From https://www.reddit.com/r/ProgrammerHumor/

REGULAR EXPRESSIONS

Luísa Coheur

TÉCNICO LISBOA

inesc id lisboa

Overview

- Learning objectives
- Topics
  - Regular Expressions
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, students should be able to know how to apply Regular Expressions and understand how useful they can be

TOPICS

Overview

- Learning objectives
- Topics
  - Regular Expressions
- Key takeaways
- Suggested readings

APPROACH: RULE-BASED

- First NLP systems were rule-based (hand-crafted)
- There are many frameworks that allow us to craft rules, which can be very sophisticated
- We will focus on Regular Expressions

REGULAR EXPRESSIONS

- Regular expressions, sometimes known as regex or re, can be used for:
  - searching
  - matching, and
  - manipulating text
- First NLP systems, as ELIZA, were totally based on regular expressions
- Currently, regular expressions can be combined with more sophisticated techniques, but are still useful

REGULAR EXPRESSIONS

- RE are case sensitive
- RE always match the biggest string
- Characters inside braces [] specify a disjunction of characters to match:
  - Example:
    - [ola]pp means “o”, “l” or “a” and will match “opp”, “lpp” and “app” and not “olapp”;
- Instead of [ABCDEFG], you can use [A–G] and instead of [0123...9] you can use [0-9]

REGULAR EXPRESSIONS

- • means any character
  - Example:
    - a.c represents abc, aac, acc, adc, a1c, a*c, etc.
- Considering what appears before:
  - ? means zero or one
  - * means zero or more (the wild card)
  - + means one or more
    - Example:
      - a? represents ε (empty string) and a
      - a* represents ε, a, aa, aaa, aaaa, ...
      - aa* represents a, aa, aaa, aaa, …
      - [ab]* represents aaa, abab, bbb, ...

REGULAR EXPRESSIONS

- ^ indicates the beginning of a line
- $ indicates the end of a line.
- | stands for the disjunction
- () groups tokens
- {n} = n occurrences of previous element
- {n,m} = between n and m occurrences of previous element
- {n,} =a t least n occurrences of previous element

REGULAR EXPRESSIONS

- \d represents any digit
- \D means any character that is not a digit
- \w means any alpha-numeric character or a space
- \W means any character that is not alpha-numeric
- \s means any space (tab, blank, ...)
- \S means any character that is not a space

REGULAR EXPRESSIONS

- *, \., \? represent, respectively, the special characters *, . and ?
- \n, \t represent a newline and tab
- [] and ^ can also be used to declare characters that should not appear in the RE.
  - Example:
    - [^a] means any character, except a

REGULAR EXPRESSIONS: APPLICATIONS

- Text Cleaning
  - Removing HTML Tags from Web Scraped Data:
    - A regex pattern like /<[^>]+>/g can be used to find and remove all HTML tags (g means “everywhere”)
- Pattern Recognition/data extraction
  - Identifying Email Addresses:
    - The regex pattern \b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b matches most email address formats
  - Extracting Dates:
    - A regex can help identify and extract the various formats of dates, useful for timeline analysis or event tracking

REGULAR EXPRESSIONS: APPLICATIONS

- Spam Detection:
  - Regular expressions can be instrumental in identifying common characteristics of spam messages, such as excessive use of capital letters, the presence of certain phrases (e.g., “BUY NOW”, “FREE”, “CLICK HERE”), or suspicious URLs

ACTIVE LEARNING MOMENT

EXERCISE

- Fill in the table, one letter in each cell
- Regular expressions indicate the letters that should appear in each column/line.
  - Example:
    - First line: you might have an H in the first cell and an E in the second, or an L in each cell or, at least, an O in one of the cells:

	[^SPEAK]+	EP|IP|EF
HE|LL|O+		
[PLEASE]+		

EXERCISE

	[^SPEAK]+	EP|IP|EF
HE|LL|O+	H	E
[PLEASE]+	L	P

KEY TAKEWAYS

KEY TAKEWAYS

- The syntax of Regular Expressions and understand that they are still very useful in NLP, even in the Deep Learning era

SUGGESTED READINGS

READINGS

- Sebenta: chapter about Regular Expressions
- Jurafsky: 2.1