**NER (Name Entity Recognition)**

1. **Mô tả bài toán NER**

NER hay còn được biết đến là nhận dạng thực thể có tên, là một phần của Xử lý ngôn ngữ tự nhiên (NLP). NER được phát triển nhằm tìm kiếm và phân loại các dữ liệu trong văn bản vào trong những danh mục đã được xác định trước đó ví dụ như tên người, tổ chức, địa danh, địa điểm, thời gian, số lượng, giá trị tiền tệ,…

Như vậy, mục tiêu của NER là tìm kiếm thực thể trong dữ liệu và phân loại thực thể được đặt tên theo danh mục, từ đó giúp cho việc xử lý và phân tích thông tin dễ dàng hơn.

Ví dụ: Cho câu sau : “Sugar sinh ngày 1 tháng 4 năm 2000, anh lớn lên và làm việc tại Thành phố Hồ Chí Minh.”

Trong câu trên các thực thể được NER nhận dạng gồm có tên người (Sugar), thời gian (ngày 1 tháng 4 năm 2000) và địa điểm (Thành phố Hồ Chí Minh)

NER biến đổi văn bản dạng thô thành dữ liệu có cấu trúc do đó nó được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau như:

* Xử lý ngôn ngữ tự nhiên (NLP): tóm tắt văn bản, trả lời câu hỏi, dịch máy,…
* Trích xuất thông tin (IE – Information Extraction): trích xuất thông tin từ các tài liệu phi cấu trúc như email, báo cáo,…
* Xây dựng Chatbot

1. **Hướng tiếp cận và xây dựng mô hình NER**

Có 3 hướng tiếp cận NER:

* Rule-based
* Statistical Learning
* Machine Learning/Deep Learning.

1. **Rule-based**

Rule-based (Dựa trên quy tắc) là phương pháp sử dụng các quy tắc ngữ pháp, từ vựng và ngữ nghĩa để xác định và phân loại các thực thể trong văn bản. Các quy tắc này thường được viết thủ công bởi các chuyên gia ngôn ngữ học và có thể bao gồm các mẫu cú pháp, từ điển, và các quy tắc logic.

**Ưu điểm:**

* Chính xác với các miền cụ thể: Khi các quy tắc được thiết kế tốt cho một miền cụ thể, phương pháp này có thể đạt độ chính xác cao.
* Dễ dàng điều chỉnh và kiểm soát: Các quy tắc có thể được điều chỉnh một cách dễ dàng dựa trên sự thay đổi trong yêu cầu hoặc dữ liệu.

**Nhược điểm:**

* Tốn thời gian và công sức: Việc thiết kế và duy trì các quy tắc đòi hỏi nhiều công sức và kiến thức chuyên môn.
* Khả năng mở rộng kém: Khó khăn trong việc áp dụng cho các miền hoặc ngôn ngữ khác nhau do tính đặc thù của các quy tắc.

Ví dụ, một quy tắc đơn giản trong NER có thể là: "Nếu từ sau là một động từ và từ trước là một danh từ, từ đó có khả năng cao là tên người."

1. **Statistical Learning**

Statistical Learning (Học thống kê) sử dụng các mô hình xác suất để xử lý nhiệm vụ NER. Các mô hình này học từ dữ liệu đã được gán nhãn để xác định các thực thể dựa trên các đặc trưng ngữ cảnh và hình thái học.

**Các phương pháp phổ biến:**

* Hidden Markov Models (HMMs): Sử dụng chuỗi các biến ẩn để mô hình hóa các mối quan hệ giữa các từ trong văn bản.
* Conditional Random Fields (CRFs): Một phương pháp mạnh mẽ hơn HMMs, CRFs mô hình hóa mối quan hệ giữa các từ và các nhãn với điều kiện trên các đặc trưng quan sát được từ dữ liệu.

**Ưu điểm:**

* Tự động học từ dữ liệu: Không cần thiết kế quy tắc thủ công.
* Tổng quát hóa tốt hơn: Có thể áp dụng cho nhiều miền và ngôn ngữ khác nhau nếu có đủ dữ liệu huấn luyện.

**Nhược điểm:**

* Yêu cầu nhiều dữ liệu: Để mô hình hoạt động tốt, cần một lượng lớn dữ liệu đã gán nhãn.
* Khó khăn trong diễn giải: Kết quả từ các mô hình thống kê có thể khó diễn giải so với các quy tắc rõ ràng.

Ví dụ, CRF có thể học rằng từ "Washington" có khả năng cao là tên địa điểm khi xuất hiện trong ngữ cảnh nhất định.

1. **Machine Learning/Deep Learning**

Machine Learning/Deep Learning (Học máy/Học sâu) là phương pháp sử dụng các thuật toán học máy và mạng nơ-ron để học các mẫu từ dữ liệu. Các mô hình này tự động học các đặc trưng từ dữ liệu huấn luyện mà không cần các quy tắc thủ công.

**Các phương pháp phổ biến:**

* Support Vector Machines (SVMs): Một phương pháp học máy tuyến tính có thể được sử dụng cho NER.
* Recurrent Neural Networks (RNNs) và Long Short-Term Memory (LSTM): Mạng nơ-ron hồi tiếp và biến thể của nó, LSTM, được sử dụng để xử lý dữ liệu chuỗi như văn bản.
* Transformers (như BERT, GPT): Các mô hình mạng nơ-ron mạnh mẽ hơn, dựa trên cơ chế tự chú ý, hiện đang là state-of-the-art cho nhiều nhiệm vụ NLP, bao gồm cả NER.

