TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

CS221.O12.KHCL

BÁO CÁO THỰC NGHIỆM ĐỒ ÁN CUỐI KÌ: PHÂN LOẠI CẢM XÚC VỚI VGCN-BERT

Nhóm: 1
Huỳnh Hoàng Khánh - 21520976
Nguyễn Dương Quốc Anh - 21521829
Nguyễn Đinh Minh Chí - 21520648
Nguyễn Thành Đăng - 21521921

Giảng viên:
Nguyễn Thị Quý

Ngày 18 tháng 12 năm 2023

Mục lục

1 Giới thiệu

1.1 Nguồn tham khảo chính

Phần source code được sử dụng trong đồ án được tham khảo từ repo github chính thức trong báo khoa học: "VGCN-BERT: Augmenting BERT with Graph Embedding for Text Classification" thuộc quyền sở hữu của nhóm tác giả Zhibin Lu, Pan Du, và Jian-Yun Nie.

Tuy nhiên, trong repo này xuất hiện một số lỗi và phần giao diện Hugging Face được nhóm tác giả đề xuất không thực sự liên quan đến đồ án môn học, nên nhóm chúng em đã code thêm một số file để đánh giá và hiện thực lại bài báo này.

1.2 Bài toán phân tích cảm xúc (sentiment analysis)

Bài toán: Phân tích cảm xúc (Sentiment Analysis) là một lĩnh vực nghiên cứu trong xử lý ngôn ngữ tự nhiên (NLP) nhằm hiểu và đánh giá ý kiến, cảm nhận, hoặc tư duy của người dùng được diễn đạt trong văn bản. Mục tiêu của bài toán là xác định cảm xúc tích cực hoặc tiêu cực của một đoạn văn bản dựa trên ngữ cảnh và nội dung của từng câu hoặc đoạn văn.Ứng dụng của phân tích cảm xúc rất đa dạng, từ đánh giá ý kiến khách hàng trên mạng xã hội đến đo lường tình cảm trong các đoạn văn trong tin tức và đánh giá sản phẩm.

Trong đồ án cuối kì này, nhóm chúng em đã thực hiện một mô hình dự đoán cảm xúc tích cực hoặc tiêu cực dựa trên bộ dataset SST-2. Bài toán này ứng dụng cụ thể trong việc đánh giá cảm xúc của khác hàng đối với những bộ phim và có thể dễ dàng ứng dụng cho các loại dữ liệu tương tự. Mặt khác, nhóm cũng đưa ra quan điểm đánh giá cá nhân về phương pháp mà bài báo khoa học: "VGCN-BERT: Augmenting BERT with Graph Embedding for Text Classification" sử dụng.

1.3 Dataset sử dụng cho bài toán

SST-2 (Stanford Sentiment Treebank) là một tập dữ liệu được thiết kế để thực hiện bài toán phân loại nhị phân cho các câu đơn với mục tiêu chính là phân tích cảm xúc. Tập dữ liệu bao gồm các câu được trích xuất từ các đánh giá phim, mỗi câu đi kèm với nhãn cảm xúc được gán bởi con người [24]. Phiên bản công khai của tập dữ liệu bao gồm 6.920 ví dụ trong tập huấn luyện, 872 ví dụ trong tập xác thực và 1.821 ví dụ trong tập kiểm thử. Tổng cộng, tập dữ liệu chứa 4.963 đánh giá tích cực và 4.650 đánh giá tiêu cực.

Độ dài trung bình của các đánh giá trong tập dữ liệu SST-2 là 19,3 từ. Tập dữ liệu này là một nguồn tài nguyên quý giá để huấn luyện và đánh giá các mô hình phân tích cảm xúc, giúp các nghiên cứu viên và chuyên gia có thể phát triển các hệ thống có khả năng dự đoán chính xác cảm xúc được thể hiện trong các câu cá nhân trong ngữ cảnh của các đánh giá phim.

1.4 Sơ lược về hướng giải quyết bài toán

Ý tưởng chính của bài báo khoa học là sự kết hợp của các đặc trưng cục bộ (trích xuất từ BERT) và các đặc trưng toàn cục (trích xuất từ Graph Convolutional Networks) để giúp cho mô hình phân loại có thể hiểu được toàn bộ ý nghĩa của văn bản mà vẫn không bỏ qua ý nghĩa của từng từ (word) có trong văn bản. Pipe line của bài toán có thể được diễn tả như sau:

Phân tích và tạo ra một bộ từ điển từ tập dữ liệu, đồng thời xây dựng một Graph Convolutional Networks dựa trên yếu tố "đồng xuất hiện" của hai từ trong các cặp sliding window.

