BÁO CÁO DỰ ÁN CUỐI KÌ HỌC SÂU
Setimentation Analysis with Transformer and Bi-RNN

Vũ Đình Thọ | Nguyễn Mạnh Hùng
Đinh Văn Sinh | Nguyễn Tuấn Thành | Nguyễn Công Thành
Viện Trí Tuệ Nhân Tạo, Đại học Công Nghệ, ĐHQGHN

Tóm tắt nội dung

Phân tích cảm xúc (Sentiment Analysis) là một ứng dụng phổ biến trong Xử lý ngôn ngữ tự nhiên (NLP), nhằm xác định cảm xúc (tích cực, tiêu cực, hoặc trung tính) trong văn bản. Dự án này tập trung triển khai và so sánh hai mô hình deep learning hiện đại: Transformer và Bi-RNN (Mạng nơ-ron hồi tiếp hai chiều), nhằm giải quyết bài toán phân tích cảm xúc trên các tập dữ liệu thực tế.
Dự án kỳ vọng sẽ góp phần nâng cao hiệu quả của các hệ thống phân tích cảm xúc hiện đại và cung cấp nền tảng khoa học cho việc ứng dụng các kỹ thuật NLP [4] tiên tiến vào thực tiễn.
Dự án này được phát triển bởi nhóm và mã nguồn được chia sẻ công khai trên GitHub. Bạn có thể tham khảo chi tiết mã nguồn và tài liệu tại `https://github.com/Liam-2603/segmentation-analysis`.

1 Giới thiệu

Mục Tiêu:

- Xây dựng và triển khai mô hình: Phát triển hai mô hình deep learning là Transformer và Bi-RNN để phân tích cảm xúc.
- Đánh giá và so sánh: So sánh hiệu suất của hai mô hình thông qua các chỉ số như độ chính xác (Accuracy), F1-Score.
- Ứng dụng thực tiễn: Ứng dụng kết quả phân tích vào các lĩnh vực như quản lý phản hồi khách hàng, theo dõi đánh giá sản phẩm hoặc giám sát cảm xúc trên mạng xã hội. Cụ thể trong dự án là đánh giá phim[2].

Phạm Vi Dự Án: Dự án sẽ tập trung triển khai và đánh giá hai mô hình Transformer và Bi-RNN trên bài toán phân tích cảm xúc. Dữ liệu sẽ được lấy từ các nguồn công khai.

Kết Quả Kỳ Vọng:

1. Xây dựng một hệ thống phân tích cảm xúc chính xác, hiệu quả, có thể áp dụng trong các bài toán thực tế.
2. Cung cấp cái nhìn sâu sắc về ưu điểm, nhược điểm của hai mô hình Transformer và Bi-RNN trong việc xử lý bài toán phân tích cảm xúc.
3. Tạo cơ sở cho các nghiên cứu hoặc ứng dụng mở rộng liên quan đến NLP trong tương lai.

2 Background about model

2.1 Sentiment Analysis

Phân loại cảm xúc bình luận [6] hiểu rõ hơn sự đón nhận của khán giả, hỗ trợ các nhà làm phim, nền tảng phát trực tuyến và nhà tiếp thị đưa ra quyết định (cắt, thêm suất chiếu, thực hiện các cơ hội hay chiến dịch nhanh chóng, đón đầu xu thế). Việc phân loại cảm xúc sẽ gặp 1 số vấn đề: ngôn ngữ đánh giá phim mang tính ẩn dụ cao, ngữ cảnh không rõ ràng, ...
Vì những vấn đề nêu trên, để phân loại 1 cách chính xác cần áp dụng các kỹ thuật Xử lý Ngôn ngữ Tự nhiên (NLP) [8] tốt nhằm nắm bắt ngôn ngữ tinh tế hơn.

