Phân loại bình luận độc hại & Ứng dụng streaming vào dữ liệu thời gian thực chủ đề COVID-19

Phạm Huỳnh Phúc[1,2], Huỳnh Khải Siếu[1,2], Phan Lực Lượng[1,2], and Trần Đình Kha[1,2]

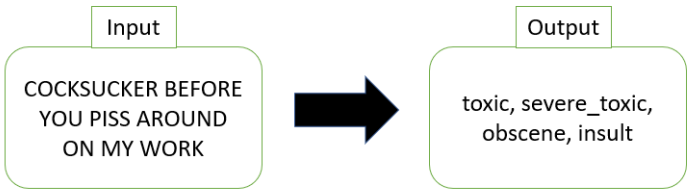
[1]Trường Đại học Công nghệ Thông tin - Đại học Quốc gia Thành phố Hồ Chí Minh
[2]{18521260 , 18520348, 18521073, 18520874}@gm.uit.edu.vn

Abstract. Trong đồ án này, chúng tôi đạt được mục tiêu chính là xử lí dữ liệu và cài đặt mô hình cho bài toán phân loại văn bản đa nhãn, sử dụng công cụ chính là pyspark. Chúng tôi chọn dữ liệu của cuộc thi Toxic Comment Classification Challenge, đây là bộ dữ liệu lớn với hơn 159.000 bình luận. Kết hợp mô hình MutilclassifierDL với các embedding khác nhau như là Universal Sentence Encoder, Bert-based Embedding, AlbertEmbeddings, ElmoEmbeddings và Sentence Embedding. Mục đích để so sánh kết quả của các mô hình tự động phân loại văn bản đa nhãn. Kết quả tốt nhất đạt được là ở mô hình MutilclassifierDL kết hợp Bert-based Embedding: 80,41% (F1-score). Bên cạnh đó, chúng tôi đã ứng dụng được streaming dữ liệu thời gian thực chủ đề COVID-19 trên mô hình tốt nhất.

1 Giới thiệu

Với cách mạng công nghiệp 4.0 dẫn đến sự phát triển không ngừng của mạng lưới internet và các diễn đàn trực tuyến, các nền tảng mạng xã hội truyền thông. Mạng xã hội truyền thông là nơi mọi người thỏa mái đưa ra suy nghĩ của họ và tự do trình bày những ý kiến về những vấn đề sự việc khác nhau. Theo Bruijn, Muhonen các mạng truyền thông xã hội đã phát triển theo cấp số nhân kể từ năm 2004. Dựa trên báo cáo của Birkland, "Người dùng Twitter tạo ra 500 triệu tweet mỗi ngày và vào năm 2019, họ đã có mức tăng trưởng 14% so với cùng kỳ năm trước về lượng sử dụng hàng ngày". Tuy nhiên, với sự phát triển nhanh chóng đó thì một số cá nhân đã có những hành động lạm dụng và quấy rối ý kiến của người khác qua những bình luận chứa ngôn từ tục tĩu, làm ảnh hưởng đến các cá nhân, tổ chức khác. Nhiều từ ngữ đe dọa có thể khiến cho người khác bị tổn hại tinh thần, rơi vào tình trạng lo sợ. Hệ thống Giám sát Hành vi Rủi ro Thanh niên năm 2017 (Trung tâm Kiểm soát và Phòng ngừa Dịch bệnh Hoa Kỳ) ước tính rằng 14,9% học sinh trung học bị bạo hành trên mạng xã hội trong 12 tháng, trước cuộc khảo sát. Những lời xúc phạm, đe dọa, bạo hành này ảnh hưởng đến tâm lí, có thể khiến cá nhân đó bị trầm cảm, thiếu lòng tự trọng và làm nảy sinh ý tưởng tự tử.
Để bảo vệ người dùng khỏi việc tiếp xúc với các ngôn ngữ xúc phạm trên các

diễn đàn trực tuyến hay các mạng xã hội, Kaggle họ đã tổ chức cuộc thi Toxic Comment Classification Challenge. Mục tiêu chính của cuộc thi này là phát triển một bộ phân loại multi-label để phân loại các nhận xét độc hại như đe dọa, tục tĩu, lăng mạ và thù hận. Tự động nhận dạng nội dung độc hại trong các diễn đàn trực tuyến và phương tiện truyền thông xã hội là một điều khoản hữu ích cho người kiểm duyệt các nền tảng công cộng cũng như người dùng có thể nhận được cảnh báo và lọc các nội dung không mong muốn. Việc phân loại bình luận không phù hợp có thể gây quấy rối quyền riêng tư và đạo đức là một trong những chủ đề quan trọng nhất liên quan đến việc mở rộng mạng xã hội hiện nay. Với tất cả những tiến bộ và cải tiến trong Công nghệ Thông tin nói chung và Khoa học Dữ liệu nói riêng, thế giới đang yêu cầu một kỹ thuật được thiết kế phù hợp để tìm và cô lập những loại nhận xét độc hại này.
Bài toán của cuộc thi được miêu tả như sau:



Hình 1: Đầu vào và đầu ra của bài toán.

