УДК 004.89

Применение метода расстояния Левенштейна библиотеки Fuzzywuzzy языка Python для исправления данных

Черкашин Александр Михайлович
Приамурский государственный университет имени Шолом-Алейхема
Студент

Научный руководитель:
Баженов Руслан Иванович
Приамурский государственный университет имени Шолом-Алейхема
к.п.н., доцент, зав. кафедрой информационных систем, математики и правовой информатики

Аннотация
Целью исследования является исправления в исходных данных, введенным человеком, до единых значений. В работе использовался алгоритм расстояния Левенштейна, реализованный в библиотеке fuzzywuzzy для python. Обработка данных позволила получить четкие значения, пригодные для восприятия машины.
Ключевые слова: Pandas, Python, Pandarallel, Fuzzywuzzy, электронная таблица, расстояние Левенштейна, неточные сравнения, нечеткие сравнения, нечеткий поиск, нечеткое соответствие.

Applying the Levenshtein distance method of the Python Fuzzywuzzy library to correct data

Cherkashin Alexander Mihailovich
Sholom-Aleichem Priamursky State University
student

Scientific adviser:
Bazhenov Ruslan Ivanovich
Sholom-Aleichem Priamursky State University
candidate of pedagogical sciences, associate professor, Head of the Department of Information Systems, Mathematics and Legal Informatics

Abstract
The aim of the study is to correct the original data entered by a person to a single value. We used the Levenshtein distance algorithm implemented in the fuzzywuzzy library for python. The processing of the data made it possible to obtain clear values suitable for the perception of the machine.
Keywords: Pandas, Python, Pandarallel, Fuzzywuzzy, spreadsheet, Levenshtein distance, fuzzy comparison, fuzzy search, fuzzy matching.

Актуальность статьи в том, что данные исходный таблицы поступают необработанным виде, имеют нечеткие значения, которые машина воспринимает как разные значения, человек воспринимает значение одинаковыми.

Ш. Гускенс и У. Херинга описали понятие определения термина расстояние Левенштейна - это мера расстояния редактирования строки. На основе лингвистических расстояний между диалектными разновидностями можно определить области диалектов [1].

М. Т. Резерфорд, Н. Д. Типански, Д. Ван, Г. Чен рассматривают решение комбинаторы синтаксического анализатора позволяющее объединить несколько простых парсеров, чтобы сформировать серию конгломератных парсеров, которые затем могут коллективно адресовать нерегулярный, сложный ввод [3].

М. Бааке, Р. Гигерих, У. Гримм рассматривают проблему метода сравнения L-расстояние с расстоянием на основе репрезентативного словаря [4].

М. Пиццол, Е. Виги, Р. Сакки применяют нечеткую логику для решение проблемы связанные с объединением данных цифровых платежей и данных ввода-вывода для изменения моделей потребления [5].

Целью исследования является исправления в исходных данных, введенным человеком, до единых значений.

В данной статье используется язык программирование python, библиотека fuzzywuzzy для нечеткого поиска строки и сравнения, библиотека Pandas для анализа и обработки таблиц, библиотека pandarallel для распараллеливания обработки таблиц по каждой строке.

Исходные данные полученные в необработанном виде, содержат нечеткие значения которые воспринимаются человеком, но непригодны для машинной обработки, например, создания сводных таблиц или логических связей с таблицами.

Исходные данные содержится в файле формата csv (Таблица 1):

id — уникальный идентификатор;

result_status — Состояние результатов;

result_value — Значение результатов.

Таблица 1 – Исходные данные

id	**result_status**	**result_value**
7	нет материала	открыта пробирка
51	нет материала	разлита пробирка
63	нет материала	пробирка пустая
72	нет материала	пусто
76	нет материала	пусто
230	нет материала	разлито
235	нет материала	пустая
832	нет материала	разлита

864	нет материала	разлита
1005	нет материала	разлита
1525	нет материала	разлита
1680	нет материала	разлита
1687	нет материала	разлита
1712	нет материала	разлита
1908	нет материала	разлита
2632	нет материала	разлита
2652	нет материала	разлита
2732	нет материала	разлита
3156	нет материала	разлита пробирка
3190	нет материала	разлита пробирка
3207	нет материала	разлита
3439	нет материала	пустая пробирка
3531	нет материала	разлита
3796	нет материала	разлита
4307	нет материала	разлита
4752	нет материала	разлито

Данная таблица (таблица 1) взята из лаборатории ФБУЗ «Центр гигиены и эпидемиологии в ЕАО» для сбора статистики, и была специально обработана. В столбец «result_value» выбраны только строки, все остальные типы данных отброшены.

