Aufgabe 1: Störung

Team-ID: 00543
Team: pip install knowledge (Imanuel Fehse)
Bearbeiter dieser Aufgabe: Imanuel Fehse
15. November 2022

Inhaltsverzeichnis

Aufgabe

Hilf Alice und Bob und schreibe ein Programm, das für einen Lückensatz nach passenden Stellen im Lieblingsbuch sucht.

Lösungsidee

Zuerst wird der Text „Alice im Wunderland“ aus einer Datei eingelesen und formatiert (Kleinschreibung und alle Satzzeichen/Formatierungen entfernen). Dann wird aus dem gegebenen Satz aus der Suchdatei ein Filter erstellt, mit dem im Text nach der gewünschten Textstelle gesucht werden kann.

Umsetzung

Im ersten Schritt wird mithilfe eines Userinputs die Datei eingelesen. Falls die Datei existiert, fährt das Programm fort, falls nicht, frägt das Programm erneut nach einer Datei.
Danach wird der in der Datei enthaltene Satz so formatiert, dass jedes Wort was gegeben oder gesucht ist in derselben Reihenfolge wie im Satz in einer Liste steht.

Mit Hilfe dieser Liste wird ein Filter für eine Regex such erstellt. Dazu wird für jedes Wort im Satz überprüft, ob das Wort gegeben ist oder nicht. Ist es gegeben, fügt das Programm das gesuchte Wort an dieser Stelle in den Filter ein. Handelt es sich an dieser Stelle um ein gesuchtes Wort, fügt das Programm an dieser Stelle eine Leerstelle ein, mit der das gesuchte Wort gefunden werden kann.
Nach der Fertigstellung des Filters wird mit Regex der Text nach der gewünschten Textstelle durchsucht und die gefundenen Stellen werden in einer Liste gespeichert.
Als Letztes werden die gefundenen Textstellen in der Konsole als Ergebnis ausgegeben.

Beispiele

Dies sind die Ausgaben, die das Programm in der Konsole macht, nachdem das Programm den Namen und Speicherort der Datei erhalten hat.

stoerung0.txt

['das kommt mir gar nicht richtig vor']

stoerung1.txt

['ich muß in clara verwandelt', 'ich muß doch clara sein']

stoerung2.txt

['fressen katzen gern spatzen', 'fressen katzen gern spatzen', 'fressen spatzen gern katzen']

stoerung3.txt

['das spiel fing an', 'das publikum fing an']

stoerung4.txt

['ein sehr schöner tag']

stoerung5.txt

['wollen sie so gut sein']

Wichtige Teile des Codes

```
def convert_search(searching):
    s = searching.split(" ")
    result = ""
    # for every word in search request
    for i in range(len(s)):
        # if this is a searched word
        if s[i] == "_":
            # add empty to pattern
            # if its the first word in search request
            if i == 0:
                # add without space
                result += "[^\s]+"
            else:
                # add with space
                result += "\s[^\s]+"
        # if the word is given
        else:
            # add word to pattern
            # if its the first word in search request
            if i == 0:
                # add without space
                result += s[i]
            else:
                # add with space
                result += "\s" + s[i]

    return result
```

Erstellt aus dem gegebenen Lückensatz den Suchfilter für Regex

```
# get story and make it lower case
text = open(filename_alice, "r", encoding="utf8").read().lower().replace("\n", " ")
# keep only necessary (no punctuation etc.)
# pattern only alphabetic characters and spaces
substract_pattern = re.compile('[\W_]+' + '|' + '\s+')
text = substract_pattern.sub(' ', text)
```

Formatiert den Text “Alice im Wunderland“ so, dass er für das Programm verwendbar ist.