

INHALT

1. A Indexierungsprojekt
2. B Klassifikationsprojekt

MEIN INDEXIERUNGSPROJEKT

....



A.1 PROBLEMSTELLUNG INDEXIERUNG

Meinen Korpus an Transkripten von Radiobeiträgen habe ich mithilfe von sechs Suchwörtern erstellt, „Literatur“, „Ausstellung“, „Konzert“, „Biografie“, „Tourismus“ und „Sport“.

Innerhalb meines Indexierungsprojektes möchte ich nun genau diesen Suchweg überprüfen und herausfinden, welche Relevanz die einzelnen Suchwörter für meinen Korpus haben. Welche von ihnen sind wirklich relevant für die Texte und welche dienen eher nur als Schlagwort? Welche Erkenntnisse kann ich daraus für die künftige Festlegung von Suchbegriffen ziehen?

*Zum einen möchte ich meine Frage mit Hilfe der Indexierung meines Korpus beantworten. Zum anderen möchte ich die Termfrequenz sowie das $tf*idf$ für die Suchbegriffe berechnen und schauen, inwieweit diese Werte bei der Bestimmung der Relevanz helfen können.*

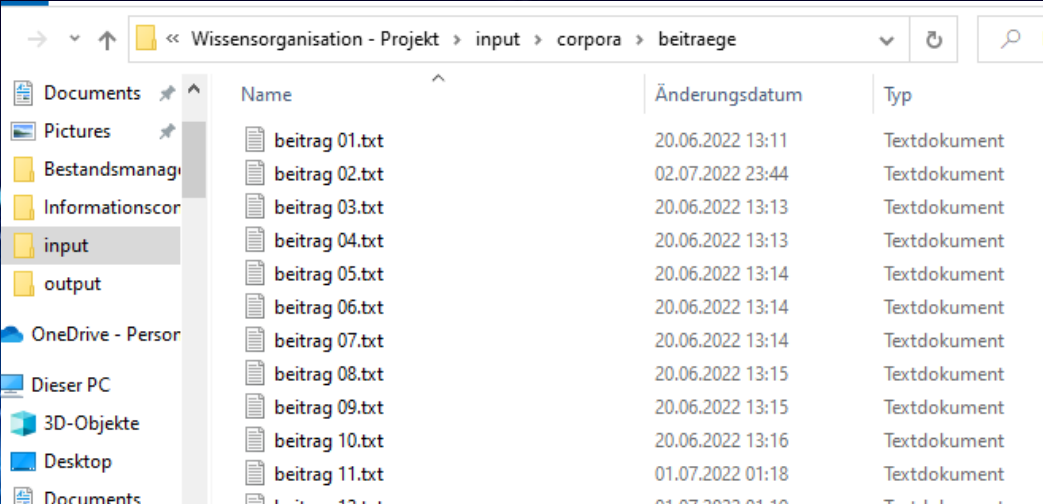
A. 2 DATEN

Mein Korpus besteht aus insgesamt 72 Transkripten von Radiobeiträgen des Norddeutschen Rundfunks (NDR). Die Radiobeiträge stammen allesamt aus den Jahren 2020 bis 2022 und wurden hauptsächlich auf den Sendern NDR 90,3 und NDR Info gesendet, aber auch auf den Kanälen NDR Kultur, NDR 1 Radio MV und NDR 1 Welle Nord. Alle Beiträge haben eine Länge zwischen zweieinhalb und dreieinhalb Minuten, weshalb sie auch von der Textlänge vergleichbar sind.

Den Bezug zu dem Thema habe ich aufgrund meiner Werkstudentenstelle beim Norddeutschen Rundfunk, wo ich hauptsächlich damit zu tun habe, Beiträge des Senders NDR 90,3 in der Hörfunkdatenbank zu katalogisieren und zu verschlagworten.

Die Audio-Transkripte der Hörfunkdatenbank sind allesamt durch ein automatisches Mining erstellt worden.

Da die entsprechende Technik noch nicht vollständig ausgereift ist, kommt es regelmäßig vor, dass die Transkripte Schreibfehler oder falsch transkribierte Worte enthalten.



The screenshot shows a Windows File Explorer window with the address bar set to 'Wissensorganisation - Projekt > input > corpora > beitraege'. The left sidebar shows the navigation pane with 'input' selected. The main pane displays a list of files with columns for Name, Änderungsdatum, and Typ.

Name	Änderungsdatum	Typ
beitrag 01.txt	20.06.2022 13:11	Textdokument
beitrag 02.txt	02.07.2022 23:44	Textdokument
beitrag 03.txt	20.06.2022 13:13	Textdokument
beitrag 04.txt	20.06.2022 13:13	Textdokument
beitrag 05.txt	20.06.2022 13:14	Textdokument
beitrag 06.txt	20.06.2022 13:14	Textdokument
beitrag 07.txt	20.06.2022 13:14	Textdokument
beitrag 08.txt	20.06.2022 13:15	Textdokument
beitrag 09.txt	20.06.2022 13:15	Textdokument
beitrag 10.txt	20.06.2022 13:16	Textdokument
beitrag 11.txt	01.07.2022 01:18	Textdokument
beitrag 12.txt	01.07.2022 01:18	Textdokument

A. 2 DATEN

Die Beiträge für den Korpus habe ich aus der Hörfunkdatenbank des NDR zusammengetragen. Hierzu habe ich mir sechs Suchwörter überlegt, nach welchen ich über die Funktion „Überall suchen“ in der regulären Schreibweise und ohne Ergänzungen recherchiert habe. Aus den Ergebnislisten habe ich pro Suchwort zwölf Beiträge ausgewählt, so dass insgesamt 72 Beiträge zusammengekommen sind. Die Beitragsauswahl ist auf diese Weise natürlich subjektiv beeinflusst worden. Die sechs Suchwörter werden im Folgenden auch in den Arbeitsaufgaben als Kategorien relevant sein.

Bei den sechs Suchbegriffen handelt es sich um folgende:

- Literatur (Beiträge 01-12)
- Ausstellung (Beiträge 13-24)
- Konzert (Beiträge 25-36)
- Biografie (Beiträge 37-48)
- Tourismus (Beiträge 49-60)
- Sport (Beiträge 61-72)

The screenshot shows the 'Recherche' (Search) interface of the NDR database. The search criteria are set to 'NDR' in the 'Ziele/Filter' field. The 'Suchobjekt' (Search object) is set to 'Konf - Korpus'. The 'Rechercheart (Vokabulare)' (Search type) is set to 'Ohne Hierarchien und Synonyme'. A table of search fields is displayed below the criteria:

Feld	Operator	Suchbegriff(e)
Titel	WITH	
Archivnummer	ANY	
Überall suchen	WITH	sport
Mitwirkende	WITH	
O-Ton von	ADJ	
O-Ton von	ADJ	
O-Ton von	ADJ	
Urheber/Produktion/Mitwirkung	WITH	
Verbreitungsdatum	=	

A. 3 VORGEHENSWEISE

Bei der Erstellung der Stopwortliste habe ich mir zunächst die häufigsten Wörter im ganzen Korpus anzeigen lassen und für die Liste dann diejenigen ausgewählt, die am häufigsten vorkommen, aber keine Relevanz für meine Fragestellung besitzen. Substantive habe ich nur im Einzelfall mit aufgenommen, etwa „sprecher“ und „sprecherin“, weil diese aufgrund des Textminings in jedem Beitrag in zweistelliger Zahl vorkommen. Ebenso kommt die Abkürzung „ndr“ in jedem Beitrag vor. Das Kürzel „mv“ ist im Korpus ebenso keine Seltenheit, da einige Beiträge vom Radiosender NDR 1 Radio MV stammen.