**Ưu điểm:**

* Hiệu suất cao: Đạt được hiệu suất vượt trội trên nhiều tập dữ liệu và nhiệm vụ.
* Tự động học đặc trưng: Không cần thiết kế đặc trưng thủ công, mô hình tự động học từ dữ liệu.

**Nhược điểm:**

* Yêu cầu tài nguyên tính toán lớn: Cần nhiều tài nguyên tính toán và thời gian huấn luyện.
* Khó khăn trong diễn giải: Các mô hình học sâu thường được coi là "hộp đen," khó diễn giải và hiểu.

Ví dụ, mô hình BERT có thể học rằng "Apple" có thể là tên công ty trong câu "Apple Inc. đã ra mắt sản phẩm mới" và là tên quả trong câu "Tôi ăn một quả táo."

1. **Đánh giá mô hình**

* Đánh giá hiệu quả của mô hình NER bằng cách sử dụng các chỉ số đánh giá như:
* Độ chính xác (Accuracy): Tỷ lệ phần trăm các thực thể được dự đoán chính xác.
* Độ chính xác theo phân loại thực thể (Precision): Tỷ lệ phần trăm các thực thể thuộc mỗi loại được dự đoán chính xác.
* Độ thu hổi (Recall): Tỷ lệ phần trăm các thực thể trong văn bản được dự đoán.
* Độ F1 (F1 Score): Trung bình điều hòa giữa độ chính xác và độ thu hồi.

1. **Lợi ích và thách thức của NER**
2. **Lợi ích**

* Cải thiện hiệu quả tìm kiếm và truy xuất thông tin: NER giúp các hệ thống tìm kiếm và truy xuất thông tin trở nên chính xác hơn bằng cách nhận diện và phân loại các thực thể quan trọng trong văn bản. Điều này giúp cung cấp các kết quả tìm kiếm liên quan và hữu ích hơn.
* Tự động hóa và tăng tốc độ trích xuất thông tin: NER tự động trích xuất thông tin có cấu trúc từ dữ liệu phi cấu trúc, tiết kiệm thời gian và công sức so với việc xử lý thủ công. Điều này đặc biệt hữu ích trong các lĩnh vực như tài chính, y tế, và pháp lý.
* Hỗ trợ xử lý ngôn ngữ tự nhiên trong các ứng dụng: NER đóng vai trò quan trọng trong việc phát triển các ứng dụng NLP như trợ lý ảo, chatbot, và hệ thống phân tích văn bản, giúp chúng hiểu và xử lý các yêu cầu của người dùng một cách chính xác và hiệu quả.
* Cải thiện trải nghiệm người dùng: Bằng cách nhận diện và phản hồi chính xác các thực thể như tên người, địa điểm, và tổ chức, NER giúp cải thiện trải nghiệm người dùng trong các ứng dụng tương tác như trợ lý ảo và chatbot.
* Hỗ trợ dịch thuật và bảo tồn ngữ nghĩa: Trong dịch thuật tự động, NER giúp nhận diện và giữ nguyên các thực thể tên riêng, đảm bảo tính chính xác và ngữ nghĩa của bản dịch, đặc biệt là trong các văn bản chuyên ngành.
* Phân tích và đánh giá thương hiệu: NER giúp nhận diện các thực thể liên quan đến thương hiệu, sản phẩm, hoặc dịch vụ trong phản hồi của khách hàng, hỗ trợ các công ty trong việc phân tích cảm xúc và đánh giá thương hiệu một cách hiệu quả.

1. **Thách thức**

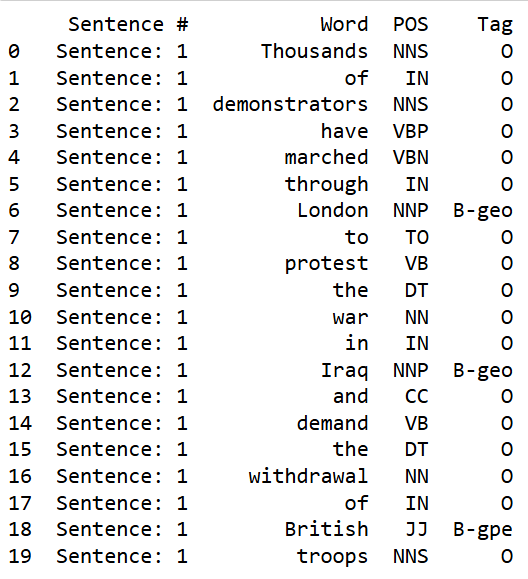
* Độ chính xác phụ thuộc vào ngữ cảnh và miền dữ liệu: Các mô hình NER có thể gặp khó khăn trong việc nhận diện và phân loại chính xác các thực thể khi ngữ cảnh hoặc miền dữ liệu thay đổi. Ví dụ, một thực thể có thể mang nghĩa khác nhau trong các lĩnh vực khác nhau.
* Yêu cầu nhiều dữ liệu huấn luyện: Các phương pháp NER hiện đại, đặc biệt là các mô hình học sâu, yêu cầu một lượng lớn dữ liệu đã gán nhãn để huấn luyện mô hình. Việc thu thập và gán nhãn dữ liệu này có thể tốn kém và mất thời gian.
* Khó khăn trong việc xử lý ngôn ngữ đa dạng: NER phải đối mặt với thách thức trong việc xử lý các ngôn ngữ có cấu trúc phức tạp hoặc không có dấu câu rõ ràng như tiếng Trung, tiếng Nhật, hoặc tiếng Ả Rập. Các ngôn ngữ có dấu diacritics như tiếng Việt cũng có thể gây khó khăn.
* Giới hạn trong việc nhận diện các thực thể mới: Các mô hình NER có thể không nhận diện được các thực thể mới hoặc ít phổ biến mà chúng chưa từng gặp trong dữ liệu huấn luyện. Điều này đòi hỏi mô hình phải được cập nhật thường xuyên với dữ liệu mới.
* Khả năng mở rộng và áp dụng trong thực tế: Mặc dù NER có thể hoạt động tốt trong môi trường nghiên cứu, việc triển khai và áp dụng trong các ứng dụng thực tế có thể gặp khó khăn, đặc biệt là khi đối mặt với khối lượng lớn dữ liệu và yêu cầu xử lý thời gian thực.
* Khó khăn trong diễn giải kết quả: Kết quả từ các mô hình NER, đặc biệt là các mô hình học sâu, có thể khó diễn giải và hiểu. Điều này gây khó khăn trong việc kiểm tra và đảm bảo tính chính xác của các thực thể được nhận diện.

