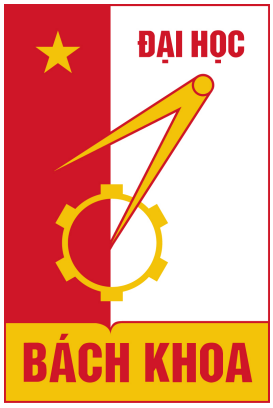
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

o0o



Nhận diện cảm xúc trong truyện tranh

Báo cáo bài tập lớn

Nhập môn Học máy và khai phá dữ liệu

Giáo viên hướng dẫn: PGS.TS Thân Quang Khoát

Sinh viên thực hiện : Nguyễn Tiến Long- 20180129
Phan Việt Hoàng- 20180086
Phạm Trần Anh- 20180018
Võ Hồng Sang - 20183973

Lớp : CTTN CNTT K63

Hà Nội - 2021

Mục lục

Chương 1

Giới thiệu

1.1 Bài toán nhận diện cảm xúc trong truyện tranh

Truyện tranh là một ngành công nghiệp tỷ đô đặc biệt phổ biến ở các khu vực Bắc Mỹ, Châu Âu và Châu Á. Ở thời kì đầu, truyện tranh được in trên sách giấy và trở thành một món ăn tinh thần không thể thiếu cho trẻ em thời bấy giờ. Những năm gần đây theo sự phát triển của công nghệ, chúng được đưa lên internet và ngày càng dễ tiếp cận với bạn đọc và trở nên phổ biến, giúp lan toả những giá trị văn hoá, giáo dục và giải trí trên toàn thế giới.

Tuy nhiên, các nội dung truyện tranh có mặt trên internet hiện tại đang gặp phải thách thức trong việc xấy dựng các công cụ đọc hiểu nội dung tự động (tương tự một số hệ thống truy vấn hình ảnh hay truy vấn video), do đó hạn chế các ứng dụng tìm kiếm nội dung trực tuyến hay các hệ thống gợi ý. Để cung cấp nội dung truyện tranh kỹ thuật số với trải nghiệm chính xác và thân thiện với người dùng trên tất cả các phương tiện, việc đọc hiểu và cân nhắc nội dung của chúng là thật sự cần thiết. Tuy nhiên ở quy mô toàn cầu, những công việc này khá tốn kém nếu thực hiện thủ công, do đó các quá trình xử lý tự động sẽ rất hữu ích để giữ chi phí cho các công việc nói trên ở mức chấp nhận được. Đây là một trong những lý do tại sao phân tích hình ảnh truyện tranh đã được nghiên cứu bởi cộng đồng phân tích dữ liệu từ khoảng hơn một thập kỷ vừa qua. Trên cơ sở đó, vẫn còn nhiều thách thức cần giải quyết trong lĩnh vực này. Mặc dù các yếu tố truyện tranh như cảnh vật, các đoạn hỗi thoại, văn bản tường thuật hiện được phát hiện và phân đoạn khá tốt (với các công cụ phân vùng ảnh và nhận diện kí tự quang học), nhưng việc phát hiện các nhân vật, nhận dạng văn bản và phân tích mối quan hệ giữa các yếu tố đó vẫn còn nhiều thách thức trong bối cảnh các tác vụ này vẫn chưa được nghiên cứu kỹ lưỡng [2]

1.2 Nhận diện cảm xúc con người

Sau đây chúng ta xem xét cách mô hình hóa cảm xúc của con người để phân tích và hiểu rõ hơn về cảm xúc trong truyện tranh qua. Bảng 1.1 trình bày bốn mô hình phổ biến cho những cảm xúc cơ bản. Với nền tảng là cuộc thi Kaggle [1], nhãn *neutral* được thêm vào vì ban tổ chức tin rằng không phải mọi trang truyện tranh đều tồn tại cảm xúc cho trước. Bên cạnh đó, nhãn *others* cũng được thêm vào để các mô hình đánh giá được tổng quan và không bị bias vào các cảm xúc cho trước. Sau cân nhắc kĩ lưỡng, cuối cùng tám nhãn đã được lựa chọn để đánh giá cảm xúc của con người trong cuộc thi này, bao gồm *angry, disgust, fear, happy, sad, surprise, neutral, and others*

Nghiên cứu	Các cảm xúc cơ bản
Ekman [3]	anger, disgust, fear, joy, sadness, surprise
Plutchik [4]	anger, anticipation, disgust, fear, joy, sadness, surprise, trust
Shaver [5]	anger, fear, joy, love, sadness, surprise
Lovheim [6]	anger, disgust, distress, fear, joy, interest, shame, surprise

Bảng 1.1: Bốn mô hình cảm xúc cơ bản [7]

1.3 Bài toán phân loại đa nhãn

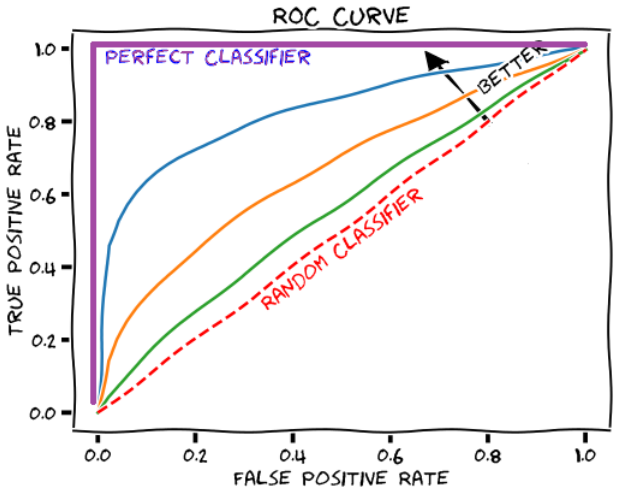
Trong cuộc thi này, người tham gia sẽ thiết kế các hệ thống học máy để tận dụng hiệu quả 2 nguồn dữ liệu: hình ảnh và văn bản (các đoạn text được trích xuất tự đồng). Mục tiêu là nhận diện cảm xúc theo từng chỉ tiêu trên từng mẫu dữ liệu. Ở giai đoạn kiểm thử, hệ thống sẽ được đưa vào một tập các tranh truyện và yêu cầu xác định xác suất của 8 nhãn mục tiêu xuất hiện trong trang truyện đó. Do đó bài toán được đặt ra là phân loại đa nhãn, tức là một điểm dữ liệu có thể thuộc nhiều nhãn.

