Xây dựng bộ dữ liệu cho bài toán chuẩn hóa tiếng Việt trên mạng xã hội

Nguyễn Thị Hoàng Anh - 20520134, Nguyễn Hà Dung - 20520165, Hồ Thanh Duy Khánh - 20521445, Nguyễn Thị Nguyệt - 20521689

Tóm tắt nội dung—Mục tiêu chính của đồ án là tạo ra bộ dữ liệu cho bài toán chuẩn hóa tiếng Việt trên mạng xã hội. Trong phạm vi đồ án, nhóm trích xuất 3400 bình luận từ bộ dữ liệu ViHSD, thực hiện gán nhãn (chuẩn hóa các token không chuẩn) cho mỗi bình luận. Hệ số Cohen's Kappa thu được cho mức độ đồng thuận là 0.9014, là một mức độ đồng thuận cao. Bên cạnh đó, để kiểm tra chất lượng bộ dữ liệu, nhóm sử dụng mô hình Seq2Seq để học dữ liệu và chuẩn hóa các bình luận tiếng Việt. Kết quả kiểm thử tốt nhất thu được trên mô hình S2SMulti, với độ đo F1 là 58.3% và độ đo BLEU-4 là 0.697. Cuối cùng để đánh giá liệu tác vụ chuẩn hóa văn bản có giúp cải thiện kết quả của các bài toán NLP khác hay không, đồ án thực hiện bài toán HSD (Hate Speech Detection). Kết luận rằng chuẩn hóa văn bản giúp tăng độ chính xác của mô hình Text-CNN từ 77% lên 79%, mô hình GRU từ 75% lên 77%.

Index Terms—xử lý ngôn ngữ tự nhiên, tiếng Việt, mạng xã hội, chuẩn hóa văn bản, deep learning, seq2seq, vihsd, hate speech detection, nlp.

1 Giới thiệu

Bài toán Chuẩn hóa văn bản (Text normalization) là quá trình biến đổi văn bản không chuẩn thành dạng chuẩn hoặc tiêu chuẩn để cải thiện khả năng hiểu và xử lý văn bản bởi các hệ thống máy tính. Bài toán có thể bao gồm nhiều tác vụ như xử lý lỗi chính tả, chuẩn hóa từ viết tắt, xử lý từ ngữ không chính thống, đồng bộ hóa biểu diễn ký tự đặc biệt và thực hiện các biến đổi văn bản khác để đảm bảo tính nhất quán và chuẩn mực của dữ liệu văn bản. Cung cấp một đầu vào văn bản chuẩn hóa cho các hệ thống xử lý ngôn ngữ tự nhiên, nhằm tăng cường khả năng xử lý và hiểu ngữ cảnh, cải thiện hiệu suất và độ chính xác của các tác vụ như dịch máy, phân loại văn bản, tóm tắt văn bản, trích xuất thông tin và hệ thống trả lời tự động.

Tiếng Việt có nhiều từ đồng âm, từ đồng nghĩa và các dạng từ ngữ phong phú. Đặc biệt, trên các trang mạng xã hội, người dùng thường viết nhanh và không tuân thủ các quy tắc ngữ pháp, chính tả, sử dụng ngôn ngữ không chuẩn, viết tắt, kết hợp tiếng Anh và tiếng Việt, và sử dụng biểu tượng cảm xúc đã khiến việc chuẩn hóa văn bản gặp một thách thức lớn. Ví dụ, chúng ta thường gặp lỗi chính tả như viết sai từ "chiện" thay vì "chuyện", hoặc lỗi đánh máy khi gõ "đunga" thay vì "đúng". Ngoài ra, có cả teencode như viết "iu" thay vì "yêu", và viết tắt như "VN" thay vì "Việt Nam". Sự thay đổi về ngữ cảnh từ mạng xã hội dẫn đến việc sử dụng ngôn ngữ viết không chuẩn và không phản ánh chính xác ngữ nghĩa ban đầu. Vì vậy, việc chuẩn hóa văn bản không chuẩn trở thành một bước tiền xử lý quan trọng nhằm tạo ra văn bản dễ đọc, dễ hiểu và thuận tiện cho các tác vụ liên quan đến ngôn ngữ tự nhiên như xử lý ngôn ngữ tự nhiên (NLP) và phân tích dữ liệu trên mạng xã hội.

Trong phạm vi đồ án, nhóm hướng tới 3 mục tiêu:

(1) Mục tiêu chính là xây dựng bộ dữ liệu cho bài toán chuẩn hóa tiếng Việt trên mạng xã hội. Bộ dữ liệu sẽ gồm các câu bình luận ở dạng không chuẩn và dạng chuẩn được gán nhãn tương ứng với nó. Một cặp câu không chuẩn và chuẩn được minh họa trong Hình 2

Hình 1. Minh họa một cặp câu không chuẩn - chuẩn

Câu không chuẩn: mình cần tìm gấp in4 anh ng iu thất lạc ạ !

Câu chuẩn: mình cần tìm gấp thông tin anh người yêu thất lạc ạ !

(2) Chuẩn hóa các câu tiếng Việt từ dạng không chuẩn thành dạng chuẩn. Bài toán được giải quyết sử dụng các mô hình sequence-to-sequence trên bộ dữ liệu vừa xây dựng,

(3) Giả định rằng chuẩn hóa văn bản giúp cải thiện hiệu suất các tác vụ NLP khác hơn so với các bước tiền xử lý đơn giản. Đồ án kiểm định giả thiết này bằng cách thí nghiệm tác vụ HSD (Hate Speech Detection) trên văn bản chưa được chuẩn hóa và đã được chuẩn hóa ở mục tiêu (2).

