Практическое задание №1. Осень 2024

Введение

Регулярные выражения - мощный инструмент для обработки текстовых данных, включая тексты на естественных языках. Регулярные выражения используются в различных задачах, таких как предварительная обработка данных, системы интеллектуального анализа информации на основе правил, сопоставление с образцом, разработка текстовых функций, валидация входных данных, извлечение данных из интернета, и т. д. Однако не каждую NLP задачу целесообразно решать с помощью регулярных выражений.

Данное задание разбито на две части: задачи первой части можно (и, пожалуй, нужно) решать с помощью регулярных выражений, а вот задачи второй части либо по определению невозможно решить, используя только регулярные выражения, либо сложность выражения настолько велика, что их применение нецелесообразно.

Постановка задачи

Часть 1.

Требуется составить регулярные выражения, для решения следующих независимых подзадач:

- проверка корректности пароля;
- проверка корректности web цвета;
- токенизация математического выражения;
- проверка корректности даты.

1.1. Проверка корректности пароля

В рамках этой подзадачи требуется разработать регулярное выражение, которым возможно проверить, может ли являться входная строка (целиком) корректным паролем.

Ограничения на пароли:

- пароль должен содержать только латинские символы, цифры и специальные символы ^$%@#&*!?
- пароль должен состоять из не менее чем восьми символов
- пароль должен содержать по крайней мере один латинский символ в верхнем регистре
- пароль должен содержать по крайней мере один латинский символ в нижнем регистре
- пароль должен содержать по крайней мере одну цифру
- пароль должен содержать по крайней мере два различных специальных символа
- пароль не должен содержать двух одинаковых символов подряд

Примеры корректных паролей:

- rtG3FG!Tr^e
- aA1!*!1Aa
- oF^a1D@y5e6
- enroi#$rkdeR#$092uwedchf34tguv394h

Примеры некорректных паролей:

- пароль
- password
- qwerty
- lOngPa$$W0Rd

1.2. Проверка корректности web цвета

В рамках этой подзадачи требуется разработать регулярное выражение, которым возможно проверить, может ли являться входная строка (целиком) корректной записью цвета в одном из трёх web форматов:

- **rgb:** `rgb(r, g, b)`, где «r, g, b» - это комбинация из трёх целых чисел (от 0 до 255) или трёх процентных значений (от 0% до 100%), перечисленных через запятую.
- **hex** (шестнадцатеричный код цвета, `#rrggbb`) – это шестизначное представление цвета в RGB пространстве. Первые две цифры (rr) – представляют собой красное значение,

следующие две – зелёное значение (gg), а последние – синее значение (bb). Перед значениями каналов предшествует символ #. Также допускается сокращённый вид записи – по одной цифре – #rgb

- **hsl** (тон, насыщенность и светлота, `hsl(h, s, l)`) - записывается похожим на rgb формат образом. Тон – целое число в диапазоне от 0 до 360, насыщенность и светлота - целочисленные процентные значения.

Примеры корректных цветов:

- #21f48D
- #888
- rgb(255, 255,255)
- rgb(10%, 20%, 0%)
- hsl(200,100%,50%)
- hsl(0, 0%, 0%)

Примеры некорректных цветов:

- #2345
- ffffff
- rgb(257, 50, 10)
- hsl(20, 10, 0.5)
- hsl(34%, 20%, 50%)

1.3. Токенизация математического выражения

Целью данной подзадачи является создание регулярного выражения, способного разбить строку, содержащую математическое выражение, на токены (элементарные части) и определить тип этих токенов. Математическое выражение может состоять из следующих элементов:

- **переменная (тип variable)** – строка из латинских букв, цифр и символа нижнего подчёркивания (_), начинающаяся не с цифры: a, var123, some_var_name;
- **число (тип number)** – строка, являющаяся целым или вещественным числом в общей форме без знака (в качестве разделителя целой и дробной части – точка): 42, 123456789, 23.567, 0.6734537
- **константа (тип constant)** – строка из списка: pi, e, sqrt2, ln2, ln10
- **функция (тип function)** – строка из списка: sin, cos, tg, ctg, tan, cot, sinh, cosh, th, cth, tanh, coth, ln, lg, log, exp, sqrt, cbrt, abs, sign
- **операция (тип operator)** – строка из списка ^, *, /, -, +
- **круглые скобки (тип `left_parenthesis` и `right_parenthesis`)**

Токены выражения могут отделяться друг от друга произвольным количеством пробелов (возможно, нулевым). Выделять пробельные символы в токены не нужно. Названия переменных не могут совпадать с именами функций и констант.