1.4 Đánh giá cho mô hình

Các bài nạp sẽ được đánh giá dựa trên độ đo ROC-AUC (Area Under the Receiver Operating Characteristic Curve). Đường con ROC minh hoạ vê hiệu năng của mô hình phân loại nhị phân khi ngưỡng dự đoán thay đổi (giá trị được chọn để phân

[1] https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/

chia một điểm dữ liệu và 2 lớp dựa theo xác suất). Trong khi đó Area Under the ROC Curve (AUC) biểu diễn đường cong ROC thành một chỉ số duy nhất (đúng bằng phần diện tích dưới đường cong ROC). AUC cho một bài toán phân loại nhiều lớp có thể được tính bằng trung bình cộng các AUC cho từng nhãn (ta coi phân loại từng nhãn là các bài toán riêng biệt). Để tính chỉ số này, nhóm chọn cài đặt sẵn của 2 thư viện Scikit-learn [2] và Tensorflow [3].



Hình 1.1: Minh hoạ độ đo ROC-AUC.

[2]http://bit.ly/scikit-learn-auc
[3]https://www.tensorflow.org/api_docs/python/tf/keras/metrics/AUC

Chương 2

Tổng quan về bộ dữ liệu

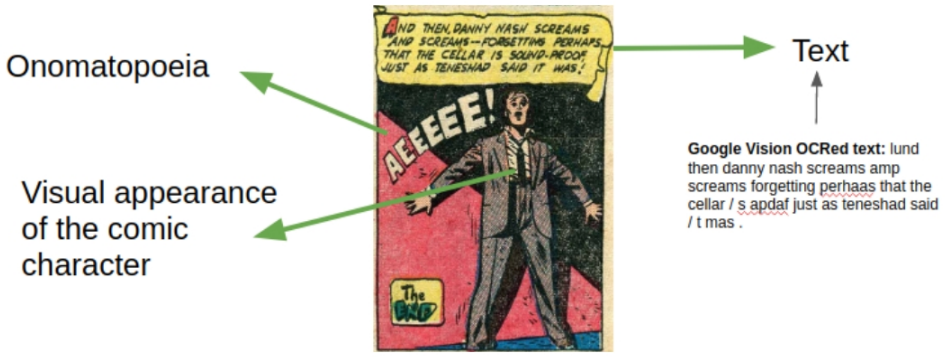
2.1 Bộ dữ liệu Emorecom

Trong cuộc thi này, nhóm được yêu cầu giải quyết một trong những thách thức của phân tích cảnh truyện tranh: nhận biết cảm xúc của cảnh truyện tranh. Cảm xúc đến từ cảm xúc của nhân vật truyện tranh trong câu chuyện và được mô hình hoá bằng thông tin hình ảnh, văn bản trong bong bóng thoại hoặc chú thích và từ tượng thanh (hình vẽ truyện tranh của các từ bắt chước ngữ âm, giống hoặc gợi ý âm thanh mà nó mô tả), xem Hình 2.1. Trong khi nhận dạng cảm xúc được nghiên cứu rộng rãi trong các lĩnh vực và dữ liệu khác, chẳng hạn như thị giác máy tính và xử lý ngôn ngữ tự nhiên, các bài toán với dữ liệu đa phương thức từ mạng xã hội, nó chưa được khai thác với hình ảnh truyện tranh chứa cả hình ảnh và văn bản. Được thúc đẩy bởi giá trị của các phương pháp tiếp cận multimodal, cuộc thi khuyến khích những người tham gia sử dụng lợi thế của các đặc trưng từ nhiều nguồn dữ liệu để suy ra cảm xúc. Do đó, nhiệm vụ này là một bài toán multimodal có thể tận dụng lợi thế từ cả hai lĩnh vực: thị giác máy tính và xử lý ngôn ngữ tự nhiên cũng là một trong những nhiệm vụ chính của cộng đồng phân tích dữ liệu.

Trong cuộc thi này, các hình ảnh được thu thập và gán nhãn theo cách crowd-sourced để cho ra 8 nhãn ứng với mỗi ảnh. Số liệu thống kê cho từng nhãn được cho trong bảng 2.1

Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	Others
4005	3678	3485	4197	1525	3435	6914	670

Bảng 2.1: Thống kê của bộ dữ liệu Emorecom với số ảnh tương ứng với mỗi nhãn



Hình 2.1: Ví dụ về một nhân vật truyện tranh với hình ảnh trực quan và văn bản chú thích. Cần lưu ý rằng các văn bản kết quả từ phương pháp OCR có thể có lỗi (ví dụ: các từ được gạch chân màu đỏ trong mẫu dữ liệu).