2 Nghiên cứu liên quan

Các phương pháp chuẩn hóa văn bản trên mạng xã hội đã trải qua sự đa dạng và phát triển theo thời gian. Trước đây, các phương pháp truyền thống thường sử dụng các quy tắc và luật ngữ cảnh để thực hiện chuẩn hóa văn bản. Một số phương pháp truyền thống bao gồm: Sử dụng từ điển, quy tắc ngữ pháp, thay thế dựa trên luật,.... Tuy nhiên, với sự tiến bộ của học máy và mô hình ngôn ngữ dựa trên dữ liệu, các phương pháp dựa trên máy học đã trở thành xu hướng phổ biến hơn, bao gồm học có giám sát, học không giám sát và học bán giám sát. Deep Learning, với khả năng mạnh mẽ trong việc học từ dữ liệu và trích xuất đặc trưng tự động, đã trở thành một phương pháp quan trọng và hiệu quả trong chuẩn hóa văn bản trên mạng xã hội. Các nghiên cứu mới nhất đã tận dụng sức mạnh của Deep Learning để tạo ra các mô hình mạnh mẽ và linh hoạt, có khả năng hiểu và xử lý các biến thể ngôn ngữ trong mạng xã hội.

Các nghiên cứu trên các ngôn ngữ khác nhau cung cấp cái nhìn đa dạng về việc chuẩn hóa văn bản. Trong nghiên

cứu về tiếng Ba Lan "Expanding Abbreviations in a Strongly Inflected Language: Are Morphosyntactic Tags Sufficient?" (Zelasko, 2018) [1], nhóm nghiên cứu tập trung vào việc khôi phục thông tin hình thái từ các dạng viết tắt trong ngôn ngữ có sự biến đổi mạnh mẽ. Kết quả của nghiên cứu này sử dụng mạng nơ-ron LSTM hai chiều kết hợp với nhúng nhãn ngữ pháp cho thấy khả năng suy ra thông tin hình thái từ nhãn ngữ pháp trong ngữ cảnh có thể cải thiện quá trình chuẩn hóa.

Trong nghiên cứu về tiếng Anh "Seq2Seq Deep Learning Models for Microtext Normalization" (Satapathy, R., Li, Y., Cavallari, S., Cambria, E., 2019) [2], các mô hình học sâu như LSTM, attentive LSTM, CNN-LSTM và GRU đã được so sánh để xử lý vấn đề chuẩn hóa văn bản ngắn trên các nền tảng mạng xã hội. Kết quả cho thấy các mô hình này cải thiện đáng kể độ chính xác của tác vụ phân loại cảm xúc.

Nghiên cứu về tiếng Việt "Non-Standard Vietnamese Word Detection and Normalization for Text-to-Speech" (Dang, H.-T., Vuong, T.-H.-Y., Phan, X.-H., 2022) [3] giới thiệu một phương pháp hai giai đoạn để xử lý từ không chuẩn trong văn bản tiếng Việt. Trong giai đoạn đầu tiên, các mô hình dựa trên thẻ nhãn như CRFs, BiLSTM-CNN-CRF và BERT-BiGRU-CRF được sử dụng để phát hiện từ không chuẩn. Sau đó, dựa vào loại từ, chúng được mở rộng thành dạng nói. Kết quả cho thấy phương pháp đề xuất đạt được độ chính xác cao.

Tổng kết lại, các nghiên cứu trên các ngôn ngữ khác nhau cho thấy sự đa dạng trong việc chuẩn hóa văn bản. Các phương pháp học sâu và mô hình sequence-to-sequence như LSTM, attentive LSTM, CNN-LSTM, GRU và các mô hình dựa trên thẻ nhãn đã chứng tỏ khả năng xử lý ngữ cảnh và cải thiện độ chính xác của việc chuẩn hóa văn bản trên mạng xã hội. Các phương pháp này có thể được áp dụng để giải quyết bài toán chuẩn hóa văn bản tiếng Việt. Dựa theo bài báo "Adapting Sequence to Sequence models for Text Normalization in Social Media" [4], chúng tôi đề xuất sử dụng mô hình mạng neural sequence-to-sequence (Seq2Seq) để giải quyết vấn đề chuẩn hóa văn bản trên mạng xã hội tiếng Việt. Việc chọn các mô hình sequence-to-sequence vì nó có khả năng xử lý các chuỗi đầu vào và đầu ra có độ dài khác nhau. Điều này rất hữu ích khi chúng ta đang làm việc với các biến thể văn bản không chính thức. Mô hình cũng giúp chúng ta tập trung vào ngữ cảnh để hiểu và xử lý văn bản một cách tốt nhất. Bên cạnh đó, việc kết hợp cả từ và ký tự trong mô hình giúp chúng ta xử lý được cả từ ngắn và từ dài một cách hiệu quả. Hơn nữa, giúp giảm thiểu hiện tượng OOV (Out-of-Vocabulary).

3 Xây dựng bộ dữ liệu

3.1 Thu Thập dữ liệu

Văn bản mạng xã hội được nhóm trích xuất từ bộ dữ liệu ViHSD. Bộ dữ liệu ViHSD được dùng để phát hiện các bình luận tiêu cực trên mạng xã hội ở Việt Nam, được công bố trong bài báo 'A Large-scale Dataset for Hate Speech Detection on Vietnamese Social Media Texts' [5] năm 2021. Bộ dữ liệu này bao gồm 33400 câu bình luận, được gán các nhãn bao gồm CLEAN (không tiêu cực), OFFENSIVE (mang tính công kích, phản cảm) và HATE (mang tính thù địch). Nhận thấy dữ liệu của bộ ViHSD khá phù hợp với yêu cầu của bài toán, các văn bản có nguồn gốc từ mạng xã hội và có chứa những từ không chuẩn (viết tắt, teencode,. . .). Nên trong phạm vi đồ án môn học, nhóm chỉ trích xuất 3400 câu bình luận đầu tiên trong tập huấn luyện của bộ ViHSD để phục vụ cho bài toán chuẩn hóa văn bản mạng xã hội.

3.2 Tiền xử lý dữ liệu

Dữ liệu cần được tiền xử lý để chuẩn bị cho việc gán nhãn. Nhóm thực hiện tách các token trong câu bằng khoảng trắng. Ví dụ như chuỗi 'Hôm nay tôi đi chơi' sẽ được biến đổi thành danh sách các token được tách bởi khoảng trắng ['Hôm', 'nay', 'tôi', 'đi', 'chơi'].

