

Поиск адреса электронной почты с помощью регулярных выражений на языке python

Эрдман Александр Алексеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Данная статья содержит описание программы, которая способна с помощью регулярных выражений находить адрес электронной почты в тексте. Программа написана на языке программирования python с использованием модуля re, способным осуществлять операции с регулярными выражениями. Практическим результатом является программа, позволяющая находить в текстовом файле адрес электронной почты и выводить его в текстовый файл.

Ключевые слова: модуль re, поиск e-mail адреса в тексте, python

Search for an email address using regular expressions in python

Erdman Alexander Alekseevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article contains a description of a program that is capable of using regular expressions to find an email address in the text. The program is written in the python programming language using the re module, capable of performing operations with regular expressions. The practical result is a program that allows you to find an email address in a text file and output it to a text file.

Keywords: re module, search for e-mail addresses in text, python

1 Введение

1.1 Актуальность

Электронная почта на сегодняшний день является одним из самых популярных способов связи между людьми. Помимо бытового уровня, электронная почта широко используется и в других сферах деятельности. Например, подавляющее большинство работодателей оставляют свои e-mail адреса для обратной связи с сотрудниками, активно используют e-mail адреса обучающиеся различных учебных заведений для обратной связи со своими преподавателями. По причине широкого применения электронной почты, всё чаще в различного рода заявлениях или иных документах, содержащих личную информацию заявителя, указывается способ обратной связи в виде e-mail адреса. В некоторых случаях человеку, обрабатывающего подобно рода

документы, нужно найти лишь e-mail адрес заявителя. Для нахождения адресов удобно было бы использовать программу, нежели искать вручную.

1.2 Обзор исследований

О.Е. Смирнов реализовал процесс поиска соответствий шаблону в целевом графе кода на Python на этапе локализации с помощью изоморфизмов подграфов [1]. Е.С Локтев, В.А Шабзон и Н.С. Бутенко применили регулярные выражения в приложении адресная книга [2]. А.А. Дубова реализовала поиск данных с использованием регулярных выражений на языке python [3]. А.В. Багаев описал принцип работы регулярных выражений в языке программирования python [4]. А.А. Мусаев и Д.А. Григорьев решили задачу на извлечения знаний из текстовых сообщений [5].

1.3 Цель Исследования

Цель исследования – используя модуль re и язык программирования python, написать программу, которая будет находить e-mail адреса в указанном txt файле с данными, в числе которых будут e-mail адреса, и выводить найденные адреса в другой txt файл.

2 Материалы и методы

Программа будет написана на языке программирования python с использованием модуля re. Pycharm будет использован в качестве IDE.

3 Результаты и обсуждения

Для написания программы понадобится встроенный модуль re. Его нужно импортировать через команду “import re”, данный модуль позволит работать с регулярными выражениями (рис.1).

```
import re
```

Рис. 1. Импорт модуля re

Так как программа будет работать с файлами, то нужно добавить команду на открытие файла для чтения, команду на открытия файла для записи, в который непосредственно будет выведен результат программы, и сохранить прочитанный текст в переменную. (рис. 2).

```
import re
fin = open('tex.txt', mode='r', encoding='utf-8')
fout = open('res.txt', mode='w', encoding='utf-8')
text = fin.read()
```

Рис. 2. Команды на открытие файлов и добавление текста в переменную text

Теперь нужно добавить функцию для поиска e-mail адреса в тексте. Для этого нужно воспользоваться методом re.findall(). Метод состоит из двух

частей: первая часть представляет из себя шаблон e-mail адреса, вторая часть ссылается на переменную с текстом (рис. 3).

```
emails = re.findall('[a-zA-Z0-9_+-.]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+', text)
```

Рис. 3. Объявление функции через переменную

Основная функция прописана. Теперь нужно вывести результат программы. Для вывода в текстовый файл нужно использовать функцию `print()` с параметром `file=fout` (рис. 4).

```
for item in emails:  
    print(item, file=fout)
```

Рис. 4. Команда на вывод результата в текстовый файл

В конечном результате получена программа для нахождения e-mail адреса в тексте (рис. 5).

```
import re  
fin = open('tex.txt', mode='r', encoding='utf-8')  
fout = open('res.txt', mode='w', encoding='utf-8')  
text = fin.read()  
emails = re.findall("[a-zA-Z0-9_+-.]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+", text)  
for item in emails:  
    print(item, file=fout)
```

