# Phát hiện span độc hại trong bình luận theo thời gian thực sử dụng Spark NLP

## (Real-time: Toxic span detection using Spark NLP)

Nguyễn Đức Duy Anh[1,2,3], Dương Quốc Lộc[1,2,3], Đặng Hoàng Quân[1,2,3], and Võ Hồng Phúc Hạnh[1,2,3]

[1] University of Information Technology, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
[3] {18520455, 18521006, 18520339,18520275}@gm.uit.edu.vn

**Abstract.** Trong bài đồ án này, chúng tôi giải quyết bài toán phát hiện span độc hại kết hợp với phân loại bình luận độc hại theo thời gian thực dựa trên Spark NLP của dữ liệu tiếng Anh. Bên cạnh đó, chúng tôi đề xuất các mô hình như ClassifierDL, Bi-LSTM, CNNs, CRFs và mô hình học máy truyền thống Logistic Regression cùng với bộ word embedings là GloVeEmbedings, UniversalSentence EncoderEmbedings, BertEmbedings, ELECTRAEmbedings. Mô hình cho kết quả F1-score cao nhất là Glove+NerDL với 68.13% cdạng BIO và 68.55% dạng IO. Việc kết hợp giữa phát hiện span độc hại và phân loại bình luận độc hại không đem lại kết quả tốt. Ngoài ra, chúng tôi còn phân tích thêm các kết quả dựa trên độ đài của từ.

**Keywords:** span · độc hại · Spark NLP · phân loại · nhận dạng thực thể

## 1 Giới thiệu

Ngày nay, các nền tảng mạng xã hội ngày càng phát triển. Tại đây nó cho phép mọi người kết nối với nhau thông qua những chia sẻ những nội dung tích cực. Tuy nhiên, nó cũng chứa những nội dung tiêu cực như đe dọa, tấn công, lăng mạ hoặc xúc phạm. Đối với môi trường mạng xã hội chỉ cần một bình luận hay một văn bản mang tính tiêu cực dễ dàng kích động những người sử dụng mạng xã hội đó. Bên cạnh đó, nó cũng có thể gây ra bạo lực đối với những người lan truyền và người tiếp nhận. Những nội dung độc hại trên các trang mạng xã hội trực tuyến thì khác nhau dựa trên những nhóm mục tiêu bị tấn công khác nhau như trẻ em, phụ nữ, người đồng tính hay phân biệt biệt chủng tộc. Trong lĩnh vực Xử lí ngôn ngữ tự nhiên (NLP), các bài toán về phân loại các câu bình luận mang sắc thái tiêu cực hay tích cực đã đạt được kết quả cao. Tuy nhiên, việc xác định các từ hoặc cụm từ độc hại trong các văn bản trên mạng xã hội còn được gọi là bài toán Toxic Span Detection là một thách thức. Hội thảo SemEval-2021 Task 5 [7] đã đề ra bộ dữ liệu phục vụ cho việc nghiên cứu bài toán này.

Bài toán Toxic span detection mang lại hiệu quả cao hơn trong việc làm nổi bật những từ hoặc cụm từ độc hại so với việc chỉ đưa ra mức độ độc hại của

toàn văn bản. Không những vậy, nó cũng trở nên quan trọng trong việc kiểm duyệt nội dung cho các trang mạng xã hội. Trong bài báo này, chúng tôi giải quyết bài toán chủ yếu dựa trên bài toán Nhận diện thực thể (Named-entity recognition) thông qua nhiệm vụ gán nhãn trình tự (Sequence tagging task) và kết hợp với bài toán phân loại văn bản (Text classification) bằng mô hình học sâu kết hợp với các kỹ thuật embedding từ truyền thống và hiện đại. Tuy nhiên, các nền tảng mạng xã hội ngày nay có hàng tỷ người dùng, nó là một nguồn dữ liệu khổng lồ, nên việc áp dụng các thuật toán để giải quyết bài toán là điều khó khăn. Ngoài ra, việc kiểm duyệt các văn bản chứa các từ hoặc cụm từ độc hại trực tuyến và liên tục để tránh nội dung lan truyền là việc vô cùng cần thiết. Do đó, chúng tôi đã sử dụng framework Apache Spark để có thể áp dụng các thuật toán cho bài toán Toxic span detection trên một nguồn dữ liệu lớn trực tuyến và liên tục.