Các dấu câu (dấu chấm – '.', dấu phẩy – ',', dấu chấm hỏi – '?', dấu chấm than – '!',. . .), dấu ngoặc (ngoặc tròn – (), ngoặc vuông – [], ngoặc nhọn – {},. . .), emoji, emoticon và một số kí tự đặc biệt khác được coi là một token riêng biệt và được tách rời ra. Lấy minh họa chuỗi. Tuy nhiên trong một số trường hợp, chuỗi kí tự có chứa các kí hiệu đặc biệt được giữ nguyên: biểu tượng cảm xúc dạng văn bản – emoticon (=)), :D, :v,. . .), đường liên kết – link, địa chỉ email, hashtags (#), nhắc – mention (@), thời gian, ngày tháng, tiền tệ.

3.3 Gán nhãn

Toàn bộ quy trình gán nhãn và phương pháp đánh giá độ đồng thuận trình bày sau đây được tham khảo từ 2 nghiên cứu 'Shared Tasks of the 2015 Workshop on Noisy User-generated Text: Twitter Lexical Normalization and Named Entity Recognition' [6] và 'MultiLexNorm: A Shared Task on Multilingual Lexical Normalization' [7]

Có tổng cộng 5 người gán nhãn cho dataset gồm 3400 bình luận. Những người tham gia gán nhãn đều đạt trình độ tốt nghiệp 12 năm phổ thông trung học, sử dụng thành thạo tiếng Việt và quen thuộc với dữ liệu trên mạng xã hội. Mỗi người sẽ gán nhãn 1360 bình luận và mỗi bình luận sẽ được gán nhãn bởi 2 người. Việc gán nhãn được thực hiện bằng công cụ Google Sheet, mỗi người gán nhãn trên một file Google Sheet riêng để đảm bảo tính độc lập trong việc gán nhãn.

Một bình luận trong bộ dữ liệu thường khá ngắn, chứa ít thông tin về ngữ cảnh của vấn đề được nhắc tới. Do đó, việc chuẩn hóa văn bản có thể rất khác nhau đối với mỗi người gán nhãn, nó tùy thuộc vào trình độ tiếng Việt, thói quen sử dụng tiếng Việt và kiến thức xã hội của mỗi người. Để đảm bảo văn bản được chuẩn hóa có độ tương đồng cao nhất có thể, nhóm đặt ra một số quy tắc chung cho việc gán nhãn. Việc gán nhãn được thực hiện trên 3 cột, cột đầu tiên là phiên bản văn bản chuẩn hóa của câu gốc, cột thứ hai thể hiện rằng liệu một token có cần chuẩn hóa hay không, cột cuối cùng thể hiện độ chắc chắn về cách chuẩn hóa của người gán nhãn. Giao diện gán nhãn được minh họa trong Hình 1. Việc gán nhãn cần theo các quy tắc chính sau:

(1) Ở cột đầu tiên, token được giữ nguyên nếu không cần chuẩn hóa, và được sửa lại nếu cần chuẩn hóa. Lưu ý rằng, một số token được coi là không chuẩn nhưng không cần phải được chuẩn hóa, ví dụ như đường link, địa chỉ gmail, icon, emoticon, các token viết bằng ngôn ngữ khác, hay dạng viết tắt thông dụng của các danh từ riêng

(2) Ở cột thứ hai, gán nhãn '#' nếu token cần được chuẩn hóa, nếu không thì để trống.

Hình 2. Giao diện gán nhãn

	index	token			
1446	Index: 110				
1447	0	cận	cẩn	#	
1448	1	thận	thận		
1449	2	nhà	nha	#	
1450	3	ae	anh em	#	?

(3) Ở cột cuối cùng, gán nhãn '?' nếu người gán nhãn không chắc chắn về cách chuẩn hóa, nếu chắc chắn thì bỏ trống.

(4) Văn bản đều được viết với kí tự in thường, kể cả với các danh từ riêng.

(5) Nếu có nhiều cách chuẩn hóa, liệt kê chúng trong cùng một cell và cách nhau bởi dấu phẩy.

(6) Các chữ số mang ý nghĩa số lượng thì giữ nguyên (1 người, 3 con gà), chữ số không mang ý nghĩa số lượng thì chuyển nó thành chữ ('1 số' - 'một số', 'chị 3' - 'chị ba').

(7) Những từ cảm thán như 'hahaaa' và 'huhuu' có thể bỏ qua.

(8) Trong trường hợp token là các từ tiếng nước ngoài nhưng được viết dưới dạng phiên âm tiếng Việt, ví dụ như 'biu ty phun' (beautiful) thì chuẩn hóa bằng cách thêm dấu gạch ngang '-' vào giữa các âm tiết ('biu ty phun' - 'biu-ty-phun' hoặc sửa lại thành từ tiếng Việt nếu có từ tương ứng.

(9) Nếu một token bị tách ra thành nhiều ô, người gán nhãn nên ghi từ chuẩn hóa ở ô đầu tiên và để các ô còn lại trống.

(10) Đặt các dấu thanh đúng vị trí. Ví dụ: 'hoá' -> 'hóa', 'nguỵ' -> 'ngụy'

Quá trình gán nhãn như trên được hoàn thành trong khoảng 2 tuần. Để đánh giá độ đồng thuận, nhóm sử dụng 2 độ đo. **(1)** Hệ số Cohen's Kappa được sử dụng để tính độ đồng thuận trong việc có cần chuẩn hóa token đó hay không. **(2)** Độ đo phần trăm đánh giá tỉ lệ số lần các người gán nhãn chuẩn hóa văn bản cùng một cách giống nhau. Hệ số Cohen's Kappa có giá trị 0.9014, điều này chỉ ra rằng các nhân viên gán nhãn có mức độ đồng thuận cao trong việc chuẩn hóa văn bản, và đánh giá của những người gán nhãn có sự tương đồng lớn với nhau. Mức độ đồng nhất trong việc chuẩn hóa từ giữa hai nhân viên gán nhãn đạt giá trị 0.7449, cho thấy mức độ đồng nhất tương đối cao giữa hai nhân viên, tức là các từ đã được chuẩn hóa có sự tương đồng lớn giữa hai nhân viên gán nhãn. Kết quả này tương thích với kết quả của hệ số Cohen's Kappa, thể hiện được tính chất chung và đồng đều của quá trình chuẩn hóa từ. Từ đó, ta có thể kết luận rằng quá trình gán nhãn và chuẩn hóa văn bản đã được thực hiện một cách đồng nhất và chính xác. Kết quả hệ số cohen's Kappa và tỷ lệ đồng ý cho thấy tính chân thực và độ tin cậy của quá trình gán nhãn.

