Contrastive Learning cho nhiệm vụ Sentiment Analysis trên mô hình MinBERT

Bùi Quang Tùng
Khoa Công nghệ thông tin, UET
Email: 22028234@vnu.edu.vn

Tóm tắt nội dung

Bài báo này đề xuất một phương pháp cải thiện hiệu suất phân loại cảm xúc (Sentiment Analysis) trên tập dữ liệu SST-2, sử dụng mô hình TinyBERT, một biến thể nhỏ gọn của BERT với 4 lớp transformer, kết hợp với kỹ thuật Contrastive Learning. Để giải quyết vấn đề hạn chế về khả năng biểu diễn đặc trưng của mô hình nhỏ, bài báo này áp dụng các kỹ thuật tăng cường dữ liệu sử dụng cho huấn luyện Contrastive như thay thế từ đồng nghĩa và xóa từ. Một lớp projection được thêm vào mô hình TinyBERT, kết hợp với hàm mất mát InfoNCE để tối ưu mô hình. Kết quả thực nghiệm cho thấy các kỹ thuật này cải thiện đáng kể độ chính xác, đạt hiệu suất gần tương đương với các mô hình lớn hơn. Nghiên cứu này nhấn mạnh tiềm năng của việc kết hợp các mô hình ngôn ngữ với Constrastive Learning để thực hiện task phân tích cảm xúc hiệu quả hơn về mặt tài nguyên.
Keywords: Sentiment Analysis, Contrastive Learning, TinyBERT, Data Augmentation, InfoNCE Loss.

I. Introduction

Phân tích cảm xúc (Sentiment Analysis) hay phân loại cảm xúc (Sentiment Classification) là một bài toán quan trọng và nổi tiếng trong lĩnh vực xử lý ngôn ngữ tự nhiên, với nhiều ứng dụng trong kinh doanh, mạng xã hội, và dịch vụ khách hàng. Mô hình Transformer-based như là BERT đã đạt được độ chính xác khá cao trong task này. Mô hình BERT đầy đủ gồm khá nhiều tham số (mô hình khá phức tạp), do đó các mô hình MinBERT được tinh chỉnh từ BERT xuất hiện. MinBERT là phiên bản nhỏ gọn của mô hình BERT, được tối ưu về tài nguyên và hiệu suất nhưng vẫn giữ được những ưu điểm mạnh mẽ của BERT. Trong bài toán phân tích cảm xúc, MinBERT thể hiện khả năng phân loại văn bản khá tốt. Tuy nhiên, vẫn có thể áp dụng các phương pháp cải tiến để nâng cao hiệu năng của MinBERT, cụ thể như huấn luyện Contrastive Learning, áp dụng kỹ thuật Paraphasing và kỹ thuật Knowledge injection. Trong bài báo này, tôi đề xuất sử dụng Contrastive Learning nhằm cải thiện hiệu suất của mô hình minBERT trong Sentiment Analysis.

II. Related Work

A. Knowledge Distillation

Knowledge Distillation (KD), lần đầu tiên được giới thiệu bởi Hinton et al. [1], là một phương pháp huấn luyện cho phép một mô hình lớn (teacher model) chuyển giao kiến thức cho một mô hình nhỏ hơn (student model). Thông qua việc học từ các phân phối xác suất mềm (soft labels) của teacher model, mô hình học sinh có thể đạt được hiệu suất gần tương đương với mô hình lớn, trong khi giảm đáng kể yêu cầu về bộ nhớ và tốc độ tính toán. KD đã được áp dụng thành

công trong việc chuyển giao kiến thức từ mô hình BERT đầy đủ sang các mô hình nhỏ gọn hơn, bao gồm TinyBERT [6] DistilBERT [3] và MiniLM [8]. Trong nghiên cứu của Tang et al. [2], kỹ thuật này đã được chứng minh là có khả năng cải thiện hiệu suất của các mô hình thu nhỏ mà vẫn duy trì tính tổng quát.

