THU THẬP DỮ LIỆU VĂN BẢN (EXTRACTING TEXT DATA)

AI Academy Vietnam

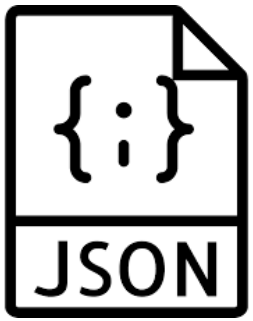
Nội dung

1. Giới thiệu

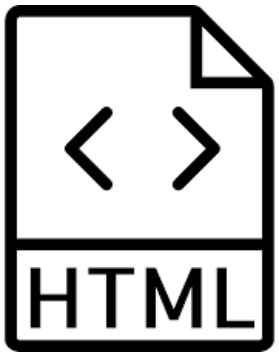
2. Các kỹ thuật trích xuất dữ liệu văn bản
   - Thu thập dữ liệu từ PDF
   - Thu thập dữ liệu từ WORD
   - Thu thập dữ liệu từ JSON
   - Thu thập dữ liệu từ HTML
   - Thu thập dữ liệu từ Web
   - Xử lý văn bản sử dụng biểu thức chính quy
   - Thao tác với chuỗi

Giới thiệu

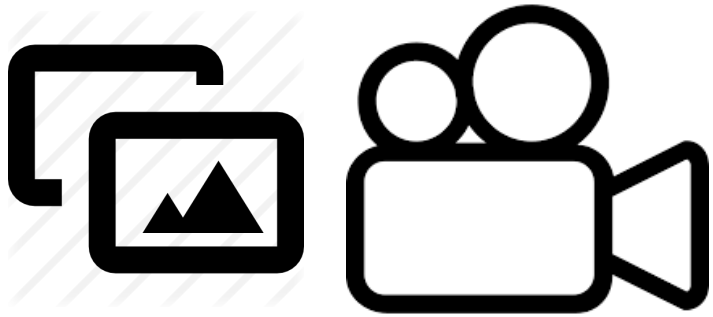

PDF


{;}
JSON

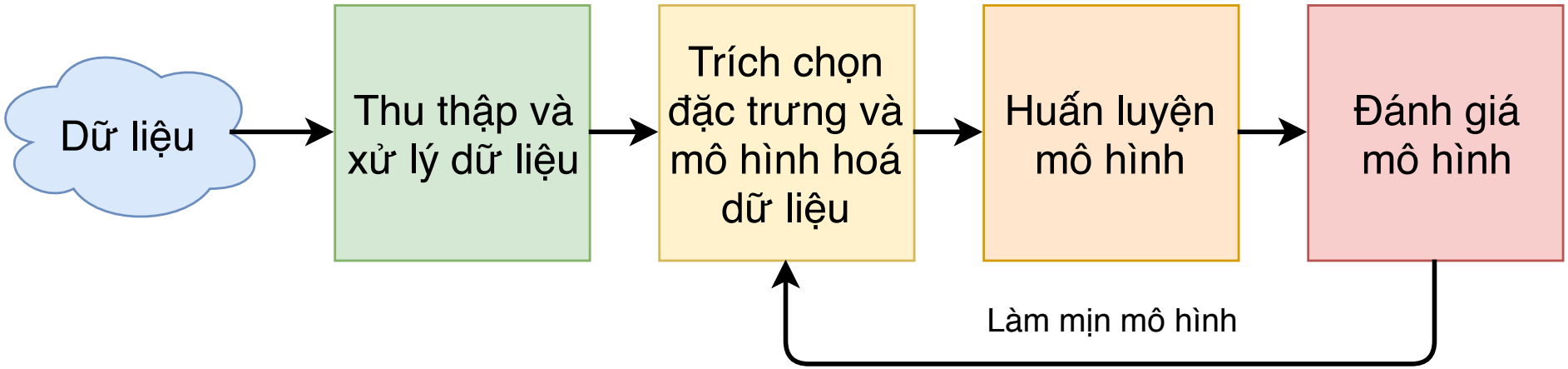

W


HTML


Favicon - Wikipedia, the free encyc
File Edit View History Bookmarks Tools Help
http://en.wikipedia.org/wiki/Favicon
Wikipedia, the free encyclop...
Wikipedia, the free...
Help us improve Wikipedia by
article
discussion
Favicon
From Wikipedia, the free encyclop
A favicon (short for favorites icon), also known as
webpage. A web designer can create such an icon
Browsers that provide favicon support typically display a
support a tabbed document interface typically show a page
to web pages.