Qua quá trình thực hiện thực nghiệm mô hình cũng đem lại độ chính xác cao khi dự đoán những bình luận trực tuyến và liên tục. Cuối cùng, bài báo cũng trình bày triển khai demo trên web. Bài báo là hướng đi mới đối với bài toán Toxic span detection khi được thực hiện trên dữ liệu lớn và liên tục. Bài báo cũng mang lại hứa hẹn tiếp tục phát triển để cải thiện kết quả để mang lại độ chính xác cao hơn cho ứng dụng thực tế.

Trong bài báo này, chúng tôi tập trung giới thiệu các thông tin liên quan đến bài toán phát hiện chiều dài(span) từ độc hại kết hợp với phân loại bình luận độc hại theo thời gian thực dựa trên Spark NLP của dữ liệu tiếng Anh. Trong mục 2, chúng tôi sẽ trình bày về các công trình nghiên cứu liên quan. Trong mục 3, giới thiệu về bộ dữ liệu VLSP 2018 ABSA. Trong mục 4, chúng tôi sẽ giới thiệu các phương pháp được sử dụng để xử lý dữ liệu và huấn luyện mô hình. Quá trình thực nghiệm sẽ được mô tả trong mục 5 và mục 6 phân tích các kết quả. Cuối cùng là kết luận và hướng phát triển 7.

## 2 Công trình liên quan

A Supervised Multi-class Multi-label Word Embeddings Approach for Toxic Comment ClassificationThe problem [2] của tác giả Salvatore Carta và các cộng sự đã mô tả Vấn đề liên quan tới những bình luận độc hại là những công kích cá nhân, xúc phạm bằng lời nói và cách tấn công nhiều người tham gia trong một cuộc thảo luận. Bằng cách khai thác framework dữ liệu lớn Apache Spark và một số word embeddings. Bài báo trình bày cách tiếp cận có thể phân loại nhiều nhãn nhiều lớp của một cuộc thảo luận trong một phạm vi độc tính.

Trong phần An Apache Spark Implementation for Sentiment Analysis on Twitter Data [1] của sách Algorithmic Aspects of Cloud Computing của tác giả Alexandros Baltas và các cộng sự đã giới thiệu phân tích tình cảm trên dữ liệu Twitter là một vấn đề khó khăn do tính chất, tính đa dạng và khối lượng của dữ liệu. Trong này, chúng tôi triển khai một hệ thống trên Apache Spark, một khuôn khổ mã nguồn mở để lập trình với dữ liệu lớn. Hệ thống đề xuất đã được đào tạo và xác thực với dữ liệu thực do Twitter thu thập thông tin.

## 3 Bộ dữ liệu

Trong đồ án này chúng tôi sử dụng bộ dữ liệu được cung cấp từ SemEval-2021 Task 5: Toxic Spans Detection. Bộ dữ liệu gồm tập huấn luyện và tập kiểm thử. Trong mỗi bộ gồm hai thành phần là nội dung của bình luận và phần span thể hiện những từ độc hại trong bình luận. Bảng 1 là các ví dụ trong bộ dữ liệu ban đầu.

| STT | Bình luận | Spans |
|---|---|---|
| 1 | That's right. They are not normal. And I am starting from the premise that they are ABNORMAL. Proceed wth the typical racist, bigot, sexist rubbish. Thanks! | [84, 85, 86, 87, 88, 89, 90, 91, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146] |
| 2 | "Watch people die from taking away their healthcare" DING DING DING! Winner of stupid post of the day award! | [81, 82, 83, 84, 85, 86] |
| 3 | More garbage pro terrorist anti Canadian drivel from the Globe, how's the subscriptions going? anyone still pay for this junk. | [] |
| 4 | Some people donâ€™t need dope to be dopey. | [34, 35, 36, 37, 38] |

