo cụm (pharase-based), và kết quả thì không được ấn tượng lắm vì cách tiếp cận của phương pháp này không thực sự giống với cách mà con người sử dụng trong dịch thuật là đọc toàn bộ câu, nắm ý nghĩa của câu và đưa ra câu dịch tương ứng
19	128	Và NMT được xây dựng hoàn toàn dựa trên cách làm này	Và NMT được xây dựng hoàn toàn dựa trên cách làm này
19	129	NMT là cách tiếp cận MT phổ biến trong khoảng 4 năm gần đây và đã cho ra các kết quả thực sự tốt, tới mức ngang hoặc hơn cả con người Cụ thể về kiến trúc thì NMT là sự kết hợp của 2 thành phần chính là Seq2Seq và Attention 2.2 Mạng nơron nhân tạo Mạng nơron nhân tạo [4] là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơron sinh học	NMT là cách tiếp cận MT phổ biến trong khoảng 4 năm gần đây và đã cho ra các kết quả thực sự tốt, tới mức ngang hoặc hơn cả con người Cụ thể về kiến trúc thì NMT là sự kết hợp của 2 thành phần chính là Seq2Seq và Attention 2.2 Mạng nơron nhân tạo Mạng nơron nhân tạo [4] là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơron sinh học
19	130	Hình 2.1 cho thấy một mạng nơron đơn giản được tạo nên bởi tầng vào, tầng ra và tầng ẩn	Hình 2.1 cho thấy một mạng nơron đơn giản được tạo nên bởi tầng vào, tầng ra và tầng ẩn

19	131	Từ “ẩn” có nghĩa là chúng ta có thể quan sát đầu vào và đầu ra trong khi cấu trúc kết nối chúng vẫn bị ẩn	Từ “ẩn” có nghĩa là chúng ta có thể quan sát đầu vào và đầu ra trong khi cấu trúc kết nối chúng vẫn bị ẩn
19	132	Hình 2-1: Mô hình mạng nơron đơn giản Mạng nơron nhân tạo được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó	Hình 2-1: Mô hình mạng nơron đơn giản Mạng nơron nhân tạo được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó
19	133	Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện	Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện
19	134	Về bản chất “học” chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron	Về bản chất “học” chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron
x	135	20 Lấy ý tưởng từ nơron sinh học của não người, một nơron có thể nhận nhiều đầu vào và cho ra một kết quả duy nhất	20 Lấy ý tưởng từ nơron sinh học của não người, một nơron có thể nhận nhiều đầu vào và cho ra một kết quả duy nhất
x	136	Hình 2-2: Ví dụ về một mạng nơron nhân tạo Mỗi nơron sẽ nhận một hoặc nhiều đầu vào x dạng nhị phân và cho ra một kết quả a dạng nhị phân duy nhất	Hình 2-2: Ví dụ về một mạng nơron nhân tạo Mỗi nơron sẽ nhận một hoặc nhiều đầu vào x dạng nhị phân và cho ra một kết quả a dạng nhị phân duy nhất
x	137	Các đầu vào sẽ được điều chỉnh sự ảnh hưởng nhờ trọng số w của nó, còn kết quả đầu ra sẽ được quyết định dựa trên một ngưỡng b nào đó	Các đầu vào sẽ được điều chỉnh sự ảnh hưởng nhờ trọng số w của nó, còn kết quả đầu ra sẽ được quyết định dựa trên một ngưỡng b nào đó
x	138	□ = { 0 □ế□ ∑□□□□ □ ≤ □□□□□□□□□□ 1 □ế□ ∑□□□□ □ > □□□□□□□□□□ Đặt b = -threshold biểu thức trên có thể viết lại thành: □ = { 0 □ế□ ∑□□□□ □ + □ ≤ 0 1 □ế□ ∑□□□□ □ + □ > 0 Với đầu vào và đầu ra dạng nhị phân, rất khó để điều chỉnh một lượng nhỏ đầu vào để đầu ra thay đổi chút ít, do vậy để linh động, chúng ta có thể mở rộng chúng ra cả khoảng [0, 1]	□ = { 0 □ế□ ∑□□□□ □ ≤ □□□□□□□□□□ 1 □ế□ ∑□□□□ □ > □□□□□□□□□□ Đặt b = -threshold biểu thức trên có thể viết lại thành: □ = { 0 □ế□ ∑□□□□ □ + □ ≤ 0 1 □ế□ ∑□□□□ □ + □ > 0 Với đầu vào và đầu ra dạng nhị phân, rất khó để điều chỉnh một lượng nhỏ đầu vào để đầu ra thay đổi chút ít, do vậy để linh động, chúng ta có thể mở rộng chúng ra cả khoảng [0, 1]
x	139	Lúc này chúng ta sử dụng một hàm đặc biệt, gọi là hàm kích hoạt để giới hạn giá trị đầu ra	Lúc này chúng ta sử dụng một hàm đặc biệt, gọi là hàm kích hoạt để giới hạn giá trị đầu ra
x	140	Các hàm kích hoạt thường rất đa dạng, có thể là hàm tuyến tính hoặc phi tuyến	Các hàm kích hoạt thường rất đa dạng, có thể là hàm tuyến tính hoặc phi tuyến
x	141	Việc lựa chọn hàm kích hoạt nào là tuỳ thuộc vào từng bài toán và kinh nghiệm của người thiết kế mạng	Việc lựa chọn hàm kích hoạt nào là tuỳ thuộc vào từng bài toán và kinh nghiệm của người thiết kế mạng
x	142	Một số hàm kích hoạt thường sử dụng trong các mô hình mạng nơron được đưa ra trong hình dưới	Một số hàm kích hoạt thường sử dụng trong các mô hình mạng nơron được đưa ra trong hình dưới
x	143	21 Hình 2-3: Một số hàm kích hoạt thông dụng Như vậy nơron nhân tạo nhận các tín hiệu đầu vào, xử lý (nhân các tín hiệu này với trọng số liên kết, tính tổng các tích thu được rồi gửi kết quả tới hàm kích hoạt), và cho một tín hiệu đầu ra (là kết quả của hàm kích hoạt)	21 Hình 2-3: Một số hàm kích hoạt thông dụng Như vậy nơron nhân tạo nhận các tín hiệu đầu vào, xử lý (nhân các tín hiệu này với trọng số liên kết, tính tổng các tích thu được rồi gửi kết quả tới hàm kích hoạt), và cho một tín hiệu đầu ra (là kết quả của hàm kích hoạt)
21	144	Một mạng nơron là sự kết hợp của các tầng nơron như hình 3.3	Một mạng nơron là sự kết hợp của các tầng nơron như hình 3.3

21	145	Dựa trên cách thức liên kết các nơron người ta chia làm hai loại: Mạng nơron truyền thẳng và mạng nơron hồi quy	Dựa trên cách thức liên kết các nơron người ta chia làm hai loại: Mạng nơron truyền thẳng và mạng nơron hồi quy
21	146	Mạng nơron truyền thẳng Mạng nơron truyền thẳng là một mạng nơron nhân tạo, trong đó các kết nối giữa các nút không tạo thành chu kỳ	Mạng nơron truyền thẳng Mạng nơron truyền thẳng là một mạng nơron nhân tạo, trong đó các kết nối giữa các nút không tạo thành chu kỳ
21	147	Mô hình mạng nơron chuyển tiếp là hình thức đơn giản nhất của mạng nơron vì thông tin chỉ được xử lý theo một hướng	Mô hình mạng nơron chuyển tiếp là hình thức đơn giản nhất của mạng nơron vì thông tin chỉ được xử lý theo một hướng
21	148	Mặc dù dữ liệu có thể đi qua nhiều nút ẩn, nó luôn đi theo một hướng và không bao giờ lùi	Mặc dù dữ liệu có thể đi qua nhiều nút ẩn, nó luôn đi theo một hướng và không bao giờ lùi
21	149	Số đặc trưng của tập dữ liệu sẽ tương ứng với số nơron trong lớp đầu vào	Số đặc trưng của tập dữ liệu sẽ tương ứng với số nơron trong lớp đầu vào
21	150	Tất cả các nơron này được kết nối với mỗi nơron trong lớp ẩn thông qua các đường liên kết gọi là “khớp thần kinh”	Tất cả các nơron này được kết nối với mỗi nơron trong lớp ẩn thông qua các đường liên kết gọi là “khớp thần kinh”
21	151	Mỗi “khớp thần kinh” sẽ được gán một trọng số (weight)	Mỗi “khớp thần kinh” sẽ được gán một trọng số (weight)
21	152	Các trọng số này sẽ được điều chỉnh trong quá trình học của mạng nơron nhân tạo để mô hình hóa mối liên hệ giữa lớp đầu vào và đầu ra	Các trọng số này sẽ được điều chỉnh trong quá trình học của mạng nơron nhân tạo để mô hình hóa mối liên hệ giữa lớp đầu vào và đầu ra
x	153	22 Hình 2-4: Mạng nơron truyền thẳng Mạng nơron hồi quy Như đã đề cập ở trên, mạng nơron gồm 3 tầng chính là tầng vào, tầng ra và tầng ẩn	22 Hình 2-4: Mạng nơron truyền thẳng Mạng nơron hồi quy Như đã đề cập ở trên, mạng nơron gồm 3 tầng chính là tầng vào, tầng ra và tầng ẩn
22	154	Có thể thấy đầu vào và đầu ra của mạng neuron này là độc lập với nhau	Có thể thấy đầu vào và đầu ra của mạng neuron này là độc lập với nhau
22	155	Như vậy mô hình này không phù hợp với những bài toán dạng chuỗi như mô tả, hoàn thành câu, ..	Như vậy mô hình này không phù hợp với những bài toán dạng chuỗi như mô tả, hoàn thành câu, ..
22	156	vì những dự đoán tiếp theo như từ tiếp theo phụ thuộc vào vị trí của nó trong câu và những từ đằng trước nó	vì những dự đoán tiếp theo như từ tiếp theo phụ thuộc vào vị trí của nó trong câu và những từ đằng trước nó
22	157	Và như vậy, RNN ra đời với ý tưởng chính là sử dụng một bộ nhớ để lưu lại thông tin từ từ những bước tính toán xử lý trước để dựa vào nó có thể đưa ra dự đoán chính xác nhất cho bước dự đoán hiện tại	Và như vậy, RNN ra đời với ý tưởng chính là sử dụng một bộ nhớ để lưu lại thông tin từ từ những bước tính toán xử lý trước để dựa vào nó có thể đưa ra dự đoán chính xác nhất cho bước dự đoán hiện tại
22	158	Mạng hồi quy là mạng có chứa các liên kết ngược	Mạng hồi quy là mạng có chứa các liên kết ngược
22	159	Khác với mạng truyền thẳng, các thuộc tính động của mạng mới quan trọng	Khác với mạng truyền thẳng, các thuộc tính động của mạng mới quan trọng
22	160	Trong một số trường hợp, các giá trị kích hoạt của các đơn vị trải qua quá trình nới lỏng (tăng giảm số đơn vị và thay đổi các liên kết) cho đến khi mạng đạt đến một trạng thái ổn định và các giá trị kích hoạt không thay đổi nữa	Trong một số trường hợp, các giá trị kích hoạt của các đơn vị trải qua quá trình nới lỏng (tăng giảm số đơn vị và thay đổi các liên kết) cho đến khi mạng đạt đến một trạng thái ổn định và các giá trị kích hoạt không thay đổi nữa
22	161	Trong các ứng dụng khác mà cách chạy động tạo thành đầu ra của mạng thì những sự thay đổi các giá trị kích hoạt là đáng quan tâm	Trong các ứng dụng khác mà cách chạy động tạo thành đầu ra của mạng thì những sự thay đổi các giá trị kích hoạt là đáng quan tâm

22	162	Hình 2-5: Mạng nơron hồi quy Một điểm nổi bật của RNN chính là ý tưởng kết nối các thông tin phía trước để dự đoán cho hiện tại	Hình 2-5: Mạng nơron hồi quy Một điểm nổi bật của RNN chính là ý tưởng kết nối các thông tin phía trước để dự đoán cho hiện tại
22	163	Đôi lúc ta chỉ cần xem lại thông tin vừa có thôi là đủ để biết được tình huống hiện tại	Đôi lúc ta chỉ cần xem lại thông tin vừa có thôi là đủ để biết được tình huống hiện tại
22	164	Ví dụ, ta có câu: “các đám mây trên bầu trời” thì ta chỉ cần đọc tới “các đám mây trên bầu” là đủ biết được chữ tiếp theo là “trời” rồi	Ví dụ, ta có câu: “các đám mây trên bầu trời” thì ta chỉ cần đọc tới “các đám mây trên bầu” là đủ biết được chữ tiếp theo là “trời” rồi
22	165	Trong tình huống này, khoảng cách tới thông tin có được cần để dự đoán là nhỏ,	Trong tình huống này, khoảng cách tới thông tin có được cần để dự đoán là nhỏ,
x	166	23 nên RNN hoàn toàn có thể học được	23 nên RNN hoàn toàn có thể học được
23	167	Nhưng trong nhiều tình huống ta buộc phải sử dụng nhiều ngữ cảnh hơn để suy luận	Nhưng trong nhiều tình huống ta buộc phải sử dụng nhiều ngữ cảnh hơn để suy luận
23	168	Ví dụ, dự đoán chữ cuối cùng trong đoạn: “I grew up in France… I speak fluent French.”	Ví dụ, dự đoán chữ cuối cùng trong đoạn: “I grew up in France… I speak fluent French.”
x	169	Rõ ràng là các thông tin gần (“I speak fluent”) chỉ có phép ta biết được đằng sau nó sẽ là tên của một ngôn ngữ nào đó, còn không thể nào biết được đó là tiếng gì	Rõ ràng là các thông tin gần (“I speak fluent”) chỉ có phép ta biết được đằng sau nó sẽ là tên của một ngôn ngữ nào đó, còn không thể nào biết được đó là tiếng gì
23	170	Muốn biết là tiếng gì, thì ta cần phải có thêm ngữ cảnh “I grew up in France” nữa mới có thể suy luận được	Muốn biết là tiếng gì, thì ta cần phải có thêm ngữ cảnh “I grew up in France” nữa mới có thể suy luận được
23	171	Rõ ràng là khoảng cách thông tin lúc này có thể đã khá xa rồi	Rõ ràng là khoảng cách thông tin lúc này có thể đã khá xa rồi
23	172	Theo Hochreiter (1991) và Bengio (1994), với khoảng cách càng lớn dần thì RNN bắt đầu không thể nhớ và học được nữa	Theo Hochreiter (1991) và Bengio (1994), với khoảng cách càng lớn dần thì RNN bắt đầu không thể nhớ và học được nữa
23	173	Và LSTM ra đời dựa trên RNN, có khả năng giải quyết vấn đề này	Và LSTM ra đời dựa trên RNN, có khả năng giải quyết vấn đề này
23	174	Mạng Long Short-Term Memory (LSTM) Long Short Term Memory networks – thường được gọi là “LSTM”, là trường hợp đặc biệt của RNN, có khả năng học với sự phụ thuộc lâu dài của các nơ-ron	Mạng Long Short-Term Memory (LSTM) Long Short Term Memory networks – thường được gọi là “LSTM”, là trường hợp đặc biệt của RNN, có khả năng học với sự phụ thuộc lâu dài của các nơ-ron
23	175	Mô hình này được giới thiệu bởi Hochreiter & Schmidhuber (1997) , và được cải tiến lại bởi Ayako Mikami (2016)	Mô hình này được giới thiệu bởi Hochreiter & Schmidhuber (1997) , và được cải tiến lại bởi Ayako Mikami (2016)
23	176	Mục tiêu chính của LSTM là quyết định thông tin nào được lưu lại và loại bỏ tại mỗi nơron của RNN	Mục tiêu chính của LSTM là quyết định thông tin nào được lưu lại và loại bỏ tại mỗi nơron của RNN
23	177	Hình 2-6: Mô hình mạng LSTM Chìa khóa của LSTM là trạng thái tế bào (cell state)	Hình 2-6: Mô hình mạng LSTM Chìa khóa của LSTM là trạng thái tế bào (cell state)
23	177	Hình 2-6: Mô hình mạng LSTM Chìa khóa của LSTM là trạng thái tế bào (cell state)	Chìa khóa của LSTM là trạng thái tế bào (cell state)
24	178	LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate) như hình 3.6	LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate) như hình 3.6