1. **Áp dụng xây dựng bài toán**

Mục tiêu của bài toán này là xác định và phân loại các thực thể có tên trong văn bản thành các loại như tên người (Person), địa điểm (Location), tổ chức (Organization), và các loại thực thể khác. Trong báo cáo này, chúng tôi sẽ xây dựng và đánh giá 2 mô hình (RNN và BiLSTM) trên cùng một bộ dữ liệu để nhận dạng các thực thể đặt tên.

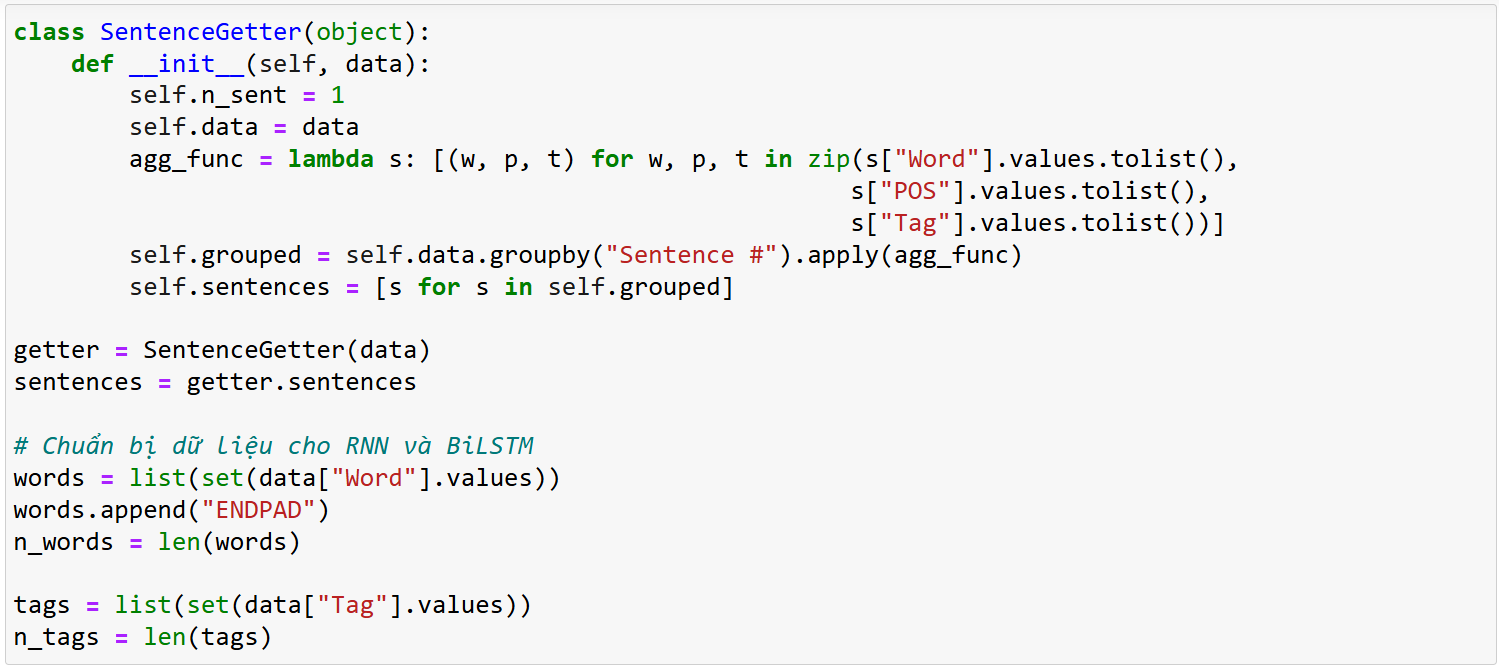
**a. Dữ liệu**

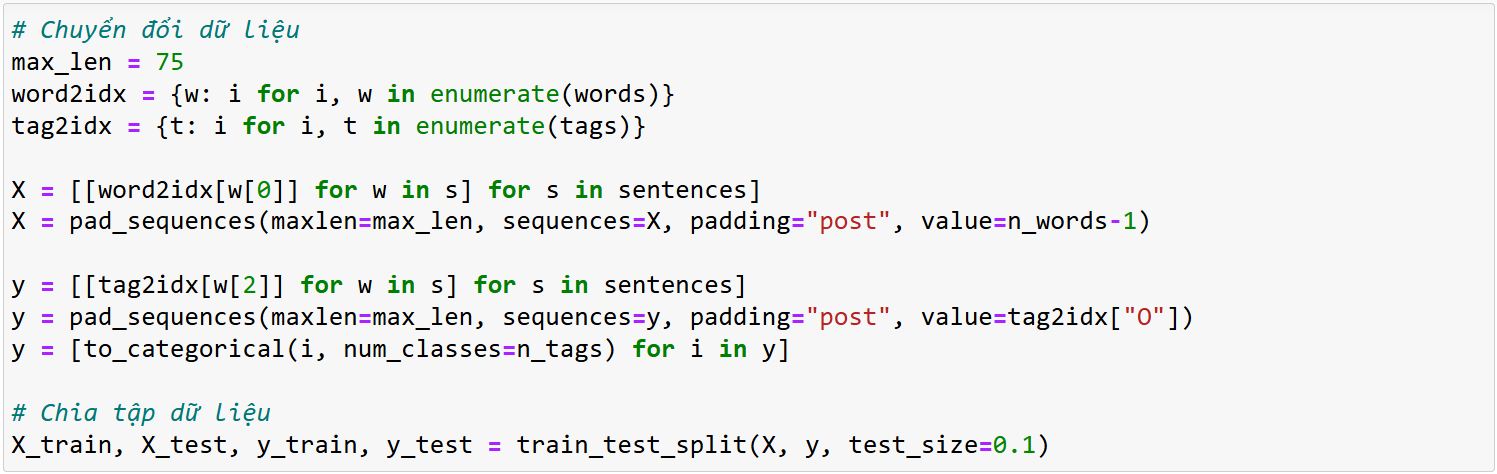
* Nguồn dữ liệu được sử dụng là bộ dữ liệu “ner\_dataset.csv”, được cung cấp bởi Kaggle.
* Cấu trúc dữ liệu bao gồm các cột Sentence #, Word, POS, và Tag.
* Sentence # biểu thị số thứ tự của câu.
* Word là từ cần được gán nhãn.
* POS là phần loại từ (Part of Speech).
* Tag là nhãn NER của từ tương ứng. Bao gồm các loại thực thể như B-geo, B-org, B-per, I-geo, I-org, I-per và các nhãn khác.



**b. Xử lý dữ liệu**

Đầu tiên, nhóm sẽ đọc và làm sạch dữ liệu, điền các giá trị bị thiếu. Sau đó, chúng tôi chuyển đổi các từ và nhãn thành các chỉ số (index) và chia dữ liệu thành các tập huấn luyện và kiểm tra.



