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

CS221.N21.KHTN

Naive Bayes Classifier for Aspect-Based Sentiment Analysis on a Vietnamese Dataset

Nhóm : 5
GV hướng dẫn : Nguyễn Thị Quý

Tp.HCM ngày 05 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 Naive Bayes Classifier. 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 Naive Bayes Classifier 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 Naive Bayes Classifier 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.

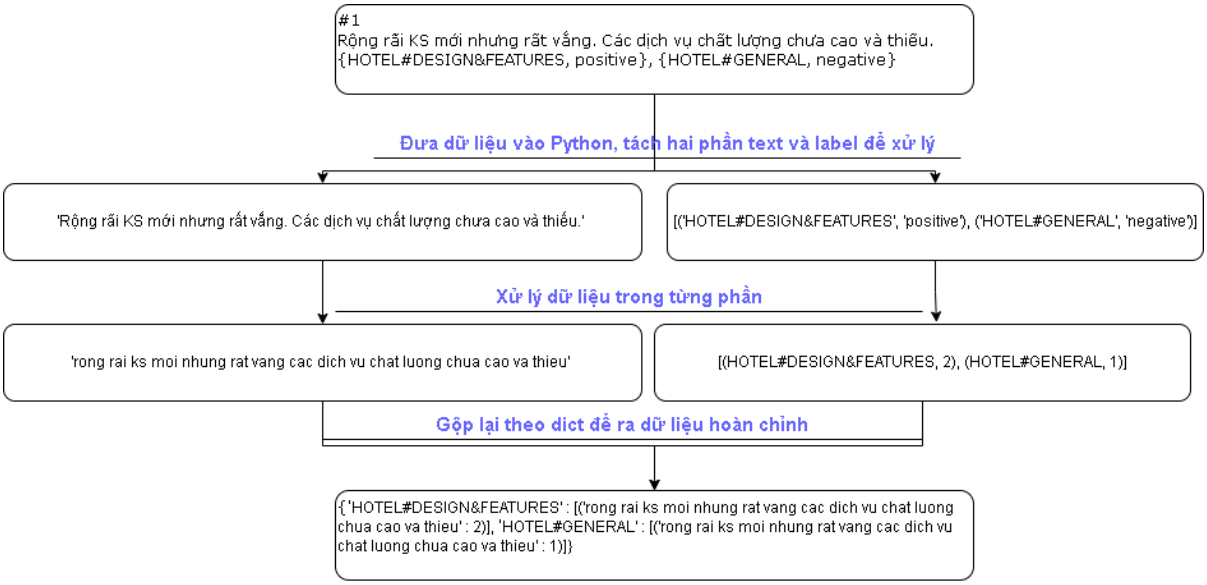
1.2.2 Dữ liệu mẫu

#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.

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

Với mỗi aspect, sử dụng **Multinomial Naive Bayes** 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] = MultinomialNB()
    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
```

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.

- Bộ dữ liệu khách sạn

TẬP TEST

	Precision	Recall	F1
SERVICE#GENERAL	0.760776	0.848558	0.802273
ROOMS#CLEANLINESS	0.570707	0.565000	0.567839
ROOM_AMENITIES#CLEANLINESS	1.000000	0.065217	0.122449
LOCATION#GENERAL	0.704846	0.723982	0.714286
ROOMS#COMFORT	0.473684	0.193548	0.274809
ROOMS#GENERAL	0.714286	0.087719	0.156250
ROOMS#DESIGN&FEATURES	0.360000	0.272727	0.310345
ROOM_AMENITIES#DESIGN&FEATURES	0.444444	0.194444	0.270531
ROOM_AMENITIES#COMFORT	NaN	0.000000	NaN
HOTEL#CLEANLINESS	0.444444	0.179104	0.255319
ROOM_AMENITIES#GENERAL	0.500000	0.031250	0.058824
ROOM_AMENITIES#QUALITY	0.235294	0.135593	0.172043
FOOD&DRINKS#STYLE&OPTIONS	0.595506	0.427419	0.497653
ROOMS#QUALITY	0.000000	0.000000	NaN
FACILITIES#DESIGN&FEATURES	0.363636	0.123077	0.183908
FOOD&DRINKS#QUALITY	0.595041	0.558140	0.576000
FACILITIES#QUALITY	0.600000	0.058824	0.107143
HOTEL#DESIGN&FEATURES	0.460000	0.270588	0.340741
HOTEL#PRICES	0.666667	0.338028	0.448598
HOTEL#QUALITY	0.000000	0.000000	NaN
ROOMS#PRICES	NaN	0.000000	NaN
HOTEL#GENERAL	0.829545	0.483444	0.610879
HOTEL#COMFORT	0.468085	0.468085	0.468085
FACILITIES#GENERAL	NaN	0.000000	NaN
HOTEL#MISCELLANEOUS	0.000000	0.000000	NaN
FACILITIES#MISCELLANEOUS	NaN	0.000000	NaN
FACILITIES#COMFORT	NaN	0.000000	NaN
FOOD&DRINKS#MISCELLANEOUS	NaN	0.000000	NaN
FACILITIES#PRICES	NaN	0.000000	NaN
FOOD&DRINKS#PRICES	NaN	0.000000	NaN
FACILITIES#CLEANLINESS	NaN	0.000000	NaN
ROOMS#MISCELLANEOUS	NaN	0.000000	NaN
ROOM_AMENITIES#MISCELLANEOUS	NaN	0.000000	NaN
ROOM_AMENITIES#PRICES	NaN	0.000000	NaN

	Precision	Recall	F1
negative	0.412121	0.210853	0.278974
positive	0.660201	0.508859	0.574734
neutral	0.000000	0.000000	NaN

F1: 0.48990016252612023

- Bộ dữ liệu nhà hàng

```
TẬP TEST
                          Precision    Recall        F1
FOOD#QUALITY               0.810865  0.881838  0.844864
FOOD#STYLE&OPTIONS         0.727723  0.729529  0.728625
DRINKS#STYLE&OPTIONS       1.000000  0.021739  0.042553
DRINKS#QUALITY             1.000000  0.014085  0.027778
RESTAURANT#GENERAL         0.554878  0.408072  0.470284
DRINKS#PRICES                   NaN  0.000000       NaN
RESTAURANT#MISCELLANEOUS        NaN  0.000000       NaN
LOCATION#GENERAL           1.000000  0.005587  0.011111
FOOD#PRICES                0.432432  0.145015  0.217195
AMBIENCE#GENERAL           0.855556  0.301961  0.446377
SERVICE#GENERAL            0.628205  0.280000  0.387352
RESTAURANT#PRICES          1.000000  0.013699  0.027027
          Precision    Recall        F1
positive   0.723795  0.566806  0.635753
neutral    0.475000  0.032149  0.060222
negative   1.000000  0.006289  0.012500
F1: 0.5128749668170959
```

=> Nhận xét :

- Dựa vào 2 tập test, mô hình đạt được kết quả tốt trong việc phân loại một số nhãn như FOOD#QUALITY, FOOD#STYLEOPTIONS, và SERVICE#GENERAL, với F1-score trên 0.7. Tuy nhiên, mô hình cũng không hoạt động hiệu quả trong việc phân loại các nhãn như DRINKS#STYLEOPTIONS, DRINKS#QUALITY và LOCATION#GENERAL. Các nhãn này có F1-score thấp hơn 0.5 hoặc không xác định được giá trị F1-score. Điều này cho thấy rằng mô hình chưa hoàn thiện và cần được cải thiện để có thể phân loại đầy đủ các nhãn một cách chính xác.

- Model cũng cho điểm F1 khá tốt ở nhãn positive, tuy nhiên nhãn negative và neutral cho kết quả khá thấp.

- Tóm lại, điều này cho thấy rằng 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 Naive Bayes làm khá ổn trong bài toán Aspect-based Sentiment Analysis. Với độ chính xác đạt được, có thể kết luận rằng, mô hình ít nhiều đã học được những đặc trưng và cho ra những dự đoán ổn. Tuy nhiên, do tính đơn giản của nó và việc giả định rằng các đặc trưng là độc lập mà Naive Bayes vẫn chưa thật sự nổi bật trong bài toán.