Đối với những token không đạt được sự đồng thuận trong việc gán nhãn, nếu một trong hai người gán nhãn chắc chắn về sự chỉnh sửa đó (không gán '?') và người còn lại không chắc về cách chuẩn hóa của mình (gán '?'), thì kết quả chuẩn hóa sẽ theo người chắc chắn về cách chuẩn hóa. Các trường hợp còn lại sẽ đưa cho một người khác không tham gia quá trình gán nhãn trước đó thực hiện gán nhãn lại. Cuối cùng, các câu không chứa bất cứ từ nào cần được chuẩn hóa sẽ bị xóa đi. Đối với những trường hợp có nhiều cách chuẩn hóa, các cặp câu bình luận nguyên bản và câu bình luận được chuẩn hóa sẽ được nhân bản, với mỗi bản sao là một trường hợp chuẩn hóa. Sau khi xóa và nhân bản các câu bình luận, số lượng câu trong bộ dữ liệu từ 3400 xuống còn 2181.

3.4 Phân tích lỗi gán nhãn

Lỗi do sai sót trong đánh nhãn của nhân viên: Nhân viên mắc phải những lỗi như gán nhãn không chính xác cho dữ liệu (với token ban đầu là "ngủ" nhưng lại bị gán thành "an", hoặc với token ban đầu là "908634738" nhưng lại bị gán thành "908634739"), hoặc thực hiện gán nhãn chuẩn hóa nhưng lại không đánh # vào cột thứ 2. Quy tắc (6) yêu cầu các chữ số mang ý nghĩa số lượng thì giữ nguyên, chữ số không mang ý nghĩa số lượng thì chuyển nó thành chữ. Có nhiều trường hợp người gán nhãn không chuyển thành chữ.

Lỗi do sự hiểu biết về ngữ cảnh: Có những câu viết không dấu và ngắn không thể hiện được ý nghĩa của câu, hoặc có thể hiện nhưng có nhiều cách hiểu khác nhau, cùng với đó là văn hóa vùng miền như "chèn", "hửi", "ứ", "méo",... gây khó khăn trong việc xác định và gán nhãn chuẩn hóa. Đôi khi, guideline gây ra sự nhập nhằng cho người gán nhãn. Ví dụ, quy tắc (9) yêu cầu nếu một chữ bị tách ra thành nhiều ô, người gán nhãn nên ghi từ chuẩn hóa ở ô đầu tiên và để các ô còn lại trống. Tuy nhiên, đôi khi việc quyết định từ chuẩn hóa đúng để ghi ở ô đầu tiên có thể gặp khó khăn. Điều này dẫn đến sự khác nhau trong cách gán nhãn của mỗi người và làm tăng sự không nhất quán trong dữ liệu gán nhãn.

Lỗi khác: Các đơn vị không được đề cập trong guideline như "20k" người gán nhãn có thể chọn các cách chuẩn hóa khác nhau "20000", "20 nghìn", "20 ngàn", dẫn đến sự không nhất quán trong việc gán nhãn và có thể ảnh hưởng đến sự đồng nhất của dữ liệu.

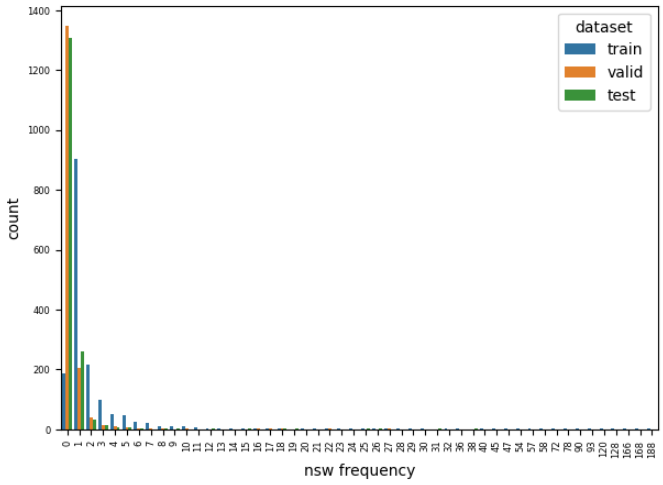
3.5 Mô tả dữ liệu

Bộ dữ liệu gồm 2181 cặp câu bình luận nguyên bản và đã được chuẩn hóa, chia ngẫu nhiên thành tập train, validation và test với tỉ lệ 8:1:1. Bảng 1 thể hiện số lượng toàn bộ các token và token không chuẩn cần được chuẩn hóa trong tập dữ liệu. Nhìn chung, số token cần chuẩn hóa chiếm khoảng 17% tổng số token của tập huấn luyện và tập kiểm thử. Số token duy nhất cần chuẩn hóa lần lượt chiếm khoảng 39% trong tập train và 25% trong tập validation và tập test.

Bảng 1
Thống kê dữ liệu trong bộ dữ liệu

	Train	Valid	Test	Full Data
Số token	26990	3429	3569	33988
Số token duy nhất	3704	1208	1259	4103
Số token cần chuẩn hóa	4901	589	641	6131
Số token duy nhất cần chuẩn hóa	1452	290	329	1638

Hình 3. Phân bố tần suất các token cần chuẩn hóa trong tập dữ liệu



Các token cần chuẩn hóa xuất hiện nhiều nhất trong tập dữ liệu được thể hiện trong Bảng 2. Từ 'ko' (không) xuất hiện nhiều nhất, 235 lần, chiếm khoảng 3.8% trong tập train, 2.7% trong tập validation và 4.8% tập test. Tuy nhiên, phần lớn token cần chuẩn hóa xuất hiện rất ít trong tập dữ liệu, chỉ từ 1 đến 2 lần, như quan sát được trong Hình 3.

