Aufgabe 1: Wörter aufräumen

Team-ID: 01011

Team: Jim Maar

Bearbeiter/-innen dieser Aufgabe:
Jim Maar

21. November 2020

Inhaltsverzeichnis

Lösungsidee

Die Wörter des Lückentextes werden im folgenden Lücken genannt. Um den vervollständigten Text zu bekommen, muss geschaut werden, welches Wort in welche Lücke einsetzbar ist, wobei ein Wort möglicherweise in mehrere Lücken einsetzbar sein kann. Wenn ein Wort in eine Lücke einsetzbar ist, dann bilden dieses Wort und die Lücke ein potenzielles Paar. Wenn ein Wort nur ein potenzielles Paar hat, dann ist dieses nun ein eindeutiges Paar, das heißt, das Wort kann in die Lücke eingesetzt werden.

Da eine eindeutige Lösung vorausgesetzt wird, muss es mindestens ein eindeutiges Paar geben. Wenn dieses Paar entfernt wird, muss es immer noch ein eindeutiges Paar geben. Wenn wir alle Wörter mit jeweils allen Lücken prüfen, ob diese ein eindeutiges Paar bilden, diese jeweils herausnehmen und das Verfahren wiederholen, haben wir irgendwann alle Paare gefunden. Das ist zwar eine Lösung, ist jedoch ineffizient, da der Algorithmus mehrmals die gleichen Paare überprüft.

Mit jedem gelöschten eindeutigen Paar kann ein potenzielles Paar zu einem eindeutigen Paar werden. Wenn das gelöschte Paar zu diesem Zeitpunkt das einzige eindeutige Paar war, passiert das auf jeden Fall. Ein potenzielles Paar wird dann zu einem eindeutigen Paar, wenn ein zweites potenzielles Paar von dem selben Wort gelöscht wird und das Wort nur in zwei Paaren enthalten ist.

Das Programm speichert nacheinander jedes potenzielle Paar von jedem Wort und jeder Lücke. Sobald ein eindeutiges Paar gefunden wurde, wird das Wort dieses Paares an der richtigen Stelle in den Lückentext eingesetzt und alle potenziellen Paare von der Lücke des Paares werden gelöscht. Wenn dadurch ein potenzielles Paar zu einem eindeutigen Paar wird, wird der Vorgang gleich wiederholt. Anschließend werden die Lücken aus den eindeutigen Paaren gelöscht, damit kein

potenzielles Paar mehr in Verbindung mit diesen entstehen kann. Wenn alle potenziellen Paare durchgegangen wurden, wurden alle eindeutigen Paare gefunden und damit alle Wörter eingesetzt.

Ein Wort kann auch mehrmals vorkommen. Solche Wörter bilden ein eindeutiges Paar mit so vielen Lücken wie die Anzahl des Wortes, bzw. wenn die Lücken auch mehrmals vorkommen, mit so vielen, dass die Summe der Anzahlen der Lücken der Anzahl des Wortes entspricht.

Umsetzung

Die Lösungsidee wird in Python implementiert. Die Wörter und Lücken werden in Dictionaries dargestellt, mit einer Liste bestehend aus den potenziellen Paaren jeder Lücke bzw. jedes Wortes jeweils als Value. Es werden außerdem Dictionaries zur Speicherung folgender Dinge genutzt:

- Die Anzahl von Wörtern und Lücken
- Eine vereinfachte Form von Lücken zur Überprüfung auf Einsetzbarkeit
- Die Position von Lücken im Satz

Um die Wörter in den Lösungssatz einzusetzen, wird eine Liste mit so vielen Elementen, wie es Wörter gibt, die auch alle Satzzeichen enthält, erstellt.

Die Funktion “aufräumen” findet alle Paare. Die Funktion “Einsetzen” setzt Wörter aus potenziellen Paaren in den Lückentext ein, dabei wird die eben genannte Liste benutzt. Die Funktion “löschen” löscht Lücken aus eindeutigen Paaren und findet dabei neue eindeutige Paare.

Das Moduls sys wird benutzt, damit das Programm mit der einzulesenden Datei oder einem Link einer in den Materialien zu findenden Beispiel Text Datei als Parameter auf der Kommandozeile aufgerufen werden kann.

Um die Links als Parameter benutzen zu können, wird das Modul urllib.request benutzt.
In dem Programm werden zwei Dictionaries mit einem Pipe kombiniert. Diese Funktion wurde in der neusten Python Version 3.9 hinzugefügt. Das Programm funktioniert also nur mit der Python Version 3.9.

Beispiele

Wir rufen das Programm mit verschiedenen bwinf-Eingabedaten auf. Diese werden aus den in den Materialien zu findenden URLs importiert.

$ ./Wörteraufräumen.py https://bwinf.de/fileadmin/bundeswettbewerb/39/raetsel0.txt

oh je, was für eine arbeit!

$ ./Wörteraufräumen.py https://bwinf.de/fileadmin/bundeswettbewerb/39/raetsel1.txt

Am Anfang wurde das Universum erschaffen. Das machte viele Leute sehr wütend und wurde allenthalben als Schritt in die falsche Richtung angesehen.

