HAW HAMBURG

BACHELORARBEIT

Konsolidierung von kontrollierten Vokabularen im Text Mining

vorgelegt im Dezember 2017 von

Xiaoyu Shi
Matr.-Nr.: 2214114

1. Prüferin: Prof. Dr. Susanne Glissmann-Hochstein
2. Prüfer: Prof. Dr. -Ing. Maika Büschenfeldt

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Department Information
Studiengang Bibliotheks- und Informationsmanagement

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Konsolidierung von kontrollierten Vokabularen im Text Mining

Bachelorarbeit vorgelegt von
Xiaoyu Shi

Abstract

Diese Arbeit beschäftigt sich mit der Frage, wie mit der Python-Bibliothek „Pandas“ die kontrollierten Vokabulare des Text-Mining-Projekts mit dem Titel „Schaffung von mehr Transparenz in der Bundestagswahl 2017“, das von Prof. Dr. Susanne Glissmann im Wintersemester 2017 am Department Information an der Hochschule für Angewandte Wissenschaften (HAW) geleitet wurde, konsolidiert werden können.

Nach der Beschreibung der theoretischen Grundlagen zu „Korpus“, dem kontrollierten Vokabular der Python-Bibliothek Pandas, und der Konsolidierung wird das Vorgehen zur Konsolidierung kontrollierter Vokabulare vorgestellt. Es gibt insgesamt 19 kontrollierte Vokabulare-Excel-Dateien, die zusammengeführt und aufbereitet werden. Der dabei genutzte Code wird mit der Python-Bibliothek „Pandas“ in der interaktiven Entwicklungsumgebung Jupyter Notebook erstellt. Das finale konsolidierte kontrollierte Vokabular ist im politischen Bereich nützlich und wertvoll. Die dazu entwickelten Skripte sind zur Analyse von Massendaten anderer Text-Mining-Projekte wiederverwendbar. Für die Nutzung der Python-Bibliothek „Pandas“ ist keine Programmiererfahrung erforderlich.

Schlagwörter: kontrolliertes Vokabular – Python – Pandas – Jupyter Notebook – Konsolidierung

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

BMBF	Bundesministerium für Bildung und Forschung
DSL	Domänenspezifische Sprachen
GPL	General Public License
HAW	Hochschule für Angewandte Wissenschaften
HGB	Handelsgesetzbuch
HTML	Hypertext Markup Language
IE	Informationsextraktion
IFRS	International Financial Reporting Standards
IR	Information Retrieval
MUI	Medien und Information
NISO	National Information Standards Organization
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OCR	Optical Character Recognition

1 Einleitung

Wie kann man sich in der ständig anwachsenden Zahl von Informationen zurechtfinden, um Entscheidungen für eigene Frage zu treffen? Mit einer enormen Informationsflut kämpfen heute viele Menschen, weil es im Internet schwierig ist zu unterscheiden, welche Informationen nützlich sind. Man erhält viele aktuelle Informationen durch Onlinezeitungen, Facebook, Twitter und viele andere Plattformen. Anhand der Überschriften ist es nicht immer leicht zu beurteilen, ob die Informationsquelle relevant ist, insbesondere dann, wenn man nach einem bestimmten Thema, zum Beispiel im Bereich der Politik, recherchiert.

Früher erhielt man wichtige Feedbacks zu einem bestimmten Thema durch Umfragen. Die Informationen wurden manuell verarbeitet und analysiert. Das war zeitaufwendig und brauchte viel Personal und Geld. Dabei bestand das Risiko, relevante Informationen zu verlieren. Heute erhält man persönliche Feedbacks zum Teil auch noch durch Umfragen. Ein großer Vorteil im Vergleich zu früher ist jedoch, dass die Daten heutzutage automatisch gespeichert und verarbeitet werden. Wie jedoch werden unstrukturierte Daten aus Onlinequellen verarbeitet und analysiert?

Bis September 2017 nutzten mehr als 31 Millionen Menschen in Deutschland den Social-Media-Kanal Facebook. Diese Zahl stammt von Facebook selbst (Roth 2017). Die Zahl der Nutzer ist von Februar 2016 bis September 2017 um über 3 Millionen gestiegen. Mit den steigenden Nutzerzahlen erhöhte sich auch die Anzahl der auf Facebook veröffentlichten Nachrichten, Bewertungen und Kommentare.

Auf dem Social-Media-Kanal Twitter hingegen gibt es mit über 500 Millionen Tweets täglich (Twitter 2017). Seit langer Zeit sind Social-Media-Kanäle wie Facebook, Twitter und andere Anwendungen nicht nur reine Kommunikationsplattformen, sondern werden als ein Ort wahrgenommen, an dem man viele aktuelle Informationen erhält.

Eine wichtige Frage ist also, welche Nachrichten und Kommentare im Social-Media-Kanal seriös und wertvoll sind. Wie kann ein Nutzer erkennen, dass eine Nachricht wertvoll und seriös ist? Zum Lösen dieses Problems werden heutzutage viele Tools angeboten, zum Beispiel das Tool „Natural Language Processing“ (NLP) und „Optical

Character Recognition“ (OCR), eine automatische Texterkennung, die es ermöglicht, gewünschte Informationen herauszufiltern.

Heutzutage ermöglicht eine gängige Text-Mining-Methode, große Mengen an Informationen aus verschiedenen Quellen zu extrahieren (Krapp 2003). Eine Projektgruppe aus dem Sommersemester 2017 am Department Information der HAW-Hamburg unter Leitung von Prof. Dr. Susanne Glissmann hat ein Projekt mit dem Titel „Schaffung von mehr Transparenz in der Bundestagswahl 2017“ durchgeführt, um onlinepolitische Artikel aus verschiedenen Datenquellen zu analysieren. Im Projekt wurde die Text-Mining-Methode genutzt, wobei wertvolle kontrollierte Vokabulare aus der Entwicklungsumgebung Jupyter der Programmiersprache Python erworben wurden. Im Projekt haben 19 Gruppen je ein eigenes kontrolliertes Vokabular erstellt. Die kontrollierten Vokabulare wurden zum Schluss im Excel gespeichert. Das Projektergebnis waren insgesamt 19 Excel-Dateien.

Anschließend stellt sich die Frage, wie die durch die Text-Mining-Methode erworbenen kontrollierten Vokabulare aus 19 verschiedenen Gruppen durch eine Programmiersprache automatisch zusammengeführt und verarbeitet werden können, um schließlich ein hochqualitatives konsolidiertes kontrolliertes Vokabular zu erhalten. Diese Arbeit beschäftigt sich mit exakt dieser Fragestellung und fokussiert auf die Vorgehensweise zur Konsolidierung kontrollierter Vokabulare.

1.1 Stand der Forschung

Ein bekanntes Projekt in der Politikwissenschaft, bei dem die Text-Mining-Methode angewandt wird, ist das „epol: Postdemokratie und Neoliberalismus“-Projekt, das vom Bundesministerium für Bildung und Forschung (BMBF) gefördert wurde (Lemke 2015, S. 5). Das zeigt, dass Text-Mining-Verfahren im politischen Bereich bereits angewendet wurden. Dabei spielt die qualitative und die quantitative Perspektive der Textanalyse zum methodischen Vorgehen eine wichtige Rolle. Dazu gibt es bereits alte und neue Fachliteraturen, beispielweise von Lemke (2015) und Gentsch (1999).

Weitere bekannte durchgeführte Text-Mining-Projekte wie die „Analyse des Firmenbildes und der externen Kommunikation” in der Branche der Versorgungsunternehmen, die „Dokument-Analyse“ bei Behörden, die „Meinungserhebung im Internet“ bei Onlineservices und weitere zeigen, wie die Text-Mining-Methode weltweit genutzt und

eingesetzt wird (Gentsch 1999, S. 190). Einige Text-Mining-Projekte wurden bereits in den Jahren 2016 und 2017 von der HAW durchgeführt. Einige davon sind die Projekte „Digital Profil“ und „Schaffung von mehr Transparenz in der Bundestagswahl 2017“. Durch diese Projekte haben die Studierenden einen Überblick über Text-Mining-Methoden erhalten.

In der Betriebswirtschaft bezieht sich Konsolidierung meistens auf den Jahresabschluss (Hans-Böckler-Stiftung 2011, S. 5). Jedoch gibt es verschiedene Methoden der Konsolidierung in der Betriebswirtschaft, die in der Literatur bereits von Sachse (2015) erläutert wurden. Konsolidierung kann auch einen zeitlichen Vergleich zwischen umfangreichen Materialen im Archiv ermöglichen (Parma 2015, S. 1). In vielen Branchen sind mit Konsolidierungen auch Fusionen und Übernahmen gemeint, was auf Englisch Merges and Acquisitions (M&A) heißt (Hans und Neumair 2010, S. 446).

Allerdings ist für die Konsolidierung kontrollierter Vokabulare im Text Mining kaum Literatur zu finden. Jedoch gibt es für die Datenanalyse zur Zusammenführung von Daten heutzutage viele kommerzielle und Open-Source-Tools wie Python, R, MATLAB und SAS (McKinney 2015, S. 2). Es gibt bereits viel Literatur zur Datenanalyse mit der Python-Bibliothek Pandas, beispielweise von McKinney (2015, 2012) und andere Literatur, darüber hinaus gibt es Beispiele zum Umgang mit Pandas auf der Webseite GitHub.