24	178	LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate) như hình 3.6	LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate) như hình 3.6: Cổng quên: Cổng này quyết định xem thông tin nào trong bộ nhớ hiện tại được giữ và thông tin nào bị bỏ lại
x	179	24 Cổng quên: Cổng này quyết định xem thông tin nào trong bộ nhớ hiện tại được giữ và thông tin nào bị bỏ lại	24 Cổng quên: Cổng này quyết định xem thông tin nào trong bộ nhớ hiện tại được giữ và thông tin nào bị bỏ lại
24	180	Thông tin đầu vào được cho vào hàm sigmoid	Thông tin đầu vào được cho vào hàm sigmoid
24	180	Thông tin đầu vào được cho vào hàm sigmoid	Thông tin đầu vào được cho vào hàm sigmoid
24	181	Đầu ra của hàm này đóng vai trò là mask để lọc thông tin từ trạng thái cell	Đầu ra của hàm này đóng vai trò là mask để lọc thông tin từ trạng thái cell
24	181	Đầu ra của hàm này đóng vai trò là mask để lọc thông tin từ trạng thái cell	Đầu ra của hàm này đóng vai trò là mask để lọc thông tin từ trạng thái cell
24	182	Chìa khóa của LSTM là trạng thái tế bào (cell state)	Hình 2-6: Mô hình mạng LSTM Chìa khóa của LSTM là trạng thái tế bào (cell state)
24	182	Chìa khóa của LSTM là trạng thái tế bào (cell state)	Chìa khóa của LSTM là trạng thái tế bào (cell state)
24	183	LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate) như hình 3.6: Cổng quên: Cổng này quyết định xem thông tin nào trong bộ nhớ hiện tại được giữ và thông tin nào bị bỏ lại	LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate) như hình 3.6
24	183	LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate) như hình 3.6: Cổng quên: Cổng này quyết định xem thông tin nào trong bộ nhớ hiện tại được giữ và thông tin nào bị bỏ lại	LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate) như hình 3.6: Cổng quên: Cổng này quyết định xem thông tin nào trong bộ nhớ hiện tại được giữ và thông tin nào bị bỏ lại
24	184	Thông tin đầu vào được cho vào hàm sigmoid	Thông tin đầu vào được cho vào hàm sigmoid
24	184	Thông tin đầu vào được cho vào hàm sigmoid	Thông tin đầu vào được cho vào hàm sigmoid
24	185	Đầu ra của hàm này đóng vai trò là mask để lọc thông tin từ trạng thái cell	Đầu ra của hàm này đóng vai trò là mask để lọc thông tin từ trạng thái cell
24	185	Đầu ra của hàm này đóng vai trò là mask để lọc thông tin từ trạng thái cell	Đầu ra của hàm này đóng vai trò là mask để lọc thông tin từ trạng thái cell
24	186	Hình 2-8: Cổng vào Cổng ra: cổng này quyết định output của từ hiện tại là gì	Hình 2-8: Cổng vào Cổng ra: cổng này quyết định output của từ hiện tại là gì
24	187	Nó được lấy thông tin từ 2 nguồn: trạng thái cell và input hiện tại	Nó được lấy thông tin từ 2 nguồn: trạng thái cell và input hiện tại
24	188	Trạng thái cell sau khi chỉnh sửa sẽ đi qua hàm tanh và input hiện tại thì được đi qua hàm sigmoid	Trạng thái cell sau khi chỉnh sửa sẽ đi qua hàm tanh và input hiện tại thì được đi qua hàm sigmoid
24	189	Kết hợp 2 kết quả trên để có được kết quả đầu ra	Kết hợp 2 kết quả trên để có được kết quả đầu ra
24	190	Kết quả đầu ra và cả trạng thái cell đều được đưa vào bước tiếp theo	Kết quả đầu ra và cả trạng thái cell đều được đưa vào bước tiếp theo

24	191	Hình 2-9: Cổng ra 2.3 Cơ chế Word Embedding Đối với xử lý ngôn ngữ tự nhiên nói chung và dịch máy nói riêng, dữ liệu thường là dạng chuỗi ký tự	Hình 2-9: Cổng ra 2.3 Cơ chế Word Embedding Đối với xử lý ngôn ngữ tự nhiên nói chung và dịch máy nói riêng, dữ liệu thường là dạng chuỗi ký tự
24	192	Con người nhìn chuỗi ký tự này và xử lý nội dung ở dạng các từ được ghép nối với nhau	Con người nhìn chuỗi ký tự này và xử lý nội dung ở dạng các từ được ghép nối với nhau
24	193	Câu hỏi được đặt ra tương tự đối với máy tính	Câu hỏi được đặt ra tương tự đối với máy tính
24	194	Làm thế nào để biểu diễn một chuỗi ký tự thành các con số để máy tính xử lý, đặc biệt trong các mô hình học máy khi mà dữ liệu đầu vào đóng vai trò cực kỳ quan trọng để xây dựng được mô hình hiệu quả	Làm thế nào để biểu diễn một chuỗi ký tự thành các con số để máy tính xử lý, đặc biệt trong các mô hình học máy khi mà dữ liệu đầu vào đóng vai trò cực kỳ quan trọng để xây dựng được mô hình hiệu quả
x	195	25 tốt và được sử dụng phổ biến hiện nay đó là Word Embedding	25 tốt và được sử dụng phổ biến hiện nay đó là Word Embedding
25	196	Kỹ thuật trên cho phép biểu diễn mỗi token bằng một vector với số chiều thấp và có sự liên hệ ngữ nghĩa giữa các vector	Kỹ thuật trên cho phép biểu diễn mỗi token bằng một vector với số chiều thấp và có sự liên hệ ngữ nghĩa giữa các vector
25	197	Word embedding là quá trình chuyển đổi văn bản thành các con số và có thể có nhiều đại diện dạng số khác nhau thể hiện cùng một văn bản	Word embedding là quá trình chuyển đổi văn bản thành các con số và có thể có nhiều đại diện dạng số khác nhau thể hiện cùng một văn bản
25	198	Word embedding là kỹ thuật để thể hiện các từ thành các vector có kích thước cố định, sao cho các từ có nghĩa tương tự hoặc gần nghĩa được thể hiện bằng các vector gần nhau (tính theo khoảng cách euclid)	Word embedding là kỹ thuật để thể hiện các từ thành các vector có kích thước cố định, sao cho các từ có nghĩa tương tự hoặc gần nghĩa được thể hiện bằng các vector gần nhau (tính theo khoảng cách euclid)
25	199	Nhiều thuật toán học máy và hầu hết tất cả các kiến trúc học sâu (Deep Learning) không thể xử lý trực tiếp các xâu hay các văn bản thông thường	Nhiều thuật toán học máy và hầu hết tất cả các kiến trúc học sâu (Deep Learning) không thể xử lý trực tiếp các xâu hay các văn bản thông thường
25	200	Chúng yêu cầu đầu vào là các con số để thực thi các tác vụ của mình như phân loại văn bản, dịch	Chúng yêu cầu đầu vào là các con số để thực thi các tác vụ của mình như phân loại văn bản, dịch
25	201	Word embedding về cơ bản sẽ thực hiện ánh xạ một từ trong một từ điển thành một vector	Word embedding về cơ bản sẽ thực hiện ánh xạ một từ trong một từ điển thành một vector
25	202	Chính vì vậy có thể hiểu word embedding là quá trình vector hóa một từ, hay tổng quát là vector hóa văn bản	Chính vì vậy có thể hiểu word embedding là quá trình vector hóa một từ, hay tổng quát là vector hóa văn bản
25	203	Cách biểu diễn Word Embedding đơn giản, những từ có nghĩa tương tự nhau, có thể thay thế cho nhau sẽ được đặt gần nhau	Cách biểu diễn Word Embedding đơn giản, những từ có nghĩa tương tự nhau, có thể thay thế cho nhau sẽ được đặt gần nhau
25	204	Khoảng cách càng gần thì càng có khả năng thay thế hơn	Khoảng cách càng gần thì càng có khả năng thay thế hơn
25	205	Như vậy, con người dịch thuật sử dụng ngôn ngữ tự nhiên, còn dịch máy sử dụng Word Embedding	Như vậy, con người dịch thuật sử dụng ngôn ngữ tự nhiên, còn dịch máy sử dụng Word Embedding
25	206	Mã hóa BPE Byte Pair Encoding là một thuật toán nén dữ liệu được giới thiệu lần đầu tiên vào năm 1994, giúp tăng hiệu quả của tất cả các mô hình NLP tiên tiến hiện nay (bao gồm cả BERT)	Mã hóa BPE Byte Pair Encoding là một thuật toán nén dữ liệu được giới thiệu lần đầu tiên vào năm 1994, giúp tăng hiệu quả của tất cả các mô hình NLP tiên tiến hiện nay (bao gồm cả BERT)

25	207	Các không gian véc tơ từ được huấn luyện như Word2vec và GloVe đã đặt nền tảng cho những thành công để máy tính có thể hiểu được ý nghĩa của các từ	Các không gian véc tơ từ được huấn luyện như Word2vec và GloVe đã đặt nền tảng cho những thành công để máy tính có thể hiểu được ý nghĩa của các từ
25	208	Trong nhiều năm, chúng là cách biểu diễn đáng tin cậy trong việc huấn luyện các mô hình học máy trong NLP khi không có nhiều dữ liệu	Trong nhiều năm, chúng là cách biểu diễn đáng tin cậy trong việc huấn luyện các mô hình học máy trong NLP khi không có nhiều dữ liệu
25	209	Mặc dù vậy, chúng không phải là công cụ toàn năng khi đối mặt với những từ hiếm xuất hiện	Mặc dù vậy, chúng không phải là công cụ toàn năng khi đối mặt với những từ hiếm xuất hiện
25	210	Các từ này được thay thế bởi tokens khi cài đặt mô hình	Các từ này được thay thế bởi tokens khi cài đặt mô hình
25	211	Để giải quyết các từ hiếm, chúng ta có giải pháp là biểu diễn văn bản dưới dạng tập hợp các ký tự	Để giải quyết các từ hiếm, chúng ta có giải pháp là biểu diễn văn bản dưới dạng tập hợp các ký tự
25	212	Các từ hiếm xét cho cùng vẫn được tạo nên từ những ký tự “không hiếm”	Các từ hiếm xét cho cùng vẫn được tạo nên từ những ký tự “không hiếm”
25	213	Mặc dù vậy, các ký tự không có được sự mô tả ngữ nghĩa trọn vẹn như các từ	Mặc dù vậy, các ký tự không có được sự mô tả ngữ nghĩa trọn vẹn như các từ
25	214	Đi tìm một cách biểu diễn dữ liệu trung hòa giữa biểu diễn dữ liệu bằng ký tự và từ, bước đột phá thực sự đầu tiên trong việc giải quyết vấn đề từ hiếm được thực hiện bởi các nhà nghiên cứu tại Đại học Edinburgh bằng cách sử dụng thành phần từ với Byte Pair Encoding (BPE)	Đi tìm một cách biểu diễn dữ liệu trung hòa giữa biểu diễn dữ liệu bằng ký tự và từ, bước đột phá thực sự đầu tiên trong việc giải quyết vấn đề từ hiếm được thực hiện bởi các nhà nghiên cứu tại Đại học Edinburgh bằng cách sử dụng thành phần từ với Byte Pair Encoding (BPE)
25	215	Ngày nay, phương pháp mã hóa này và các biến thể của nó trở thành chuẩn mực trong hầu hết các mô hình tiên tiến bao gồm cả BERT, GPT-2, RoBERTa,..	Ngày nay, phương pháp mã hóa này và các biến thể của nó trở thành chuẩn mực trong hầu hết các mô hình tiên tiến bao gồm cả BERT, GPT-2, RoBERTa,..
25	216	Nguyên lý hoạt động của BPE dựa trên phân tích trực quan rằng hầu hết các từ đều có thể phân tích thành các thành phần con	Nguyên lý hoạt động của BPE dựa trên phân tích trực quan rằng hầu hết các từ đều có thể phân tích thành các thành phần con
x	217	26 lowest đều là hợp thành bởi low và những đuôi phụ er, est	26 lowest đều là hợp thành bởi low và những đuôi phụ er, est
26	218	Những đuôi này rất thường xuyên xuất hiện ở các từ	Những đuôi này rất thường xuyên xuất hiện ở các từ
26	219	Như vậy khi biểu diễn từ lower chúng ta có thể mã hóa chúng thành hai thành phần từ phụ (subwords) tách biệt là low và er	Như vậy khi biểu diễn từ lower chúng ta có thể mã hóa chúng thành hai thành phần từ phụ (subwords) tách biệt là low và er
26	220	Theo cách biểu diễn này sẽ không phát sinh thêm một index mới cho từ lower và đồng thời tìm được mối liên hệ giữa lower, lowest và low nhờ có chung thành phần từ phụ là low	Theo cách biểu diễn này sẽ không phát sinh thêm một index mới cho từ lower và đồng thời tìm được mối liên hệ giữa lower, lowest và low nhờ có chung thành phần từ phụ là low
26	221	Phương pháp BPE sẽ thống kê tần suất xuất hiện của các từ phụ cùng nhau và tìm cách gộp chúng lại nếu tần suất xuất hiện của chúng là lớn nhất	Phương pháp BPE sẽ thống kê tần suất xuất hiện của các từ phụ cùng nhau và tìm cách gộp chúng lại nếu tần suất xuất hiện của chúng là lớn nhất
26	222	Cứ tiếp tục quá trình gộp từ phụ cho tới khi không tồn tại các subword để gộp nữa, ta sẽ thu được tập subwords cho toàn bộ bộ văn bản mà mọi từ đều có thể biểu diễn được thông qua subwords	Cứ tiếp tục quá trình gộp từ phụ cho tới khi không tồn tại các subword để gộp nữa, ta sẽ thu được tập subwords cho toàn bộ bộ văn bản mà mọi từ đều có thể biểu diễn được thông qua subwords

26	223	Quá trình này gồm các bước như sau: • Bước 1: Khởi tạo bộ từ điển	Quá trình này gồm các bước như sau: • Bước 1: Khởi tạo bộ từ điển
26	224	• Bước 2: Biểu diễn mỗi từ trong bộ văn bản bằng cách kết hợp của các ký tự @@ ở cuối cùng để đánh dấu sự kết thúc của một từ	• Bước 2: Biểu diễn mỗi từ trong bộ văn bản bằng cách kết hợp của các ký tự @@ ở cuối cùng để đánh dấu sự kết thúc của một từ
26	225	• Bước 3: Thống kê tần suất xuất hiện theo cặp của toàn bộ token theo trong bộ từ điển	• Bước 3: Thống kê tần suất xuất hiện theo cặp của toàn bộ token theo trong bộ từ điển
26	226	• Bước 4: Gộp các cặp câu có tần suất xuất hiện lớn nhất để tạo thành một n-gram theo level character cho từ điển	• Bước 4: Gộp các cặp câu có tần suất xuất hiện lớn nhất để tạo thành một n-gram theo level character cho từ điển
26	227	• Bước 5: Lặp lại bước 3 và bước 4 cho tới khi số bước triển khai merge đạt đỉnh hoặc kích thước kỳ vọng của từ điển đạt được	• Bước 5: Lặp lại bước 3 và bước 4 cho tới khi số bước triển khai merge đạt đỉnh hoặc kích thước kỳ vọng của từ điển đạt được
26	228	Giả sử từ điển của chúng ta gồm các từ với tần suất như sau: vocab = {'low@@': 5, 'lower@@': 2, 'newest@@': 6, 'widest@@': 3}	Giả sử từ điển của chúng ta gồm các từ với tần suất như sau: vocab = {'low@@': 5, 'lower@@': 2, 'newest@@': 6, 'widest@@': 3}
26	229	Coi mỗi ký tự là một token	Coi mỗi ký tự là một token
26	230	Khi đó thống kê tần suất xuất hiện của các cặp ký tự như sau: ('d', 'e'): 3, ('e', 'r'): 2, ('e', 's'): 9, ('e', 'w'): 6, ('i', 'd'): 3, ('l', 'o'): 7, ('n', 'e'): 6, ('o', 'w'): 7, ('r', '@@'): 2, ('s', 't'): 9, ('t', @@'): 9, ('w', '@@'): 5, ('w', 'e'): 8, ('w', 'i'): 3	Khi đó thống kê tần suất xuất hiện của các cặp ký tự như sau: ('d', 'e'): 3, ('e', 'r'): 2, ('e', 's'): 9, ('e', 'w'): 6, ('i', 'd'): 3, ('l', 'o'): 7, ('n', 'e'): 6, ('o', 'w'): 7, ('r', '@@'): 2, ('s', 't'): 9, ('t', @@'): 9, ('w', '@@'): 5, ('w', 'e'): 8, ('w', 'i'): 3
26	231	Lựa chọn cặp từ phụ có tần suất xuất hiện nhỏ nhất và merge chúng thành một từ phụ mới	Lựa chọn cặp từ phụ có tần suất xuất hiện nhỏ nhất và merge chúng thành một từ phụ mới
26	232	Ta nhận thấy qua các lượt merge từ phụ, độ dài của các từ phụ trong từ điển tăng dần	Ta nhận thấy qua các lượt merge từ phụ, độ dài của các từ phụ trong từ điển tăng dần
26	233	Thuật toán hội tụ trước 1000 vòng lặp vì toàn bộ các từ phụ đã được merge và đạt ngưỡng của từng từ đơn	Thuật toán hội tụ trước 1000 vòng lặp vì toàn bộ các từ phụ đã được merge và đạt ngưỡng của từng từ đơn
26	234	Khi giới hạn kích thước của từ điển hoặc số lượng lượt merge ta sẽ thu được một từ điển từ phụ là thành phần của các từ trong từ điển	Khi giới hạn kích thước của từ điển hoặc số lượng lượt merge ta sẽ thu được một từ điển từ phụ là thành phần của các từ trong từ điển
26	235	Khi đó mọi từ mới dường như sẽ có thể biểu diễn được theo từ phụ	Khi đó mọi từ mới dường như sẽ có thể biểu diễn được theo từ phụ
26	236	Ví dụ: Khi dừng số lượt merge tại bước 10 ta thu được từ điển: {'low@@': 5, 'low': 2, 'e': 2, 'r': 2, '@@': 2, 'newest@@': 6, 'wid': 3, 'est@@': 3}	Ví dụ: Khi dừng số lượt merge tại bước 10 ta thu được từ điển: {'low@@': 5, 'low': 2, 'e': 2, 'r': 2, '@@': 2, 'newest@@': 6, 'wid': 3, 'est@@': 3}
26	237	Khi đó ta có thể biểu diễn một token mới chưa từng xuất hiện trong từ điển là wider thành wider	Khi đó ta có thể biểu diễn một token mới chưa từng xuất hiện trong từ điển là wider thành wider
26	238	2.4 Mô hình Seq2Seq Sequence to Sequence Model (Seq2seq) là một mô hình Deep Learning với mục đích tạo ra một output sequence từ một input sequence mà độ dài của 2 sequences này có thể khác nhau	2.4 Mô hình Seq2Seq Sequence to Sequence Model (Seq2seq) là một mô hình Deep Learning với mục đích tạo ra một output sequence từ một input sequence mà độ dài của 2 sequences này có thể khác nhau
26	239	Seq2seq được giới thiệu bởi nhóm nghiên cứu	Seq2seq được giới thiệu bởi nhóm nghiên cứu
x	240	27 của Google vào năm 2014 trong bài báo Sequence to Sequence with Neural Networks	27 của Google vào năm 2014 trong bài báo Sequence to Sequence with Neural Networks