Stopwortliste: *sprecher, sprecherin, ja, schon, ganz, ndr, mal, mehr, gibt, einfach, immer, sagt, natürlich, heute, geht, eben, wirklich, viele, wurde, eigentlich, gut, zwei, neue, kommt, ersten, macht, drei, gerade, mv, weiß, vielleicht, seit, große, gar, sagen, sieht, bisschen, erste, beim, kam, ab*

A. 3 VORGEHENSWEISE

Meinen Thesaurus habe ich auf Grundlage der Schlagwörter erstellt, die ich für meine 12 ausgewählten Texte vergeben habe (jeweils 2 Texte pro Suchbegriff). Die Schlagwortliste befindet sich in einer extra PDF-Datei im Ordner.

Ebenfalls im Thesaurus zu finden sind die genannten sechs Suchbegriffe sowie die fünf häufigsten Begriffe des Korpus, wobei es hier Überschneidungen gibt.

Des weiteren wurden Wortvarianten mit einbezogen, die auf dem selben Wortstamm basieren.

Beitrag 04: Manfred Ertel – „Akte B.“

Suchworte: Uwe Barschel, Ministerpräsident, Kriminalroman

Beitrag 50: Perspektiven für den Tourismus in M-V

Suchworte: Mecklenburg-Vorpommern, Urlaub, Übernachtung

A. 3 VORGEHENSWEISE

Für die Erstellung des Thesaurus, insbesondere für die Auswahl der Schlagworte und Oberbegriffe habe ich die ARD-Sachklassifikation genutzt. Dies hat sich angeboten, da die Beiträge im NDR-Schallarchiv ebenfalls mit dieser verschlagwortet werden.

The screenshot shows the 'Klasse/Deskriptor' software interface. On the left, a 'Hierarchie' tree lists various literary genres, with 'Kriminalroman' selected. On the right, a search window shows the search term 'Kriminalroma' and a search result table.

Hierarchie

- SD Gegenwartsliteratur
- KL Literarischer Stoff
- KL Literarisches Werk
 - SD Abenteuerroman
 - SD Arbeiterliteratur
 - SD Ausländerliteratur
 - KL Bildergeschichte
 - SD Biographischer Roman
 - SD Erotische Literatur
 - SD Exilliteratur
 - SD Experimentelle Literatur
 - SD Fantasy-Literatur
 - SD Frauenliteratur
 - SD Gesellschaftsroman
 - SD Groschenheft
 - SD Heimatliteratur
 - SD Historischer Roman
 - SD Horrorliteratur
 - SD Jugendliteratur
 - SD Kinderliteratur
 - SD **Kriminalroman**
 - SD Liebesroman
 - SD Mundartliteratur

Suchen

Kriminalroma

Alles Nur Klassen
 Nur Sachdeskriptoren Alles ohne Freie Sachdeskriptoren

Suchergebnis Seite 1 / 1 (Treffer gesamt: 1)

Vokabelname	Typ	Anza...
Kriminalroman	SD	233

Auswahl

Übernehmen Abbrechen

A. 3 VORGEHENSWEISE

Den Thesaurus habe ich in fünf Spalten eingeteilt:

deskriptor_stemming	Regelgestemmtes Wort
nicht_deskriptor	Tokens, die auf den jeweiligen Wortstamm zurückgeführt werden sollen
deskriptor_kontrolliert	Intellektuell vergebenes Suchwort bzw. Schlagwort
oberbegriff	Oberbegriff bzw. Definition für das Schlag- bzw. Suchwort
begrueundung	Quellennachweis für die Begriffswahl bzw. Erläuterung

1 **deskriptor_stemming** **nicht_deskriptor**

2 **deskriptor_kontrolliert** **oberbegriff**

deskriptor_kontrolliert **oberbegriff** **begrueundung**

A. 3 VORGEHENSWEISE

Zur Klärung meiner Fragestellung konnte ich zunächst auf die Suchfunktion zurückgreifen, welche ich als Teil der Arbeitsaufgabe erstellt habe. Neben meinen Suchwörtern und den fünf häufigsten Wörtern lässt sich damit auch der Kontext aller sechs Suchwörter darstellen, mit denen ich meinen Korpus erstellt habe.

Mit Hilfe dessen kann man zunächst grob auswerten, wie oft jedes Suchwort jeweils in den Texten des eigenen Themenbereiches vorkommt.

```
for satz in result_df[ 'satz' ]:  
    gefiltert = result_df.loc[result_df['satz'] == satz]  
    print(gefiltert['dokument'], '\n', 'Suchbegriff:', gefiltert['suchterm'], '\n\n', satz, '\n\n')
```

```
12  beitrag 64.txt  
Name: dokument, dtype: object  
Suchbegriff: 12  sport  
Name: suchterm, dtype: object  
  
Auch Frauen die gerne beim Sport zuschauen sind in Saudi-Arabien eigentlich immer noch verpönt.  
  
13  beitrag 64.txt  
Name: dokument, dtype: object  
Suchbegriff: 13  sport  
Name: suchterm, dtype: object  
  
Was die Spielerin Al Hashim über die Bedeutung der Liga für den Sport sagt gilt daher auch für die ganze Gesellschaft.  
  
14  beitrag 64.txt  
Name: dokument, dtype: object  
Suchbegriff: 14  sport
```

A. 3 VORGEHENSWEISE

Anschließend habe ich die Termfrequenz und die Inverse Dokument Frequenz für die einzelnen Dokumente berechnet. Um die Auswertung für meine Fragestellung übersichtlicher beantworten zu können, habe ich die Sammlungen der Dokumente bei der Anzeige der Termfrequenz entsprechend den sechs Suchwörtern aufgeteilt.

```
#Literatur
#txtDocs = [txt1, txt2, txt3, txt4, txt5, txt6, txt7, txt8, txt9, txt10, txt11, txt12]

#Ausstellung
#txtDocs = [txt13, txt14, txt15, txt16, txt17, txt18, txt19, txt20, txt21, txt22, txt23, txt24]

#Konzert
#txtDocs = [txt25, txt26, txt27, txt28, txt29, txt30, txt31, txt32, txt33, txt34, txt35, txt36]

#Biografie
#txtDocs = [txt37, txt38, txt39, txt40, txt41, txt42, txt43, txt44, txt45, txt46, txt47, txt48]

#Tourismus
#txtDocs = [txt49, txt50, txt51, txt52, txt53, txt54, txt55, txt56, txt57, txt58, txt59, txt60]

#Sport
txtDocs = [txt61, txt62, txt63, txt64, txt65, txt66, txt67, txt68, txt69, txt70, txt71, txt72]

docs = [list(nltk.tokenize.word_tokenize(txtDoc)) for txtDoc in txtDocs]
```

