

Практическое задание №2. Осень 2022

Введение

Регулярные выражения - мощный инструмент для обработки текстовых данных, включая тексты на естественных языках, однако использовать только их для решения различных NLP задач далеко не всегда целесообразно. В этом задании представлены задачи, идеальное решение которых с помощью регулярных выражений либо в принципе невозможно либо крайне затруднительно и неэффективно.

Постановка задачи

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;

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. 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.

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**.

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 развернется 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 развернется 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 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 и 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](https://docs.python.org/3/library/re.html) - Документация на библиотеку регулярных выражений в Python3
- [Habr](https://habr.com/ru/post/444444/) - Регулярные выражения в Python. От простого к сложному:
- [regex101](https://regex101.com/) - Тестирование и отладка регулярных выражений с возможностью выбора языка программирования:
- [towardsdatascience](https://towardsdatascience.com/regular-expressions-for-nlp-1a1e1e1e1e1e) - Применение регулярных выражений для NLP:

Тестирование

На личной странице (2022.tpc.ispras.ru/submissions/regex2) находится статистика со всеми результатами в т.ч. результатами последнего тестирования (дата, метрики качества).

На странице 2022.tpc.ispras.ru/results/regex2 доступны результаты всех участников. Решения перезапускаются раз в неделю по средам в 00:00.

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 скобочного выражения на корректность (PARENTHESIS_REGEX);
 2. Регулярное выражение для разбиения на предложения (SENTENCES_REGEX);
 3. Регулярное выражение для поиска персон (PERSONS_REGEX);
 4. Регулярное выражение для извлечения данных о сериалах (SERIES_REGEX).
- 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 regex. В противном случае система проверки выдаст ошибку.

Пример решения, возвращающего пустые результаты для всех подзадач:

```
PARENTHESIS_REGEX = r''  
SENTENCES_REGEX = r''  
PERSONS_REGEX = r''  
SERIES_REGEX = 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 подзадачи валидации используется F_1 мера для задачи бинарной классификации

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

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

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

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