BÁO CÁO BÀI TẬP LỚN HỌC SÂU

Nhận diện thực thể có tên bằng RNN, BiRNN, CNN

Hồ Hà Ngọc Nhất * Vũ Đình Quang Huy * Hồ Cảnh Quyền *

Phạm Anh Quân * Nguyễn Hải Nam

Viện trí tuệ nhân tạo - UET - VNU

Abstract

Mạng nơ-ron hồi tiếp (RNN) đã đạt được nhiều tiến bộ đáng kể trong các bài toán mô hình hóa chuỗi dữ liệu thuộc nhiều lĩnh vực như xử lý ngôn ngữ tự nhiên và phân tích chuỗi thời gian. Báo cáo này tập trung nghiên cứu ba kiến trúc nổi bật: Đơn vị hồi tiếp có cổng (GRU), Bộ nhớ dài ngắn hạn (LSTM), Mạng nơ-ron hồi tiếp hai chiều (BiRNN) và Mạng nơron tích chập (CNN) kết hợp với BiRNN. Chúng tôi thực hiện so sánh kiến trúc, hiệu năng trên các bài toán chuỗi, cùng phân tích chi tiết quá trình huấn luyện. Kết quả của thực nghiệm chỉ ra rằng mạng RNN cơ bản GRU và LSTM mang lại kết quả không thực sự tốt cho bài toán nhận diện thực thể, BiRNN mang đến kết quả tốt hơn nhờ cơ chế học từ cả hai chiều thuận và nghịch nhưng thiếu khả năng học ngữ cảnh tốt. Ở phương pháp kết hợp Mạng CNN với Mạng BiRNN mang lại kết quả tối ưu nhất trong ba phương pháp đã nêu nhờ cơ chế trích xuất các đặc trưng quan trong trong văn bản. Quá trình nghiên cứu đánh ra được rằng, Với bộ dữ liệu CoNLL-2003, độ đo F1-score trên tập test với mô hình BLSTM-CNN là 88%

1 Introduction

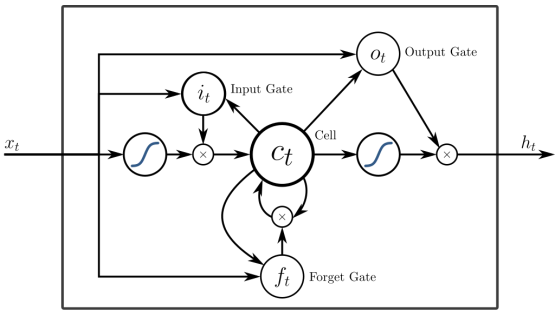
Nhận dạng thực thể (Named Entity Recognition- NER) là một bài toán trong Xử lý Ngôn ngữ Tự nhiên (NLP), nhằm mục tiêu xác định và phân loại các thực thể được đề cập trong văn bản vào các nhóm danh mục cụ thể. Các phương pháp tiếp cận hiệu suất cao đã được sử dụng như CRF, SVM. Tuy nhiên, các mô hình học máy truyền thống thường không thu được kết quả cao đối với dữ liệu phức tạp, các bộ dữ liệu có nhiều nhập nhằng vì ngữ nghĩa. Cùng với sự phát triển mạnh mẽ của GPU, các mô hình Deep Learning sử dụng mạng neural nhân tạo dần được đưa vào bài toán nhận dạng thực thể và nhanh chóng đem lại kết quả tốt với bộ dữ liệu lớn. Trong phạm vi bài viết này, chúng tôi sẽ sử dụng các mô hình Deep Learning như GRU, LSTM, CNN và các kỹ thuật deep learning như BiRNN để giải bài toán nhận diện thực thể này. Bộ dữ liệu chúng tôi sử dụng là CoNLL-2003 (Collobert and Weston, 2008; Mikolov et al., 2013).

2 Mô hình

Trong nội dung của bài viết, chúng tôi sẽ sử dụng 5 loại model Neural Network: LSTM, GRU, BLSTM, BGRU, BLSTM-CNN để so sánh cũng như đánh giá kết quả của các mô hình trên bộ dữ liệu CoNLL-2003.

2.1 LSTM

LSTM được phát triển để giải quyết vấn đề gradient biến mất (S. Hochreiter and J. Schmidhuber). Kiến trúc của nó gồm các cổng chính: Forget gate layer, Input gate layer, Output gate layer.