B. MinBERT

MinBERT là một mô hình thu gọn của BERT (Bidirectional Encoder Representations from Transformers), được thiết kế để giảm thiểu tài nguyên tính toán nhưng vẫn đảm bảo hiệu quả. Được giới thiệu lần đầu trong nghiên cứu của Sanh et al. [3] về mô hình DistilBERT, các kỹ thuật như giảm số lượng tầng (layers) và tham số đã được sử dụng để tối ưu hóa MinBERT. Theo các nghiên cứu gần đây như TinyBERT [6] và ALBERT [7], các mô hình MinBERT không chỉ giảm kích thước và độ phức tạp của mô hình mà còn có hiệu suất khá tốt trong các task NLP, gần tiệm cận với với mô hình BERT đầy đủ. Tuy nhiên, với các bài toán yêu cầu sự tinh vi như phân tích cảm xúc, MinBERT có thể gặp khó khăn trong việc nắm bắt đầy đủ các sắc thái cảm xúc phức tạp. Do đó một số kỹ thuật nâng cao có tiềm năng sẽ cải thiện khả năng của mô hình khi được áp dụng chúng trong quá trình huấn luyện.

C. Stanford Sentiment Treebank (SST và SST-2)

Stanford Sentiment Treebank [11] là một tập dữ liệu chuẩn được sử dụng rộng rãi trong nhiệm vụ phân tích cảm xúc. Trong đó, tập dữ liệu SST-2 tập trung vào phân loại nhị phân, cung cấp các nhãn cảm xúc nhị phân (positive, negative) cho các đoạn văn bản tiếng Anh, cụ thể là các câu đánh giá về phim. Tập dữ liệu này đã được tích hợp trong nhiều benchmark nổi tiếng như GLUE [12], tạo thành một bài toán chuẩn để đánh giá hiệu suất của các mô hình NLP hiện đại, bao gồm cả BERT [13] và các phiên bản thu gọn của nó.

Trong nghiên cứu này, SST-2 được sử dụng để đánh giá hiệu suất mô hình TinyBERT sau khi áp dụng kỹ thuật Contrastive Learning. Mặc dù SST-2 là một tập dữ liệu có cấu trúc tốt và cân bằng, các thách thức như phân biệt sắc thái cảm xúc trong ngữ cảnh cụ thể vẫn đòi hỏi các phương pháp tiên tiến để đạt được kết quả tốt nhất.

D. Contrastive Learning Method

Contrastive Learning đã trở thành một phương pháp phổ biến trong học sâu, đặc biệt với các ứng dụng học biểu diễn (representation learning). Phương pháp này được biết đến rộng rãi sau khi Chen và các cộng sự [5] giới thiệu framework SimCLR, thường được áp dụng trong lĩnh vực Computer Vision, trong đó các cặp mẫu dương (positive pairs) được kéo gần nhau và các cặp mẫu âm (negative pairs) bị đẩy xa ra trong không gian biểu diễn. Trong lĩnh vực NLP, Gao và các cộng sự [4] đã áp dụng thành công Contrastive Learning vào việc học biểu diễn ngữ nghĩa với framework SimCSE, cải thiện đáng kể hiệu suất của các mô hình Transformer trên các bài toán không giám sát. Một số nghiên cứu khác như ConSERT [9] đã tích hợp Contrastive Learning với BERT để xử lý các nhiệm vụ phân loại văn bản, đạt được hiệu quả cao hơn trong việc phân tách các nhãn gần giống nhau. Việc áp dụng Contrastive Learning trong nghiên cứu này nhằm tăng cường khả năng phân biệt giữa các nhãn cảm xúc của mô hình TinyBERT.

