4.2 n-Gramme zählen: ein Sprachmodell von Hand trainieren

Leitidee

Ein n-Gramm-Modell lernt Sprache, indem es zählt. Das klingt einfach – und das ist es auch. In diesem Abschnitt machst du genau das, was der Computer beim **Training** eines n-Gramm-Modells tut: Du nimmst einen kurzen Text, zerlegst ihn in Wortpaare, zählst sie und berechnest daraus, mit welcher Wahrscheinlichkeit auf ein Wort ein bestimmtes anderes Wort folgt. Am Ende hast du ein kleines Sprachmodell auf Papier. Zur Kontrolle lässt du Python nachzählen.

Lernziele

Nach diesem Abschnitt kannst du

- erklären, was ein n-Gramm ist, und Unigramme, Bigramme und Trigramme unterscheiden,
- die Begriffe Textkorpus und Textkörperanalyse erläutern,
- in einem kurzen Text alle Bigramme von Hand ermitteln und zählen,
- aus den Häufigkeiten die Wahrscheinlichkeiten für Folgewörter berechnen und in einer Tabelle festhalten,
- eine Handauszählung mit einem Python-Programm überprüfen.

4.2.1 Was ist ein n-Gramm?

Nimm den Satz „der hund bellt laut“. Du kannst ihn auf verschiedene Arten in Stücke schneiden:

- in **einzelne Wörter**: „der“, „hund“, „bellt“, „laut“,
- in **Wortpaare**: „der hund“, „hund bellt“, „bellt laut“,
- in **Dreiergruppen**: „der hund bellt“, „hund bellt laut“.

Jedes dieser Stücke ist ein **n-Gramm**. Das *n* gibt an, wie viele Wörter direkt hintereinander zu einem Stück gehören. Die Stücke überlappen sich: Jedes neue Stück beginnt **ein Wort weiter rechts** als das vorige.

Name	n	Stücke aus „der hund bellt laut“	Anzahl
Unigramm	1	der · hund · bellt · laut	4
Bigramm	2	der hund · hund bellt · bellt laut	3
Trigramm	3	der hund bellt · hund bellt laut	2

Mit jedem Wort mehr pro Stück gibt es ein Stück weniger. Allgemein gilt: Ein Text mit N Wörtern hat $N-n+1$ n-Gramme. Für Bigramme ($n = 2$) sind das $N - 1$.

Merke

- Ein **n-Gramm** ist eine Folge von n aufeinanderfolgenden Wörtern (oder Zeichen) aus einem Text.
  - **Unigramm** ($n = 1$): einzelne Wörter
  - **Bigramm** ($n = 2$): Wortpaare
  - **Trigramm** ($n = 3$): Wortdreiergruppen

Statt Wörtern kann man auch **Zeichen** zu n-Grammen zusammenfassen. Die Zeichen-Bigramme von „hund“ sind „hu“, „un“ und „nd“. In diesem Kapitel geht es aber immer um Wörter.

4.2.2 Textkorpus und Textkörperanalyse

Ein n-Gramm-Modell braucht Text, aus dem es lernen kann. Diese Textsammlung heißt **Textkorpus** oder kurz **Korpus** (lateinisch *corpus* = Körper; deshalb sagt man auch **Textkörper**). Bei echten Sprachmodellen besteht der Korpus aus riesigen Mengen an Büchern, Zeitungsartikeln und Internetseiten. Bei uns sind es zunächst nur ein oder zwei Sätze. Das Prinzip ist dasselbe.

Definition: Textkorpus

Ein **Textkorpus** (auch Textkörper) ist eine Sammlung von Texten, die als **Trainingstext** für ein Sprachmodell dient.

Definition: Textkörperanalyse

Bei der **Textkörperanalyse** wird ermittelt, wie oft jedes n-Gramm in einem Textkorpus vorkommt. Aus diesen Häufigkeiten werden anschließend Wahrscheinlichkeiten berechnet.

Merke

Bei der **Textkörperanalyse** zählt man, wie oft jedes n-Gramm in einem **Textkorpus** (Trainingstext) vorkommt.

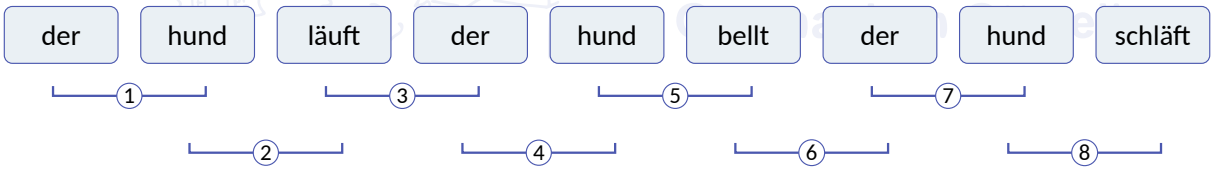
Damit das Zählen eindeutig ist, schreiben wir in diesem Kapitel alle Wörter **klein** und lassen Satzzeichen weg. Sonst wären „Der“ und „der“ für den Computer zwei verschiedene Wörter.

4.2.3 Bigramme von Hand zählen

Wir untersuchen diesen Textkorpus:

> der hund läuft der hund bellt der hund schläft

Der Text hat 9 Wörter, also gibt es $9 - 1 = 8$ Bigramme. Du findest sie, indem du ein „Fenster“ aus zwei Wörtern über den Text schiebst – immer um ein Wort weiter:



Ein Fenster aus zwei Wörtern wandert über den Text. Jede Klammer markiert ein Bigramm, die Zahl gibt die Reihenfolge an

Als Liste geschrieben sind das die acht Bigramme: der hund · hund läuft · läuft der · der hund · hund bellt · bellt der · der hund · hund schläft.

Jetzt wird gezählt. Am einfachsten geht das mit einer **Strichliste**: Du gehst die Liste von links nach rechts durch. Kommt ein Bigramm zum ersten Mal vor, bekommt es eine neue Zeile. Kommt es noch einmal vor, machst du einen weiteren Strich.

Bigramm	Strichliste	Anzahl
der hund	III	3
hund läuft	I	1
läuft der	I	1
hund bellt	I	1
bellt der	I	1
hund schläft	I	1
Summe		**8**

Die Summe muss genau die Anzahl der Bigramme ergeben, hier 8. Das ist eine gute Kontrolle, ob du keines vergessen hast.

Das Ergebnis: Auf das Wort „der“ folgt in diesem Text **dreimal** „hund“ – und nie etwas anderes. Das ist mit Abstand das häufigste Bigramm mit „der“ als erstem Wort. Auf „hund“ folgen dagegen drei verschiedene Wörter, jedes genau einmal.

Tipp

Beim Zählen von Hand passieren leicht Fehler. Nummeriere die Wörter des Textes vorher durch und streiche jedes Bigramm in der Liste ab, sobald du es gezählt hast.

Arbeitsauftrag 11: Von Hand zählen

Gegeben ist der Textkorpus:

> die katze schläft die katze spielt die katze frisst die katze schläft

1. Ermittle **von Hand** alle Bigramme und zähle, wie oft jedes vorkommt. Lege dazu eine Tabelle mit Strichliste an.
2. Welches Wort folgt im Korpus am häufigsten direkt auf „die“?
3. Welches Wort folgt am häufigsten auf „katze“?

Meine Auszählung:

4.2.4 Von der Häufigkeit zur Wahrscheinlichkeit

Häufigkeiten allein reichen noch nicht. Ein Sprachmodell will wissen: **Wenn** gerade das Wort „der“ dasteht – **wie wahrscheinlich** ist es dann, dass als nächstes „hund“ kommt? Dafür vergleicht man, wie oft „der hund“ vorkommt, mit der Anzahl **aller** Bigramme, die mit „der“ beginnen.

Wir rechnen das an einem neuen Korpus Schritt für Schritt durch. Er ist etwas abwechslungsreicher, damit interessantere Zahlen entstehen:

> der hund bellt der hund läuft der ball rollt der hund bellt

Beispiel: Wahrscheinlichkeiten Schritt für Schritt

Schritt 1: Wörter und Bigramme ermitteln. Der Text hat 12 Wörter, also $12 - 1 = 11$ Bigramme:

der hund · hund bellt · bellt der · der hund · hund läuft · läuft der · der ball · ball rollt · rollt der · der hund · hund bellt

Schritt 2: Bigramme zählen und nach dem ersten Wort ordnen. Alle Bigramme mit demselben ersten Wort stehen untereinander. Das erste Wort nennen wir **Vorgängerwort**, das zweite **Folgewort**.

Vorgängerwort	Folgewort	Anzahl
der	hund	3
der	ball	1
hund	bellt	2
hund	läuft	1
bellt	der	1
läuft	der	1
ball	rollt	1
rollt	der	1

Kontrolle: $3 + 1 + 2 + 1 + 1 + 1 + 1 + 1 = 11$. ✓

Schritt 3: Wie oft steht jedes Vorgängerwort vorne in einem Bigramm? Dazu addierst du die Anzahlen in seinen Zeilen: „der“ $3 + 1 = 4$, „hund“ $2 + 1 = 3$, „bellt“ 1, „läuft“ 1, „ball“ 1, „rollt“ 1.

Schritt 4: Wahrscheinlichkeit berechnen. Teile die Anzahl des Bigramms durch die Zahl aus Schritt 3. Für „der“, gefolgt von „hund“:

$$P(\text{der} \to \text{hund}) = \frac{3}{4} = 0{,}75 = 75\,\%$$

Für „hund“, gefolgt von „bellt“:

$$P(\text{hund} \to \text{bellt}) = \frac{2}{3} \approx 0{,}67 = 67\,\%$$

Das Zeichen P steht für *probability*, englisch für Wahrscheinlichkeit. $P(\text{der} \to \text{hund})$ liest du: „die Wahrscheinlichkeit, dass auf ‚der‘ das Wort ‚hund‘ folgt“.

Rechnest du das für alle Zeilen aus, erhältst du die vollständige **Wahrscheinlichkeitstabelle** des Modells:

Vorgängerwort	Folgewort	Anzahl	Rechnung	Wahrscheinlichkeit
der	hund	3	$\frac{3}{4}$	0,75 = 75 %
der	ball	1	$\frac{1}{4}$	0,25 = 25 %
hund	bellt	2	$\frac{2}{3}$	≈ 0,67 = 67 %
hund	läuft	1	$\frac{1}{3}$	≈ 0,33 = 33 %
bellt	der	1	$\frac{1}{1}$	1 = 100 %
läuft	der	1	$\frac{1}{1}$	1 = 100 %
ball	rollt	1	$\frac{1}{1}$	1 = 100 %
rollt	der	1	$\frac{1}{1}$	1 = 100 %

Für jedes Vorgängerwort ergeben die Wahrscheinlichkeiten zusammen genau 1, also 100 %: bei „der“ $0{,}75 + 0{,}25 = 1$, bei „hund“ $\frac{2}{3} + \frac{1}{3} = 1$. Das ist logisch, denn **irgendein** Folgewort kommt nach „der“ ja immer. Diese Summe ist deine Kontrolle.

Merke

Die **Wahrscheinlichkeit**, dass auf ein Wort A das Wort B folgt, berechnet man so:

$$P(A \to B) = \frac{\text{Anzahl des Bigramms „A B“}}{\text{Anzahl aller Bigramme, die mit A beginnen}}$$

Für jedes Vorgängerwort ergeben die Wahrscheinlichkeiten aller Folgewörter zusammen 1 (also 100 %).

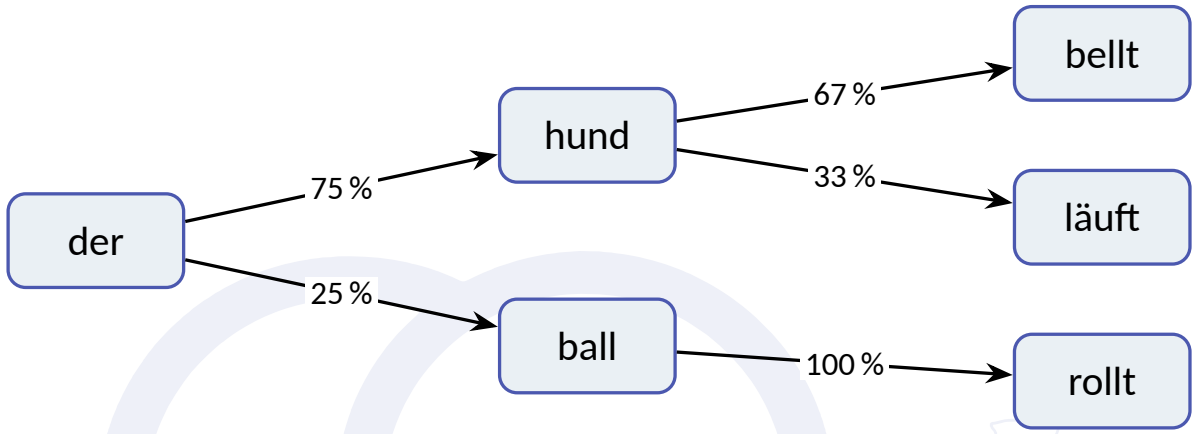
Gymnasium Ottweiler

Stolperstein

Im Nenner steht **nicht** die Gesamtzahl aller Bigramme im Text, sondern nur die Anzahl der Bigramme, die mit dem Vorgängerwort **beginnen**. $\frac{3}{11}$ wäre für „der hund“ also falsch.

Vorsicht auch beim letzten Wort: „bellt“ steht zweimal im Text. Das zweite „bellt“ ist aber das letzte Wort, danach folgt nichts mehr. Deshalb beginnt nur **ein** Bigramm mit „bellt“, und es gilt $P(\text{bellt} \to \text{der}) = \frac{1}{1}$, nicht $\frac{1}{2}$. Zähle im Nenner also nur Vorkommen, auf die noch ein Wort folgt.

Man kann ein solches Modell auch als **Wortgraph** zeichnen: Jedes Wort ist ein Kästchen, und ein Pfeil führt zu jedem möglichen Folgewort. Am Pfeil steht die Wahrscheinlichkeit.



Ausschnitt aus dem Bigramm-Modell zu „der hund bellt der hund läuft der ball rollt der hund bellt“: die Folgewörter von „der“, „hund“ und „ball“. Die Pfeile zurück zu „der“ sind weggelassen

Merke

Aus den Häufigkeiten lässt sich die **Wahrscheinlichkeit** ableiten, mit der auf ein bestimmtes Wort (bzw. eine Wortfolge) ein bestimmtes Folgewort folgt. Das ist die Grundlage für die spätere **Textgenerierung**.

So kannst du es sagen

So beschreibst du deine Ergebnisse in Fachsprache:

- *„Nach der kommt meistens hund.“* → *„Im Korpus folgt auf ‚der‘ in drei von vier Fällen ‚hund‘.“*
- *„hund bellt ist 67 %.“* → *„Die Wahrscheinlichkeit, dass auf ‚hund‘ das Wort ‚bellt‘ folgt, beträgt* $\frac{2}{3} \approx 67\,\%$*.“*

Arbeitsauftrag 12: Wahrscheinlichkeiten berechnen

Verwende deine Häufigkeiten aus dem Auftrag *Von Hand zählen* (Korpus „die katze schläft die katze spielt …“).

1. Berechne die **Wahrscheinlichkeit** für jedes Wort, das auf „die“ folgt: Anzahl des jeweiligen Bigramms geteilt durch die Gesamtanzahl aller Bigramme, die mit „die“ beginnen.

2. Berechne ebenso die Wahrscheinlichkeiten für alle Wörter, die auf „katze“ folgen. Gib sie als Bruch, als Dezimalzahl und in Prozent an.

Meine Rechnung:

Arbeitsauftrag 13: Fehler finden

Tim soll für den Korpus „ich mag tee ich mag kaffee ich trinke tee“ die Wahrscheinlichkeitstabelle aufstellen. Er schreibt:

Vorgängerwort	Folgewort	Wahrscheinlichkeit
ich	mag	$\frac{2}{8}$
ich	trinke	$\frac{1}{8}$
tee	ich	$\frac{1}{2}$

1. Beschreibe Tims zwei Fehler.
2. Stelle die vollständige und richtige Wahrscheinlichkeitstabelle für diesen Korpus auf.

Fehler und richtige Tabelle:

Gymnasium Ottweiler

4.2.5 Trigramme: mehr Wörter als Kontext

Bei Bigrammen hängt das Folgewort nur vom **einen** Wort davor ab. Bei Trigrammen schaut man auf die **zwei** Wörter davor. Diese Wörter, auf die das Modell beim Vorhersagen schaut, heißen **Kontext**.

Im Korpus „der hund bellt der hund läuft der ball rollt der hund bellt“ gibt es $12 - 2 = 10$ Trigramme. Drei davon beginnen mit dem Kontext „der hund“:

Kontext	Folgewort	Anzahl	Wahrscheinlichkeit
der hund	bellt	2	$\frac{2}{3} \approx 67\,\%$
der hund	läuft	1	$\frac{1}{3} \approx 33\,\%$

Gerechnet wird genau wie bei Bigrammen: Anzahl des Trigramms geteilt durch die Anzahl aller Trigramme mit demselben Kontext. In diesem kleinen Text ändert sich dadurch noch nichts. In langen Texten macht der längere Kontext aber einen großen Unterschied: Nach „die“ kann fast alles kommen, nach „die katze“ schon viel weniger. Wie sich das auf erzeugte Texte auswirkt, untersuchst du im Abschnitt über Textgenerierung.

4.2.6 Kontrolle mit Python

Bei kurzen Texten geht das Zählen von Hand. Bei langen Texten ist es mühsam und fehleranfällig. Hier hilft ein Programm. Zuerst muss der Text in einzelne Wörter zerlegt werden.

Werkzeug

Mit `text.split()` zerlegst du eine Zeichenkette an den Leerzeichen in ihre Wörter. Das Ergebnis ist eine **Liste** von Zeichenketten. Mehrere Leerzeichen und Zeilenumbrüche hintereinander zählen dabei wie ein einziges Leerzeichen.

```
satz = "der hund bellt laut"
woerter = satz.split()
print(woerter)
print(len(woerter))
```

AUSGABE

```
['der', 'hund', 'bellt', 'laut']
4
```

Mit dieser Liste kannst du alle Bigramme ausgeben. Das Bigramm Nummer `i` besteht aus dem Wort an der Stelle `i` und dem Wort an der Stelle `i + 1`:

```
korpus = "der hund läuft der hund bellt der hund schläft"
woerter = korpus.split()
for i in range(len(woerter) - 1):
    print(woerter[i], woerter[i + 1])
```

AUSGABE

```
der hund
hund läuft
läuft der
der hund
hund bellt
bellt der
der hund
hund schläft
```

- **Zeile 2:** `woerter` ist jetzt die Liste `['der', 'hund', 'läuft', …]` mit 9 Elementen. Die Indizes laufen von 0 bis 8.
- **Zeile 3:** Die Schleife läuft nur bis `len(woerter) - 1`, also für `i` von 0 bis 7. Das sind genau 8 Durchläufe – so viele, wie es Bigramme gibt.
- **Zeile 4:** Ausgegeben werden das Wort an Stelle `i` und sein Nachfolger an Stelle `i + 1`. Das ist das „Fenster“ aus der Abbildung oben.

Stolperstein

Schreibst du in der Schleife nur `range(len(woerter))`, läuft `i` bis zum letzten Index. Im letzten Durchlauf will das Programm dann das Wort **nach** dem letzten Wort lesen. Das gibt es nicht. Hier ein kurzes Beispiel mit drei Wörtern (Indizes 0 bis 2): Im dritten Durchlauf ist `i = 2`, und `woerter[3]` existiert nicht.

```
1 woerter = "der hund bellt".split()
2 for i in range(len(woerter)):
3     print(woerter[i], woerter[i + 1])
```

AUSGABE

```
der hund
hund bellt
```

FEHLERMELDUNG

```
IndexError: list index out of range
```

Das letzte Wort hat keinen Nachfolger. Deshalb gibt es ein Bigramm weniger als Wörter.

Jetzt soll das Programm auch zählen. Dazu speichern wir jedes Bigramm als Zeichenkette aus zwei Wörtern, zum Beispiel `"die katze"`, in einer Liste. Dann gehen wir so vor wie bei der Strichliste: Zuerst sammeln wir alle **verschiedenen** Bigramme, dann zählen wir für jedes, wie oft es in der Liste vorkommt.

```
1  def zaehle(liste, gesucht):
2      anzahl = 0
3      for element in liste:
4          if element == gesucht:
5              anzahl += 1
6      return anzahl
7
8  korpus = "die katze schläft die katze spielt "
9  korpus = korpus + "die katze frisst die katze schläft"
10 woerter = korpus.split()
11
12 bigramme = []
13 for i in range(len(woerter) - 1):
14     bigramme.append(woerter[i] + " " + woerter[i + 1])
15
16 verschiedene = []
17 for bigramm in bigramme:
18     if bigramm not in verschiedene:
19         verschiedene.append(bigramm)
20
21 for bigramm in verschiedene:
22     print(bigramm, ":", zaehle(bigramme, bigramm))
```

AUSGABE

```
die katze : 4
katze schläft : 2
schläft die : 1
katze spielt : 1
spielt die : 1
katze frisst : 1
frisst die : 1
```

- **Zeilen 1–6:** Eine solche Zählfunktion kennst du von den Listenalgorithmen. `zaehle` zählt, wie oft `gesucht` in der Liste vorkommt.
- **Zeilen 8–10:** Der Korpus ist für eine Zeile zu lang. Deshalb wird er in zwei Teilen gebildet und mit `+` zusammengesetzt. Dann wird er in Wörter zerlegt.
- **Zeilen 12–14:** Für jede Position `i` wird das Bigramm als Zeichenkette gebildet, z. B. `"die" + " " + "katze"` ergibt `"die katze"`, und an die Liste `bigramme` angehängt. Sie enthält danach alle 11 Bigramme, auch die doppelten.
- **Zeilen 16–19:** In `verschiedene` kommt jedes Bigramm nur einmal, nämlich beim ersten Auftreten. Das entspricht dem Anlegen einer neuen Zeile in der Strichliste.
- **Zeilen 21–22:** Für jedes verschiedene Bigramm wird gezählt, wie oft es in `bigramme` steht.

Vergleiche die Ausgabe mit deiner Lösung zum Auftrag *Von Hand zählen*. Die Zahlen müssen übereinstimmen.

Versuch: Probiere es aus

Setze in den Zeilen 8 und 9 den Korpus „der hund bellt der hund läuft der ball rollt der hund bellt“ ein und vergleiche mit der Tabelle aus dem Beispiel *Wahrscheinlichkeiten Schritt für Schritt*. Ergänze dann eine Zeile, die die Anzahl aller Bigramme mit `len(bigramme)` ausgibt.

4.2.7 Übungen

Arbeitsauftrag 14 (Zusatz): Trigramme

Erweitere das Programm aus dem Abschnitt *Kontrolle mit Python* so, dass es **Trigramme** (Dreiergruppen) statt Bigramme zählt. Verwende deinen eigenen, etwas längeren Beispielsatz.

4.2.8 Weitere Übungen

Hier kannst du das Gelernte weiter üben. Die Punkte im Titel zeigen, wie schwer eine Aufgabe ist: ●○○ leicht, ●●○ mittel, ●●● knifflig. Fang mit den leichten an.

Arbeitsauftrag 15: n-Gramme aufschreiben ●○○

1. Bestimme für den Satz „wir fahren morgen nach saarbrücken“ alle Unigramme, Bigramme und Trigramme und gib jeweils die Anzahl an.
2. n-Gramme kann man auch aus **Zeichen** bilden. Bestimme die Zeichen-Bigramme des Wortes „kino“.

n-Gramme	Anzahl
Unigramme	
Bigramme	
Trigramme	
Zeichen-Bigramme von „kino“	

Arbeitsauftrag 16: Wie viele Bigramme? ●○○

Bestimme für jeden Text die Anzahl der Wörter, die Anzahl der Bigramme und die Anzahl der **verschiedenen** Bigramme.

Text	Wörter	Bigramme	verschiedene Bigramme
ich bin ich			
la la la la			
heute ist heute und morgen ist morgen			
du und ich und du und ich			

Arbeitsauftrag 17: Strichliste ●○○

Gegeben ist der Textkorpus:

> wir spielen fußball wir spielen basketball wir schauen fußball

1. Ermittle alle Bigramme und zähle sie mit einer Strichliste.
2. Welches Wort folgt am häufigsten auf „wir“?

Bigramm	Strichliste	Anzahl
Summe		

Arbeitsauftrag 18: Wahrscheinlichkeiten aus einer Tabelle ●○○

In einem großen Korpus wurde gezählt, welche Wörter auf „gehe“ folgen: 5-mal „ins“, 3-mal „nach“ und 2-mal „zum“. Andere Folgewörter gibt es nicht.

1. Berechne die Wahrscheinlichkeit für jedes Folgewort als Bruch, als Dezimalzahl und in Prozent.
2. Kontrolliere mit der Summe.
3. Welches Folgewort ist am wahrscheinlichsten?