Các bước của mô hình học máy



0.12	**0.34**	**0.44**	**0.05**
0.53	0.23	0.78	0.72
0.45	0.64	0.38	0.34

- Dữ liệu: huấn luyện, kiểm tra, đánh giá
- Huấn luyện
- Đánh giá: K-fold cross validation, 3-1-1

Nội dung

1. Giới thiệu

2. Các kỹ thuật trích xuất dữ liệu văn bản
   - Thu thập dữ liệu từ PDF
   - Thu thập dữ liệu từ WORD
   - Thu thập dữ liệu từ JSON
   - Thu thập dữ liệu từ HTML
   - Thu thập dữ liệu từ Web
   - Xử lý văn bản sử dụng biểu thức chính quy
   - Thao tác với chuỗi

Thu thập dữ liệu

- Là bước đầu tiên của mọi bài toán xử lý ngôn ngữ tự nhiên
- Dữ liệu
  - Dữ liệu thô (raw data)
  - Dữ liệu đã xử lý (feature vectors)
- Một số loại dữ liệu cơ bản
  - Văn bản
    - Tin tức (news)
    - Ý kiến người dùng (tweets, comments, reviews)
    - Văn bản y sinh
  - Âm thanh
  - Hình ảnh

Trích xuất dữ liệu từ PDF

- **Đầu vào**: Một file pdf chứa nội dung
- **Đầu ra**: Dữ liệu văn bản trong file pdf đó
- **Giải pháp**:
  - Sử dụng thư viện PyPDF2 [1]
    - !pip install PyPDF2
    - import PyPDF2
  - Một số đặc tính
    - Trích xuất thông tin
    - Chia các trang văn bản
    - Gộp văn bản
    - Cắt trang
    - Mã hoá và giải mã các tệp tin PDF
- **Ví dụ**

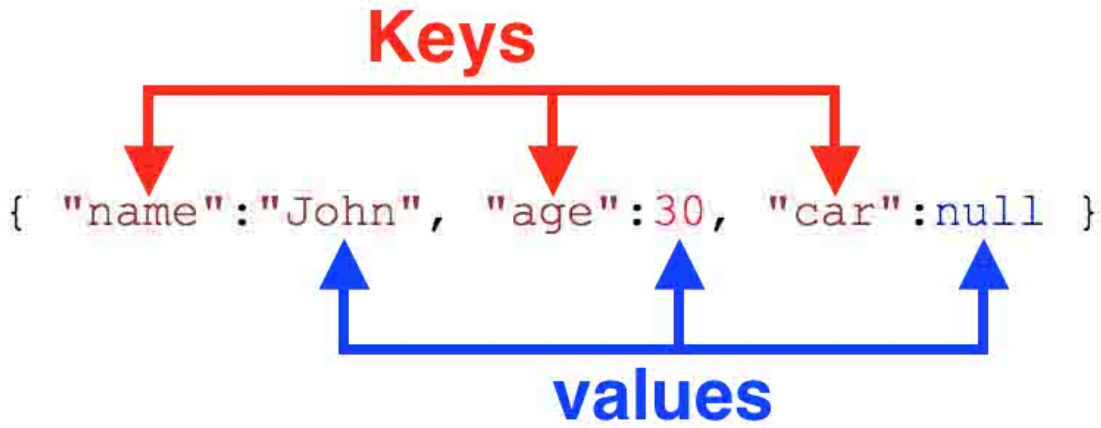
[1] https://pypi.org/project/PyPDF2/

Trích xuất dữ liệu từ WORD

- **Đầu vào**: Một file word chứa nội dung
- **Đầu ra**: Dữ liệu văn bản trong file word đó
- **Giải pháp**:
  - Sử dụng thư viện docx [2]
    - !pip install docx
    - from docx import document
  - Một số đặc tính
    - Cho phép làm việc với văn bản doc
    - Xem link đính kèm
- **Ví dụ**