Примеры выражений и ожидаемых токенов:

- **выражение:** `"sin(x) + cos(y) * 2.5"`
  токены:
  ```
  {"type": "function", "span": [0, 3]},
  {"type": "left_parenthesis", "span": [3, 4]},
  {"type": "variable", "span": [4, 5]},
  {"type": "right_parenthesis", "span": [5, 6]},
  {"type": "operator", "span": [7, 8]},
  {"type": "function", "span": [9, 12]},
  {"type": "left_parenthesis", "span": [12, 13]},
  {"type": "variable", "span": [13, 14]},
  {"type": "right_parenthesis", "span": [14, 15]},
  {"type": "operator", "span": [16, 17]},
  {"type": "number", "span": [18, 21]}
  ```
- **выражение:** `"pi    +        usO5NlMvU"`
  токены:
  ```
  {"type": "constant", "span": [0, 2]},
  {"type": "operator", "span": [6, 7]},
  {"type": "variable", "span": [15, 24]}
  ```

- **выражение:** `"(  63393394.98 /8505  )"`
  токены:
  ```
  {"type": "left_parenthesis", "span": [0, 1]},
  {"type": "number", "span": [10, 21]},
  {"type": "operator", "span": [22, 23]},
  {"type": "number", "span": [23, 27]},
  {"type": "right_parenthesis", "span": [33, 34]}
  ```

1.4. Проверка корректности даты

Требуется разработать регулярное выражение, способное определить, является ли входная строка (целиком) датой в одном из нескольких форматов. Допускаются следующие форматы даты:

- **день.месяц.год** (14.09.2023, 5.02.1995, 01.4.2012)
- **день/месяц/год** (14/09/2023, 5/02/1995, 01/4/2012)
- **день-месяц-год** (14-09-2023, 5-02-1995, 01-4-2012)
- **год.месяц.день** (2023.09.14, 1995.02.5, 2012.4.01)
- **год/месяц/день** (2023/09/14, 1995/02/5, 2012/4/01)
- **год-месяц-день** (2023-09-14, 1995-02-5, 2012-4-01)
- **день месяц_rus год** (14 сентября 2023, 5 февраля 1995, 01 апреля 2012)
- **Месяц_eng день, год** (September 14, 2023, February 5, 1995, April 01, 2012)
- **Мес_eng день, год** (Sep 14, 2023, Feb 5, 1995, Apr 01, 2012)
- **год, Месяц_eng день** (2023, September 14, 1995, February 5, 2012, April 01)
- **год, Мес_eng день** (2023, Sep 14, 1995, Feb 5, 2012, Apr 01)

Примеры корректных дат:

- 20 января 1806
- 1924, July 25
- 26/09/1635
- 3.1.1506

Примеры некорректных дат:

- 25.08-1002
- декабря 19, 1838
- 8.20.1973
- Jun 7, -1563

Примечание: год должен быть неотрицательным.

Часть 2.

Требуется составить регулярные выражения, для решения следующих независимых подзадач:

- проверка корректности скобочного выражения;
- разбиение текста на предложения;
- поиск в тексте именованных сущностей типа PERSON;
- извлечение данных из HTML страницы;

2.1. Проверка корректности скобочного выражения

В рамках этой подзадачи требуется разработать регулярное выражение, которым возможно проверить, является ли входная строка (целиком) корректным скобочным выражением. Скобки могут быть трёх типов: (), {} и [].
Правильная скобочная последовательность формально определяется следующим образом:

- пустая строка — правильная скобочная последовательность;
- правильная скобочная последовательность, взятая в скобки — правильная скобочная последовательность;
- правильная скобочная последовательность, к которой приписана слева или справа правильная скобочная последовательность — тоже правильная скобочная последовательность.

Примеры корректных выражений	Примеры некорректных выражений
() {[]} {[{[()]}]}	) (}[] {{[{])((){

2.2. Разбиение текста на предложения

В рамках этой подзадачи требуется разработать регулярное выражение, которым возможно извлечь из текста предложения (разбить текст на предложения). В качестве источника текстов используются рецензии к фильмам на сайте кинопоиска. Примеры можно найти по ссылке: https://www.kinopoisk.ru/reviews/type/comment/period/month.
Регулярное выражение должно представлять из себя именованную группу sentence: `(?P<sentence>)`.