27	241	Mặc dù mục đích ban đầu của Model này là để áp dụng trong Machine Translation, tuy nhiên hiện nay Seq2seq cũng được áp dụng nhiều trong các hệ thống khác như Speech recognition, Text summarization, Image captioning,… Seq2seq gồm 2 phần chính là Encoder và Decoder	Mặc dù mục đích ban đầu của Model này là để áp dụng trong Machine Translation, tuy nhiên hiện nay Seq2seq cũng được áp dụng nhiều trong các hệ thống khác như Speech recognition, Text summarization, Image captioning,… Seq2seq gồm 2 phần chính là Encoder và Decoder
27	242	Cả hai thành phần này đều được hình thành từ các mạng Neural Networks, trong đó Encoder có nhiệm vụ chuyển đổi dữ liệu đầu vào (input sequence) thành một representation với lower dimension còn Decoder có nhiệm vụ tạo ra output sequence từ representation của input sequence được tạo ra ở phần Encoder	Cả hai thành phần này đều được hình thành từ các mạng Neural Networks, trong đó Encoder có nhiệm vụ chuyển đổi dữ liệu đầu vào (input sequence) thành một representation với lower dimension còn Decoder có nhiệm vụ tạo ra output sequence từ representation của input sequence được tạo ra ở phần Encoder
27	243	Mô hình Seq2Seq cơ bản có nhược điểm là yêu cầu RNN decoder sử dụng toàn bộ thông tin mã hóa từ chuỗi đầu vào cho dù chuỗi đó dài hay ngắn	Mô hình Seq2Seq cơ bản có nhược điểm là yêu cầu RNN decoder sử dụng toàn bộ thông tin mã hóa từ chuỗi đầu vào cho dù chuỗi đó dài hay ngắn
27	244	Thứ hai, RNN encoder cần phải mã hóa chuỗi đầu vào thành một vector duy nhất và có độ dài cố định	Thứ hai, RNN encoder cần phải mã hóa chuỗi đầu vào thành một vector duy nhất và có độ dài cố định
27	245	Ràng buộc này không thực sự hiệu quả vì trong thực tế, việc sinh ra từ tại một bước thời gian trong chuỗi đầu ra có khi phụ thuộc nhiều hơn vào một số những thành phần nhất định trong chuỗi đầu vào	Ràng buộc này không thực sự hiệu quả vì trong thực tế, việc sinh ra từ tại một bước thời gian trong chuỗi đầu ra có khi phụ thuộc nhiều hơn vào một số những thành phần nhất định trong chuỗi đầu vào
27	246	Ví dụ, khi dịch một câu từ tiếng nước này sang tiếng nước khác, chúng ta thường quan tâm nhiều hơn đến ngữ cảnh xung quanh từ hiện tại so với các từ khác trong câu	Ví dụ, khi dịch một câu từ tiếng nước này sang tiếng nước khác, chúng ta thường quan tâm nhiều hơn đến ngữ cảnh xung quanh từ hiện tại so với các từ khác trong câu
27	247	Kỹ thuật attention được đưa ra để giải quyết vấn đề đó	Kỹ thuật attention được đưa ra để giải quyết vấn đề đó
27	248	Kỹ thuật attention được đưa ra lần đầu vào năm 2014 bởi Bahdanau và cộng sự trong công trình nghiên cứu về dịch máy	Kỹ thuật attention được đưa ra lần đầu vào năm 2014 bởi Bahdanau và cộng sự trong công trình nghiên cứu về dịch máy
27	249	Ở mức trừu tượng, kỹ thuật attention nới lỏng điều kiện rằng toàn bộ chuỗi đầu vào được mã hóa bằng một vector duy nhất	Ở mức trừu tượng, kỹ thuật attention nới lỏng điều kiện rằng toàn bộ chuỗi đầu vào được mã hóa bằng một vector duy nhất
27	250	Thay vào đó các từ trong chuỗi đầu vào sẽ được RNN encoder mã hóa thành một dãy các vector	Thay vào đó các từ trong chuỗi đầu vào sẽ được RNN encoder mã hóa thành một dãy các vector
27	251	Sau đó RNN decoder áp dụng kỹ thuật attention mềm dẻo (soft attention) bằng cách lấy tổng có trọng số của dãy các vector mã hóa	Sau đó RNN decoder áp dụng kỹ thuật attention mềm dẻo (soft attention) bằng cách lấy tổng có trọng số của dãy các vector mã hóa
27	252	Các trọng số trong mô hình này được tính bằng một mạng neural truyền thẳng	Các trọng số trong mô hình này được tính bằng một mạng neural truyền thẳng
27	253	RNN encoder, RNN decoder và các tham số trong kỹ thuật attention được huấn luyện đồng thời từ dữ liệu	RNN encoder, RNN decoder và các tham số trong kỹ thuật attention được huấn luyện đồng thời từ dữ liệu
27	254	Hình dưới đây minh họa mô hình Seq2Seq sử dụng attention trong bài toán dịch máy	Hình dưới đây minh họa mô hình Seq2Seq sử dụng attention trong bài toán dịch máy

x	255	28 Hình 2-10: Attention với mô hình Seq2Seq trong dịch máy 2.5 Mô hình Transformer Giới thiệu mô hình Transformer RNN và LSTM là các phương pháp tiếp cận hiện đại thường được sử dụng trong mô hình về xử lý ngôn ngữ và từ đó, đã có nhiều nỗ lực cải tiến mô hình ngôn ngữ và kiến trúc mã hóa - giải mã	28 Hình 2-10: Attention với mô hình Seq2Seq trong dịch máy 2.5 Mô hình Transformer Giới thiệu mô hình Transformer RNN và LSTM là các phương pháp tiếp cận hiện đại thường được sử dụng trong mô hình về xử lý ngôn ngữ và từ đó, đã có nhiều nỗ lực cải tiến mô hình ngôn ngữ và kiến trúc mã hóa - giải mã
28	256	Các mô hình hồi quy thường tính toán theo vị trị các ký tự các chuỗi đầu vào và đầu ra	Các mô hình hồi quy thường tính toán theo vị trị các ký tự các chuỗi đầu vào và đầu ra
28	257	Việc căn chỉnh vị trí này trong các bước tính toán, sẽ tạo ra các trạng thái ẩn	Việc căn chỉnh vị trí này trong các bước tính toán, sẽ tạo ra các trạng thái ẩn
28	258	Bản chất tuần tự vốn có này loại trừ tính đồng thời trong các mẫu huấn luyện, điều này trở nên quan trọng khi mà chuỗi dài hơn do các ràng buộc bộ nhớ bị giới hạn theo các mẫu ví dụ	Bản chất tuần tự vốn có này loại trừ tính đồng thời trong các mẫu huấn luyện, điều này trở nên quan trọng khi mà chuỗi dài hơn do các ràng buộc bộ nhớ bị giới hạn theo các mẫu ví dụ
28	259	Các nghiên cứu gần đây đã đạt được những cải tiến đáng kể về hiệu quả tính toán thông qua các thủ thuật và tính toán có điều kiện, đồng thời cải thiện hiệu suất của mô hình	Các nghiên cứu gần đây đã đạt được những cải tiến đáng kể về hiệu quả tính toán thông qua các thủ thuật và tính toán có điều kiện, đồng thời cải thiện hiệu suất của mô hình
28	260	Các cơ chế attention đã trở thành một phần không thể thiếu trong mô hình tuần tự, cho phép mô hình hóa các phụ thuộc mà không quan tâm đến khoảng cách của chuỗi đầu vào và đầu ra	Các cơ chế attention đã trở thành một phần không thể thiếu trong mô hình tuần tự, cho phép mô hình hóa các phụ thuộc mà không quan tâm đến khoảng cách của chuỗi đầu vào và đầu ra
28	261	Gần như trong tất cả các trường hợp, các cơ chế attention được sử dụng cùng với mạng hồi quy	Gần như trong tất cả các trường hợp, các cơ chế attention được sử dụng cùng với mạng hồi quy
28	262	Dưới đây, mô hình Transformer được đề xuất, một kiến trúc mô hình tránh việc hồi quy mà thay vào đó là hoàn toàn dựa vào cơ chế attention để đưa ra sự phụ thuộc giữa đầu vào và đầu ra	Dưới đây, mô hình Transformer được đề xuất, một kiến trúc mô hình tránh việc hồi quy mà thay vào đó là hoàn toàn dựa vào cơ chế attention để đưa ra sự phụ thuộc giữa đầu vào và đầu ra
28	263	Tổng quan về mô hình Đây là mô hình được các kỹ sư của Google giới thiệu vào năm 2017 trong bài báo Attention Is All You Need [5]	Tổng quan về mô hình Đây là mô hình được các kỹ sư của Google giới thiệu vào năm 2017 trong bài báo Attention Is All You Need [5]
28	264	Giống như những mô hình dịch máy khác, kiến trúc tổng quan của mô hình transformer bao gồm 2 phần lớn là encoder và decoder	Giống như những mô hình dịch máy khác, kiến trúc tổng quan của mô hình transformer bao gồm 2 phần lớn là encoder và decoder
28	265	Encoder dùng để học vector biểu của câu với mong muốn	Encoder dùng để học vector biểu của câu với mong muốn
x	266	29 rằng vector này mang thông tin hoàn hảo của câu đó	29 rằng vector này mang thông tin hoàn hảo của câu đó
29	267	Decoder thực hiện chức năng chuyển vector biểu diễn kia thành ngôn ngữ đích	Decoder thực hiện chức năng chuyển vector biểu diễn kia thành ngôn ngữ đích
29	268	Chi tiết các thành phần của bộ mã hóa và giải mã được thể hiện như hình 3.11, bộ mã hóa và giải mã lần lượt nằm ở cột bên trái và bên phải của hình vẽ	Chi tiết các thành phần của bộ mã hóa và giải mã được thể hiện như hình 3.11, bộ mã hóa và giải mã lần lượt nằm ở cột bên trái và bên phải của hình vẽ
29	269	Hình 2-11: Kiến trúc mô hình Transformer	28 Hình 2-11: Kiến trúc mô hình Transformer .....
29	269	Hình 2-11: Kiến trúc mô hình Transformer	Hình 2-11: Kiến trúc mô hình Transformer

x	270	30 Hình 2-12: Bộ mã hóa và giải mã trong mô hình Transformer Một trong những ưu điểm của Transformer là mô hình có khả năng xử lý song song cho các từ	30 Hình 2-12: Bộ mã hóa và giải mã trong mô hình Transformer Một trong những ưu điểm của Transformer là mô hình có khả năng xử lý song song cho các từ
30	271	Đầu vào sẽ được đẩy vào cùng một lúc	Đầu vào sẽ được đẩy vào cùng một lúc
30	272	Bộ mã hóa của mô hình transformer bao gồm một tập gồm N = 6 lớp giống nhau, mỗi lớp bao gồm 2 lớp con	Bộ mã hóa của mô hình transformer bao gồm một tập gồm N = 6 lớp giống nhau, mỗi lớp bao gồm 2 lớp con
30	273	Lớp đầu tiên là cơ chế multi-head self-attention, và lớp thứ 2 là mạng feed-forward kết nối đầy đủ	Lớp đầu tiên là cơ chế multi-head self-attention, và lớp thứ 2 là mạng feed-forward kết nối đầy đủ
30	274	Đầu ra của mỗi lớp con là LayerNorm(x + Sublayer(x)), trong đó Sublayer(x) là một hàm được thực hiện bởi chính lớp con đó	Đầu ra của mỗi lớp con là LayerNorm(x + Sublayer(x)), trong đó Sublayer(x) là một hàm được thực hiện bởi chính lớp con đó
30	275	Bộ giải mã: cũng bao gồm tập gồm N = 6 lớp giống nhau	Bộ giải mã: cũng bao gồm tập gồm N = 6 lớp giống nhau
30	276	Ngoài hai lớp con giống như bộ mã hóa, bộ giải mã còn có một lớp để thực hiện multi-head attention	Ngoài hai lớp con giống như bộ mã hóa, bộ giải mã còn có một lớp để thực hiện multi-head attention
x	277	31 trên đầu ra của lớp giải mã	31 trên đầu ra của lớp giải mã
31	278	Ở đây sẽ có thay đổi cơ chế self-attention trong bộ mã hóa	Ở đây sẽ có thay đổi cơ chế self-attention trong bộ mã hóa
31	279	Dưới đây sẽ trình bày chi tiết về bộ mã hóa và giải mã của mô hình transformer	Dưới đây sẽ trình bày chi tiết về bộ mã hóa và giải mã của mô hình transformer
31	280	Cơ chế Self Attention Trước khi đi chi tiết vào mô hình, em sẽ trình bày về self-attention – “trái tim” của mô hình transformer	Cơ chế Self Attention Trước khi đi chi tiết vào mô hình, em sẽ trình bày về self-attention – “trái tim” của mô hình transformer
31	281	Self Attention cho phép mô hình khi mã hóa một từ có thể sử dụng thông tin của những từ liên quan tới nó	Self Attention cho phép mô hình khi mã hóa một từ có thể sử dụng thông tin của những từ liên quan tới nó
x	282	Có thể tưởng tượng self- attention giống như cơ chế tìm kiếm	Có thể tưởng tượng self- attention giống như cơ chế tìm kiếm
31	283	Với một từ cho trước, cơ chế này sẽ cho phép mô hình tìm kiếm trong các từ còn lại để xác định từ nào liên quan để sau đó thông tin sẽ được mã hóa dựa trên tất cả các từ trên	Với một từ cho trước, cơ chế này sẽ cho phép mô hình tìm kiếm trong các từ còn lại để xác định từ nào liên quan để sau đó thông tin sẽ được mã hóa dựa trên tất cả các từ trên
31	284	Đầu vào của self-attention là 3 vector query, key, value	Đầu vào của self-attention là 3 vector query, key, value
31	285	Các vector này được tạo ra bằng cách nhân ma trận biểu diễn các từ đầu vào với ma trận học tương ứng	Các vector này được tạo ra bằng cách nhân ma trận biểu diễn các từ đầu vào với ma trận học tương ứng
31	286	Query vector: vector dùng để chứa thông tin của từ được tìm kiếm, so sánh	Query vector: vector dùng để chứa thông tin của từ được tìm kiếm, so sánh
31	287	Key vector: vector dùng để biểu diễn thông tin các từ được so sánh với từ cần tìm kiếm ở trên	Key vector: vector dùng để biểu diễn thông tin các từ được so sánh với từ cần tìm kiếm ở trên
31	288	Value vector: vector biểu diễn nội dung, ý nghĩa của các từ	Value vector: vector biểu diễn nội dung, ý nghĩa của các từ