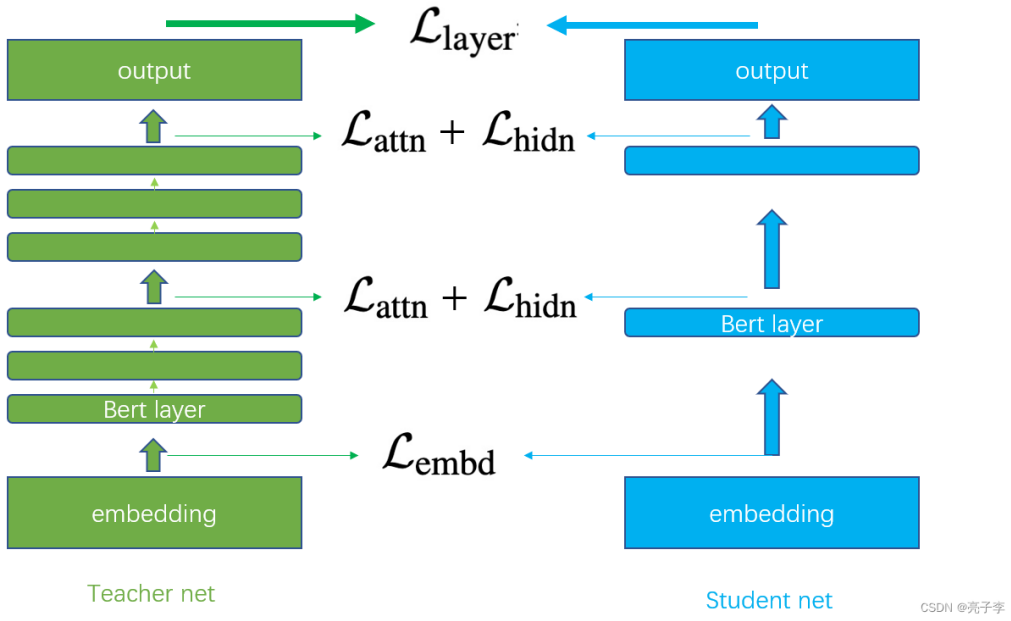
III. Model

A. *Tổng quan về TinyBERT*

Nghiên cứu này sử dụng TinyBERT$_4$ làm mô hình cơ sở cho nhiệm vụ phân tích cảm xúc trên tập dữ liệu SST-2. TinyBERT là một phiên bản thu gọn của BERT, được thiết kế để giảm thiểu kích thước và yêu cầu tính toán mà vẫn giữ được hiệu suất cao. TinyBERT đạt được điều này thông qua kỹ thuật Knowledge Distillation (KD), trong đó một mô hình lớn (teacher model) truyền tri thức cho một mô hình nhỏ hơn (student model).

Với cấu trúc gồm 4 tầng Transformer, TinyBERT$_4$ có số lượng parameter là 14.5M (so với 109M parameters của BERT base), chiều vector ẩn $d_{hid} = 312$ và kích thước feed-forward là $d_i = 1200$. Token đặc biệt `[CLS]` từ tầng cuối cùng được sử dụng để dự đoán nhãn cảm xúc qua một lớp Dense ở đầu ra.

Quá trình huấn luyện TinyBERT bao gồm hai giai đoạn. Giai đoạn đầu tiên là *General Distillation*, trong đó mô hình học các biểu diễn ngữ nghĩa chung từ BERT bằng cách sử dụng dữ liệu tiền huấn luyện. Giai đoạn thứ hai là *Task-Specific Distillation*, TinyBERT sẽ được tinh chỉnh trên tập dữ liệu của một task cụ thể với sự hỗ trợ từ các soft labels của teacher model.



Hình 1. Quá trình học của TinyBERT với Embedding-layer Distillation và Transformer Distillation.

Trong toàn bộ quá trình, hàm lỗi kết hợp giữa các layer được sử dụng để tối ưu hóa mô hình. Hàm loss tổng hợp cho quá trình Knowledge Distillation của TinyBERT:

$$\mathcal{L}_{\text{layer}} = \begin{cases} \mathcal{L}_{\text{embd}}, & m = 0, \\ \mathcal{L}_{\text{hidn}} + \mathcal{L}_{\text{attn}}, & M \geq m > 0, \\ \mathcal{L}_{\text{pred}}, & m = M + 1, \end{cases} \tag{1}$$

Trong đó:

$\mathcal{L}_{\text{embd}}$: Loss tại embedding layer,

$\mathcal{L}_{\text{hidn}}$: Loss tại các hidden layer,

$\mathcal{L}_{\text{attn}}$: Loss tại các attention layer,

$\mathcal{L}_{\text{pred}}$: Loss dự đoán đầu ra (output prediction) tại layer cuối cùng (m = M + 1),

m : Chỉ số layer hiện tại,

M : Tổng số layer trong mô hình.

B. Kết hợp Contrastive Learning với task phân loại cảm xúc

Trong bài toán phân tích cảm xúc nhị phân, Contrastive Learning giúp mô hình phân biệt các cảm xúc tích cực, tiêu cực tốt hơn bằng cách làm nổi bật sự khác biệt giữa các văn bản có nội dung tương tự (mẫu dương) và trái ngược (mẫu âm). Mô hình có thể học được các đặc điểm quan trọng mà không cần phải gắn nhãn cho các mẫu trong tập huấn luyện.