A. 3 VORGEHENSWEISE

*Auch bei der Berechnung der Termfrequenz*Inverse Dokument Frequenz (tf*idf) habe ich diese Unterteilung nach den sechs Suchbegriffen vorgenommen. Da es bei der Berechnung von tf*idf aber auch interessant ist, diese in Bezug auf den gesamten Korpus zu analysieren, habe ich hier zusätzlich eine entsprechende Sammlung definiert.*

```
# Dokumente nach Themen geordnet
#Literatur
#txtDocs = [txt1, txt2, txt3, txt4, txt5, txt6, txt7, txt8, txt9, txt10, txt11, txt12]

#Ausstellung
#txtDocs = [txt13, txt14, txt15, txt16, txt17, txt18, txt19, txt20, txt21, txt22, txt23, txt24]

#Konzert
#txtDocs = [txt25, txt26, txt27, txt28, txt29, txt30, txt31, txt32, txt33, txt34, txt35, txt36]

#Biografie
#txtDocs = [txt37, txt38, txt39, txt40, txt41, txt42, txt43, txt44, txt45, txt46, txt47, txt48]

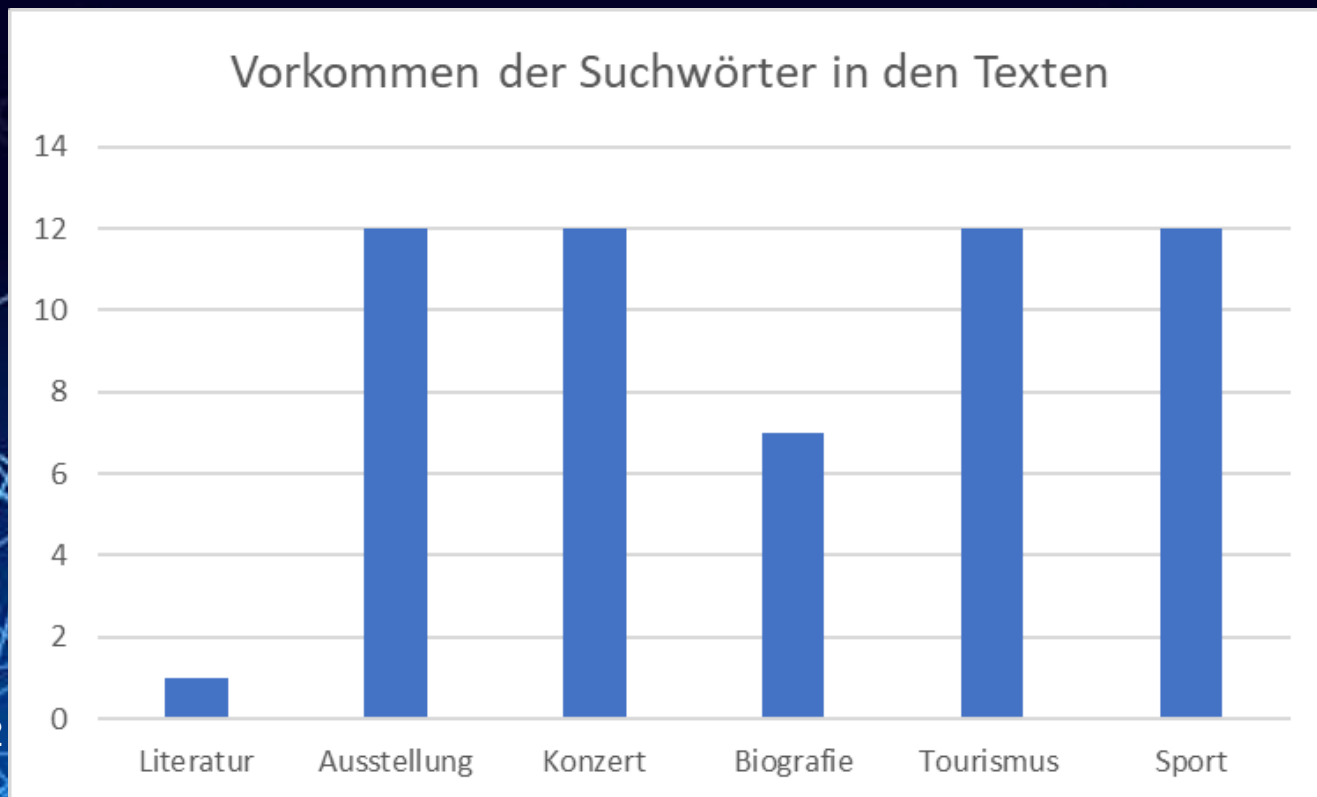
#Tourismus
#txtDocs = [txt49, txt50, txt51, txt52, txt53, txt54, txt55, txt56, txt57, txt58, txt59, txt60]

#Sport
#txtDocs = [txt61, txt62, txt63, txt64, txt65, txt66, txt67, txt68, txt69, txt70, txt71, txt72]

#Gesamter Korpus
txtDocs = [txt1, txt2, txt3, txt4, txt5, txt6, txt7, txt8, txt9, txt10, txt11, txt12, txt13, txt14, txt15, txt16, txt17]
```

A.4 ERGEBNISSE

Mit Hilfe der Suchfunktion konnte ich auswerten, in wie vielen Dokumenten des jeweiligen Themenblocks die Suchwörter vorkommen. Dabei ging es nur um die Substantive in exakter Schreibweise und Einzahl. Vier der Suchbegriffe, Ausstellung, Konzert, Tourismus und Sport, kommen in allen Texten ihres Themenbereiches vor. Bei Biografie sind es sieben Texte, Literatur kommt nur in einem Text vor.



A.4 ERGEBNISSE

*Mit den beschriebenen Mitteln habe ich die Termfrequenz sowie die $tf*idf$ berechnet, letzteres einmal bezogen auf den eigenen Themenbereich – also jeweils die 12 Dokumente, die das Thema behandeln – sowie einmal bezogen auf den gesamten Korpus.*

Auf den folgenden Folien habe ich für jeden Suchbegriff zwei Beiträge als Beispiel aufgeführt (außer Literatur, da der Begriff nur einmal in seinem Themenabschnitt auftaucht).

Im Notebook werden auch die absoluten Worthäufigkeiten pro Text angezeigt. Daran lässt sich ebenfalls ablesen, wie oft der entsprechende Begriff pro Beitrag vorkommt.

A.4 ERGEBNISSE

„Literatur“

Beitrag 01:

Tf: 0.0031847133757961785

Tf*idf (Themenkomplex):
0.007913715445184715

Tf*idf (Gesamter Korpus):
0.010121190542509382

„Ausstellung“

Beitrag 13:

Tf: 0.011673151750972763

Tf*idf (Themenkomplex): 0.0

Tf*idf (Gesamter Korpus):
0.019116055907402296

Beitrag 20:

Tf: 0.00423728813559322

Tf*idf (Themenkomplex): 0.0

Tf*idf (Gesamter Korpus):
0.0069390202940711725

A.4 ERGEBNISSE

„Konzert“

Beitrag 25:

Tf: 0.008097165991902834

Tf*idf (Themenkomplex): 0.0

Tf*idf (Gesamter Korpus):
0.012178764346366593

Beitrag 36:

Tf: 0.01639344262295082

Tf*idf (Themenkomplex): 0.0

Tf*idf (Gesamter Korpus):
0.024657006504529087

„Biografie“

Beitrag 41:

Tf: 0.00425531914893617

Tf*idf (Themenkomplex):
0.002293602130777391

Tf*idf (Gesamter Korpus):
0.009349891818451998

Beitrag 46:

Tf: 0.010169491525423728

Tf*idf (Themenkomplex):
0.005481320346434104

Tf*idf (Gesamter Korpus):
0.022344656718673417

A.4 ERGEBNISSE

„Tourismus“

Beitrag 49:

Tf: 0.0036101083032490976

Tf*idf (Themenkomplex): 0.0

Tf*idf (Gesamter Korpus):
0.006179482893698623

Beitrag 60:

Tf: 0.010238907849829351

Tf*idf (Themenkomplex): 0.0

Tf*idf (Gesamter Korpus):
0.017526110186565035

„Sport“

Beitrag 61:

Tf: 0.009900990099009901

Tf*idf (Themenkomplex): 0.0

Tf*idf (Gesamter Korpus):
0.015530850672414309

Beitrag 72:

Tf: 0.008130081300813009

Tf*idf (Themenkomplex): 0.0

Tf*idf (Gesamter Korpus):
0.012752974942388987

A.4 ERGEBNISSE

*Die Berechnung von tf sowie $tf*idf$ zeigt, dass die Relevanz von Begriffen nicht unbedingt von der Häufigkeit ihres Vorkommens im Text abhängt.*

*Bei meinem Beispiel lässt sich – aufgrund der relativ ähnlichen Länge der Texte – die Relevanz relativ schnell anhand der Anzahl der Nullen hinter dem Komma einschätzen. Bei nur einer Null hat der Begriff eine höhere Relevanz im Text oder im Korpus, bei mehr als einer Null sinkt die Relevanz. Keiner meiner Suchbegriffe kommt in einem Beitrag an erster Stelle, was die tf - und $tf*idf$ -Werte betrifft. An letzter Stelle kommen nur Ausstellung, Konzert, Tourismus und Sport bei der Berechnung des $tf*idf$ innerhalb ihres eigenen Themenabschnittes. Da all diese Begriffe in jedem Text ihres Abschnittes vorkommen, ergibt der $tf*idf$ -Wert hierfür immer 0.0. Ansonsten lassen sich die Suchbegriffe bei allen Berechnungen immer ins vordere bis hintere Mittelfeld einordnen. Ein konkretes Muster gibt es nicht.*

A.4 ERGEBNISSE

*Die Auswertung aller tf - und $tf*idf$ -Werte zeigt, dass die Höhe der Werte nicht unbedingt etwas mit der Häufigkeit im Text oder der Anzahl der Dokumente zu tun hat. Ausstellung, Konzert, Tourismus und Sport kommen zwar alle häufig vor, dies macht die Auswertung aber nicht unbedingt einfacher. Literatur und Biografie kommen – wie aufgezeigt – fast gar nicht oder nur in Teilen ihrer Texte vor. Es ändert nichts daran, dass auch die Dokumente, in denen sie vorkommen, hohe $tf*idf$ -Werte haben können.*

*Bezogen auf meine Fragestellung bedeutet es, dass eine pauschale Beantwortung anhand der $tf*idf$ -Werte relativ schwierig ist.*

A.4 ERGEBNISSE

Zusammenfassend lässt sich für Projektaufgabe A sagen, dass ich mit Hilfe von sechs Suchbegriffen 72 Texte zusammengetragen habe. Mit Hilfe meiner Indexierung und der Suchfunktion konnte ich feststellen, dass vier der Suchbegriffe jeweils in all ihren 12 Texten vorkommen. Bei den Begriffen Biografie, welcher nur in sieben seiner Texte vorkommt, und Literatur, wo es nur ein Text ist, spielt offensichtlich die Verschlagwortung und Klassifizierung in der Datenbank eine viel größere Rolle. Begriffe wie Konzert, Ausstellung oder Sport sind klarer umrissen und lassen sich so auch im Text selbst nutzen. Ein Text, den man mit Konzert, Ausstellung oder Sport klassifizieren würde, handelt demnach auch von einem Konzert, einer konkreten Ausstellung oder einem sportlichen Ereignis, weshalb das Wort genutzt wird. Literatur und Biografie sind per se schon unspezifizierter. Ein mit Literatur klassifizierter Text kann vieles beinhalten, eine Buchrezension, Informationen zu einer Gattung, zu einem Literaturfestival, zu einem Autor. So ist es auch bei meinem Korpus der Fall.

A.4 ERGEBNISSE

Genauso ist es auch mit Biografie. Mit diesem Begriff klassifizierte Texte können Porträts von Personen sein, runde Geburtstage und Jubiläen oder Rezensionen zu literarischen Biografien. Letzteres wäre eine Überschneidung mit Literatur. Beide Begriffe sind also sehr unspezifisch und werden deshalb im Text auch selbst nicht oft genutzt sondern eher bei der Verschlagwortung und Klassifizierung selbst.

*Auf die Relevanz der Begriffe in den Texten hat das wenig Einfluss, wie die Berechnung von tf sowie $tf*idf$ gezeigt haben. Hier gibt es viele Faktoren, die für die Werte relevant sind und Anzahl der Texte bzw. die Anzahl im Text ist nur ein Teil davon. In Zweifel können hier Begriffe, die nicht so oft vorkommen, sogar höhere Werte erreichen.*

A.4 ERGEBNISSE

Man wird zwar auch mit Begriffen wie Literatur und Biografie Texte finden, es werden aber nicht so viele sein wie mit möglichen anderen Suchbegriffen. Und aufgrund der schwierigen Eingrenzung ist immer die Frage, ob die gefundenen Texte das beinhalten, wonach man sucht.

Meine Erkenntnis aus den gewonnenen Ergebnissen ist, dass ich mir künftig mehr Gedanken über die Auswahl meiner Suchwörter machen werde, erst recht, wenn es sich nicht um eine Datenbank mit verschlagworteten Datensätzen handelt. Anstatt Literatur hätte ich beispielsweise „Rezension“ als Suchbegriff verwenden können und anstatt Biografie „Porträt“. Das wäre eine bessere Eingrenzung gewesen.

A.4 ERGEBNISSE

Mit der Funktion meiner Indexierung bin ich sehr zufrieden. Die zwölf verschlagworteten Texte lassen sich mit der Suchfunktion problemlos finden, das Information-Retrieval ist gut. Gleiches gilt für die fünf häufigsten Wörter im Korpus. Auch die anderen Funktionen, das Einfügen der Stopwörter, der Thesaurus sowie das Anzeigen der Worthäufigkeiten, haben keine Probleme bereitet.

*Ohne Probleme funktioniert hat auch Erstellung des Notebooks für die Berechnung von $tf*idf$, wobei diese mit einiger Fleißarbeit verbunden war. Auch musste ich mir etwas einfallen lassen, um das $tf*idf$ einerseits nur für die Themenabschnitte und andererseits auch für den ganzen Korpus berechnen zu können. Dies ist mir aber gelungen. Wie schon dargelegt, habe ich allerdings bei der Auswertung festgestellt, dass mir die Werte hieraus nicht so weitergeholfen haben, wie ich zu Beginn gedacht habe. Sie haben einiges über das Verhältnis von Relevanz und Häufigkeit ausgesagt, aber für die Beantwortung meiner Frage haben mir die Funktionen der Suchmaschine mehr weitergeholfen.*

A.5 SCHWIERIGKEITEN

Um auftretende Schwierigkeiten zu lösen war am wichtigsten, zu lernen, die Fehlermeldungen richtig zu interpretieren, was mir mit etwas Übung in vielen Fällen gelungen ist. So konnte ich Schreibfehler im Code oder auch andere Gründe dafür, weshalb der Code nicht funktioniert, ausfindig machen. Häufig haben Kleinigkeiten dazu geführt, weil ich bspw. Änderungen am Thesaurus vorgenommen habe, und ich dann vergessen habe, diesen nochmal neu im Notebook zu öffnen.