31	289	Vector attention cho một từ thể hiện tính tương quan giữa 3 vector này được tạo ra bằng cách nhân tích vô hướng giữa chúng và sau đó được chuẩn hóa bằng hàm softmax	Vector attention cho một từ thể hiện tính tương quan giữa 3 vector này được tạo ra bằng cách nhân tích vô hướng giữa chúng và sau đó được chuẩn hóa bằng hàm softmax
31	290	Cụ thể quá trình tính toán như sau: Bước 1: Tính ma trận query, key, value bằng cách khởi tạo 3 ma trận trọng số query, key, vector	Cụ thể quá trình tính toán như sau: Bước 1: Tính ma trận query, key, value bằng cách khởi tạo 3 ma trận trọng số query, key, vector
31	291	Sau đó nhân input với các ma trận trọng số này để tạo thành 3 ma trận tương ứng	Sau đó nhân input với các ma trận trọng số này để tạo thành 3 ma trận tương ứng
31	292	Nhân 2 ma trận key, query vừa được tính ở trên với nhau để với ý nghĩa là so sánh giữa câu query và key để học mối tương quan	Nhân 2 ma trận key, query vừa được tính ở trên với nhau để với ý nghĩa là so sánh giữa câu query và key để học mối tương quan
31	293	Sau đó thì chuẩn hóa về đoạn [0-1] bằng hàm softmax	Sau đó thì chuẩn hóa về đoạn [0-1] bằng hàm softmax
31	294	1 có nghĩa là câu query giống với key, 0 có nghĩa là không giống	1 có nghĩa là câu query giống với key, 0 có nghĩa là không giống
31	295	Nhân attention weights với ma trận value	Nhân attention weights với ma trận value
31	296	Điều này có nghĩa là chúng ta biểu diễn một từ bằng trung bình có trọng số (attention weights) của ma trận value	Điều này có nghĩa là chúng ta biểu diễn một từ bằng trung bình có trọng số (attention weights) của ma trận value
x	297	32 Hình 2-13: Quá trình tính toán vector attention Hai hàm attention phổ biến được sử dụng là additive attention và dot-product attention	32 Hình 2-13: Quá trình tính toán vector attention Hai hàm attention phổ biến được sử dụng là additive attention và dot-product attention
x	298	Một điểm đặc biệt dot-product attention là chia cho √□ố □□□ề□ □ủ□ □□□□□□ □□□ , additive attention tính toán sử dụng một mạng feed- forward với một tầng ẩn	Một điểm đặc biệt dot-product attention là chia cho √□ố □□□ề□ □ủ□ □□□□□□ □□□ , additive attention tính toán sử dụng một mạng feed- forward với một tầng ẩn
x	299	Về mặt lý thuyết, cả hai giống nhau về độ phức tạp nhưng dot-product attention trong thực tế nhanh hơn và hiệu quả hơn vì có thể sử dụng thuật toán nhân ma trận tối ưu	Về mặt lý thuyết, cả hai giống nhau về độ phức tạp nhưng dot-product attention trong thực tế nhanh hơn và hiệu quả hơn vì có thể sử dụng thuật toán nhân ma trận tối ưu
x	300	Trong đồ án sẽ sử dụng dot-product attention	Trong đồ án sẽ sử dụng dot-product attention
x	301	Để hiểu rõ hơn về cách tính toán chúng ta sẽ thực hiện tính toán vector attention qua các bước như sau: (1) Chuẩn bị đầu vào, (2) Khởi tạo trọng số, (3) Xác định key, query và value, (4) Tính điểm số attention với đầu vào 1, (5) Tính giá trị hàm softmax, (6) Nhân điểm số trên với value, (7) Tính đầu ra cho đầu vào 1, (8) Lặp lại các bước từ 4 đến 7 với các đầu vào khác	Để hiểu rõ hơn về cách tính toán chúng ta sẽ thực hiện tính toán vector attention qua các bước như sau: (1) Chuẩn bị đầu vào, (2) Khởi tạo trọng số, (3) Xác định key, query và value, (4) Tính điểm số attention với đầu vào 1, (5) Tính giá trị hàm softmax, (6) Nhân điểm số trên với value, (7) Tính đầu ra cho đầu vào 1, (8) Lặp lại các bước từ 4 đến 7 với các đầu vào khác
x	302	Cụ thể như sau: Trong ví dụ này, chúng ta sẽ thực hiện tính toán với 3 đầu vào với số chiều là 4	Cụ thể như sau: Trong ví dụ này, chúng ta sẽ thực hiện tính toán với 3 đầu vào với số chiều là 4
x	303	Input 1: [1, 0, 1, 0] Input 2: [0, 2, 0, 2] Input 3: [1, 1, 1, 1] Mỗi một đầu vào phải có 3 biểu diễn đó là key, query và value	Input 1: [1, 0, 1, 0] Input 2: [0, 2, 0, 2] Input 3: [1, 1, 1, 1] Mỗi một đầu vào phải có 3 biểu diễn đó là key, query và value
x	304	Giả sử chúng ta muốn các vector này có số chiều là 3, trong khi đầu vào có kích thước là 4	Giả sử chúng ta muốn các vector này có số chiều là 3, trong khi đầu vào có kích thước là 4
x	305	Do đó, một tập hợp trọng số phải là ma trận có kích thước là 4×3	Do đó, một tập hợp trọng số phải là ma trận có kích thước là 4×3

x	306	Chúng ta sẽ khởi tạo bộ trọng số như sau: Trọng số key	Chúng ta sẽ khởi tạo bộ trọng số như sau: Trọng số key
x	307	33 [[0, 0, 1], [1, 1, 0], [0, 1, 0], [1, 1, 0]] Trọng số query [[1, 0, 1], [1, 0, 0], [0, 0, 1], [0, 1, 1]] Trọng số value [[0, 2, 0], [0, 3, 0], [1, 0, 3], [1, 1, 0]] Lưu ý rằng trong cài đặt mạng nơ ron, các trọng số này thường là các số rất nhỏ, được khởi tạo ngẫu nhiên bằng cách sử dụng các phân phối ngẫu nhiên như Gaussian, Xavier và Kaimin	33 [[0, 0, 1], [1, 1, 0], [0, 1, 0], [1, 1, 0]] Trọng số query [[1, 0, 1], [1, 0, 0], [0, 0, 1], [0, 1, 1]] Trọng số value [[0, 2, 0], [0, 3, 0], [1, 0, 3], [1, 1, 0]] Lưu ý rằng trong cài đặt mạng nơ ron, các trọng số này thường là các số rất nhỏ, được khởi tạo ngẫu nhiên bằng cách sử dụng các phân phối ngẫu nhiên như Gaussian, Xavier và Kaimin
33	308	Khởi tạo này sẽ được thực hiện một lần trước khi huấn luyện	Khởi tạo này sẽ được thực hiện một lần trước khi huấn luyện
33	309	Bây giờ chúng ta đã có 3 bộ trọng số, chúng ta sẽ tính các ma trận biểu diễn key, query và value cho các đầu vào	Bây giờ chúng ta đã có 3 bộ trọng số, chúng ta sẽ tính các ma trận biểu diễn key, query và value cho các đầu vào
33	310	Ma trận key cho đầu vào 1: [0, 0, 1] [1, 0, 1, 0] x [1, 1, 0] = [0, 1, 1] [0, 1, 0] [1, 1, 0] Tính toán tương tự cho dầu vào 2 và đầu vào 3 ta có: [0, 0, 1] [0, 2, 0, 2] x [1, 1, 0] = [4, 4, 0] [0, 1, 0] [1, 1, 0] [0, 0, 1] [1, 1, 1, 1] x [1, 1, 0] = [2, 3, 1] [0, 1, 0] [1, 1, 0] Để tính toán nhanh hơn, chúng ta có thể tính toán ma trận key như sau:	Ma trận key cho đầu vào 1: [0, 0, 1] [1, 0, 1, 0] x [1, 1, 0] = [0, 1, 1] [0, 1, 0] [1, 1, 0] Tính toán tương tự cho dầu vào 2 và đầu vào 3 ta có: [0, 0, 1] [0, 2, 0, 2] x [1, 1, 0] = [4, 4, 0] [0, 1, 0] [1, 1, 0] [0, 0, 1] [1, 1, 1, 1] x [1, 1, 0] = [2, 3, 1] [0, 1, 0] [1, 1, 0] Để tính toán nhanh hơn, chúng ta có thể tính toán ma trận key như sau:
33	310	Ma trận key cho đầu vào 1: [0, 0, 1] [1, 0, 1, 0] x [1, 1, 0] = [0, 1, 1] [0, 1, 0] [1, 1, 0] Tính toán tương tự cho dầu vào 2 và đầu vào 3 ta có: [0, 0, 1] [0, 2, 0, 2] x [1, 1, 0] = [4, 4, 0] [0, 1, 0] [1, 1, 0] [0, 0, 1] [1, 1, 1, 1] x [1, 1, 0] = [2, 3, 1] [0, 1, 0] [1, 1, 0] Để tính toán nhanh hơn, chúng ta có thể tính toán ma trận key như sau:	34 [0, 0, 1] [1, 0, 1, 0] [1, 1, 0] [0, 1, 1] [0, 2, 0, 2] x [0, 1, 0] = [4, 4, 0] [1, 1, 1, 1] [1, 1, 0] [2, 3, 1] Tính toán tương tự với ma trận value: [0, 2, 0] [1, 0, 1, 0] [0, 3, 0] [1, 2, 3] [0, 2, 0, 2] x [1, 0, 3] = [2, 8, 0] [1, 1, 1, 1] [1, 1, 0] [2, 6, 3] Tính toán tương tự với ma trận query: [1, 0, 1] [1, 0, 1, 0] [1, 0, 0] [1, 0, 2] [0, 2, 0, 2] x [0, 0, 1] = [2, 2, 2] [1, 1, 1, 1] [0, 1, 1] [2, 1, 3] Tiếp đến chúng ta sẽ tính toán điểm số attention với từng đầu vào
x	311	34 [0, 0, 1] [1, 0, 1, 0] [1, 1, 0] [0, 1, 1] [0, 2, 0, 2] x [0, 1, 0] = [4, 4, 0] [1, 1, 1, 1] [1, 1, 0] [2, 3, 1] Tính toán tương tự với ma trận value: [0, 2, 0] [1, 0, 1, 0] [0, 3, 0] [1, 2, 3] [0, 2, 0, 2] x [1, 0, 3] = [2, 8, 0] [1, 1, 1, 1] [1, 1, 0] [2, 6, 3] Tính toán tương tự với ma trận query: [1, 0, 1] [1, 0, 1, 0] [1, 0, 0] [1, 0, 2] [0, 2, 0, 2] x [0, 0, 1] = [2, 2, 2] [1, 1, 1, 1] [0, 1, 1] [2, 1, 3] Tiếp đến chúng ta sẽ tính toán điểm số attention với từng đầu vào	Ma trận key cho đầu vào 1: [0, 0, 1] [1, 0, 1, 0] x [1, 1, 0] = [0, 1, 1] [0, 1, 0] [1, 1, 0] Tính toán tương tự cho dầu vào 2 và đầu vào 3 ta có: [0, 0, 1] [0, 2, 0, 2] x [1, 1, 0] = [4, 4, 0] [0, 1, 0] [1, 1, 0] [0, 0, 1] [1, 1, 1, 1] x [1, 1, 0] = [2, 3, 1] [0, 1, 0] [1, 1, 0] Để tính toán nhanh hơn, chúng ta có thể tính toán ma trận key như sau:
x	311	34 [0, 0, 1] [1, 0, 1, 0] [1, 1, 0] [0, 1, 1] [0, 2, 0, 2] x [0, 1, 0] = [4, 4, 0] [1, 1, 1, 1] [1, 1, 0] [2, 3, 1] Tính toán tương tự với ma trận value: [0, 2, 0] [1, 0, 1, 0] [0, 3, 0] [1, 2, 3] [0, 2, 0, 2] x [1, 0, 3] = [2, 8, 0] [1, 1, 1, 1] [1, 1, 0] [2, 6, 3] Tính toán tương tự với ma trận query: [1, 0, 1] [1, 0, 1, 0] [1, 0, 0] [1, 0, 2] [0, 2, 0, 2] x [0, 0, 1] = [2, 2, 2] [1, 1, 1, 1] [0, 1, 1] [2, 1, 3] Tiếp đến chúng ta sẽ tính toán điểm số attention với từng đầu vào	34 [0, 0, 1] [1, 0, 1, 0] [1, 1, 0] [0, 1, 1] [0, 2, 0, 2] x [0, 1, 0] = [4, 4, 0] [1, 1, 1, 1] [1, 1, 0] [2, 3, 1] Tính toán tương tự với ma trận value: [0, 2, 0] [1, 0, 1, 0] [0, 3, 0] [1, 2, 3] [0, 2, 0, 2] x [1, 0, 3] = [2, 8, 0] [1, 1, 1, 1] [1, 1, 0] [2, 6, 3] Tính toán tương tự với ma trận query: [1, 0, 1] [1, 0, 1, 0] [1, 0, 0] [1, 0, 2] [0, 2, 0, 2] x [0, 0, 1] = [2, 2, 2] [1, 1, 1, 1] [0, 1, 1] [2, 1, 3] Tiếp đến chúng ta sẽ tính toán điểm số attention với từng đầu vào
34	312	Giá trị này sẽ được tính bằng cách nhân ma trận query của từng đầu vào với ma trận chuyển vị của ma trận value	Giá trị này sẽ được tính bằng cách nhân ma trận query của từng đầu vào với ma trận chuyển vị của ma trận value

34	313	Cụ thể như sau: [0, 4, 2] [1, 0, 2] x [1, 4, 3] = [2, 4, 4] [1, 0, 1] Tiếp đến chúng ta sẽ chuẩn hóa giá trị trên bằng hàm softmax: Softmax ([2, 4, 4]) = [0.0, 0.5, 0.5] Sau đó, chúng ta sẽ nhân giá trị hàm softmax với ma trận value: 1: 0.0 * [1, 2, 3] = [0.0, 0.0, 0.0] 2: 0.5 * [2, 8, 0] = [1.0, 4.0, 0.0] 3: 0.5 * [2, 6, 3] = [1.0, 3.0, 1.5] Các giá trị trên sẽ được cộng với nhau để thu được ma trận attention [0.0, 0.0, 0.0] + [1.0, 4.0, 0.0] + [1.0, 3.0, 1.5] ---------- = [2.0, 7.0, 1.5] Ma trận [2.0, 7.0, 1.5] chính là biểu diễn attention của đầu vào 1, nó thể hiện mối quan hệ với tất cả các key khác và cả chính nó	Cụ thể như sau: [0, 4, 2] [1, 0, 2] x [1, 4, 3] = [2, 4, 4] [1, 0, 1] Tiếp đến chúng ta sẽ chuẩn hóa giá trị trên bằng hàm softmax: Softmax ([2, 4, 4]) = [0.0, 0.5, 0.5] Sau đó, chúng ta sẽ nhân giá trị hàm softmax với ma trận value: 1: 0.0 * [1, 2, 3] = [0.0, 0.0, 0.0] 2: 0.5 * [2, 8, 0] = [1.0, 4.0, 0.0] 3: 0.5 * [2, 6, 3] = [1.0, 3.0, 1.5] Các giá trị trên sẽ được cộng với nhau để thu được ma trận attention [0.0, 0.0, 0.0] + [1.0, 4.0, 0.0] + [1.0, 3.0, 1.5] ---------- = [2.0, 7.0, 1.5] Ma trận [2.0, 7.0, 1.5] chính là biểu diễn attention của đầu vào 1, nó thể hiện mối quan hệ với tất cả các key khác và cả chính nó
34	314	Thực hiện tương tự với đầu vào 2 và 3, ta có ma trận attention với các đầu vào như sau: [[2.0, 7.0, 1.5], # attention 1 [2.0, 8.0, 0.0], # attention 2 [2.0, 7.8, 0.3]]) # attention 3	Thực hiện tương tự với đầu vào 2 và 3, ta có ma trận attention với các đầu vào như sau: [[2.0, 7.0, 1.5], # attention 1 [2.0, 8.0, 0.0], # attention 2 [2.0, 7.8, 0.3]]) # attention 3
x	315	35 Bộ mã hóa – Encoder Encoder của mô hình transformer có thể bao gồm nhiều encoder layer tượng tự nhau	35 Bộ mã hóa – Encoder Encoder của mô hình transformer có thể bao gồm nhiều encoder layer tượng tự nhau
x	316	Mỗi encoder layer của transformer lại bao gồm 2 thành phần chính là multi head attention và feedforward network, ngoài ra còn có cả skip connection và normalization layer	Mỗi encoder layer của transformer lại bao gồm 2 thành phần chính là multi head attention và feedforward network, ngoài ra còn có cả skip connection và normalization layer
x	317	Các thành phần của 1 lớp được biểu diễn như sau: Hình 2-14: Encoder của mô hình Transformer 2.5.4.1	Các thành phần của 1 lớp được biểu diễn như sau: Hình 2-14: Encoder của mô hình Transformer 2.5.4.1
x	318	Embedding Layer Đầu tiên, các từ được biểu diễn bằng một vector sử dụng một ma trận word embedding có số dòng bằng kích thước của tập từ vựng	Embedding Layer Đầu tiên, các từ được biểu diễn bằng một vector sử dụng một ma trận word embedding có số dòng bằng kích thước của tập từ vựng
x	319	Sau đó các từ trong câu được tìm kiếm trong ma trận này, và được nối nhau thành các dòng của một ma trận 2 chiều chứa ngữ nghĩa của từng từ riêng biệt	Sau đó các từ trong câu được tìm kiếm trong ma trận này, và được nối nhau thành các dòng của một ma trận 2 chiều chứa ngữ nghĩa của từng từ riêng biệt
x	320	Nhưng như các bạn đã thấy, transformer xử lý các từ song song, do đó, với chỉ word embedding mô hình không thể nào biết được vị trí các từ	Nhưng như các bạn đã thấy, transformer xử lý các từ song song, do đó, với chỉ word embedding mô hình không thể nào biết được vị trí các từ
x	321	Như vậy, chúng ta cần một cơ chế nào đó để đưa thông tin vị trí các từ vào trong vector đầu vào	Như vậy, chúng ta cần một cơ chế nào đó để đưa thông tin vị trí các từ vào trong vector đầu vào
x	322	Đó là lúc positional encoding xuất hiện và giải quyết vấn đề của chúng ta	Đó là lúc positional encoding xuất hiện và giải quyết vấn đề của chúng ta
x	323	Position Encoding Phương pháp của tác giả đề xuất không gặp những hạn chế mà chúng ta vừa nêu	Position Encoding Phương pháp của tác giả đề xuất không gặp những hạn chế mà chúng ta vừa nêu
x	324	Vị trí của các từ được mã hóa bằng một vector có kích thước bằng word embedding và được cộng trực tiếp vào word embedding	Vị trí của các từ được mã hóa bằng một vector có kích thước bằng word embedding và được cộng trực tiếp vào word embedding
x	325	Giá trị này được tính như sau: □□(□□□,2□) = □□□ (□□□ 10000 2□ □□□□□□)	Giá trị này được tính như sau: □□(□□□,2□) = □□□ (□□□ 10000 2□ □□□□□□)