Исходный код обработки для таблицы 1.

```
df = pd.read_excel("journal.xlsx")
a = df["Unnamed: 14"][df["Unnamed: 14"].isin(["открыта пробирка", "разлита пробирка", "пробирка пустая", "пусто", "разлито", "пустая", "разлита", "пустая пробирка"])]
df_out = pd.DataFrame({"id": a.index, "result_status": df["Результат"][a.index], "result_value": a})
df_out.to_csv("list_data.csv", sep='\t', encoding='utf-8', index=False)
```

Таблица 2. Список корректных значений

Идентификатор	**Значение**
0	Значение
1	открыта
2	пустая
3	разлита
4	разбита
5	повреждено
6	отсутствуют
7	зарезервировано

Загрузка библиотеки.

```
import pandas as pd
from fuzzywuzzy import process
from pandarallel import pandarallel
```

Инициализация библиотеки «pandarallel».

```
pandarallel.initialize()
```

Чтение файла «list_data.csv» (таблица 1).

```
df = pd.read_csv("list_data.csv", sep="\t")
```

Определение переменных «result_value_type» (Таблица 2).

```
result_value_type = [
        "Значение",
        "открыта",
        "пустая",
        "разлита",
        "разбита",
        "повреждено",
        "отсутствуют",
        "зарезервировано"
]
```

Обработка таблицы. (Таблица 1)

```
res_raw = pd.Series(df.result_value.unique()).parallel_apply(
   lambda x: list(process.extractOne(x, result_value_type)) + [x]
)
```

Функция pd.Series создает одномерный массив (временный ряд) [2]. df.result_value — выбирает столбец «result_value», функция unique выбирает уникальные строки из таблицы 1, в столбеце «result_value». Функция parallel_apply взята из библиотеки «pandarallel» выполняет многопоточную обработку, каждый поток читает строки из таблицы и вызывается в lambda. Функция process.extractOne выполняет поиск и сравнение строки с использованием алгоритма расстояние Левенштейна, и находит максимальную точность совпадения. Переменная x — текущая строка таблицы, result_value_type — Список (таблица 2). Возвращает значение виде списка.

Переменная res_raw, полученная в результате вычисления, представляется в виде массива (временного ряда), а для преобразования в таблицу используется следующий код.

Преобразование из массива в таблицу заданным заголовки.

```
res = pd.DataFrame(list(res_raw), columns=["name", "p", "orig_name"])
```

Условная выборка res.p больше 60, отбрасывает все неточные значение меньше 60.

```
res_p = res[res.p > 60]
```

Таблица 3 – Результаты вычисления переменной rse_p

id	**name**	**p**	**orig_name**
0	открыта	90	открыта пробирка
1	разлита	90	разлита пробирка
2	пустая	90	пробирка пустая
3	пустая	73	пусто
4	разлита	86	разлито
5	пустая	100	пустая
6	разлита	100	разлита
7	пустая	90	пустая пробирка

В таблице 3. name — взята из переменной result_value_type (таблица 2). p — точность, значение от 0 до 100, 0 — неточный, 100 — точный. orig_name — столбец, взят из таблицы 1 столбец result_value.

Замена значения в таблице 1, столбец result_value в таблица 3 столбец name.

```
df.result_value.replace(list(res_p.orig_name), list(res_p.name), inplace=True)
```

Таблица 4 – Обработанная таблица

id	**result_status**	**result_value**
7	нет материала	открыта
51	нет материала	разлита
63	нет материала	пустая
72	нет материала	пустая
76	нет материала	пустая
230	нет материала	разлита
235	нет материала	пустая
832	нет материала	разлита

864	нет материала	разлита
1005	нет материала	разлита
1525	нет материала	разлита
1680	нет материала	разлита
1687	нет материала	разлита
1712	нет материала	разлита
1908	нет материала	разлита
2632	нет материала	разлита
2652	нет материала	разлита
2732	нет материала	разлита
3156	нет материала	разлита
3190	нет материала	разлита
3207	нет материала	разлита
3439	нет материала	пустая
3531	нет материала	разлита
3796	нет материала	разлита
4307	нет материала	разлита
4752	нет материала	разлита

Вывод: в результате обработки данных (Таблица 4) ячейка result_value содержит четкие значения, пригодные для восприятия человеком.

Библиографический список

1. Gooskens C., Heeringa W. Perceptive evaluation of Levenshtein dialect distance measurements using Norwegian dialect data // Language variation and change. 2004. Т. 16. № 3. С. 189.
2. pandas.Series — pandas 1.2.1 documentation // pandas URL: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.html (дата обращения: 2021-01-21).
3. Rutherford M. et al. Processing of ICARTT data files using fuzzy matching and parser combinators //2014 International Conference on Artificial Intelligence (ICAI'14). 2014. С. 217-220.
4. Michael Baake, Robert Giegerich, Uwe Grimm Surprises in approximating Levenshtein distances // Theoretical Biology. 2006. Т. 243. С. 279-282.
5. Pizzol M., Vighi E., Sacchi R. Challenges in coupling digital payments data and input-output data to change consumption patterns //Procedia CIRP. 2018. Т. 69. С. 633-637.