Strings, Regular Expressions & Web Scraping

MMAI 5400 – lecture 1
Fall 2024

Text book (optional)

Speech & Language Processing

By Dan Jurafsky & James H. Martin

Freely available at Speech & Language Processing.

Office hours

Wednesdays 1:00 to 2:30 pm.

Room G238

Please send me a message *in advance* to arrange a meeting.

Assignments & exams

Assignments

- **Number**: 3

Exams

- **Number**: 2, midterm & final exams
- **When**:
  - Midterm - October 30
  - Final - December 11
- **Format**: open-book, but no electronics
- **Materials**: printable lecture notes & slides

Content

The data type

- *string*

A tool to operate on strings

- *Regular expressions*

A common way to extract data

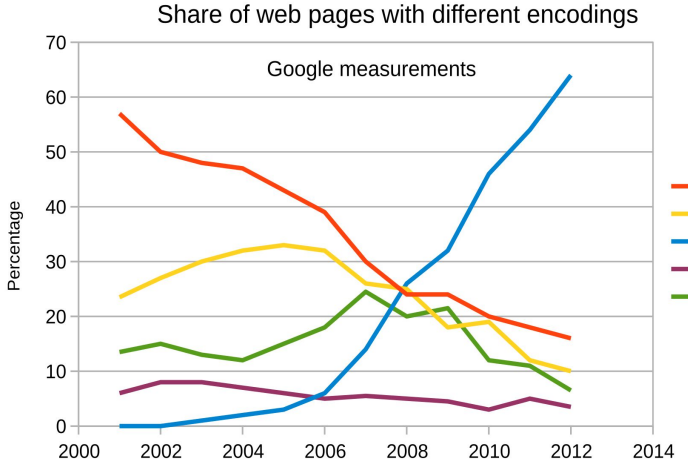
- *Web scraping*

Prepare the data for machine learning

- *Tokenization*

Strings

Image credits: Chris55 - CC BY-SA 4.0,
https://commons.wikimedia.org/w/index.php?curid=51421096



Strings

A string is a sequence of characters with some character encoding (e.g. ASCII, UTF-8, ...)

Character encodings

- Traditionally: 7 bits per character, for a total of $2^7 = 128$ unique characters.
  - Example: ASCII
  - Problem: Many languages needed more characters, e.g. Chinese, Japanese & Korean (even Swedish)
- Unicode:
  - Unicode specifies a relationship between characters & numbers representing those characters (code points)
  - UTF-8: a particular way of encoding Unicode; a variable width encoding made up of 1 to 4 bytes. Can encode all 1,112,064 unicode characters
  - For a more complete explanation see this StackOverflow post: https://stackoverflow.com/questions/643694/what-is-the-difference-between-utf-8-and-unicode

Question: Why not?
Question: How to print a backslash?

Strings in Python

The string data type in Python is called `str` & holds Unicode strings

Strings may be delimited with either single, double or triple (later slides) quotes

- This allows for nesting.
- Ok: `"My name is not 'Hjalmar'"`
- Not ok: `"My name is not "Hjalmar""`

Escape sequences in strings

The backslash character (\), aka the escape character, indicates the beginning of an escape sequence

Try the following:

- `print("My name is not "Hjalmar"")`
- `print("My name is not \"Hjalmar\"")`
- `print("New lines\nare important")`
- `print("... and so are \ttabs")`

The escape character allows the following character(s) to “escape” their normal meaning.

Strings in Python

- raw strings

Indicated by a preceding `r` or `R`

E.g. `r"this is a raw string"` or `R"this is a raw string"`

Escape sequences in a raw string are **NOT** translated

Try:

```
- print("My name is not \"Hjalmar\"")
- print(r"My name is not \"Hjalmar\"")
- print("New lines\nare important")
- print(R"New lines\nare important")
```

Or better yet:

```
- print("New lines are \no\t important")
- print(r"New lines are \no\t important")
```

Strings in Python

- triple quoted

Strings can also be delimited by triple single or double quotation marks.

E.g. `"""This is a string delimited by triple double quotes"""`

In triple quoted strings both single & double quotes can be included without escaping them.

E.g. `'''This is "a crazy mess" of 'ndom quotes'''`

Also, they allow for multi-line strings.

