Regular Expression in NLP

Regular Expression

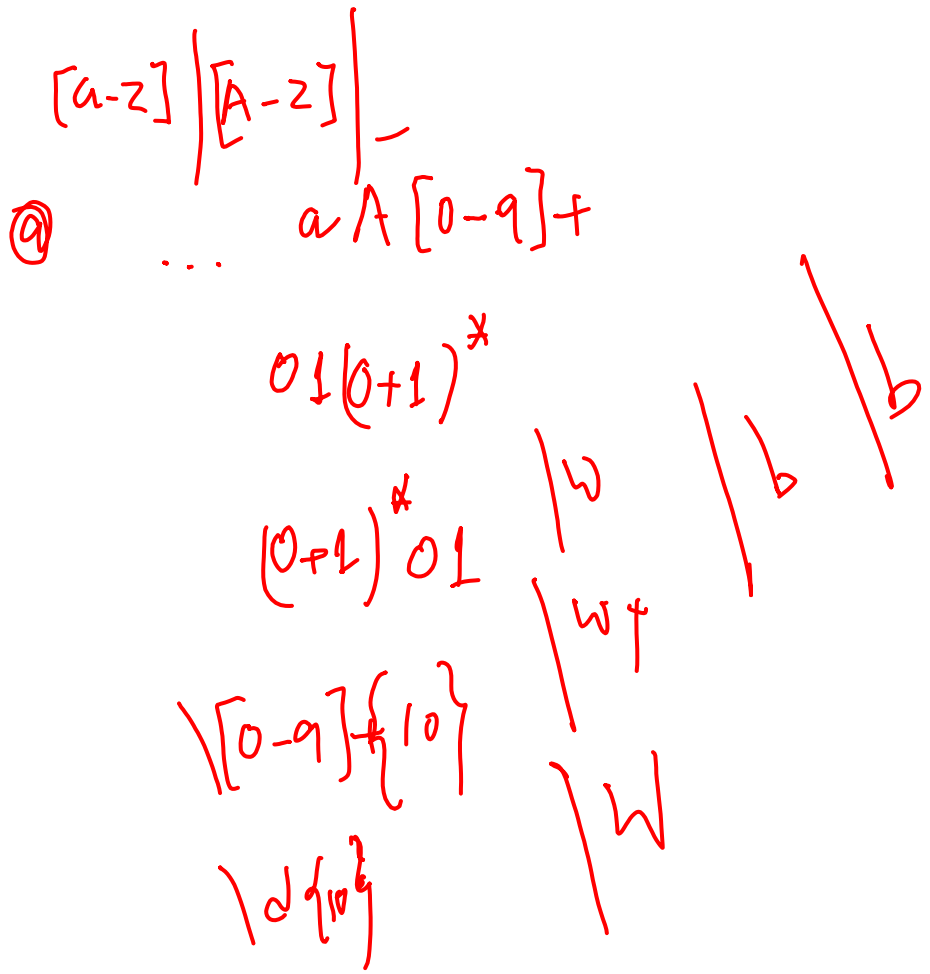
- A Regular Expression (or Regex) is a pattern (or filter) that describes a set of strings that matches the pattern.
- In other words, a regex accepts a certain set of strings and rejects the rest.
- A regex consists of a sequence of characters, metacharacters (such as .; \d, \D, \s, \S, \w, \W) and operators (such as +, *, ?, |, ^).

Matching a Single Character

- Most characters, including all letters (a-z and A-Z) and digits (0-9), match itself. For example, the regex x matches substring "x"; z matches "z"; and 9 matches "9".
- Regex is used to match single character.
- In Python:
  - Import re
  - re.findall(r'a', 'abcabc')
- Non-alphanumeric characters without special meaning in regex also matches itself. For example, = matches "="; @ matches "@".

Regex Special Characters and Escape Sequences

- Regex's Special Characters
- metacharacter: dot (.)
- bracket list: [ ]
- position anchors: ^, $
- occurrence indicators: +, *, ?, { }
- parentheses: ( )
- or: |
- escape and metacharacter: backslash (\)



Regular expression functions

re.findall("\d{10}", text)

Function	Meaning
findall	Returns a list containing all matches
search	Returns a Match object if there is a match anywhere in the string
split	Returns a list where the string has been split at each match
sub	Replaces one or many matches with a string

txt = "hi is my phone number
7985061200
9543210125"

Metacharacters

Character	Meaning	Example
[]	A set of characters	"[a-m]"
\	Signals a special sequence (can also be used to escape special characters)	"\d"
.	Any character (except newline character)	"he..o"
^	Starts with	"^hello"
$	Ends with	"planet$"
*	Zero or more occurrences	"he.*o"
+	One or more occurrences	"he.+o"
?	Zero or one occurrences	"he.?o"
{}	Exactly the specified number of occurrences	"he.{2}o"
|	Either or	"falls|stays"