Table 1: Các ví dụ trong bộ dữ liệu ban đầu

Để phù hợp với từng nhiệm vụ con, chúng tôi tiến hành biến đổi dữ liệu. Cụ thể đối với nhiệm vụ phát hiện bình luận độc hại chúng tôi gán nhãn 1 là độc hại cho những câu có từ độc hại trong bình luận ngược lại là 0.

| Tập | Nhãn | Số lượng |
|---|---|---|
| Training | 1 | 8101 |
| | 0 | 528 |
| Test | 1 | 1606 |
| | 0 | 394 |

Table 2: Thống kê nhãn của bộ dữ liệu cho nhiệm vụ phân loại

Bảng 2 trình bày thống kê nhãn của bộ dữ liệu cho nhiệm vụ phân loại bình luận độc hại. Có sự chênh lệch khá lớn giữa nhãn 1 và nhãn 0. Phần lớn bộ dữ liệu là các câu bình luận có nhãn độc hại.

Bảng 3 là trình bày thống kê cụ thể nhãn cho mô hình NER theo format Conll. Có hai kiểu biểu diễn span là IOB và IO. Cụ thể với IOB có ba loại nhãn là B-T đại diện cho token đầu tiên của nằm trong span độc hại, I-T đại diện cho các token liền sau token đầu tiên của nằm trong span độc hại và O đại diện cho các token không nằm trong span độc hại. Đối với IO thì có hai loại nhãn

| | Tập | Nhãn | Số lượng |
|---|---|---|---|
| **IO** | Training | I-T | 17507 |
| | | O | 302229 |
| | Test | I-T | 2178 |
| | | O | 67220 |
| **IOB** | Training | B-T | 9846 |
| | | I-T | 7661 |
| | | O | 302229 |
| | Test | B-T | 1821 |
| | | I-T | 357 |
| | | O | 67220 |

Table 3: Thống kê nhãn của bộ dữ liệu cho nhiệm vụ phát hiện span

là I-T đại diện cho token nằm trong span độc hại và O đại diện cho các token không nằm trong span độc hại. Số lượng nhãn O luôn vượt trội hơn hẳn so với các nhãn khác ở cả dạng IOB và IO