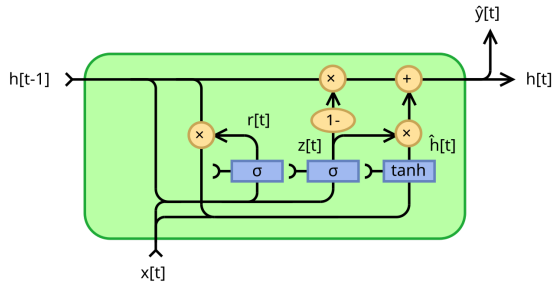


LSTM phù hợp cho các chuỗi dài như văn bản phức tạp nhờ:

- Có cơ chế bộ nhớ Cell State (được cập nhật qua ba cửa Forget, Input, và Output).
- Loại bỏ thông tin không quan trọng như Forget Gate.

2.2 GRU

GRU (R. Dey and F. M. Salem) là biến thể đơn giản hơn của LSTM, với hai cửa chính: Reset Gate và Update Gate.

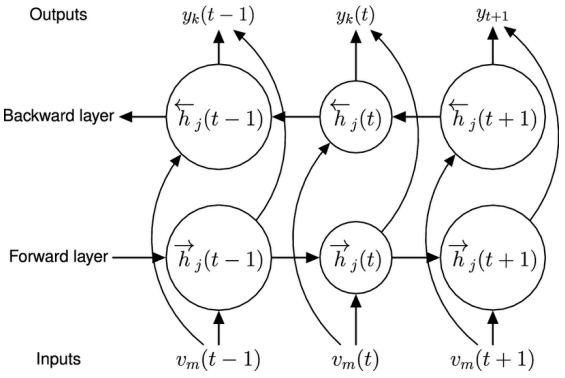


GRU xử lý chuỗi hiệu quả, đáp ứng nhằm lược yêu cầu của bài toán NER như:

- Xử lý ngữ cảnh ngắn.
- Hiệu quả tính toán cao hơn nhờ giảm số tham số

2.3 BiRNN

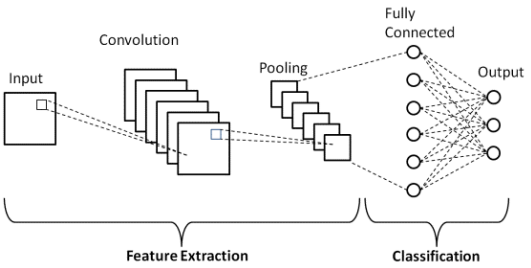
BiRNN (M. Schuster and K. K. Paliwal) là một kiến trúc mạng nơron được sử dụng cho sequential data.



Trong BiRNN, thông tin được xử lý theo cả 2 chiều thuận và nghịch, kĩ thuật này khiến dữ liệu được học thông tin ngữ cảnh ở cả hai chiều thuận và nghịch. Nhờ đó, BiRNN đem lại kết quả xuất sắc cho các bài toán như nhận diện thực thể có tên.

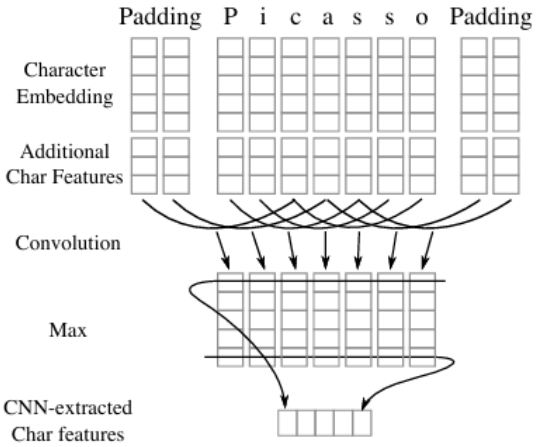
2.4 CNN + BiRNN

Mạng CNNs được cấu thành từ 4 lớp chính: Convolutional Layer, Activation Layer, Pooling Layer, Fully Connected Layer. Khi xếp chồng các lớp lên nhau, kiến trúc CNN được hình thành.



Trong NLP nói chung và bài toán NER nói riêng, CNN đóng vai trò quan trọng trong việc trích xuất các từ văn bản bằng cách sử dụng các filters để phát hiện các mẫu trong chuỗi từ (Jason P.C. Chiu and Eric Nichols).

Ý tưởng của mô hình là sử dụng thuật toán BiRNN kết hợp với CNN để đưa ra kết quả của bài toán NER.



Trong phương pháp này, CNN sử dụng các lớp tích chập để trích xuất các đặc trưng từ các vector nhúng. Tiếp theo sử dụng pooling để giảm kích thước và giữ lại các đặc trưng quan trọng của dữ liệu.