Cuộc thi được tổ chức trên diễn đàn Codalab [1] từ ngày 15 tháng 12 năm 2020 đến 31 tháng 3 năm 2021 và thu hút 145 lượt đăng kí, 21 đội tham gia tới vòng private test và 7 đội hoàn thành mọi chặng thi. Timeline của các vòng có thể được tham khảo tại đây [2]

2.2 Chuẩn bị dữ liệu

Bộ dữ liệu Emorec bao gồm các trang truyện được lấy từ bộ dữ liệ publiu COMICS [3]. COMICS (120 GB) bao gồm 1,2 triệu cảnh truyện cùng với các đoạn text được đọc ra bởi Google Vision OCR, xem ảnh 2.1

Warm-Up	Public Training	Public Testing	Private Testing
100	6,112	2,046	2,041

Bảng 2.2: Số lượng dữ liệu cho các giai đoạn cuộc thi

Bộ dữ liệu cuối cùng bao gồm training set, public test set và private test set (xem 2.2, có 6112 điểm dữ liệu cho training set tương ứng với 2046 điểm (bao gồm ảnh và văn bản) ở giai đoạn public. Các đội thi có thể xem được kết quả bài dự đoán của mình trên trang chủ Codalab.

[1] https://competitions.codalab.org/competitions/27884
[2] https://emoreccom.univ-lr.fr
[3] https://obj.umiacs.umd.edu/comics/index.html

1. **Warm Up:** Từ 16/12/2020 đến 10/1/2021 người tham gia được cung cấp bộ dữ liệu warm up gồm 100 điểm dữ liệu để quen với format của dữ liệu Emorecom

2. **Public data:** Từ 10/1/2021 đến 24/3/2021 người tham gia được cung cấp 6112 điểm dữ liệu huấn luyện tương ứng với 2046 điểm dữ liệu kiểm thử (không có nhãn) và có thể nạp bài dự đoán lên diễn đàn để xem kết quả và ranking hiện tại trên bảng tổng sắp

3. **Private Test:** Từ 24/3/2021 đến 31/3/2021 người tham gia được cung cấp 2041 điểm dữ liệu không có nhãn và yêu cầu nạp dự đoán cho bộ này trước thời hạn kết thuc để đánh giá kết quả cuối cùng

2.3 Phương thức đánh giá

Script đánh giá được cài đặt và chạy tự động trên nền tảng Codalab và điểm của bài nạp sẽ được đánh giá một cách tự động. Như đề cập ở trên, có tổng cộng 8 nhãn bao gồm *0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral, 7=Others.*. Người tham gia được yêu cầu nạp file dự đoán theo đúng thứ tự trên với xác suất biểu thị cho sự xuất hiện của từng trạng thái cảm xúc trong trang truyện đó theo format sau:

image_id	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	Others
0_27_5	0.55	0.06	0.09	0.37	0.44	0.83	0.28	0.47
17_10_3	0.09	0.31	0.39	0.35	0.74	0.95	0.02	0.15

Chương 3

Tiếp cận bài toán

3.1 Tiền xử lý dữ liệu

Dữ liệu của BTC cho gồm 5 thành phần chính

- **train_transcriptions.json:** chứa dữ liệu huấn luyện dạng văn bản
- **train:** folder chứa các tranh truyện trong tập huấn luyện
- **train_emotion_labels.csv:** chứa nhãn của tập dữ liệu huấn luyện
- **additional_infor:emotion_polarity.csv [1]:** chứa xác suất của các nhãn trong tập train
- **test_transcriptions.json:** chứa dữ liệu kiểm thử dạng văn bản
- **test:** folder chứa các tranh truyện trong tập kiểm thử

3.1.1 Dữ liệu hình ảnh

Để thuận tiện cho việc huấn luyện mô hình, nhóm quyết định thay đổi hình dạng ảnh về chung kích cỡ là 256×256 và quyết định không tiến hành thêm các kĩ thuật augmentation ảnh vì chúng sẽ làm ảnh hưởng đến các thông tin dạng chữ có trong tranh truyện

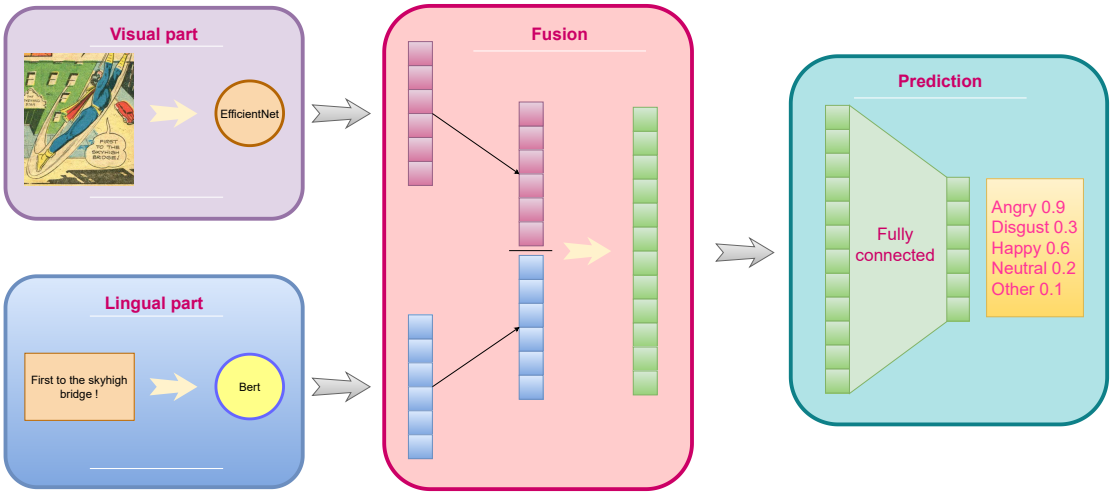
3.1.2 Dữ liệu văn bản

Vì đầu ra của Google Vision OCR là không thật sự tốt, có thể một phần vì các tranh truyện đã cũ nên bị khá nhiều lỗi, do đó nhóm quyết định dùng luật để sửa lại một số chỗ và chuyển hết các chữ về dạng chữ thường (đúng như output của Google Vision OCR).

