ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

BÁO CÁO BÀI TẬP LỚN
MÔN: CÁCH TIẾP CẬN HIỆN ĐẠI TRONG XỬ LÝ NGÔN NGỮ TỰ NHIÊN - 055256

ASSIGNMENT: SENTIMENT ANALYSIS

Lớp: 1

GVHD: PGS.TS Quản Thành Thơ

Sinh viên thực hiện: Lê Đức Huy - 1810166

TP. HỒ CHÍ MINH, THÁNG 11/2021

MỤC LỤC

DANH SÁCH HÌNH VẼ

1. Giới thiệu bài toán

Sentiment analysis – phân tích cảm xúc khách hàng là một cách tuyệt vời giúp cho các doanh nghiệp hiểu được điểm mạnh, điểm yếu trong sản phẩm, dịch vụ của mình; đồng thời nhanh chóng nắm bắt được tâm ký và nhu cầu khách hàng để mang đến cho họ sản phẩm, dịch vụ hoàn hảo nhất. Chúng ta sẽ dựa trên những thông tin mà khách hàng cung cấp như bình luận, đánh giá, bài chia sẻ để giải bài toán này.

2. Tập dữ liệu và tiền xử lý dữ liệu

2.1 Tập dữ liệu VLSP

Tập dữ liệu được sử dụng trong bài tập lớn này là VLSP. Bao gồm các bình luận về các sản phẩm công nghệ như smartphone, laptop, phụ kiện,… thu thập được từ cuộc thi VLSP năm 2016.

Tập dữ liệu này bao gồm 5100 mẫu huấn luyện về 1050 mẫu kiểm thử. Các đặc điểm của tập dữ liệu được mô tả như sau:

Đặc điểm	Thông số
Ngôn ngữ	Tiếng Việt
Số mẫu huấn luyện	5100
Số mẫu kiểm thử	1050
Độ dài bình luận ngắn nhất	1 từ
Độ dài bình luận tối đa	2624 từ
Độ dài bình luận trung bình	29 từ
Số lượng cảm xúc cần phân lớp	3 loại: negative, neutral, positive
Số từ trong từ điểm (bao gồm cả Unknown)	7919 từ

2.2 Tiền xử lý dữ liệu

Chúng ta sẽ tiến hàng tiền xử lý tập dữ liệu để loại bỏ đi các thành phần không quan trọng, các thành phần gây nhiễu. Các bước được tiến hành như sau:

1/ Loại bỏ các ký tự số

2/ Chuyển tất cả các chữ trong tập dữ liệu thành chữ in thường

3/ Tokenize từng văn bản

4/ Tạo tập từ điển cho tập dữ liệu

5/ Mã hóa các từ bằng index của nó có trong tập từ điển

6/ Những văn bản dài hơn 300 từ sẽ được cắt bỏ xuống còn đúng 300 từ

3. Các mô hình huấn luyện

Tham số dùng chung cho tất cả mô hình:

Tham số	Giá trị
Độ dài câu tối đa	300 từ
Số chiều vector embedding của mỗi từ	400
Mô hình Word2Vec	CBOW
Kích thước cửa sổ Word2Vec	5

3.1 Mô hình CNN

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 300)]        0           []

 embedding (Embedding)          (None, 300, 400)     3167600     ['input_1[0][0]']

 reshape (Reshape)              (None, 300, 400, 1)  0           ['embedding[0][0]']

 conv2d (Conv2D)                (None, 298, 1, 100)  120100      ['reshape[0][0]']

 conv2d_1 (Conv2D)              (None, 297, 1, 100)  160100      ['reshape[0][0]']

 conv2d_2 (Conv2D)              (None, 296, 1, 100)  200100      ['reshape[0][0]']

 max_pooling2d (MaxPooling2D)   (None, 1, 1, 100)    0           ['conv2d[0][0]']

 max_pooling2d_1 (MaxPooling2D)  (None, 1, 1, 100)   0           ['conv2d_1[0][0]']

 max_pooling2d_2 (MaxPooling2D)  (None, 1, 1, 100)   0           ['conv2d_2[0][0]']

 concatenate (Concatenate)      (None, 3, 1, 100)    0           ['max_pooling2d[0][0]',
                                                                  'max_pooling2d_1[0][0]',
                                                                  'max_pooling2d_2[0][0]']

 flatten (Flatten)              (None, 300)          0           ['concatenate[0][0]']

 dropout (Dropout)              (None, 300)          0           ['flatten[0][0]']

 dense (Dense)                  (None, 3)            903         ['dropout[0][0]']