Die Python-Bibliothek Pandas bietet umfassende Funktionen zur Auswertung und Verarbeitung tabellarischer Daten. Mit der Python-Bibliothek Pandas kann man einfach und schnell gut strukturierte Daten verarbeiten (McKinney 2015, S. 5). Pandas ermöglicht es mit der individuellen Zielsetzung, die strukturierten Daten zusammenzuführen. Das Handbuch mit dem Titel „Internet-Suchmaschinen“ von Lewandowski gibt einen Hinweis darauf, was man vor einer Konsolidierung unterschiedlicher Daten machen soll, zum Beispiel Indexierungen definieren (Lewandowski 2011, S. 144).

Diese Arbeit versucht, die Konsolidierung mehrerer kontrollierter Vokabulare im Text Mining durch Anwendung der Python-Bibliothek Pandas zu realisieren.

1.2 Zielsetzung der Arbeit

Diese Arbeit basiert auf dem Endergebnis des Projekts „Schaffung von mehr Transparenz in der Bundestagswahl 2017“, das von Prof. Dr. Susanne Glissmann im Wintersemester 2017 am Department Information an der Hochschule für Angewandte Wissenschaften geleitet wurde. Ziel des Projekts war es, ein kontrolliertes Vokabular zur Beschlagwortung verschiedenster politische Beiträge zu entwickeln (Glissmann 2017, S. 3). Das Projekt wurde Ende Mai 2017 mit einem positiven Ergebnis abgeschlossen.

Ziel dieser Bachelorarbeit ist die Konsolidierung der kontrollierten Vokabulare, die seitens der Studierenden der 19 Gruppen des Projekts resultieren. Hierbei sollen entsprechende Vorgehensweisen im Projekt vorgestellt werden. Das Hauptaugenmerk dieser Arbeit liegt auf der Zusammenführung und Aufbereitung kontrollierter Vokabulare. Zu erwarten ist, dass sich die kontrollierten Vokabulare aus den 19 Excel-Dateien durch die Python-Bibliothek Pandas zusammenführen lassen. Anschließend wird die zusammengeführte Datei von Homonymen, Synonymen und Ausdrücken von Begriffen bereinigt, um ein kontrolliertes Vokabular mit höherer Qualität zu entwickeln.

Darüber hinaus soll erreicht werden, dass das entwickelte Verfahren zur Konsolidierung kontrollierter Vokabulare nicht nur für dieses Projekt, sondern in Zukunft auch für die Datenverarbeitung anderer Projekte angewendet werden kann.

1.3 Aufbau der Arbeit

In der vorliegenden Bachelorarbeit wird zunächst ein theoretischer Rahmen gesetzt, gefolgt von einem praktischen Teil. Diese Bachelorarbeit ist insgesamt in fünf Kapitel untergliedert.

Das erste Kapitel führt in die aktuelle Forschung, die Zielsetzung dieser Arbeit und den Aufbau dieser Arbeit ein.

Das zweite Kapitel stellt eine theoretische Einführung dar. Es werden wichtige Definitionen bezüglich wesentlicher Elemente dieser Arbeit erklärt, die zum Verständnis der Arbeit erforderlich sind.

Das dritte Kapitel stellt eine Beschreibung des Projekts „Schaffung von mehr Transparenz in der Bundestagswahl 2017“ dar. In diesem Kapitel werden die Ablaufprozesse des Projekts erläutert.

Das vierte Kapitel beinhaltet den Praxisteil. Hierin wird das Vorgehen zur Konsolidierung kontrollierter Vokabulare in einzelnen Schritten erläutert. Die dazu angewendeten kontrollierten Vokabulare sind die Ergebnisse des Projekts „Schaffung von mehr Transparenz in der Bundestagswahl 2017“. Es gibt insgesamt 19 kontrollierte Vokabulare, die am Ende in einer Datei zusammengeführt und ausgewertet werden. Als Programmiersprache zur Konsolidierung der kontrollierten Vokabulare wird sich in dieser Arbeit auf Python beschränkt. Die Skripte zur Konsolidierung wurden mit der Python-Bibliothek Pandas in der Entwicklungsumgebung Jupyter Notebook erstellt, und sind im Anhang dieser Arbeit beigefügt. Am Ende dieses Kapitels wird ein Ergebnis zur Konsolidierung der kontrollierten Vokabulare wiedergegeben.

Im fünften Kapitel erfolgen eine Zusammenfassung dieser Arbeit und ein Ausblick auf die zukünftige Weiterentwicklung.

2 Theoretische Grundlagen

In den folgenden Abschnitten werden theoretische Grundlagen über das Korpus, kontrollierte Vokabulare und die Programmiersprache Python mit der Bibliothek Pandas vorgestellt.

2.1 Korpus

Korpus wird im Englisch „corpus“ geschrieben. In einem Korpus werden in der Regel schriftliche und gesprochene Texte gesammelt und auf Rechnern digitalisiert. Die Daten sind auf Computerservern ständig abrufbar (Lemnitzer und Zinsmeier 2015, S. 13). Ein Korpus kann aus einem oder mehreren kurzen oder langen Dokumenten bestehen (Heyer, Quasthoff und Wittig 2012, S. 202). Die Dokumente können sich dabei auf diverse Themen oder das gleiche Thema beziehen. Der Dokumenttyp im Korpus ist flexibel. Häufig wird ein Korpus bei verschiedenen wissenschaftlichen Arbeiten verwendet und für linguistische Zwecke eingesetzt. Zurzeit existieren weltweit viele elektronische Korpora. Das bekannteste Korpus ist eine Sammlung deutscher Sprachen und Literatur, das als Textkorpora des Instituts für Deutsche Sprache (IDS) in Mannheim bezeichnet wird (Institut für Deutsche Sprache 2017).

Bei der Bearbeitung des Korpus werden häufig Natural Language Toolkits (NLTK) verwendet, da NLTK viele unterschiedliche „corpus reader“ enthalten. NLTK ist ein Modul der Programmiersprache Python und bietet umfassende Funktionen wie Klassifizierung, Tokenisierung, Stemming, Tagging, Chunking, Parsing, semantisches Argumentieren usw. an. Darüber hinaus kann NLTK Probleme und Aufgabenstellungen der Computerlinguistik lösen (Bird 2006, S. 69).

Die im Projekt „Schaffung von mehr Transparenz in der Bundestagswahl 2017“ genutzten Dokumente im Korpus bestehen aus Zeitungsartikeln, Artikeln von Facebook und anderen Onlineinformationsquellen über Politik, zum Beispiel Twitter. Die Dokumente wurden im Textformat (*.txt) im Korpus abgespeichert.

Die Studierenden im Projekt haben am Ende des Projekts, bei der Schlusspräsentation, konkrete Beschreibungen zu eigenen Korpora erläutert. Im Folgenden wird ein Überblick über das Korpus im Projekt gegeben.

Im Projekt enthält jeder Korpus ungefähr 1000 Dokumente. Die Anzahlen der Wörter in den Dokumenten sind unterschiedlich. Die wesentlichen Bestandteile eines Dokuments im Korpus des Projekts beinhalten den Dokumentnamen, das Datum, den Titel, den Text usw. Die Studierenden haben den Inhalt eines Dokuments im Korpus auf den Folien einer Schlusspräsentation deutlich dargestellt. Folgende Abbildungen zeigen die Struktur eines Dokuments des jeweiligen Korpus der Gruppe C04 und C05.