[1] Nhóm quyết định không dùng nguồn dữ liệu này vì không làm tăng hiệu năng mô hình

3.2 Tổng quan mô hình

Như trình bày trực quan trong hình 3.1, mô hình đề xuất gồm có 3 thành phần chính là 2 khối pretrained model để biểu diễn dữ liệu dạng ảnh là Efficient Net [8] và dữ liệu văn bản là Pre-training of Deep Bidirectional Transformers for Language Understanding [9] . Các đặc trưng được trích xuất ra sẽ được kết hợp và đưa ra tầng cuối cùng để phân loại.



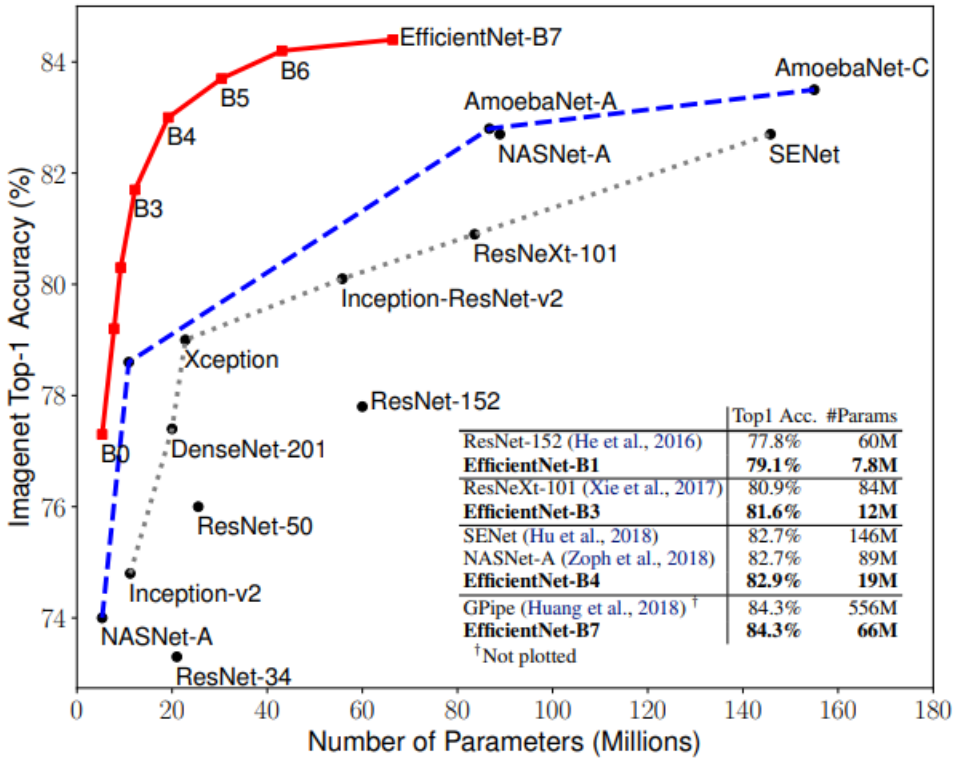
Hình 3.1: Mô hình đề xuất với cơ chế early fusion, kết hợp các đặc trưng từ ảnh và văn bản.

3.2.1 EfficientNet

Mạng Nơ-ron tích chập (Convolutional Neural Networks - ConvNets) thường được phát triển với ngân sách tài nguyên cố định và sau đó được thu phóng để có độ chính xác tốt hơn nếu có nhiều tài nguyên hơn. (Nguyên văn: Convolutional Neural Networks (ConvNets) are commonly developed at a fixed resource budget, and then scaled up for better accuracy if more resources are available.). Bởi vậy nên nhóm tác giả Mingxing Tan và Quoc V. Le đã nghiên cứu một cách có hệ thống và nhận thấy rằng việc cân bằng một cách có hệ thống độ sâu, chiều rộng và độ phân giải mạng (network depth, width, and resolution) có thể mang đến hiệu suất tốt hơn.

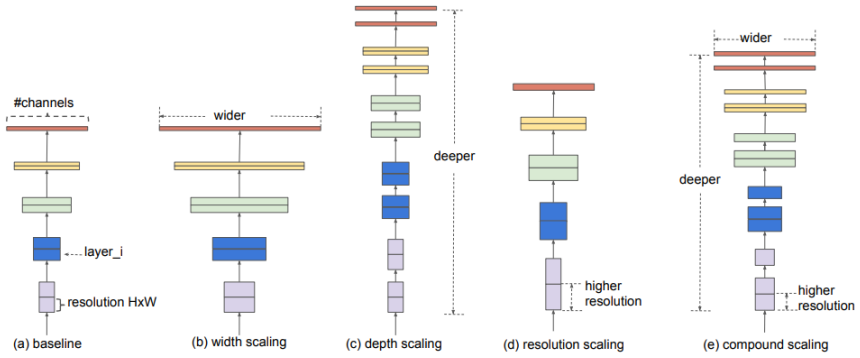
Như ta đã biết, có ba kích thước tỷ lệ của CNN: depth, width, và resolution:

- **Depth** là độ sâu của mạng tương đương với số lớp trong đó.
- **Width** là độ rộng của mạng. Ví dụ: một thước đo chiều rộng là số kênh trong lớp Conv.



