ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

CS221.N21.KHTN

Huấn luyện mô hình Logistic Regression cho bài toán Phân tích cảm xúc theo yếu tố trên tập dữ liệu Tiếng Việt

Nhóm : 5
GV hướng dẫn: TS. Nguyễn Thị Quý

Tp.HCM ngày 18 tháng 04 năm 2023

Contents

1 Giới thiệu

1.1 Giới thiệu bài toán

Aspect-Based Sentiment Analysis (ABSA) là một bài toán phức tạp trong lĩnh vực xử lý ngôn ngữ tự nhiên, chúng ta phải phân tích ý kiến của người dùng về một sản phẩm hoặc dịch vụ dựa trên những khía cạnh (aspect) của sản phẩm đó. Nhiệm vụ chính của ABSA là phân loại những đánh giá thành các lớp cảm xúc tương ứng với mỗi aspect.

Một trong những phương pháp hiệu quả để giải quyết bài toán ABSA là sử dụng mô hình Logistic Regression. Với mô hình này, chúng ta có thể xây dựng một bộ phân loại dựa trên xác suất để phân loại những đánh giá vào các lớp cảm xúc tương ứng. Các mô hình Logistic Regression cho ABSA được xây dựng trên các khía cạnh (aspect) cụ thể của sản phẩm hoặc dịch vụ, với mỗi khía cạnh có thể có nhiều thuộc tính (attribute).

Trong báo cáo này, chúng tôi sẽ trình bày phương pháp sử dụng mô hình Logistic Regression để giải quyết bài toán ABSA trên tập dữ liệu về khách sạn và nhà hàng. Ngoài ra, chúng tôi cũng đề xuất một cải tiến cho phương pháp này bằng cách thêm một lớp cảm xúc mới (not mentioned) để thể hiện những khía cạnh (aspect) không được đề cập trong đánh giá của người dùng.

1.2 Dataset VLSP2018-SA

1.2.1 Các loại nhãn

A. Nhãn sentiment: positive, negative hoặc neutral.

B. Nhãn aspect: có dạng [entity]#[attribute]

- Với dữ liệu Hotel:
  - Có 7 loại entity: HOTEL, ROOMS, ROOMS_AMENITIES, FACILITIES, SERVICE, LOCATION, FOOD&DRINKS.
  - Có 8 loại attribute: GENERAL, PRICES, DESIGNFEATURES, CLEANLINESS, COMFORT, QUALITY, STYLEOPTIONS, MISCELLANEOUS.

- Với dữ liệu Restaurant:
  - Có 6 loại entity: restaurant, ambience, location, food, service, drinks.
  - Có 5 loại attribute: general, quality, price, style_option, miscellaneous.

