

Bài 2

Thu thập dữ liệu văn bản

Extracting the Text Data

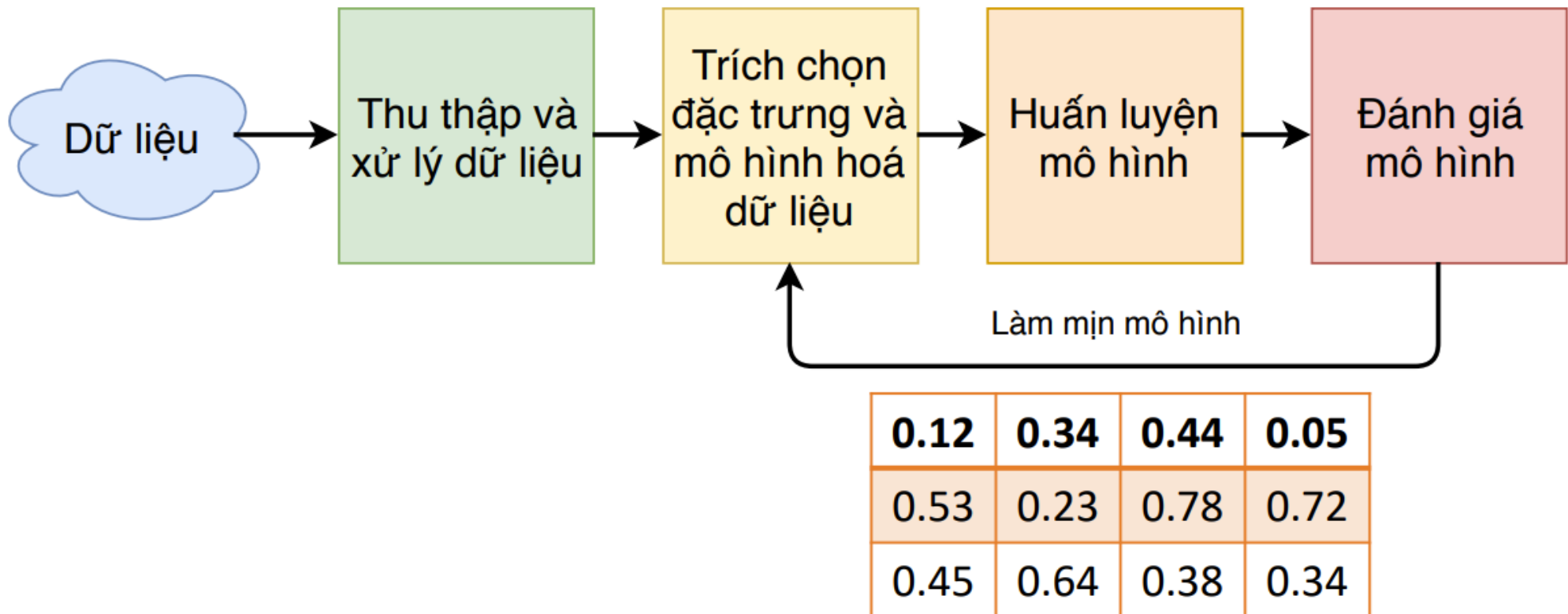
Lê Thanh Hương
Trường Công nghệ Thông tin và Truyền thông, ĐHBKHN

Nội dung

- Giới thiệu
- Các kỹ thuật thu thập dữ liệu văn bản
 - Trích rút dữ liệu từ PDF
 - Trích rút dữ liệu từ WORD
 - Trích rút dữ liệu từ JSON
 - Trích rút dữ liệu từ HTML
 - Trích rút dữ liệu từ WEB
 - Phân tích văn bản sử dụng biểu thức chính qui
 - Xử lý chuỗi

Giới thiệu

- Các bước của mô hình học máy



- Dữ liệu: Huấn luyện (train), kiểm tra (validation), đánh giá (test)
- Sử dụng: k-fold cross validation

Thu thập dữ liệu

1. Là bước đầu tiên của mọi bài toán NLP
2. Dữ liệu
 - Dữ liệu thô (raw data)
 - Dữ liệu đã xử lý (feature vectors)
3. 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

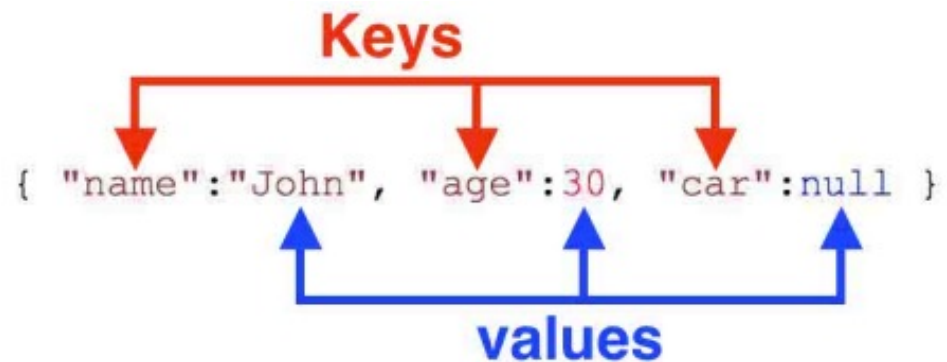
Trích rút dữ liệu từ PDF

- 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
(pypi.org/project/PyPDF2)
 - !pip install PyPDF2
 - Import PyPDF2
 - Một số công việc:
 - Chia trang văn bản
 - Gộp văn bản
 - Cắt trang
 - Mã hóa và giải mã các tệp PDF

Trích rút 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 (pypi.org/project/python-docx hoặc python-docx.readthedocs.io/en/latest)
 - !pip install docx
 - from docx import document
 - Một số công việc:
 - Xem link đính kèm

Trích rút 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

Trích rút 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

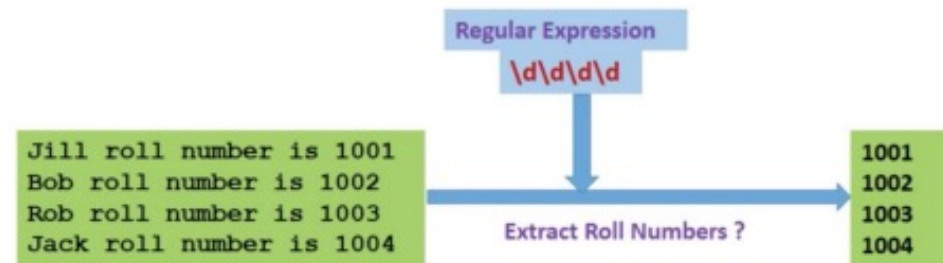
Trích rút dữ liệu từ Web

- Trích xuất thông tin từ website (web scraping, web harvesting, web data extraction) là kỹ thuật trích xuất một lượng lớn dữ liệu từ các trang web
- Đầu vào: website cần trích rút thông tin
- Đầu ra: Dữ liệu văn bản trích rút được từ website
- Giải pháp:
 - Sử dụng thư viện BeautifulSoup
 - `!pip install bs4`
 - `!pip install requests`
 - `from bs4 import BeautifulSoup`
 - `Import requests`

Phân tích văn bản sử dụng biểu thức chính qui

What is a regular expression?

"A string that defines a text matching pattern"



- Đầu vào: văn bản
- Đầu ra: Dữ liệu văn bản được xử lý bởi biểu thức chính qui
- Giải pháp:
 - Sử dụng biểu thức chính qui
 - `import re`

Một số biểu thức chính qui

- **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` là từ
- **Regex:** `\W` không phải là từ
- **Regex:** `(a|b)` khớp một trong hai

Một số biểu thức chính qui (tiếp)

- **Regex:** `a?` ; `?` xuất hiện 0 hoặc 1 lần, nhưng không nhiều hơn 1
- **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:** `a3` xuất hiện đúng 3 lần
- **Regex:** `a3, 6` xuất hiện đồng thời giữa 3 và 6
- **Regex:** `^` bắt đầu một chuỗi ký tự
- **Regex:** `$` kết thúc một chuỗi ký tự

Các hàm:

- **`re.match()`:** so khớp chuỗi
- **`re.search()`:** tìm chuỗi
- **`re.findall()`:** tìm chuỗi

Xử lý chuỗi

Các hàm xử lý chuỗi:

- **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

Q&A

Thank you!