

Введение

Решение задачи о фильтрации представлено при помощи парсинга строки ввода пользователей и нахождения в ней ключевых слов (имя, фамилия, отчество) и поиск совпадений с записями о пользователях в хранилище данных, либо же поиск в хранилище по подстроке.

Алгоритм фильтрации

Введённую пользователем строку мы разбиваем на слова, каждое слово анализируем и находим частичное или полное совпадение в «словаре» со словами. Также каждому анализируемому слову сопоставляем score (качество совпадения от 0 до 1). Основываясь на теге (имя нарицательное/собственное) в словаре и score (рассматриваем только 0.5 и более), делаем вывод о том является ли это слово полноценным именем. Если да, то делаем нормальную форму и ищем в хранилище совпадение, если нет, то пытаемся найти вхождение введённой строки в чьё-то имя.

Когда мы находим совпадение в базе, то мы складываем score этого слова (или длину подстроки/длину слова) в общий результат (массив с очками совпадения). После обработки каждого введённого слова, мы ищем максимальную сумму очков и элемент в хранилище, который ему соответствует. Если такой найден, то выводим данные о найденном пользователе. Если максимальный score нулевой, то мы не можем вывести соответствующий элемент.

Алгоритм старается всегда найти хоть какое-то совпадение, например, для строки Андрей Петров будет выведено совпадение с пользователем Иваном Петровым, если лучшего совпадения нет.

Инструменты

Для реализации данного алгоритма я выбрал ЯП Python, библиотеку rymorphy3 с русским словарём. Python + rymorphy3 позволяет сделать быстрое решение с относительно качественным уровнем анализа слов.

Программная реализация

Github с решением (файл task3.py)

<https://github.com/andreyvydra/olimp>

```
import pymorphy3

morph_ru = pymorphy3.MorphAnalyzer(lang='ru')

data = (
    ("Пётр", "Иванов", "Василиевич"),
    ("Иван", "Пётров", "Василиевич"),
    ("Василий", "Пётров", "Иванович"),
    ("Андрей", "Выдра", "Михаилович"),
    ("Дмитрий", "Примеров", "Тестович"),
    ("Дмитрий", "Викторов", "Алексеевич"),
    ("Александр", "Недвижимов", "Ильич")
)

inputs = (
    "Петр Васильевич Иванов",
    "Иванов Петр Васильевич",
    "Петр Иванов",
    "иванов Петр",
    "Петр",
    "михалыч",
    "Выдра",
    "выдр",
    "Ивано",
    "движ",
    "ор",
    "Андр",
    "Примеров Дмитрий",
    "Иванов Васильевич",
    "Андрей Пётров"
)

for inp in inputs:
    user_input = inp.strip()
    points = [0 for i in range(len(data))]
    for word in user_input.split():
        best_parse = morph_ru.parse(word)[0]
        tag = best_parse.tag
        idx = -1
        if best_parse.score > 0.5:
            if "Name" in tag:
                idx = 0
            elif "Patr" in tag:
                idx = 2
            elif "Surn" in tag:
                idx = 1

        for i, item in enumerate(data):
            if idx == -1:
                for element in item:
                    points[i] += len(word) / len(element) if word.lower() in element.lower() else 0
                continue
            if best_parse.normal_form.lower() == item[idx].lower():
```

```

        points[i] += best_parse.score

    idx = 0
    max_points = 0
    for i, point in enumerate(points):
        if point > max_points:
            max_points = point
            idx = i

    print("_____")
    print("Для строки -", user_input)
    if max_points > 0:
        print("Совпадение найдено, данные о пользователе: ", data[idx])
    else:
        print("Совпадения не обнаружено")

```

Демонстрация на тестовых примерах

В хранилище данных находятся следующие записи:

```

data = (
    ("Пётр", "Иванов", "Василиевич"),
    ("Иван", "Пёторов", "Василиевич"),
    ("Василий", "Пёторов", "Иванович"),
    ("Андрей", "Выдра", "Михаилович"),
    ("Дмитрий", "Примеров", "Тестович"),
    ("Дмитрий", "Викторов", "Алексеевич"),
    ("Александр", "Недвижимов", "Ильич")
)

```

Работа программы при различных вводах:

Для строки - Петр Васильевич Иванов

Совпадение найдено, данные о пользователе: ('Пётр', 'Иванов', 'Василиевич')

Для строки - Иванов Петр Васильевич

Совпадение найдено, данные о пользователе: ('Пётр', 'Иванов', 'Василиевич')

Для строки - Петр Иванов

Совпадение найдено, данные о пользователе: ('Пётр', 'Иванов', 'Василиевич')

Для строки - иванов Петр

Совпадение найдено, данные о пользователе: ('Пётр', 'Иванов', 'Василиевич')

Для строки - Петр

Совпадение найдено, данные о пользователе: ('Пётр', 'Иванов', 'Василиевич')

Для строки - михалыч

Совпадение найдено, данные о пользователе: ('Андрей', 'Выдра', 'Михаилович')

Для строки - Выдра

Совпадение найдено, данные о пользователе: ('Андрей', 'Выдра', 'Михаилович')

Для строки - выдр

Совпадение найдено, данные о пользователе: ('Андрей', 'Выдра', 'Михаилович')

Для строки - Ивано

Совпадение найдено, данные о пользователе: ('Пётр', 'Иванов', 'Василиевич')

Для строки - движ

Совпадение найдено, данные о пользователе: ('Александр', 'Недвижимов', 'Ильич')

Для строки - ор

Совпадение найдено, данные о пользователе: ('Иван', 'Пёторов', 'Василиевич')

Для строки - Андр

Совпадение найдено, данные о пользователе: ('Андрей', 'Выдра', 'Михаилович')

Для строки - Примеров Дмитрий

Совпадение найдено, данные о пользователе: ('Дмитрий', 'Примеров', 'Тестович')

Для строки - Иванов Васильевич

Совпадение найдено, данные о пользователе: ('Пётр', 'Иванов', 'Василиевич')

Для строки - Андрей Пётров

Совпадение найдено, данные о пользователе: ('Иван', 'Пётров', 'Василиевич')

Заключение

В этом задании я написал фильтр для поиска пользователей по текстовому запросу. Алгоритм показал свою эффективность на различных входных данных.