E.g.
```
"""This string stretch across

    2  lines"""
```

Regular Expressions

Operate on strings: regular expressions

Regular expressions (aka **regex**) are extremely useful for extracting information from text/strings

- A miniature language used to specify search **patterns for strings**
  - E.g.: Say that you want to automatically extract the price of bikes from the following text: "*10 blue bikes cost $500, while 23 pink bikes cost $623.*"
- Supported by many languages & platforms, e.g.: SQL, Python, R, Alteryx, Tableau, LibreOffice, Java, Scala, .NET, & Go
- Text editors & IDEs often support searching with regex, e.g.: VSCode, Notepad++, Emacs, Vim, Intellij IDEA, & PyCharm
- Very important in data science

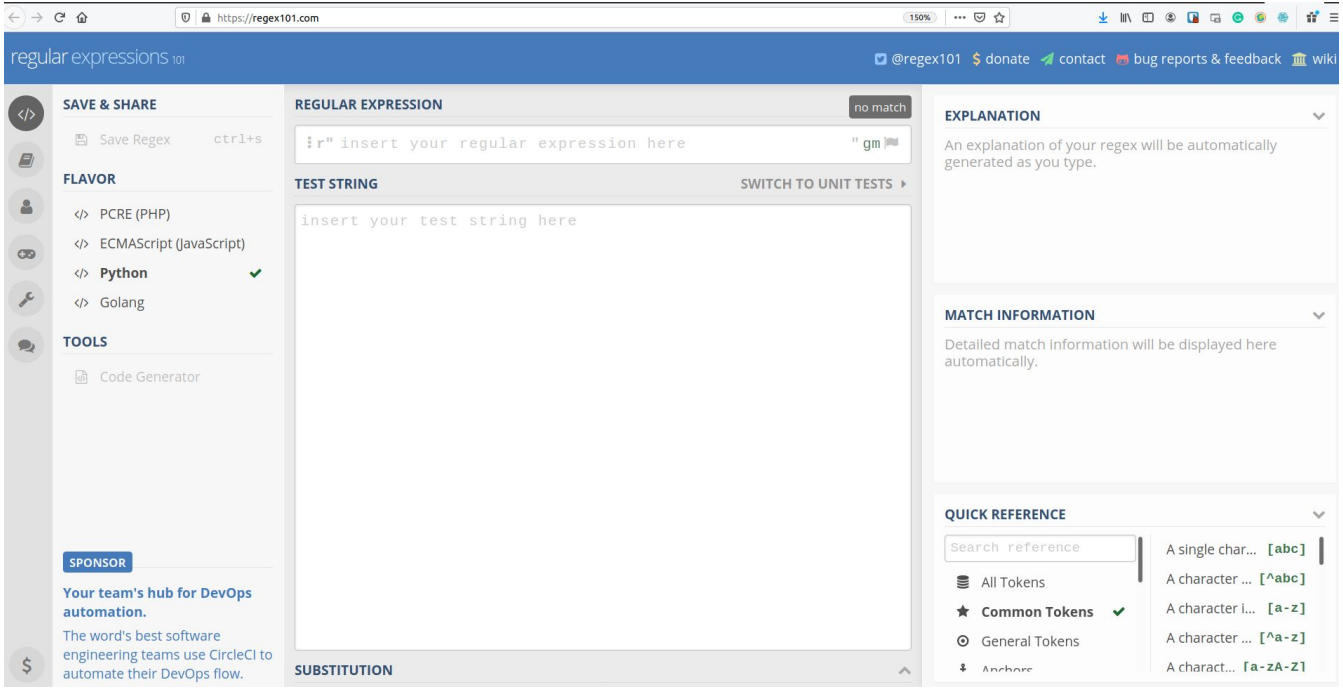
Formally, a regular expression is an algebraic notation for characterizing a set of strings

Example uses of regular expressions

- Web scraping
- Extracting information, e.g., numbers from text (code, log files, spreadsheets, ...)
- Data wrangling
- Simple parsing
- Tokenization
- Input validation (email addresses, user names, passwords)
- Use the command `grep` to search for a particular pattern in all text files in a folder
- ...