2.2 RNN and Bi-RNN

2.2.1 RNN

RNN [1] là một loại mạng nơ-ron nhân tạo được thiết kế để xử lý dữ liệu theo chuỗi, rất phù hợp với xử lý chuỗi văn bản. Trong phân tích cảm xúc, RNN xử lý các bình luận phim theo từng từ, duy trì một trạng thái ẩn (hidden state) để nắm bắt ý nghĩa ngữ cảnh của các từ đã gặp trước đó. Từ đó tự tạo ngữ cảnh tuần tự để hiểu cảm xúc đa dạng không cứng nhắc mà mềm dẻo hơn.
Dù vậy, RNN truyền thống khó khăn khi nắm bắt các phụ thuộc xa (long-term dependencies), đặc biệt khi các từ quan trọng cách nhau một khoảng dài trong câu.

2.2.2 Bi-RNN

Bi-RNN khắc phục hạn chế này bằng cách xử lý chuỗi dữ liệu theo cả hai hướng: từ đầu đến cuối (forward) và từ cuối đến đầu (backward). Hiệu quả hơn trong phân tích cảm xúc vì hiểu cảm xúc của một bình luận thường đòi hỏi phải xét đến cả ngữ cảnh trước và sau.
Dù vậy, RNN và Bi-RNN vẫn gặp hạn chế về tốc độ huấn luyện chậm và vấn đề biến mất gradient khi xử lý các chuỗi dữ liệu dài như các bình luận chi tiết về phim.

2.3 Attention and Transformer

2.3.1 Attention

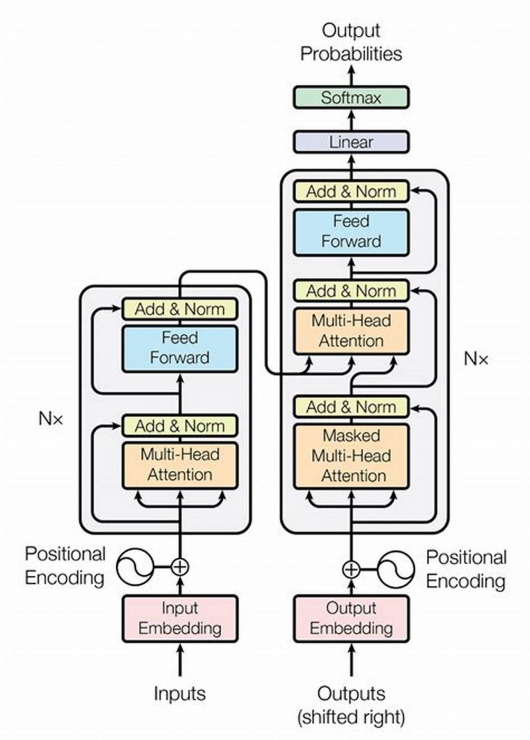
Cơ chế Attention cải thiện hiệu suất của các mô hình tuần tự (như RNN) bằng cách cho phép mô hình tập trung vào các phần quan trọng nhất của câu trong khi dự đoán cảm xúc. Thay vì dựa hoàn toàn vào trạng thái ẩn cuối cùng, Attention gán trọng số động cho từng từ trong câu dựa trên mức độ liên quan của chúng đến dự đoán đánh giá hiện tại.

2.3.2 Transformer

Mô hình Transformer được giới thiệu trong bài báo "Attention Is All You Need"[7] (Vaswani et al., 2017), đã cách mạng hóa NLP bằng cách thay thế hoàn toàn các lớp hồi tiếp (recurrent layers) bằng cơ chế self-attention. Điều này giúp mô hình xử lý toàn bộ chuỗi dữ liệu song song, nắm bắt các mối quan hệ toàn cục hiệu quả hơn.
Các đặc điểm chính của Transformer trong phân tích cảm xúc từ bình luận phim:
Khả năng mở rộng: Xử lý hiệu quả các bình luận dài (phổ biến trong đánh giá phim).

- Self-Attention: Mô hình hóa mối quan hệ giữa tất cả các từ trong bình luận, bất kể vị trí của chúng.

- Positional Encoding: Cung cấp thông tin về thứ tự từ trong chuỗi khi không có cơ chế hồi tiếp.
- Khả năng mở rộng: Xử lý hiệu quả các bình luận dài (phổ biến trong đánh giá phim).