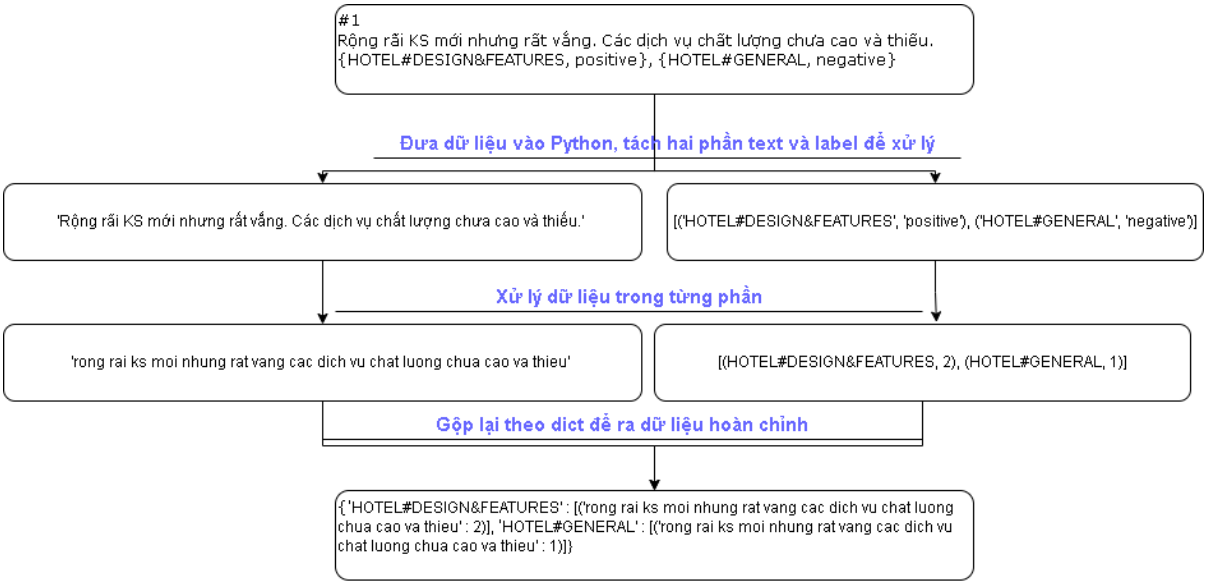
1.2.2 Dữ liệu mẫu

#1
Rộng rãi KS mới nhưng rất vắng. Các dịch vụ chất lượng chưa cao và thiếu.
{HOTELDESIGNFEATURES, positive}, {HOTELGENERAL, negative}

#2
Địa điểm thuận tiện, trong vòng bán kính 1,5km nhiều quán ăn ngon
{LOCATIONGENERAL, positive}
...

2 Phương pháp thực hiện

2.1 Chuẩn bị dữ liệu



Các bước chuẩn bị dữ liệu

2.2 Tiền xử lý dữ liệu

Tiền xử lý là giai đoạn làm cho dữ liệu dễ dàng hơn hoặc phù hợp để sử dụng trong quá trình khai thác . Tiền xử lý được thực hiện với mục đích thống nhất và dễ đọc cũng như quá trình phân loại. Việc xử lý văn bản được thực hiện bằng một số thao tác sau:

- Đầu vào là đường dẫn tới file văn bản cần xử lý.
- Chuyển tất cả các ký tự thành chữ thường.
- Thay thế các ký tự tiếng Việt có dấu bằng các ký tự tương ứng không dấu.
- Loại bỏ tất cả các ký tự không phải là chữ cái hoặc khoảng trắng.

```python
def process_raw_dataset(text):
    '''
    Xử lí các vấn đề sau:
    - Viết hoa thường lẫn lộn
    - Viết dấu, không dấu lẫn lộn
    - Xoá số, dấu câu, kí tự đặc biệt
    '''
    text = text.lower()
    VNalpha = 'àáãạảăắằẳẵặâấầẩẫậèéẹẻẽêềếểễệđìíĩỉịòóõọỏôốồổỗộ
    VNkodau = 'aaaaaaaaaaaaaaaaaeeeeeeeeeeeediiiiioooooooooo
    for idx in range(len(VNalpha)):
        text = text.replace(VNalpha[idx], VNkodau[idx])
    text = re.sub(r'[^(a-zA-Z )]+', '', text)
    return text
```

```python
process_raw_dataset("Hôm nay, tôi ĐI HỌC!")
```

```
'hom nay toi di hoc'
```

2.3 Gom nhóm dữ liệu theo label

Để gom nhóm dữ liệu theo label, đầu tiên ta duyệt qua từng cặp (text, label) trong dữ liệu đã qua tiền xử lý. Trong quá trình này, ta tìm danh sách các aspect và sentiment từ dữ liệu. Nếu aspect của một tuple (text, label) không có trong danh sách aspect đã tìm được, nó sẽ được coi là **not mentioned**. Khi đó, nhãn sẽ được đặt bằng 0. Nếu aspect và sentiment của tuple (text, label) có trong danh sách aspect và sentiment đã tìm được, nhãn sẽ được đặt bằng số tương ứng với sentiment (**1: negative, 2: positive, 3: neutral**).

2.4 Trích xuất đặc trưng

Để rút trích đặc trưng của các văn bản trong bộ dữ liệu, ta sử dụng class **CountVectorizer** của thư viện **Scikit-learn**. Class này giúp biến đổi các văn bản thành các vectơ đặc trưng bằng phương pháp **Bag-of-words**.

```
corpus = [text for txt in list(data.values()) for text in list(txt.keys())]
vectorizer = CountVectorizer()
vectorizer.fit_transform(corpus)
```

2.5 Xây dựng và huấn luyện model

Trước đi bắt đầu xây dựng mô hình, load dữ liệu xử lý theo các hàm được cài đặt trong phần tiền xử lý dữ liệu trước đó.