Пример 1:
Что сразу бросается в глаза, так это нестандартная рисовка и отсутствие эмоций на лицах, в первом сезоне "смешные" моменты были со вставками глупых лиц, как в аниме начала 2000х. Потом, видимо, поняли, что это уже не круто и от таких ходов отказались. Если не обращать внимание на картинку, а полностью окунуться в сюжет, в принципе очень даже смотрибельно. Интересно следить за развитием персонажа, как он сначала вершит правосудие над обидчиками своего отца, а потом глубоко погружается в овладение магией разного толка. Присутствует жестокость и почти нет фансервиса, что радует. Монстры от сезона к сезону от топорных моделек переходят в состояние "неплохо", авторы исправляют свои ошибки, как и все, за что берётся копировать китайская нация. В общем вас ждет вырвиглазная рисовка с неплохим сюжетом и поиском приемлемой озвучки.
7 из 10

Особенности разбиения текстов со списками.
Для рецензий, содержащих списки, ожидается следующий алгоритм разбиения:

- если элементы списка состоят из нескольких предложений, то предложение перед списком завершается до списка, а каждый пункт списка разбивается на независимые предложения, причём первое предложение пункта включает в себя маркер списка;

- в противном случае (обычно пункты таких списков завершаются символом ; за исключением последнего пункта, завершающегося точкой) предложением является весь список и предшествующее ему предложение.

Пример 2:
Резюмируя можно сказать:
1. Герои объединяются под сомнительным предлогом;
2. Их отношения выглядят неестественно;
3. Карьерный рост Кэсси не прокатил бы даже в диснеевской сказке.
Несмотря на то, что оценка в общепринятом понимании относится к серой зоне, субъективно фильм оставляет приятное послевкусие.

Пример 3.
Кратко и по пунктам:
1. Начал смотреть, потому что новый сериал по подписке.
2. Сразу "проглотил" полторы серии, запнулся на отсылке к "лихим 90-м" и бандитам, не нравится мне такое. Решил не досматривать.
3. Через день всё-таки любопытство взяло верх. Сказал себе: если будет неожиданный поворот в банальном сюжете, досмотрю. Поворот случился, пришлось смотреть весь сериал.

Внимание: тексты рецензий не обязательно являются полностью корректными относительно правил русского языка. Следует учитывать это при составлении регулярных выражений.

2.3. Поиск в тексте именованных сущностей типа PERSON

Целью данной подзадачи является создание регулярного выражения, способного найти в тексте на русском языке именованные сущности типа PERSON. Под персонами следует понимать следующее определение: человек (реальный или вымышленный) со своими индивидуальными особенностями с социокультурной точки зрения. Регулярное выражение должно находить персон с помощью именованной группы person: `(?P<person>)`.

Пример:
Нургалиев уволил начальника УВД Томской области.
Начальник УВД Томской области Виктор Гречман освобожден от занимаемой должности. Как сообщает "Интерфакс" со ссылкой на пресс-службу МВД, это решение принял глава ведомства Рашид Нургалиев по поручению президента РФ Дмитрия Медведева.