Bảng 2
Các token cần được chuẩn hóa xuất hiện nhiều nhất trong tập dữ liệu

Xếp hạng	Train		Valid		Test		Full Data	
1	ko	188	t	27	t	38	ko	235
2	dm	168	k	22	ko	31	t	231
3	t	166	dm	18	đc	26	dm	205
4	đc	128	m	18	m	25	đc	171
5	k	120	đc	17	dm	19	k	160
6	vl	93	ko	16	k	18	m	133
7	m	90	e	10	a	15	vl	112
8	a	78	vl	10	vn	12	a	101
9	e	72	đm	10	e	9	e	91
10	vn	58	vn	8	vl	9	vn	78

4 Chuẩn hóa văn bản

4.1 Phương pháp chuẩn hóa

Đồ án sẽ huấn luyện mô hình Seq2Seq ở mức token sử dụng cơ chế attention (S2S), sau đó so sánh kết quả huấn luyện với 2 biến thể của nó: S2SSelf và S2SMulti. Mô hình được sử dụng sẽ hoạt động ở mức độ token, tức là mô hình sẽ xem các đơn âm tiết là đơn vị nhỏ nhất trong các chuỗi input, target và output.

4.1.1 Mô hình S2S

Đồ án sẽ sử dụng mô hình sequence-to-sequence để biến đổi chuỗi đầu vào (câu không chuẩn) thành chuỗi đầu ra (câu được chuẩn hóa). Một câu bình luận không chuẩn được biểu diễn dưới dạng một chuỗi đầu vào độ dài T gồm các âm tiết $\vec{x} = [x_1, x_2, ..., x_T]$, mục tiêu là sẽ tạo ra một chuỗi đầu ra $\vec{y} = [y_1, y_2, ..., y_T]$ cũng có độ dài T và mang ý nghĩa tương ứng với chuỗi $\vec{x}$. Bài toán chuẩn hóa tiếng Việt được định nghĩa dưới dạng sequence-to-sequence nhằm ánh xạ một chuỗi này sang một chuỗi khác. Đồ án xây dựng mô hình dựa trên framework encoder-decoder (Cho và cộng sự, 2014; Sutskever, Vinyals, và Le 2014) [8], [9], cả 2 nghiên cứu trên đều sử dụng mô hình RNN được cài đặt cơ chế attention.

Có nhiều biến thể của mạng RNN, trong đó kể đến kiến trúc LSTM, GRU,... Trong phạm vi đồ án môn học, nhóm chọn mô hình Bi-LSTM được cài đặt cơ chế attention để chuẩn hóa văn bản.

BiLSTM: LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào. Việc hiểu nội dung token không chuẩn để có thể trả nó về dạng chuẩn không chỉ phụ thuộc vào các thông tin phía trước token đang xét và cả các thông tin phía sau. Tuy nhiên, một kiến trúc LSTM truyền thống chỉ có thể trích xuất thông tin của các token nằm phía trước nó. BiLSTM có thể khắc phục điểm yếu trên. Một kiến trúc BiLSTM thường chứa 2 mạng LSTM đơn được sử dụng đồng thời và độc lập để mô hình hoá chuỗi đầu vào theo 2 hướng: từ trái sang phải (forward LSTM) và từ phải sang trái (backward LSTM)

Attention là một kỹ thuật được (Bahdanau và cộng sự, 2014) [10] và (Luong và cộng sự, 2015) [11] giới thiệu trong các bài báo của họ. Ý tưởng của nó là cho phép decoder sử dụng thông tin của toàn bộ chuỗi đầu vào, nhưng chỉ tập trung vào những phần quan trọng tại mỗi timeStep.

Encoder đọc chuỗi đầu vào $\vec{x}$ và chuyển nó thành các chuỗi hidden state tương ứng với ngữ cảnh cụ thể $\vec{h} = [h_1, h_2, ..., h_T]$. Trong mô hình bidirectional, hai encoder được sử dụng, một encoder đọc văn bản ở chế độ forward và một encoder đọc văn bản theo chiều ngược lại. Hidden state cuối cùng tại thời điểm t là sự kết hợp của hai mô đun encoder $\vec{h_t} = [g_f(x_t, h_{t-1}); g_b(x_t, h_{t+1})]$, trong đó g_f và g_b lần lượt là forward và backward encoder units. Tương tự, decoder định nghĩa một chuỗi các hidden states $\vec{s_j} = g_s(s_{j-1}, y_{j-1}, c_j)$ dựa trên từ trước đó y_{j-1}, decoder state s_{j-1} và context vector c_j, được tính bằng tổng trọng số của các hidden states dựa trên cơ chế attention (Bahdanau, Cho, và Bengio 2014) [10]:

$$c_j = \sum_{i=k}^{|t|} \alpha_{jk} h_k \tag{1}$$

trong đó $\alpha_{jk} = Softmax(f(s_{j-1}, h_k))$ và $f(s_{j-1}, h_k) = s_{j-1}^T W h_k$ là hàm tổng quát theo (Luong, Pham, và Manning 2015) [10]. Sau đó, mỗi âm tiết đầu ra sẽ được dự đoán bởi bộ phân loại Softmax $y_j \sim p(y_j \mid y_{<j}, \vec{x}) = Softmax(\psi(s_j))$, trong đó ψ là một hàm biến đổi affine ánh xạ decoder state thành một vector có kích thước bằng tập từ vựng (vocabulary-sized vector).

Cho tập dữ liệu huấn luyện D, mô hình Seq2Seq được huấn luyện bằng cách tối đa hóa log-likelihood:

$$L(\theta) = -\sum_{(\vec{x}, \vec{y}) \in D} \sum_{j=1}^{|L|} log p_\theta(y_j \mid y_{<j}, \vec{x}) \tag{2}$$

Lưu ý rằng trong quá trình huấn luyện, một dự đoán sai có thể gây ra lỗi tích lũy dần trong các bước huấn

Hình 4. Minh họa các chuỗi đầu vào, chuỗi đầu ra mục tiêu và chuỗi đầu ra dự đoán

Input: [‘mình’, ‘cần’, ‘tìm’, ‘gấp’, ‘in4’, ‘anh’, ‘ng’, ‘iu’, ‘thất’, ‘lạc’, ‘ạ’, ‘!’]

Target: [‘mình’, ‘cần’, ‘tìm’, ‘gấp’, ‘thông tin’, ‘anh’, ‘người’, ‘yêu’, ‘thất’, ‘lạc’, ‘ạ’, ‘!’]