x	326	36 □□(□□□,2□+1) = □□□ (□□□ 10000 2□ □□□□□□) Trong đó: • pos là vị trí của từ trong câu • PE là giá trị phần tử thứ i trong embeddings có độ dài □□□□□□ Như vậy bộ mã hóa sẽ nhận ma trận biểu diễn của các từ đã được cộng với thông tin vị trí thông qua positional encoding	36 □□(□□□,2□+1) = □□□ (□□□ 10000 2□ □□□□□□) Trong đó: • pos là vị trí của từ trong câu • PE là giá trị phần tử thứ i trong embeddings có độ dài □□□□□□ Như vậy bộ mã hóa sẽ nhận ma trận biểu diễn của các từ đã được cộng với thông tin vị trí thông qua positional encoding
x	327	Hình 2-15: Ví dụ biểu diễn từ đầu vào Sau đó, ma trận này sẽ được xử lý bởi Multi Head Attention	Hình 2-15: Ví dụ biểu diễn từ đầu vào Sau đó, ma trận này sẽ được xử lý bởi Multi Head Attention
x	328	Multi Head Attention thực chất là là sử dụng nhiều self-attention	Multi Head Attention thực chất là là sử dụng nhiều self-attention
x	329	Multi Head Attention Vấn đề của self-attention là attention của một từ sẽ luôn "chú ý" vào chính nó	Multi Head Attention Vấn đề của self-attention là attention của một từ sẽ luôn "chú ý" vào chính nó
x	330	Chúng ta muốn mô hình có thể học nhiều kiểu mối quan hệ giữ các từ với nhau	Chúng ta muốn mô hình có thể học nhiều kiểu mối quan hệ giữ các từ với nhau
x	331	Ý tưởng là thay vì sử dụng một self-attention thì chúng ta sẽ sử dụng nhiều self- attention	Ý tưởng là thay vì sử dụng một self-attention thì chúng ta sẽ sử dụng nhiều self- attention
8	332	Hình 2-16: Quá trình tính toán vector attention với nhiều "head"	36 Hình 2-16: Quá trình tính toán vector attention với nhiều "head" .....
8	332	Hình 2-16: Quá trình tính toán vector attention với nhiều "head"	Hình 2-16: Quá trình tính toán vector attention với nhiều "head"
x	333	37 Đơn giản là cần nhiều ma trận query, key, value	37 Đơn giản là cần nhiều ma trận query, key, value
x	334	Mỗi "head" sẽ cho ra output riêng, các ma trận này sẽ được kết hợp với nhau và nhân với ma trận trọng số để có được ma trận attention duy nhất	Mỗi "head" sẽ cho ra output riêng, các ma trận này sẽ được kết hợp với nhau và nhân với ma trận trọng số để có được ma trận attention duy nhất
x	335	MultiHead(Q, K, V) = Concat(□□□□1, ...,□□□□□)□ 0 Mỗi encoder và decoder trong Transformer sử dụng N attention	MultiHead(Q, K, V) = Concat(□□□□1, ...,□□□□□)□ 0 Mỗi encoder và decoder trong Transformer sử dụng N attention
x	336	Mỗi attention sẽ biến đổi tuyến tính q, k, k với một ma trận có thể huấn luyện khác nhau tương ứng	Mỗi attention sẽ biến đổi tuyến tính q, k, k với một ma trận có thể huấn luyện khác nhau tương ứng
x	337	Mỗi phép biến đổi cung cấp cho chúng ta một phép chiếu khác nhau cho q, k và v	Mỗi phép biến đổi cung cấp cho chúng ta một phép chiếu khác nhau cho q, k và v
x	338	Vì vậy, N attention cho phép xem mức độ phù hợp từ N quan điểm khác nhau	Vì vậy, N attention cho phép xem mức độ phù hợp từ N quan điểm khác nhau
x	339	Điều này cuối cùng đẩy độ chính xác tổng thể cao hơn, ít nhất là theo kinh nghiệm	Điều này cuối cùng đẩy độ chính xác tổng thể cao hơn, ít nhất là theo kinh nghiệm
x	340	Việc chuyển đổi cũng làm giảm kích thước đầu ra của chúng, do đó, thậm chí N attention được sử dụng, độ phức tạp tính toán vẫn giữ nguyên	Việc chuyển đổi cũng làm giảm kích thước đầu ra của chúng, do đó, thậm chí N attention được sử dụng, độ phức tạp tính toán vẫn giữ nguyên
x	341	Trong multi-head attention, ghép các vectơ đầu ra theo sau là một phép biến đổi tuyến tính	Trong multi-head attention, ghép các vectơ đầu ra theo sau là một phép biến đổi tuyến tính
x	342	Bộ giải mã – Decoder Bộ giải mã thực hiện chức năng giải mã vector của câu nguồn thành câu đích, do đó bộ giải mã sẽ nhận thông tin từ bộ mã hóa là 2 vector key và value	Bộ giải mã – Decoder Bộ giải mã thực hiện chức năng giải mã vector của câu nguồn thành câu đích, do đó bộ giải mã sẽ nhận thông tin từ bộ mã hóa là 2 vector key và value

x	343	Kiến trúc của bộ giải mã rất giống với bộ mã hóa, ngoại trừ có thêm một masked multi- head attention nằm ở giữ dùng để học mối liên quan giữ từ đang được dịch với các từ được ở câu nguồn	Kiến trúc của bộ giải mã rất giống với bộ mã hóa, ngoại trừ có thêm một masked multi- head attention nằm ở giữ dùng để học mối liên quan giữ từ đang được dịch với các từ được ở câu nguồn
x	344	Hình 2-17: Bộ giải mã của mô hình transformer Masked multi-head attention tất nhiên là multi-head attention mà chúng ta đã nói đến ở trên, tuy nhiên đi các từ ở tương lai chưa được mô hình dịch đến được	Hình 2-17: Bộ giải mã của mô hình transformer Masked multi-head attention tất nhiên là multi-head attention mà chúng ta đã nói đến ở trên, tuy nhiên đi các từ ở tương lai chưa được mô hình dịch đến được
38	345	Trong bộ giải mã còn có một multi-head attention khác có chức năng chú ý các từ ở bộ mã hóa, layer này nhận vector key và value từ bộ mã hóa, và output từ layer phía dưới	Trong bộ giải mã còn có một multi-head attention khác có chức năng chú ý các từ ở bộ mã hóa, layer này nhận vector key và value từ bộ mã hóa, và output từ layer phía dưới
38	346	Đơn giản bởi vì chúng ta muốn so sánh sự tương quan giữa từ đang được dịch với các từ nguồn	Đơn giản bởi vì chúng ta muốn so sánh sự tương quan giữa từ đang được dịch với các từ nguồn
38	347	Ứng dụng Attention trong mô hình Transformer Mô hình Transformer sử dụng multi-head attention theo 3 cách khác nhau	Ứng dụng Attention trong mô hình Transformer Mô hình Transformer sử dụng multi-head attention theo 3 cách khác nhau
38	348	Thứ nhất là trong lớp “encoder-decoder attention”, câu truy vấn đến từ lớp giải mã trước đó và các khóa và giá trị đến từ đầu ra của bộ mã hóa	Thứ nhất là trong lớp “encoder-decoder attention”, câu truy vấn đến từ lớp giải mã trước đó và các khóa và giá trị đến từ đầu ra của bộ mã hóa
38	349	Điều này cho phép tất cả các vị trí trong bộ giải mã sẽ tham gia vào tất cả các vị trí trong chuỗi đầu vào	Điều này cho phép tất cả các vị trí trong bộ giải mã sẽ tham gia vào tất cả các vị trí trong chuỗi đầu vào
x	350	Nó tương tự như cơ chế encoder-decoder attention trong mô hình sequence-to- sequence	Nó tương tự như cơ chế encoder-decoder attention trong mô hình sequence-to- sequence
38	351	Tiếp đến là bộ mã hóa chứa các lớp self-attention	Tiếp đến là bộ mã hóa chứa các lớp self-attention
x	352	Trong một lớp self- attention, tất cả các khóa, giá trị và truy vấn đều đến từ cùng một nơi, trong trường hợp này đó là đầu ra của lớp trước đó trong mã hóa	Trong một lớp self- attention, tất cả các khóa, giá trị và truy vấn đều đến từ cùng một nơi, trong trường hợp này đó là đầu ra của lớp trước đó trong mã hóa
38	353	Mỗi vị trí trong bộ mã hóa có thể tham gia vào tất cả các vị trí trong lớp trước của bộ mã hóa	Mỗi vị trí trong bộ mã hóa có thể tham gia vào tất cả các vị trí trong lớp trước của bộ mã hóa
38	354	Ngoài ra, các lớp self-attention trong bộ giải mã cho phép mỗi vị trí trong bộ giải mã tham dự tất cả các vị trí trong bộ giải mã bao gồm cả vị trí của nó	Ngoài ra, các lớp self-attention trong bộ giải mã cho phép mỗi vị trí trong bộ giải mã tham dự tất cả các vị trí trong bộ giải mã bao gồm cả vị trí của nó
38	355	Chúng ta cần ngăn chặn luồng thông tin bên trái trong bộ giải mã để bảo toàn thuộc tính tự động hồi quy	Chúng ta cần ngăn chặn luồng thông tin bên trái trong bộ giải mã để bảo toàn thuộc tính tự động hồi quy
38	356	Điều này được thực hiện bên trong scaled dot-product attention bằng cách che đi tất cả các giá trị trong đầu vào của softmax tương ứng với các kết nối không hợp lệ	Điều này được thực hiện bên trong scaled dot-product attention bằng cách che đi tất cả các giá trị trong đầu vào của softmax tương ứng với các kết nối không hợp lệ

38	357	2.6 Bộ công cụ mã nguồn mở cho dịch máy mạng nơron OpenNMT Giới thiệu mô hình OpenNMT OpenNMT là bộ công cụ mã nguồn mở cho dịch máy mạng nơron (NMT) và sinh ngôn ngữ tự nhiên (NLG), được phát hành vào tháng 12 năm 2016 bởi nhóm NLP Harvard và SYSTRAN, kể từ đó nó đã được sử dụng trong một số nghiên cứu và ứng dụng công nghiệp	2.6 Bộ công cụ mã nguồn mở cho dịch máy mạng nơron OpenNMT Giới thiệu mô hình OpenNMT OpenNMT là bộ công cụ mã nguồn mở cho dịch máy mạng nơron (NMT) và sinh ngôn ngữ tự nhiên (NLG), được phát hành vào tháng 12 năm 2016 bởi nhóm NLP Harvard và SYSTRAN, kể từ đó nó đã được sử dụng trong một số nghiên cứu và ứng dụng công nghiệp
38	358	Hiện nay, nó đang được duy trì bởi SYSTRAN và Ubiqus [6]	Hiện nay, nó đang được duy trì bởi SYSTRAN và Ubiqus [6]
38	359	Bộ công cụ bao gồm nhiều mã nguồn để có thể xử lý toàn bộ quy trình công việc trong học máy: từ chuẩn bị dữ liệu đến tăng tốc suy luận (inference acceleration)	Bộ công cụ bao gồm nhiều mã nguồn để có thể xử lý toàn bộ quy trình công việc trong học máy: từ chuẩn bị dữ liệu đến tăng tốc suy luận (inference acceleration)
x	360	Với mục tiêu hỗ trợ nghiên cứu về kiến trúc mô hình, biểu diễn tính năng và phương thức nguồn, đồng thời duy trì tính ổn định của API và hiệu suất cạnh tranh cho các ứng dụng sản xuất thì hệ thống ưu tiên tính hiệu quả, tính mô-đun và khả năng mở rộng	Với mục tiêu hỗ trợ nghiên cứu về kiến trúc mô hình, biểu diễn tính năng và phương thức nguồn, đồng thời duy trì tính ổn định của API và hiệu suất cạnh tranh cho các ứng dụng sản xuất thì hệ thống ưu tiên tính hiệu quả, tính mô-đun và khả năng mở rộng
38	361	OpenNMT đã được sử dụng trong một số hệ thống MT sản xuất và được trích dẫn trong hơn 700 bài báo nghiên cứu	OpenNMT đã được sử dụng trong một số hệ thống MT sản xuất và được trích dẫn trong hơn 700 bài báo nghiên cứu
38	362	OpenNMT hỗ trợ một loạt các kiến trúc mô hình (transformer, lstm,…) và quy trình đào tạo cho dịch máy mạng nơron cũng như các tác vụ như sinh ngôn ngữ tự nhiên và mô hình hóa ngôn ngữ	OpenNMT hỗ trợ một loạt các kiến trúc mô hình (transformer, lstm,…) và quy trình đào tạo cho dịch máy mạng nơron cũng như các tác vụ như sinh ngôn ngữ tự nhiên và mô hình hóa ngôn ngữ
38	363	Cộng đồng mã nguồn mở về dịch máy mạng nơron rất đa dạng và cũng có một số dự án khác có chung mục tiêu tương tự với OpenNMT như Fairseq (Ott et al., 2019), Sockeye (Hieber et al., 2017) hay Marian (Junczys-Dowmunt et al., 2018)	Cộng đồng mã nguồn mở về dịch máy mạng nơron rất đa dạng và cũng có một số dự án khác có chung mục tiêu tương tự với OpenNMT như Fairseq (Ott et al., 2019), Sockeye (Hieber et al., 2017) hay Marian (Junczys-Dowmunt et al., 2018)
x	364	39 OpenNMT [7] được xây dựng dựa trên các nghiên cứu cải tiến mô hình NMT truyền thống, cho phép mô hình dịch tự động quan sát toàn bộ chuỗi đầu vào để khởi tạo những từ mới ở đầu ra, cho các kết quả tốt khi dịch các câu dài	39 OpenNMT [7] được xây dựng dựa trên các nghiên cứu cải tiến mô hình NMT truyền thống, cho phép mô hình dịch tự động quan sát toàn bộ chuỗi đầu vào để khởi tạo những từ mới ở đầu ra, cho các kết quả tốt khi dịch các câu dài
39	365	Đồng thời, OpenNMT cho phép tối ưu hóa bộ nhớ, tăng tốc độ tính toán khi sử dụng bộ xử lý đồ họa GPU	Đồng thời, OpenNMT cho phép tối ưu hóa bộ nhớ, tăng tốc độ tính toán khi sử dụng bộ xử lý đồ họa GPU
39	366	OpenNMT có hai phiên bản chính được duy trì và phát triển là : - OpenNMT-py: là phiên bản sử dụng PyTorch của OpenNMT, thân thiện với người dùng, mang tính dễ sử dụng và tính linh hoạt của PyTorch	OpenNMT có hai phiên bản chính được duy trì và phát triển là : - OpenNMT-py: là phiên bản sử dụng PyTorch của OpenNMT, thân thiện với người dùng, mang tính dễ sử dụng và tính linh hoạt của PyTorch
39	367	- OpenNMT-tf: là phiên bản sử dụng TensorFlow 2 của OpenNMT, một triển khai theo kiểu mô-đun, ổn định được hỗ trợ bởi hệ sinh thái TensorFlow 2	- OpenNMT-tf: là phiên bản sử dụng TensorFlow 2 của OpenNMT, một triển khai theo kiểu mô-đun, ổn định được hỗ trợ bởi hệ sinh thái TensorFlow 2
39	368	Hai phiên bản này đều cung cấp các tiện ích dòng lệnh và thư viện Python để cấu hình, huấn luyện và chạy các mô hình	Hai phiên bản này đều cung cấp các tiện ích dòng lệnh và thư viện Python để cấu hình, huấn luyện và chạy các mô hình

39	369	Mỗi phiên bản đều có thiết kế và bộ tính năng riêng, nhưng cả hai đều mang chung đặc điểm OpenNMT: dễ sử dụng, hiệu quả, tính mô-đun, tính hiệu quả và tính sẵn sàng sản xuất	Mỗi phiên bản đều có thiết kế và bộ tính năng riêng, nhưng cả hai đều mang chung đặc điểm OpenNMT: dễ sử dụng, hiệu quả, tính mô-đun, tính hiệu quả và tính sẵn sàng sản xuất
39	370	Các tính năng được hỗ trợ của hai phiên bản được so sánh trong dưới	Các tính năng được hỗ trợ của hai phiên bản được so sánh trong dưới
39	371	Hình 2-18: Các tính năng được OpenNMT-py (cột py) và OpenNMT-tf (cột tf) triển khai	37 Hình 2-18: Các tính năng được OpenNMT-py (cột py) và OpenNMT-tf (cột tf) triển khai
39	371	Hình 2-18: Các tính năng được OpenNMT-py (cột py) và OpenNMT-tf (cột tf) triển khai	Hình 2-18: Các tính năng được OpenNMT-py (cột py) và OpenNMT-tf (cột tf) triển khai
39	372	Trong khuôn khổ đồ án này, em đã chọn sử dụng OpenNMT-py vì tính dễ sử dụng và than thiện với người dùng hơn của nó	Trong khuôn khổ đồ án này, em đã chọn sử dụng OpenNMT-py vì tính dễ sử dụng và than thiện với người dùng hơn của nó
x	373	40 Tổng quan về mã OpenNMT-py Hình 2-19: Sơ đồ tổng quan về mã OpenNMT-py Tiền xử lý: Tiền xử lý dữ liệu là quá trình tạo ra các từ vựng và chuỗi chỉ số được sử dụng cho quá trình huấn luyện	40 Tổng quan về mã OpenNMT-py Hình 2-19: Sơ đồ tổng quan về mã OpenNMT-py Tiền xử lý: Tiền xử lý dữ liệu là quá trình tạo ra các từ vựng và chuỗi chỉ số được sử dụng cho quá trình huấn luyện
40	374	Quy trình gồm các bước sau: • Mã hóa (tokenization – cho tệp văn bản): tách tập tin thành các mã (token) được phân tách bằng dấu cách, có thể gắn với các đặc trưng	Quy trình gồm các bước sau: • Mã hóa (tokenization – cho tệp văn bản): tách tập tin thành các mã (token) được phân tách bằng dấu cách, có thể gắn với các đặc trưng
40	375	• Tiền xử lý: Xây dựng một tệp dữ liệu từ nguồn dữ liệu huấn luyện và kiểm định đã được mã hóa, có thể tùy chọn xáo trộn các câu và sắp xếp theo độ dài câu	• Tiền xử lý: Xây dựng một tệp dữ liệu từ nguồn dữ liệu huấn luyện và kiểm định đã được mã hóa, có thể tùy chọn xáo trộn các câu và sắp xếp theo độ dài câu
40	376	Mục tiêu chính của quá trình tiền xử lý là xây dựng bộ từ vựng với các từ, đặc trưng của từ và gán mỗi từ vào một chỉ mục trong những bộ từ điển này	Mục tiêu chính của quá trình tiền xử lý là xây dựng bộ từ vựng với các từ, đặc trưng của từ và gán mỗi từ vào một chỉ mục trong những bộ từ điển này
40	377	Kích thước bộ từ vựng mặc định là 50000, ta có thể chọn kích thước mong muốn cho bộ từ vựng bằng src_vocab_size và tgt_vocab_size	Kích thước bộ từ vựng mặc định là 50000, ta có thể chọn kích thước mong muốn cho bộ từ vựng bằng src_vocab_size và tgt_vocab_size
40	378	Huấn luyện: Ngoài các cài đặt kích thước tiêu chuẩn như số lớp, kích thước thứ nguyên ẩn,..	Huấn luyện: Ngoài các cài đặt kích thước tiêu chuẩn như số lớp, kích thước thứ nguyên ẩn,..
40	379	OpenNMT cũng cung cấp nhiều kiến trúc mô hình khác nhau	OpenNMT cũng cung cấp nhiều kiến trúc mô hình khác nhau
40	380	• Encoder (bộ mã hóa) Encoder trong OpenNMT bao gồm: - Default Encoder (encoder mặc định) - Bidirectional encoder (encoder hai chiều) - Pyramidal deep bidirectional encoder - Deep bidirectional encoder - Google's NMT encoder - Convolutional encoder (encoder tích chập) Encoder mặc định: là một cấu trúc RNN đơn giản (LSTM, GRU)	• Encoder (bộ mã hóa) Encoder trong OpenNMT bao gồm: - Default Encoder (encoder mặc định) - Bidirectional encoder (encoder hai chiều) - Pyramidal deep bidirectional encoder - Deep bidirectional encoder - Google's NMT encoder - Convolutional encoder (encoder tích chập) Encoder mặc định: là một cấu trúc RNN đơn giản (LSTM, GRU)
x	381	41 Bidirectional encoder (-encoder_type brnn): bao gồm hai bộ mã hóa độc lập: một mã hóa trình tự bình thường và một mã hóa trình tự đảo ngược	41 Bidirectional encoder (-encoder_type brnn): bao gồm hai bộ mã hóa độc lập: một mã hóa trình tự bình thường và một mã hóa trình tự đảo ngược