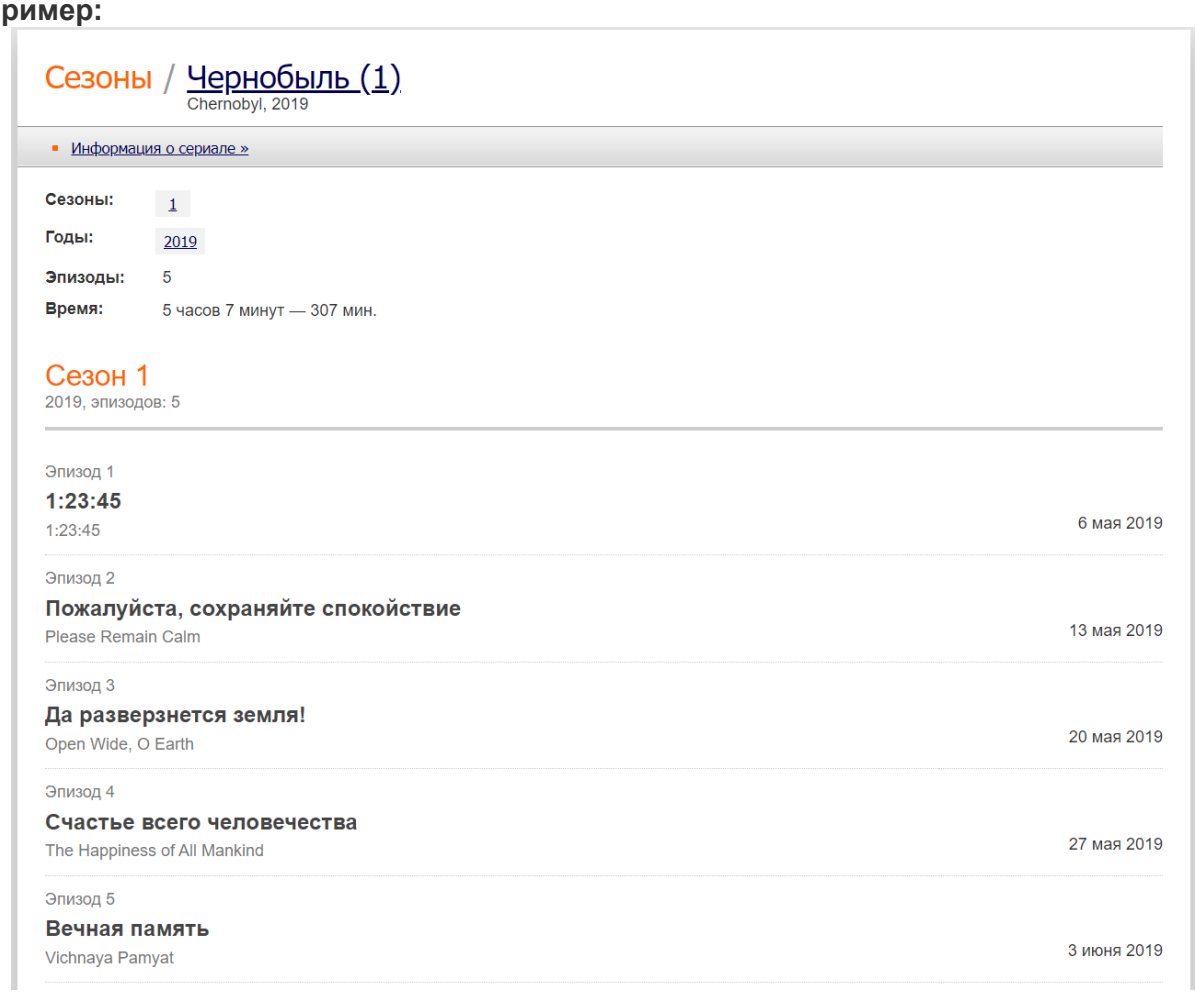
2.4. Извлечение данных из HTML страницы

Требуется разработать регулярное выражение, способное выделить из html кода страницы различные сведения о сериалах. В качестве источника используются страницы с эпизодами на Кинопоиске вида https://www.kinopoisk.ru/film/{id}/episodes/, где вместо {id} находится идентификатор сериала, состоящий из цифр.

Извлекаемые данные:
- **общая информация:** название сериала (`name`), общее количество эпизодов в сериале (`episodes_count`);
- **информация об эпизоде:** номер (`episode_number`), название (`episode_name`), оригинальное название (`episode_original_name`), дата выхода (`episode_date`);
- **информация о сезоне:** номер сезона (`season`), год (`season_year`), количество эпизодов (`season_episodes`).

В скобках указаны именованные группы, в которые необходимо заключить искомую информацию.

Пример:



Сезоны / Чернобыль (1)
Chernobyl, 2019

Информация о сериале »

Сезоны: 1
Годы: 2019
Эпизоды: 5
Время: 5 часов 7 минут — 307 мин.

Сезон 1
2019, эпизодов: 5

Эпизод 1
1:23:45
1:23:45
6 мая 2019

Эпизод 2
Пожалуйста, сохраняйте спокойствие
Please Remain Calm
13 мая 2019

Эпизод 3
Да разверзнется земля!
Open Wide, O Earth
20 мая 2019

Эпизод 4
Счастье всего человечества
The Happiness of All Mankind
27 мая 2019

Эпизод 5
Вечная память
Vichnaya Pamyat
3 июня 2019

Извлекаемая информация:

- "Чернобыль (1)" (name)
- "5" (episodes_count)
- "1" (season)
- "2019" (season_year)
- "5" (season_episodes)
- "1" (episode_number)
- "1:23:45" (episode_name)
- "1:23:45" (episode_original_name)
- "6 мая 2019" (episode_date)
- "2" (episode_number)
- "Пожалуйста, сохраняйте спокойствие" (episode_name)
- "Please Remain Calm" (episode_original_name)
- "13 мая 2019" (episode_date)
- "3" (episode_number)
- "Да разверзнется земля!" (episode_name)
- "Open Wide, O Earth" (episode_original_name)
- "20 мая 2019" (episode_date)
- "4" (episode_number)
- "Счастье всего человечества" (episode_name)
- "The Happiness of All Mankind" (episode_original_name)
- "27 мая 2019" (episode_date)
- "5" (episode_number)
- "Вечная память" (episode_name)
- "Vichnaya Pamyat" (episode_original_name)
- "3 июня 2019" (episode_date)