## 4 Phương pháp

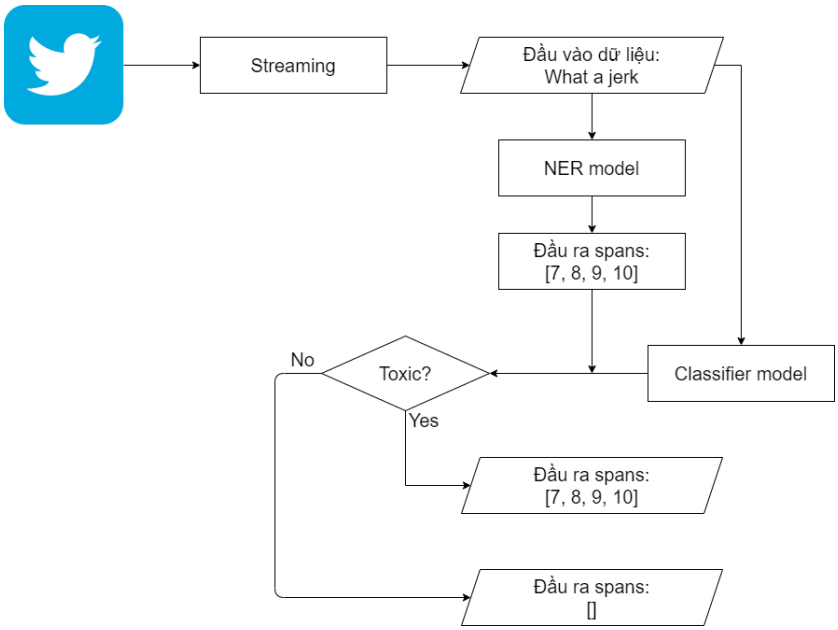


Fig. 1: Hướng tiếp cận bài toán

Chúng tôi đề ra hai hướng tiếp cận trong bài báo này, được thể hiện trong hình 1. Hướng tiếp cận thứ nhất là giải quyết theo bài toán phát hiện nhận dạng thực thể (Named Entity Recognitionn) thông qua nhiệm vụ gán nhãn trình tự (sequence tagging) để xác định các từ hoặc cụm từ độc hại trong câu bình luận. Chúng tôi thử nghiệm thông qua hai kiểu gán nhãn cho mỗi câu bình luận. Kiểu thứ nhất là IOB, trong đó B đại diện cho từ đầu tiên của cụm từ đầu ra là độc hại, I đại diện cho từ tiếp theo của cụm từ đó và O là đại diện cho những từ còn lại. Kiểu gán nhãn thứ hai là IO, trong đó I đại diện cho những từ đầu ra là độc hai và O là đại diện cho những từ còn lại.

Đối với hướng tiếp cận thứ nhất, chúng tôi sử dụng mô hình biểu diễn từ được huấn luyện từ trước (pre-trained word embedding) bao gồm Glove, BERT, Electra kết hợp với mô hình học sâu được nhúng trong thư viện của framework sparknlp là NerDL và mô hình học sâu NerCrf. Các mô hình này sữ dự đoán các từ hoặc cụm từ độc hại.

Hướng tiếp cận thứ hai là phân loại các câu bình luận là độc hại hoặc không độc hại bằng cách sử dụn mô hình biểu diễn từ được huấn luyện từ trước (pre-trained word embedding) là Universal Sentence Encoder và Glove kết hợp với mô hình học sâu được nhúng trong thư viện của framework sparknlp là classifierDL và mô hình học máy truyền thống Logistic Regression. Đối với những câu được phân loại là nhãn độc hại sẽ tiếp tục được phân loại để xác định những từ hoặc cụm từ độc hại theo cách tiếp cận thứ nhất.

### 4.1 Tiền xử lý dữ liệu

Trong xử lý ngôn ngữ tự nhiên nói chung, và xử lý ngôn ngữ tiếng Việt nói riêng thì bước đầu tiên mà chúng ta cần làm là tiền xử lý dữ liệu. Việc xử lý này nếu phù hợp với bài toán cũng như là phù hợp với dữ liệu có thể làm tăng hiệu suất. Do đó, tiền xử lý dữ liệu là quá trình chuẩn hóa dữ liệu và loại bỏ các thành phần không có ý nghĩa cho việc xử lý bài toán. Vì vậy chúng tôi đã thiết kế một số bước để xử lý dữ liệu nhằm khai thác tốt hơn các thông tin so với dữ liệu gốc. Các bước được thực hiện lần lược theo mô tả bên dưới:

### 4.2 Word embeddings

**Global Vectors for Word Representation (GloVe)** GloVe được giới thiệu bởi Jeffrey Pennington và các cộng sự năm 2014 [8]. Đây là mô hình học không giám sát để thu được các biểu diễn vector cho các từ. Mô hình dựa trên ý tưởng rằng tỷ lệ xác suất đồng xuất hiện giữa các từ có thể mã hóa một số dạng ý nghĩa đặc biệt dưới dạng các vector khác nhau. Có nghĩa rằng ánh xạ các từ vào một không gian vector, ở đó khoảng cách giữa các vector liên quan đến sự tương đồng về nghĩa của các từ.

**Universal Sentence Encoder** : Universal Sentence Encoder là bộ mã hóa các câu thành các vectơ chiều cao có thể được sử dụng để phân loại văn bản, tương tự ngữ nghĩa, phân cụm và các nhiệm vụ ngôn ngữ tự nhiên khác [3]. Mô hình