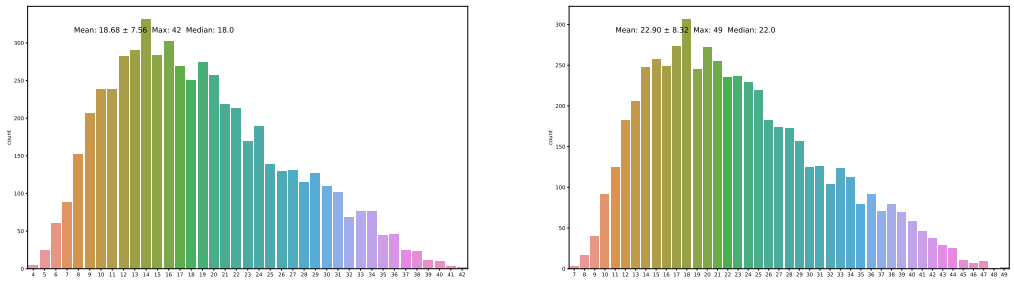
Hình 3.2: Hiệu năng của các mô hình EfficientNet trên bộ dữ liệu Imagenet [2]

- **Resolution** là độ phân giải hình ảnh được chuyển đến CNN.



Hình 3.3: Ý tưởng về việc thu phóng mô hình trên các kích thước khác nhau.

Chúng ta hay tự hỏi rằng tại sao việc thu phóng mô hình lại quan trọng. Câu trả lời là, ta có thể nói rằng việc thu phóng thường được thực hiện để cải thiện độ chính xác của mô hình đối với một tác vụ nhất định, chẳng hạn như phân loại ImageNet. Việc thu phóng quy mô, nếu được thực hiện đúng cách, cũng có thể



Hình 3.4: Độ dài câu theo mức từ (trái) và BPE token (phải)

giúp cải thiện hiệu quả của một mô hình.

Để chứng minh tốt hơn hiệu quả của phương pháp thu phóng quy mô của mình, nhóm tác giả cũng đã phát triển một mạng cơ sở kích thước di động, được gọi là EfficientNet. Các mô hình EfficientNet của nhóm tác giả thường sử dụng thứ tự các tham số và FLOPS ít hơn so với các ConvNets khác với độ chính xác tương tự. Đặc biệt, EfficientNet-B7 của chúng tôi đạt độ chính xác top1 84,3% với thông số 66M và 37B FLOPS, chính xác hơn nhưng nhỏ hơn 8,4 lần so với GPipe tốt nhất trước đây. Những lợi ích này đến từ cả kiến trúc tốt hơn, thu phóng quy mô tốt hơn và cài đặt đào tạo tốt hơn được tùy chỉnh cho EfficientNet.

3.2.2 BERT

Trong xử lý ngôn ngữ tự nhiên, việc biểu diễn một từ thành một vector đóng một vai trò cực kỳ quan trọng. Nó lợi ích rất nhiều trong việc thể hiện sự tương đồng, đối lập về ngữ nghĩa giữa các từ, giúp mô hình hóa vector cho 1 câu hay đoạn văn, tìm các câu có nghĩa tương đồng... Word embedding là một nhóm các kỹ thuật đặc biệt trong xử lý ngôn ngữ tự nhiên, có nhiệm vụ ánh xạ một từ hoặc một cụm từ trong bộ từ vựng tới một vector số thực. Từ không gian một chiều cho mỗi từ tới không gian các vector liên tục. Các vector từ được biểu diễn theo phương pháp word embedding thể hiện được ngữ nghĩa của các từ, từ đó ta có thể nhận ra được mối quan hệ giữa các từ với nhau(tương đồng, trái nghịch,...).

Trong năm 2013, một ý tưởng được đưa ra bởi Tomas Mikolov- một kỹ sư đang làm tại Google đã giải quyết được các vấn đề trên bằng một mô hình hoàn toàn khác. Mô hình được sử dụng tốt cho đến ngày nay và được gọi là mô hình word2vec [11]. Word2vec là một mạng neural 2 lớp với duy nhất 1 tầng ẩn, lấy đầu vào là một corpus lớn và sinh ra không gian vector(với số chiều khoảng vài trăm), với mỗi từ duy nhất trong corpus được gắn với một vector tương ứng trong không gian.

Hình 3.4 cho ta thống kê về độ dài của các câu có trong bộ dữ liệu dạng văn

bản, từ đây ta chọn các tham số về độ dài đoạn văn tương ứng. Cụ thể ta chọn 42 cho độ dài lớn nhất của câu ở mức từ và 56 cho độ dài lớn nhất khi các câu được tách ra thành các BPE tokens [3].

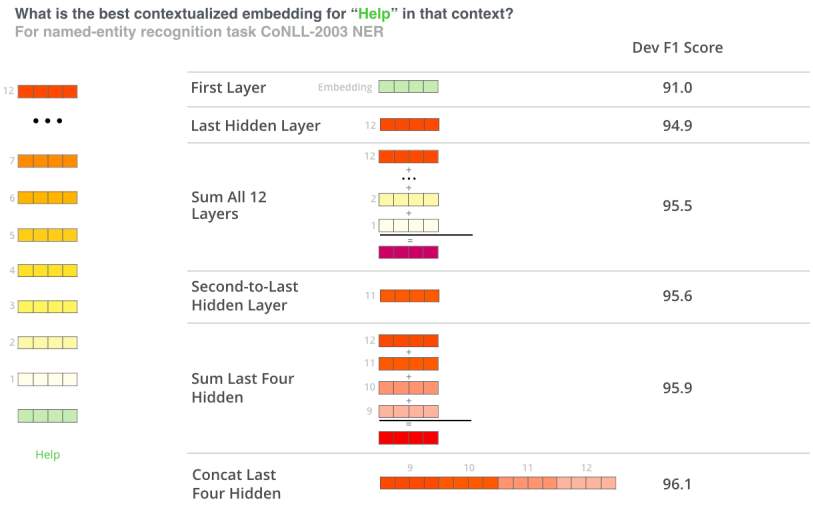
Các word vectors được xác định trong không gian vector sao cho những từ có chung ngữ cảnh trong corpus được đặt gần nhau trong không gian. Dự đoán chính xác cao về ý nghĩa của một từ dựa trên những lần xuất hiện trước đây. Về mặt lý thuyết, các kỹ thuật khác như Word2vec, FastText hay Glove cũng tìm ra đại diện của từ thông qua ngữ cảnh chung của chúng. Tuy nhiên, những ngữ cảnh này là đa dạng trong dữ liệu tự nhiên.Trong khi các mô hình như Word2vec, fastText tìm ra 1 vector đại diện cho mỗi từ dựa trên 1 tập ngữ liệu lớn nên không thể hiện được sự đa dạng của ngữ cảnh. Việc tạo ra một biểu diễn của mỗi từ dựa trên các từ khác trong câu sẽ mang lại kết quả ý nghĩa hơn nhiều. BERT mở rộng khả năng của các phương pháp trước đây bằng cách tạo các biểu diễn theo ngữ cảnh dựa trên các từ trước và sau đó để dẫn đến một mô hình ngôn ngữ với ngữ nghĩa phong phú hơn.