$ ./Wörteraufräumen.py https://bwinf.de/fileadmin/bundeswettbewerb/39/raetsel2.txt

Als Gregor Samsa eines Morgens aus unruhigen Träumen erwachte, fand er sich in seinem Bett zu einem ungeheueren Ungeziefer verwandelt.

$ ./Wörteraufräumen.py https://bwinf.de/fileadmin/bundeswettbewerb/39/raetsel3.txt

Informatik ist die Wissenschaft von der systematischen Darstellung, Speicherung, Verarbeitung und Übertragung von Informationen, besonders der automatischen Verarbeitung mit Digitalrechnern.

$ ./Wörteraufräumen.py https://bwinf.de/fileadmin/bundeswettbewerb/39/raetsel4.txt

Opa Jürgen blättert in einer Zeitschrift aus der Apotheke und findet ein Rätsel. Es ist eine Liste von Wörtern gegeben, die in die richtige Reihenfolge gebracht werden sollen, so dass sie eine lustige Geschichte ergeben. Leerzeichen und Satzzeichen sowie einige Buchstaben sind schon vorgegeben.

Quellcode

```
# In Satz werden später alle Wörter eingesetzt
Satz = []
# Lückenindex enthält Indizes von Lücken
Lückenindex = {}
# Lückeneinfach enthält eine vereinfachte Form der Lücken
Lückeneinfach = {}
for Index, Lücke in enumerate(Lücken):
    # Die Satzzeichen werden rausgenommen und in Satz gespeichert
    if Lücke[-1] == "_" or Lücke[-1].isalpha():
        Satz += [""]
    else:
        Satz += [Lücke[-1]]
        Lücke = Lücke[:-1]
        Lücken[Index] = Lücke
    # Indizes der Lücken werden zu Lückenindex hinzugefügt
    if Lücke in Lückenindex:
        Lückenindex[Lücke] += [Index]
    else:
        Lückenindex[Lücke] = [Index]
    # Lückeneinfach enthält die Buchstaben mit ihren Indizes der Lücken
    Lückeeinfach = []
    for Index in range(0, len(Lücke)):
        if Lücke[Index] != "_":
            Lückeeinfach += [[Index, Lücke[Index]]]
    Lückeneinfach[Lücke] = Lückeeinfach

# Anzahl enthält die Anzahl jedes Wortes und jeder Lücke
Anzahl = {Wort: Wörter.count(Wort) for Wort in Wörter} | {
```

```
    Lücke: Lücken.count(Lücke) for Lücke in Lücken
}
# Wörter und Lücken enthalten nun auch die potenziellen Paare der Wörter und Lücken
Wörter = {Wort: [] for Wort in Wörter}
Lücken = {Lücke: [] for Lücke in Lücken}


def einsetzbar(Wort, Lücke):
    # Sie sind nicht einsetzbar, wenn die Längen unterschiedlich sind
    if len(Wort) != len(Lücke):
        return False
    Lücke = Lückeneinfach[Lücke]
    for Index, Buchstabe in Lücke:
        # Sie sind nicht einsetzbar, wenn ein Buchstabe nicht übereinstimmt
        if Wort[Index] != Buchstabe:
            return False
    # Sie sind einsetzbar, wenn noch nichts zurückgegeben wurde
    return True


def eintragen(Wort):
    # Die Lücke ist ein eindeutiges Paar von dem Wort
    for Lücke in Wörter[Wort]:
        # Das Wort wird in den Endsatz eingesetzt
        for i in range(0, Anzahl[Lücke]):
            Index = Lückenindex[Lücke][i]
            Satz[Index] = Wort + Satz[Index]
        # Die Lücke wird aus dem Dictionary entfernt
        del Lücken[Lücke]


def löschen(Wort):
    # Die Lücke bildet ein eindeutiges Paar mit dem Wort
    for Lücke in Wörter[Wort]:
        # falschesWort bildete ein potenzielles Paar mit der Lücke
        for falschesWort in Lücken[Lücke]:
            if falschesWort != Wort:
                # Die Lücke wird aus den potenziellen Paaren von falschesWort entfernt
                Wörter[falschesWort].remove(Lücke)
                if (
                    sum([Anzahl[Lücke] for Lücke in Wörter[falschesWort]])
                    == Anzahl[falschesWort]
                ):
                    # Wenn die potenziellen Paare von falschesWort jetzt eindeutige Paare sind,
                    # wird die Funktion noch mal mit diesen Paaren/Paar ausgeführt
                    # und falschesWort wird in die Lücken eingesetzt
                    löschen(falschesWort)
```

```
                eintragen(falschesWort)


def aufräumen():
    for Wort in Wörter:
        for Lücke in Lücken:
            # Wenn ein Wort in eine Lücke einsetzbar ist, wird das gespeichert
            if einsetzbar(Wort, Lücke):
                Wörter[Wort] += [Lücke]
                Lücken[Lücke] += [Wort]
        if sum([Anzahl[Lücke] for Lücke in Wörter[Wort]]) == Anzahl[Wort]:
            # Wenn die potenziellen Paare vom Wort eindeutig sind,
            # wird die Löschfunktion aufgerufen und das Wort wird in die Lücken eingesetzt
            löschen(Wort)
            eintragen(Wort)


aufräumen()
# Ergebnisse werden ausgegeben
Satz = " ".join(Satz)
print(Satz)
```