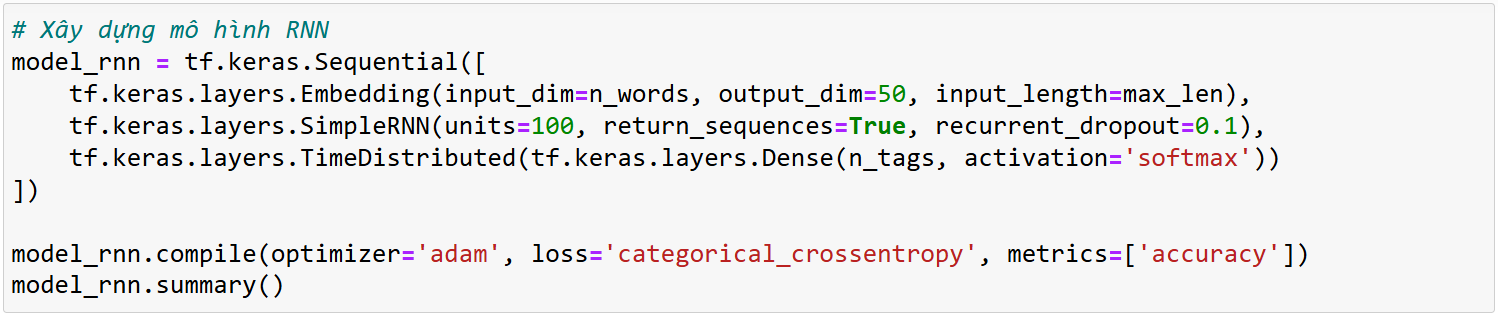


**c. Xây dựng các mô hình**

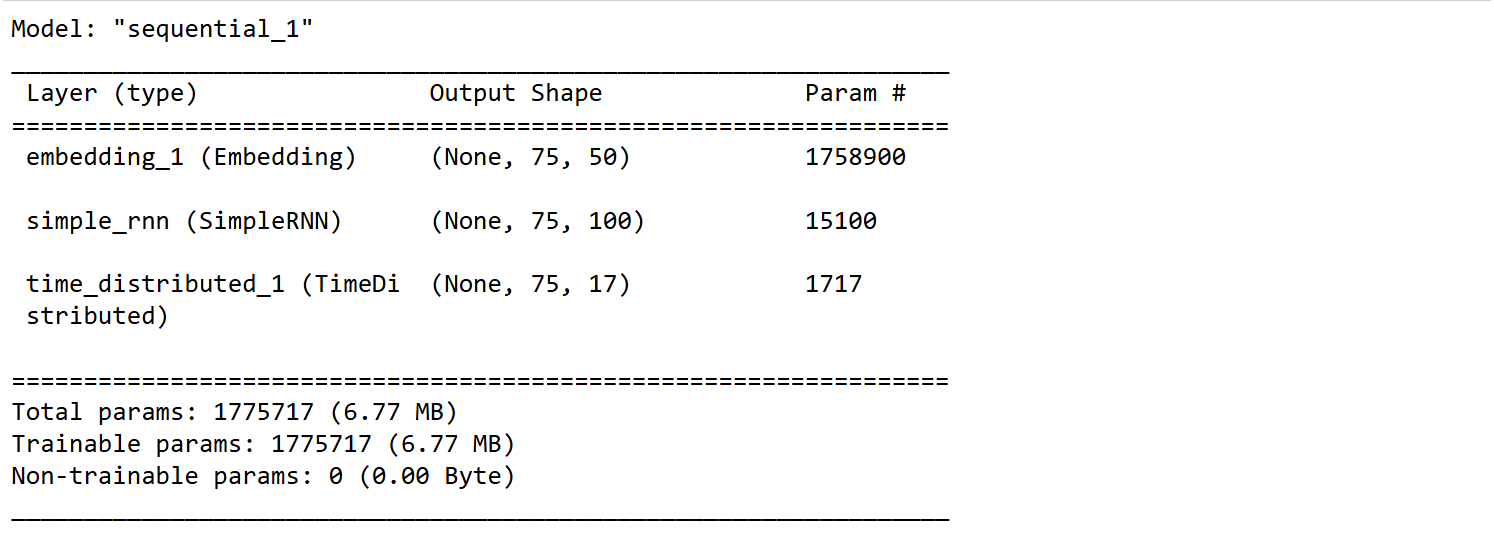
Nhóm xây dựng trên 2 mô hình gồm RNN, BiLSTM.

***Với mô hình RNN:***

* Sử dụng lớp Embedding để biểu diễn từ vựng thành các vectơ số. Mỗi từ trong từ điển sẽ được biểu diễn dưới dạng một vectơ có số chiều cố định. Việc sử dụng lớp Embedding giúp mô hình học được các mối quan hệ ngữ nghĩa giữa các từ trong văn bản.
* Lớp SimpleRNN là một lớp mạng nơ-ron tái phát (recurrent neural network) đơn giản. Sử dụng lớp SimpleRNN để xây dựng mô hình RNN.