Kết hợp giữa đặt trưng trích xuất từ word embedding và Grap Convolutional Network để đưa vào bộ BERT. Đầu ra sẽ được đưa qua các lớp Fully connected để tạo thành bộ phân loại ứng với số lớp mà chúng ta mong muốn.

2 Preprocessing Data

2.1 Mục đích:

Giai đoạn tiền xử lý dữ liệu trong quy trình này có mục tiêu loại bỏ dấu câu, chuẩn hóa văn bản thành chữ thường và thực hiện tách từ (tokenization). Điều này sẽ chuẩn hóa dữ liệu và chuẩn hóa việc biểu diễn các từ trong văn bản, phù hợp để xử lý các bước tiếp theo.

2.2 Thiết lập môi trường

Đầu tiên, chúng ta phải clone repo github tại url chính thức được nhóm tác giả công bố: https://github.com/Louis-udm/VGCN-BERT.git

```
!git clone https://github.com/Louis-udm/VGCN-BERT.git
```

Tiếp theo, chúng ta cần phải cài đặt một số thư viện, framwork mà dự án cần sử dụng.

```
    !pip install NLTK
    !pip install python-dotenv
    !pip install boto3
```

- NLTK (Natural Language Toolkit)
  - NLTK là một thư viện Python phổ biến được sử dụng trong xử lý ngôn ngữ tự nhiên (NLP). Nó cung cấp nhiều công cụ và tài nguyên, bao gồm các dữ liệu ngôn ngữ và các chức năng cho các nhiệm vụ như phân loại văn bản, tách từ, đánh giá cảm xúc, v.v.
- python-dotenv
  - python-dotenv là một thư viện Python giúp quản lý biến môi trường thông qua tệp .env. Điều này làm cho việc lưu trữ cấu hình, thông tin đăng nhập, hoặc bất kỳ biến môi trường nào khác trở nên thuận tiện hơn và an toàn hơn.
- boto3
  - Boto3 là một SDK (Software Development Kit) Python cho Amazon Web Services (AWS). Nó cung cấp các chức năng để tương tác với các dịch vụ của AWS như S3, DynamoDB, EC2, và nhiều dịch vụ khác.

2.3 Cấu trúc của bộ dữ liệu SST-2

<test.txt>
0 no movement , no yuks , not much of anything .
0 a gob of drivel so sickly sweet , even the eager consumers of moore 's pasteurized ditties will retch it up like rancid crème brûlée .
...
1 as an entertainment , the movie keeps you diverted and best of all , it lightens your wallet without leaving a sting .
0 aspires for the piquant but only really achieves a sort of ridiculous sourness .
1 chicago is , in many ways , an admirable achievement .

Các file txt này cần được tổ chức lưu trữ trong ./data/<tên bộ dữ liệu>/. Các tập train, valid và test được lưu dưới dạng <train.txt, valid.txt, test.txt> tương ứng trong đường dẫn trên. Trong mỗi file txt bao gồm nhiều hàng dữ liệu, mỗi hàng là một điểm dữ liệu gồm hai phần. Phần đầu là nhãn và phần sau là một câu đánh giá các nhau bởi dấu cách.

2.4 Tiền xử lí dữ liệu

Để tiền xử lí dữ liệu, chúng ta chỉ cần thực hiện câu lệnh sau:

```
!python -m vgcn_bert.prepare_data --ds [dataset]
```

Trong đó [dataset] có thể là "cola" hoặc "sst" tùy vào mục đích của người sử dụng. Nhóm sẽ phân tích cụ thể những bước mà hàm này thực hiện trong việc tiền xử lí dữ liệu.

Các công đoạn trong hàm prepare-data

- Làm sạch và tách từ câu
- Tokenization bằng một tokenizer BERT
- Loại bỏ stopword

```
from pytorch_pretrained_bert import (  # for Huggingface
   transformer 0.6.2)
        BertTokenizer,
    )
from nltk.corpus import stopwords
```

- Tính trọng số TF-IDF

```
for i in range(n_docs):
    doc_words = shuffled_clean_docs[i]
    words = doc_words.split()
    doc_word_set = set()
    tfidf_vec = []
    for word in words:
        if word in doc_word_set:
            continue
        j = vocab_map[word]
        key = str(i) + "," + str(j)
        tf = doc_word_freq[key]
        tfidf_row.append(i)
        if i < train_size:
            row.append(i)
        else:
            row.append(i + vocab_size)
        tfidf_col.append(j)
        col.append(train_size + j)
        # smooth
        idf = log((1.0 + n_docs) / (1.0 + word_doc_freq[vocab[j]])
           ) + 1.0
```

```
        # weight.append(tf * idf)
        if tfidf_mode == "only_tf":
            tfidf_vec.append(tf)
        else:
            tfidf_vec.append(tf * idf)
        doc_word_set.add(word)
    if len(tfidf_vec) > 0:
        weight.extend(tfidf_vec)
        tfidf_weight.extend(tfidf_vec)
```

- Xây dựng GRAPH dựa trên tính đồng xuất hiện

3 Huấn luyện lại mô hình với tập dữ liệu SST-2

Trong phần này, chúng ta sẽ xem xét cách thiết lập môi trường để huấn luyện và cách sử dụng các hàm được nhóm tác giả công bố để huấn luyện lại mô hình.

3.1 Thiết lập môi trường

Với paper này, chúng ta có thể dễ dàng huấn luyện lại mô hình phiên bản "vanilla" với colab. Các tham số mà nhóm đã thiết lập với môi trường colab trong điều kiện kết nối mạng tốt.

Bảng 1: Cấu hình mô hình

GPU	T4 GPU
EPOCH	9
BATCH SIZE	16
Learning Rate	0.00001
Load Checkpoint	FALSE
MAX_SEQ_LENGTH	200

3.2 Cách thực hiện

Để huấn luyện lại mô hình với dữ liệu sst-2, chúng ta chỉ cần thực hiện câu lệnh sau:

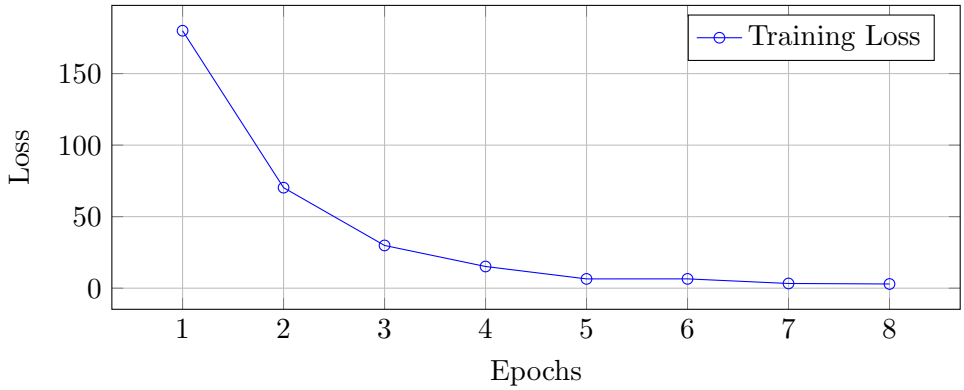
```
!python -m vgcn_bert.train_vanilla_vgcn_bert --ds sst
```

- Để load lại các checkpoint thì chúng ta có thể dùng args "–load 1". Khi đó code sẽ mặc định load checkpoint tại đường dẫn sau: `/output/my_Vanilla_VGCN_BERT1_model_sst_cle_sw0.pt`

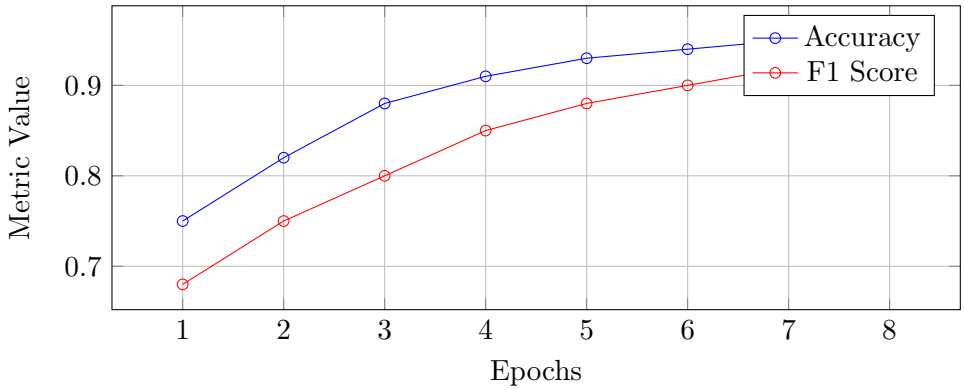
3.3 Quá trình huấn luyện mô hình của nhóm

Thời gian thực hiện huấn luyện mô hình với 9 epochs (đề xuất của nhóm tác giả) mất 45 60 phút. Tuy nhiên nếu tiếp tục huấn luyện thì độ chính xác của mô hình có thể tiếp tục tăng.