Kontext	Folgewort	Anzahl	Bruch	Dezimalzahl	Prozent
gehe	ins	5			
gehe	nach	3			
gehe	zum	2			

Arbeitsauftrag 19: Ein Modell von Hand trainieren ●●○

Gegeben ist der Textkorpus:

> ich höre musik ich höre podcasts ich lese comics ich höre musik

1. Ermittle alle Bigramme und ihre Häufigkeiten.
2. Stelle die vollständige Wahrscheinlichkeitstabelle auf (Kontext, Folgewort, Anzahl, Wahrscheinlichkeit als Bruch und in Prozent).
3. Kontrolliere: Ergeben für jeden Kontext die Wahrscheinlichkeiten zusammen 1?

Meine Auszählung und meine Tabelle:

Arbeitsauftrag 20: Mehr Kontext mit Trigrammen ●●○

Gegeben ist der Textkorpus:

> er spielt gitarre sie spielt geige er spielt gitarre sie spielt klavier

1. Bestimme mit Bigrammen die Wahrscheinlichkeiten aller Folgewörter von „spielt“.
2. Bestimme alle Trigramme, die mit dem Kontext „er spielt“ beginnen, und alle, die mit dem Kontext „sie spielt“ beginnen. Berechne die Wahrscheinlichkeiten der Folgewörter.
3. Vergleiche: Was ändert sich, wenn das Modell zwei Wörter als Kontext benutzt statt einem?

Meine Rechnungen und mein Vergleich:

Gymnasium Ottweiler

Arbeitsauftrag 21: Ausgabe vorhersagen ●●○

Bestimme **ohne Computer**, was das Programm ausgibt. Wie viele Zeilen sind es?

```
korpus = "ich mag eis ich mag pizza"
woerter = korpus.split()
for i in range(len(woerter) - 2):
    print(woerter[i], woerter[i + 1], woerter[i + 2])
```

Ausgabe:

Arbeitsauftrag 22: Korpus gesucht ●●●

1. Schreibe einen möglichst kurzen Korpus, für den gilt:
   - Auf „ich“ folgt mit der Wahrscheinlichkeit $\frac{2}{3}$ „lache“ und mit $\frac{1}{3}$ „weine“.
   - Auf „lache“ folgt immer „ich“.
2. Kontrolliere deinen Korpus, indem du die Wahrscheinlichkeitstabelle aufstellst.
3. Begründe, warum ein solcher Korpus aus mindestens 6 Wörtern bestehen muss.

Mein Korpus, meine Kontrolle und meine Begründung:

Gymnasium Ottweiler

4.2.9 Das Wichtigste in Kürze

- Ein **n-Gramm** ist eine Folge von n aufeinanderfolgenden Wörtern: Unigramm (1), Bigramm (2), Trigramm (3). Ein Text mit N Wörtern hat $N - 1$ Bigramme.
- Der **Textkorpus** ist der Trainingstext. Bei der **Textkörperanalyse** zählt man, wie oft jedes n-Gramm darin vorkommt.
- Bigramme findet man mit einem „Fenster“ aus zwei Wörtern, das Wort für Wort weiterrückt. Gezählt wird mit einer Strichliste.
- $P(A \to B)$ = Anzahl von „A B“ geteilt durch die Anzahl aller Bigramme, die mit A beginnen.
- Für jedes Vorgängerwort ergeben die Wahrscheinlichkeiten zusammen 1 (100 %).
- Bei Trigrammen besteht der **Kontext** aus zwei Wörtern.
- In Python zerlegt `text.split()` einen Text in eine Liste von Wörtern. Damit kann man alle n-Gramme bilden und zählen.

Selbstcheck

- ☐ Ich kann zu einem Satz alle Unigramme, Bigramme und Trigramme angeben.
- ☐ Ich kann erklären, was ein Textkorpus und eine Textkörperanalyse sind.
- ☐ Ich kann in einem kurzen Text alle Bigramme von Hand zählen.
- ☐ Ich kann die Wahrscheinlichkeit für ein Folgewort berechnen und weiß, durch welche Zahl ich teilen muss.
- ☐ Ich kann eine Wahrscheinlichkeitstabelle aufstellen und mit der Summe 1 kontrollieren.
- ☐ Ich kann mit `split()` und einer Schleife alle Bigramme eines Textes ausgeben lassen.

Gymnasium Ottweiler