* Sử dụng lớp TimeDistributed được sử dụng để áp dụng lớp Dense (hoặc lớp kích hoạt) cho mỗi mẫu và mỗi bước thời gian. Trong mô hình này, lớp Dense giúp chúng ta dự đoán xác suất cho mỗi nhãn NER.

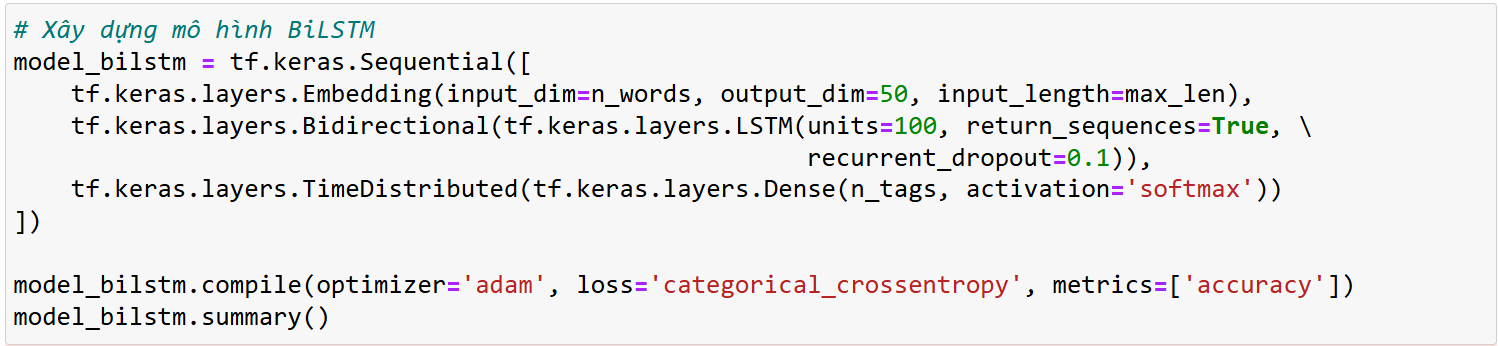


* Kết quả tổng quan về mô hình: Mô hình có tổng cộng 1,775,717 tham số, trong đó có 1,758,900 tham số từ lớp Embedding, 15,100 tham số từ lớp SimpleRNN và 1,717 tham số từ lớp TimeDistributed. Tất cả các tham số đều được huấn luyện (trainable params), tức là chúng sẽ được điều chỉnh trong quá trình huấn luyện. Mô hình có thể được tối ưu bằng thuật toán tối ưu hóa Adam với hàm mất mát là cross-entropy và các metric đánh giá là accuracy.