2 Công trình liên quan

Từ cuộc thi Toxic Comment Classification Challenge do Kaggle, một số phương pháp đề xuất đạt hiệu quả tối ưu nhất đã được công bố. Mujahed A.Saif cùng cộng sự [7] đã đề xuất 4 mô hình để giải quyết bài toán của cuộc thi. Bốn mô hình gồm Logistic Regression và ba mô hình neural networks là Convolutional neural network (Conv), Long short term momery (LSTM) và Conv+LSTM. Tất cả mô hình được triển khai trên Python 3, tuy cấu trúc đơn giản nhưng đã được điều chỉnh để phù hợp với bài toán. Tất cả các mô hình đều thể hiện khả năng phân loại văn bản nhưng trong đó mô hình Conv + LSTM đạt hiệu quả tốt nhất, vì nó cung cấp độ chính xác cao nhất.
Ngoài ra, Betty van Akan cùng cộng sự [1] đã so sánh các phương pháp học sâu và nông cạn khác nhau trên tập dữ liệu này. Và từ đó Betty van Akan đã phân tích lỗi và áp dụng các giải pháp để cải thiện hiệu suất của mô hình. Những phương pháp họ thực nghiệm là Logistic Regression, Recurrent Neural Networks, Convolutional Neural Networks, (Sub)-Word Embeddings và Ensemble Learning. Và họ đã phát hiện và giải quyết một số lỗi như thiếu bối cảnh trong câu bình luận và nhãn tập dữ liệu không nhất quán với nhau.

3 Bộ dữ liệu

Chúng tôi sử dụng bộ dữ liệu của cuộc thi Toxic Comment Classification Challenge do Kaggle - một cộng đồng trực tuyến nổi bật trong lĩnh vực Khoa học Dữ liệu tổ chức, bộ dữ liệu có tổng cộng 4 file csv:

- train.csv - tập huấn luyện: chứa 159.571 bình luận có nhãn nhị phân của các loại tính độc hại cho các bình luận này.
- test.csv - tập thử nghiệm: gồm 153.146 bình luận và ID tương ứng.
- test_labels.csv: nhãn cho dữ liệu thử nghiệm. Giá trị của -1 cho biết nó không được sử dụng để cho điểm(đây là tệp được thêm vào sau khi cuộc thi kết thúc)
- sample_submission.csv: tệp gửi mẫu ở định dạng chính xác.

Trong phần này, chúng tôi chỉ sử dụng 3 tệp là train.csv, test.csv và test_label.csv. Bộ dữ liệu gồm 8 thuộc tính, 2 thuộc tính cơ bản là **ID** và **Comment**, 6 thuộc tính còn lại đại diện cho 6 loại tính độc hại xuất hiện trong câu bình luận là: **toxic, severe_toxic, obscene, threat, insult, identity_hate**. Tương ứng mỗi loại tính độc hại là một thuộc tính của bộ dữ liệu. Trong đó, các thuộc tính được mô tả trong bảng 1 như sau:

Tên thuộc tính	**Ý nghĩa**	**Miền giá trị**
ID	Mã định danh của bình luận	Không xác định
Comment	Nội dung bình luận	Không xác định
toxic	Có phát hiện yếu tố độc hại hay không	1: có, 0: không
severe_toxic	Có phát hiện yếu tố độc hại nghiêm trọng hay không	1: có, 0: không
obscene	Có phát hiện từ ngữ tục tĩu không	1: có, 0: không
threat	Có phát hiện yếu tố đe dọa không	1: có, 0: không
insult	Có phát hiện yếu tố sỉ nhục không	1: có, 0: không
identity_hate	Có phát hiện ngôn từ thù ghét hay không	1: có, 0: không

Bảng 1: Bảng mô tả thuộc tính bộ dữ liệu

Trong bảng 2, ta thấy chỉ cần bình luận có bất kì nhãn 1 nào trong 5 thuộc tính: severe_toxic, obscene, threat, insult, identity_hate, đều sẽ có nhãn 1 ở thuộc tính “toxic”.