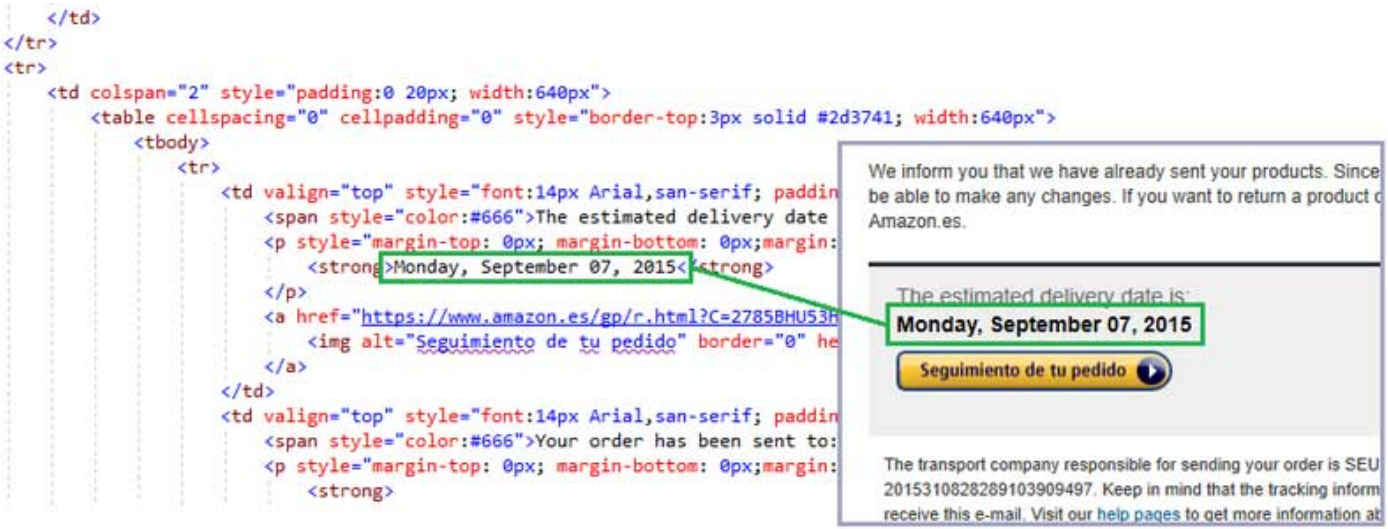
[2] https://pypi.org/project/python-docx/ hoặc https://python-docx.readthedocs.io/en/latest/

Trích xuất dữ liệu từ JSON



- **Đầu vào**: Một file JSON chứa nội dung
- **Đầu ra**: Dữ liệu văn bản trong file JSON đó
- **Giải pháp**:
  - Sử dụng thư viện có sẵn
    - import json
    - Tích hợp sẵn trong python các phiên bản
  - Lấy dữ liệu thông qua các key
- **Ví dụ**

Trích xuất dữ liệu từ HTML



- **Đầu vào**: Một file HTML chứa nội dung
- **Đầu ra**: Dữ liệu văn bản trong file HTML đó
- **Giải pháp**:
  - Sử dụng thư viện:
    - !pip install bs4
    - import urllib.request as urllib2
    - from bs4 import BeautifulSoup
  - Lấy dữ liệu thông qua các **tag**
- **Ví dụ**

Trích xuất dữ liệu từ một trang web

Trích xuất thông tin từ website[3] (web scraping, web harvesting, or web data extraction) là kỹ thuật để trích xuất một lượng lớn dữ liệu từ các websites.

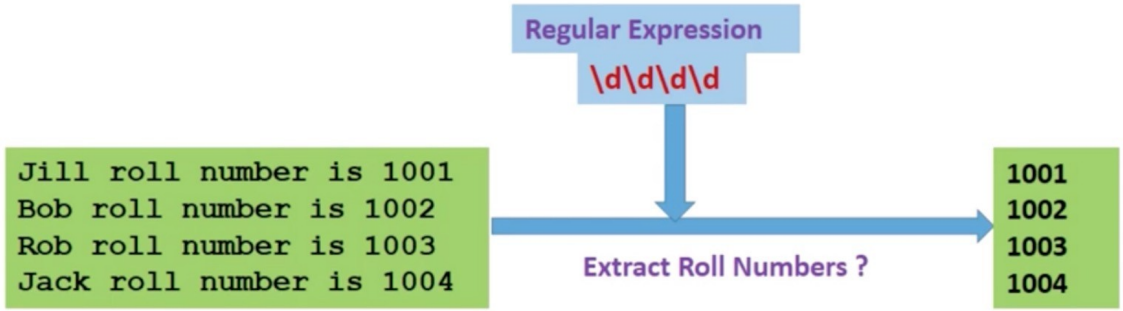
- **Đầu vào**: Website cần trích xuất thông tin
- **Đầu ra**: Dữ liệu văn bản trích xuất được từ website
- **Giải pháp**: sử dụng thư việnBeautifulSoup:
  - !pip install bs4
  - !pip install requests
  - from bs4 import BeautifulSoup
  - import requests
- **Ví dụ**