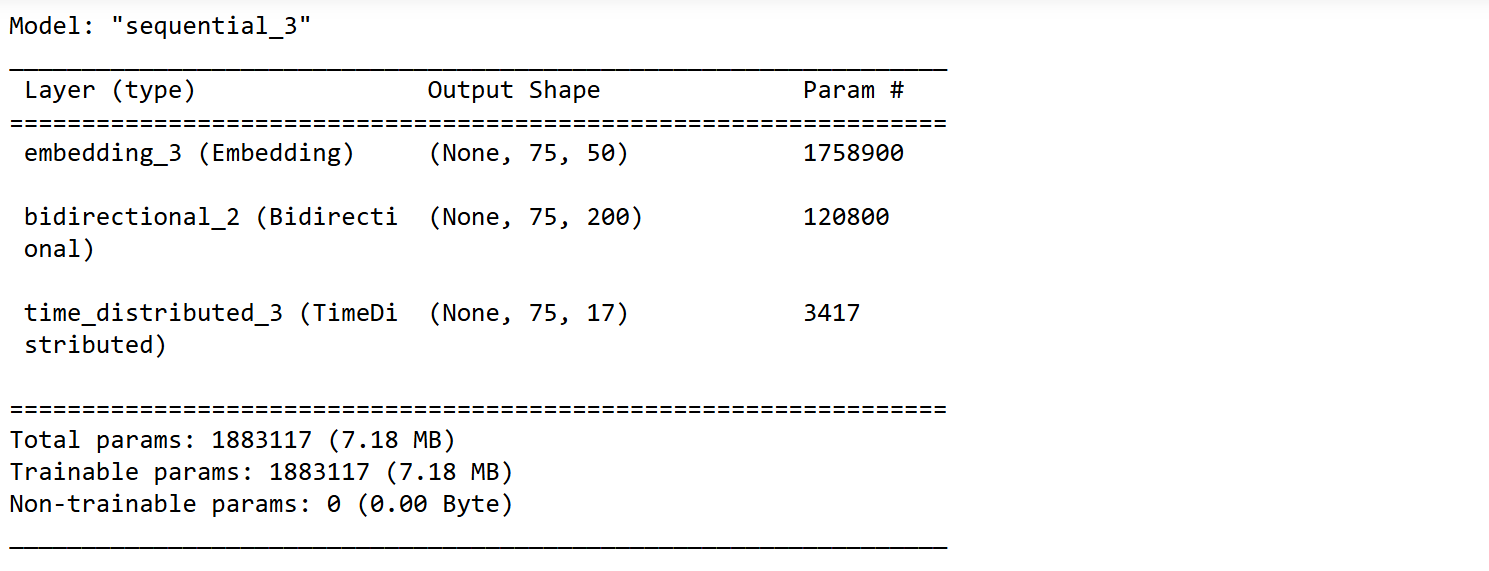


***Với mô hình BiLSTM***.

* Sử dụng lớp Embedding để biểu diễn từ vựng thành các vectơ số. Mỗi từ trong từ điển sẽ được biểu diễn dưới dạng một vectơ có số chiều cố định. Việc sử dụng lớp Embedding giúp mô hình học được các mối quan hệ ngữ nghĩa giữa các từ trong văn bản.
* Lớp Bidirectional LSTM (BiLSTM) là một biến thể của mạng LSTM, cho phép mô hình học thông tin từ cả hai hướng trong chuỗi từ. Sử dụng lớp Bidirectional(LSTM) để xây dựng mô hình BiLSTM.
* Sử dụng lớp TimeDistributed được sử dụng để áp dụng lớp Dense (hoặc lớp kích hoạt) cho mỗi mẫu và mỗi bước thời gian. Trong mô hình này, lớp Dense giúp chúng ta dự đoán xác suất cho mỗi nhãn NER.

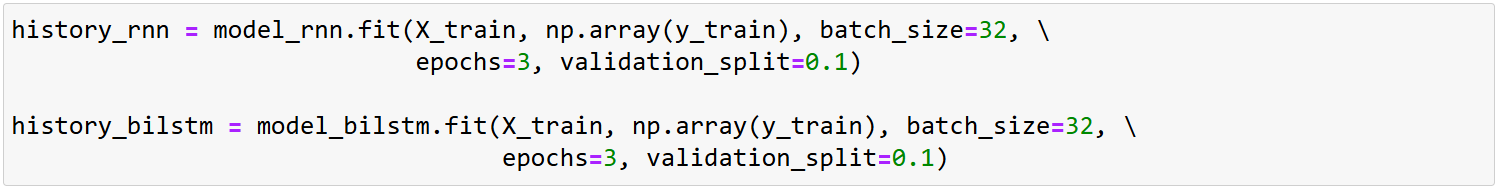


* Kết quả tổng quan về mô hình: Mô hình có tổng cộng 1,883,117 tham số, trong đó có 1,758,900 tham số từ lớp Embedding, 120,800 tham số từ lớp Bidirectional LSTM, và 3,417 tham số từ lớp TimeDistributed. Tất cả các tham số đều được huấn luyện (trainable params), tức là chúng sẽ được điều chỉnh trong quá trình huấn luyện. Mô hình có thể được tối ưu bằng thuật toán tối ưu hóa Adam với hàm mất mát là cross-entropy và các metric đánh giá là accuracy.