ID	Comment	toxic	severe_toxic	obscene	threat	insult	identity_hate
0001d958c54c6e35	You, sir, are my hero. Any chance you remember what page that's on?	0	0	0	0	0	0
0020e7119b96eeeb	Stupid peace of shit stop deleting my stuff asshole go die and fall in a hole go to hell!	1	1	1	0	1	0
0020fd96ed3b8c8b	=Tony Sidaway is obviously a fistfuckee. He loves an arm up his ass.	1	0	1	0	1	0
0036621e4c7e10b5	Would you both shut up, you don't run wikipedia, especially a stupid kid.	1	0	0	0	1	0
fdce660ddcd6d7ca	I think he is a gay fag!!!	1	0	0	0	0	1

Bảng 2: Vài dòng của bộ dữ liệu

4 Tiền xử lý dữ liệu

Đầu tiên, chúng tôi gom tập test.csv và tập test_label.csv thành 1 tập có cấu trúc như tập train.csv. Vì bộ dữ liệu khá lớn dẫn đến vấn đề về bộ nhớ, dung lượng và cấu hình hệ thống khi huấn luyện mô hình ngoại tuyến và thử nghiệm hệ thống trực tuyến, chúng tôi quyết định lọc bỏ những bình luận không có tính độc hại nào, cụ thể là những bình luận có nhãn 0 ở tất cả 6 loại tính độc hại. Sau khi lọc, dữ liệu còn lại là tập huấn luyện gồm 16.225 bình luận và tập thử nghiệm gồm 6243 bình luận. Chúng tôi tiến hành chuyển nhãn ở 6 thuộc tính về 1 thuộc tính do pyspark chỉ hỗ trợ duy nhất một thuộc tính nhãn.
Dữ liệu của chúng tôi là dữ liệu mạng xã hội vì vậy cần thực hiện làm sạch văn bản để tránh bị nhiễu khi cài đặt mô hình. Chúng tôi sử dụng biểu thức chính quy để loại bỏ mention, hastag, URL, email, số và các ký tự và các khoảng trắng thừa, rồi chuẩn hoá tất cả các từ trong văn bản thành viết thường, giữ lại các dòng mà đoạn văn bản có nội dung. Bảng 3 dưới đây cho thấy văn bản trước và sau khi xử lý.

Câu bình luận trước khi xử lí	Câu bình luận sau khi xử lí
!!!	have you heard about dynamic ips about proxies about internet clubs you can t ban me from editing the true history of my folk you pathethic monkeydonian piece of shit
Have you heard about dynamic IPs? About proxies? About internet clubs? You can't ban me from editing the true history of my folk, you pathethic monkeydonian piece of shit	
"	has pictoral brown eyes then is jealous she s not a brown eyed girl remember the good times ip hopping around chicago miss but remember that only naturally brown eyes are attractive brown eyes blue eyes are brown only when they re full of shit
75.3.64.232 has pictoral brown eyes then.	
Is 75.3.64.232 jealous she's not a Brown-Eyed Girl? (-:	
Remember the good times IP hopping around Chicago, Miss 75.3.64.232.	
But remember that only naturally brown eyes are attractive brown eyes.	
Blue eyes are ""brown"" only when they're full of shit."	

Bảng 3: Bình luận trước và sau khi xử lý

5 Phương pháp & Thực nghiệm

Trong phần này, chúng tôi xây dựng mô hình phân loại văn bản đa nhãn MultiClassifierDL [8] kết hợp với các embedding ở 4.1, 4.2, 4.3, 4.4, 4.5. Sử dụng thư viện SparkNLP[1] do Apache Spark chưa hỗ trợ các công cụ để xây dựng mô hình Deep Learning trên Spark. SparkNLP là 1 thư viện mã nguồn mở cho xử lý ngôn ngữ tự nhiên trên Python, Java và Scala, được xây dựng dựa trên Apache Spark và thư viện Spark ML(MLib). Thư viện bao gồm nhiều pre-trained neural network models, pipelines, embeddings và cũng cho phép huấn luyện custom models.