Để áp dụng Contrastive Learning với tập dữ liệu SST-2, nghiên cứu này sử dụng các kỹ thuật augmentation cho dữ liệu văn bản để tạo ra các mẫu dương ứng với từng dữ liệu. Các kỹ thuật augmentation sử dụng trong nghiên cứu này bao gồm:

- **Thay thế từ đồng nghĩa:** Một số từ ngẫu nhiên trong câu được thay thế bằng các từ đồng nghĩa của nó.
- **Xóa từ:** Một từ ngẫu nhiên trong câu được loại bỏ. Điều này giúp mô hình học cách xử lý và duy trì nghĩa của câu mặc dù có sự mất mát thông tin.

Trong khi đó, các mẫu dữ liệu âm là các mẫu còn lại trong batch. Ví dụ, với batch size là 32 thì ta có 31 mẫu âm và một cặp dữ liệu dương.

Hàm lỗi: InfoNCE Loss

Để tối ưu hóa việc học trong Contrastive Learning, chúng tôi sử dụng hàm lỗi InfoNCE (Information Noise Contrastive Estimation) [15], một hàm lỗi phổ biến trong các bài toán học tương phản. Hàm lỗi InfoNCE được định nghĩa:

$$\mathcal{L}_{InfoNCE} = -\log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)}{\sum_{k=1,k\neq i}^{K} \exp(\mathbf{z}_i \cdot \mathbf{z}_k/\tau)}$$

Trong đó:

$\mathbf{z}_i$ là biểu diễn của dữ liệu gốc,

$\mathbf{z}_j$ là biểu diễn của mẫu dữ liệu dương ứng với dữ liệu gốc,

$\mathbf{z}_k$ là các biểu diễn của các mẫu dữ liệu còn lại,

τ(temperature) là tham số điều chỉnh độ nhạy của hàm loss,

K là số mẫu dữ liệu trong batch.

Hàm lỗi này giúp mô hình tối thiểu hóa khoảng cách giữa các cặp dữ liệu dương trong không gian nhúng, đồng thời tối đa hóa khoảng cách giữa các cặp dữ liệu âm. Mục tiêu là làm cho

các cặp dữ liệu dương trở nên gần nhau, trong khi các cặp dữ liệu âm được đẩy ra xa, giúp cho mô hình học được các sắc thái cảm xúc tốt hơn.

Hàm lỗi của mô hình trong quá trình huấn luyện là tổng 2 hàm lỗi InfoNCE Loss và CrossEntropy Loss (phân loại 2 lớp positive và negative):

$$\mathcal{L} = \lambda \mathcal{L}_{InfoNCE} + \mathcal{L}_{CE}$$

với λ là trọng số của hàm lỗi InfoNCE Loss.

Để áp dụng Contrastive Learning, nghiên cứu này thực hiện một số thay đổi trong cấu trúc mô hình, bao gồm việc thêm một lớp projection vào mô hình. Cụ thể, sau khi TinyBERT xử lý dữ liệu đầu vào qua các lớp Transformer, một lớp projection (kích thước bằng 312 tương đương với chiều vector ẩn của TinyBERT) được thêm vào song song với lớp classifier.

Lớp projection này giúp biến đổi biểu diễn đầu ra của TinyBERT thành không gian nhúng mới với số chiều bằng số chiều của lớp ẩn cuối cùng, sau đó đưa 2 projection output của mẫu gốc và mẫu dương của nó vào hàm InfoNCE Loss để huấn luyện Contrastive. Điều này giúp mô hình nhận biết các sắc thái cảm xúc của dữ liệu tốt hơn, qua đó giúp TinyBERT trả về lớp ẩn đầu ra để đưa vào lớp classifier chất lượng hơn.

C. Khái quát quá trình tiến hành huấn luyện:

1) Chuẩn bị dữ liệu huấn luyện (chia dữ liệu, tokenize dữ liệu, áp dụng các kỹ thuật augmentation, tạo dataloader)
2) Khởi tạo mô hình pretrained, định nghĩa mô hình của nghiên cứu.
3) Khởi tạo hàm lỗi, optimizer, scheduler,...
4) Huấn luyện mô hình.
5) Đánh giá mô hình ứng với những cài đặt thí nghiệm khác nhau.