Với mỗi aspect (khía cạnh đánh giá), sử dụng **Logistic Regression** để huấn luyện một mô hình với các features được trích xuất bằng **CountVectorizer** từ các văn bản tương ứng với aspect đó trong tập dữ liệu và dự đoán sentiment của chúng (0, 1, 2, hoặc 3 tương ứng với not mentioned, negative, positive, neutral).

```
asp_lst = get_aspect_list(processed_txt_data)
model = {}
for asp in asp_lst:
    model[asp] = LogisticRegression(verbose=0, solver='liblinear',random_state=0,
        C=5, penalty='l2',max_iter=100)
    x = [txt for txt, label in data[asp].items()]
    y = [label for txt, label in data[asp].items()]
    x = vectorizer.transform(x)
    model[asp].fit(x.toarray(), y)


def predict(text, model, vectorizer, asp_lst):

    result = []
    asp_map = {1: 'negative', 2: 'positive', 3: 'neutral'}
    for asp in asp_lst:
        pred = model[asp].predict([vectorizer.transform([text]).toarray()[0]])[0]
        if pred != 0:
            result.append((asp, asp_map[pred]))
    return result
```

Hàm trả về các gía trị True Positive giữa giá tri dự đoán (pred) và giá trị thật (truth) cùng với tính toán giá trị F1

```
def check_pred(pred, truth):

    TP = len(list(set(pred) & set(truth)))
    return TP
def check_pred_asp(pred, truth):

    pred = [asp for asp, sent in pred]
    truth = [asp for asp, sent in truth]
    TP = len(list(set(pred) & set(truth)))
    return TP
def F1(TP, pred, truth):

    precision = TP / pred
    recall = TP / truth
    return 2 * precision * recall / (precision + recall)
```

3 Đánh giá mô hình

Để đánh giá hiệu suất của mô hình trên các nhãn, ta cần tính toán các chỉ số precision, recall và F1-score cho mỗi nhãn.

TẬP TEST

	Precision	Recall	F1
SERVICE#GENERAL	0.830882	0.814904	0.822816
ROOM_AMENITIES#CLEANLINESS	0.529412	0.195652	0.285714
LOCATION#GENERAL	0.817308	0.769231	0.792541
ROOMS#CLEANLINESS	0.661538	0.645000	0.653165
ROOMS#COMFORT	0.405405	0.322581	0.359281
ROOMS#GENERAL	0.439024	0.315789	0.367347
HOTEL#GENERAL	0.649254	0.576159	0.610526
ROOMS#DESIGN&FEATURES	0.503597	0.353535	0.415430
ROOM_AMENITIES#DESIGN&FEATURES	0.528571	0.256944	0.345794
HOTEL#CLEANLINESS	0.370787	0.492537	0.423077
ROOM_AMENITIES#COMFORT	0.566667	0.250000	0.346939
ROOM_AMENITIES#GENERAL	0.419355	0.406250	0.412698
ROOM_AMENITIES#QUALITY	0.317073	0.220339	0.260000
FOOD&DRINKS#STYLE&OPTIONS	0.610000	0.491935	0.544643
ROOMS#QUALITY	0.125000	0.100000	0.111111
FACILITIES#DESIGN&FEATURES	0.437500	0.215385	0.288660
FACILITIES#QUALITY	0.500000	0.196078	0.281690
HOTEL#DESIGN&FEATURES	0.452055	0.388235	0.417722
HOTEL#PRICES	0.547945	0.563380	0.555556
HOTEL#QUALITY	0.000000	0.000000	NaN
ROOMS#PRICES	0.222222	0.206897	0.214286
FACILITIES#GENERAL	0.250000	0.238095	0.243902
HOTEL#COMFORT	0.446602	0.489362	0.467005
HOTEL#MISCELLANEOUS	0.500000	0.073529	0.128205
FOOD&DRINKS#QUALITY	0.590164	0.558140	0.573705
FACILITIES#MISCELLANEOUS	NaN	0.000000	NaN
FACILITIES#COMFORT	0.384615	0.192308	0.256410
FACILITIES#PRICES	0.000000	0.000000	NaN
FOOD&DRINKS#MISCELLANEOUS	NaN	0.000000	NaN
FOOD&DRINKS#PRICES	0.000000	0.000000	NaN
FACILITIES#CLEANLINESS	0.200000	0.200000	0.200000
ROOMS#MISCELLANEOUS	NaN	0.000000	NaN
ROOM_AMENITIES#MISCELLANEOUS	NaN	0.000000	NaN
ROOM_AMENITIES#PRICES	NaN	0.000000	NaN

	Precision	Recall	F1
negative	0.464646	0.285271	0.353506
positive	0.638247	0.596899	0.616881
neutral	0.117647	0.015038	0.026667