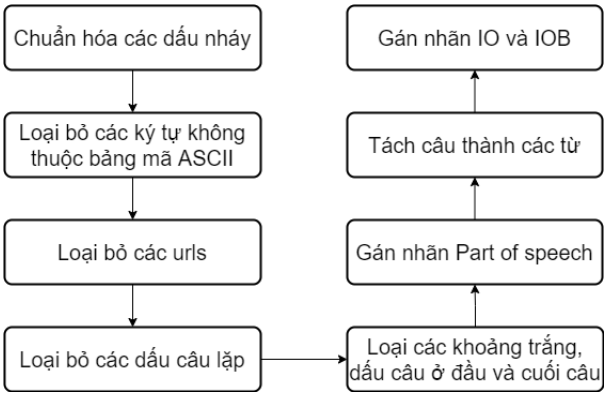


Fig. 2: Các bước tiền xử Lý

được đào tạo và tối ưu hóa cho văn bản dài hơn một từ, chẳng hạn như câu, cụm từ hoặc đoạn văn ngắn. Nó được đào tạo trên nhiều nguồn dữ liệu và nhiều nhiệm vụ khác nhau với mục đích đáp ứng một cách linh hoạt nhiều loại nhiệm vụ hiểu ngôn ngữ tự nhiên. Đầu vào là văn bản tiếng Anh có độ dài thay đổi và đầu ra là vectơ 512 chiều.

**Bidirectional Encoder Representation from Transformer (BERT)** BERT là mô hình biểu diễn ngôn ngữ được Google giới thiệu vào năm 2018[]. Tại thời điểm công bố, nó đã mang đến sự cải thiện đáng kể cho các bài toán thuộc lĩnh vực NLP. BERT được dùng để huấn luyện trước các biểu diễn từ (pre-train word embedding). Một điểm đặc biệt ở BERT mà các mô hình word embedding chưa từng có là kết quả huấn luyện có thể fine-tuning theo các nhiệm vụ huấn luyện. Mô hình BERT có thể học ngữ cảnh của từ bởi được đào tạo theo hai chiến lược sau:

**Masked LM:** Trước khi đưa vào mô hình BERT thì 15% số từ trong chuỗi được thay thế bởi token [MASK], khi đó mô hình sẽ dự đoán được từ thay thế bởi [MASK] với ngữ cảnh là các từ không bị thay thế bởi [MASK]

**Next sentence Prediction (NSP):** Trong chiến lược này, thì mô hình sử dụng một cặp câu câu dữ liệu đầu vào và dự đoán câu thứ hai có phải là câu kế tiếp của câu thứ nhất không. Trong quá trình huấn luyện, 50% lượng dữ liệu đầu vào là cặp câu trong đó câu thứ hai thực sự là câu tiếp theo của câu thứ nhất và 50% còn lại thì câu thứ hai được chọn ngẫu nhiên từ tập dữ liệu.

**Pre-training Text Encoders as Discriminators Rather Than Generators (ELECTRA)** ELECTRA[4] là mô hình huấn luyện trước biểu diễn từ (pre-train word embedding) được Google giới thiệu gần đây. Mô hình này được các giả được xây dựng dựa trên mô hình BERT nhưng quá trình trình học hiệu quả hơn đồng thời tiết kiệm được nguồn lực về tính toán qua các kết quả mà tác giả công bố.

### 4.3 Mô hình học máy truyền thống

**Logistics regression** [4] là một phương pháp phổ biến để dự đoán một phản hồi nhị phân. Đó là một trường hợp đặc biệt của mô hình Generalized Linear dự đoán xác suất của kết quả.

### 4.4 Mô hình học sâu

**ClassifierDL** [5] là một annotator của Spark NLP, nó được sử dụng cho bài toán phân loại đa lớp. ClassifierDL sử dụng SOTA Universal Sentence Encoder là đầu vào cho phân loại văn bản. Bên cạnh đó, ClassifierDL sử dụng mô hình học sâu (DNNs) để xây dựng bên trong TensorFlow và hỗ trợ lên tới 100 lớp (class). Annotator này chấp nhận nhãn thuộc kiểu dữ liệu String, Int, Float hoặc Double.

**NerDL** [6] là mô hình Bidirectional LSTM-CNNs-CRF dựa trên mô hình Bidirectional LSTM-CNNs[9] và mô hình Bidirectional LSTM-CRF[6], được mô tả trong hình 3. Lớp biểu diễn từ (word embedding) có nhiệm vụ biến đổi câu đầu vào thành các vector đặc trưng từ (word features). Các vector đặc trưng từ tiếp tục qua lớp CNNs sẽ được trích xuất thành các đặc trưng mới. Tiếp theo, các đặc trưng mới này sau khi đi qua lớp Bi-LSTM sẽ cho xác suất của mỗi nhãn ứng với mỗi từ của câu đầu vào. Cuối cùng những xác suất này đi qua lớp CRF sẽ cho đầu ra là tập nhãn ứng với mỗi từ của câu đầu vào.