IV. Data

Dữ liệu sử dụng trong nghiên cứu được lấy từ nguồn công khai là tập dữ liệu SST-2 (Stanford Sentiment Treebank 2), một phiên bản rút gọn của tập dữ liệu SST, tập trung vào bài toán phân loại cảm xúc nhị phân (Binary Sentiment Classification). SST-2 bao gồm các câu hoặc đoạn văn bản ngắn được trích từ các bài đánh giá phim, với nhãn cảm xúc được phân loại thành hai loại: tích cực (*nhãn 1*) và tiêu cực (*nhãn 0*).

Tập dữ liệu SST-2 đã được chia sẵn thành ba phần chính:

- **Tập huấn luyện (Train):** Gồm 6,920 mẫu, được sử dụng để huấn luyện mô hình.
- **Tập kiểm định (Validation):** Gồm 872 mẫu, được sử dụng để điều chỉnh siêu tham số và đánh giá hiệu suất mô hình trên dữ liệu không tham gia huấn luyện.
- **Tập kiểm tra (Test):** Gồm 1,821 mẫu, được sử dụng để đánh giá cuối cùng hiệu suất của mô hình.

Mỗi mẫu trong tập dữ liệu bao gồm một câu văn bản và nhãn cảm xúc tương ứng. Các câu trong tập dữ liệu có độ dài trung bình ngắn, được thiết kế phù hợp để đánh giá khả năng phân tích cảm xúc của các mô hình học sâu.

Việc sử dụng SST-2 trong nghiên cứu không chỉ đảm bảo tính khách quan khi so sánh với các mô hình khác, mà còn tận dụng được một tập dữ liệu opensource chuẩn trong lĩnh vực xử lý ngôn ngữ tự nhiên.

https://github.com/YJiangcm/SST-2-sentiment-analysis/tree/master/data.

V. Experiments

A. Cài đặt mô hình

Các thí nghiệm được thực hiện trên nền tảng Google Colab, sử dụng cấu hình phần cứng tiêu chuẩn gồm GPU NVIDIA Tesla T4 với bộ nhớ VRAM từ 12GB đến 16GB, RAM 12.7GB, và Disk 112.6GB dung lượng lưu trữ tạm thời. Môi trường này cung cấp đủ khả năng tính toán để huấn luyện mô hình nhẹ như TinyBERT.

Trong thí nghiệm này tôi chia lại tập dữ liệu SST-2 với tập train gồm 7820 mẫu (900 mẫu được lấy từ tập test), tập validation gồm 872 mẫu và tập test gồm 921 mẫu. Bên cạnh đó, thí nghiệm sẽ bao gồm 3 kiểu sinh dữ liệu mẫu dương: "delete one word", "replace with synonym"và mixed cả 2 kỹ thuật trên. Kỹ thuật xóa một từ sẽ xóa ngẫu nhiên một từ trong câu, trong khi đó kỹ thuật thay thế từ đồng nghĩa sẽ thay các từ trong câu với xác suất một từ bị thay thế là 0.6. Kiểu mixed sẽ chọn ngẫu nhiên 1 trong 2 kỹ thuật trên với mỗi câu trong tập dữ liệu. B

Mô hình sử dụng chia thành 2 phần, gồm phần Transformer (sử dụng BERTModel) và phần Dense layer. Phần Transfomer là mô hình TinyBert pretrained: *TinyBERT-General-4L-312D* được tải về từ Huggingface. Phần Dense gồm 2 lớp cùng lấy đầu ra ẩn cuối cùng của mạng Transformer làm đầu vào là lớp projection (dùng để huấn luyện Contrastive) và lớp classifier (dùng để phân lớp). Trong 2 lớp trên, dropout với tỷ lệ 0.2 được sử dụng nhằm giảm thiểu hiện tượng overfitting. Mô hình sử dụng hàm lỗi InfoNCE với tham số temperature τ = 0.05, với trọng số $\lambda = 1$ trong công thức tính tổng lỗi của mô hình. Hàm CrossEntropy Loss với output đầu ra có 2 logits tương ứng với nhãn âm và nhãn dương được sử dụng làm hàm lỗi phân loại.