Tất cả các bước tập hợp dữ liệu, tách từ, nhận dạng câu và biểu diễn dưới dạng vector trước MultiClassifierDL có thể được thực hiện trong một pipeline được chỉ định là một chuỗi các giai đoạn và mỗi giai đoạn là một Transformer hoặc một Estimator. Các giai đoạn này được chạy theo thứ tự và DataFrame đầu vào được chuyển đổi khi nó đi qua từng giai đoạn. Đó là, dữ liệu được chuyển qua pipeline mà trước đó đã cài đặt theo thứ tự. Phương thức biến đổi của mỗi giai đoạn cập nhật tập dữ liệu và chuyển nó sang giai đoạn tiếp theo. Với sự trợ giúp của Pipelines, chúng tôi có thể đảm bảo rằng dữ liệu đào tạo và kiểm tra trải qua các bước xử lý tính năng giống hệt nhau.

5.1 Universal Sentence Encoder (USE)

Universal Sentence Encoder [2]: là một mô hình giúp giải mã văn bản thành các vector chiều cao có thể được sử dụng để phân loại văn bản, từ đồng nghĩa, phân cụm và các tác vụ NLP khác.

5.2 Bert-based Embedding

BERT-based Embedding [3] là mô hình biểu diễn từ theo 2 chiều ứng dụng kỹ thuật Transformer, được thiết kế để huấn luyện trước các biểu diễn từ. Điểm đặc biệt ở BERT đó là nó có thể điều hoà cân bằng bối cảnh theo cả 2 chiều trái và phải.

5.3 AlbertEmbeddings

Albert Embedding [5] - mô hình học sâu NLP. Mô hình giúp giải quyết vấn đề về kích thước quá khổ của các mô hình pre-trained. Albert sử dụng hai cách tối ưu hoá để giảm kích thước mô hình: phân tích nhân tử của lớp embedding và chia sẻ tham số trên các lớp ẩn.

5.4 ElmoEmbeddings

Elmo Embeddings [6] là một phương pháp biểu diễn từ ngữ theo ngữ cảnh sâu. Nhiệm vụ chính của Elmo là trích xuất các tính năng từ văn bản đầu vào, cung cấp cho các nhiệm vụ về NLP. Elmo được tính toán trên mô hình ngôn ngữ hai chiều có hai lớp xếp chồng lên nhau.

[1] https://nlp.johnsnowlabs.com/

5.5 Sentence embedding

Sentence Embedding [4] có ứng dụng rộng rãi trong NLP như truy xuất thông tin, phân cụm, chấm điểm bài văn tự động và xét nghĩa tương đồng của văn bản về ngữ nghĩa. Khác với các embeddings khác, Sentence Embedding sẽ nhúng cả một câu thành vector có độ dài cố định thay vì nhúng từng từ. Sau đó các vector này được sử dụng để đánh giá mức độ tương đồng cosin của chúng bằng cách phản ánh các phán đoán của con người về ngữ nghĩa liên quan.

6 Phân tích kết quả

Để đánh giá hiệu suất của MultiClassifierDL khi kết hợp với các embedding, chúng tôi đã sử dụng ba độ đo phổ biến cho task classification là Precision, Recall và F1-score.

Bảng 4: Kết quả của MultiClassifierDL kết hợp với các embedding

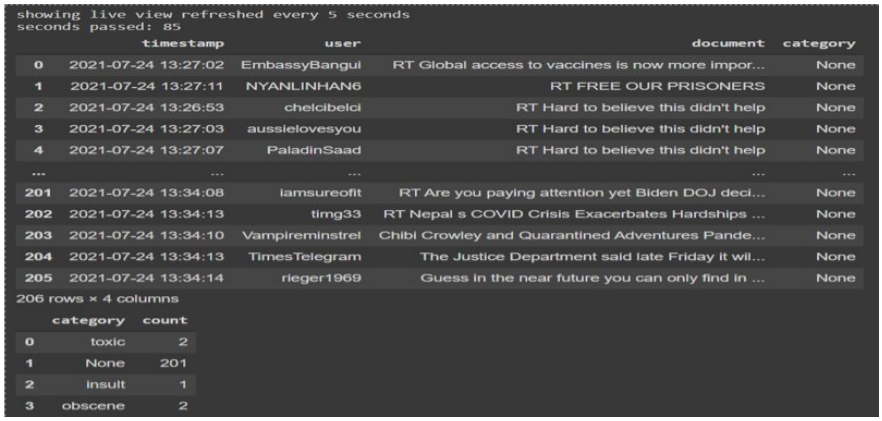
	Precision	**Recall**	**F1-score**
Universal Sentence Encoder	92.23	67.83	78.17
Bert-based-embedding	90.60	72.28	**80.41**
Sentence Embedding	91.42	63.93	75.24
Elmo Embedding	90.52	70.13	79.03
Albert Embedding	90.41	67.43	77.25

Bảng 4 thể hiện kết quả của MutilclassifierDL với các embedding. Ta có thể thấy, các kết quả không có sự chênh lệch đáng kể và giao động từ 75.24% đến 80.41%. Trong đó, thuật toán MutilclassifierDL + Bert based embedding đạt kết quả cao nhất với F1-score là 80.41%. Và Elmo embedding đạt kết quả thấp nhất với 75.24% F1-score.

