Aufgabe 2: Twist

Anton Schütz

18. November 2018

Inhaltsverzeichnis

1 Lösungsidee

Teilaufgabe twisten

Nach dem Twisten sind der Anfangs- und Endbuchstabe eines Wortes unverändert, das heißt, um ein Wort zu twisten muss man lediglich die Reihenfolge der Buchstaben in der Mitte des Wortes vertauschen. Dies funktioniert offenkundig nur bei Wörtern, die mehr als 3 Buchstaben haben, sowie der Mittelteil nicht nur aus gleichen Buchstaben besteht.

Teilaufgabe enttwisten

Beim Enttwisten macht man sich die Eigenschaft zunutze, dass beim Twisten Anfangs- und Endbuchstabe unverändert bleiben. Auch ändert sich die Länge des Wortes und dessen Zeichenzusammensetzung nicht. Anhand dieser Parameter kann versucht werden, ein passendes Wort in einem Wörterbuch zu finden. An Grenzen stößt das Verfahren, wenn es für ein getwistetes Wort mehrere mögliche Ausgangswörter besitzt. Hier kann nicht eindeutig gesagt werden, welches das korrekte Ausgangswort ist. Auch ist der Erfolg ausschlaggebend von dem genutzten Wörterbuch. Neologismen können mit einem Standartwörterbuch zum Beispiel nicht gefunden werden. Auch Eigennamen können ein Problem darstellen.

2 Umsetzung

Teilaufgabe twisten

Das zu twistende Wort wird in 3 Teile aufgeteilt:
den ersten Buchstaben, den Mittelteil und den letzten Buchstaben. Es wird überprüft, ob das Wort länger als 3 Buchstaben ist und der Mittelteil aus verschiedenen Buchstaben besteht. Ist dies nicht gegeben, kann das Wort nicht getwistet werden. Der Mittelteil wird nun in eine Liste zerteilt, wobei jeder Buchstabe ein Element der Liste ist. Diese wird nun mittels der Erweiterung 'random' gemischt. Aus dem Anfangsbuchstaben, dem gemischten Mittelteil und dem letzten Buchstaben wird nun wieder ein Wort zusammengesetzt. Zum Schluss wird überprüft, ob das neue Wort und das Ausgangswort ungleich ist. Sie könnten theoretisch gleich sein, da das Mischen der Liste die Möglichkeit zulässt, dass diese unverändert bleibt.

Teilaufgabe enttwisten

Anhand des Anfangsbuchstaben des zu enttwistenden Wortes werden aus der Wörterliste alle Wörter mit gleichem Anfangsbuchstaben in einer Liste gespeichert. Aus dieser werden nun alle Wörter mit einer anderen Länge als der des Ausgangswortes, sowie alle mit ungleichem letzten Buchstaben aussortiert. Bei den übrig geblieben Wörtern wird geschaut, ob sie mit der Zeichenzusammensetzung, also Anzahl und Art der Buchstaben, übereinstimmen. Trifft dies zu ist das Wort wahrscheinlich das enttwistete Ausgangswort.

3 Beispiele

Teilaufgabe twisten

Ungetwistet:
Augusta Ada Byron King, Countess of Lovelace, war eine britische Adelige und Mathematikerin,
die als die erste Programmiererin überhaupt gilt.
Bereits 100 Jahre vor dem Aufkommen der ersten Programmiersprachen ersann sie eine
Rechen-Mechanik,
der einige Konzepte moderner Programmiersprachen vorwegnahm.

Getwistet:
Augtusa Ada Boryn Knig, Cteunoss of Llocveae, war enie bhtiirsce Aigldee und Merimetiktahan,
die als die ertse Preiimraoergrmn üraehbput glit.
Bierets 100 Jarhe vor dem Aemuomkfn der etersn Prcmmpsaearrrgihoen esrann sie enie
Rehecn-Micenhak,

der egiine Koznepte moeednrr Paprrecrsiomgahermn vrhonaewgm.

Ausschließlich die Wörter werden getwistet. Sowohl die Formatierung, als auch die Sonderzeichen, wie Kommas etc. bleiben unverändert. Auch wurden die nicht twistbaren Wörter übernommen, wie zum Beispiel 'war' oder 'vor'. Man erkennt auch, dass das Verfahren Eigennamen twistet.

Teilaufgabe enttwisten

Getwistet:
Der Twsit
(Eigsnclh tiwst = Duenrhg, Venurdrehg)
war ein Mdaotenz im 4/4-Tkat,
der in den frhüen 1960er Jearhn populär
wrude und zu
Rcok'n'Roll, Ryhthm and Bleus oedr sielezpler
Twsit-Msuik gnzteat wrid.

Enttwistet:
Der Twsit
(Eigsnclh tiwst = Drehung, Verdrehung)
war ein Mdaotenz im 4/4-Takt,
der in den frühen 1960er Jahren populär
wurde und zu
Rock'n'Roll, Ryhthm and Blues oder spezieller
Twsit-Musik getanzt wird.

Sowohl Formatierung, als auch Sonderzeichen bleiben auch hier unverändert. Viele Wörter konnten enttwistet werden, wie zum Beispiel 'Drehung' oder 'Takt'. Es gibt aber Wörter, zu den kein passendes Ausgangswort gefunden wurde, da dieses in der genutzten Wörterliste nicht vorhanden war.

4 Quellcode (ausschnittsweise)

Teilaufgabe twisten

```
def shuffeler(word):
    # Wort kann nicht getwistet werden, da es zu kurz ist
    if len(word) <= 3:
        return word

    # Wort kann nicht getwistet werden, da nur gleiche Zeichen in der
    # Wortmitte vorhanden sind
    worker = list(word[1:len(word) - 1])
    last_c = worker[0]
    worker.pop(0)

    res = False
    for c in worker:
        if last_c != c:
```

```
            res = True
        last_c = c
    if not res:
        return word

    out = word
    # Wort muss getwistet sein (Bei z.B. symmetrischem Mittelteil notwendig)
    while out == word:
        mid = list(word[1:len(word) - 1])
        random.shuffle(mid)
        out = word[0] + ''.join(mid) + word[len(word) - 1]
    return out
```

Die Funktion **shuffeler** hat die Aufgabe, zu überprüfen, ob ein Wort twistbar ist und wenn ja, dieses zu twisten.

Teilaufgabe enttwisten

```
def enttwister(word):
    # Wort kann nicht getwistet sein
    if len(word) < 4:
        return word

    # moegliche Woerter suchen
    char_ind = char_indice[word[0]]
    poswords_raw = wordlist[char_ind:]  # Woerter mit gleichem
    # Anfangsbuchstaben
    poswords = []  # Woerter mit geichem Anfangs- und Endzeichen

    for w in poswords_raw:
        if word[0] != w[0]:
            break
        if word[-1] == w[-1]:
            if len(word) == len(w):
                poswords.append(w)

    # Keine moeglichen Woerter gefunden
    if (len(poswords)) == 0:
        return word

    # moegliche Woerter durchgehen und auf Richtigkeit ueberpruefen
    for e in poswords:
        el = list(e)
        fit = True
```

```
        for c in word:
            if c in el:
                el.remove(c)
            else:
                # Wort kann nicht passen
                fit = False
                break
        if fit and len(el) == 0:
            # Passendes Wort gefunden
            return e
    # Kein moegliches Wort hat gepasst
    return word
```

Die Funktion **enttwister** hat die Aufgabe, aus einer Liste der möglichen Wörter passende heraus zu suchen und diese dann auf ihre Richtigkeit zu überprüfen