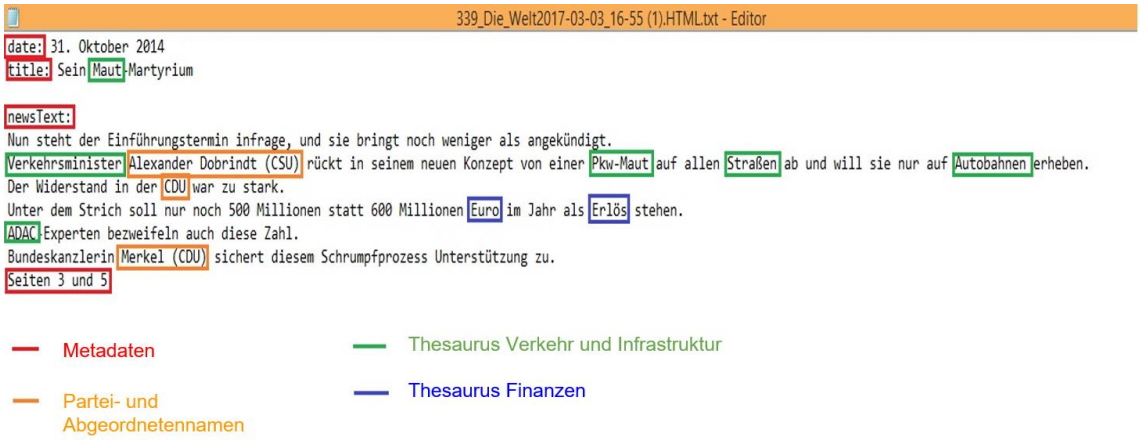

Abbildung 1: Aufbau eines Korpus-Dokuments von Borchert und Klipp

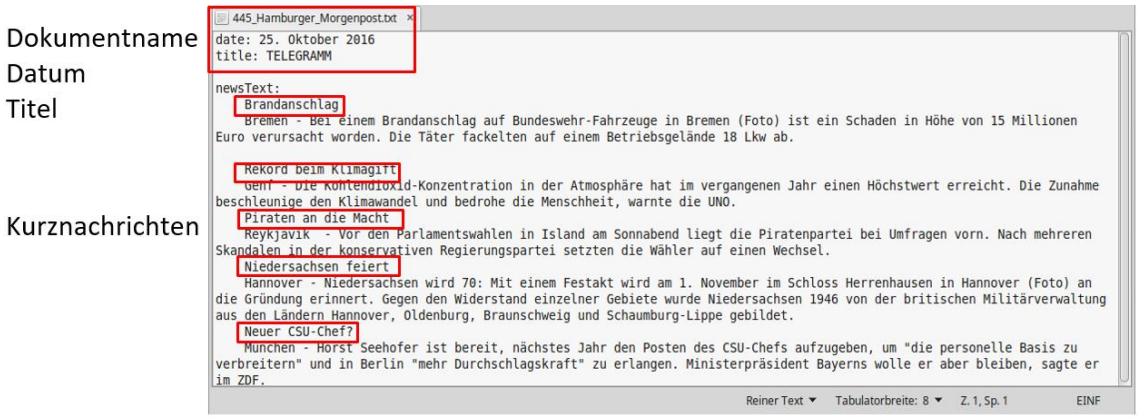

Abbildung 2: Inhalt eines Korpus- Dokuments von Gebhardt und Jaß

Ein guter Korpus ist für wissenschaftliche Aufgaben sehr wichtig, da im Korpus vorkommende repräsentative Wörter als kontrollierte Vokabulare erstellt werden können.

Die Dokumente im Korpus sind der Grundstein der kontrollierten Vokabulare. Im Projekt wurden die Korpora in der interaktiven Entwicklungsumgebung Jupyter Notebook geladen und mit der Programmiersprache Python durchgeführt. Dadurch wurden die kontrollierten Vokabulare erstellt.

2.2 Kontrollierte Vokabulare

„Kontrollierte Vokabulare“ ist ein weit gefasster Begriff. Hierbei ist in der Regel eine Sammlung von Wörtern und Phrasen gemeint. Kontrollierte Vokabulare sind geeignete Mittel, um Nutzern sinnvolle Informationen von Dokumenten zur Verfügung zu stellen. Sie beschreiben den Inhalt eines Dokumentes (zum Beispiel einer Nachricht) und andere wichtige Aspekte des Inhalts (National Information Standards Organization 2010, S. 10).

Eine englische Definition für „kontrolliertes Vokabular“ findet sich in der Literatur unter dem Begriff *Guidelines for the Construction, Format, and Monolingual Controlled Vacabularies* von der National Informations Standards Organization (NISO).

> *A list of terms that have been enumerated explicitly. This list is controlled by and is available from a controlled vocabulary registration authority. All terms in a controlled vocabulary must have an unambiguous, non-redundant definition.*
>
> *At a minimum, the following two rules must be enforced:*
>
> 1. *If the same term is commonly used to mean different concepts, then its name is explicitly qualified to resolve this ambiguity.*
> 2. *If multiple terms are used to mean the same thing, one of the terms is identified as the preferred term in the controlled vocabulary and the other terms are listed as synonyms or aliases (National Information Standards Organization 2010, S. 5).*

Die Verwendungszwecke kontrollierter Vokabulare sind:

1. *Translation: Provide a means for converting the natural language of authors, indexers and users into a vocabulary that can be used for indexing and retrieval.*
2. *Consistency: Promote uniformity in term format and in the assignment of terms*
3. *Indication of relationships: Indicate semantic relationships among terms*
4. *Label and browse: Provide consistent and clear hierarchies in a navigation system to help, users locate desired content objects*
5. *Retrieval: Serve as a searching aid in locating content objects (National Information Standards Organization 2010, S. 11).*

Kontrollierte Vokabulare spielen eine wichtige Rolle bei der inhaltlichen Erschließung des Internets oder eines Dokuments. Ein bevorzugter Begriff wird vom kontrollierten Vokabular festgelegt, da Begriffe im kontrollierten Vokabular Mehrdeutigkeiten und Redundanzen vermeiden helfen (National Information Standards Organization 2010, S. 12). List, Synonym Ring, Taxonomy und Thesaurus sind die vier wichtigsten Varianten von kontrolliertem Vokabular (National Information Standards Organization 2010, S. 16f.).

Die wichtigsten Prinzipien für ein kontrolliertes Vokabular sind:

1. Vermeidung von Mehrdeutigkeit
2. Kontrolle der Synonyme
3. Erstellung passender Beziehungen zwischen entsprechenden Begriffen
4. Prüfung und Validierung der Begriffe (National Information Standards Organization 2010, S. 12).

Nur repräsentative Begriffe werden im kontrollierten Vokabular gesammelt. Falls ein Homonym existiert, muss der Begriff auf Mehrdeutigkeit kontrolliert werden. Man soll sich Gedanken darübermachen, ob dieser Begriff für den Nutzer nötig und sinnvoll ist (National Information Standards Organization 2010, S. 12). In der folgenden Abbildung ist ein Beispiel von Mehrdeutigkeit dargestellt:

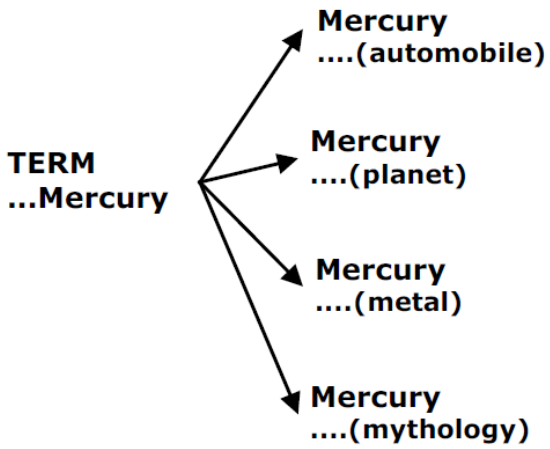

Abbildung 3: Mehrdeutigkeit des kontrollierten Vokabulars von NISO

Es kann vorkommen, dass mehrere Wörter im kontrollierten Vokabular den gleichen Begriff beschreiben, was als Synonym bezeichnet wird. Zum Bespiel beschreiben die

Begriffe „Artificial consciousness“, „Biocomputers“ und „Electronic brains“ den Fachbegriff „Conscious automata“ (National Information Standards Organization 2010, S. 13).

Ein bekanntes Beispiel ist *The Art and Architectur Thesaurus®* (National Information Standards Organization 2010, S. 15). In Abbildung 4 werden verschiedene Modelle kontrollierter Vokabulare von NISO dargestellt.

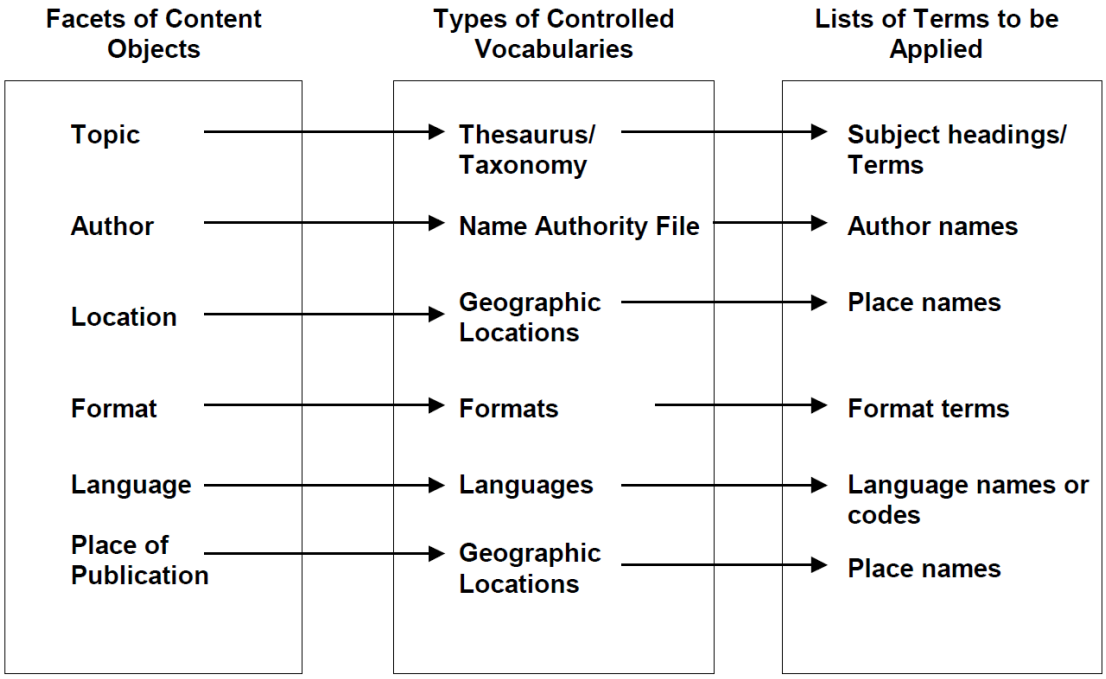

Abbildung 4: Modell kontrollierter Vokabulare von NISO

Das wichtigste Ziel des kontrollierten Vokabulars ist sicherzustellen, dass verschiedene Begriffe durch eine einfache linguistische Form erklärt werden können (National Information Standards Organization 2010, S. 12).