41	382	Trạng thái đầu ra và trạng thái cuối cùng được nối hoặc tổng hợp tùy thuộc vào -brnn_merge tùy chọn	Trạng thái đầu ra và trạng thái cuối cùng được nối hoặc tổng hợp tùy thuộc vào -brnn_merge tùy chọn
x	383	Hình 2-20: bidirectional encoder Pyramidal deep bidirectional encoder (-encoder_type pdbrnn): là một bộ mã hóa hai chiều thay thế giúp giảm thứ nguyên thời gian sau mỗi lớp dựa trên - pdbrnn_reduction và sử dụng -pdbrnn_merge để giảm	Hình 2-20: bidirectional encoder Pyramidal deep bidirectional encoder (-encoder_type pdbrnn): là một bộ mã hóa hai chiều thay thế giúp giảm thứ nguyên thời gian sau mỗi lớp dựa trên - pdbrnn_reduction và sử dụng -pdbrnn_merge để giảm
41	384	Hình 2-21:Pyramidal deep bidirectional encoder Deep bidirectional encoder (-encoder_type dbrnn): là một bộ mã hóa hai chiều thay thế trong đó kết quả đầu ra của mọi lớp được tổng hợp (hoặc nối) trước khi cấp cho lớp tiếp theo	Hình 2-21:Pyramidal deep bidirectional encoder Deep bidirectional encoder (-encoder_type dbrnn): là một bộ mã hóa hai chiều thay thế trong đó kết quả đầu ra của mọi lớp được tổng hợp (hoặc nối) trước khi cấp cho lớp tiếp theo
41	385	Đó là một trường hợp đặc biệt của Pyramidal deep bidirectional encoder mà không giảm thời gian (tức là -pdbrnn_reduction = 1)	Đó là một trường hợp đặc biệt của Pyramidal deep bidirectional encoder mà không giảm thời gian (tức là -pdbrnn_reduction = 1)
x	386	42 Hình 2-22: Deep bidirectional encoder Google's NMT encoder (-encoder_type gnmt): là bộ mã hóa có một lớp hai chiều như được mô tả trong [8]	42 Hình 2-22: Deep bidirectional encoder Google's NMT encoder (-encoder_type gnmt): là bộ mã hóa có một lớp hai chiều như được mô tả trong [8]
42	387	Các trạng thái hai chiều được nối và các kết nối dư được mặc định là có	Các trạng thái hai chiều được nối và các kết nối dư được mặc định là có
42	388	Hình 2-23: Google's NMT encoder Convolutional encoder (): là bộ mã hóa dựa trên một số lớp tích chập như được mô tả trong [9] • Decoder (Bộ giải mã) Bộ giải mã mặc định áp dụng attention trên chuỗi nguồn và thực hiện cấp dữ liệu đầu vào theo mặc định	Hình 2-23: Google's NMT encoder Convolutional encoder (): là bộ mã hóa dựa trên một số lớp tích chập như được mô tả trong [9] • Decoder (Bộ giải mã) Bộ giải mã mặc định áp dụng attention trên chuỗi nguồn và thực hiện cấp dữ liệu đầu vào theo mặc định
42	389	Input feeding (đầu vào cung cấp) là một cách tiếp cận để cung cấp các vectơ attention làm đầu vào cho các bước tiếp theo để thông báo cho mô hình về các quyết định liên kết trong quá khứ - "as inputs to the next time steps to inform the model about past alignment decisions" [10]	Input feeding (đầu vào cung cấp) là một cách tiếp cận để cung cấp các vectơ attention làm đầu vào cho các bước tiếp theo để thông báo cho mô hình về các quyết định liên kết trong quá khứ - "as inputs to the next time steps to inform the model about past alignment decisions" [10]
x	390	Có thể tắt nó bằng cách cài đặt - input_feed 0	Có thể tắt nó bằng cách cài đặt - input_feed 0
x	391	43 Hình 2-24: decoder mặc định Dịch (Transtale): Sau khi huấn luyện, model sẽ được lưu lại và khi dịch, model đã huấn luyện sẽ được sử dụng cùng với thuật toán beamsearch để cho ra bản dịch	43 Hình 2-24: decoder mặc định Dịch (Transtale): Sau khi huấn luyện, model sẽ được lưu lại và khi dịch, model đã huấn luyện sẽ được sử dụng cùng với thuật toán beamsearch để cho ra bản dịch
44	392	TỰ ĐỘNG PHÁT HIỆN VÀ SỬA LỖI CHÍNH TẢ DỰA VÀO MÔ HÌNH TRANSFORMER 3.1 Mô hình đề xuất Tổng quan mô hình Hình 3-1: Tổng quan mô hình đề xuất Bài toán sửa lỗi chính tả tiếng Việt có dữ liệu đầu vào là một câu sai chính tả và đầu ra là một câu đúng chính tả nên ta sẽ quy về bài toán seq2seq	TỰ ĐỘNG PHÁT HIỆN VÀ SỬA LỖI CHÍNH TẢ DỰA VÀO MÔ HÌNH TRANSFORMER 3.1 Mô hình đề xuất Tổng quan mô hình Hình 3-1: Tổng quan mô hình đề xuất Bài toán sửa lỗi chính tả tiếng Việt có dữ liệu đầu vào là một câu sai chính tả và đầu ra là một câu đúng chính tả nên ta sẽ quy về bài toán seq2seq

44	393	Do bộ dữ liệu tiếng Việt còn chứa nhiều từ tiếng Anh và các từ viết tắt tên của các tổ chức, thuật ngữ nên khi đưa trực tiếp vào huấn luyện sẽ làm cho độ chính xác của mô hình giảm đi nên em đã đánh dấu những từ tiếng Anh bằng nhãn ENG_ + số thứ tự và lưu các từ vào 1 tệp riêng và sau khi dịch sẽ gán lại các từ đó vào câu	Do bộ dữ liệu tiếng Việt còn chứa nhiều từ tiếng Anh và các từ viết tắt tên của các tổ chức, thuật ngữ nên khi đưa trực tiếp vào huấn luyện sẽ làm cho độ chính xác của mô hình giảm đi nên em đã đánh dấu những từ tiếng Anh bằng nhãn ENG_ + số thứ tự và lưu các từ vào 1 tệp riêng và sau khi dịch sẽ gán lại các từ đó vào câu
44	394	Mô hình đề xuất là mô hình Transformer có cấu trúc như hình dưới:	Mô hình đề xuất là mô hình Transformer có cấu trúc như hình dưới:
x	395	45 Hình 3-2: mô hình Transformer Trong bài toán này, em đã thử sử dụng BPE (một kỹ thuật nén dữ liệu hoạt động bằng cách thay thế các cặp byte liên tiếp có tần suất lớn bằng một byte không tồn tại trong dữ liệu) khi tiền xử lý dữ liệu	45 Hình 3-2: mô hình Transformer Trong bài toán này, em đã thử sử dụng BPE (một kỹ thuật nén dữ liệu hoạt động bằng cách thay thế các cặp byte liên tiếp có tần suất lớn bằng một byte không tồn tại trong dữ liệu) khi tiền xử lý dữ liệu
45	396	Nhưng do đây là bài toán sửa lỗi chính tả với dữ liệu đầu vào là dữ liệu sai chính tả (nó không có một quy tắc từ nhất định) nên khi áp dụng BPE sẽ không giúp ích trong việc nén dữ liệu hay xử lý được những từ chưa gặp bao giờ do khi nén dữ liệu đã tạo ra những quy tắc sai khiến độ chính xác mô hình giảm	Nhưng do đây là bài toán sửa lỗi chính tả với dữ liệu đầu vào là dữ liệu sai chính tả (nó không có một quy tắc từ nhất định) nên khi áp dụng BPE sẽ không giúp ích trong việc nén dữ liệu hay xử lý được những từ chưa gặp bao giờ do khi nén dữ liệu đã tạo ra những quy tắc sai khiến độ chính xác mô hình giảm
45	397	3.2 Môi trường thực nghiệm Việc huấn luyện các mô hình Deep Learning thì việc có GPU sử dụng là điều cần thiết	3.2 Môi trường thực nghiệm Việc huấn luyện các mô hình Deep Learning thì việc có GPU sử dụng là điều cần thiết
45	398	GPU cho phép xử lý phép tính song song nên sẽ nhanh hơn nhiều so với CPU	GPU cho phép xử lý phép tính song song nên sẽ nhanh hơn nhiều so với CPU
45	399	GPU sẽ hỗ trợ chạy những thuật toán Deep Learning rất tốt	GPU sẽ hỗ trợ chạy những thuật toán Deep Learning rất tốt
45	400	Và tất nhiên, thay vì chi tiền cho một GPU, chúng ta có thể sử dụng Google Colab	Và tất nhiên, thay vì chi tiền cho một GPU, chúng ta có thể sử dụng Google Colab
45	401	Đây thực sự là một điều tuyệt vời của Google khi cung cấp một service dựa trên Jupyter Notebooks và hỗ trợ GPU miễn phí	Đây thực sự là một điều tuyệt vời của Google khi cung cấp một service dựa trên Jupyter Notebooks và hỗ trợ GPU miễn phí
45	402	Điều này không chỉ là công cụ tuyệt vời giúp nâng cao khả năng code, mà nó còn cho phép người dùng phát triển các ứng dụng Deep Learning trên các thư viện phổ biến	Điều này không chỉ là công cụ tuyệt vời giúp nâng cao khả năng code, mà nó còn cho phép người dùng phát triển các ứng dụng Deep Learning trên các thư viện phổ biến
45	403	Google Colab là miễn phí nên trong một tiến trình kết nối sẽ bị giới hạn về thời gian kết nối cũng như giới hạn về lưu lượng	Google Colab là miễn phí nên trong một tiến trình kết nối sẽ bị giới hạn về thời gian kết nối cũng như giới hạn về lưu lượng
45	404	Hơn nữa khi dùng chúng chỉ cung cấp các GPU cơ bản nên tốc độ huấn luyện còn chậm	Hơn nữa khi dùng chúng chỉ cung cấp các GPU cơ bản nên tốc độ huấn luyện còn chậm
x	405	46 đồ án em đã sử dụng bản Google Colab Pro (bản trả phí) với cấu hình máy: 24GB RAM, Tesla V100-SXM2	46 đồ án em đã sử dụng bản Google Colab Pro (bản trả phí) với cấu hình máy: 24GB RAM, Tesla V100-SXM2
46	406	3.3 Xây dựng bộ dữ liệu Bộ dữ liệu thô Để huấn luyện mô hình ngôn ngữ, ta cần dữ liệu là văn bản để làm dữ liệu huấn luyện	3.3 Xây dựng bộ dữ liệu Bộ dữ liệu thô Để huấn luyện mô hình ngôn ngữ, ta cần dữ liệu là văn bản để làm dữ liệu huấn luyện
46	407	May mắn là trong bài toán này không cần dán nhãn cho các mô hình ngôn ngữ mà chỉ cần tập văn bản thô là được	May mắn là trong bài toán này không cần dán nhãn cho các mô hình ngôn ngữ mà chỉ cần tập văn bản thô là được

46	408	Trong phạm vi đồ án này em đã sử dụng bộ dữ liệu của viwikipedia file viwiki-20200501-pages-articles.xml.bz2	Trong phạm vi đồ án này em đã sử dụng bộ dữ liệu của viwikipedia file viwiki-20200501-pages-articles.xml.bz2
x	409	Em đã sử dụng trực tiếp tệp train_tieng_viet.txt từ nguồn https://drive.google.com/file/d/1- 7lE RkqCoID1691yCXLAOyZoJqYPqhGq/view	Em đã sử dụng trực tiếp tệp train_tieng_viet.txt từ nguồn https://drive.google.com/file/d/1- 7lE RkqCoID1691yCXLAOyZoJqYPqhGq/view
x	410	Đây là tệp dữ liệu viwwiki- 20200501 ở trên sau khi đã giải nén và chọn những câu hợp lệ do độ dữ liệu bao gồm cả những câu không đạt tiêu chuẩn vì chứa các ký tự tiếng Trung, Hàn, Nhật,… Tệp dữ liệu đã được lọc bỏ những câu không đạt tiêu chuẩn bằng cách tạo hàm kiểm tra tính hợp lệ của câu	Đây là tệp dữ liệu viwwiki- 20200501 ở trên sau khi đã giải nén và chọn những câu hợp lệ do độ dữ liệu bao gồm cả những câu không đạt tiêu chuẩn vì chứa các ký tự tiếng Trung, Hàn, Nhật,… Tệp dữ liệu đã được lọc bỏ những câu không đạt tiêu chuẩn bằng cách tạo hàm kiểm tra tính hợp lệ của câu
46	411	Sau khi lọc bỏ những câu không đạt tiêu chuẩn, bộ dữ liệu còn khoảng hơn 6 triệu câu	Sau khi lọc bỏ những câu không đạt tiêu chuẩn, bộ dữ liệu còn khoảng hơn 6 triệu câu
x	412	Đánh dấu từ tiếng Anh Vì trong bộ dữ liệu có chứa nhiều câu có cả những từ tiếng Anh hoặc từ viết tắt của tên các tổ chức,… nên sẽ ảnh hưởng nhiều đến kết quả sau khi sửa lỗi	Đánh dấu từ tiếng Anh Vì trong bộ dữ liệu có chứa nhiều câu có cả những từ tiếng Anh hoặc từ viết tắt của tên các tổ chức,… nên sẽ ảnh hưởng nhiều đến kết quả sau khi sửa lỗi
46	413	Vì vậy em đã sử dụng thư viện enchant để đánh dấu những từ tiếng Anh, thay chúng bằng ENG_stt để tăng hiệu suất của mô hình	Vì vậy em đã sử dụng thư viện enchant để đánh dấu những từ tiếng Anh, thay chúng bằng ENG_stt để tăng hiệu suất của mô hình
46	414	Tạo lỗi sai chính tả Để có thể huấn luyện cho mô hình, ta cần các câu chứa lỗi chính tả để huấn luyện cho mô hình	Tạo lỗi sai chính tả Để có thể huấn luyện cho mô hình, ta cần các câu chứa lỗi chính tả để huấn luyện cho mô hình
46	415	Do đó ta sẽ cài đặt hàm noise_maker () là hàm sẽ chuyển đổi các câu thành các câu có lỗi chính tả, nó sẽ được sử dụng làm dữ liệu đầu vào	Do đó ta sẽ cài đặt hàm noise_maker () là hàm sẽ chuyển đổi các câu thành các câu có lỗi chính tả, nó sẽ được sử dụng làm dữ liệu đầu vào
46	416	Các lỗi chính tả này được tạo ra trong hàm này theo một trong bốn cách sau: - Tạo từ viết tắt, teencode (Ví dụ: đc ~ được, ko ~ không,…) - Thứ tự của hai ký tự sẽ được đổi chỗ (kôhng ~ không) - Một ký tự sẽ được thêm vào (ytuổi ~ tuổi) - Một ký tự sẽ bị loại bỏ (ến ~ đến) Khả năng xảy ra của lỗi tạo từ viết tắt, teencode là 3.5 % và các lỗi còn lại có xác suất xảy ra như nhau là 7% Cuối cùng, tạo ra các lô dữ liệu sai chính tả bằng hàm noise_maker ()	Các lỗi chính tả này được tạo ra trong hàm này theo một trong bốn cách sau: - Tạo từ viết tắt, teencode (Ví dụ: đc ~ được, ko ~ không,…) - Thứ tự của hai ký tự sẽ được đổi chỗ (kôhng ~ không) - Một ký tự sẽ được thêm vào (ytuổi ~ tuổi) - Một ký tự sẽ bị loại bỏ (ến ~ đến) Khả năng xảy ra của lỗi tạo từ viết tắt, teencode là 3.5 % và các lỗi còn lại có xác suất xảy ra như nhau là 7% Cuối cùng, tạo ra các lô dữ liệu sai chính tả bằng hàm noise_maker ()
46	417	Sau đó ta chia bộ dữ liệu ra làm 3 tập: + Tập huấn luyện: gồm 6500000 câu + Tập kiểm định: gồm 200000 câu + Tập kiểm thử: gồm 10000 câu	Sau đó ta chia bộ dữ liệu ra làm 3 tập: + Tập huấn luyện: gồm 6500000 câu + Tập kiểm định: gồm 200000 câu + Tập kiểm thử: gồm 10000 câu
x	418	3.4 Huấn luyện mô hình https://drive.google.com/file/d/1-7lERkqCoID1691yCXLAOyZoJqYPqhGq/view https://drive.google.com/file/d/1-7lERkqCoID1691yCXLAOyZoJqYPqhGq/view	46 3.4 Huấn luyện mô hình ....
x	418	3.4 Huấn luyện mô hình https://drive.google.com/file/d/1-7lERkqCoID1691yCXLAOyZoJqYPqhGq/view https://drive.google.com/file/d/1-7lERkqCoID1691yCXLAOyZoJqYPqhGq/view	3.4 Huấn luyện mô hình https://drive.google.com/file/d/1-7lERkqCoID1691yCXLAOyZoJqYPqhGq/view https://drive.google.com/file/d/1-7lERkqCoID1691yCXLAOyZoJqYPqhGq/view
x	419	47 Trong mô hình em đã sử dụng thư viện mã nguồn mở OpenNMT-py để huấn luyện	47 Trong mô hình em đã sử dụng thư viện mã nguồn mở OpenNMT-py để huấn luyện

