PHÂN TÍCH DỮ LIỆU DẠNG VĂN BẢN VỚI NLTK

Nội dung

- Giới thiệu về thư viện NLTK
- Tìm 1 từ với NLTK
- Phân tích tần số của các từ
- Lựa chọn các từ trong văn bản
- Bigrams và collocations
- Sử dụng văn bản trên mạng
- Rút trích văn bản từ trang html
- Phân tích cảm xúc người dùng

Giới thiệu về thư viện NLTK

- Trong các phần trước, ta đã học cách phân tích dữ liệu dạng số hoặc dạng bảng thông qua các biểu thức toán học hoặc kỹ thuật thống kê. Nhưng hầu hết dữ liệu bao gồm dạng văn bản, quy tắc ngữ pháp khác nhau tùy vào ngôn ngữ nào đó. Trong một văn bản, các từ và nghĩa của từ có thể là nguồn thông tin rất hữu ích. Trong lab này, ta sẽ tìm hiểu một số kỹ thuật phân tích văn bản bằng thư viện NLTK cho phép bạn thực hiện các phép toán phức tạp. Hơn nữa giúp ta hiểu được một phần quan trọng trong phân tích dữ liệu.

Giới thiệu về thư viện NLTK

- Cài đặt thư viện NLTK:
- Vào của sổ command line, gõ lệnh
  - pip install NLTK
- Import thư viện NLTK và download công cụng NLTK
- Gõ lệnh:
  - import nltk
  - nltk.download_shell()

Giới thiệu về thư viện NLTK

- Để có thể tạo một số các ví dụ để tìm hiểu về thư viện, bạn cần một số các văn bản để làm việc. Một nguồn văn bản tuyệt vời phù hợp cho mục đích này là kho ngữ liệu Gutenberg, có trong bộ sưu tập kho ngữ liệu. Kho ngữ liệu Gutenberg gồm các văn bản được trích xuất từ kho lưu trữ điện tử có tên là dự án Gutenberg (http: // www.gutenberg.org/). Có hơn 25.000 sách điện tử trong kho lưu trữ này.
- Để tải xuống gói này, trước tiên hãy nhập tùy chọn d để tải xuống. Công cụ sẽ hỏi bạn cho tên gói, vì vậy bạn nhập tên gutenberg.

Giới thiệu về thư viện NLTK

- Sau khi hoàn tất, bạn có xem tên các tập tin có trong gói 'gutenberg' bằng lệnh fileids().
  - gb = nltk.corpus.gutenberg
  - print("Gutenberg files : ", gb.fileids())

Giới thiệu về thư viện NLTK

- Để truy cập nội dung bên trong của một trong các tập tin này, trước tiên bạn chọn một tập tin, chẳng hạn Macbeth của Shakespeare (shakespeare-macbeth.txt), sử dụng hàm words (), rồi gán cho một biến nào đó.
  - macbeth = nltk.corpus.gutenberg.words('shakespeare-macbeth.txt')

Giới thiệu về thư viện NLTK

- Nếu bạn muốn biết chiều dài của văn bản trên (bao nhiêu từ), bạn dùng hàm len()
  - len(macbeth)

Giới thiệu về thư viện NLTK

- Nếu bạn muốn hiển thị 10 từ đầu tiên của tập tin
  - macbeth [:10]
- Hàm trên trích ra 10 từ đầu tiên là tiêu đề của tập tin, dấu ngoặc vuông được tính là 1.
- Ta muốn trích 5 câu đầu tiên của tập tin (một câu được kẹp trong cặp ngoặc vuông), ta dùng hàm sent()
  - macbeth_sents = nltk.corpus.gutenberg.sents('shakespeare-macbeth.txt')
  - macbeth_sents[:5]

Tìm 1 từ với NLTK:

- Tìm từ ‘Stage’ xuất hiện trong văn bản text
  - text = nltk.Text(macbeth)
  - text.concordance('Stage')
- Tìm từ xuất hiện trước và sau từ ‘Stage’
  - text.common_contexts(['Stage'])
- Tìm từ tương tự từ ‘Stage’
  - text.similar('Stage')

Phân tích tần số của các từ:

- Muốn xem 10 từ thông dụng nhất trong văn bản xuất hiện bao nhiêu lần, dùng lệnh most_common()
  - fd = nltk.FreqDist(macbeth)
  - fd.most_common(10)
- Stopword là những từ thông dụng, thường ít có ý nghĩa trong quá trình phân tích văn bản và thường cần được loại bỏ.
- Muốn download stopword
  - nltk.download('stopwords')

Phân tích tần số của các từ:

- Muốn xem các stopword trong tiếng Anh, dùng lệnh
  - sw = set(nltk.corpus.stopwords.words('english'))
  - print(len(sw))
  - list(sw)[:10]
- Có 179 stopword trong từ vựng tiếng Anh. Ta sẽ loại bỏ các từ stopword trong biến macbeth
  - macbeth_filtered = [w for w in macbeth if w.lower() not in sw]
  - len(macbeth_filtered)

  - fd = nltk.FreqDist(macbeth_filtered)
  - fd.most_common(10)

Phân tích tần số của các từ:

- Bây giờ, 10 từ phổ biến nhất đầu tiên được trả về, các stopword đã được loại bỏ nhưng vẫn còn các dấu câu, ta cần loại bỏ các dấu câu theo lệnh sau:
  - import string
  - punctuation = set(string.punctuation)
  - macbeth_filtered2 = [w.lower() for w in macbeth if w.lower( ) not in sw and w.lower() not in punctuation]
  - fd = nltk.FreqDist(macbeth_filtered2)
  - fd.most_common(10)

Bigrams và collocations

- Lọc các bigram sau khi đã loại các stopword và các dấu câu, dùng lệnh sau:
  - bgrms = nltk.FreqDist(nltk.bigrams(macbeth_filtered2))
  - bgrms.most_common(15)
- Ngoài bigram ra, còn có trigram, sự kết hợp của 3 từ, ta dùng lệnh trigrams()
  - tgrms = nltk.FreqDist(nltk.trigrams (macbeth_filtered2))
  - tgrms.most_common(10)

Sử dụng văn bản trên mạng

- Import thư viện và mở url để đọc file:
  - from urllib import request
  - url = "http://www.gutenberg.org/files/2554/2554-0.txt"
  - response = request.urlopen(url)
  - raw = response.read().decode('utf8')
  - raw[:75]
- Kết quả:
  - '\ufeffThe Project Gutenberg EBook of Crime and Punishment, by Fyodor Dostoevsky\r’

Sử dụng văn bản trên mạng

- Thay bằng các lệnh sau:
  - from urllib import request
  - url = "http://www.gutenberg.org/files/2554/2554-0.txt"
  - response = request.urlopen(url)
  - raw = response.read().decode('utf8-sig')
  - raw[:75]
- Kết quả:
  - 'The Project Gutenberg EBook of Crime and Punishment, by Fyodor Dostoevsky\
  - r\n’

Sử dụng văn bản trên mạng

- Thực hiện các lệnh sau:
  - tokens = nltk.word_tokenize (raw)
  - webtext = nltk.Text (tokens)
  - webtext[:12]
- Kết quả:
  - ['The','Project','Gutenberg','EBook','of','Crime','and', 'Punishment',',','by','Fyodor','Dostoevsky']

Rút trích văn bản từ trang html

- Trong phần này, chúng ta sẽ tìm hiểu cách rút trích văn bản từ trang html.
  - url = "http://news.bbc.co.uk/2/hi/health/2284783.stm"
  - html = request.urlopen(url).read().decode('utf8')
  - html[:120]
- Ta dùng thư viện bs4 (BeautifulSoup) cung cấp cho bạn các trình phân tích cú pháp phù hợp có thể nhận dạng HTML và trích xuất văn bản.
  - from bs4 import BeautifulSoup
  - raw = BeautifulSoup(html, "lxml").get_text()
  - tokens = nltk.word_tokenize(raw)
  - text = nltk.Text(tokens)

Phân tích cảm xúc người dùng

- Phân tích cảm xúc là một lĩnh vực nghiên cứu gần đây để đánh giá quan điểm của người dùng về một chủ đề nào đó. Việc phân tích này dựa trên về các kỹ thuật khác nhau sử dụng phân tích văn bản, các lĩnh vực trên các mạng xã hội và các diễn đàn.
- Nhờ nhận xét và đánh giá của người dùng, các thuật toán phân tích cảm xúc có thể đánh giá mức độ đánh giá hoặc đánh giá dựa trên các từ khóa nhất định. Mức độ đánh giá được gọi là ý kiến và có ba giá trị có thể có: **positive, neutral hoặc negative**. Việc đánh giá ý kiến này do đó trở thành một dạng phân loại.