**NerCrf** [7] Đây là mô hình học sâu kết hợp với phương pháp phân loại dựa trên xác suất có điều kiện được đề xuất bởi J.Laffety và các cộng sự (năm 2001) [5] chúng có thể tích hợp được các thuộc tính đa dạng của chuỗi dữ liệu quan sát nhằm hỗ trợ cho quá trình phân lớp. Tuy nhiên CRFs là các mô hình đồ thị vô hướng. Điều này cho cho phép CRFs có thể định nghĩa phân phối xác suất cho toàn bộ chuỗi trạng thái với điều kiện biết chuỗi quan sát cho trước.

## 5 Thực nghiệm

Với ClassifierDL chúng tôi sử dụng Universal Sentence Embeddings và GloVe Embeddings và cài đặt tham số như sau MaxEpochs 10, BatchSize 64, ValidationSplit 0.2, Learningrate 3e-6, Dropout 0.5, Device GPU.
Với NERDL chúng tôi sử dụng Bert Embeddings, Electra Embeddings, GloVe 100d và cài đặt tham số như sau MaxEpochs 5, BatchSize 32, ValidationSplit 0.2, Learningrate 1e-3, Dropout 0.5, Device GPU.
Đối với Logistic regression các tham số như sau maxIter 100, regParam 0, tol 1e-06, threshold = 0.5, aggregation_depth = 2.

[4] https://spark.apache.org/docs/latest/api/java/index.html?org/apache/spark/ml/classification/LogisticRegression.ht
[5] https://nlp.johnsnowlabs.com/api/com/johnsnowlabs/nlp/annotators/classifier/dl/ClassifierDLApproach
[6] https://nlp.johnsnowlabs.com/api/com/johnsnowlabs/nlp/annotators/ner/dl/NerDLApproach
[7] https://nlp.johnsnowlabs.com/api/com/johnsnowlabs/nlp/annotators/ner/crf/NerCrfApproach

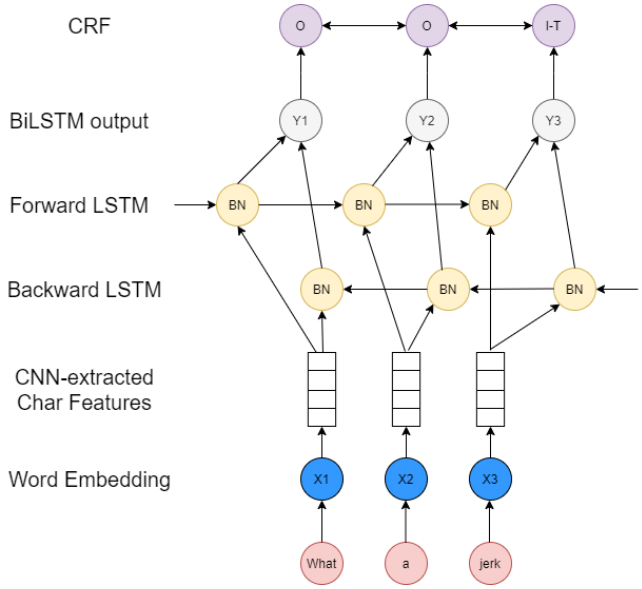


Fig. 3: Kiến trúc model NerDL

## 6 Phân tích kết quả

| Phương pháp | Độ dài từ | F1-score |
|---|---|---|
| BERT+NerDL | 1 | 79.33 |
| | 2-4 | 2.99 |
| | 5 | 0 |
| electra+NerDL | 1 | 72.95 |
| | 2-4 | 8.98 |
| | 5 | 0 |
| GloVe+NerDL | 1 | 81.89 |
| | 2-4 | 20.90 |
| | 5 | 0 |
| GloVe+NerCrf | 1 | 57.76 |
| | 2-4 | 40.83 |
| | 5 | 42.32 |