Output: [‘mình’, ‘cần’, ‘tìm’, ‘gấp’, ‘in4’, ‘anh’, ‘người’, ‘yêu’, ‘thất’, ‘lạc’, ‘ạ’, ‘!’]

luyện tiếp theo. Vì vậy, khi tính toán xác suất có điều kiện $p_\theta\left(y_j \mid y_{<j}, \vec{x}\right)$, người ta thường sử dụng phương pháp Scheduled Sampling (Bengio và cộng sự, 2015) [11]. Đây là một phương pháp luân phiên giữa việc sử dụng kết quả dự đoán của mô hình ở bước trước $\hat{y}_{j-1}$ và đầu ra tương ứng của nó y_{j-1} để giảm bớt các lỗi tích lũy từ trước.

Mô hình Seq2Seq ở mức token có thể nắm bắt nội dung ngữ nghĩa ở mức độ token và các phụ thuộc ngữ cảnh dài hạn giúp ánh xạ đúng các từ có nhiều cách chuẩn hóa. Hình 4 minh họa các chuỗi đầu vào (input), chuỗi đầu ra mục tiêu (target) và chuỗi đầu ra dự đoán (output) được chuẩn hóa một cách phù hợp dựa vào ngữ cảnh.

4.1.2 Mô hình S2SSelf

Một ký hiệu đặc biệt ‘@self’ sẽ được sử dụng để đánh dấu token trong chuỗi đầu vào mà không cần thay đổi trong quá trình huấn luyện. Ví dụ, nếu một chuỗi đầu vào là ‘Nước là chất điện phân’ và chuỗi đầu ra là ‘Nc là chất điện phân’, thì chuỗi đầu ra sẽ được biến đổi thành ‘Nc @self @self @self @self’. Khi thực hiện dự đoán, ta sẽ thay thế kí hiệu đặc biệt này bằng từ tương ứng trong câu gốc.

4.1.3 Mô hình S2SMulti

Đầu tiên, ta thực hiện chuyển đổi các token chỉ có một cách chuẩn hóa duy nhất, sau đó sẽ sử dụng mô hình S2S để dịch các token có nhiều cách sửa trong câu.

4.2 Độ đo đánh giá

Nhóm sử dụng 2 độ đo riêng biệt để đánh giá khả năng dự đoán của mô hình sequence-to-sequence.

F1 score: Độ đo F1 là một độ đo phổ biến để đánh giá bài toán phân loại. Trong phạm vi đồ án này, độ đo F1 được dùng để đánh giá tỷ lệ gán nhãn đúng đối với các token cần được chuẩn hóa. Độ đo F1 được tính như sau

$$p = \text{correct norm}/\text{total norm} \tag{3}$$

$$r = \text{correct norm}/\text{total nsw} \tag{4}$$

$$f1 = \frac{2 \times p \times r}{p + r} \tag{5}$$

trong đó, correct norm là số token được chuẩn hóa đúng, total norm số token được chuẩn hóa trong câu dự đoán, tức là có thay đổi so với câu gốc và total nsw là tổng số token cần được chuẩn hóa trong chuỗi đầu vào.

Độ đo F1 được tính trên từng cặp câu, cuối cùng là lấy giá trị trung bình. Đối với nhứng câu có nhiều cách chuẩn hóa, tức là có nhiều cặp câu với chuỗi đầu vào giống nhau và khác chuỗi target, ta lấy giá trị F1 cao nhất.

BLEU-4: BLEU score là một trong những độ đo phổ biến nhất khi có một hoặc nhiều người dịch câu theo nhiều cách khác nhau. Tập dữ liệu sử dụng trong đồ án cũng có đặc điểm tương tự nên nhóm sử dụng BLEU-4 làm một trong hai độ đo đánh giá mô hình. Để tính được BLEU score, ta lần lượt tính precision score cho 1-gram đến 4-gram. N-gram là tần suất xuất hiện của n kí tự (hoặc từ) liên tiếp nhau có trong câu. Công thức tính precision score cho N-gram được thể hiện như sau

$$p_n = \frac{\text{Số lượng n-grams được chuẩn hóa đúng}}{\text{Tổng số n-grams trong câu}} \tag{6}$$

Tiếp theo, chúng ta kết hợp các Precision Score sử dụng công thức dưới đây. Cái này có thể tính toán cho các N khác nhau và các giá trị weight khác nhau. Cụ thể, ta cho $N = 4$ và uniform weights $\omega_n = N/4$

$$\begin{aligned}\text{Geometric Average Precision (N)} &= \prod_{n=1}^{N} (p_n)^{\omega_n} \\ &= \prod_{n=1}^{N} (p_1)^{\frac{1}{4}} \times \prod_{n=1}^{N} (p_2)^{\frac{1}{4}} \times \prod_{n=1}^{N} (p_3)^{\frac{1}{4}} \times \prod_{n=1}^{N} (p_4)^{\frac{1}{4}}\end{aligned} \tag{7}$$

Tuy nhiên, nếu chuỗi dự đoán chứa các từ đơn, 1-gram precision sẽ là 1/1 = 1. Điều này sẽ khuyến khích mô hình sinh đầu ra ngắn hơn và điểm cao hơn. Do đó, ta tính Brevity Penalty để penalize các câu trả lời quá ngắn. Nếu mô hình dự đoán càng ít từ so với câu đúng, giá trị này sẽ càng nhỏ

$$\text{BP} = \begin{cases} 1 & c > r \\ e^{1-r/c} & c \leq r \end{cases} \tag{8}$$

trong đó, c là số lượng token có trong câu dự đoán, r là số lượng token có trong câu mục tiêu.