[3] **Lưu ý**: Cần phải chắc chắc website đó cho phép thu thập dữ liệu

Xử lý văn bản sử dụng biểu thức chính quy

What is a regular expression?

"A string that defines a text matching pattern"



- **Đầu vào**: Một văn bản chứa nội dung
- **Đầu ra**: Dữ liệu văn bản đã được xử lý bởi biểu thức chính quy (Regular Expression)
- **Giải pháp**:
  - Sử dụng biểu thức chính quy
  - import re

Một số biểu thức chính quy (1/2)

- **Regex: [ab]** tìm sự xuất hiện đơn của ký tự a và b
- **Regex: [^ab]** tìm các ký tự ngoại trừ a và b
- **Regex: [a-z]** tìm các ký tự trong khoảng a đến z
- **Regex: [^a-z]** tìm các ký tự ngoại trừ từ a tới z
- **Regex: [a-zA-Z]** tìm tất cả ký tự từ a đến z và A đến Z
- **Regex:.** bất kỳ ký tự đơn nào
- **Regex: \s** bất kỳ ký tự cách nào
- **Regex: \S** bất kỳ ký tự không cách nào
- **Regex: \d** bất kỳ số nào
- **Regex: \D** không phải là số
- **Regex: \w** bất kỳ các chữ cái tạo nên từ; tương ứng [A-Za-z0-9_]
- **Regex: \W** các ký tự ngoại trừ các ký tự tạo nên từ; tương ứng [^A-Za-z0-9_]
- **Regex: (a|b)** khớp một trong hai

Một số biểu thức chính quy (2/2)

- **Regex: a?** ; **?** xuất hiện 0 hoặc 1 lần, nhưng không nhiều hơn 1
- **Regex: a*** ; ***** xuất hiện 0 lần hoặc nhiều hơn 0
- **Regex: a+** ; **+** xuất hiện một hoặc nhiều lần
- **Regex: a{3}** xuất hiện đúng 3 lần
- **Regex: a{3, 6}** xuất hiện đồng thời giữa 3 và 6
- **Regex: a{3,}** xuất hiện đồng thời 3 hoặc nhiều hơn
- **Regex: a{, 6}** xuất hiện đồng thời đến 6 lần
- **Regex: ^** khớp bắt đầu một chuỗi ký tự
- **Regex: $** khớp với kết thúc một chuỗi ký tự

Một số hàm xử lý trong re

- **re.match()**: So khớp biểu thức ở đầu chuỗi. Nếu khớp thì trả về đối tượng *match*, còn không trả về *None*.
- **re.search()**: Tìm trong cả chuỗi và trả về vị trí đầu tiên tìm thấy. Nếu khớp thì trả về đối tượng *match*, còn không trả về *None*.
- **re.findall()**: Tìm và trả ra kết quả là một danh sách các chuỗi khớp với biểu thức chính quy.
- **re.sub()**: Thay thế chuỗi được tìm thấy bằng một chuỗi mới.
- **re.split()**: Tách chuỗi, trả ra một danh sách các chuỗi con.