(a) IOB

| Phương pháp | Độ dài từ | F1-score |
|---|---|---|
| BERT+NerDL | 1 | 72.57 |
| | 2-4 | 29.95 |
| | 5 | 0 |
| electra+NerDL | 1 | 74.77 |
| | 2-4 | 21.84 |
| | 5 | 0 |
| GloVe+NerDL | 1 | 80.37 |
| | 2-4 | 27.94 |
| | 5 | 4.54 |
| GloVe+NerCrf | 1 | 72.91 |
| | 2-4 | 31.97 |
| | 5 | 4.54 |

(b) IO

Table 4: Kết quả các phương pháp theo độ dài từ

Bảng 4 trình bày kết quả của các phương pháp theo độ dài từ và được tính bằng F1-score macro. Trong đó, dạng IOB có mô hình NerDL sử dụng Bert Em-

| **Phương pháp** | **F1-score** |
|---|---|
| BERT+NerDL+LR | 55.22 |
| BERT+NerDL-ClassifierDL+GloVe | **57.21** |
| electra+NerDL+LR | 56.04 |
| electra+NerDL-ClassifierDL+GloVe | **58.17** |
| GloVe+NerDL+LR | 57.55 |
| GloVe+NerDL-ClassifierDL+GloVe | **59.83** |
| GloVe +NerCrf+LR | 54.47 |
| GloVe +NerCrf-ClassifierDL+GloVe | **56.48** |
| BERT+NerDL | 64.50 |
| electra+NerDL | **65.39** |
| GloVe+NerDL | **68.13** |
| GloVe+NerCrf | 63.37 |

(a) IOB

| **Phương pháp** | **F1-score** |
|---|---|
| BERT+NerDL+LR | 54.59 |
| BERT+NerDL-ClassifierDL+GloVe | **56.73** |
| electra+NerDL+LR | 56.01 |
| electra+NerDL-ClassifierDL+GloVe | **58.05** |
| GloVe+NerDL+LR | 57.99 |
| GloVe+NerDL-ClassifierDL+GloVe | **60.30** |
| GloVe +NerCrf+LR | 43.70 |
| GloVe +NerCrf-ClassifierDL+GloVe | **45.26** |
| BERT+NerDL | 63.91 |
| electra+NerDL | **64.51** |
| GloVe+NerDL | **68.55** |
| GloVe+NerCrf | 51.19 |

(b) IO

Table 5: Kết quả các phương pháp

beddings với F1-score là 79.33%, mô hình NerDL sử dụng Electra Embeddings với F1-score là 72.95%, mô hình NerDL sử dụng GloVe Embeddings với F1-score là 81.89% cho kết quả cao với độ dài 1, cao nhất là mô hình NerDL sử dụng GloVe Embeddings. Bên cạnh đó, kết quả lại rất thấp với độ dài 2-4 và không dự đoán được gì ở độ dài 5 F1-score đều bằng 0%. Ngoài ra, mô hình NerCrf sử dụng GloVe Embeddings cho kết quả khá đồng đều với độ dài 1, 2-4 và 5 lần lượt là 57.76%, 40.83%, 42.32%. Mặc dù kết quả ở độ dài 1,2-4 không cao bằng những mô hình khác nhưng ở độ 5 thì lại vượt trội so với mô hình khác. Đối với dạng IO có mô hình NerDL sử dụng Bert Embeddings, mô hình NerDL sử dụng Electra Embeddings, mô hình NerDL sử dụng GloVe Embeddings, mô hình NerCrf sử dụng GloVe Embeddings cho kết quả thấp với độ dài 2-4 và rất thấp ở độ dài 5; với độ dài 1 cả 4 phương pháp đều cho kết quả khá cao, đều trên 72%. Tương tự với dạng IOB, mô hình NerDL sử dụng GloVe Embeddings đạt được kết quả F1-score 80.37%, cao nhất trong các mô hình của dạng IO. Trong cả hai dạng IO và IOB phương pháp GloVe Embeddings kết hợp với NerDL thể hiện ưu việt nhất. Ở dạng IO phương pháp chỉ kém 1.5% ở độ dài 1 từ so với dạng IOB. Tuy nhiên, ở độ dài 2-4 và 5 từ phương pháp ở dạng IO lại nhỉnh hơn IOB.