Cuối cùng, để tính Bleu Score, ta nhân Brevity Penalty với Geometric Average of Precision Scores.

$$\text{BLEU (N)} = \text{BP} \times \text{Geometric Average Score (N)} \tag{9}$$

4.3 Kết quả thực nghiệm

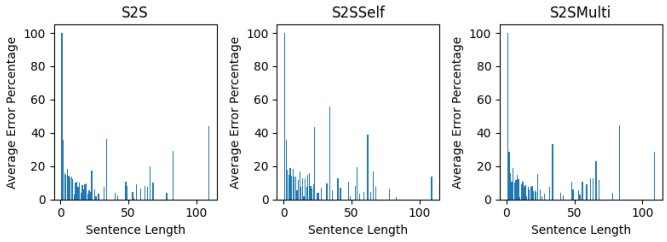
Dựa vào Bảng 3, ta có thể thấy kết quả chuẩn hóa chuẩn hóa văn bản không quá tốt, do tất cả độ đo đều thấp hơn 70%. Mô hình S2SMulti có khả năng tốt nhất trong việc chuẩn hóa văn bản, cả trong việc xử lý ngữ cảnh và tổng quát hóa dữ liệu mới. Điều này được chứng minh độ đo F1 và BLEU-4 thu được trên tập validation và tập test đều cao nhất. Độ đo F1 thu được trên tập validation và test lần lượt là 55.63% và 58.3%, độ đo BLEU-4 thu được trên tập validation và test lần lượt là 0.6526 và 0.6971. Kết quả thu được trên mô hình S2S gần xấp xỉ với mô hình S2SMulti. Tuy nhiên, mô hình S2SSelf thu được kết quả thấp nhất trên cả 2 độ đo, khoảng 40% đối với độ đo F1 và chưa tới 60% đối với độ đo BLEU-4.

Để biến đổi đúng token về dạng chuẩn, mô hình phải học được nội dung trong bối cảnh của câu. Việc thế kí hiệu self vào các token không cần bị biến đổi trong quá trình học khiến chuỗi đầu vào mất đi nhiều thông tin quan trọng. Do đó, mô hình S2SSelf không thể tận dụng ngữ cảnh vào việc chuẩn hóa câu. Mô hình S2SMulti đảm bảo các token có duy nhất một cách diễn giải được biến đổi chính xác trong quá trình học. Đồng thời, cơ chế này cũng giúp tăng độ tin cậy

Bảng 3
Kết quả thực nghiệm chuẩn hóa văn bản

	S2S		S2SSelf		S2SMulti	
	F1	BLEU-4	F1	BLEU-4	F1	BLEU-4
Valid	0.5426	0.6504	0.4088	0.5756	0.5563	0.6526
Test	0.5616	0.6871	0.3765	0.5840	0.5830	0.6971

Hình 5. Biểu đồ tác động của độ dài câu đến tỷ lệ lỗi của các mô hình



của ngữ cảnh câu trong quá trình học. Trong khi đó, mô hình S2S truyền thống không đảm bảo dịch đúng những token chỉ có một cách diễn giải. Do vậy trong một số trường hợp thì mô hình S2SMulti lại chuẩn hóa câu tốt hơn.

4.4 Phân tích lỗi chuẩn hóa văn bản

Quan sát từ Hình 5, ta nhận thấy rằng nếu câu quá dài hoặc quá ngắn thì tỉ lệ lỗi của mô hình càng cao. Trong trường hợp này, độ dài lý tưởng của các chuỗi đưa vào mô hình nằm trong khoảng từ 25 đến 50. Nhận thấy rằng độ dài câu có tác động đáng kể đến tỷ lệ lỗi của các mô hình, nên việc xử lý độ dài câu có thể là một khía cạnh quan trọng để cải thiện hiệu suất và giảm tỷ lệ lỗi của các mô hình này.

Một số lỗi gán nhãn có thể ảnh hưởng đến kết quả dự đoán của mô hình. Trường hợp xảy ra thường xuyên nhất là sự không đồng bộ gán nhãn đối với các câu có nhiều cách chuẩn hóa. Ví dụ rằng token 't' có thể hiểu là 'tôi', 'tao', hoặc 'tớ'. Nhưng trong khi huấn luyện, với câu 't thích nghe kpop', kết quả gán nhãn cho ra 3 trường hợp: 'tao thích nghe kpop', 'tôi thích nghe kpop', 'tớ thích nghe kpop'. Khi dự đoán câu 'chỉ t vs' chỉ có 1 trường hợp chuẩn hóa là 'chỉ tao với', kết quả của mô hình lại là 'chỉ tôi với'. Câu này vẫn đúng nhưng nếu so với tập các trường hợp có thể đúng thì không khớp, và độ đo thu được trong trường hợp này là 0.

5 Hate Speech Detection

5.1 Thiết kế thực nghiệm

Thí nghiệm này nhằm đánh giá tầm quan trọng và mức độ cần thiết của việc chuẩn hóa văn bản trong giai đoạn tiền xử lý dữ liệu cho các tác vụ NLP phức tạp khác, cụ thể là Hate Speech Detection.

Bài toán Hate Speech Detection nhằm phát hiện ra các bình luận thù ghét và tiêu cực trên mạng xã hội. Các bình luận được đưa vào mô hình học và đầu ra là các nhãn thể hiện tính chất của câu bình luận đó: CLEAN, OFFENSIVE và HATE. Nhóm sẽ sử dụng mô hình Text-CNN và GRU để dự đoán nhãn và sử dụng độ đo F1-micro, F1-macro và Accuracy để đánh giá hiệu suất của mô hình. Nhóm cài đặt

Bảng 4
Kết quả thực nghiệm Hate Speech Detection (HSD)

	Dữ liệu chưa chuẩn hóa		Dữ liệu được chuẩn hóa	
	Text-CNN	**GRU**	**Text-CNN**	**GRU**
F1-micro	77.27%	75%	79.55%	77.27%
F1-macro	35.61%	28.57%	41.45%	35.61%
Accuracy	77.27%	75%	79.55%	77.27%

các thông số của mô hình giống hệt như nghiên cứu của thầy Lưu Thanh Sơn [5].

Thí nghiệm sẽ được thực hiện lần lượt trên các câu bình luận chưa được chuẩn hóa và đã được chuẩn hóa. Dữ liệu được chuẩn hóa lấy từ kết quả của mô hình S2SMulti, là kết quả tốt nhất. Có tổng cộng 438 câu bình luận được chuẩn hóa được gộp từ tập validation (219 câu) và tập test (219) câu. Tập dữ liệu chưa được chuẩn hóa được lấy tương ứng với 438 câu bình luận đã chuẩn hóa trên. Các nhãn tương ứng với mỗi bình luận được trích xuất từ bộ dữ liệu ViHSD. Sau cùng, tỉ lệ tập train, validation và test của cả 2 thí nghiệm được chia theo tỉ lệ 8:1:1.