Рис. 5. Программа для поиска и вывода e-mail адреса в текстовый файл

Принцип действия программы можно показать на примере. Допустим дана практическая задача: “Преподавателю университета нужно собрать данные о студентах конкретной группы. Эти данные включают в себя: e-mail адрес, сколько лет студенту, работает или не работает, какой язык программирования использует, какую программу пишет”. Часто на практике студенты пишут требуемые данные не в строгом порядке, то есть могут менять местами конкретные данные. Допустим преподавателю важно в первую очередь узнать e-mail адрес студента. Учитывая, что данные записаны не в строгом порядке, e-mail адрес не виден сразу и его приходится искать вручную, то есть нужно прочесть все данные чтобы увидеть. Чтобы облегчить поиск e-mail адресов можно воспользоваться программой для их поиска.

Дан входной файл с данными студента и нужно найти почту и для удобства вывести её отдельно в другой файл (рис. 6).

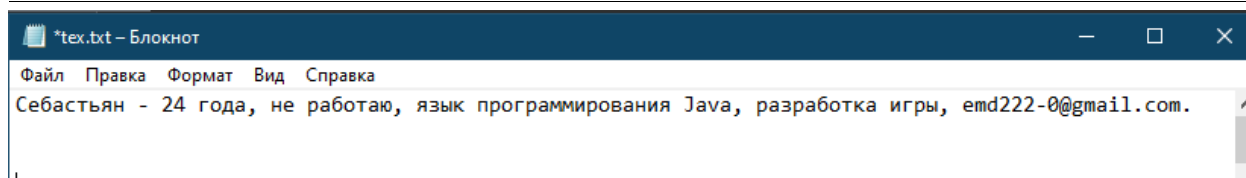


Рис. 6. Файл с входными данными

После выполнения программы получим результат (рис. 7).

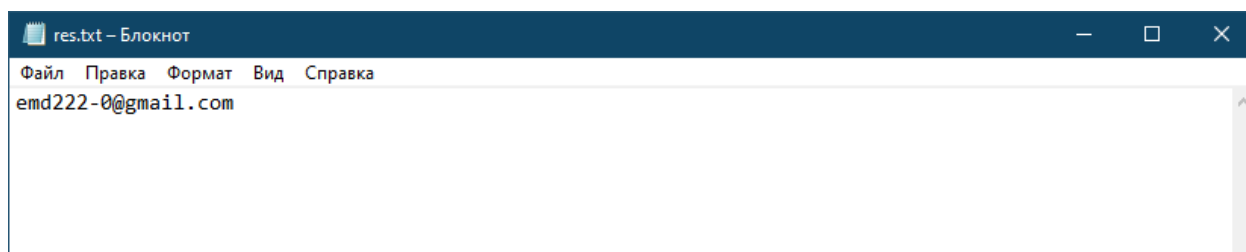


Рис. 7. Результат работы программы

4 Выводы

В ходе написания статьи была разработана программа для поиска e-mail адресов в тексте. Также был приведён реальный практический пример использования данной программы, с которым программа справилась.

Библиографический список

1. Дубова А.А. Поиск данных с использованием регулярных выражений // Международная научно-техническая конференция молодых ученых БГТУ им. ВГ Шухова. 2018. С. 3881-3885.
2. Мусаев А.А., Григорьев Д.А. Обзор современных технологий извлечения знаний из текстовых сообщений // COMPUTER. 2021. Т. 13. №. 6. С. 1291-1315.
3. Багаев А.В. Регулярные выражения в языке программирования python 3.5 // "Информационные технологии в образовании" ИТО Саратов 2016". 2016. С. 15-17.
4. Смирнов О.Е. Автоматическое изменение кода на Python в соответствии в графовыми шаблонами. 2021. С. 9-13
5. Локтев Е.С., Шабзон В.А., Бутенко Н.С. Использование регулярных выражений на примере приложения адресная книга // Концепции устойчивого развития науки в современных условиях. 2018. С. 177-179.