Bidirectional Encoder Representations from Transformers là một mô hình học máy xử lý ngôn ngữ tự nhiên do Google phát triển. BERT được tạo và xuất bản vào năm 2018 bởi Jacob Devlin và các đồng nghiệp của ông từ Google. Nó có thể được sử dụng trong nhiều bài toán NLP như:

- Phân loại văn bản
- Question answering
- Dịch máy
- Nhận dạng thực thể có tên
- Tóm tắt văn bản
- ...

Không nghi ngờ gì nữa, BERT là một bước đột phá trong việc sử dụng học máy trong xử lý ngôn ngữ. Hơn thế nữa, nó là mã nguồn mở và cho phép tinh chỉnh nhanh chóng, phạm vi ứng dụng thực tế là rất lớn.
Văn bản ban đầu, sau khi được tách thành các BPE tokens (token_id) truncate và padding nhờ vào bộ tách từ cũng của thư viện open source Huggingface [4] sẽ được đưa vào mô hình cùng với attention_mask và token_type_id

[3]https://huggingface.co/transformers/tokenizer_summary.html
[4]https://huggingface.co/



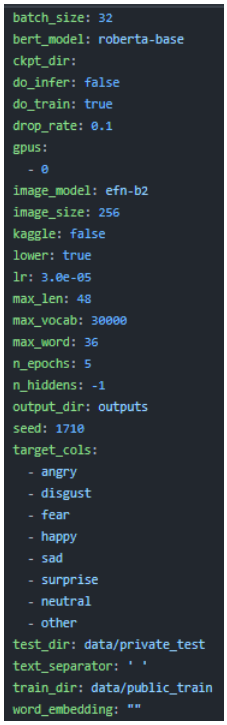
Hình 3.5: So sánh các cách kết hợp embedding mức từ của BERT

Lấy ý tưởng từ paper gốc của BERT và kết quả của Bert-base trên share task CoNLL-2003 NER. Mô hình được đề xuất có thử nghiệm cả việc lấy embedding của token [CLS] và ghép 4 trạng thái ẩn cuối cùng của BERT model 3.5 . Kết quả thực nghiệm cho thấy việc lấy 4 trạng thái ẩn cho kết quả cao hơn trên nhiều seed khác nhau, do đó đây là cài đặt được sử dụng cho mô hình cuối cùng

3.3 Cài đặt

Để có một cách đánh giá trực quan và công bằng giữa các lần chạy thực nghiệm, nhóm quyết định sử dụng chiến lược K-fold Cross-Validation [5]. Với K được chọn bằng 5.

[5]https://machinelearningmastery.com/k-fold-cross-validation/



```
batch_size: 32
bert_model: roberta-base
ckpt_dir:
do_infer: false
do_train: true
drop_rate: 0.1
gpus:
  - 0
image_model: efn-b2
image_size: 256
kaggle: false
lower: true
lr: 3.0e-05
max_len: 48
max_vocab: 30000
max_word: 36
n_epochs: 5
n_hiddens: -1
output_dir: outputs
seed: 1710
target_cols:
  - angry
  - disgust
  - fear
  - happy
  - sad
  - surprise
  - neutral
  - other
test_dir: data/private_test
text_separator: ' '
train_dir: data/public_train
word_embedding: ""
```

Với mỗi lần chạy, cấu hình của mô hình sẽ được lưu lại để dễ dàng đánh giá và hơn thế nữa, file **config.yaml** dùng để dựng lại mô hình, load lại trọng số của mạng mà sẽ dùng để dự đoán tập private test. Như vậy với mỗi cấu hình, ta có 5 mô hình tương ứng với 5 fold. Dự đoán của các mô hình đó sẽ được cộng trung bình để đưa ra dự đoán cho bất kì tập dữ liệu nào đưa vào.

Việc sử dụng một số các mô hình pretrained khá lớn như Bert hay Effcientnet bên cạnh khả năng học biểu diễn vô cùng mạnh mẽ vẫn còn đó có một số nhược điểm mà nổi bật nhất trong số đó là vấn đề overfitting. Để cố gắng hạn chế vấn đề này, nhóm quyết định sử dụng dropout[13], đặt vào ngay sau các model Bert và Efficient Net để tránh việc các mô hình pretrained này khớp với dữ liệu quá nhanh. Hiểu 1 cách đơn giản thì Dropout là việc bỏ qua các đơn vị (tức là 1 nút mạng) trong quá trình đào tạo 1 cách ngẫu nhiên. Bằng việc bỏ qua này thì đơn vị đó sẽ không được xem xét trong quá trình forward và backward. Theo đó, p được gọi là xác suất giữ lại 1 nút mạng trong mỗi giai đoạn huấn luyện, vì thế xác suất nó bị loại bỏ là (1 - p).
Để đa dạng thực nghiệm, nhóm quyết định sử dụng một vài biến thể của Bert:

- DistilBert [18]
- RoBerta [17]
- Albert [19]
- ...

hay thậm chí thay EfficientNet bằng một số mô hình huấn luyện sẵn trên bộ Image Net như:

- VGG [16]

- Resnet [14]
- ResNext [15]
- ...