5.2 Kết quả thực nghiệm

Quan sát Bảng 4, ta thấy rằng chuẩn hóa văn bản giúp tăng độ chính xác của tác vụ HSD lên 2% trên độ đo F1-micro và Accuracy, khoảng 6 - 7% trên độ đo F1-macro đối với cả 2 mô hình Text-CNN và GRU. Kết luận rằng so với các bước tiền xử lý đơn giản thông thường, thì trong trường hợp này chuẩn hóa văn bản giúp tăng hiệu suất của bài toán HSD. Tuy rằng độ chính xác của việc chuẩn hóa văn bản không quá cao (dưới 70%), nhưng thí nghiệm này cho thấy tiềm năng của việc chuẩn hóa văn bản trong việc cải thiện hiệu suất của các tác vụ NLP phức tạp hơn.

6 Kết luận

Nhìn chung, đồ án này đã hoàn thành ba nhiệm vụ chính. Đầu tiên, nhóm đã xây dựng được bộ dữ liệu dùng cho bài toán chuẩn hóa tiếng Việt sử dụng trên mạng xã hội. Thao tác gán nhãn cho các token cần được chuẩn hóa có mức độ đồng thuận khá cao, đảm bảo chất lượng dữ liệu đầu vào cho các mô hình học sâu. Thứ hai, nhóm áp dụng mô hình sequence-to-sequence để chuẩn hóa các token không chuẩn. Kết quả cho thấy các mô hình chưa thực sự có hiệu quả cao trong việc chuẩn hóa văn bản khi các độ đo đều thấp hơn 70%. Cuối cùng, tuy rằng mô hình sequence-to-sequence chuẩn hóa văn bản không quá chính xác, nhưng trong thí nghiệm của nhóm thực hiện, nó đủ để cải thiện hiệu suất của tác vụ Hate Speech Detection.

Tuy nhiên, trong quá trình thực hiện đồ án, nhóm còn gặp nhiều khó khăn và do đó còn tồn tại nhiều hạn chế. Về vấn đề dữ liệu, bộ dữ liệu ViHSD được thu thập vào khoảng thời gian năm 2019-2020 nên dữ liệu đa số là các nội dung phổ biến những năm trước, không còn phù hợp với mạng xã hội ở thời điểm hiện tại. Bên cạnh đó, các bài viết trong bộ dữ liệu này đa số về chủ đề chính trị và game, nghĩa là chủ đề chưa thực sự đa dạng đối với các trang mạng xã hội. Trong vấn đề gán nhãn, việc gán nhãn trong khoảng thời gian ngắn khiến các thành viên tham gia mắc phải nhiều lỗi

sai mặc dù đã có hướng dẫn gán nhãn tham khảo. Bên cạnh đó, nhiều quy định trong bản hướng dẫn cũng chưa thực sự rõ ràng. Đối với vấn đề huấn luyện mô hình, do dung lượng máy tính có hạn nên chưa thể khai thác tối đa khả năng huấn luyện của các mô hình và tiêu tốn khá nhiều thời gian.

Trong tương lai, để có thể cải thiện chất lượng của bộ dữ liệu, nhóm có thể thu thập dữ liệu mạng xã hội trong thời gian gần nhất có thể. Bên cạnh đó, trong tương lai cần thiết trau dồi thêm kiến thức về tiếng Việt, kiến thức xã hội và cải thiện các phương pháp gán nhãn. Để cải thiện hiệu suất chuẩn hóa văn bản, nhóm cần tạo ra bộ dữ liệu đủ nhiều cho mô hình học sâu. Khi đã cải thiện được hiệu suất chuẩn hóa văn bản, khả năng cao có thể mang lại thay đổi lớn trong các tác vụ NLP phức tạp khác.

TÀI LIỆU

[1] P. Żelasko, "Expanding abbreviations in a strongly inflected language: Are morphosyntactic tags sufficient?" 2018.

[2] R. Satapathy, Y. Li, S. Cavallari, and E. Cambria, "Seq2seq deep learning models for microtext normalization," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[3] H.-T. Dang, T.-H.-Y. Vuong, and X.-H. Phan, "Non-standard vietnamese word detection and normalization for text-to-speech," 2022. [Online]. Available: httpsarxiv.orgabs2209.02971

[4] I. Lourentzou, K. Manghnani, and C. Zhai, "Adapting sequence to sequence models for text normalization in social media," *CoRR*, vol. abs1904.06100, 2019. [Online]. Available: httparxiv.orgabs1904.06100

[5] S. T. Luu, K. V. Nguyen, and N. L. Nguyen, "A large-scale dataset for hate speech detection on vietnamese social media texts," *CoRR*, vol. abs/2103.11528, 2021. [Online]. Available: https://arxiv.org/abs/2103.11528

[6] T. Baldwin, M. C. de Marneffe, B. Han, Y.-B. Kim, A. Ritter, and W. Xu, "Shared tasks of the 2015 workshop on noisy user-generated text Twitter lexical normalization and named entity recognition," in *Proceedings of the Workshop on Noisy User-generated Text*. Beijing, China: Association for Computational Linguistics, url = httpsaclanthology.orgW15-4319, jul 2015, pp. 126–135.

[7] R. van der Goot, A. Ramponi, A. Zubiaga, B. Plank, B. Muller, I. San Vicente Roncal, N. Ljubevsi'c, O. cCetinouglu, R. Mahendra, T. cColakouglu, T. Baldwin, T. Caselli, and W. Sidorenko, "MultiLexNorm a shared task on multilingual lexical normalization," in *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*. Online: Association for Computational Linguistics, nov 2021, pp. 493–509. [Online]. Available: httpsaclanthology.org2021.wnut-1.55

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014. [Online]. Available: httpsarxiv.orgabs1409.0473

[9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014. [Online]. Available: httpsarxiv.orgabs1409.3215

[10] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015. [Online]. Available: httpsarxiv.orgabs1508.04025

[11] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," 2015. [Online]. Available: httpsarxiv.orgabs1506.03099