REPORT

DETECT SARA COMMENTS AUTOMATICALLY

Thực hiện : Lê Trọng Tuân

Hồ Chí Minh, ngày 14 tháng 4 năm 2019

DECTECT SARA COMMENTS AUTOMATICALLY IN INDONESIAN LANGUAGE

Nội dung

I. MÔ TẢ BÀI TOÁN :

1. GIỚI THIỆU :

Ứng dụng WeVojiBabe đây là ứng dụng tin tức phổ biến nhất ở Indonesia. Có hàng ngàn bình luận mới về ứng dụng WeVojiBabe mỗi ngày. Nhưng nhiều ý kiến là bình luận của Sara. Bình luận của Sara là những bình luận có nội dung phân biệt tôn giáo và báng bổ.

2. YÊU CẦU:

Từ các tài nguyên dữ liệu sau :

- Từ điển Indonesia : Danh sách 202115 từ phổ biến nhất trong tiếng Indonesia.
- Từ vô nghĩa trong tiếng Indonesia : Danh sách gồm 7224 từ vô nghĩa trong tiếng Indonesia.
- Dữ liệu training : Gồm có 60392 bình luận hợp lệ và 14908 bình luận là của Sara.
- Dữ liệu test: Gồm có 6830 bình luận hợp lệ và 2816 bình luận của Sara.

Hãy phát triển một chương trình máy tính để tự động nhận ra các bình luận của Sara bằng ngôn ngữ lập trình Python hoặc Scala hoặc Java.

II. PHƯƠNG PHÁP GIẢI QUYẾT BÀI TOÁN:

1. Ý TƯỞNG:

Ta dễ dàng nhận ra bài toán trên thuộc dạng bài toán phân lớp.

Các bước xử lý bái toán như sau :

- Đặc trưng hóa dữ liệu :

- Tính xác suất mỗi từ trong trường hợp là bình luận hợp lệ và trường hợp là bình luận của Sara.
- Từ đó ta có thể tính được xác suất của bình luận trong 2 trường hợp.
- Xác suất nào lớn hơn,thì đó là kết quả.

2. THUẬT TOÁN SỬ DỤNG:

Thuật toán tôi sử dụng để giải quyết bài toán này là Naive Bayes Classifier.

Mô tả thuật toán **Naive Bayes Classifier**:

- Với các bài toán phân lớp dữ liệu cho c class khác nhau,thay vì đi tìm chính xác label của mỗi điểm dữ liệu,ta có thể đi tìm xác suất của mỗi điểm dữ liệu rơi vào các class : P(c|x). Biểu thức này thể hiện xác suất của điểm dữ liệu x thuộc class c,với đầu vào x là một vector được xử lý từ dữ liệu ban đầu. Và label của điểm dữ liệu x sẽ là class có xác suất x rơi vào cao nhất :

$$c = \arg\max_{c \in \{1,..,C\}} p(c \mid x)$$

- Dùng quy tắc Bayes biến đổi công thức trên :

$$c = \arg\max_{c} \text{p(c|x)} = \arg\max_{c} \frac{p(x \mid c).p(c)}{p(x)} = \arg\max_{c} p(x \mid c).p(c)$$

- Với p(x|c) chúng ta sẽ tính dựa trên mô hình phân bố xác suất **Multinomial naive Bayes** :

$$\lambda_{ci} = p(x_i \mid c) = \frac{N_{ci}}{N_c}$$

Trong đó : N_{ci} là tổng số lần từ thứ i (trong dictionary) xuất hiện trong văn bản của class c.

N_c là tổng số từ (kể cả lặp) xuất hiện trong class c.

λ_{ci} là xác suất từ thứ i (trong dictionary) xuất hiện trong class c.

Nhưng với công thức trên,khi một từ trong từ điển không xuất hiện trong class c thì sẽ làm cho kết quả phép tính xác suất cho một câu bằng 0,dẫn đến kết quả sai. Để khắc phụ điều này,ta dùng kỹ thuật **Laplace smoothing :**

$$\lambda_{ci}^{'} = \frac{N_{ci} + \alpha}{N_c + d.\alpha}$$

Trong đó : d là số kích thước của từ điển (số từ trong từ điển).

α thường được lấy là 1, có ý nghĩa thêm mỗi từ trong từ điển 1 lần vào class c.

III. HIỆN THỰC THUẬT TOÁN :

1. BƯỚC XỬ LÝ DỮ LIỆU :

- Viết hàm tạo label cho dữ liệu (tranning set lẫn test set) : Hàm nhận vào file normal.txt và sara.txt, trả về list có giá trị 0 hoặc 1 (0: khi hàng dữ liệu thuộc tập normal,1 khi hàng dữ liệu thuộc tập sara).
- Viết hàm tạo list stop words : Hàm nhận vào file stop_words.txt và trả về một biến kiểu list gồm có các stop words có trong file.
- Viết hàm tạo dictionary : Hàm nhận vào file id_full.txt và trả về một biến kiểu list gồm các từ có trong file.

- Xử lý dữ liệu(các dòng bình luận trong các file normal và sara): Nhận vào file normal và sara, trả về file features.txt với mỗi dòng sẽ có 3 giá trị :
  <số thứ tự dòng > <thứ tự từ xét (trong dictionary)> <tần suất từ đó trong dòng bình luận >
  - Loại bỏ ký tự đặc biệt và số có trong dòng bình luận.
  - Lọc stop words cho dòng bình luận.
  - Tách dòng bình luận thành các từ riêng lẽ.
  - Sắp xếp từ theo thứ tự [a➔z]
  - Tạo biến kiểu dictionary với : key: từ , value : tần suất từ trong dòng bình luận.
  - Với từng key trong biến trên ta sẽ tìm id trong biến dictionary rồi viết ra file đặc trưng.
- Viết hàm nhận vào file dữ liệu đã xử lý để hình thành ma trận thưa thớt (Sparse Matrix).

2. DÙNG THƯ VIỆN SKLEARN ĐỂ BUILD MODEL:

- Dùng hàm **fit , predict** trong class **MultinomialNB** để chạy model.
- Dùng hàm **accuracy_score** trong thư viện **Sklearn.matrics** để tính độ chính xác của mô hình.

IV. KẾT QUẢ :

1. ĐỘ CHÍNH XÁC CỦA MÔ HÌNH :

```
Training size = 75300, accuracy = 82.17%
PS C:\Users\trtua\Downloads\Test>
```

V. THẢO LUẬN:

1. Naive Bayes Classifier thường được sử dụng cho các bài toán phân loại văn bản vì hiệu quả tương đối cao,và dường như đơn giản dễ thực hiện.
2. Trong bài toán này tôi đã kiểm tra mô hình bằng phân phối **Multinomial naive Bayes** và **Bernoulli Naive Bayes** nhưng kết quả của mô hình **Bernoulli Naive Bayes** thấp hơn rất nhiều nên trong báo cáo tôi chỉ mô tả sử dụng mô hình **Multinomial naive Bayes**.
3. Đối với dictionary trong data source bài toán này thì không cần phải thực hiện bước **stemming** cho các từ,vì dictionary hầu như đã bao phủ hết các trường hợp, tôi đã dùng thư viện **Sastrawi** (thư viện stemming cho các từ trong tiếng Indonesia) để kiểm tra.

VI. DANH SÁCH TÀI LIỆU THAM KHẢO:

1. Machine Learning cơ bản – Vũ Hữu Tiệp
2. Thư viện Stemming từ trong tiếng Indonesia (https://pypi.org/project/Sastrawi/)
3. Natural Language Toolkit (https://www.nltk.org/)
4. Thư viện Scikit learn cho thuật toán Naive Bayes Classifier (https://scikit-learn.org/stable/modules/naive_bayes.html).