Learning & testing regular expressions

Regular expressions 101 is a web site for testing regular expressions: https://regex101.com/



Question: Which of the 3 strings match the regex? (use regex 101 to check)

Regular expressions

- a trivial example

Regex: “My name”

Strings: [“I’m called Moz”, “My name isn’t Zumbi”, “my name is Hjalmar”]

Regular expressions

- a trivial example, continued

Regex: “`My name`”

Strings: [“`I’m called Moz`”, “`My name isn’t Zumbi`”, “`my name is Hjalmar`”]

How would we match both “`My name`” & “`my name`”?

Regex with Python exercise

Open `MMAI5400_class01_RE.ipynb`

Strings & regex

- terminology

String

Character encoding

Unicode

Escape sequence

regex

Web scraping

What is web scraping?

Aka *Screen Scraping*, *Web Data Extraction* & *Web Harvesting*

A technique to extract data from web pages

It can be done manually, but automation makes is faster, more efficient & less error-prone

Some understanding of how web pages are displayed is useful for web scraping

Examples of web scraping

Online stores often scrape web pages of their competitors for prices to adjust their own prices.

“Contract scraping” for personal information to use for marketing purposes.

Researchers create data sets for text mining.

- E.g. collections of journal articles, wikipedia articles or digitized texts.

eXtensible Markup Language

HyperText Markup Language

XML & HTML documents

XML & HTML documents are text files written in ***m**arkup **l**anguages* (X**ML** & HT**ML**)

- Markup language: a system for annotating a document in a way that is syntactically distinguishable from the text content. The markup language is used to format the document & affects how the text content is displayed, but is not itself shown.
- Examples are LaTeX, Scribe, XML, HTML, SGML, Markdown, & many more

XML & HTML are very similar

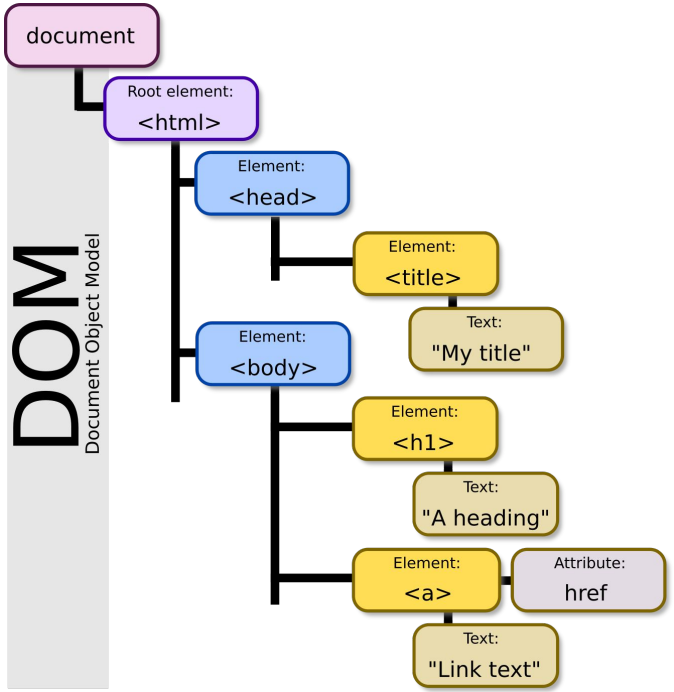
- Starting with HTML5, HTML documents are essentially a dialect of XML

Basic XML document structure

- Hierarchical structure (i.e. like a tree) with nodes: *element nodes*, *attribute nodes* & *text nodes*
- Every element is an *element node*
- **Element nodes** must have both opening & closing tags, e.g. `<boring_teacher>` & `</boring_teacher>`
- Tags are case sensitive, e.g. `<boring_teacher>` does **not** equal `<Boring_Teacher>`
- Elements must be properly nested:

```
<boring_teacher>
     <name>Hjalmar Turesson</name>
          <address>Rua Genesio Pitanga 160, Pium, Parnamirim, RN, Brazil</address>
     <date>April 02, 2016</date>
</boring_teacher>
```

Basic XML document structure



- **Attribute nodes** contain values (must be quoted)
- Example: `<boring_teacher type="extremely"></boring_teacher>`
- **Text nodes** what is contained within the opening & closing tags, e.g. `<name>Hjalmar Turesson</name>`

Image credit: Birger Eriksson - CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=18034500

Question: How is the source displayed?