2.3 Python und Pandas

Python wurde von Guido van Rossum im Jahr 1989 entwickelt. Genau wie Perl wird Python auch unter der GNU General Public License (GPL) veröffentlicht. Bei Linguisten ist Python besonders beliebt (Weigend 2016, S. 39).

Python ist eine einfache und moderne Programmiersprache. Vorkenntnisse sind nicht zwingend erforderlich (Weigend 2016, S. 23). Python konzentriert sich auf eine schnelle und einfache Programmierbarkeit, Vollständigkeit und Anpassbarkeit, weswegen

Python sowie Perl häufig als Skriptsprachen bezeichnet werden (McKinney 2015, S. 2). Das mit Python geschriebene Programm läuft auch wesentlich schneller als solche, die in anderer Sprache wie C++ und Java geschrieben wurden (McKinney 2015, S. 3).

Die Programmiersprache Python hat einen eigenen Python-Interpreter (Eingabeaufforderung), in dem man die Anweisungen hineinschreibt (Sweigart 2016, S. 4). In der Python Shell von IDLE lässt sich auch der Code eingeben und ausführen. Die Anweisungen werden durch den Python-Interpreter ausgearbeitet (Weigend 2016, S. 43). Beim Programmieren mit Python ist es auch möglich, Anweisungen im Texteditor zunächst zu schreiben und als Python-Skript-Format(*.py) zu speichern. Das Python-Skript wird zum Schluss im Python-Interpreter aufgerufen, um das Programm dann dort auszuführen (Weigend 2016, S. 66). Aktuell gibt es eine bessere Distribution von Python, die als Anaconda bekannt ist.

Distribution Anaconda ist ein kostenloses Installationsprogramm. Das Programm ist in Python und C geschrieben (Woyand 2017, S. 252). Distribution Anaconda enthält ein interaktives Jupyter Notebook. Im Jupyter Notebook werden die Python-Skripte direkt geschrieben und ausgeführt. Das Jupyter Notebook stammt aus einem Jupyter-Projekt, das früher als Ipython-Projekt bekannt war. „Die Jupyter Notebook-Dateien enthalten beliebigen Texte, mathematische Formeln, Eingabecode, Ergebnisse, Grafiken, Videos und beliebige andere Medientypen, die in modernen Webbrowsern angezeigt werden können“ (Microsoft Azure 2016).

Tools wie Python, R, MATLAB, SAS und Stata sind im Bereich der Datenanalyse sehr bekannt (McKinney 2015, S. 2). Im Bereich der Datenverarbeitung wird Python häufig mit R, MATLAB, SAS, Stata und anderen domänenspezifischen Sprachen (DSL) und Tools verglichen. Die Programmiersprache R ist ein freies Open-Source-Programm. R ist ein leistungsstarkes Werkzeug und ermöglicht es, große Datenmengen zu analysieren (Wollschläger 2014, S. 2). Die Programmiersprache Python bietet jedoch eine stabilere und effektivere Umgebung zur Datenanalyse (McKinney 2015, S. 5). Die Datenanalyse mit Python ist mithilfe der Bibliothek Pandas deutlich einfacher geworden (McKinney 2015, S. 2). Denn die Python-Bibliothek Pandas kann viele Probleme der Datenverarbeitung sehr effektiv lösen (McKinney 2015, S. 115).

Pandas heißt auf Deutsch „Panel-Daten“, was sich vom englischen Begriff „panel data“ ableitet. Pandas stellt umfassende Funktionen zur Datenanalyse und Datenstrukturie-

rung zur Verfügung. Die wichtigsten Pandas-Objekte sind DataFrame und Series (McKinney 2015, S. 5). Der DataFrame in der Python-Bibliothek Pandas ist ähnlich wie der DataFrame in R. Er basiert grundlegend auf Tabellen. Die Struktur eines DataFrames kann man als tabellarische Datenstruktur ansehen. DataFrame hat sowohl einen Zeilen- als auch einen Spaltenindex. Ein Spalt kann dabei nur einen Datentyp besitzen. Aber verschiedene Spalten können aus unterschiedlichen Datentypen bestehen, zum Beispiel numerisch, String, boolesch usw. (McKinney 2015, S. 119). Series ist ein Objekt, das ähnlich wie ein eindimensionales Array ist. Series kann verschiedene Datentypen aufnehmen. Das einfachste Series kann aus einer Liste von Daten bestehen (McKinney 2015, S. 116).

Die Basisfunktionen von DataFrame und Series sind Indexierung, Selektieren, Filtern, Arithmetik und Datenausrichtung, Sortieren und Rangbildung sowie Operationen zwischen DataFrame und Series (McKinney 2015, S. 130ff.). Eine wichtige Grundlage in Python ist NumPy. NumPy ist die Abkürzung für Numerical Python (McKinney 2012, S. 79) und stellt die Funktionen für mathematische und numerische Routinen bereit (McKinney 2015, S. 4).

Im Projekt „Schaffung von mehr Transparenz in der Bundestagswahl 2017“ haben die Studierenden bereits das Jupyter Notebook verwendet, um kontrollierte Vokabulare zu erstellen. Aufgrund der Überlegenheit gegenüber anderen Datenanalysetools wird sich in dieser Arbeit für Python mithilfe der Bibliothek Pandas als Programmiersprache zur Konsolidierung der kontrollierten Vokabulare entschieden. Um die kontrollierten Vokabulare zu analysieren, werden Funktionen wie Indexierung, Sortieren und Filtern am häufigsten genutzt. Auf Details dazu wird in Kapitel 4 näher eingegangen.

2.4 Konsolidierung

Die Verbform von „Konsolidierung“ ist „konsolidieren“. Im Wirtschaftsbereich bedeutet es „festigen“, „sichern“ bzw. „umwandeln“ oder „zusammenlegen“ (Götz, Haensch und Wellmann 2003, S. 599). In der Betriebswirtschaft und in der Volkswirtschaft hat der Begriff „Konsolidierung“ unterschiedliche Bedeutungen.

In der Betriebswirtschaft ist eine Konsolidierung für Unternehmen nicht fremd, insbesondere dann, wenn das Unternehmen mehrere Tochterunternehmen oder Partner hat. Bei der Erstellung des Jahresabschlusses bedeutet Konsolidierung das Zusammenfassen

von Einzelabschlüssen mehrerer Tochterunternehmen. Wenn ein Mutterunternehmen sein Tochterunternehmen direkt oder indirekt beherrscht, spricht man von „Vollkonsolidierung“. Die Tochterunternehmen müssen dann voll konsolidiert werden (Hans-Böckler-Stiftung 2011, S. 12).

Es gibt auch eine Quotenkonsolidierung, das heißt, nur ein Teil wird konsolidiert (Hans-Böckler-Stiftung 2011, S. 13). Der Konsolidierungskreis wird nach Handelsgesetzbuch (HGB) und nach International Financial Reporting Standards (IFRS) abgegrenzt. Es gibt allgemeine Konsolidierungsregeln im Unternehmenserwerb (Sachse 2015, S. 5). Eine Konsolidierungsmaßnahme beinhaltet unterschiedliche Schritte wie die Kapitelkonsolidierung, die Schuldenkonsolidierung, die Zwischenergebniseliminierung sowie die Aufwands- und Ertragskonsolidierung (Hans-Böckler-Stiftung 2011, S. 19). Die Kapitelkonsolidierung, die Schuldenkonsolidierung und die Zwischenergebniskonsolidierung sind drei grundlegende Methoden der Konsolidierung im Konzernabschluss. Egal ob Konsolidierung nach HGB oder IFRS, häufig wird die Equity-Methode angewendet. Die Equity-Methode ist ein Verfahren zur Bewertung von Beteiligungen im Jahresabschluss (Hans-Böckler-Stiftung 2011, S. 13). Die Konsolidierung in der Wirtschaft stellt alle Informationen aus verschiedenen Seiten zusammen, wodurch ein Mutterunternehmen seine Tochterunternehmen bequem verwaltet.

Im Sachrecht und Finanzwesen hat Konsolidierung noch andere Bedeutungen. Es wird hier aber nicht näher darauf eingegangen, da es kaum Literatur zur Konsolidierung kontrollierter Vokabulare gibt. In der vorliegenden Arbeit wird jedoch der Sinn von Konsolidierung auf ähnliche Weise wie bei einem Konzernabschluss betrachtet.

Die Konsolidierung kontrollierter Vokabulare in dieser Arbeit bezieht sich sowohl auf die Zusammenführung der 19 kontrollierten Vokabulare aus dem Projekt als auch auf das notwendige Bereinigen der Daten, welche die wichtigsten Eigenschaften sowie eine gute Qualität des finalen kontrollierten Vokabulars gewährleisten. Die Vorgehensweise zur Durchführung der Konsolidierung wird in Kapitel 4 dieser Arbeit vorgestellt.

3 Projektvorstellung

Das Projekt „Schaffung von mehr Transparenz in der Bundestagswahl 2017“ wurde von Studierenden an der Hochschule für Angewandte Wissenschaften im Studiengang *Medien und Information (MUI)* im Sommersemester 2017 durchgeführt. Es war eine Zusammenarbeit zwischen dem HAW DataLab, abgeordnetenwatch.de (AW) sowie der Datenschule (Glissmann 2017, S. 2) und wurde von Prof. Dr. Glissmann geleitet. In diesem Projekt wurde ein Text-Mining-Verfahren angewendet. Der nötige Programmiercode im Projekt wurde von Prof. Dr. Glissmann in der Entwicklungsumgebung Jupyter Notebook mit Python erstellt.

Abgeordnetenwatch.de ist ein unabhängiges und überparteiliches Internetportal, auf dem Bürgerinnen und Bürger Politikerinnen und Politiker zu bestimmten politischen Themen direkt befragen und die Politikerinnen und Politiker den Bürgerinnen und Bürgern direkt antworten können (Abgeornetenwatch.de 2017). Die Fragen und Antworten sind online veröffentlicht. Folgende Abbildung zeigt das Infoportal Abgeornetenwatch.de.

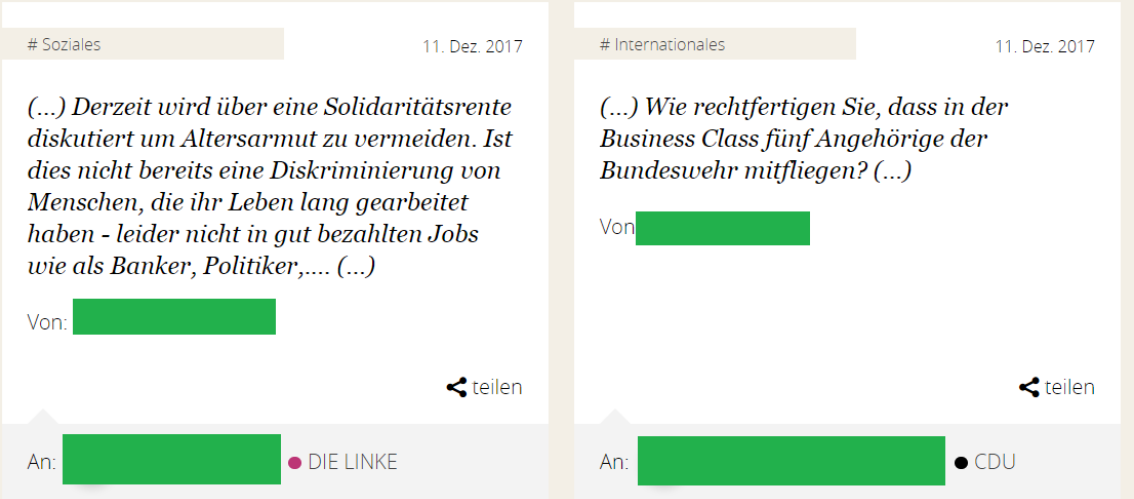

Abbildung 5: Fragen an Abgeordnete der EU auf Abgeordnete.de

Die Datenschule ist eine gemeinnützige Einrichtung in Berlin, die sich mit dem Bereich Daten und Technologie beschäftigt, ähnlich wie digitale Tools und Daten (Glissmann 2017, S. 2). Ziel der Datenschule ist es, Wissen zu vermitteln. Dabei lässt sich der gesellschaftliche Wandel positiv gestalten und mit anderen Organisationen oder Projekten

kommunizieren (Datenschule 2017). Die im Projekt zum Teil genutzten Daten wurden von dem Infoportal Abgeornetenwatch.de ausgewählt. Andere Datenquellen sind SPD auf Facebook, Bündnis 90/Die Grünen auf Facebook und andere Social-Media-Kanäle.

Ziel des gesamten Projekts ist es, kontrollierte Vokabulare zur Verschlagwortung verschiedenster politischen Beiträge zu entwickeln (Glissmann 2017, S. 3). Das Projekt wurde im Mai 2017 mit positivem Ergebnis erfolgreich abgeschlossen.

In diesem Kapitel wird zunächst die Methode zur Informationsgewinnung im Projekt vorgestellt. Anschließend werden der Ablaufprozess und entsprechende Ergebnisse des Projekts vorgestellt.

3.1 Methode zur Informationsgewinnung

Wie werden Informationen zum aktuellen Thema gewonnen? Dafür gibt es viele verschiedene Methoden. In der Informationswissenschaft wird von „Information Retrieval" als Wissenschaftsdisziplin beim Wiederauffinden gespeicherten Wissens gesprochen (Stock 2007, S. 38). Information Retrieval stammt aus dem Bereich Database. Allerdings reicht Information Retrieval allein zur Verarbeitung von Massendaten nicht aus. Häufig wird Information Retrieval mit einer Informationsextraktion kombiniert und es wird beides zusammen angewendet (Seidenfaden 2007, S. 155).

> *Die Hauptaufgaben im Bereich IR liegen in der Indizierung und Suche im gesamten Dokumentenbestand (Volltextsuche), in der Textklassifikation und im Textclustering (Textgruppierung). […] Im Bereich der IE liegt der Forschungsschwerpunkt in der Extraktion von Eigennamen und Relationen und in der Gruppierung und Fusion gefundener Entitäten (Carstensen, Ebert, Jekat, Klabunde und Langer 2010, S. 584).*

Die Bewertung der resultierenden Ergebnisse aus Information Retrieval und Informationsextraktion kann per Precision and Recall geschehen.

Durch die Zusammenarbeit von Information Retrieval und Informationsextraktion können relevante Informationen in der Datenbank gesucht und gefunden werden. Jedoch reichen diese häufig nicht aus, um neues Wissen aus Texten als Wissensrohstoff zu gewinnen. Deshalb wird heutzutage die Text-Mining-Methode umfangreich angewendet: um unstrukturierte Daten mithilfe der Verarbeitung in natürliche Sprache auszuwerten, was einen automatischen Prozess der Gewinnung neuer Informationen aus Textdokumenten ermöglicht (Hippner und Rentzmann 2006, S. 287).

Text Mining bezeichnet eine Methode, bei der Informationen und Erkenntnisse aus Textdokumenten gewonnen werden. Mit Text Mining lassen sich neue und wertvolle Informationen extrahieren. Die wichtigsten Begriffe für Text Mining sind „Text“, „Information“ und „Wissen“ (Heyer, Quasthoff und Wittig 2012, S. 1). Darüber hinaus wird Text Mining als „semantische Textanalyse“ bezeichnet (Heyer, Quasthoff und Wittig 2012, S. 3). Beim Text Mining werden vier Prozessschritte durchgeführt: Suche, Vorverarbeitung, Bewertung und Extraktion. Das Information Retrieval dient in einem ersten Schritt der Suche, die Informationsextraktion als letzter Schritt wird ebenfalls als Teilgebiet des Text Mining betrachtet. In der Vorverarbeitung wird eine statistische Sprachverarbeitung mit Verfahren der Computerlinguistik durchgeführt. In der Bewertung werden die vorverarbeiteten Dokumente durch Klassifizierungsverfahren und Clustering zueinander in Beziehung gesetzt (Sullivan 2001, S. 324).

Im Projekt „Schaffung von mehr Transparenz in der Bundestagswahl 2017“ wurde die Text-Mining-Methode zur Erstellung kontrollierter Vokabulare angewendet. Das angewendete Vorgehensmodell wurde von Schieber und Hilbert dargestellt. Wardazky hat in ihrer Arbeit diese Vorgehensweise in einer Tabelle zusammengefasst. Folgende Abbildung zeigt das Text-Mining-Vorgehensmodell von Wardazky im Projekt (Wardazky 2016, S. 26).

Phase	***Schritte***
I	• Analyseziele definieren • Anwendungsdomäne bestimmen
II	• Quellsysteme und -dokumente bestimmen • Eigenschaften der Dokumente feststellen • Dokumente in Arbeitsbereich übertragen

III	• Terme identifizieren
	• Linguistische Aufbereitung durchführen
	  o Lexikalische Analyse
	  o Syntaktische Analyse
	  o Semantische Analyse
	• Technische Aufbereitung durchführen
	  o Terme indexieren und gewichten
	  o Terme reduzieren
	  o Datenstruktur anpassen
	  o Vorbereitende Analyse durchführen

IV	• Klassifikationsverfahren • Segmentierungsverfahren • Abhängigkeitsanalysen • Meinungsanalyseverfahren • Sonstige Verfahren
V	• Kennzahlen auswählen • Ergebnisse auswerten • Vorgehen überprüfen und nächste Schritte definieren
VI	• Maßnahmen ableiten und Ergebnisse anwenden

Abbildung 6: Text-Mining-Vorgehensmodell von Wardazky

Im Projekt haben die Studierenden die Entwicklung kontrollierter Vokabulare hauptsächlich in Phase IV durchgeführt. Die Dokumente im Korpus zur Erstellung kontrollierter Vokabulare wurden von Professorin Dr. Glissmann vorbreitet. Im folgenden Kapitel wird der Prozess des Projekts genauer erläutert. Es wird beschrieben, womit sich die MUI-Studierenden im Projekt beschäftigt haben.

3.2 Prozess des Projekts

Im Projekt „Schaffung von mehr Transparenz in der Bundestagswahl 2017“ haben die Studierenden im Wesentlichen nach dem Vorgehensmodell der Text-Mining-Analyse in der Programmiersprache Python gearbeitet. Insgesamt gab es 19 Gruppen, wobei sich eine Gruppe immer aus zwei Studierenden zusammensetzte. Jede Gruppe erhielt dabei einen Korpus. Ein Korpus beinhaltete ungefähr 1000 .txt-Dateien, die aus XML-, HTML- und PDF-Formate umgewandelt wurden. Die Dateien enthielten verschiedenste politische Beiträge aus dem Internet, zum Beispiel Beiträge aus *Facebook-Posts der Grünen*, politische Zeitungsartikel aus *Die Zeit* usw.

Die Studierenden haben die Dokumente aus dem Korpus in der interaktiven Entwicklungsumgebung Jupyter Notebook eingelesen und durchlaufen. Anschließend wurden die Ergebnisse im Excel-Format (*.xlsx) gespeichert. Die Python-Skripte haben die Studierenden von Prof. Dr. Glissmann bekommen. Mit den Skripten konnten die Studierenden auch eine Liste mit sogenannten Stoppwörtern erstellen. Stoppwörter sind Wörter, bezüglich derer bei zu analysierenden Daten keine Bedeutung zu erwarten ist, zum Beispiel „ein“, „der“, „die“ und „das“.

Einige Teilaufgaben haben die Studierenden manuell erfüllt. Das heißt, die Studierenden haben Dokumente im Korpus selbst gelesen, um repräsentative Begriffe selbst auszuwählen, zum Beispiel den Begriff „Demokratie“. Die selbst ausgewählten Begriffe wurden anschließend mit automatisch ausgewählten Begriffen verglichen. Da es im Projekt 19 Gruppen gab, wurden zum Schluss 19 kontrollierte Vokabulare erstellt. Die Anzahl der Begriffe in den kontrollierten Vokabularen einer jeden Excel-Datei war unterschiedlich, zudem waren die Begriffe vom Inhalt des politischen Artikels abhängig. Die 19 kontrollierten Vokabulare in den Excel-Dateien enthielten zwischen 200 und 1500 Begriffe.

Zum Schluss wurden die Begriffe den politischen Topics zugeordnet, zum Beispiel „Flucht“, „Anschlag“ usw., das heißt, sie wurden beschlagwortet. Während der Festlegung der Topics diskutierten die Studierenden viel darüber, wie die Begriffe in den kontrollierten Vokabularen bestimmten Topics zugeordnet werden können. Dadurch sollte die Mehrdeutigkeit eines Begriffs in den kontrollierten Vokabularen vermieden werden. Außerdem fügten die Studierenden in den Excel-Tabellen die Quellen der Begriffe, deren Häufigkeit usw. ein. Folgende Abbildung zeigt den Inhalt eines kontrollierten Vokabulars in einer Excel-Tabelle des Projekts.

synonym (gestemmt)	synonym (original)	topic	topic_art	Quelle	Quellentyp
abendschul	abendschule	Schule	Political_AW	wortliste	sortiert nach
abgasaffar	abgasaffäre	Wirtschaft	Political_AW	Hamburger Abendblatt	Zeitung
abgasbetrug	abgasbetrug	Wirtschaft	Political_AW	Hamburger Abendblatt	Zeitung
abgasnorm	abgasnorm, abgasnormen	Umwelt	Political_AW	Hamburger Abendblatt	Zeitung
abgasskandal	abgasskandal	Wirtschaft	Political_AW	Hamburger Abendblatt	Zeitung
abgelt	abgeltung	Finanzen	Political_AW	wortliste	sortiert nach
abrust	abrüstung	Sicherheit	Political_AW	Hamburger Abendblatt	Zeitung
abschieb	abschiebung, abschiebungen, abschieben	Flucht	Political_NEW	Hamburger Abendblatt	Zeitung
abschw	Abschwung	Wirtschaft	Political_AW	Hamburger Abendblatt	Zeitung
adac	adac	Verkehr und Infrastruktur	Political_AW	Eigenes Wort	Einfall
akademikerkind	akademikerkind	Kinder und Jugend	Political_AW	Eigenes Wort	Einfall
akti	Aktie	Wirtschaft	Political_AW	Hamburger Abendblatt	Zeitung
akzeptanz	akzeptanz	Gesellschaft	Political_AW	wortliste	sortiert nach
alleinerzieh	alleinerziehende, alleinerziehenden,	Familie	Political_AW	wortliste	sortiert nach
alphabetisier	alphabetisierung	Bildung und Forschung	Political_AW	eigenes Wort	Einfall

Abbildung 7: Strukturierung kontrollierter Vokabulare

3.3 Zusammenfassung

Das Projekt wurde mit positiven Ergebnissen abgeschlossen. Die kontrollierten Vokabulare wurden durch eine automatische Analyse mithilfe der Programmiersprache Python erstellt und mit entsprechenden Topics in den Excel-Dateien gespeichert. Die Ergebnisse des Projekts wurden in 19 verschiedenen Dateien gespeichert. Die 19 unterschiedlichen kontrollierten Vokabulare enthalten über 9000 Begriffe. Die 19 Dateien sind in einem Korpus gespeichert. Die im Projekt entwickelten 19 kontrollierten Voka-

bulare sollen noch in einem endgültigen kontrollierten Vokabular konsolidiert werden, welche die relevante Aufgabe dieser Arbeit darstellt. In Kapitel 4 wird die Vorgehensweise der Konsolidierung und Aufbereitung kontrollierter Vokabulare erläutert.

4 Konsolidierung kontrollierter Vokabulare

In diesem Kapitel wird das methodische Vorgehen zur Konsolidierung kontrollierter Vokabulare vorgestellt. Die dazu verwendeten kontrollierten Vokabulare stammen aus dem vorgestellten Projekt „Schaffung von mehr Transparenz in der Bundestagswahl 2017“. Der entwickelte Programmcode zur Konsolidierung wurde mit der Python-Bibliothek Pandas in der interaktiven Entwicklungsumgebung Jupyter Notebook erstellt.

Zunächst werden die Forschungsfragen zur Konsolidierung konkretisiert, dann das Konzept zur Realisierung der Konsolidierung erstellt. Anschließend werden die einzelnen Arbeitsschritte zur Durchführung der Konsolidierung der kontrollierten Vokabulare vorgestellt.

4.1 Forschungsfrage

Vor der Konsolidierung der kontrollierten Vokabulare aus den 19 Gruppen des Projekts sollten die Forschungsfragen klar definiert werden. Denn mit der Beantwortung der Forschungsfragen werden konkrete Schritte eingeleitet, mit denen sich die Konsolidierung der kontrollierten Vokabulare zielorientiert und argumentiert realisieren lässt.

Da sich die 19 Gruppen im Projekt mit sich überlappenden Themenbereichen beschäftigt haben, kann es nach der physikalischen Zusammenführung der 19 Vokabulare Duplikate und Mehrdeutigkeiten im zusammengeführten Vokabular geben. Gemäß den Prinzipien des kontrollierten Vokabulars, die in Kapitel 2 erwähnt wurden, muss ein gutes kontrolliertes Vokabular Mehrdeutigkeiten und Duplikate vermieden werden (Pellegrini und Blumauer 2006, S. 362). Ein gutes kontrolliertes Vokabular soll demnach auf Synonyme und Beziehungen zwischen mehreren Begriffen achten (National Informations Standards Organization, S. 12).

Unter diesen Gesichtspunkten bezieht sich die Konsolidierung der kontrollierten Vokabulare dieser Arbeit sowohl auf die Zusammenführung der 19 kontrollierten Vokabulare aus dem Projekt als auch auf das notwendige Bereinigen dieser Daten, um wichtige Eigenschaften gemäß Definition kontrolliertes Vokabulars sowie eine gute Qualität des

finalen kontrollierten Vokabulars zu gewährleisten. Deshalb liegt der Fokus dieser Arbeit auf der Frage, wie die kontrollierten politischen Vokabulare der 19 Gruppen aus dem Projekt mithilfe der Python-Bibliothek Pandas zusammengeführt sowie bereinigt werden können und wie die Begriffe bestimmten Topics angemessen zugeordnet werden können.

Jedes in Excel gespeicherte kontrollierte Vokabular des Projekts enthält mehrere Spalten. Die ausgewählten Begriffe in den kontrollierten Vokabularen werden möglicherweise Suchworte beim Suchen beeinflussen. Nach der Konsolidierung der 19 kontrollierten Vokabulare aus dem Projekt könnte es dazu kommen, dass zwei oder mehr identische Begriffe in dem kontrollierten Vokabular existieren. Es kann auch vorkommen, dass ein Begriff gleichzeitig unterschiedlichen Topics zugeordnet wird und mehrdeutige Begriffe beziehungsweise unpassende Topics im kontrollierten Vokabular vorhanden sind.

Unter oben genannten Gesichtspunkten sind die folgenden wichtigen Unterfragen zur Konsolidierung der im Rahmen dieser Arbeit kontrollierten Vokabulare zu beantworten:

1. Sind die in allen Spalten im konsolidierten Vokabular enthaltenden Informationen für die Konsolidierung relevant? Mit welchen Daten in den kontrollierten Vokabularen sollte man sich bei der Konsolidierung beschäftigen?
2. Wie lassen sich die Zusammenführung kontrollierter Vokabulare und die Aufbereitung von Massendaten mithilfe der Python-Bibliothek Pandas durchführen?
3. Wie soll man Duplikate und mehrdeutige Begriffe sowie unpassende Topics während oder nach der Zusammenführung der kontrollierten Vokabulare behandeln?
4. Wie soll man entscheiden, welche Begriffe zum finalen kontrollierten Vokabular gehören?

4.2 Konzeptualisierung

Um die 19 kontrollierten Vokabulare im Projekt zu konsolidieren und schließlich ein finales hochqualitatives kontrolliertes Vokabular zu entwickeln, musste zunächst ein konzeptioneller Plan ausgearbeitet werden. Gemäß dem Konzept sollte eine methodische Vorgehensweise zur Realisierung der Konsolidierung angewendet werden, um die einzelnen Arbeitsschritte bestimmen zu können.

Gemäß den oben genannten Forschungsfragen kann die Konsolidierung kontrollierter Vokabulare grundsätzlich in zwei Phasen unterteilt werden. Die erste Phase ist Zusammenführung der 19 kontrollierten Vokabulare aus 19 verschiedenen Gruppen im Projekt, und die zweite Phase ist das Bereinigen der Daten im zusammengeführten Datei.

Aufgrund der Forschungsfragen müssen in der ersten Phase die kontrollierten Vokabulare aus allen Gruppen betrachtet werden, um herauszufinden, wie die Zusammenführung kontrollierter Vokabulare effektiv verwirklicht werden kann. Die 19 kontrollierten Vokabulare werden hierbei einzeln verarbeitet, zum Schluss sollten sie in einem neuen Korpus gespeichert werden.

Die einzeln kontrollierten Vokabulare wurden zunächst in der Python-Bibliothek Pandas eingelesen. Da nicht alle Informationen im kontrollierten Vokabular relevant sind, wurde der konsolidierte Bereich des kontrollierten Vokabulars eingegrenzt, denn dadurch erhöht sich die Effektivität der Datenverarbeitung bei der Konsolidierung. Somit konnte der Konsolidierungsbereich mit relevanten Objekten des kontrollierten Vokabulars in Pandas extrahiert werden. Während der Eingrenzung des Konsolidierungsbereichs wurden Duplikate entfernt.

Die ausgewählten Bereiche wurden anschließend in einer neuen Datei gespeichert. Zuvor war es jedoch nötig, der Datei eine neue Spalte hinzuzufügen, wofür ein Sheet-Name in Pandas vergeben wurde. Von dort aus kann die Quelle des kontrollierten Vokabulars nach der Zusammenführung noch erkannt werden. Anschließend konnte die Zusammenführung der 19 neuen gespeicherten kontrollierten Vokabulare durchgeführt werden, wobei ein zusammengeführtes Vokabular generiert wurde. Somit war die erste Phase der Konsolidierung abgeschlossen. Allerdings war die zusammengeführte Datei noch kein gutes kontrolliertes Vokabular, weil sie noch nicht die Eigenschaften eines guten kontrollierten Vokabulars besaß.

In der zweiten Phase sollte überlegt werden, wie die Begriffe in der neu generierten zusammengeführten Datei aufbereitet werden können, damit ein gutes kontrolliertes Vokabular entsteht.

Zu diesem Zeitpunkt war es noch möglich, dass die 19 kontrollierten Vokabular-Dateien aus dem Projekt gleiche Begriffe enthalten, sodass nach der Zusammenführung manche Begriffe zwei- oder mehrfach vorhanden sind. Daher sollten die zusammengelegten Begriffe nach der Zusammenfassung der kontrollierten Vokabulare automatisch

miteinander verglichen werden. Für den Fall, dass sich zwei oder mehr Zeilen zu 100% im zusammengeführten Vokabular gleichen, sollten diese Duplikate identifiziert und automatisch gelöscht werden. Für den Fall, dass zwei oder mehr Zeilen inhaltlich ähnlich, aber nicht identisch waren, sollten diese zunächst beibehalten werden. Nach der Zusammenführung sollten die Duplikate und fehlende Daten wie Lücken im gesamten Umfang der 19 Vokabulare geprüft und durch entsprechende Funktionen von Pandas automatisch entfernt werden.

Darüber hinaus sollte die Häufigkeit des Topics in der zusammengeführten Datei in absteigender Reihenfolge gezeigt werden können. Die Topics sollten darauf geprüft werden, ob sie eventuell umzubenennen sind. Falls die Bedeutung eines Topics zu umfangreich war bzw. das Topic sich nicht auf ein bestimmtes Thema bezog, wurde eine manuelle Bearbeitung durchgeführt.

Darauf folgte eine Prüfung auf Mehrdeutigkeiten der Begriffe im zusammengeführten Vokabular. Um Mehrdeutigkeiten zu eliminieren, sollten Begriffe manuell geprüft und bearbeiten werden, da es sich hierbei um konkrete Bedeutungen der Begriffe handelte, sodass eine vollautomatische Bearbeitung nicht möglich war.

Zum Schluss sollte eine allgemeine Qualitätskontrolle durchgeführt werden, um das Ergebnis des kontrollierten Vokabulars zu bewerten. Gemäß den oben genannten Überlegungen wurde die Vorgehensweise zur Realisierung der Konsolidierung wie in der folgenden Tabelle dargestellt konzeptualisiert:

Arbeitsvorgang	**Schritte**
I	• Forschungsfrage
II	• Konzeptualisierung

III	• Zusammenführung kontrollierter Vokabulare (Phase I)
	  ○ Vorbereitung
	  ○ Zusammenführung

IV	• Datenaufbereitung (Phase II)
	  ○ Duplikate und Lücken prüfen und entfernen
	  ○ Umbenennung des Topics
	  • Manuelle Bearbeitung
	  ○ Mehrdeutigkeit
	  • Manuelle Bearbeitung

V	• Ergebnis

Tabelle 1: Plan der Konsolidierung von kontrollierten Vokabularen

4.3 Zusammenführung kontrollierter Vokabulare

In diesem Unterkapitel wird eine Zusammenführung der 19 kontrollierten Vokabulare als ersten Schritt der Konsolidierung dargestellt. Es gab insgesamt 19 kontrollierte Vokabular-Dateien aus den 19 Gruppen dieses Projekts. Im Projekt waren die 19 Gruppen in die drei großen Gruppen A, B und C aufgeteilt. In dieser Arbeit wird die Zusammenführung der Vokabulare gemäß Gruppenaufteilung in zwei Schritten durchgeführt. Folgende Tabelle gibt einen Einblick, wie die kontrollierten Vokabulare schrittweise zusammengelegt wurden.

Die kontrollierten Vokabulare Excel-Datei	Schritt 1. Zusammenlegung	Schritte 2. Zusammenführung
A01_Vokabular.xlsx	Gruppe A_Vokabular.xlsx	Kontrollierte_Vokabulare.xlsx
A02_Vokabular.xlsx		
A03_Vokabular.xlsx		
A04_Vokabular.xlsx		
A06_Vokabular.xlsx		
A10_Vokabular.xlsx		
B01_Vokabular.xlsx	Gruppe B_Vokabular.xlsx	
B03_Vokabular.xlsx		
B05_Vokabular.xlsx		
B06_Vokabular.xlsx		
B07_Vokabular.xlsx		
B09_Vokabular.xlsx		
B10_Vokabular.xlsx		
C03_Vokabular.xlsx	Gruppe C_Vokabular.xlsx	
C04_Vokabular.xlsx		
C05_Vokabular.xlsx		
C06_Vokabular.xlsx		
C07_Vokabular.xlsx		
C10_Vokabular.xlsx		

Abbildung 8: Verfahren zur Zusammenführung kontrollierter Vokabulare

Zunächst wurden die 19 kontrollierten Vokabulare je nach Gruppe (A, B und C) in drei Dateien zusammengeführt und anschließend in einer Datei zusammengeführt, die dann noch bearbeitet und bereinigt werden musste. Dabei ließen sich Ähnlichkeiten und Unterschiede der Vokabulare der drei Bereiche beobachten. Die 19 kontrollierten Vokabular-Dateien hätten natürlich auch auf einmal zusammengeführt werden können, was das gleiche Ergebnis wie bei der Zusammenführung von Abbildung 8 geliefert hätte.