Hình 1: The Transformer Model

3 Experiments

3.1 Data set

1. Làm sạch dữ liệu: Loại bỏ các câu trùng lặp hoặc có lỗi chính tả nghiêm trọng. => Thu được dataset bao gồm 50,000 câu đánh giá phim bằng tiếng việt , được sử dụng cho bài toán phân loại cảm xúc của câu có 2 nhãn là : positive(1) và negative(0)
2. Kích thước:
   - Train: 40,000 câu (80%)
   - Validation: 5,000 câu (10%)
   - Test: 5,000 câu (10%)
3. Tiền xử lý:
   - Tokenization sử dụng Phow(là pre train tokenization của PHOBERT) có vocab size lên đến 64,000 tokens
   - Áp dụng zero-padding cho các câu dưới 50 token.

3.2 Transformer

Ta sẽ tinh chỉnh và đánh giá 2 mô hình transformers chỉ sử dụng bộ Encode với tập dữ liệu trên cho bài toán phân loại cảm xúc. 2 mô hình được sử dụng là DistilBERT-base [5](SANH et al., 2020) và PhoBERT-base[3](Nguyen et al., 2020).
Phần cứng được sử dụng trong quá trình tinh chỉnh là 2 GPU T4 trên môi trường của Kaggle, các mô hình sẽ được đánh giá thông qua 2 thang đo là độ chính xác và điểm số F1 trên bộ validation. Cấu hình của 2 mô hình được mô tả trong ảnh sau:
Các mô hình được tinh chỉnh với learning rate là 2e-5 trong 5 epochs, sử dụng optimizer AdamW và batch size được tùy chỉnh theo như bảng dưới đây.
Sau khi tinh chỉnh, ta thấy với batch size nhỏ thì kết quả trả về chính xác hơn nhưng đồng thời thời gian huấn luyện cũng lâu hơn. Mô hình PhoBERT cho ra kết quả tốt hơn so với

```
{
  "activation": "gelu",
  "architectures": [
    "DistilBertForMaskedLM"
  ],
  "attention_dropout": 0.1,
  "dim": 768,
  "dropout": 0.1,
  "hidden_dim": 3072,
  "initializer_range": 0.02,
  "max_position_embeddings": 512,
  "model_type": "distilbert",
  "n_heads": 12,
  "n_layers": 6,
  "pad_token_id": 0,
  "qa_dropout": 0.1,
  "seq_classif_dropout": 0.2,
  "sinusoidal_pos_embds": false,
  "tie_weights_": true,
  "transformers_version": "4.10.0.dev0",
  "vocab_size": 30522
}
```

```
{
  "architectures": [
    "RobertaForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "eos_token_id": 2,
  "gradient_checkpointing": false,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-05,
  "max_position_embeddings": 258,
  "model_type": "roberta",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 1,
  "tokenizer_class": "PhobertTokenizer",
  "type_vocab_size": 1,
  "vocab_size": 64001
}
```


Hình 2: 1.1. Cấu hình distilBERT-base 1.2. Cấu hình PhoBERT-base

Models	Acc	F1-score	Train Runtime
distilBERT(batch$_size = 32$)	87.03	87.28	5543.02
PhoBERT(batch$_size = 32$)	90.7	90.93	5465.87
distilBERT(batch$_size = 64$)	86.59	87	5357.07
PhoBERT(batch$_size = 64$)	90.5	90.76	5048.47
distilBERT(batch$_size = 128$)	86,7	86.86	5150.61
PhoBERT(batch$_size = 128$)	90.09	90.39	4838.89

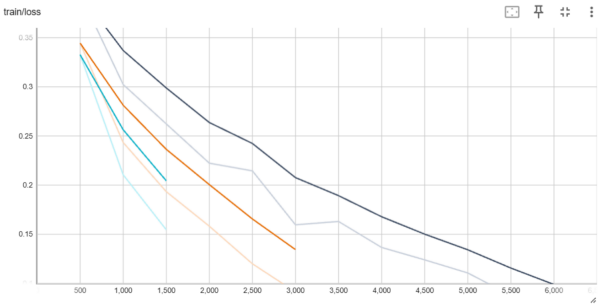
Bảng 1: Kết quả và thời gian chạy của các mô hình với batch size khác nhau, thời gian huấn luyện tính bằng giây

distilBERT là bởi PhoBERT là mô hình được pre-trained trên bộ dữ liệu tiếng Việt.Do hạn chế về tài nguyên huấn luyện nên ta chỉ có thể đánh giá 2 mô hình trên.