F1: 0.5394793000426803

Kết quả đánh giá bộ dữ liệu chủ đề khách sạn

Đối với bộ dữ liệu chủ đề khách sạn, xét theo các khía cạnh (aspect), mô hình logistics regression đạt kết quả rất tốt trong việc phân loại nhãn ở hai khía cạnh **SERVICE#GENERAL** và **LOCATION#GENERAL** với F1-score lần lượt là **0.82** và **0.79**. Xét theo 3 nhãn phân loại, mô hình đạt được F1-score tốt nhất ở nhãn **positive** với giá trị là **0.62**.

```
TẬP TEST
                          Precision    Recall        F1
DRINKS#QUALITY             0.666667  0.366197  0.472727
RESTAURANT#GENERAL         0.533088  0.650224  0.585859
DRINKS#STYLE&OPTIONS       0.459459  0.369565  0.409639
FOOD#QUALITY               0.816239  0.835886  0.825946
DRINKS#PRICES              0.291667  0.092105  0.140000
FOOD#STYLE&OPTIONS         0.727735  0.709677  0.718593
RESTAURANT#PRICES          0.239437  0.232877  0.236111
AMBIENCE#GENERAL           0.608108  0.529412  0.566038
SERVICE#GENERAL            0.459459  0.388571  0.421053
FOOD#PRICES                0.418831  0.389728  0.403756
LOCATION#GENERAL           0.419355  0.290503  0.343234
RESTAURANT#MISCELLANEOUS   0.181818  0.030769  0.052632
           Precision    Recall        F1
positive    0.643468  0.675854  0.659264
neutral     0.404605  0.208122  0.274860
negative    0.239437  0.106918  0.147826
F1: 0.5577303716736309
```

Kết quả đánh giá bộ dữ liệu chủ đề nhà hàng

Đối với bộ dữ liệu chủ đề khách sạn, xét theo các khía cạnh (aspect), mô hình logistics regression đạt kết quả rất tốt trong việc phân loại nhãn ở hai khía cạnh **FOOD#QUALITY** và **FOOD#STYLEOPTIONS** với F1-score lần lượt là **0.83** và **0.72**. Xét theo 3 nhãn phân loại, mô hình đạt được F1-score tốt nhất ở nhãn **positive** với giá trị là **0.66**.

Nhận xét chung:

- Ở cả hai bộ dữ liệu theo chủ đề khách sạn và chủ đề nhà hàng, mô hình logistic regression đều đạt được giá trị F1-score tương đối tốt, lần lượt là **0.54** và **0.56**.
- Ở cả hai bộ dữ liệu, mô hình đạt được F1-score ở nhãn positive khá ấn tượng, trên **0.6**. Tuy nhiên, mô hình vẫn chưa thể hiện tốt ở 2 nhãn **neutral** và **negative**.
- Tóm lại, mô hình vẫn còn có phần chưa hoàn thiện và cần được cải thiện thêm. Có thể cần phải xem xét lại cách tiền xử lý dữ liệu, tăng cường số lượng dữ liệu huấn luyện và cải thiện kiến trúc mô hình để đạt được hiệu quả tốt hơn.

4 Kết luận

Mô hình Logistic Regression làm khá ổn trong bài toán Aspect-based Sentiment Analysis. Trong bài báo cáo trước đó, chúng tôi đã huấn luyện mô hình Naive Bayes cho cùng tập dữ liệu.
Bảng kết quả sau đây so sánh kết quả dự đoán của hai mô hình với cùng bài toán trên

- Mô hình Naive Bayes

Dataset		Precision	Recall	F1
Hotel	positive	0.599606	0.539492	0.567963
	negative	0.361722	0.218473	0.272414
	neutral	0.111111	0.008850	0.016393
Restaurant	positive	0.599606	0.539492	0.567963
	negative	0.361722	0.218473	0.272414
	neutral	0.111111	0.008850	0.016393

- Mô hình Logistic Regression

Dataset		Precision	Recall	F1
Hotel	positive	0.638247	0.596899	0.616881
	negative	0.464646	0.285271	0.353506
	neutral	0.117647	0.015038	0.026667
Restaurant	positive	0.643468	0.675854	0.659264
	negative	0.239437	0.106918	0.147826
	neutral	0.404605	0.208122	0.274860

- Dánh giá: Mô hình Logistic Regression hoạt động tốt hơn trong đa số trường hợp ở cả hai dataset. Từ đó mang đến hiệu quả phân tích quan điểm tốt hơn cho bài toán nói trên.