3 Evaluation

Đánh giá được thực hiện trên bộ dữ liệu CoNLL-2003 (Collobert and Weston, 2008; Mikolov et al., 2013).

Với mỗi thử nghiệm, chúng tôi đánh giá và đưa ra kết quả trung bình ba độ đo Precision, Recall, F1-score của 10 thử nghiệm có kết quả tối nhất và thu được bảng thống kê sau:

Model	Dev			Test		
	Prec.	Recall	F1	Prec.	Recall	F1
GRU	0.87	0.69	0.74	0.83	0.69	0.74
LSTM	0.87	0.81	0.82	0.82	0.77	0.79
BGRU	0.88	0.84	0.86	0.82	0.82	0.82
BLSTM	0.91	0.83	0.87	0.86	0.81	0.84
BGRU-CNN	0.92	0.89	0.91	0.86	0.88	0.87
BLSTM-CNN	**0.93**	**0.92**	**0.93**	**0.87**	**0.89**	**0.88**

Bảng 1: Kết quả trung bình chạy thử nghiệm các mô hình GRU, LSTM, BGRU, BLSTM, BGRU-CNN, BLSTM-CNN

3.1 Tiền xử lý dữ liệu

Trên toàn bộ tập dữ liệu, chúng tôi thực hiện lần lượt các bước tiền xử lý dữ liệu.

- Tách câu thành các token và đánh dấu các token là từ đặc biệt như số, chữ hoa,..
- Chia các từ thành các kí tự phục vụ mục đích làm đầu vào cho CNN.
- Trước khi huấn luyện, tập dữ liệu được xáo và chia làm các batch size nhỏ.

3.2 Dataset

Dataset CoNLL-2003 (Tjong Kim Sang và De Meulder, 2003) bao gồm các nhãn từ kho dữ liệu RCV1 của Reuters được gán nhãn 4 loại thực thể bao gồm: location, organization, person, and miscellaneous. Bộ dữ liệu đã được chia sẵn thành train, test, dev. Chúng tôi sử dụng tập train để train mô hình, dev để đánh giá và tập test để đưa ra kết quả thử nghiệm cuối cùng.

3.3 Tối ưu hoá siêu tham số

Với nhiều lần thử nghiệm khác nhau, chúng tôi tìm ra được bộ siêu tham số bao gồm learning rate, batch size, dropout,... tối ưu nhất cho từng mô hình.

3.4 Kết quả

Bảng 1 đã cho ta thấy kết quả của các thí nghiệm trên bộ dữ liệu ConNLL-2003. Thí nghiệm của chúng tôi đưa ra kết quả tốt nhất với chỉ số F1-score là 0.88 với mô hình BLSTM kết hợp với CNN.

4. Kết luận

Chúng tôi đã chứng minh rằng mô hình mạng nơ-ron của mình, kết hợp giữa LSTM hai chiều (BiLSTM) và mạng CNN giải quyết các kí tự, đạt được kết quả khá cao trong nhận dạng thực thể (NER). Mô hình của chúng tôi cải thiện so với các kết quả mô hình cũ khá nhiều về bài toán NER, cho thấy khả năng học các mối quan hệ phức tạp từ lượng dữ liệu lớn.

Trong tương lai, chúng tôi cũng muốn mở rộng mô hình để thực hiện các nhiệm vụ tương tự, chẳng hạn như mô hình nhận diện thực thể cho đa ngôn ngữ. Chúng tôi cũng có ý định sẽ sử dụng các mô hình Embedding linh hoạt hơn cũng như các mô hình pre-train như BERT để cải thiện khả năng học bài toán NER..

5. Tài liệu tham khảo

Ma, X. "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF." arXiv preprint arXiv:1603.01354 (2016).

Lample, Guillaume. "Neural architectures for named entity recognition." arXiv preprint arXiv:1603.01360 (2016).

Chiu, Jason PC, and Eric Nichols. "Named entity recognition with bidirectional LSTM-CNNs." Transactions of the association for computational linguistics 4 (2016): 357-370.

M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093.

R. Dey and F. M. Salem, "Gate-variants of Gated Recurrent Unit (GRU) neural networks," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 2017, pp. 1597-1600, doi: 10.1109/MWSCAS.2017.8053243

S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

Jacovi, Alon, Oren Sar Shalom, and Yoav Goldberg. "Understanding convolutional neural networks for text classification." arXiv preprint arXiv:1809.08037 (2018).

Link github dự án: [ngocnhatAI/Named-Entity-Recognition]