*Eine Schwierigkeit bestand darin, wie ich das Notebook für die tf*idf-Berechnung erstelle. Durch den Aufbau meines Korpus, bestehend aus einzelnen Themenabschnitten, wollte ich die Werte auch auf die einzelnen Themen bezogen errechnen lassen. Hier ist es mir gelungen, anhand der Notebook-Vorlage und etwas logischem Nachdenken herauszufinden, dass ich die einzelnen Textabschnitte auch im Code durch Klammern definieren muss, damit es funktioniert.*

A.6 OFFENE FRAGEN

Ein paar Punkte sind bei mir aufgetreten, die ich nicht lösen konnte:

*Bei der Anzeige von $tf*idf$ im entsprechenden Notebook wurden mir immer die Werte aller Texte zweimal angezeigt. Das Problem hat nicht die Umsetzung behindert, aber lösen konnte ich es auch nicht.*

*Außerdem wurden die Werte für tf sowie $tf*idf$ immer für die „Dokumente 1-12“ angezeigt, hier hätte ich es auch gerne so gehabt, dass die angezeigte Zahl mit meiner Dokumentennummer übereinstimmt.*

*Die Umsetzung von $tf*idf$ mit einem großen Korpus ist – mit meinen aktuellen Kenntnissen – relativ umständlich und besteht aus viel Fleißarbeit, weil man alle Beiträge einzeln aufführen muss. Hierfür gibt es sicherlich Möglichkeiten, die mir noch nicht bekannt sind.*

*Auch die Auswertung bei $tf*idf$ ist relativ umständlich, weil man alle Listen einzeln durchsehen muss. Hier wäre es gut, wenn es die Möglichkeit gäbe, die Listen nicht nur nach Werten, sondern auch alphabetisch sortiert auszugeben.*

MEIN KLASSIFIKATIONS- PROJEKT



B.1 PROBLEMSTELLUNG KLASSIFIKATION

Für mein Klassifikationsprojekt möchte ich das statische Verfahren mit dem lernenden Verfahren vergleichen.

Für diesen Zweck dienen meine sechs Suchwörter, auf deren Grundlage ich den Korpus zusammengestellt habe, als Kategorien. Ich möchte beide Klassifikationssysteme darin vergleichen, ob sie die Radiobeiträge den richtigen Kategorien zuordnen.

Dies ist insofern auch eine Fortführung meiner Fragestellung aus der Indexierungsaufgabe, bei welcher ich die Relevanz meiner sechs Suchbegriffe untersucht habe. Nun überprüfe ich, ob die Begriff dafür geeignet sind, als Kategorie zu dienen.

B. 2. DATEN

Für die Klassifikation nutze ich zum großen Teil wieder den selben Korpus aus Aufgabe A.

Gerade für die statische Klassifikation ist es allerdings interessant, noch weitere Texte außerhalb des Korpus hinzuzuziehen, da die Wortlisten für die Einordnung auf Grundlage der Korpus Texte erstellt wurden und es langweilig wäre, nur die selben Texte damit wieder einzuordnen.

Für das lernende Verfahren besteht das Problem weniger, da hier für den Trainingsbestand nur ein Teil der Texte benötigt wird.

B. 2. DATEN

Da ich beim statischen Klassifikationsverfahren nur zwei Kategorien, nämlich „Literatur“ und „Biografie“ miteinander vergleiche, habe ich zur Überprüfung der Klassifikation zwei weitere, kleine Textsammlungen erstellt, „literatur_klassifikation“ und „biografie_klassifikation“. Ähnlich, wie beim regulären Korpus, habe ich mit den Suchwörtern Literatur und Biografie jeweils 10 weitere Texte in der NDR-Hörfunkdatenbank herausgesucht und aus diesen zwei kleine Corpora gebildet.

- beitraege
- biografie_klassifikation
- literatur_klassifikation

- literatur 01.txt
- literatur 02.txt
- literatur 03.txt
- literatur 04.txt
- literatur 05.txt
- literatur 06.txt
- literatur 07.txt
- literatur 08.txt
- literatur 09.txt
- literatur 10.txt

- biografie 01.txt
- biografie 02.txt
- biografie 03.txt
- biografie 04.txt
- biografie 05.txt
- biografie 06.txt
- biografie 07.txt
- biografie 08.txt
- biografie 09.txt
- biografie 10.txt

B. 3 VORGEHENSWEISE

Für das statische Klassifikationsverfahren habe ich mich aufgrund der Funktionsweise des Verfahrens dazu entschieden, nur zwei Kategorien miteinander zu vergleichen, nämlich Literatur und Biografie.

Für diese beiden Begriffe habe ich mich entschieden, da sie gewisse Überschneidungen haben können, was die Auswertung interessanter gestaltet. Eine Literaturkritik über eine Biografie könnte beispielsweise in beide Kategorien passen.

Andere Kategorien, wie etwa Biografie und Tourismus, wären sehr gegensätzlich für einen Einzelvergleich zweier Kategorien.

Im Rahmen der statischen Klassifikation habe ich dann sowohl die Zuordnung der Texte aus den Themenblöcken „Literatur“ und „Biografie“ aus dem regulären Korpus überprüft wie auch die Texte aus den beiden zusätzlichen Corpora „literatur_klassifikation“ und „biografie_klassifikation“.

B. 3 VORGEHENSWEISE

Für die Zuordnungslisten des statischen Verfahrens habe ich mir für die Themenblöcke Literatur und Biografie jeweils die häufigsten Wörter anzeigen lassen. In die Listen habe ich diejenigen mit aufgenommen, die pro Themenblock mindestens sieben Mal vorkommen.

Eigennamen wurden dabei nicht berücksichtigt.

2) Listen für die Zuordnung zu den Kategorien Literatur und Biografie

```
In [96]: # Liste Wörter Literatur. In die Liste wurden nur Wörter aufgenommen,  
# die in den Beitragstexten zum Thema Literatur mindestens 7 Mal vorkommen, aber keine Eigennamen von Personen darstellen.  
literatur_words=['roman', 'buch', 'ddr', 'vater', 'autor', 'geschichte', 'dorf', 'spielen', 'sohn', 'fast', 'erzählt', 'liebe', 'nacht']  
print(literatur_words)
```

```
['roman', 'buch', 'ddr', 'vater', 'autor', 'geschichte', 'dorf', 'spielen', 'sohn', 'fast', 'erzählt', 'liebe', 'nacht']
```

```
In [99]: # Liste Wörter Biografie. In die Liste wurden nur Wörter aufgenommen,  
# die in den Beitragstexten zum Thema Biografie mindestens 7 Mal vorkommen und keine Eigennamen von Personen darstellen.  
biografie_words=['biografie', 'später', 'tagesthemen', 'deutschland', 'frau', 'niemand', 'lassen', 'familie', 'ab']  
print(biografie_words)
```

```
['biografie', 'später', 'tagesthemen', 'deutschland', 'frau', 'niemand,lassen', 'familie', 'ab']
```


B. 3 VORGEHENSWEISE

Nach Berechnung wird die Kategorie mithilfe einer if/else-Klausel zugeordnet.

Zuordnung des Beitrages

Mit Hilfe einer if/else-Klausel werden die berechneten Zahlen zugeordnet.

```
In [18]: # if neg < pos erzielt kein Ergebnis. Also müssen wir noch weitere Bedingungen prüfen.  
if literatur > biografie:  
    print("Der Beitrag ist vermutlich aus der Kategorie Literatur.")  
elif biografie > literatur:  
    print("Der Beitrag ist vermutlich aus der Kategorie Biografie.")  
else: # hier auch eingerückt die else-Klausel, wenn keine der beiden Bedingungen zutrifft  
    print("Eindeutig zuordnen kann man den Beitrag nicht.")
```

Der Beitrag ist vermutlich aus der Kategorie Literatur.

B. 3 VORGEHENSWEISE

Beim dynamischen und lernenden Klassifikationsverfahren habe ich die Zuordnung mit allen sechs Kategorien getestet. Dies habe ich auf Grundlage der Bibliothek TextBlob durchgeführt. Zwölf Beiträge meines Korpus habe ich für den Trainingsbestand genutzt. Anschließend habe ich die Klassifizierung der anderen 60 Beiträge überprüft. Gleiches habe ich mit den Beiträgen aus den beiden kleinen Corpora `literatur_klassifikation` und `biografie_klassifikation` unternommen.

B. 3 VORGEHENSWEISE

Für die dynamische und lernende Klassifikation habe ich aus 12 Beiträgen (je zwei aus jedem Themenkomplex) einen Trainingsbestand gebildet. Es handelt sich um die 12 Texte, die ich auch zu Beginn des Projektes gesichtet und verschlagwortet habe. Die Schlagworte habe ich natürlich thematisch angepasst, damit die thematische Klassifikation funktioniert.

```
train = [  
    ('roman krimi fiktiv', 'literatur'),  
    ('heimatroman roman gegenwartsliteratur', 'literatur'),  
    ('ausstellung bestaunen besucher wanderausstellung', 'ausstellung'),  
    ('ausstellung museum besuch fotos', 'ausstellung'),  
    ('symphoniker tournee orchester musiker,', 'konzert'),  
    ('philharmoniker musikfestival orchester', 'konzert'),  
    ('karriere schauspieler frau ausnahmefigur', 'biografie'),  
    ('porträt biografie moderatorin', 'biografie'),  
    ('urlauber urlaub tourismus', 'tourismus'),  
    ('jugendherberge klassenfahrt reise tourismus', 'tourismus'),  
    ('liga football sport', 'sport'),  
    ('fußball liga fußballverband sport', 'sport')
```

B. 3 VORGEHENSWEISE

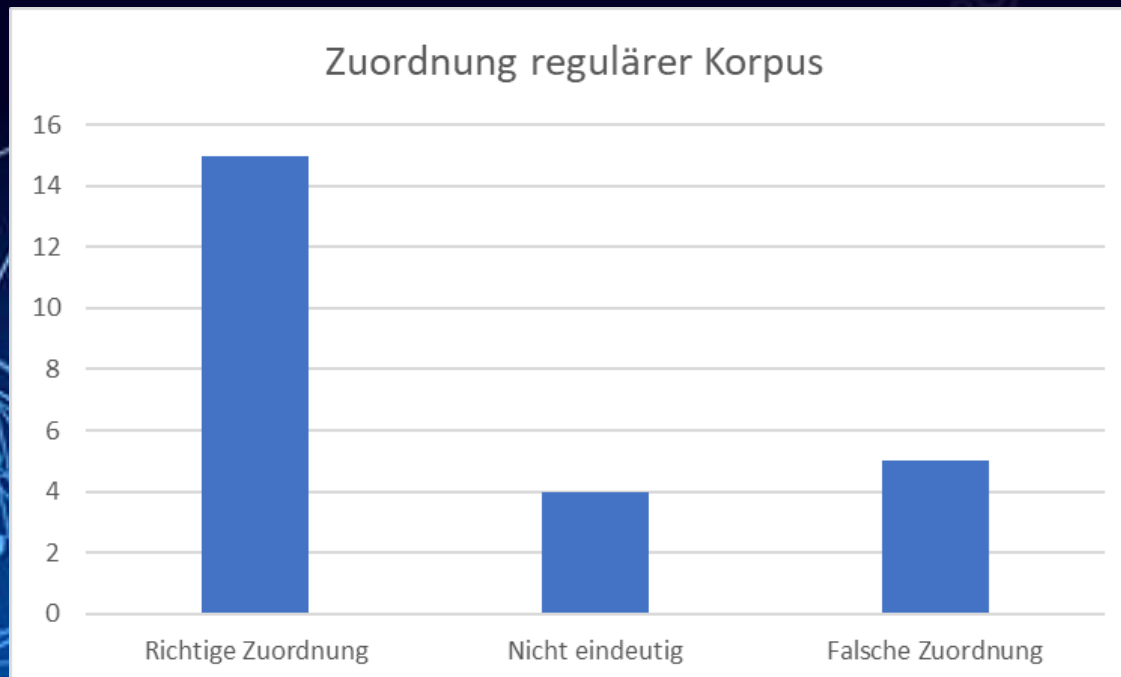
Der Vollständigkeit halber habe ich im Notebook die selbe Klassifikation noch einmal über eine externe CSV-Datei mit den Trainingsdaten eingebaut, die ich mit Excel erstellt habe.

```
#Öffnen der Datei mit den Trainingsdaten, die im Input-  
with open('input/klassifikation.csv','r') as file:  
    cl2= NaiveBayesClassifier(file, format="csv")
```

B. 4 ERGEBNISSE

Beim statischen Verfahren hat es folgendes Ergebnis gegeben:

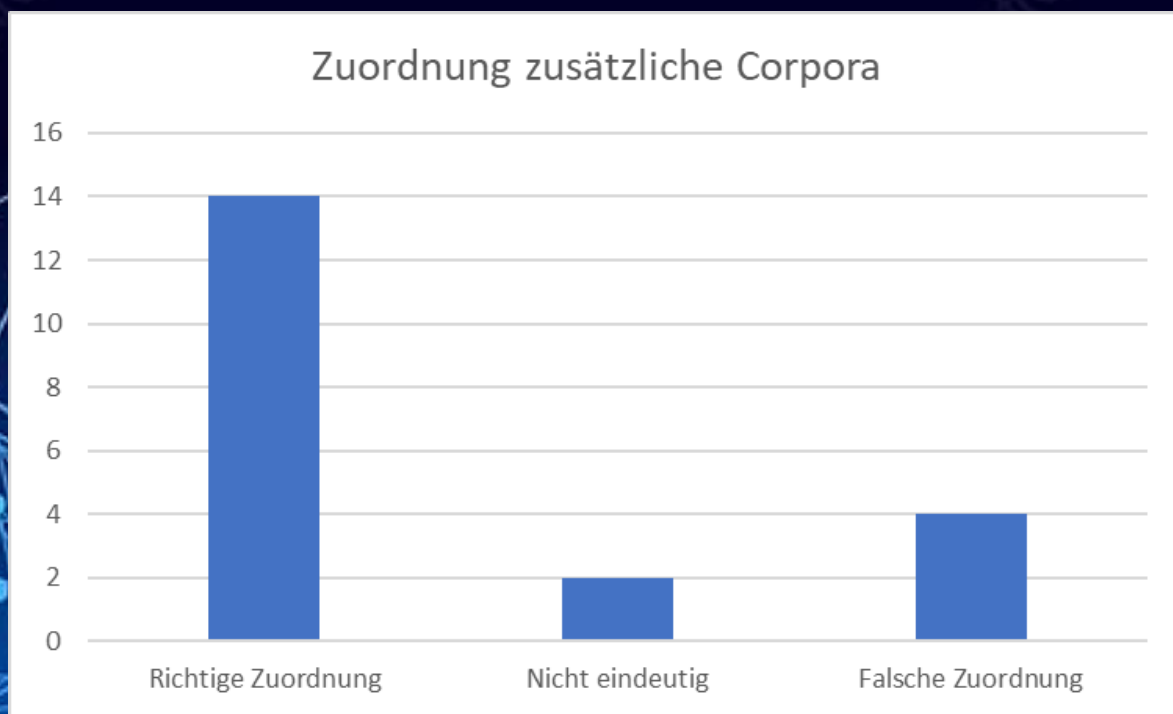
Von den 24 Texten aus dem regulären Korpus (je 12 aus den Themenblöcken Literatur und Biografie) wurden insgesamt 16 richtig zugeordnet. Vier Beiträge konnte das System nicht eindeutig zuordnen, davon einen aus dem Themenbereich Literatur und drei aus dem Bereich Biografie. Falsch zugeordnet wurden fünf Beiträge, davon einer aus dem Bereich Literatur und vier aus dem Bereich Biografie. Bei der nicht eindeutigen und falschen Zuordnung überwiegt also der Bereich Biografie sehr eindeutig.



B. 4 ERGEBNISSE

Beim statischen Verfahren hat es folgendes Ergebnis gegeben:

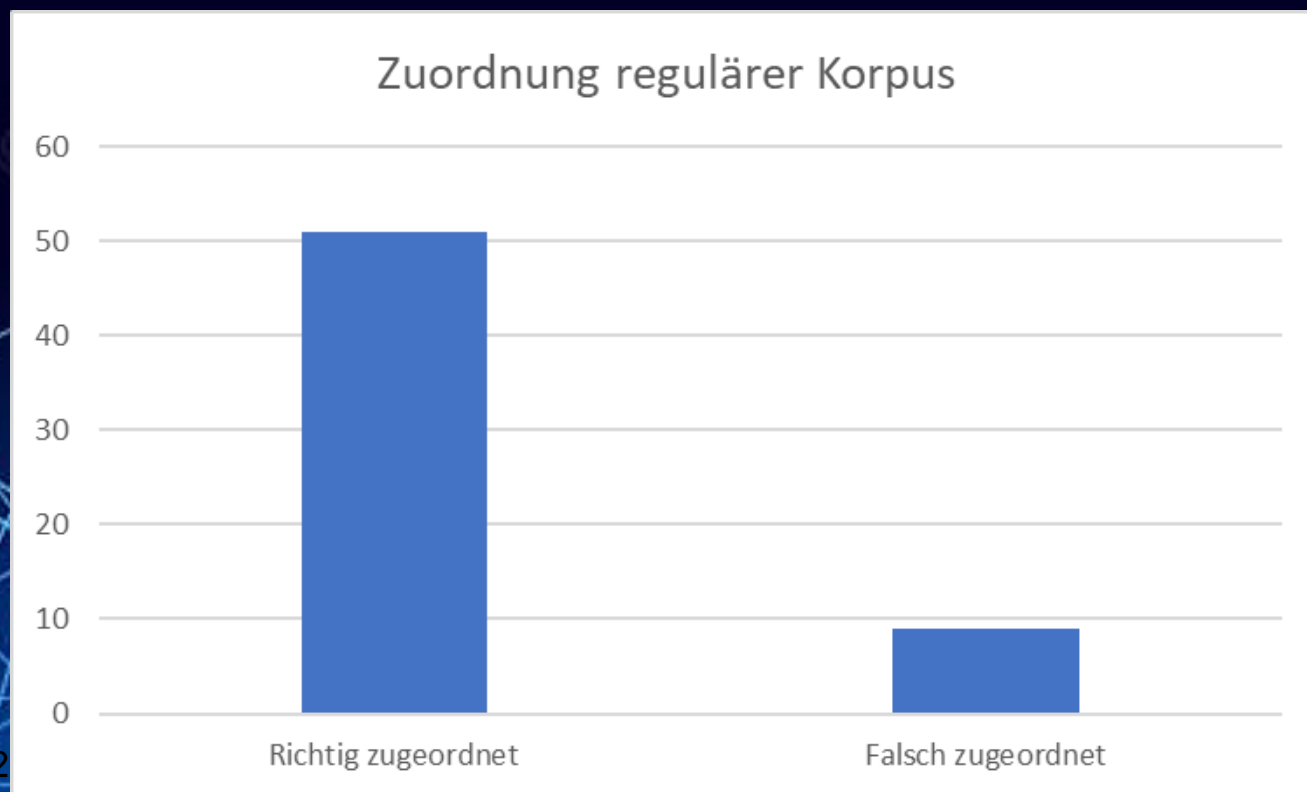
Von den 20 Texten aus den beiden neuen Corpora „literatur_klassifikation“ sowie „biografie_klassifikation“ wurden insgesamt 14 richtig zugeordnet. Zwei Texte konnten nicht eindeutig zugeordnet werden, jeweils einer aus dem Korpus „literatur_klassifikation“ und dem Korpus „biografie_klassifikation“. Vier Beiträge wurden falsch zugeordnet, alle aus dem Korpus „biografie_klassifikation“. Auch hier dominiert bei den falschen und nicht eindeutigen Texten der Bereich Biografie.



B. 4 ERGEBNISSE

Beim dynamischen Verfahren hat es folgendes Ergebnis gegeben:

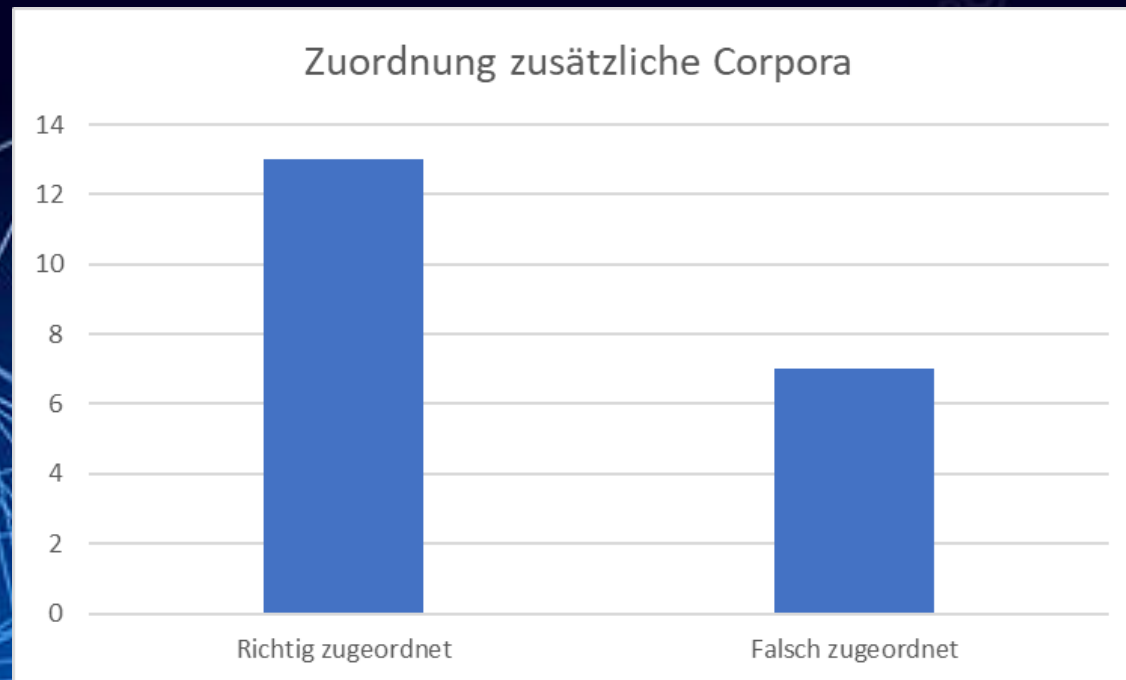
Von den 60 überprüften Dokumenten wurden 51 richtig zugeordnet, 9 Dokumente wurden falsch eingeordnet. Die meisten falsch zugeordneten Texten kamen aus den Themenbereichen Literatur und Konzert, fast alle falsch zugeordneten Texte wurden vom System der Kategorie Biografie zugeordnet.