47	420	Em đã chọn OpenNMT-py để thực hiện đồ án vì: nó có thể mở rộng và triển khai nhanh chóng với tính dễ sử dụng của PyTorch	Em đã chọn OpenNMT-py để thực hiện đồ án vì: nó có thể mở rộng và triển khai nhanh chóng với tính dễ sử dụng của PyTorch
47	421	Em đã tiến hành thử nghiệm với những kịch bản sau: • Kịch bản 1: Tiền xử lý dữ liệu với BPE và huấn luyện bằng mô hình Transformer • Kịch bản 2: Tiền xử lý dữ liệu với BPE, đánh dấu từ tiếng Anh và huấn luyện bằng mô hình Transformer • Kịch bản 3: Đánh dấu từ tiếng Anh và huấn luyện bằng mô hình Transformer ở mức từ • Kịch bản 4: Tiền xử lý dữ liệu với PhoBERT BPE Đối với tất cả các thử nghiệm các tham số huấn luyện chính của mô hình được cài đặt như sau: - Kích thước lô (batch size): một tập dữ liệu huấn luyện có thể chia nhỏ thành các batch, mỗi một batch sẽ chứa các training samples, số lượng các samples này được gọi là batch size	Em đã tiến hành thử nghiệm với những kịch bản sau: • Kịch bản 1: Tiền xử lý dữ liệu với BPE và huấn luyện bằng mô hình Transformer • Kịch bản 2: Tiền xử lý dữ liệu với BPE, đánh dấu từ tiếng Anh và huấn luyện bằng mô hình Transformer • Kịch bản 3: Đánh dấu từ tiếng Anh và huấn luyện bằng mô hình Transformer ở mức từ • Kịch bản 4: Tiền xử lý dữ liệu với PhoBERT BPE Đối với tất cả các thử nghiệm các tham số huấn luyện chính của mô hình được cài đặt như sau: - Kích thước lô (batch size): một tập dữ liệu huấn luyện có thể chia nhỏ thành các batch, mỗi một batch sẽ chứa các training samples, số lượng các samples này được gọi là batch size
47	422	Việc lựa chọn batch size lớn hay nhỏ sẽ ảnh hưởng đến tốc độ tính toán và thông lượng đào tạo	Việc lựa chọn batch size lớn hay nhỏ sẽ ảnh hưởng đến tốc độ tính toán và thông lượng đào tạo
47	423	Tốc độ tính toán sẽ giảm dần khi tăng dần batch size bởi không phải tất cả hoạt động của các GPU đều hoạt động song song theo lô	Tốc độ tính toán sẽ giảm dần khi tăng dần batch size bởi không phải tất cả hoạt động của các GPU đều hoạt động song song theo lô
47	424	Ngược lại, thông lượng đào tạo tăng tuyến tính với kích thước của lô	Ngược lại, thông lượng đào tạo tăng tuyến tính với kích thước của lô
47	425	Ở đây sẽ lựa chọn batch size bằng 4096	Ở đây sẽ lựa chọn batch size bằng 4096
47	426	- Bộ mã hóa và giải mã là tổng hợp xếp chồng lên nhau của 6 layer	- Bộ mã hóa và giải mã là tổng hợp xếp chồng lên nhau của 6 layer
47	427	Mỗi layer bao gồm 2 layer con (sub-layer) trong đó sub-layer đầu tiên là multi-head self-attention với số head là 8	Mỗi layer bao gồm 2 layer con (sub-layer) trong đó sub-layer đầu tiên là multi-head self-attention với số head là 8
47	428	Đầu ra của mỗi sub-layer này sẽ có số chiều là 1024	Đầu ra của mỗi sub-layer này sẽ có số chiều là 1024
47	429	Bảng 3-1: Các tham số sử dụng huấn luyện mô hình Tham số Giá trị N 6 hiden size 1024 batch size 4096 num head 8 optimizer adam warmup_steps 16.000	Bảng 3-1: Các tham số sử dụng huấn luyện mô hình Tham số Giá trị N 6 hiden size 1024 batch size 4096 num head 8 optimizer adam warmup_steps 16.000
x	430	48 train steps 150.000 learning_rate 2 save_checkpoint_steps 1000 word_vec_size 512 3.5 Kết quả đánh giá Phương pháp đánh giá Sau khi tham khảo các nghiên cứu khác, em đã chọn điểm BLEU score để đánh giá mô hình này vì nó là thang điểm thông dụng hay được dùng cho bài toán sửa lỗi chính tả	48 train steps 150.000 learning_rate 2 save_checkpoint_steps 1000 word_vec_size 512 3.5 Kết quả đánh giá Phương pháp đánh giá Sau khi tham khảo các nghiên cứu khác, em đã chọn điểm BLEU score để đánh giá mô hình này vì nó là thang điểm thông dụng hay được dùng cho bài toán sửa lỗi chính tả
x	431	BLEU (Bilingual Evaluation Understudy Score) được Kishore Papineni và cộng sự đề xuất lần đầu vào năm 2002 qua bài nghiên cứu “A Method for Automatic Evaluation of Machine Translation”	BLEU (Bilingual Evaluation Understudy Score) được Kishore Papineni và cộng sự đề xuất lần đầu vào năm 2002 qua bài nghiên cứu “A Method for Automatic Evaluation of Machine Translation”
x	432	BLEU được tính dựa trên số lượng n-grams giống nhau giữa câu dịch của mô hình (output) với các câu tham chiếu tương ứng (label) có xét tới yếu tố độ dài của câu	BLEU được tính dựa trên số lượng n-grams giống nhau giữa câu dịch của mô hình (output) với các câu tham chiếu tương ứng (label) có xét tới yếu tố độ dài của câu

x	433	Số n-grams tối đa của BLEU là không giới hạn, nhưng vì xét về ý nghĩa, cụm từ quá dài thường không có nhiều ý nghĩa, và nghiên cứu cũng đã cho thấy là với 4-gram, điểm số BLEU trung bình cho khả năng dịch thuật của con người cũng đã giảm khá nhiều nên n-grams tối đa thường được sử dụng là 4-gram	Số n-grams tối đa của BLEU là không giới hạn, nhưng vì xét về ý nghĩa, cụm từ quá dài thường không có nhiều ý nghĩa, và nghiên cứu cũng đã cho thấy là với 4-gram, điểm số BLEU trung bình cho khả năng dịch thuật của con người cũng đã giảm khá nhiều nên n-grams tối đa thường được sử dụng là 4-gram
x	434	Công thức để tính điểm đánh giá như sau: □□□□□ = □□□ {∑□□ log(□□) − max (□□□□ □□□□ − 1.0) □ □=1 } □□ = ∑ □□□□ ∑ □□□□ Trong đó: • □□□ là số lượng các n-grams trong phân đoạn j của bản dịch dùng để tham khảo • □□□ là số lượng các n-grams trong phân đoạn j của bản dịch bằng máy	Công thức để tính điểm đánh giá như sau: □□□□□ = □□□ {∑□□ log(□□) − max (□□□□ □□□□ − 1.0) □ □=1 } □□ = ∑ □□□□ ∑ □□□□ Trong đó: • □□□ là số lượng các n-grams trong phân đoạn j của bản dịch dùng để tham khảo • □□□ là số lượng các n-grams trong phân đoạn j của bản dịch bằng máy
x	435	• □□□□là số lượng các n-grams trong phân đoạn j của bản dịch bằng máy	• □□□□là số lượng các n-grams trong phân đoạn j của bản dịch bằng máy
x	436	• □□□□ là số lượng các từ trong bản dịch bằng máy	• □□□□ là số lượng các từ trong bản dịch bằng máy
x	437	Giá trị score đánh giá mức độ tương ứng giữa hai bản dịch và nó được thực hiện trên từng phân đoạn, ở đây phân đoạn được hiểu là đơn vị tối thiểu trong các bản dịch, thông thường mỗi phân đoạn là một câu hoặc một đoạn	Giá trị score đánh giá mức độ tương ứng giữa hai bản dịch và nó được thực hiện trên từng phân đoạn, ở đây phân đoạn được hiểu là đơn vị tối thiểu trong các bản dịch, thông thường mỗi phân đoạn là một câu hoặc một đoạn
x	438	Việc thống kê đồ trùng khớp của các n-grams dựa trên tập hợp các ngrams trên các phân đoạn,	Việc thống kê đồ trùng khớp của các n-grams dựa trên tập hợp các ngrams trên các phân đoạn,
x	439	49 trước hết là nó được tính trên từng phân đoạn, sau đó tính lại giá trị này trên tất cả các phân đoạn	49 trước hết là nó được tính trên từng phân đoạn, sau đó tính lại giá trị này trên tất cả các phân đoạn
49	440	Kết quả Dưới đây là điểm số BLEU khi sử dụng hệ thống để tự động sửa lỗi chính tả câu	Kết quả Dưới đây là điểm số BLEU khi sử dụng hệ thống để tự động sửa lỗi chính tả câu
49	441	Để đánh giá hệ thống, em đã sử dụng cùng bộ dữ liệu kiểm thử	Để đánh giá hệ thống, em đã sử dụng cùng bộ dữ liệu kiểm thử
52	442	- Áp dụng mô hình Transformer và tối ưu các tham số của mô hình cho bài toán tự động sửa lỗi chính tả Tiếng Việt	Nội dung chính của đồ án là áp dụng mô hình Transformer cho bài toán tự động phát hiện và sửa lỗi chính tả Tiếng Việt

Kết quả kiểm trùng với tài liệu: Tài liệu hệ thống - final_datn_ngadt_20167304_2.7m.txt

Tỉ lệ sao chép: **10.235%**

Trang	Chỉ số	Tài liệu kiểm tra	Tài liệu gốc
x	92	16 - Cấu trúc của một âm tiết theo cách tiếp cận 3 thành phần sẽ được viết lại như sau: Âm tiết = [Âm đầu][Âm cuối] Trong đó những thành phần nằm trong cặp dấu <> là bắt buộc phải có, những thành phần nằm trong cặp dấu [ ] thì có thể có hoặc không	Vì vậy, một cách tiếp cận mới ngắn gọn hơn để dễ dàng cho việc kiểm tra chính tả trên máy tính được đưa ra đó là cách tiếp cận theo hướng tổ hợp âm (cấu trúc âm tiết ba thành phần) Cấu trúc của một âm tiết theo cách tiếp cận 3 thành phần sẽ được viết lại như sau: Âm tiết = [Âm đầu][Âm cuối] Trong đó những thành phần nằm trong cặp dấu <> là bắt buộc phải có, những thành phần nằm trong cặp dấu [ ] thì có thể có hoặc không Trong cấu trúc âm tiết 3 thành phần thì tổ hợp âm giữa là thành phần bắt buộc phải có, nó là thành phần chính cấu tạo nên âm tiết
16	93	• Tổ chức lưu trữ luật âm tiết Dựa trên những phân tích về âm tiết 3 thành phần, chúng ta có thể tổ chức lưu trữ từ điển luật theo Tổ hợp âm giữa trên file dữ liệu như sau: Structure CT_AM Tong_Am_Dau : LongInt To_Hop_Am_Giua : String(3) Tong_Am_Cuoi : LongInt End Structure Trong đó: Tong_Am_Dau là giá trị tổng của các Âm đầu có thể đi với tổ hợp âm giữa Tong_Am_Cuoi là giá trị tổng của các Âm cuối có thể đi với tổ hợp âm giữa Lưu cấu trúc âm này (có sắp xếp) thành một từ điển các cấu trúc âm để sau này chúng ta kiểm tra các âm tiết ở trong từ điển	Dựa trên những phân tích về âm tiết 3 thành phần, chúng ta có thể tổ chức lưu trữ từ điển luật theo Tổ hợp âm giữa trên file dữ liệu như sau: Structure CT_AM Tong_Am_Dau : LongInt To_Hop_Am_Giua : String(3) Tong_Am_Cuoi : LongInt End Structure Trong đó: Tong_Am_Dau là giá trị tổng của các Âm đầu có thể đi với tổ hợp âm giữa Tong_Am_Cuoi là giá trị tổng của các Âm cuối có thể đi với tổ hợp âm giữa Lưu cấu trúc âm này (có sắp xếp) thành một từ điển các cấu trúc âm để sau này chúng ta kiểm tra các âm tiết ở trong từ điển
16	95	- Phương pháp: + Bước 1: Tách âm tiết ra làm 3 phần: âm đầu, tổ hợp âm giữa, âm cuối và chuyển thành một cấu trúc âm tiết X, tương ứng theo âm đầu, tổ hợp âm giữa và âm cuối	Phương pháp: 1) Tách âm tiết ra làm 3 phần: âm đầu, tổ hợp âm giữa, âm cuối và chuyển thành một cấu trúc âm tiết X, tương ứng theo âm đầu, tổ hợp âm giữa và âm cuối
16	96	+ Bước 2: Tìm tổ hợp âm giữa trong từ điển theo phương pháp tìm kiếm nhị phân	2) Tìm tổ hợp âm giữa trong từ điển theo phương pháp tìm kiếm nhị phân
16	97	+ Bước 3: Nếu tìm thấy thì tiếp tục bước 4, nếu không thì nhảy đến bước 6	3) Nếu tìm thấy thì tiếp tục bước 4, nếu không thì nhảy đến bước 6
16	98	+ Bước 4: Ta lấy được một cấu trúc âm tiết CTAM tương ứng trong từ điển	4) Ta lấy được một cấu trúc âm tiết CT_AM tương ứng trong từ điển
16	99	+ Bước 5: Kiểm tra xem âm đầu, âm cuối của X có trong trong cấu trúc âm tiết CTAM đó hay không	5) Kiểm tra xem âm đầu, âm cuối của X có trong trong cấu trúc âm tiết CT_AM đó hay không
16	100	Nếu có thì kết luận là âm tiết đúng, nhảy đến bước 7	Nếu có thì kết luận là âm tiết đúng, nhảy đến bước 7
16	101	+ Bước 6: Kết luận âm tiết sai	11 6) Kết luận âm tiết sai
16	102	+ Bước 7: Kết thúc Việc kiểm tra toàn bộ các âm tiết của văn bản là việc kiểm tra tất cả các âm tiết có trong từ điển hay không	Việc kiểm tra toàn bộ các âm tiết của văn bản là việc kiểm tra tất cả các âm tiết có trong từ điển hay không
16	103	Với phương pháp này chúng ta kiểm tra được tất cả các âm tiết trong văn bản có đúng chính tả hay không	Với phương pháp này chúng ta kiểm tra được tất cả các âm tiết trong văn bản có đúng chính tả hay không