Try at w3schools.com:
https://www.w3schools.com/tags/tag_code.asp

HTML document

- example

Source:

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

Displayed as:

??

How to view the source code of a web page

Right-click on a webpage & select “View page source”

Safari seems to require that you turn on “Develop” under preferences > advanced (I haven’t tested this)

HTML tags (a small subset)

Headline tags

- `<h1> ... </h1>`
- `<h2> ... </h2>`
- `<h3> ... </h3>`
- ...

Paragraph tag

- `<p> ... </p>`

Bold font tag

- `<b> ... </b>`

Anchor (link) tag

- `<a> ... </a>`

Table tags

- `<table> ... </table>`
- Table row
  - `<tr> ... </tr>`
- Table cell
  - `<td> ... </td>`

For a more complete list see:
https://www.geeksforgeeks.org/most-commonly-used-tags-in-html/

Web scraping with Python

`Scrapy`

- End-to-end: downloads, cleans & saves
- For a robust & reusable web scraper
- Great tutorial: https://librarycarpentry.org/lc-webscraping/

`BeautifulSoup`

- Extracts data from an already downloaded web page (e.g. by using `requests` or `urllib3`)
- For static content
- Easiest to begin with
- Use for exercise & assignment

`Selenium`

- Dynamic content, e.g. scroll to load more content, click next to see shopping offers, click button to view complete details, …

Web scraping exercise

Extract names & link to personal page for all ministers in the Legislative Assembly of Ontario.

URL: https://www.ola.org/en/members/current/ministers

Use `requests`, `BeautifulSoup` & `csv`

Open `MMAI5400_class01_WebScraping.ipynb`

Ministers

MPP	Policy Area
Bethlenfalvy, Hon. Peter	President of the Treasury Board
Calandra, Hon. Paul	Minister Without Portfolio
Cho, Hon. Raymond Sung Joon	Minister for Seniors and Accessibility
Clark, Hon. Steve	Minister of Municipal Affairs and Housing
Downey, Hon. Doug	Attorney General
Dunlop, Hon. Jill	Minister Without Portfolio
Dunlop, Hon. Jill	Associate Minister of Children and Women's Issues
Elliott, Hon. Christine	Minister of Health

Members
Current
Contact information
Ministers
Opposition critics
Parliamentary assistants
Composite list
In memoriam
All
Expense disclosure

Legality (NOT LEGAL ADVICE)

The legality of web scraping varies across the world

In general, the act of automatically downloading a web page & extracting information from it is not illegal (very similar to viewing a page through a browser, which is legal).

However, issues can arise is the scraped data is copyrighted & used for commercial purposes

Example:

- Mongohouse.com scraped the Toronto Real Estate Board (TREB) for listings, re-published them on their own site, & made money from real estate-related advertisement as competitors to TREB.
- This was deemed illegal: https://www.canadianlawyermag.com/news/opinion/federal-court-makes-clear-website-scraping-is-illegal/276128

Question: What does DoS mean?

Legality (NOT LEGAL ADVICE)

Advice (but still not legal advice)

- Read terms & conditions on the the web pages you intend to scrape
- Ask if doubtful
- Be careful with frequent requests to small web sites
  - You don't want to DoS them
  - Scrape during off-peak hours
  - Pause in-between requests (e.g. with Python's `time.sleep(num_seconds)`)

For more: https://robertorocha.info/on-the-ethics-of-web-scraping/

An alternative to scraping

Web APIs (Application Program Interface)

Many sites provide an API to extract structured data

Examples:

- Facebook API, Twitter API & YouTube comments API, cryptocurrency exchanges, PubMed

To start:

- Find the documentation to the API you are interested in (google for “site name” + “API”)
- Use `requests.get()` with parameters or find a Python package specific to the API

Web scraping

- terminology

- Web Scraping
  - Screen Scraping
  - Web Data Extraction
  - Web Harvesting
- Markup language
- XML/HTML
  - HTML Tree
  - Element/attribute/text nodes
  - HTML tags
- Web API

Python Debugging?