Phân tích cảm xúc người dùng

- Mục đích của ví dụ dưới đây là tìm các từ lặp lại nhiều nhất dạng positive và dạng negative hoặc các từ khóa liên quan đến một ý kiến nào đó, sử dụng thuật toán Naïve Bayes để phân loại cảm xúc người dùng dựa vào các đoạn văn bản movie_reviews của họ.

Phân tích cảm xúc người dùng

- Trước tiên, download các movie review dùng lệnh sau:
  - nltk.download('movie_reviews')
- Sau đó xây dựng tập train dựa vào copus trên. Tạo 1 mảng tên là documents. Mảng này chứa cột đầu tiên là nội dung review của người dùng, và cột thứ 2 là cột đánh giá **positive, neutral, or negative.** Sau đó trộn các dòng này ngẫn nhiên.
  - import random
  - reviews = nltk.corpus.movie_reviews
  - documents = [(list(reviews.words(fileid)), category)
  - for category in reviews.categories()
  - for fileid in reviews.fileids(category)]
  - random.shuffle(documents)

Phân tích cảm xúc người dùng

- Xem nội dung review đầu tiên (dòng 0, cột 0)
  - first_review = ' '.join(documents[0][0])
  - print(first_review)
- Xem kết quả review đầu tiên (dòng 0, cột 1)
  - documents[0][1]
- Ta cần tạo bảng phân phối tần số các từ trong copus, bảng này cần chuyển sang dạng list, ta dùng hàm list()
  - all_words = nltk.FreqDist(w.lower() for w in reviews.words())
  - word_features = list(all_words)

Phân tích cảm xúc người dùng

- Sau đó, bước tiếp theo là xác định một hàm để tính toán các đặc trưng, tức là những từ đủ quan trọng để thiết lập ý kiến của một review.

```
def document_features(document, word_features):
    document_words = set(document)
    features = {}
    for word in word_features:
        features['{}'.format(word)] = (word in document_words)
    return features
```

Phân tích cảm xúc người dùng

- Khi bạn định nghĩa hàm document_features(), bạn tạo 1 tập các documents
  - featuresets = [(document_features(d,word_features), c) for (d,c) in documents]
  - len(featuresets)

Phân tích cảm xúc người dùng

- Tạo tập train và tập test: 1500 dòng đầu dùng cho tập train và 500 dòng còn lại dùng cho tập test để đánh giá độ chính xác của mô hình.
  - train_set, test_set = featuresets[1500:], featuresets[:500]
  - classifier = nltk.NaiveBayesClassifier.train(train_set)
- Dùng thuật toán Naïve Bayes để phân loại, dùng thư viện NLTK. Sau đó tính toán độ chính xác của thuật toán
  - train_set, est_set = featuresets[1500:], featuresets[:500]
  - classifier = nltk.NaiveBayesClassifier.train(train_set)
  - print(nltk.classify.accuracy(classifier, test_set))

Phân tích cảm xúc người dùng

- Chúng ta đã hoàn tất việc phân tích, dưới đây các từ với trọng số lớn nhất của các review được đánh giá là positive và negative
  - classifier.show_most_informative_features(10)
- Kết quả:

```
            badly = True              neg : pos    =     11.1 : 1.0
            julie = True              neg : pos    =      9.5 : 1.0
           finest = True              pos : neg    =      9.0 : 1.0
           forgot = True              neg : pos    =      8.8 : 1.0
            naked = True              neg : pos    =      8.8 : 1.0
       refreshing = True              pos : neg    =      7.9 : 1.0
           stolen = True              pos : neg    =      7.3 : 1.0
          luckily = True              pos : neg    =      7.3 : 1.0
          directs = True              pos : neg    =      7.3 : 1.0
             rain = True              neg : pos    =      7.3 : 1.0
```