19	124	CƠ SỞ LÝ THUYẾT 2.1 Dịch máy mạng nơron (NMT) NMT (Neural Machine Translation) là sự kết hợp của dịch máy (Machine Translation - MT) và mạng nơron nhân tạo (Artificial Neural Network - NN)	Mô hình NMT NMT (Neural Machine Translation) là sự kết hợp của dịch máy (Machine Translation - MT) và mạng nơ-ron nhân tạo [20] (Artificial Neural Network - NN)
25	216	Nguyên lý hoạt động của BPE dựa trên phân tích trực quan rằng hầu hết các từ đều có thể phân tích thành các thành phần con	Nguyên lý hoạt động của BPE dựa trên phân tích trực quan rằng hầu hết các từ đều có thể phân tích thành các thành phần con
x	217	26 lowest đều là hợp thành bởi low và những đuôi phụ er, est	Chẳng hạn như từ: low, lower, lowest đều là hợp thành bởi low và những đuôi phụ er, est
26	218	Những đuôi này rất thường xuyên xuất hiện ở các từ	Những đuôi này rất thường xuyên xuất hiện ở các từ
26	219	Như vậy khi biểu diễn từ lower chúng ta có thể mã hóa chúng thành hai thành phần từ phụ (subwords) tách biệt là low và er	Như vậy khi biểu diễn từ lower chúng ta có thể mã hóa chúng thành hai thành phần từ phụ (subwords) tách biệt là low và er
26	220	Theo cách biểu diễn này sẽ không phát sinh thêm một index mới cho từ lower và đồng thời tìm được mối liên hệ giữa lower, lowest và low nhờ có chung thành phần từ phụ là low	Theo cách biểu diễn này sẽ không phát sinh thêm một index mới cho từ lower và đồng thời tìm được mối liên hệ giữa lower, lowest và low nhờ có chung thành phần từ phụ là low
26	221	Phương pháp BPE sẽ thống kê tần suất xuất hiện của các từ phụ cùng nhau và tìm cách gộp chúng lại nếu tần suất xuất hiện của chúng là lớn nhất	Phương pháp BPE sẽ thống kê tần suất xuất hiện của các từ phụ cùng nhau và tìm cách gộp chúng lại nếu tần suất xuất hiện của chúng là lớn nhất
26	222	Cứ tiếp tục quá trình gộp từ phụ cho tới khi không tồn tại các subword để gộp nữa, ta sẽ thu được tập subwords cho toàn bộ bộ văn bản mà mọi từ đều có thể biểu diễn được thông qua subwords	35 tục quá trình gộp từ phụ cho tới khi không tồn tại các subword để gộp nữa, ta sẽ thu được tập subwords cho toàn bộ bộ văn bản mà mọi từ đều có thể biểu diễn được thông qua subwords
26	223	Quá trình này gồm các bước như sau: • Bước 1: Khởi tạo bộ từ điển	Quá trình này gồm các bước như sau: Bước 1: Khởi tạo từ điển (vocabulary)
26	224	• Bước 2: Biểu diễn mỗi từ trong bộ văn bản bằng cách kết hợp của các ký tự @@ ở cuối cùng để đánh dấu sự kết thúc của một từ	Bước 2: Biểu diễn mỗi từ trong bộ văn bản bằng kết hợp của các ký tự với token <\w> ở cuối cùng đánh dấu kết thúc một từ (lý do thêm token sẽ được giải thích bên dưới)
26	225	• Bước 3: Thống kê tần suất xuất hiện theo cặp của toàn bộ token theo trong bộ từ điển	Bước 3: Thống kê tần suất xuất hiện theo cặp của toàn bộ token trong từ điển
26	226	• Bước 4: Gộp các cặp câu có tần suất xuất hiện lớn nhất để tạo thành một n-gram theo level character cho từ điển	Bước 4: Gộp các cặp có tần suất xuất hiện lớn nhất để tạo thành một n- gram theo level character mới cho từ điển
26	227	• Bước 5: Lặp lại bước 3 và bước 4 cho tới khi số bước triển khai merge đạt đỉnh hoặc kích thước kỳ vọng của từ điển đạt được	Bước 5: Lặp lại bước 3 và bước 4 cho tới khi số bước triển khai merge đạt đỉnh hoặc kích thước kỳ vọng của từ điển đạt được
x	318	Embedding Layer Đầu tiên, các từ được biểu diễn bằng một vector sử dụng một ma trận word embedding có số dòng bằng kích thước của tập từ vựng	24 được biểu diễn bằng một vector sử dụng một ma trận word embedding có số dòng bằng kích thước của tập từ vựng
x	319	Sau đó các từ trong câu được tìm kiếm trong ma trận này, và được nối nhau thành các dòng của một ma trận 2 chiều chứa ngữ nghĩa của từng từ riêng biệt	Sau đó các từ trong câu được tìm kiếm trong ma trận này, và được nối nhau thành các dòng của một ma trận 2 chiều chứa ngữ nghĩa của từng từ riêng biệt

x	320	Nhưng như các bạn đã thấy, transformer xử lý các từ song song, do đó, với chỉ word embedding mô hình không thể nào biết được vị trí các từ	Nhưng Transformer xử lý các từ song song, do đó, chỉ với Word Embedding thì mô hình không thể nào biết được vị trí các từ
x	321	Như vậy, chúng ta cần một cơ chế nào đó để đưa thông tin vị trí các từ vào trong vector đầu vào	Như vậy, cần một cơ chế nào đó để đưa thông tin vị trí các từ vào trong vector đầu vào
x	324	Vị trí của các từ được mã hóa bằng một vector có kích thước bằng word embedding và được cộng trực tiếp vào word embedding	Vị trí của các từ được mã hóa bằng một vector có kích thước bằng word embedding và được cộng trực tiếp vào word embedding
38	357	2.6 Bộ công cụ mã nguồn mở cho dịch máy mạng nơron OpenNMT Giới thiệu mô hình OpenNMT OpenNMT là bộ công cụ mã nguồn mở cho dịch máy mạng nơron (NMT) và sinh ngôn ngữ tự nhiên (NLG), được phát hành vào tháng 12 năm 2016 bởi nhóm NLP Harvard và SYSTRAN, kể từ đó nó đã được sử dụng trong một số nghiên cứu và ứng dụng công nghiệp	CÁCH TIẾP CẬN ĐỀ XUẤT 3.1 Giới thiệu mô hình OpenNMT OpenNMT là bộ công cụ mã nguồn mở cho dịch máy bằng mạng nơron, được release lần đầu tiên vào tháng 12 năm 2016 bởi nhóm NLP Harvard và SYSTRAN, dự án kể từ đó đã được sử dụng trong một số nghiên cứu và ứng dụng trong ngành
38	358	Hiện nay, nó đang được duy trì bởi SYSTRAN và Ubiqus [6]	Hiện nay, nó được duy trì bởi SYSTRAN và Ubiqus
x	364	39 OpenNMT [7] được xây dựng dựa trên các nghiên cứu cải tiến mô hình NMT truyền thống, cho phép mô hình dịch tự động quan sát toàn bộ chuỗi đầu vào để khởi tạo những từ mới ở đầu ra, cho các kết quả tốt khi dịch các câu dài	OpenNMT được xây dựng dựa trên các nghiên cứu cải tiến mô hình NMT [19] truyền thống, cho phép mô hình dịch tự động quan sát toàn bộ chuỗi đầu vào để khởi tạo những từ mới ở đầu ra, cho các kết quả tốt khi dịch các câu dài
39	365	Đồng thời, OpenNMT cho phép tối ưu hóa bộ nhớ, tăng tốc độ tính toán khi sử dụng bộ xử lý đồ họa GPU	Đồng thời, OpenNMT cho phép tối ưu hóa bộ nhớ, tăng tốc độ tính toán khi sử dụng bộ xử lý đồ họa GPU
x	373	40 Tổng quan về mã OpenNMT-py Hình 2-19: Sơ đồ tổng quan về mã OpenNMT-py Tiền xử lý: Tiền xử lý dữ liệu là quá trình tạo ra các từ vựng và chuỗi chỉ số được sử dụng cho quá trình huấn luyện	Tiền xử lý Việc tiền xử lý là để tạo ra các từ vựng và chuỗi chỉ số được sử dụng cho quá trình huấn luyện
40	374	Quy trình gồm các bước sau: • Mã hóa (tokenization – cho tệp văn bản): tách tập tin thành các mã (token) được phân tách bằng dấu cách, có thể gắn với các đặc trưng	Quy trình chung bao gồm một số bước như sau: • Mã hóa (đối với tệp văn bản): là tách tập tin thành các mã thông báo được phân tách bằng dấu cách, có thể được liên kết với các tính năng
40	375	• Tiền xử lý: Xây dựng một tệp dữ liệu từ nguồn dữ liệu huấn luyện và kiểm định đã được mã hóa, có thể tùy chọn xáo trộn các câu và sắp xếp theo độ dài câu	• Tiền xử lý: là xây dựng một tệp dữ liệu từ kho ngữ liệu xác thực và đào tạo nguồn được mã hóa, tùy chọn xáo trộn các câu và sắp xếp theo độ dài câu
40	376	Mục tiêu chính của quá trình tiền xử lý là xây dựng bộ từ vựng với các từ, đặc trưng của từ và gán mỗi từ vào một chỉ mục trong những bộ từ điển này	Mục tiêu chính của quá trình tiền xử lý là xây dựng từ và tính năng từ vựng và gán mỗi từ vào một chỉ mục trong các từ điển này

40	380	• Encoder (bộ mã hóa) Encoder trong OpenNMT bao gồm: - Default Encoder (encoder mặc định) - Bidirectional encoder (encoder hai chiều) - Pyramidal deep bidirectional encoder - Deep bidirectional encoder - Google's NMT encoder - Convolutional encoder (encoder tích chập) Encoder mặc định: là một cấu trúc RNN đơn giản (LSTM, GRU)	Encoder Encoder trong OpenNMT bao gồm: • Default Encoder (encoder mặc định) • Bidirectional encoder (encoder hai chiều) • Pyramidal deep bidirectional encoder (encoder hai chiều sâu hình chóp) • Deep bidirectional encoder (encoder hai chiều sâu) • Google's NMT encoder (encoder NMT của Google) • Convolutional encoder (encoder tích chập) Encoder mặc định của model trong OpenNMT là mô hình RNN đơn giản (LSTM hoặc GRU), ngoài ra còn có các lựa chọn encoder khác như Transformer, Transformer Language Model ..
x	381	41 Bidirectional encoder (-encoder_type brnn): bao gồm hai bộ mã hóa độc lập: một mã hóa trình tự bình thường và một mã hóa trình tự đảo ngược	Bidirectional encoder (-encoder_type brnn) bao gồm hai bộ encoder độc lập: một encoder trình tự bình thường và một encoder trình tự đảo ngược
41	382	Trạng thái đầu ra và trạng thái cuối cùng được nối hoặc tổng hợp tùy thuộc vào -brnn_merge tùy chọn	Đầu ra và trạng thái cuối cùng được nối hoặc tổng hợp tùy thuộc vào tùy chọn - brnn_merge
x	383	Hình 2-20: bidirectional encoder Pyramidal deep bidirectional encoder (-encoder_type pdbrnn): là một bộ mã hóa hai chiều thay thế giúp giảm thứ nguyên thời gian sau mỗi lớp dựa trên - pdbrnn_reduction và sử dụng -pdbrnn_merge để giảm	Hình 3.5 Bidirectional encoder Pyramidal deep bidirectional encoder (-encoder_type pdbrnn) là một bộ mã hóa hai chiều thay thế giúp giảm thứ nguyên thời gian sau mỗi lớp dựa trên yếu tố -pdbrnn_reduction và sử dụng -pdbrnn_merge làm hành động giảm (tổng hoặc nối)
41	384	Hình 2-21:Pyramidal deep bidirectional encoder Deep bidirectional encoder (-encoder_type dbrnn): là một bộ mã hóa hai chiều thay thế trong đó kết quả đầu ra của mọi lớp được tổng hợp (hoặc nối) trước khi cấp cho lớp tiếp theo	31 Hình 3.6 Pyramidal deep bidirectional encoder Deep bidirectional encoder (-encoder_type dbrnn) là một bộ mã hóa hai chiều thay thế trong đó kết quả đầu ra của mọi lớp được tổng hợp (hoặc nối) trước khi cấp cho lớp tiếp theo
41	385	Đó là một trường hợp đặc biệt của Pyramidal deep bidirectional encoder mà không giảm thời gian (tức là -pdbrnn_reduction = 1)	Đây là một trường hợp đặc biệt của bộ mã hóa hai chiều hình chóp sâu mà không giảm thời gian (tức là -pdbrnn_reduction = 1)
x	386	42 Hình 2-22: Deep bidirectional encoder Google's NMT encoder (-encoder_type gnmt): là bộ mã hóa có một lớp hai chiều như được mô tả trong [8]	Hình 3.7 Deep bidirectional encoder Bộ mã hóa của Google (-encoder_type gnmt) là bộ mã hóa có một lớp hai chiều như được mô tả trong Wu et al
42	387	Các trạng thái hai chiều được nối và các kết nối dư được mặc định là có	Các trạng thái hai chiều được nối và các kết nối dư được bật theo mặc định
x	390	Có thể tắt nó bằng cách cài đặt - input_feed 0	Có thể tắt điều này bằng cách đặt -input_feed 0
x	395	45 Hình 3-2: mô hình Transformer Trong bài toán này, em đã thử sử dụng BPE (một kỹ thuật nén dữ liệu hoạt động bằng cách thay thế các cặp byte liên tiếp có tần suất lớn bằng một byte không tồn tại trong dữ liệu) khi tiền xử lý dữ liệu	Đây là là một kỹ thuật nén dữ liệu hoạt động bằng cách thay thế các cặp byte liên tiếp có tần suất lớn bằng một byte không tồn tại trong dữ liệu
53	443	[2] Nguyễn Gia Định, Trần Thanh Lương, "THUẬT TOÁN KIỂM TRA ÂM TIẾT TIẾNG VIỆT DỰA TRÊN LUẬT CẤU TẠO ÂM TIẾT," 2016	10 Kiểm tra lỗi chính tả dựa trên luật cấu tạo âm tiết tiếng Việt ..

Kết quả kiểm trùng với tài liệu:
https://tailieu.vn/docview/tailieu/2014/20140828/votinhdon91/tom_tat_la__8157.pdf

Tỉ lệ sao chép: **0.640%**

Trang	Chỉ số	Tài liệu kiểm tra	Tài liệu gốc
19	132	Hình 2-1: Mô hình mạng nơron đơn giản Mạng nơron nhân tạo được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó	Nó được tạo lên từ một số lượng lớn các phần tử (gọi là nơron) kết nối với nhau thông qua các liên kết (gọi là trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó
19	133	Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện	Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện
19	134	Về bản chất "học" chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron	Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron sao cho giá trị hàm lỗi là nhỏ nhất

Kết quả kiểm trùng với tài liệu: Tài liệu hệ thống - letuananh_1.4m.txt

Tỉ lệ sao chép: **0.640%**

Trang	Chỉ số	Tài liệu kiểm tra	Tài liệu gốc
15	89	- Do bộ từ điển được tạo thủ công nên khó có thể bao quát hết được tất cả các lỗi	Bộ từ điển được xây dựng một cách thủ công nên khó có thể bao quát hết toàn bộ các lỗi thực tế
17	113	Do đó, nghiên cứu và phát triển một ứng dụng phát hiện và sửa lỗi chính tả tiếng Việt sử dụng thông tin ngữ cảnh sẽ giúp cho việc sửa lỗi chính tả đạt hiệu quả cao hơn	Do đó, nghiên cứu và phát triển một ứng dụng phát hiện và sửa lỗi chính tả tiếng Việt sử dụng thông tin ngữ cảnh, thời gian tính toán nhanh hơn và mô hình có thể xử lý được các từ tiếng Anh không có trong từ điển tiếng Việt sẽ giúp cho việc sửa lỗi chính tả đạt hiệu quả cao hơn
18	117	Bài toán có dữ liệu đầu vào là một câu sai chính tả và đầu ra là một câu đúng chuẩn chính tả nên ta thấy mô hình dạng seq2seq sẽ thích hợp cho bài toán	Bài toán có dữ liệu đầu vào là một câu sai chính tả và đầu ra là một câu đúng chuẩn chính tả nên nhiều người sẽ nghĩ là mô hình dạng seq2seq sẽ thích hợp cho bài toán

Kết quả kiểm trùng với tài liệu: Tài liệu hệ thống - tom_tat_la__8157.txt

Tỉ lệ sao chép: **0.640%**

Trang	Chỉ số	Tài liệu kiểm tra	Tài liệu gốc
19	132	Hình 2-1: Mô hình mạng nơron đơn giản Mạng nơron nhân tạo được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó	Nó được tạo lên từ một số lượng lớn các phần tử (gọi là nơron) kết nối với nhau thông qua các liên kết (gọi là trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó
19	133	Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện	Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện
19	134	Về bản chất “học” chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron	Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron sao cho giá trị hàm lỗi là nhỏ nhất

Kết quả kiểm trùng với tài liệu: Tài liệu hệ thống - vi-do_truong_manh_1.9m.txt

Tỉ lệ sao chép: **0.640%**

Trang	Chỉ số	Tài liệu kiểm tra	Tài liệu gốc
11	29	- Tiếng Việt có 6 thanh: thanh ngang (còn gọi là thanh không), thanh huyền, thanh sắc, thanh hỏi, thanh ngã, thanh nặng	7 các thanh có âm điệu không bằng phẳng hay còn gọi là thanh trắc gồm có: thanh ngã, thanh sắc, thanh hỏi, thanh nặng
27	246	Ví dụ, khi dịch một câu từ tiếng nước này sang tiếng nước khác, chúng ta thường quan tâm nhiều hơn đến ngữ cảnh xung quanh từ hiện tại so với các từ khác trong câu	Ví dụ khi dịch một câu từ tiếng nước A sang nước B, chúng ta thường quan tâm nhiều đến ngữ cảnh xung quanh từ hiện tại hơn là các từ ở vị trí cách xa nó trong câu
27	248	Kỹ thuật attention được đưa ra lần đầu vào năm 2014 bởi Bahdanau và cộng sự trong công trình nghiên cứu về dịch máy	Kỹ thuật attention được đưa ra lần đầu vào năm 2014 bởi Baddanau và cộng sự [16] trong công trình nghiên cứu về dịch máy và đã được phát triển bởi Minh-Thang Luong [18] và các công sự

Kết quả kiểm trùng với tài liệu: Tài liệu hệ thống - tv_phat_trien_cac_cau_truc_thuat_hoc_cua_mang_noron_tu_to_chuc_6964.txt

Tỉ lệ sao chép: **0.213%**

Trang	Chỉ số	Tài liệu kiểm tra	Tài liệu gốc
19	132	Hình 2-1: Mô hình mạng nơron đơn giản Mạng nơron nhân tạo được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó	Dựa theo mạng nơron sinh học, ANN được tạo ra từ một lượng lớn các phần tử xử lý (gọi là nơron1) kết nối với nhau thông qua các liên kết (gọi là trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó