

# Aufgabe 2: Twist

Team: Richard Wohlbold

Team-ID: 00012

25. Oktober 2018

## Inhaltsverzeichnis

<b>1</b>	<b>Lösungsidee</b>	<b>2</b>
1.1	Wörtererkennung . . . . .	2
1.2	Twisten . . . . .	2
1.3	Enttwisten . . . . .	2
1.3.1	Die Wörterliste im Speicher . . . . .	2
1.3.2	Bestimmung der enttwisteten Wörter . . . . .	3
1.3.3	Optimierung . . . . .	3
<b>2</b>	<b>Umsetzung</b>	<b>3</b>
<b>3</b>	<b>Beispiele</b>	<b>3</b>
3.1	Twisten . . . . .	3
3.1.1	Triviales Beispiel: Keine Eingabedatei . . . . .	3
3.1.2	Triviales Beispiel: Kein Text . . . . .	4
3.1.3	Beispiel 1 von der bwinf-Website . . . . .	4
3.1.4	Beispiel 2 von der bwinf-Website . . . . .	4
3.1.5	Beispiel 3 von der bwinf-Website . . . . .	4
3.1.6	Beispiel 4 von der bwinf-Website . . . . .	4
3.1.7	Beispiel 5 von der bwinf-Website . . . . .	4
3.2	Enttwisten . . . . .	4
3.2.1	Triviales Beispiel: Keine Eingabedatei . . . . .	5
3.2.2	Triviales Beispiel: Kein Text . . . . .	5
3.2.3	Triviales Beispiel: Keine Wörterliste . . . . .	5
3.2.4	Beispiel von der bwinf-Website . . . . .	5
3.2.5	Ausgabe von 3.1.4 . . . . .	5
3.2.6	Ausgabe von 3.1.5 . . . . .	6
3.2.7	Ausgabe von 3.1.6 . . . . .	6
3.2.8	Ausgabe von 3.1.7 . . . . .	6
3.3	Beides . . . . .	6
3.3.1	Triviales Beispiel: Keine Eingabedatei . . . . .	6
3.3.2	Triviales Beispiel: Kein Text . . . . .	6
3.3.3	Triviales Beispiel: Keine Wörterliste . . . . .	7
3.3.4	Beispiel 1 von der bwinf-Website . . . . .	7
3.3.5	Beispiel 2 von der bwinf-Website . . . . .	7
3.3.6	Beispiel 3 von der bwinf-Website . . . . .	7
3.3.7	Beispiel 4 von der bwinf-Website . . . . .	8
3.3.8	Beispiel 5 von der bwinf-Website . . . . .	8
<b>4</b>	<b>Quellcode (ausschnittsweise)</b>	<b>8</b>

# 1 Lösungsidee

Um einen Text zu twisten, müssen zuerst die Wörter von anderen Zeichen, wie Kommas, Klammern und Ähnlichem getrennt werden, denn wir möchten nicht, dass solche Symbole in der Mitte eines Wortes stehen. Ebenfalls müssen beim Enttwisten eines Texts zuerst die getwisteten Wörter von Kommas und anderen Satzzeichen getrennt werden, um diese in einer Wörterliste zu finden.

## 1.1 Wörtererkennung

**Hinweis:** Im Folgenden verwende ich die Schreibweise regulärer Ausdrücke, wie sie in Python mit dem `re`-Modul verwendet wird. Die Konzepte sind jedoch auch auf andere Schreibweisen übertragbar.

Um alle deutschen Wörter in einem Text zu erkennen, benutze ich reguläre Ausdrücke. Ein deutsches Wort ist demnach eine Folge von einem oder mehr deutschen Buchstaben ( $a$ - $z$ ,  $A$ - $Z$ ,  $\text{Ä}$ ,  $\text{ä}$ ,  $\text{Ö}$ ,  $\text{ö}$ ,  $\text{Ü}$ ,  $\text{ü}$  und  $\beta$ ). Um damit einen deutschen Buchstaben zu erkennen, erstelle ich eine neue Buchstabenklasse, indem ich den regulären Ausdruck `[a-zA-ZÄäÖöÜü]` verwende. Um nun einen oder mehr als einen solchen Buchstaben hintereinander zu erfassen, füge ich dem regulären Ausdruck noch ein `+` hinzu, das *eins oder mehr* bedeutet: `[a-zA-ZÄäÖöÜü]+`.

Mit diesem regulären Ausdruck kann man nun alle deutschen Wörter aus dem Text extrahieren. Die Zeichen zwischen zwei Wörtern werden weder getwistet noch enttwistet, sondern werden immer so gelassen wie sie sind. Die Benutzung eines regulären Ausdrucks hat den Vorteil, dass nicht spezifiziert wird, welche Zeichen Wörter trennen, sondern welche Zeichen Wörter ausmachen. Da es sehr viele Satzzeichen geben könnte, ist dieses Verfahren einfacher, solange die Aufgabe spezifiziert, welches Alphabet verwendet wird (hier ist es das Deutsche).

## 1.2 Twisten

Da wir nun alle Wörter in einem Text erkennen können, können wir alle Wörter getrennt betrachten. Das Twisten eines Worts geschieht in drei Schritten:

1. Falls das Wort nur einen oder zwei Buchstaben lang ist, bleibt das Wort so wie es ist und wir sind fertig. Ansonsten wird der erste und letzte Buchstabe entfernt.
2. Die verbleibenden Buchstaben werden zufällig angeordnet.
3. Der erste und letzte Buchstabe werden wieder jeweils an den Anfang oder an das Ende hinzugefügt.

Mit diesem Verfahren lässt sich jedes in 1.1 ermittelte Wort zufällig twisten.

## 1.3 Enttwisten

### 1.3.1 Die Wörterliste im Speicher

Das Enttwisten ist herausfordernder als das Twisten, da das Programm ohne eine Wörterliste (oder Machine Learning) zurechtkommen kann, da es nicht weiß, welche Wörter überhaupt möglich sind. Als Vorbereitung zum Enttwisten muss eine Wörterliste eingelesen und vorbereitet werden. Damit das eigentliche Verfahren zum Enttwisten mit einer Geschwindigkeit von  $O(n)$  ( $n$  ist die Anzahl der Wörter im Text) und unabhängig von der Länge der Wörterliste laufen kann, muss eine Datenstruktur verwendet werden, die gegeben einer Eingabe eines getwisteten Worts eine Liste von möglichen Wörtern in  $O(1)$  (konstanter Zeit) zuordnet. Eine geeignete Datenstruktur hierfür ist die Hashtabelle, da diese eine Zugriffszeit von  $O(1)$  bietet, die ungenutzten Felder aufgrund geringer Datenmenge kein großes Problem darstellen und da diese nach initialer Befüllung nur noch zum Lesen benutzt wird.

Da jedes getwistete Wort eine beliebige Reihenfolge der mittleren Buchstaben innehaben kann, muss ein Verfahren bestehen, jede getwistete Version eines Originalworts in die gleiche Zeichenfolge umzuwandeln, ohne das Originalwort zu kennen. Währenddessen muss jedoch auch so viel Information wie möglich behalten werden (vor Allem der erste und letzte Buchstabe). Dies ist möglich, indem die mittleren Buchstaben eines getwisteten Worts alphabetisch sortiert werden, der erste und letzte jedoch gleich bleiben. Somit ist immernoch klar, was der erste und letzte Buchstabe waren und welche anderen Buchstaben in dem Wort enthalten sind. Dieses Wortform nenne ich **reduzierte Form**.

Dieses Verfahren eignet sich als Schlüssel für die Hashtabelle. Aus jedem getwisteten Wort kann diese Form abgeleitet werden und eine Liste von möglichen enttwisteten Wörtern mit gleicher reduzierter Form kann in  $O(1)$  gefunden werden.

Somit erstellen wir aus der Wörterliste eine Hashtabelle: Für jedes Wort wird die reduzierte Form als Schlüssel verwendet und der Liste von Wörtern mit diesem Schlüssel wird das richtige Wort angehängt.

### 1.3.2 Bestimmung der enttwisteten Wörter

Wenn ein getwistetes Wort gegeben ist, muss dieses Wort nun die reduzierte Form des Wortes gebildet werden, die der Hashtabelle als Schlüssel gegeben wird. Nun gibt es drei Möglichkeiten:

1. Es gibt kein Wort für die reduzierte Form in der Hashtabelle. Hier wird das getwistete Wort so belassen, wie es ist. Je nach Länge der Wörterliste tritt dieses Problem unterschiedlich häufig auf. Außerdem wird eine Warnung ausgegeben.
2. Es gibt genau ein Wort für die reduzierte Form in der Hashtabelle. Dies ist der Idealfall, das Wort wird einfach in den Ergebnistext übernommen.
3. Es gibt mehrere Wörter für die reduzierte Form. Wir können irgendeins der Wörter nehmen, da wird nicht wissen können, welches hier gemeint ist. Die anderen Möglichkeiten sollten mit einer Warnung ausgegeben werden.

### 1.3.3 Optimierung

Wenn ein Wort am Satzanfang steht, wird es immer großgeschrieben, nach manchen Satzzeichen, wie einem Doppelpunkt manchmal auch. Verben werden nach einem Artikel auch großgeschrieben (*das Schwimmen*). Damit das Verfahren nicht alle diese Sonderfälle einzeln betrachtet, werden alle Schlüssel der Hashtabelle kleingeschrieben. Damit der Ausgabertext trotzdem zu einem Großteil richtig geschrieben ist, wird die Groß- und Kleinschreibung der Wörter in der Wörterliste in den Ergebnislisten der Hashtabelle übernommen. Außerdem wird ein Wort im Ausgabertext großgeschrieben, wenn das getwistete Wort im Eingabetext großgeschrieben ist, wie beispielsweise am Satzanfang.

## 2 Umsetzung

Ich habe das Verfahren in Python umgesetzt. Zur Erkennung von Wörtern mithilfe regulärer Ausdrücke benutze ich das integrierte Modul `re`. Um die Wörter von Satzzeichen zu trennen, benutze ich die Funktion `split`, die alle Treffer des Musters von allen anderen Zeichen trennt und in einer Liste zurückgibt. Zur zufälligen Anordnung mittlerer Buchstaben beim twisten benutze ich die Funktion `shuffle` aus dem Modul `random`. Als Hashtabelle verwende ich die standardmäßigen Python-dicts.

Um mit einem Skript zu twisten, enttwisten und beides nacheinander zu tun, parse ich mit `argparse` die Argumente. Das Skript wird mit `twist.py <aktion> <eingabedatei> [-w woerterliste]` aufgerufen, wobei Aktion entweder `twist`, `enttwist` oder `beides` sein kann, wobei `beides` den Eingabetext twistet und wieder enttwistet, um z.B. die Qualität einer Wörterliste zu testen. Eine Wörterliste wird nur bei `enttwist` oder `beides` benötigt.

## 3 Beispiele

Die Beispiele zur Twistfunktion des Programms sind von der bwinf-Website.

### 3.1 Twisten

#### 3.1.1 Triviales Beispiel: Keine Eingabedatei

```

1 $ python3 twist.py twist
  usage: twist.py [-h] [-w WOERTER] {twist,enttwist,beides} eingabe
3 twist.py: error: the following arguments are required: eingabe

```

### 3.1.2 Triviales Beispiel: Kein Text

Hier ist `leer.txt` eine leere Textdatei. Die Ausgabe sollte deshalb auch leer sein:

```
1 $ python3 twist.py twist ../beispieldaten/leer.txt
```

### 3.1.3 Beispiel 1 von der bwinf-Website

```
$ python3 twist.py twist ../beispieldaten/twist1.txt
2 Der Twist
  (Eiscnlgh tiswt = Dhureng, Vreuedhnrng)
4 war ein Moedtnaz im 4/4-Tkat,
  der in den fheurn 1960er Jaerhn päolupr
6 wrdue und zu
  Rcok'n'Rlol, Rhtyhm and Beuls oder sieezlpler
8 Tsiwt-Misuk gazentt wrid.
```

### 3.1.4 Beispiel 2 von der bwinf-Website

```
$ python3 twist.py twist ../beispieldaten/twist2.txt
2 Hat der alte Hmeixeenster scih doch eniaml wegebeegbn! Und nun soelln sneie Getsier auch
  nach meniem Wellin leebn. Sinee Wort und Werke mkret ich und den Bruach, und mit
  Gietsästksere tu ich Wuednr acuh.
```

### 3.1.5 Beispiel 3 von der bwinf-Website

```
$ python3 twist.py twist ../beispieldaten/twist3.txt
2 Ein Rrauseatnt, wehecls a la crate aebirett, beeitt sien Agnboet ohne eine vhrefor
  fleeggeste Mrfnneoehiülgee an. Dducarh heabn die Gätse zawr mher Sielpaurm bei der
  Whal ierhr Seepisn, für das Rnrauætst eeteshnn jeodch zueähczstlr Afnuwad, da
  wengeir Panrisgnhlhucsieet vohneadrn ist.
```

### 3.1.6 Beispiel 4 von der bwinf-Website

```
$ python3 twist.py twist ../beispieldaten/twist4.txt
2 Autusga Ada Bryon Knig, Ctnouess of Levlaoce, war enie birsthice Adliege und
  Mitetiakmerahn, die als die etsre Pragrioemriermn üpubaerht glit. Brtiees 100 Jahre
  vor dem Akuomefmn der etsren Paerprmrgeahomsrin enarsn sie enie Rcehen-Mhnaceik, der
  egniie Kzponete mneeodr Paprhoariemcrmregsn vgowanhrem.
```

### 3.1.7 Beispiel 5 von der bwinf-Website

Hinweis: Im folgenden Beispiel erscheint die Programmausgabe nach weniger als einer Sekunde, nur L<sup>A</sup>T<sub>E</sub>Xweigert sich, sie zu drucken.

```
$ python3 twist.py twist ../beispieldaten/twist5.txt
2 [...]
```

## 3.2 Enttwisten

Das erste Beispiel zum Enttwisten ist von der bwinf-Website, für alle Anderen wurde die Programmausgabe der Beispiele von 3.1 verwendet.

### 3.2.1 Triviales Beispiel: Keine Eingabedatei

```
$ python3 twist.py enttwist -w ../beispieldaten/woerterliste.txt
2 usage: twist.py [-h] [-w WOERTER] {twist,enttwist,beides} eingabe
twist.py: error: the following arguments are required: eingabe
```

### 3.2.2 Triviales Beispiel: Kein Text

```
1 $ python3 twist.py enttwist ../beispieldaten/leer.txt -w ../beispieldaten/woerterliste.
txt
```

### 3.2.3 Triviales Beispiel: Keine Wörterliste

```
1 $ python3 twist.py beides ../beispieldaten/enttwist.txt
Fehler: Keine Wörterliste angegeben!
```

### 3.2.4 Beispiel von der bwinf-Website

```
$ python3 twist.py enttwist ../beispieldaten/enttwist.txt -w ../beispieldaten/
woerterliste.txt
2 Kein enttwistetes Wort für "Twsit" gefunden...
Kein enttwistetes Wort für "tiwst" gefunden...
4 Mehrere enttwistete Wörter für "ein" gefunden: ['ein', 'ein'] , benutze "ein" ...
Kein enttwistetes Wort für "Mdaotenz" gefunden...
6 Mehrere enttwistete Wörter für "frhüen" gefunden: ['frühen', 'führen'] , benutze "frühen
" ...
Kein enttwistetes Wort für "n" gefunden...
8 Kein enttwistetes Wort für "Roll" gefunden...
Kein enttwistetes Wort für "Ryhythm" gefunden...
10 Kein enttwistetes Wort für "and" gefunden...
Kein enttwistetes Wort für "Twsit" gefunden...
12
Der Twsit
14 (Englisch tiwst = Drehung, Verdrehung)
war ein Mdaotenz im 4/4-Takt,
16 Der in den frühen 1960er Jahren populär
wurde und zu
18 Rock'n'Roll, Ryhythm and Blues oder spezieller
Twsit-Musik getanzt wird.
```

### 3.2.5 Ausgabe von 3.1.4

```
1 $ python3 twist.py enttwist ../beispieldaten/ausgabe_von_2.txt -w ../beispieldaten/
woerterliste.txt
Kein enttwistetes Wort für "wegebeegbn" gefunden...
3 Kein enttwistetes Wort für "tu" gefunden...

5 Hat der alte Hexenmeister sich doch einmal wegebeegbn! Und nun sollen seine Geister auch
nach meinem Willen leben. Seine Wort Und Werke merkt ich Und den Brauch, Und mit
Geistesstärke tu ich Wunder auch.
```

### 3.2.6 Ausgabe von 3.1.5

```

1 $ python3 twist.py enttwist ../beispieldaten/ausgabe_von_3.txt -w ../beispieldaten/
  woerterliste.txt
  Kein enttwistetes Wort für "a" gefunden...
3 Kein enttwistetes Wort für "la" gefunden...
  Kein enttwistetes Wort für "crate" gefunden...
5 Mehrere enttwistete Wörter für "aebirett" gefunden: ['arbeitet', 'abrietet'] , benutze "
  arbeitet" ...
  Kein enttwistetes Wort für "Mrfnneoehiülgee" gefunden...
7 Kein enttwistetes Wort für "Panrisgnhlhucsieet" gefunden...

9 Ein Restaurant, welches a la crate arbeitet, bietet sein Angebot ohne eine vorher
  festgelegte Mrfnneoehiülgee an. Dadurch haben die Gäste zwar mehr Spielraum bei der
  Wahl ihrer Speisen, für das Restaurant entstehen jedoch zusätzlicher Aufwand, da
  weniger Panrisgnhlhucsieet vorhanden ist.

```

### 3.2.7 Ausgabe von 3.1.6

```

1 $ python3 twist.py enttwist ../beispieldaten/ausgabe_von_4.txt -w ../beispieldaten/
  woerterliste.txt
  Kein enttwistetes Wort für "Autusga" gefunden...
3 Kein enttwistetes Wort für "Bryon" gefunden...
  Kein enttwistetes Wort für "Knig" gefunden...
5 Kein enttwistetes Wort für "Ctnouess" gefunden...
  Kein enttwistetes Wort für "of" gefunden...
7 Kein enttwistetes Wort für "Levlaoce" gefunden...
  Mehrere enttwistete Wörter für "Brtiees" gefunden: ['bereits', 'Bieters', 'breites', '
  breites'] , benutze "Bereits" ...

9 Autusga Ada Bryon Knig, Ctnouess of Levlaoce, 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.

```

### 3.2.8 Ausgabe von 3.1.7

Hinweis: Im folgenden Beispiel erscheint die Programmausgabe nach weniger als einer Sekunde, nur  $\LaTeX$ weigert sich, sie zu drucken.

```

$ python3 twist.py enttwist ../beispieldaten/ausgabe_von_5.txt -w ../beispieldaten/
  woerterliste.txt
2 [...]

```

## 3.3 Beides

### 3.3.1 Triviales Beispiel: Keine Eingabedatei

```

$ python3 twist.py beides -w ../beispieldaten/woerterliste.txt
2 usage: twist.py [-h] [-w WOERTER] {twist,enttwist,beides} eingabe
  twist.py: error: the following arguments are required: eingabe

```

### 3.3.2 Triviales Beispiel: Kein Text

```

1 $ python3 twist.py beides ../beispieldaten/leer.txt -w ../beispieldaten/woerterliste.
  txt

```

### 3.3.3 Triviales Beispiel: Keine Wörterliste

```
1 $ python3 twist.py beides ../beispieldaten/twist1.txt
Fehler: Keine Wörterliste angegeben!
```

### 3.3.4 Beispiel 1 von der bwinf-Website

```
$ python3 twist.py beides ../beispieldaten/enttwist.txt -w ../beispieldaten/woerterliste.
txt
2 Kein enttwistetes Wort für "Twist" gefunden...
Kein enttwistetes Wort für "tsiwt" gefunden...
4 Kein enttwistetes Wort für "Mdetanoz" gefunden...
Mehrere enttwistete Wörter für "frehün" gefunden: ['frühen', 'führen'] , benutze "frühen"
...
6 Kein enttwistetes Wort für "n" gefunden...
Kein enttwistetes Wort für "Roll" gefunden...
8 Kein enttwistetes Wort für "Rthyhm" gefunden...
Kein enttwistetes Wort für "and" gefunden...
10 Kein enttwistetes Wort für "Twist" gefunden...

12 Der Twist
(Englisch tsiwt = Drehung, Verdrehung)
14 war ein Mdetanoz im 4/4-Takt,
Der in den frühen 1960er Jahren populär
16 wurde und zu
Rock'n'Roll, Rthyhm and Blues oder spezieller
18 Twist-Musik getanzt wird.
```

### 3.3.5 Beispiel 2 von der bwinf-Website

```
$ python3 twist.py beides ../beispieldaten/twist2.txt -w ../beispieldaten/woerterliste.
txt
2 Kein enttwistetes Wort für "weeegegbbn" gefunden...
Kein enttwistetes Wort für "tu" gefunden...
4
Hat der alte Hexenmeister sich doch einmal weeegegbbn! Und nun sollen seine Geister auch
nach meinem Willen leben. Seine Wort Und Werke merkt ich Und den Brauch, Und mit
Geistesstärke tu ich Wunder auch.
```

### 3.3.6 Beispiel 3 von der bwinf-Website

```
1 $ python3 twist.py beides ../beispieldaten/twist3.txt -w ../beispieldaten/woerterliste.
txt
Kein enttwistetes Wort für "a" gefunden...
3 Kein enttwistetes Wort für "la" gefunden...
Kein enttwistetes Wort für "ctrae" gefunden...
5 Mehrere enttwistete Wörter für "aieretbt" gefunden: ['arbeitet', 'abrietet'] , benutze "
arbeitet" ...
Kein enttwistetes Wort für "Mregoneühneilfe" gefunden...
7 Kein enttwistetes Wort für "Paelignhruhcnest" gefunden...

9 Ein Restaurant, welches a la ctrae arbeitet, bietet sein Angebot ohne eine vorher
festgelegte Mregoneühneilfe an. Dadurch haben die Gäste zwar mehr Spielraum bei der
Wahl ihrer Speisen, für das Restaurant entstehen jedoch zusätzlicher Aufwand, da
weniger Paelignhruhcnest vorhanden ist.
```

### 3.3.7 Beispiel 4 von der bwinf-Website

```

1 $ python3 twist.py beides ../beispieldaten/twist4.txt -w ../beispieldaten/woerterliste.
  txt
  Kein enttwistetes Wort für "Asugtua" gefunden...
3 Kein enttwistetes Wort für "Broyn" gefunden...
  Kein enttwistetes Wort für "King" gefunden...
5 Kein enttwistetes Wort für "Ctuosens" gefunden...
  Kein enttwistetes Wort für "of" gefunden...
7 Kein enttwistetes Wort für "Lleacove" gefunden...
  Mehrere enttwistete Wörter für "Beitres" gefunden: ['bereits', 'Bieters', 'breites', '
    breites'] , benutze "Bereits" ...
9
  Asugtua Ada Broyn King, Ctuosens of Lleacove, war eine britische Adelige und
  Mathematikerin, die als die erste Programmiererin überhaupt gilt. Bereits 100 Jahre
  vor dem Aufkommen der ersten Programmiersprachenerkennung eine Rechen-Mechanik, der
  einige Konzepte moderner Programmiersprachen vorwegnahm.

```

### 3.3.8 Beispiel 5 von der bwinf-Website

Hinweis: Im folgenden Beispiel erscheint die Programmausgabe nach weniger als einer Sekunde, nur  $\text{\LaTeX}$ weigert sich, sie zu drucken.

```

$ python3 twist.py twist ../beispieldaten/twist5.txt
2 [...]

```

## 4 Quellcode (ausschnittsweise)

Die Funktion `wort_twisten`, die ein Wort twistet:

```

def wort_twisten(wort):
2     if len(wort) <= 2:
        return wort
4     mittlere_buchstaben = list(wort[1:-1])
        shuffle(mittlere_buchstaben)
6     return wort[0] + ''.join(mittlere_buchstaben) + wort[-1]

```

code/wort\_twisten.py

Die Funktion `text_twisten`, die einen Text twistet:

```

import re
2
def text_twisten(text):
4     # Der reguläre Ausdruck zum Finden von Wörtern im Text
    muster = re.compile(r'([a-zA-ZäöüÄÖÜß]+)')
    result = ""
6     elemente = re.split(muster, text)
    for element in elemente:
8         if re.fullmatch(muster, element):
            result += wort_twisten(element)
10        else:
            result += element
12    return result

```

code/text\_twisten.py

Die Funktion `wort_zu_schluessel`, die einen Schlüssel für die Hashtabelle aus dem Wort macht:

```

1 def wort_zu_schluessel(wort):
    # Sortiere mittlere Buchstaben des Worts
3     mittlere_buchstaben = ''.join(sorted(list(wort[1:-1])))
    return wort[0].lower() + mittlere_buchstaben + wort[-1].lower()

```

code/wort\_zu\_schluessel.py



Die Funktion `enttwist_text`, die einen Text versucht zu enttwisten:

```

import re
2
def enttwist_text(text: str, woerterbuch_text: str):
4     woerterbuch = woerterbuch_erstellen(woerterbuch_text.split("\n"))
    enttwisteter_text = []
6     # Der reguläre Ausdruck zum Finden von Wörtern im Text
    muster = re.compile(r'([a-zA-ZäöüÄÖÜß]+)')
8     woerter_und_zeichen = re.split(muster, text)
    for element in woerter_und_zeichen:
        # Ist das Element ein Wort?
        if re.fullmatch(muster, element):
12             schluessel = wort_zu_schluessel(element)
            moegliche_woerter = woerterbuch.get(schluessel)
14             # Keine möglichen Wörter gefunden
            if moegliche_woerter is None:
16                 print("Kein enttwistetes Wort für", "'" + element + "'", "gefunden...")
                enttwisteter_text.append(element)
18             # Ein mögliches Wort gefunden
            elif len(moegliche_woerter) == 1:
20                 if (element[0].isupper()):
                    moegliche_woerter[0] = moegliche_woerter[0][0].upper() +
moegliche_woerter[0][1:]
22                 enttwisteter_text.append(moegliche_woerter[0])
            # Mehrere mögliche Wörter gefunden
24             else:
                benutztes_wort = moegliche_woerter[0]
26                 if (element[0].isupper()):
                    benutztes_wort = benutztes_wort[0].upper() + benutztes_wort[1:]
28
                # Warnung ausgeben, falls alle Wörter nicht gleich sind
30                 if len(set(moegliche_woerter)) != 1:
                    print("Mehrere enttwistete Wörter für", "'" + element + "'", "
gefunden:", moegliche_woerter, ", benutze", "'" + benutztes_wort + "'", "...")
32                 enttwisteter_text.append(benutztes_wort)
            # Das Element ist kein Wort
34             else:
                enttwisteter_text.append(element)
36     print()
    return ''.join(enttwisteter_text)

```

code/enttwist\_text.py