Cụ thể về điểm khác nhau cũng như ưu/nhược điểm của từng mô hình nhóm xin phép không trình bày ở đây. Qua một số thử nghiệm, mô hình cuối cùng (mô hình đạt được hiệu năng tốt nhất trên đánh giá) là sự kết hợp của Roberta và Efficientnet B2. Về cơ bản, Roberta sử dụng chung kiến trung như Bert và điểm khác nhau nằm ở việc huấn luyện khi Roberta được huấn luyện lâu hơn batch size lớn hơn, câu văn dài hơn và nhiều dữ liệu hơn. Phần tác vụ next sentence prediction trong paper gốc của Bert [9]. Tác vụ Mask language model bên cạnh đó cũng có một số thay đổi

Bên cạnh đó, nhóm còn thử nghiệm thêm các cách cài đặt Early/Late Fusion [12] hay các mô hình chỉ sử dụng hình ảnh/văn bản và so sánh kết quả. Phần tiếp theo của báo cáo sẽ đi sâu vào kết quả thực nghiệm.

Chương 4

Kết quả và đánh giá

4.1 Đánh giá từng thành phần

Để có một cái nhìn tốt hơn về độ quan trọng của từng thành phần của mô hình cũng như khả năng khai thác thông tin từ các nguồn dữ liệu hình ảnh và văn bản, nhóm huấn luyện 4 mô hình khác với cách sử dụng dữ liệu đầu vào khác nhau:

1. Roberta với dữ liệu văn bản: mô hình huấn luyện đơn giản là mô hình phân loại văn bản
2. EfficientNet B3 với dữ liệu ảnh: tương tự đây là mô hình phân loại hình ảnh
3. Early fusion với cả 2 nguồn dữ liệu: biểu diễn của hình ảnh và văn bản được kết hợp trước khi đưa ra phân loại
4. Late fusion với cả 2 nguồn dữ liệu: là mô hình được huấn luyện trên từng nguồn dữ liệu, sau đó kết hợp kết quả để đưa ra kết quả dự đoán cuối cùng

Kết quả được ghi lại trong bảng sau:

Cách huấn luyện	Pretrained model	ROC-AUC
Chỉ dùng văn bản	Roberta	0.6358
Chỉ dùng hình ảnh	Efficienet (B3)	0.5412
Early fusion	**Efficienet (B3) + Roberta**	**0.6423**
Late fusion	Efficienet (B3) + Roberta	0.6288

Bảng 4.1: Kết quả của một số cách huấn luyện khác nhau

Rõ ràng dữ liệu dạng văn bản chiếm ưu thế quan trọng trong cuộc thi này (đạt được đến tận 0.6358 AUC) trong khi dữ liệu dạng ảnh sẽ không thực sự tốt nếu chỉ dùng một mình-chỉ đạt được kết quả xấp xỉ dự đoán ngẫu nhiên (0.5412). Kết hợp cả 2 nguồn dữ liệu này cho kết quả rất tốt với cách huấn luyện early fusion, đã trình bày ở trên (0.6423) - đây cũng sẽ là cách huấn luyện chung cho các mô hình sau này.

4.2 Kết quả

Dựa vào nhận xét từ mục 4.1 và cách cài đặt như đã được trình bày ở trên. Nhóm huấn luyện một số cặp pretrained image-text và ghi lại kết quả vào bảng 4.2. Cấu hình đạt được kết quả cao nhất là cặp EfficientNet B2 + Roberta base, với 0.652 ROC-AUC trên trung bình 5-folds. Tuy nhiên ta vẫn giữ lại cấu hình, file dự đoán và trọng số của các cặp mô hình còn lại để ensemble [1]

Mô hình	DistilBERT	Bert base uncased	Bert base cased	Roberta base
EfficientNet B0	0.6340	0.6393	0.6365	0.6448
EfficientNet B2	0.6343	0.6429	0.6379	**0.652**
EfficientNet B5	0.6261	0.6384	0.6309	0.6444

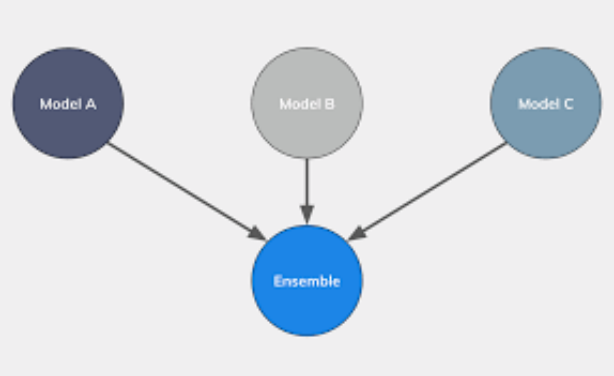
Bảng 4.2: Kết quả của các mô hình pretrained

4.3 Ensemble và kết quả tổng sắp

Với tổng cộng 15 cấu hình với các cặp pretrained khác nhau, mỗi cấu hình ta có 5 mô hình (như được trình bày ở 3.3). Mỗi fold được huấn luyện 5 epoch với thời gian khoảng 2 phút/epoch hay $2 \times 5 \times 5 = 50$ phút cho mỗi cấu hình.$15 \times 5 = 75$ mô hình được trình bày trên sẽ được đưa vào ensemble.