3.3 Bi-RNN

Tham số liên quan đến mô hình :

- Lớp embedding
  - input_dimension = vocab_size = 1585076
  - embedding_dimension = 100
- Drop out = 0.5
- Lớp hidden: sử dụng bidirectional



Hình 3: Giá trị loss của mô hình PhoBERT qua từng step (màu xanh là mô hình tinh chỉnh với batch$_size = 128, mucaml64, muxml32$)

  - Thử nghiệm trên 2 lớp LSTM[1] hoặc 3 lớp LSTM
  - hidden_dimension = 100

Tham số liên quan đến quá trình training :

- optimizer : thử nghiệm bằng Adam hoặc SGD
- batch size : 16, 64, 100
- momentum (đối với SGD) : 0,9
- learning rate :
  - đối với Adam : mặc định
  - đối với SGD : thử với các learning rate khác nhau (0,1 ; 0,01 ; 0,001)

Tiến hành training bằng x 2 GPU : T4 qua 5 epoch
Kết quả của việc tuning các hyperparameter :

Nhận xét:

- Khi tăng batch size lên dẫn đến thời gian training giảm

Optimizer	Batch size	Learning Rate	F1-Score	Accuracy	Training Time (s)
Adam	100	default	91.51%	91.75%	1765
	64		**92,56%**	**92,11%**	2786
	16		91,2%	90,31%	3200
SGD(momentum=0,9)	100	0,1	68.28%	69,58%	**1820**
	64	0,01	48.62%	49,52%	2826
	16	0,001	45,67%	46,52%	3360

Hình 4: Bi-RNN với 2 lớp LSTM

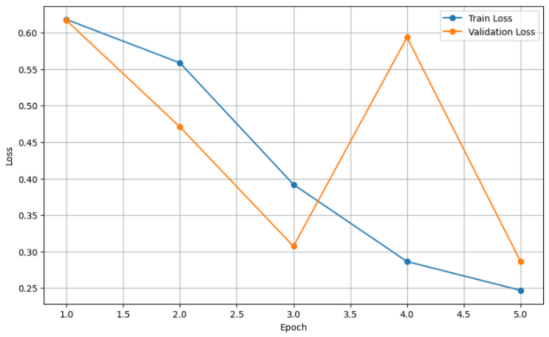
Optimizer	Batch size	Learning Rate	F1-Score	Accuracy	Training Time (s)
Adam	100	default	87,9%	87,74%	**1885**
	64		**95,23%**	**95,26%**	2906
	16		92,13%	91,71%	4478
SGD(momentum=0,9)	100	0,1	56,23%	57,02%	1960
	64	0,01	49,23%	50,49%	3125
	16	0,001	46,37%	49,51%	4698

Hình 5: Bi-RNN với 3 lớp LSTM

- Tăng learning rate thì acc của mô hình tăng lên (đối với optimizer bằng SGD) Phương pháp optimizer bằng Adam tốt hơn SGD

- Khi mô hình ở batch size = 64 cho ra kết quả tốt nhất so với các batch size khác

- Như vậy từ bảng trên có thể thấy rằng, mô hình có acc cao nhất là 95,09% khi mà mô hình có 3 lớp LSTM ở hidden layer , batch size = 64 và optimizer bằng adam



Hình 6: Giá trị loss tốt nhất của mô hình Bi-RNN (batch size = 64, optimizer bằng adam)

4 Result experiments

Mô hình	F1-score	Accuracy
Transformer	90,93 %	90,7 %
Bi-RNN	95,23%	95,26 %

Bảng 2: So sánh kết quả tốt nhất giữa Transformer và Bi-RNN

Bi-RNN vượt trội hơn Transformer trong cả F1-Score (95,14% so với 90,93%) và Accuracy (95,09% so với 90,7%), với mức chênh lệch khoảng 4%. Điều này cho thấy Bi-RNN phù hợp hơn với bài toán với bộ dữ liệu đã cho
Transformer có thể chưa phát huy tối đa tiềm năng, có thể do chưa được tối ưu siêu tham số hoặc dữ liệu không đủ lớn.