Ví dụ 1: Làm sạch văn bản

In [19]:
```
sentence = '''Looking ahead, the US' climate plans are more ambitious than
            China's -- US President Joe Biden has pledged to at least halve US emissions
            by 2030, from 2005 levels -- but China is at a different stage of development,
            so that should be a factor in working out what the country's fair share of
            climate action should be. China is also ahead of the US on renewables.
            China expresses its commitments in terms of "carbon intensity" which allows for
            more emissions the more its GDP rises'''
re.sub('[^A-Za-z0-9.]+' , ' ', sentence)
```

Out[19]:
```
'Looking ahead the US climate plans are more ambitious than China s US President Joe Biden ha
s pledged to at least halve US emissions by 2030 from 2005 levels but China is at a different
stage of development so that should be a factor in working out what the country s fair share
of climate action should be. China is also ahead of the US on renewables. China expresses its
commitments in terms of carbon intensity which allows for more emissions the more its GDP ris
es'
```

Ví dụ 2: Tìm các con số trong văn bản

In [2]:
```
import re
re.findall(r'[0-9]', 'abc123 def456 789ghi')
```

Out[2]:
```
['1', '2', '3', '4', '5', '6', '7', '8', '9']
```

In [3]:
```
re.findall(r'[0-9]+', 'abc123 def456 789ghi')
```

Out[3]:
```
['123', '456', '789']
```

In [4]:
```
#Tìm những text bắt đầu bằng số
re.findall(r'\b[0-9]+.*\b', 'abc123 def456 789ghi')
```

Out[4]:
```
['789ghi']
```

Ví dụ 3: Giữ lại text loại bỏ các phần đi kèm với con số

In [5]:
```
#Cách 1
re.findall(r'\b[A-Za-z]+\b', '123abc def456 7g8h9i jkl')
```

Out[5]:
```
['jkl']
```

In [6]:
```
#Cách 2
re.findall(r'\b[A-Za-z]+\b', '123abc def456 7g8h9i jkl')
```

Out[6]:
```
['jkl']
```

Ví dụ 4: Tìm kiếm ngày trong năm

In [24]:
```
import re
date = '23 oct 2019 23 oct,2019 23 october,2019 oct 26,2020'
# tìm kiếm với dấu cách nằm giữa ngày tháng năm
x = re.findall('\d{2} [a-z]{3} \d{4}',date)
print(x)

# tìm kiếm với dấu cách hoặc dấu phẩy nằm giữa ngày tháng năm
y = re.findall('\d{2}[ |,][a-z]{3}[ |,]\d{4}',date)
print(y)
```

```
['23 oct 2019']
['23 oct 2019', '23 oct,2019']
```

Ví dụ 5: Tìm kiếm hashtag

In [27]:
```
#tìm kiếm hashtag
tweet = "wow!,it is a natural beauty.#nature #_beautiful #"
x = re.findall('#[_]*[a-z]+',tweet)
print(x)
```

```
['#nature', '#_beautiful']
```

Ví dụ 6: Tìm tên file ảnh

```
In [7]: re.findall(r'\w+.(png|jpeg)', 'file names: image_1.png and image_2.jpeg')
Out[7]: ['png', 'jpeg']

In [8]: re.findall(r'\w+.(?:png|jpeg)', 'file names: image_1.png and image_2.jpeg')
Out[8]: ['image_1.png', 'image_2.jpeg']
```

Ví dụ 7: Tìm địa chỉ HTTP

In [14]:
```
re.findall(r'http://\S*/',
           '''addresses: http://www.google.com/,
           http://www.bad address.com/,
           and https://www.kaggle.com/ ''')
```

Out[14]:
```
['http://www.google.com/']
```

Thao tác với chuỗi

Các đơn giản nhất là ta sử dụng các hàm xử lý chuỗi sau:

- **s.find(t)**: vị trí đầu tiên của chuỗi t trong s (-1 nếu không tìm thấy)
- **s.rfind(t)**: vị trí cuối cùng của chuỗi t trong s (-1 nếu không tìm thấy)
- **s.index(t)**: giống như s.find(t), nhưng trả ra lỗi ValueError nếu không tìm thấy t
- **s.rindex(t)**: giống như s.rfind(t), nhưng trả ra ValueError nếu không tìm thấy t
- **s.join(text)**: nối các từ thành một chuỗi
- **s.split(t)**: chia chuỗi s thành một danh sách (list) các từ, với t là chuỗi phân tách.
- **s.lower()**: đổi s thành chữ thường
- **s.upper()**: đổi s thành chữ hoa
- **s.strip()**: copy s bằng cách loại bỏ các dấu cách ở đầu và cuối
- **s.replace(t, u)**: thay thế t thành u trong s

THANK YOU

Q&A