**d. Huấn luyện các mô hình**

Nhóm sử dụng cùng một tập dữ liệu và các tham số huấn luyện cho tất cả các mô hình để đảm bảo tính công bằng khi so sánh. Các tham số huấn luyện bao gồm số epochs: 3, batch size: 32, và tỷ lệ validation split: 0.1.

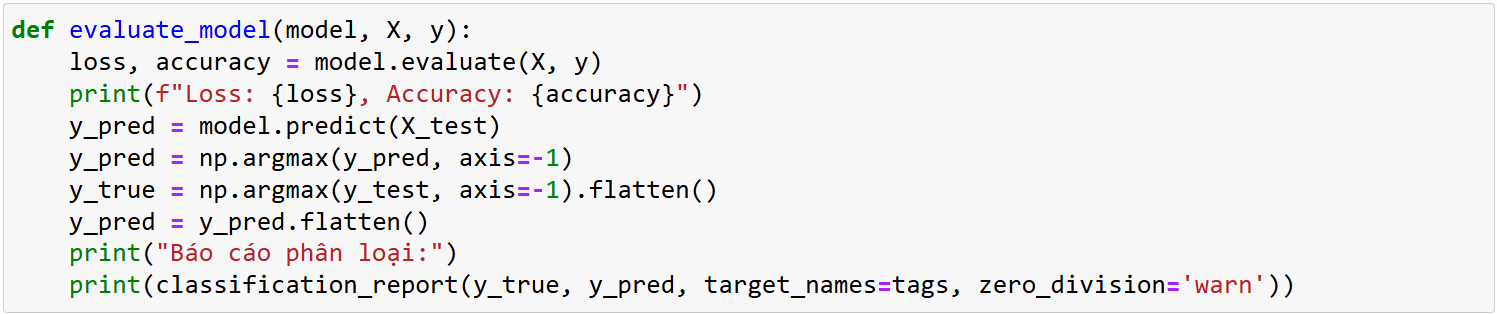


**e. Đánh giá các mô hình**

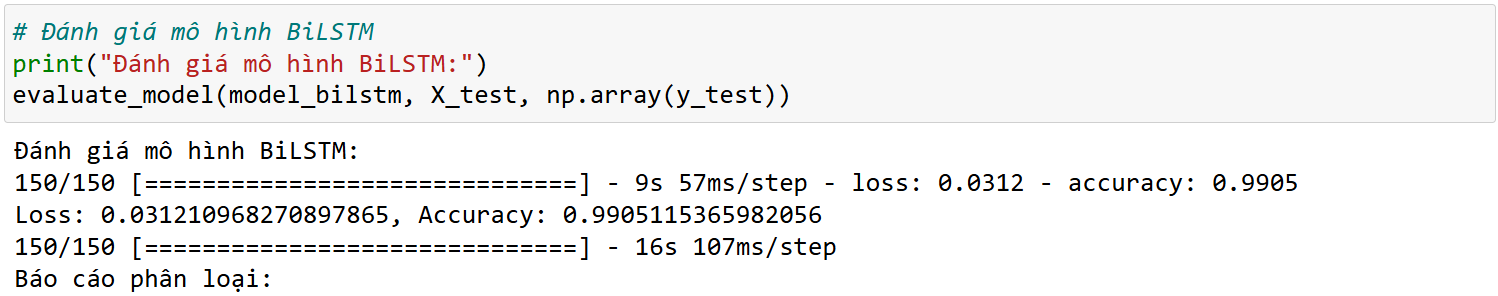
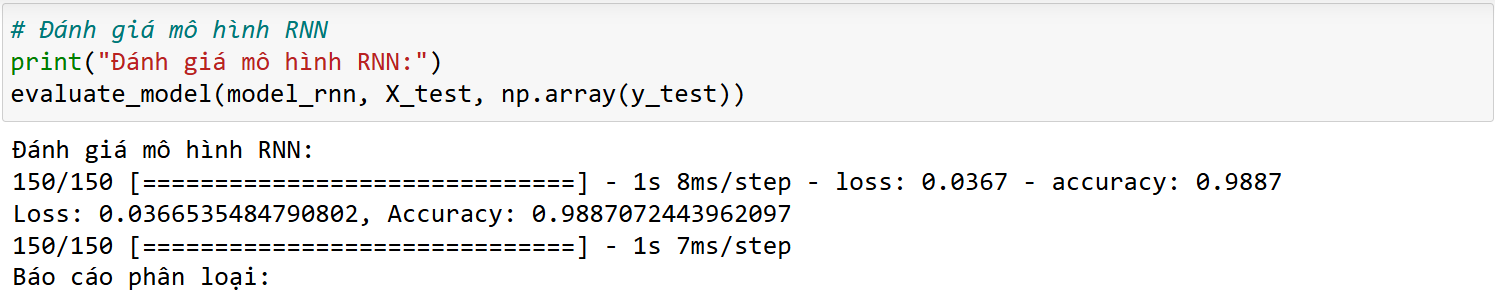
Xây dựng hàm evaluate\_model để đánh giá hiệu suất của mô hình NER thông qua các chỉ số như precision, recall và F1-score cho mỗi nhãn.

Hàm này nhận ba đối số đầu vào: “model” là mô hình cần được đánh giá, “X” là dữ liệu đầu vào và “y” là nhãn tương ứng với dữ liệu đầu vào.