Dựa trên cơ sở cuộc thảo luận ở đây [2] Ta huấn luyện thêm một mô hình cuối cùng (Logistic Regression [3] nhận đầu vào là các file dự đoán và yêu cầu đầu ra cũng là 1 file dự đoán để kết hợp hiệu năn của cả 75 mô hình trước đó. Kết quả nhận được, ROC-AUC đã cải thiện từ 65.2 $\rightarrow$ 66.3, trở nên vô cùng vượt trội.

[1] https://machinelearningmastery.com/ensemble-learning-algorithms-with-python/
[2] https://www.kaggle.com/c/instant-gratification/discussion/93526
[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html



Hình 4.1: Minh hoạ cho ensemble

Nạp bài dự đoán này lên trên nền tảng Codalab, cả nhóm đã dành được top 2 trên bảng tổng sắp public và top 1 trên bảng tổng sắp cuối cùng.

Chương 5

Tổng kết

Bên cạnh việc sử dụng các mô hình học sâu có hiệu năng lớn, một trong những yếu tố yêu cầu cho người làm học máy là công việc xử lý dữ liệu, trích chọn đặc trưng để đưa vào huấn luyện. Công việc này yêu cầu người kĩ sư có cảm quan thật tốt và hiểu rõ về dữ liệu, bên cạnh đó là việc thực nghiệm cần thiết trong tiền xử lý, phân tích dữ liệu và xác định các siêu tham số trong tuning mô hình.

Quay lại với bài toán được đặt ra của cả nhóm, đây là một trong những hướng nghiên cứu rất tiềm năng trong thời gian gần đây - các bài toán về multimodal - xây dựng những hệ thống đa tác vụ, có khả năng xử lý nhiều dạng dữ liệu. Một số ứng dụng có thể kể đến là các hệ gợi ý : truyện tranh, video clip ... Nhóm đã có một số đề xuất, đi từ cơ sở là giải quyết các mô hình riêng lẻ để giải quyết từng vấn đề được đặt ra cho đến xây dựng một mô hình lớn, có khả năng trích xuất được nhiều nguồn dữ liệu để tận dụng tối đa nguồn dữ liệu giá trị này. Một mô hình đơn lẻ có vẻ như sẽ không đủ để dành chiến thắng trong một cuộc thi học máy - nơi người tham gia dành rất nhiều thời gian và kĩ thuật trong việc stacking/ensemble. Đột phá trong cách tiếp cận của nhóm phải kể đến kĩ thuật ensemble với Logistic Regression được nhắc đến trong mục , giúp tăng đáng kể hiệu năng của mô hình và giảm bias. Tuy nhiên, việc ensemble trong thực tế lại khá hạn chế vì tốc độ dự đoán cũng như độ phức tạp, dung lượng bộ nhớ yêu cầu để huấn luyện và lưu trữ các mô hình là rất lớn - xấp xỉ 75 lần một mô hình riêng lẻ trong cách tiếp cận của nhóm. Do đó đôi khi chúng ta phải đánh đổi dữa độ chính xác và tốc độ tính toán cũng như bộ nhớ,

Chúng em xin chân thành cảm ơn sự góp ý từ thầy và anh giản, giúp đỡ trong suốt quá trình hoàn thành đề tài môn học Nhập môn học máy và khai phá dữ liệu cũng như kiến thức học được trong học kì vừa qua. Đây sẽ là những kiến thức vô cùng quý giá cho sinh viên ngành khoa học máy tính bọn em!

Tài liệu tham khảo

[1] Khoat Than Quang, Introduction to Machine Learning and Data Mining (2021), School of Information and Communication Technology Hanoi University of Science and Technology.

[2] Augereau, O., Iwata, M., Kise, K. A survey of comics research in computer science. Journal of Imaging 4 (04/2018)

[3] Ekman, P.: An argument for basic emotions. Cogn. Emot. 6(3-4), 169–200 (1992)

[4] Plutchik, R., Kellerman, H.: Emotion: Theory, research and experience. Academic Press 3 (1986)

[5] Shaver, P., Schwartz, J., Kirson, D., O´connor, C.: Emotion knowledge: Further exploration of a prototype approach. J. Pers. Soc. Psychol. 52(6) (1987)

[6] Lovheim, H.: A new three-dimensional model for emotions and monoamine neuro transmitters. Med. Hypoth. 78(2), 341–348 (2012)

[7] , A., Shahraki, A.G., Zaiane, O.R.: Current state of text sentiment analysis from opinion to emotion mining. ACM Comput. Surv. 50(2), Article 25 (2017)

[8] Mingxing Tan, Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. International Conference on Machine Learning, 2019

[9] J Devlin, MW Chang, K Lee, K Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805v2, 2019

[10] What Are Word Embeddings for Text? `https://machinelearningmastery.com/what-are-word-embeddings/`

[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean Distributed Representations of Words and Phrases and their Compositionality. arXiv preprint arXiv:1310.4546v1, 2013

[12] Konrad Gadzicki; Razieh Khamsehashari; Christoph Zetzsche. Early vs Late Fusion in Multimodal Convolutional Neural Networks. 2020 IEEE 23rd International Conference on Information Fusion (FUSION)

[13] T. Mikolov, I. Sutskever, K. Chen, G. Corrado Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research (2013)

[14] K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778

[15] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1492-1500

[16] K Simonyan, A Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556, 2014.

[17] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint at arXiv:1907.11692, 2019.

[18] 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.

[19] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. arXiv preprint arXiv:1909.11942v6. 2019