Nhóm chúng em sử dụng colab pro nên sẽ không bị giới hạn thời gian kết nối với môi trường colab. Nếu chúng ta chạy colab thường thì cần chú ý đến việc ngắt kết nối do giới hạn tài nguyên. Khi đó checkpoint có thể chưa kịp lưu lại.



Hình 1: Training và Validation Loss qua các Epochs



Hình 2: Accuracy và F1 Score qua các Epochs

4 Đánh giá

Trong phần này, nhóm đã sử dụng tập dữ liệu test có sẵn từ bộ dữ liệu (được đề cập ở trên) và độ đo phổ biến cho bài toán phân loại (precision, recall, F1-score) để đánh giá hiệu suất mô hình.

Trong bài báo, tác giả chủ yếu dùng F1-score để đánh giá tính hiệu quả của mô hình. Vì thể, trong source code mà họ công bố, hàm f1_score từ thư viện scikit-learn được dùng để tính toán F1-score.

```
    f1_metrics = f1_score(
        np.array(all_label_ids).reshape(-1),
        np.array(predict_out).reshape(-1),
        average="weighted",
    )
```

Chúng ta có thể dùng accuracy cho bài toán phân lớp (tiêu cực và tích cực)

```
    ev_acc = correct / total
```

5 Thực nghiệm với các mẫu thử của nhóm

Vì nhóm tác giả không cung cấp inference để người dùng có thể kiểm tra trực tiếp kết quả và họ cũng không công bố các file weigth của mô hình pretrain nên chúng ta cần tự huấn luyện lại

mô hình. Đồng thời nhóm cũng cố gắng code thêm hai file "prepare_data.py" và "test.py" để có thể chạy thử trên những câu đánh giá của chính chúng ta.

- Các file do nhóm thêm vào được đặt trong drive project tại link sau
- Chúng ta cần đặt hai file "prepare_data.py" và "test.py" này vào folder ./vgcn_bert/
- Thực thi hàm MakePredict của nhóm chúng mình để kiểm tra câu đánh giá.

Nhóm tiến hành kiểm tra trên một số mẫu thử như sau:

Comment	**Target Result**	**Model Prediction**
I found the Harry Potter series to be overrated. The plot seemed predictable, and the characters, including the protagonist, lacked depth. The magical world, which initially intrigued me, became repetitive and failed to hold my interest throughout the entire series.	Negative Sentiment	Negative Sentiment
3 Idiots is a film, it is good.	Positive Sentiment	Positive Sentiment
The cast delivers exceptional performances, bringing the characters to life with depth and authenticity.	Positive Sentiment	Positive Sentiment
Good camera with a bad plot film.	Negative Sentiment	Negative Sentiment
The actors perform well, but the scenes are shot too casually.	Negative Sentiment	Negative Sentiment
The actors perform not too well, but the scenes are shot perfectly.	Positive Sentiment	Positive Sentiment

6 Hướng phát triển khác

Trong trường hợp các bạn muốn tạo ra một mô hình để classification trên các tập dữ liệu với nhãn khác. Chúng ta hoàn toàn có thể làm điều này dễ dàng với repo này.

- Chúng ta chỉ cần chuẩn bị tập training dạng txt như nhóm chúng mình đã trình bày ở phần **2.3**
- Các bạn có thể finetune lại cách đánh trọng số bằng countword hay bất kì thuật toán nào mà các bạn muốn bằng cách thay thế hàm TF-IDF trong file prepare_data.py
- Vì nhóm tác giả code from scrath lại một số operator như: TF-IDF, build graph nên chúng ta có thể đọc và hiểu được tương đối dễ dàng.
- Ngoài ra các bạn có thể tham khảo mô hình classification sử dụng RoBERTA model với bộ dữ liệu IBDM trong drive của nhóm mình.