Bảng 5 trình bày kết quả của các phương pháp phối hợp giữa các mô hình phát hiện span độc hại với mô hình phân loại bình luận độc hại và cũng được tính bằng F1-score macro. Đối với các trường hợp có sử dụng kết hợp mô hình phân loại, mô hình ClassifierDL sử dụng GloVe Embeddings cho kết quả tốt hơn Logistic Regression ở cả hai dạng IOB và IO. Mô hình NerDL sử dụng GloVe Embeddings kết hợp với mô hình phân loại ClassifierDL cùng với GloVe Embeddings cho kết quả tốt nhất, cụ thể 59.83% ở IOB và 60.30% ở IO. Còn đối với trường hợp không kết hợp mô hình phân loại, mô hình NerDL kết hợp với GloVe Embeddings cho kết quả cao nhất ở cả hai dạng IOB và IO, cụ thể

lần lượt là 68.13% và 68.55%. Ở cả hai dạng IOB và IO, các mô hình đều đem lại kết quả cao hơn khi không kết hợp với mô hình phân loại.

## 7 Kết luận và hướng phát triển

Trong đồ án này, chúng tôi đã giải quyết được bài toán phát hiện span độc hại trong các bình luận kết hợp với phân loại bình luận độ hại trên bộ dữ liệu tiếng Anh theo thời gian thực dựa trên Spark NLP. Kết quả thu được khá khả quan mô hình GloVe+NerDL cho kết quả F1-score cao nhất ở IOB và IO, cụ thể 68.13% dạng IOB và 68.55% dạng IO. Trong khi đó, mô hình GloVe+NerCrf+LR cho kết quả thấp nhất ở IOB và IO, cụ thể 54.47% dạng IOB và 43.70% dạng IO. Ngoài ra, các mô hình phát hiện span độc hại khi phối hợp mô hình với các mô hình phân loại cho kết quả thấp hơn các mô hỉnh mô hình phát hiện span độc hại riêng lẻ. Bên cạnh đó, chúng tôi đã tiến hành phân tích kết quả mô hình dựa trên độ dài span. Những span dạng IOB có kết quả F1-score của các mô hình BERT+NerDL, electra+NerDL, GloVe+NerDL lần lượt là 79.33%, 72.95%, 81.89% . Chúng cho kết quả cao với độ dài 1, kết quả thấp với độ dài 2-4 và 5. Trong khi đó mô hình GloVe+NerCrf cho kết quả F1-score khá đồng đều với độ dài 1, 2-4 và 5 lần lượt là 57.76%, 40.83% và 42.32%. Đối với span dạng IO có BERT+NerDL, electra+NerDL, GloVe+NerDL,GloVe+NerCrf cho kết quả thấp với độ dài 2-4 và 5; với độ dài 1 cả 4 phương pháp đều cho kết quả khá cao, tương tự dạng IOB GloVe+NerDL cao nhất với F1-score 80.37%.

Trong tương lai, chúng tôi dự định sẽ tìm hiểu các thư viện xây dựng mô hình và tiến hành thực nghiệm các bộ embeddings khác.

## References

1. Alexandros Baltas, Andreas Kanavos, and Athanasios K Tsakalidis. An apache spark implementation for sentiment analysis on twitter data. In *International Workshop of Algorithmic Aspects of Cloud Computing*, pages 15–25. Springer, 2016.
2. Salvatore Carta, Andrea Corriga, Riccardo Mulas, Diego Reforgiato Recupero, and Roberto Saia. A supervised multi-class multi-label word embeddings approach for toxic comment classification. 2019.
3. Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.
4. Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.
5. John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
6. Rrubaa Panchendrarajan and Aravindh Amaresan. Bidirectional lstm-crf for named entity recognition. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, 2018.
7. John Pavlopoulos, Léo Laugier, Jeffrey Sorensen, and Ion Androutsopoulos. Semeval-2021 task 5: Toxic spans detection. *Proceedings of SemEval*, 2021.

8. Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
9. Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.