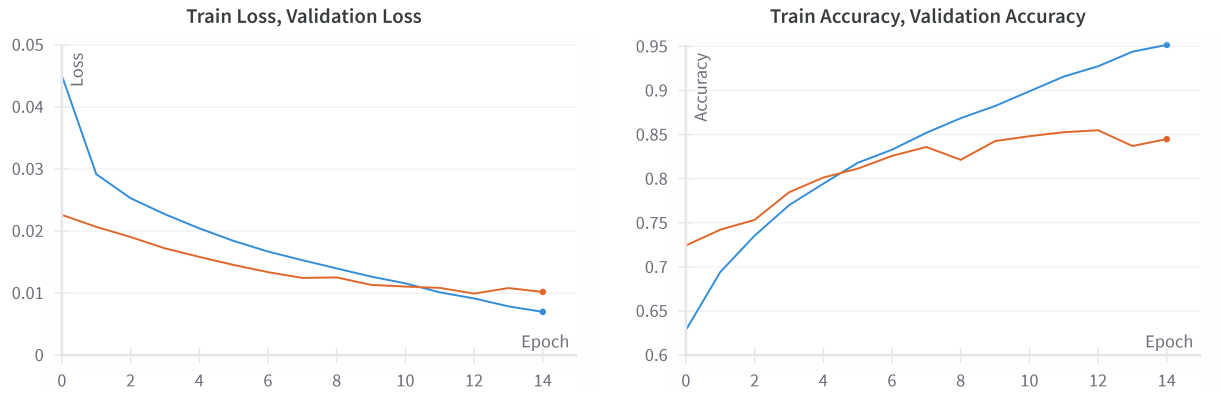
Quá trình huấn luyện được thực hiện với batch size là 32, thuật toán tối ưu `AdamW`, với tốc độ học được thiết lập ban đầu là 2×10^{-5}, kết hợp với một scheduler cho learning rate `StepLR`, giảm tốc độ học đi một nửa sau mỗi hai epoch.

B. Đánh giá hiệu suất

Hiệu suất của mô hình được đánh giá dựa trên bốn chỉ số chính:

- **Accuracy:** Đo lường tỷ lệ mẫu được dự đoán chính xác so với tổng số mẫu.
- **Precision:** Đo lường tỷ lệ dự đoán đúng trong số các mẫu được dự đoán là tích cực.
- **Recall:** Đo lường tỷ lệ dự đoán đúng trong số các mẫu thực sự mang nhãn tích cực.
- **F1-Score:** Là trung bình điều hòa giữa Precision và Recall, phản ánh sự cân bằng giữa hai chỉ số này.

Tập dữ liệu test được sử dụng để đánh giá mô hình sau khi quá trình huấn luyện và tinh chỉnh được hoàn tất. Các thông số và kết quả được ghi nhận để phân tích, từ đó đánh giá hiệu quả của phương pháp cũng như so sánh với mô hình không áp dụng Contrastive Learning.



(a) Loss (b) Accuracy

Hình 2. Quá trình huấn luyện với setting kỹ thuật synonym augmentation

VI. Results

Mô hình được đánh giá trên tập dữ liệu test với các chỉ số chính: **Accuracy**, **Precision**, **Recall**, và **F1-Score**. Kết quả cho thấy phương pháp học tương phản kết hợp với TinyBERT đã cải thiện tương đối hiệu suất trong việc phân loại cảm xúc nhị phân so với mô hình ban đầu (chưa huấn luyện Contrastive).

Bảng I
KẾT QUẢ TRÊN TẬP KIỂM TRA

Experiment	Accuracy	Precision	Recall	F1-Score
$\text{BERT}_{\text{BASE}}$	90.8	90.8	90.8	90.8
$\text{TinyBERT}_{\text{BASE}}$	87.8	87.8	87.8	87.8
Delete one word	89.4	89.7	89.4	89.5
Synonym	90.0	90.1	90.0	90.0
Mixed	89.5	89.7	89.5	89.6

VII. Analysis and Discussion

Kết quả thực nghiệm được trình bày trong Bảng I cho thấy hiệu suất của các mô hình trên tập kiểm tra SST-2 khi áp dụng các phương pháp khác nhau. Các phân tích cụ thể như sau:

A. Ảnh hưởng của Contrastive Learning

Khi áp dụng kỹ thuật Contrastive Learning với dữ liệu được augment (Delete one word, Synonym, và Mixed), hiệu suất của $TinyBERT_{BASE}$ được cải thiện tương đối. Cụ thể:

- **Phương pháp Delete one word:** Đạt Accuracy 89.4%, cải thiện 1.6% so với TinyBERT gốc. Điều này cho thấy việc xóa từ giúp tăng cường khả năng học biểu diễn từ các ngữ cảnh thiếu hụt.

- **Phương pháp Synonym:** Đạt Accuracy cao nhất 90.0%, chứng tỏ việc thay thế từ đồng nghĩa tạo ra những mẫu phong phú mà vẫn giữ khá sát ý nghĩa ban đầu, giúp mô hình học tốt hơn.
- **Phương pháp Mixed:** Kết hợp cả hai kỹ thuật augment trên, đạt Accuracy 89.5%, cao hơn một chút với phương pháp chỉ dùng Delete one word nhưng không quá đáng kể.

B. So sánh các kỹ thuật augmentation

Phương pháp Delete one word và Mixed tuy cải thiện hiệu suất so với mô hình gốc nhưng vẫn thấp hơn Synonym. Điều này có thể được lý giải rằng việc xóa từ trong câu có thể làm mất một phần ý nghĩa ban đầu, dẫn đến sự thay đổi trong ngữ cảnh, từ đó khiến mô hình khó học hơn. Các kết quả trên chỉ ra rằng việc áp dụng Unsupervised Contrastive Learning đã cải thiện tương đối đáng kể hiệu suất của mô hình khi kết hợp với các kỹ thuật augmentation hợp lý.

VIII. Conclusion

Trong bài báo này, tôi đã đề xuất và thực hiện một phương pháp kết hợp mô hình TinyBERT với kỹ thuật Contrastive Learning nhằm cải thiện hiệu suất phân loại cảm xúc trên tập dữ liệu SST-2. Bằng cách sử dụng các kỹ thuật tăng cường dữ liệu như thay thế từ đồng nghĩa và xóa từ, cùng với việc áp dụng hàm mất mát InfoNCE và thêm một lớp projection vào mô hình, nghiên cứu đã thành công trong việc cải thiện độ chính xác của mô hình. Kết quả thực nghiệm cho thấy phương pháp đề xuất không chỉ cải thiện hiệu suất của TinyBERT mà còn đạt được hiệu quả gần tương đương với các mô hình lớn hơn như $\text{BERT}_{\text{BASE}}$.

Nghiên cứu này khẳng định tính hiệu quả của việc kết hợp học tương phản trong quá trình huấn luyện trong các bài toán phân loại cảm xúc. Trong tương lai, tôi dự kiến mở rộng nghiên cứu bằng cách áp dụng phương pháp này trên các tập dữ liệu lớn hơn và đa dạng hơn, sử dụng các phương pháp data augmentation đa dạng hơn, cũng như thử nghiệm với các biến thể khác của MinBERT để đánh giá tính tổng quát của phương pháp.

References

Tài liệu

[1] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean, *Distilling the Knowledge in a Neural Network*, arXiv preprint arXiv:1503.02531, 2015.

[2] Tang, Raphael, et al., *Distilling Task-Specific Knowledge from BERT into Simple Neural Networks*, arXiv preprint arXiv:1903.12136, 2019.

[3] Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf, *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*, arXiv preprint arXiv:1910.01108, 2019.

[4] Gao, Tianyu, Xingcheng Yao, and Danqi Chen, *SimCSE: Simple Contrastive Learning of Sentence Embeddings*, Proceedings of EMNLP 2021, 2021.

[5] Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, *A Simple Framework for Contrastive Learning of Visual Representations*, International Conference on Machine Learning (ICML), 2020.

[6] Jiao, Xiaoqi, et al., *TinyBERT: Distilling BERT for natural language understanding*, arXiv preprint arXiv:1909.10351.

[7] Lan, Zhenzhong, et al., *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*, International Conference on Learning Representations (ICLR), 2020.

[8] Wang, Wenhui, et al., *MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers*, NeurIPS, 2020.

[9] Yan, Yanzhu, et al., *ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer*, ACL 2021.

[10] Sun, Chi, et al., *How to Fine-Tune BERT for Text Classification?*, CCF NLPCC, 2019.

[11] Socher, Richard, et al., *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank*, EMNLP 2013, 2013.

[12] Wang, Alex, et al., *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*, ICLR 2019, 2019.

[13] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, NAACL-HLT 2019, 2019.

[14] A. Vaswani, N. Shazeer, N. Parmar, et al., *Attention is all you need*, Advances in Neural Information Processing Systems (NeurIPS), 2017.

[15] A. v. d. Oord, Y. Li, and O. Vinyals, *Representation learning with contrastive predictive coding* in Proceedings of NeurIPS, 2018.