Hàm trước tiên sử dụng phương thức evaluate của mô hình để tính loss và accuracy trên tập dữ liệu kiểm tra. Sau đó, nó sử dụng phương thức predict của mô hình để dự đoán nhãn cho tập dữ liệu kiểm tra. Kết quả dự đoán và nhãn thực tế được chuyển về dạng 1 chiều và sau đó in ra báo cáo phân loại sử dụng hàm classification\_report từ thư viện sklearn.metrics.

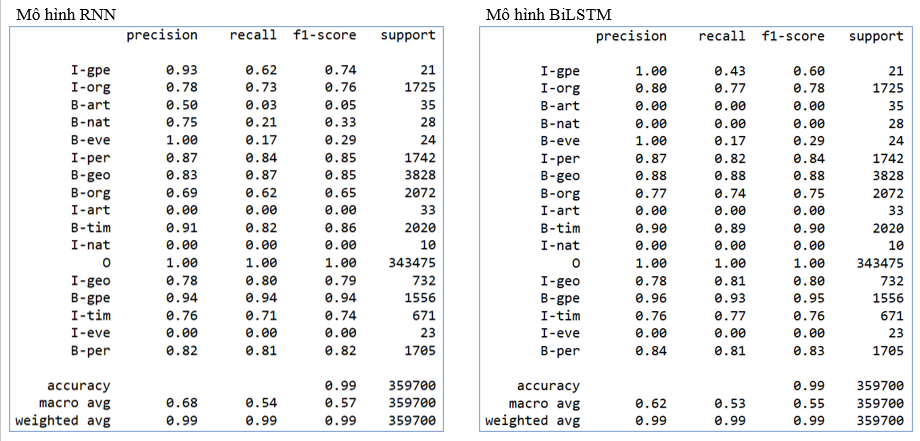


Kết quả đánh giá mô hình RNN và BiLSTM cho thấy cả hai mô hình đều có hiệu suất tốt với độ chính xác (accuracy) lớn hơn 99% trên tập dữ liệu kiểm tra.



Tuy nhiên, khi xem xét classification report, có một số điểm cần được lưu ý:

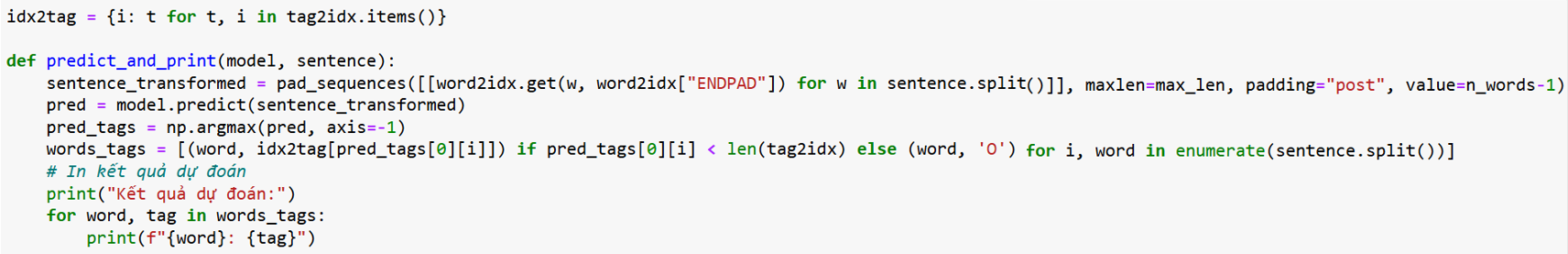
* **Với mô hình RNN:** Precision và recall cho một số nhãn như “B-art”, “B-nat”, “I-art”, “I-nat”, “I-eve” là 0 hoặc gần bằng 0, tức là mô hình không thể dự đoán hoặc không dự đoán đúng các nhãn này. Mặc dù có độ chính xác cao, nhưng mô hình có thể cần được điều chỉnh để cải thiện khả năng dự đoán cho các nhãn hiếm.
* **Với mô hình BiLSTM:** Precision và recall cho các nhãn “B-art”, “B-nat”, “I-art”, “I-nat”, “I-eve” cũng thấp hoặc gần bằng 0.Mặc dù có hiệu suất tương tự như mô hình RNN, nhưng BiLSTM có thể cung cấp thông tin hơn về các từ trong chuỗi nhờ khả năng học từ cả hai hướng.



**f. Dự đoán nhãn**

Xây dựng hàm predict\_and\_print để dự đoán nhãn cho các từ trong một câu và in ra kết quả dự đoán. Hàm này nhận hai đối số đầu vào: “model” là mô hình đã được huấn luyện để dự đoán nhãn cho các từ trong câu và “sentence” là câu cần dự đoán nhãn.

* Câu được chuyển đổi thành dạng số hóa bằng cách sử dụng từ điển word2idx, sau đó được thêm padding để có độ dài bằng max\_len.
* Mô hình được sử dụng để dự đoán nhãn cho câu văn đã được chuyển đổi.
* Kết quả dự đoán được chuyển thành các nhãn tương ứng với từng từ trong câu và được in ra màn hình.



Kết quả dự đoán từ hai mô hình RNN và BiLSTM cho câu “John Lee is the chief of CBSE, Americans suffered from H5N1 virus in 2002” là như sau:

