

Python (part 3)

String manipulation and Regular expression



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

The need to process string data in data science

- In data science, we often encounter strings and need to do string processing operations, especially in preprocessing step
- Example: normalize strings, find and extract substrings containing important info
- With simple string operations: use methods of string in Python
- With complex string operations: ...

Example

- Convert text into a standard format

	County	State	Voted
0	De Witt County	IL	97.8
1	Lac qui Parle County	MN	98.8
2	Lewis and Clark County	MT	95.2
3	St John the Baptist Parish	LA	52.6

	County	State	Population
0	DeWitt	IL	16,798
1	Lac Qui Parle	MN	8,067
2	Lewis & Clark	MT	55,716
3	St. John the Baptist	LA	43,044

Example

- Extract a piece of text to create a feature

169.237.46.168 - -

[26/Jan/2004:10:47:58 -0800]"GET /stat141/Winter04 HTTP/1.1" 301 328

"http://anson.ucdavis.edu/courses"

"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NET CLR 1.1.4322)"

Example

□ Transform text into features

unclean **or** degraded floors walls **or** ceilings

inadequate **and** inaccessible handwashing facilities

inadequately cleaned **or** sanitized food contact surfaces

wiping cloths **not** clean **or** properly stored **or** inadequate sanitizer

foods **not** protected **from** **contamination**

unclean nonfood contact surfaces

unclean **or** unsanitary food contact surfaces

unclean hands **or** improper use of gloves

inadequate washing facilities **or** equipment

These new features can be used **in** an analysis of food safety scores.

Example

☐ Text analysis

State of the Union Address

George Washington

January 8, 1790

Fellow-Citizens of the Senate and House of Representatives:

I embrace with great satisfaction the opportunity which now presents itself of congratulating you on the present favorable prospects of our public ...

Basic String Manipulation

Basic String Manipulation

- ☐ Transform upper case characters to lower case (or vice versa).
- ☐ Replace a substring with another or delete a substring.
- ☐ Split a string into pieces at a particular character.
- ☐ Slice a string at specified locations.

Basic String Manipulation

	County	State	Voted		County	State	Population
0	De Witt County	IL	97.8	0	DeWitt	IL	16,798
1	Lac qui Parle County	MN	98.8	1	Lac Qui Parle	MN	8,067
2	Lewis and Clark County	MT	95.2	2	Lewis & Clark	MT	55,716
3	St John the Baptist Parish	LA	52.6	3	St. John the Baptist	LA	43,044

- ☐ Capitalization: qui vs Qui
- ☐ Omission of words: County and Parish are absent from right table
- ☐ Different abbreviation conventions: & vs and
- ☐ Different punctuation conventions: St. vs St
- ☐ Use of whitespace: DeWitt vs De Witt

Basic String Manipulation

```
def clean_county(county):  
    return (  
        county.lower()  
        .replace("county", "")  
        .replace("parish", "")  
        .replace("&", "and")  
        .replace(".", "")  
        .replace(" ", "")  
    )
```

Basic String Manipulation

☐ Complete list methods

Method	Description
str.lower()	Returns a copy of a string with all letters converted to lowercase
str.replace(a, b)	Replaces all instances of the substring a in str with the substring b
str.strip()	Removes leading and trailing whitespace from str
str.split(a)	Returns substrings of str split at a substring a
str[x:y]	Slices str, returning indices x (inclusive) to y (not inclusive)

Splitting Strings to Extract Pieces of Text

- How to get Date from this string information

```
169.237.46.168 - - [26/Jan/2004:10:47:58 -0800]"GET /stat141/Winter04
HTTP/1.1"
301 328 "http://anson.ucdavis.edu/courses""Mozilla/4.0 (compatible; MSIE
6.0;
Windows NT 5.0; .NET CLR 1.1.4322)"
```

Splitting Strings to Extract Pieces of Text

- How to get Date from this string information

```
log_entry.split('[')
```

```
log_entry.split('[')[1].split(':')[0]
```

```
(log_entry.split('[')[1]  
.split(':')[0]  
.split('/'))
```

Regular expression

Regular expression

- ☐ Example about a quite complex string operation
- ☐ Find and extract phone number (according to US phone number format: **3 numbers, then -, then 3 numbers, then -, then 4 numbers**) from a string (e.g., “My phone is 123-456-7890”)
- ☐ Use string methods in Python: how many lines of code?
- ☐ Use Regex: one line of code :-)

Regular expression - Regex

- Allow to do **complex** string operations via **string patterns**
- A string pattern is written according to **Regex syntax**
- E.g., pattern of US phone number (3 numbers, then -, then 3 numbers, then -, then 4 numbers): **\d{3}-\d{3}-\d{4}**
- Regex is supported in most programming languages, most editors, and some Linux commands (e.g. find, grep)

The plan

- ☐ First, learn about Regex syntax for writing patterns
- ☐ Then, learn about string functions in Python which use patterns written in Regex syntax

Regex — syntax

Literal

- ☐ A character is called a literal if this character means itself
- ☐ Demo ...

Character set

Syntax	Meaning (note: “the set of ...” means “a char in this set ...”)
[]	A set of chars; e.g. [1a.] is the set of chars 1, a, .
[start-end]	The set of all chars from start to end; e.g. [0-5] is the set of all chars from 0 to 5, [a-z] is the set of all chars from a to z
[^]	A negated set; e.g. [^12] is the set of all chars except 1, 2
.	The set of all chars except newline
\w & \W	The set of word chars (a-z, A-Z, 0-9, _) & the negated set
\d & \D	The set of digit chars & the negated set
\s & \S	The set of whitespace chars (space, tab, newline) & the negated set

Quantifier

Syntax	Meaning
{m}	The char right before repeats m times
{m,n}	The char right before repeats m-n times
{m,}	The char right before repeats $\geq m$ times
{,n}	The char right before repeats $\leq n$ times
*	Is the shorthand of {0,}
+	Is the shorthand of {1,}
?	Is the shorthand of {0,1}

Quantifier

- ☐ The **greedy** property of quantifier: quantifier will get the longest result
- ☐ Demo

Quantifier

- ☐ Quantifier only has effect on one char right before it
- ☐ To make quantifier having effect on more than one char right before it, we can use () to group chars
- ☐ Demo ...

Anchor

Syntax	Meaning
^	The location of the pattern after is the start of string
\$	The location of the pattern before is the end of string
\b	The location of the pattern after is the start of a word
\B	The location of the pattern after is not the start of a word

☐ Demo

Or

Syntax: |

Meaning: or pattern before |, or pattern after |

- ☐ Normally, pattern before | is all before | in Regex expression, and pattern after | is all after |
- ☐ We can use group if we just want a part before | and a part after |:

(the part before | the part after)

Demo ...

Regex — in Python

Use Regex in Python

import re # Built-in lib

- ☐ Here we just talk about some common used functions of this lib
- ☐ When needed, you can search [document](#)

Regex function

- `re.search(...)`
- `re.finditer(...)`
- `re.findall(...)`
- `re.sub(...)`
- `re.split(...)`
- Use flag

Reference

- <https://docs.python.org/3/library/re.html>
- The “Principles and Techniques of Data Science” book, chapter 13 - Working with Text.
URL:
https://www.textbook.ds100.org/ch/13/text_strings.html