Special Sequences

A special sequence is a \ followed by one of the characters in the list below, and has a special meaning:

Character	Meaning	Example
\A	Returns a match if the specified characters are at the beginning of the string	"\AThe"
\b	Returns a match where the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\bain" r"ain\b"
\B	Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\Bain" r"ain\B"
\d	Returns a match where the string contains digits (numbers from 0-9)	"\d"
\D	Returns a match where the string DOES NOT contain digits	"\D"
\s	Returns a match where the string contains a white space character	"\s"
\S	Returns a match where the string DOES NOT contain a white space character	"\S"

Character	Meaning	Example
\w	Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore _ character)	"\w"
\W	Returns a match where the string DOES NOT contain any word characters	"\W"
\Z	Returns a match if the specified characters are at the end of the string	"Spain\Z
\A	Returns a match if the specified characters are at the beginning of the string	"\AThe"
\b	Returns a match where the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\bain" r"ain\b"
\w	Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore _ character)	"\w"
\W	Returns a match where the string DOES NOT contain any word characters	"\W"
\Z	Returns a match if the specified characters are at the end of the string	"Spain\Z

Sets

A set is a set of characters inside a pair of square brackets [] with a special meaning:

Set	Meaning
[arn]	Returns a match where one of the specified characters (a, r, or n) is present
[a-n]	Returns a match for any lower case character, alphabetically between a and n
[^arn]	Returns a match for any character EXCEPT a, r, and n
[0123]	Returns a match where any of the specified digits (0, 1, 2, or 3) are present
[0-9]	Returns a match for any digit between 0 and 9
[0-5][0-9]	Returns a match for any two-digit numbers from 00 and 59
[a-zA-Z]	Returns a match for any character alphabetically between a and z, lower case OR upper case
[+]	In sets, +, *, ., |, (), $,{} has no special meaning, so [+] means: return a match for any + character in the string
[arn]	Returns a match where one of the specified characters (a, r, or n) is present
[a-n]	Returns a match for any lower case character, alphabetically between a and n
[^arn]	Returns a match for any character EXCEPT a, r, and n

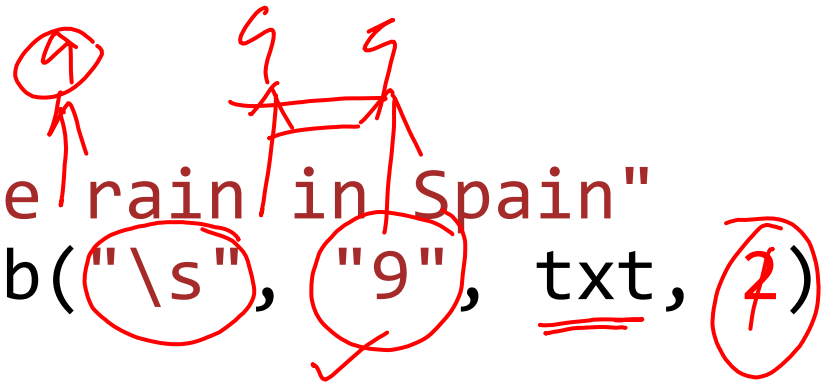
Examples

- ```
  import re

  txt = "The rain in Spain"
  x = re.findall("Portugal", txt)
  print(x)
  ```
- ```
  import re

  txt = "The rain in Spain"
  x = re.findall(“ain", txt)
  print(x)
  ```

- ```
  import re

  txt = "The rain in Spain"
  x = re.split("\s", txt)
  print(x)
  ```

- ```
  import re

  txt = "The rain in Spain"
  x = re.split("\s", txt, 1)
  print(x)
  ```

-
```
import re

txt = "The rain in Spain"
x = re.sub("\s", "9", txt, 2)
print(x)
```



- **Numbers `[0-9]+` or `\d+`**
- **Full Numeric Strings `^[0-9]+$` or `^\d+$`**
- Positive Integer Literals [1-9][0-9]*|0 or [1-9]\d*|0
- Full Integer Literals ^[+-]?[1-9][0-9]*|0$ or ^[+-]?[1-9]\d*|0$
- Identifiers (or Names) [a-zA-Z_][0-9a-zA-Z_]* or [a-zA-Z_]\w*
- Image Filenames ^\w+\.(gif|png|jpg|jpeg)$

- Extracting emails from a Text Document
- any character a-z, any digit 0-9 and symbol '_' followed by a '@' symbol and after this symbol we can again have any character, any digit and especially a dot.
- r"[\w.-]+@[\w.-]+“
- Date:
- r"(\d{4})-(\d{2})-(\d{2})
- Phone:
- \d{10}