7 Streaming Data

Từ MulticlassifierDL + Bert-based-embedding đạt F1-score là 80.41%. Chúng tôi đã tiền hành streaming data từ mạng xã hội Twitter với chủ đề là Covid-19. Chúng tôi đã tiền hành crawl dữ liệu với các thông tin là user, timestamp và text. Sau khi thu thập dữ liệu, chúng tôi đã tiến hành streaming trên mô hình đạt kết quả tốt nhất. Với số mẫu thử thu thập được là 206 tweet chúng tôi đã phân loại các bình luận tiêu cực. Trong đó 201 bình luận không có từ ngữ tiêu cực, 2 bình luận có từ ngữ toxic, 1 bình luận có từ ngữ sỉ nhục và 2 bình luận có từ ngữ tục tĩu. Phần lớn các bình luận không có từ ngữ tiêu cực là do chủ đề crawl là covid-19 nên đa số đều mang tính chất thời sự, ít từ ngữ tiêu cực.

Hình 2: Thu thập dữ liệu từ Twitter.



Hình 3: Streaming dữ liệu trên mô hình cao nhất

8 Kết luận & Hướng phát triển

Với bài toán đặt ra, chúng tôi đã cài đặt mô hình phân loại đa nhãn MulticlassifierDL với 5 loại embedding khác nhau đã trình bày ở phần 5. So sánh kết quả thực thi ở 5 loại embedding với nhau, chúng tôi thấy rằng mô hình kết hợp với Bert-based-embedding khả quan nhất, đạt 80.41% với độ đo F1-score.

Từ kết quả đó, chúng tôi tiếp tục sử dụng mô hình tốt nhất trên, ứng dụng Streaming lên bộ dữ liệu thời gian thực. Và mô hình của chúng tôi đã phân loại được các bình luận độc hại ở bộ dữ liệu này.

Bên cạnh đó, do dung lượng bộ dữ liệu khá lớn và sự hạn chế về mặt bộ nhớ, chúng tôi chỉ thực thi mô hình được trên một phần bộ dữ liệu. Điều đó làm cho kết quả đạt được thiếu sự khách quan, tin cậy. Chúng tôi sẽ cố gắng khắc phục điểm yếu này.

Đồng thời, hiệu suất mô hình cho ra dù đạt nhưng chỉ ở mức trung bình, chưa đạt yêu cầu của một mô hình phân loại. Chúng tôi hy vọng sẽ cải thiện thêm được hiệu suất mô hình.

Cuối cùng, chúng tôi hy vọng chủ đề phân loại và phát hiện bình luận tiêu cực này sẽ được quan tâm, phát triển và nghiên cứu nhiều hơn nữa. Hạn chế được những hậu quả tiêu cực xảy ra với người dùng trên mạng xã hội, diễn đàn trực tuyến.

References

[1] Betty van Aken et al. *Challenges for Toxic Comment Classification: An In-Depth Error Analysis*. 2018. arXiv: 1809.07572 [cs.CL].

[2] Daniel Cer et al. *Universal Sentence Encoder*. 2018. arXiv: 1803.11175 [cs.CL].

[3] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[4] Allyson Ettinger et al. "Assessing Composition in Sentence Vector Representations". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1790–1801. URL: https://aclanthology.org/C18-1152.

[5] Zhenzhong Lan et al. *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. 2020. arXiv: 1909.11942 [cs.CL].

[6] Matthew E. Peters et al. *Deep contextualized word representations*. 2018. arXiv: 1802.05365 [cs.CL].

[7] M.A Saif et al. "Classification of online toxic comments using the logistic regression and neural networks models". In: vol. 2048. Dec. 2018, p. 060011. DOI: 10.1063/1.5082126.

[8] William C. Sleeman IV and Bartosz Krawczyk. "Multi-class imbalanced big data classification on Spark". In: *Knowledge-Based Systems* 212 (2021), p. 106598. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2020.106598. URL: https://www.sciencedirect.com/science/article/pii/S0950705120307279.