Общая информация

Для получения выделяемых регулярными выражениями данных для токенизации выражения, а также всех задач второй части следует использовать следующий код:

```
entities = set()
for match in regexp.finditer(html):
    for key, value in match.groupdict().items():
        if value is not None:
            start, end = match.span(key)
            entities.add((start, end, key))
```

Примеры входных данных для заданий 2.2 и 2.4 доступны по ссылке:

- в папке sentences представлены тексты рецензий, в которых каждое предложение выделено символами { и };
- в папке series представлены примеры html разметки страниц с информацией о сериалах.

Внимание: примеры данных даются исключительно в ознакомительных целях для выполнения данного задания. Использование их для других целей запрещено.

Решение задачи

Теоретические аспекты

- docs.python - Документация на библиотеку регулярных выражений в Python3
- Habr - Регулярные выражения в Python. От простого к сложному:
- regex101 - Тестирование и отладка регулярных выражений с возможностью выбора языка программирования:

Тестирование

На личной странице (practicum.tpc.ispras.ru/submissions/regexp) находится статистика со всеми результатами в т.ч. результатами последнего тестирования (дата, метрики качества).

На странице practicum.tpc.ispras.ru/results доступны результаты всех участников. Таблица обновляется раз в неделю.

Загрузка решения

Загружаемый файл должен представлять собой zip архив с любым именем. Архив должен обязательно содержать:

- Решение **в файле solution.py**. В файле должны содержаться следующие строки, содержащие регулярные выражения:
  1. Регулярное выражение для проверки пароля на корректность (`PASSWORD_REGEXP`)
  2. Регулярное выражение для проверки цвета (`COLOR_REGEXP`)
  3. Регулярное выражение для токенизации выражения (`EXPRESSION_REGEXP`)
  4. Регулярное выражение для проверки дат (`DATES_REGEXP`)
  5. Регулярное выражение для проверки скобочного выражения на корректность (`PARENTHESIS_REGEXP`);
  6. Регулярное выражение для разбиения на предложения (`SENTENCES_REGEXP`);
  7. Регулярное выражение для поиска персон (`PERSONS_REGEXP`);
  8. Регулярное выражение для извлечения данных о сериалах (`SERIES_REGEXP`).
- **Описание** найденных регулярных выражений в файле description.txt. Пожалуйста, напишите подробное описание, как были найдены регулярные выражения. Это описание будет выложено вместе с решением после завершения курса.

Каждое регулярное выражение должно являться строкой, записанной по правилам python regexp. В противном случае система проверки выдаст ошибку.

Пример решения, возвращающего пустые результаты для всех подзадач:

```
PASSWORD_REGEXP = r''
COLOR_REGEXP = r''
EXPRESSION_REGEXP = r''
DATES_REGEXP = r''

PARENTHESIS_REGEXP = r''
SENTENCES_REGEXP = r''
PERSONS_REGEXP = r''
SERIES_REGEXP = r''
```

Ограничения

- Каждую неделю можно послать не более 10 решений.
- Внимание! Итоговое тестирование будет проводиться на **последнем загруженном** решении.
- Размер загружаемого архива не должен превышать 15Мб.
- Время тестирования каждого регулярного выражения не должно превышать 3 секунд на тексте из 5000 символов.
- На проверяющей машине доступно 16 Гб оперативной памяти.

Оценка качества

Для оценки задания используется усредненная F_1 мера по каждой из подзадач. Для подзадач валидации используется F_1 мера для задачи бинарной классификации

$$P = \frac{tp}{tp + fp}, \qquad R = \frac{tp}{tp + fn}, \qquad F_1 = \frac{2PR}{P + R};$$

Для оценки остальных подзадач используется micro-averaged F_1, мера точного совпадения границ искомых подстрок:

$$P = \frac{|correct|}{|predicted|}, \qquad R = \frac{|correct|}{|expected|}, \qquad F_1 = \frac{2PR}{P + R};$$

При проверке результатов валидации строки, в случае превышения ограничения по времени, считается, что ответ противоположен правильному.