5 Limitations and future work

1. Hạn chế
    - Độ chính xác hạn chế: Mô hình có thể không đạt độ chính xác cao với các văn bản có cảm xúc phức tạp hoặc mơ hồ.
    - Sự phụ thuộc vào dữ liệu: Chất lượng dữ liệu đào tạo ảnh hưởng trực tiếp đến hiệu suất mô hình.
    - Khả năng tổng quát hóa: Mô hình có thể không hoạt động tốt với các ngôn ngữ hoặc lĩnh vực khác nhau.
    - Tính phức tạp của mô hình: Cấu trúc phức tạp của Transformer và Bi-RNN đòi hỏi nguồn lực tính toán lớn.
    - Khả năng giải thích: Mô hình khó giải thích kết quả phân tích cảm xúc.
2. Hướng phát triển tương lai
    - Cải thiện độ chính xác: Tích hợp kỹ thuật mới như Attention Mechanism cải tiến, sử dụng pre-trained language model hiệu suất cao hơn.
    - Mở rộng dữ liệu đào tạo: Tăng cường đa dạng dữ liệu, bao gồm các lĩnh vực và ngôn ngữ khác nhau.
    - Tối ưu hóa mô hình: Sử dụng kỹ thuật tối ưu hóa như quantization, pruning để giảm tính phức tạp.
    - Áp dụng trong các lĩnh vực mới: Xây dựng ứng dụng trong y tế, giáo dục, tài chính và các lĩnh vực khác.
    - Tích hợp đa mô hình: Kết hợp Transformer và Bi-RNN với các mô hình khác để cải thiện hiệu suất.

6 Conclusion

Trong bài báo cáo, chúng em đã thực hiện phân tích và so sánh hai mô hình deep learning hiện đại là Transformer (PhoBERT, DistilBERT) và Bi-RNN (với các lớp LSTM) trong bài toán phân tích cảm xúc dựa trên đánh giá phim bằng tiếng Việt. Kết luận chung:

1. PhoBERT (Transformer) với độ chính xác cao nhất 90.93% là lựa chọn phù hợp cho các bài toán xử

lý ngôn ngữ tiếng Việt nhờ khả năng xử lý toàn cục và hiệu suất cao, đặc biệt trong các tập dữ liệu phức tạp. Nhưng hạn chế khi thời gian huấn luyện của dài hơn so với Bi-RNN khi batch size tăng.

2. Bi-RNN (với LSTM) với 95.26% độ chính xác vẫn là một phương pháp mạnh mẽ và hiệu quả, đặc biệt khi tài nguyên huấn luyện hạn chế. Khó khăn khi chuỗi dài và có tốc độ huấn luyện chậm hơn khi số lượng layer tăng.

Kết quả mang lại thể hiện hiệu quả của các mô hình deep learning trong bài toán phân tích cảm xúc và góp phần đóng góp thực tiễn cho phát triển các ứng dụng NLP.

Tài liệu

[1] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[2] Nhan Cach Dang, María N. Moreno-García, and Fernando De la Prieta. Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3), 2020.

[3] datquocnguyen. Phow2v. `https://github.com/datquocnguyen/PhoW2V`.

[4] Christopher Manning, Richard Socher, and other faculty members. Cs224n: Natural language processing with deep learning. `https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1214/`, 2014. Truy cập ngày 15 tháng 12, 2024.

[5] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

[6] Ben Trevett. pytorch-sentiment-analysis. `https://github.com/bentrevett/pytorch-sentiment-analysis`.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia

Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[8] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2024.