B. 4 ERGEBNISSE

Beim dynamischen Verfahren hat es folgendes Ergebnis gegeben:

Von den 20 Texten aus den Corpora `literatur_klassifikation` und `biografie_klassifikation` wurden insgesamt 13 richtig zugeordnet, 7 falsch. Die Verteilung unter den beiden kleinen Corpora ist allerdings ungleich. Während das System bei den Biografie-Texten 8 richtig und nur zwei falsch eingeordnet hat, waren es bei den Literatur-Texten nur 5, die richtig eingeordnet wurden, 5 wurden falsch eingeordnet. Interessant ist, dass alle falsch eingeordneten Texte des Korpus `literatur_klassifikation` der Klasse `Biografie` zugeordnet wurden.



B. 4 ERGEBNISSE

Vergleicht man beide Klassifikationsprogramme miteinander, so lässt sich feststellen, dass beim statischen Verfahren von insgesamt 44 Texten aus den drei Corpora 29 richtig zugeordnet wurden, was einem Anteil von 65,9 Prozent entspricht.

Beim dynamischen, lernenden Verfahren wurden von 80 Texten aus den drei Corpora 64 Texte richtig klassifiziert wurden, dies entspricht einem Anteil von 80 Prozent.

Grundsätzlich hat also das dynamische Verfahren besser gearbeitet. Einschränkend gilt natürlich, dass beide Verfahren unterschiedlich angewendet wurden, auf Grundlage unterschiedlicher Wortlisten gearbeitet haben und auch die Zusammenstellung der Wortlisten subjektiv erfolgt ist.

B. 4 ERGEBNISSE

Sowohl beim statischen wie beim dynamischen Verfahren ist mir aufgefallen, dass die Kategorien Literatur und Biografie relativ schwer fassbar sind, Biografie noch schwerer als Literatur. Dies stützt auch die Erkenntnisse der Projektaufgabe A, laut der beide Begriffe weniger Relevanz für die Texte haben, als die anderen.

Beim statischen Verfahren waren es insbesondere Texte der Kategorie Biografie, die falsch zugeordnet wurden, hier gab es auch keinen Unterschied zwischen dem regulären Korpus und den beiden Ergänzungscorpora.

Beim dynamischen Verfahren wurden insbesondere Texte der Kategorie Literatur falsch zugeordnet, dies zeigte sich auch in den beiden Ergänzungscorpora „literatur_klassifikation“ und „biografie_klassifikation“. Fast alle falsch zugeordneten Texte ordnete das System der Klasse Biografie zu, was die Uneindeutigkeit dieser Kategorie noch einmal aufzeigt.

Diese Erkenntnis wird beim dynamischen System dadurch gestützt, da hier ja auch die Texte aus den anderen Kategorien klassifiziert wurden, bis auf ein paar Ausnahmen der Kategorie Konzert alle richtig.

B.5 SCHWIERIGKEITEN

Beim Klassifikationsprojekt bin ich auf die Schwierigkeit gestoßen, dass ich beim dynamischen Verfahren ursprünglich einen deutlich größeren Trainingsbestand nutzen wollte, den ich auch in einer CSV-Datei angelegt habe. Hierfür habe ich mir für jeden Text, der für den Trainingsbestand vorgesehen war, die häufigsten Wörter anzeigen lassen und diejenigen ausgewählt, die zur Klassifikation passen. Bei der praktischen Anwendung der Liste gab das System nur eine Fehlermeldung aus. Zwar konnte ich ein bis zwei Fehler finden, die ich beim Erstellen der Liste mit Anführungszeichen und Komma gemacht habe, allerdings ließ sich das Grundproblem so nicht lösen. Da es bei vorherigen Versuchen bereits mit kleineren Listen funktioniert hatte, griff ich erneut auf diese kleinere Variante zurück, was auch geklappt hat, sowohl im Notebook selbst als auch in der CSV-Datei.

Für den künftigen Umgang mit Python würde ich dieses Problem gerne noch lösen.

3. LESSONS LEARNED

Durch das Projekt bin ich mit einer weiteren Programmiersprache in Berührung gekommen. Das Thema Python fand ich sehr interessant, insbesondere die Möglichkeiten aus Aufgabe A, Texte inhaltlich auszuwerten. Ich hatte lange Zeit mit dem Thema Programmieren noch keine Berührungspunkte und habe jetzt im Studium bereits mehrere Möglichkeiten kennengelernt. Insbesondere, was die Möglichkeiten der automatischen Textauswertung angeht, kamen mir direkt Ideen, wie man mit Hilfe von Python historische Quellen etc. auswerten könnte.

Beim Klassifizieren fand ich die Unterschiede zwischen beiden Systemen interessant. Es war aufschlussreich, zu welchen unterschiedlichen Ergebnissen beide Systeme gekommen sind.

Die Erkenntnisse aus beiden Projekten haben mich dazu gebracht, mehr über die Funktion von Recherchemitteln und die Bedeutung von Suchbegriffen nachzudenken.

3. LESSONS LEARNED

Bei künftigen Projekten würde ich vor allem die Erfahrungen dieses Projektes nutzen und alles mehr vom Ende her denken und mir Gedanken machen, was eigentlich ein realistisches Ergebnis sein könnte und mit welchen Mitteln dieses zu erreichen ist. Entsprechende Methoden zur Umsetzung würde ich mir teilweise schon im Vorfeld anschauen, um überhaupt realistisch einschätzen zu können, ob ich das Projekt so umsetzen kann.

Dieses Mal fiel mir die Findung der richtigen Fragestellung noch schwer, weil ich auch schlecht einschätzen konnte, wofür die Kenntnisse tatsächlich ausreichen.

Jetzt würde es mir viel einfacher fallen, die Erstellung des Korpus, des Thesaurus sowie der Stopwortliste richtig einzuschätzen.

Mir hat das Thema Python in jedem Fall Spaß gebracht und ich werde es weiter verfolgen und mir hoffentlich neue Kenntnisse aneignen.