==================================================================================================
Total params: 3,648,803
Trainable params: 3,648,803
Non-trainable params: 0
```

Hình 1: Mô hình Sentiment Analysis dùng CNN

Tính toán số lượng tham số ở mỗi tầng:

- Tầng embedding: vocab_size * embedding_dim = 7919 * 400 = **3167600**

- Tầng CNN với 100 filter, mỗi filter có chiều dài là 3, số chiều vector embedding là 400:

num_filters * (filter_height * filter_width + bias)

= 100 * (3 * 400 + 1) = **120100**

- Tầng CNN với 100 filter, mỗi filter có chiều dài là 4, số chiều vector embedding là 400:

num_filters * (filter_height * filter_width + bias)

= 100 * (4 * 400 + 1) = **160100**

- Tầng CNN với 100 filter, mỗi filter có chiều dài là 5, số chiều vector embedding là 400:

num_filters * (filter_height * filter_width + bias)

= 100 * (5 * 400 + 1) = **200100**

- Tầng fully-connected với đầu vào là vector đặc trưng 300 chiều và đầu ra là 3 lớp:

num_classes * (input_dim + bias) = 3 * (300 + 1) = **903**

3.2 Mô hình LSTM

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_5 (InputLayer)        [(None, 300)]             0

 embedding (Embedding)       (None, 300, 400)          3167600

 lstm_3 (LSTM)               (None, 512)               1869824

 flatten_2 (Flatten)         (None, 512)               0

 dropout_1 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 3)                 1539

=================================================================
Total params: 5,038,963
Trainable params: 5,038,963
Non-trainable params: 0
```

Hình 2: Mô hình Sentiment Analysis dùng LSTM

Tính toán số lượng tham số ở mỗi tầng:

- Tầng embedding: vocab_size * embedding_dim = 7919 * 400 = **3167600**

- Tầng LSTM: với số chiều vector hidden là 512:

4 * (input_dim * hidden_dim + hidden_dim * hidden_dim + bias * hidden_dim)

= 4 * (400 * 512 + 512 * 512 + 1 * 512) = **1869824**

- Tầng fully-connected với đầu vào là vector đặc trưng 512 chiều và đầu ra là 3 lớp:

num_classes * (input_dim + bias) = 3 * (512 + 1) = **1539**

3.3 Mô hình kết hợp CNN + LSTM

```
Model: "model_2"
_________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
=================================================================================================
 input_6 (InputLayer)           [(None, 300)]        0           []

 embedding (Embedding)          (None, 300, 400)     3167600     ['input_6[0][0]']

 reshape_6 (Reshape)            (None, 300, 400)     0           ['embedding[5][0]']

 conv1d (Conv1D)                (None, 300, 100)     120100      ['reshape_6[0][0]']

 conv1d_1 (Conv1D)              (None, 300, 100)     160100      ['reshape_6[0][0]']

 conv1d_2 (Conv1D)              (None, 300, 100)     200100      ['reshape_6[0][0]']

 concatenate_1 (Concatenate)    (None, 300, 300)     0           ['conv1d[0][0]',
                                                                  'conv1d_1[0][0]',
                                                                  'conv1d_2[0][0]']

 lstm_4 (LSTM)                  (None, 512)          1665024     ['concatenate_1[0][0]']

 dropout_2 (Dropout)            (None, 512)          0           ['lstm_4[0][0]']

 dense_2 (Dense)                (None, 3)            1539        ['dropout_2[0][0]']

=================================================================================================
Total params: 5,314,463
Trainable params: 5,314,463
Non-trainable params: 0
```

Hình 3: Mô hình Sentiment Analysis dùng CNN + LSTM

Tính toán số lượng tham số ở mỗi tầng:

- Tầng embedding: vocab_size * embedding_dim = 7919 * 400 = **3167600**

- Tầng CNN với 100 filter, mỗi filter có chiều dài là 3, số chiều vector embedding là 400:

num_filters * (filter_height * filter_width + bias)

= 100 * (3 * 400 + 1) = **120100**

- Tầng CNN với 100 filter, mỗi filter có chiều dài là 4, số chiều vector embedding là 400:

num_filters * (filter_height * filter_width + bias)

= 100 * (4 * 400 + 1) = **160100**

- Tầng CNN với 100 filter, mỗi filter có chiều dài là 5, số chiều vector embedding là 400:

num_filters * (filter_height * filter_width + bias)

= 100 * (5 * 400 + 1) = **200100**

- Tầng LSTM: với số chiều vector đặc trưng đầu vào là 300, vector hidden là 512:

4 * (input_dim * hidden_dim + hidden_dim * hidden_dim + bias * hidden_dim)

= 4 * (300 * 512 + 512 * 512 + 1 * 512) = **1665024**

- Tầng fully-connected với đầu vào là vector đặc trưng 512 chiều và đầu ra là 3 lớp:

num_classes * (input_dim + bias) = 3 * (512 + 1) = **1539**

4. Kết quả thí nghiệm

Chúng tôi đã tiến hành thử nghiệm tập dữ liệu VLSP với 3 mô hình đã trình bày ở Mục 3. Kết quả tổng hợp được như sau:

Mô hình	**Độ chính xác**
CNN	59.52%
LSTM	61.24%
CNN + LSTM	62.76%

Với kết quả trên, ta thấy độ chính xác tăng dần từ CNN, đến LSTM và cuối cùng là mô hình kết hợp CNN + LSTM. Từ đó, ta rút ra được kết luận rằng sự kết hợp giữa 2 môn hình CNN và LSTM với nhau đã có hiệu quả, giúp tăng độ chính xác so với các mô hình chỉ dùng đơn thuần CNN hay LSTM. Tuy nhiên, cả 3 mô hình cho thấy sự chênh lệch độ chính xác không quá cao, chỉ từ 1% - 3%. Đây là tỉ lệ chỉ có ý nghĩa tăng trên mặt lý thuyết, còn trên thực tế nếu chỉ tăng từ 1% - 3% thì là con số rất nhỏ, sẽ không thể hiện được sự cải tiến vượt bậc.