Im folgenden Unterkapitel wird das konkrete Vorgehen zur Zusammenführung kontrollierter Vokabulare ausführlich erklärt. Folgende Infobox zeigt, was in dieser Phase gemacht wurde und was dabei herauskommen ist. In jeder Phase wurde zunächst eine Infobox erstellt.

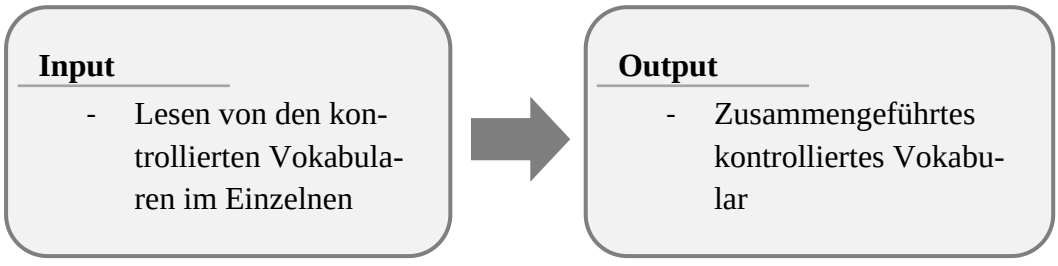

Abbildung 9: Input und Output von Phase I

4.3.1 Vorbereitung vor der Zusammenführung

In diesem Unterkapitel werden zunächst das genutzte Tool und die Schritte zu der Zusammenführung erläutert.

Die Python-Bibliothek Pandas ermöglicht es, viele Formate in DataFrames wie CSV, Excel, Tabellen, HTML und anderen Datenformaten einzulesen. Am meisten werden die Funktion *read_csv* und *read_table* genutzt (Wes McKinney 2015, S. 161). Die beliebteste Funktion in der Python-Bibliothek Pandas ist das Lesen tabellarischer Daten. Diese Funktion ist für die Konsolidierung kontrollierter Vokabulare sehr gut geeignet. Da diese Arbeit die Datenanalyse fokussiert, wird hierbei kein Hinweis zum Programmierstil von Python gegeben, sondern Anweisungen der Python-Bibliothek Pandas zur Konsolidierung kontrollierter Vokabulare im Jupyter Notebook vorgestellt.

Sobald man die Python-Bibliothek Pandas und entsprechende Objekte wie DataFrames und Series in die interaktive Entwicklungsumgebung Jupyter Notebook importiert, kann das kontrollierte Vokabular einzeln eingelesen werden. Voraussetzung ist, dass Jupyter Notebook und die Dateien in einem Ordner speichert sind. Um eine Datei aus einem richtigen Ort aufzurufen, kann man den Befehl *os.getcwd()* verwenden. Diese Anweisung zeigt ein aktuelles Arbeitsverzeichnis.

Die Python-Bibliothek Pandas bietet zwar die Funktion an, alle Dateien auf einmal in Jupyter einzulesen und zu bearbeiten, jedoch ist es in diesem Fall nötig, die Dateien im Einzelnen in der interaktiven Entwicklungsumgebung Jupyter Notebook einzulesen. Das erleichtert es, einen passenden Sheet-Name festzulegen.

Excel-Datei mit *pandas.read_excel* laden

Pandas stellt viele Funktionen zur Verfügung, um tabellarische Daten einzulesen. Folgende Tabelle gibt die Funktionen von Pandas zum Lesen tabellarischer Daten wieder.

Funktion	Beschreibung
read_csv	Die zu bearbeitenden Daten werden aus einer Datei, URL oder einem ähnlichen Objekt geladen. In diesem Fall sind Kommata als Trennzeichen zu verwenden.
read_table	Die zu bearbeitenden Daten werden aus einer Datei, URL oder einem ähnlichen Objekt geladen. In diesem Fall sind Tabulatoren ('\t') als Trennzeichen zu verwenden.
read_excel	Die zu bearbeitenden Daten aus einer Excel-Datei werden geladen.
read_fwf	Die Daten werden mit unflexibler Spaltenbreite ohne Trennzeichen gelesen.
read_clipboard	Diese Funktion ist für Daten aus Webseiten geeignet.

Tabelle 2: Funktionen zum Parsen in Pandas (McKinney 2015, S.161)

In dieser Arbeit wird der Befehl *read_excel* zum Einlesen der 19 kontrollierten Vokabulare angewendet, da diese in einer Excel-Datei gespeichert sind. Natürlich kann man auch die Excel-Dateien in CSV-Dateien konvertieren, um die Funktion *read_csv* zu verwenden. Aber zwingend notwendig ist die Konvertierung zur Zusammenführung der kontrollierten Vokabulare nicht.

In der Python-Bibliothek Pandas wird das kontrollierte Vokabular aus 19 Gruppen einzeln eingelesen. Dabei zeigt sich im DataFrame die Struktur der Datei. Folgende Abbildung stellt die Struktur des Vokabulars aus der Gruppe C10 beispielhaft dar:

	synonym (gestemmt)	synonym (original)	topic	topic_art	Quelle	Quellentyp	Quellen Erklärung
0	bundespressekonferenz	bundespressekonferenz	Ankündigungen	Others	wortliste	sortiert nach	häufigkeit
1	flugblatt	flugblatt	Ankündigungen	Others	wortliste	sortiert nach	häufigkeit
2	interview	interview	Ankündigungen	Others	wortliste	sortiert nach	häufigkeit
3	jubilaumsveranstalt	jubiläumsveranstaltung	Ankündigungen	Others	wortliste	sortiert nach	häufigkeit
4	kommenti	kommentieren	Ankündigungen	Others	wortliste	sortiert nach	häufigkeit

Abbildung 10: Die Struktur eines kontrollierten Vokabulars in DataFrame

Um grundlegende Informationen der Arbeitsmappe zu erhalten, kann der Befehl *df_t.shape g*enutzt werden. Mit *df.dtypes* kann der Datentyp von allen Elementen angezeigt werden. Eine vollständige Zusammenfassung kann mit dem Befehl *df.info()* angefordert werden.

Konsolidierungsbereich eingrenzen

Wie in Abbildung 7 und 10 dargestellt, enthält die kontrollierte Vokabular-Datei mehrere Spalten mit unterschiedlichen Namen wie *Synonym (Original), Synonym (gestemmt)*, *topic, topic-art, Quelle, Bemerkung* usw. Nach Betrachtung der kontrollierten Vokabulare kann festgelegt werden, dass nur bei drei Spalten nötig ist, diese zusammenzuführen. Diese sind Synonym (gestemmt), Synonym (original) und Topic. Unter der Spalte *Synonym (gestemmt)* sind gestemmte Formen der Begriffe gelistet. Das bedeutet, dass ein Vokabular durch eine Programmiersprache automatisch gestemmt wird. Zum Beispiel lautet das Original *Flüchtling.* Entsprechend gestemmt lauten *flucht* oder *fluch*. In der Spalte *Synonym (original)* wird der Begriff im Original gezeigt, zum Beispiel „Flüchtling“ und „Terroranschlag“. In der Spalte *Topic* wird dann angezeigt, zu welchem Topic ein Begriff gehört, zum Beispiel ist „Finanzen“ dem Topic *Aktie* zugeordnet. Diese drei Spalten enthalten alle relevanten Informationen der ursprünglichen Vokabulare und stellen alle wichtigen Inhalte im finalen konsolidierten Vokabular dar. Deswegen werden *Synonym (gestemmt), Synonym (original)* und *Topic* als Konsolidierungsbereiche ausgewählt.

Während der Eingrenzung des Konsolidierungsbereichs wird eine Bereinigung von Duplikaten mit der Funktion *df.drop_duplicates(subset=["synonym (gestemmt)", "synonym (original)", "topic"])* durchgeführt. An dieser Stelle wird von einer Entfernung von Duplikaten in einem einzelnen Vokabular gesprochen.

Sheet-Name definieren

Nach Eingrenzung des Konsolidierungsbereichs in der interaktiven Entwicklungsumgebung Jupyter Notebook soll eine neue Spalte mit einem neuen Sheet-Namen hinzugefügt werden, um klar zu zeigen, aus welcher Gruppe das neu gespeicherte Vokabular und die Begriffe im finalen konsolidierten Vokabular stammen. Sheet-Name bedeutet den Namen der Arbeitsblätter des kontrollierten Vokabulars (Sweigart 2016, S. 312). In diesem Schritt wird eine neue Spalte im DataFrame hinzugefügt und schließlich ein

Sheet-Name zu jeder Zeile in der Datei definiert. Als Sheet-Name werden in dieser Arbeit die Gruppennamen des Projekts verwendet, um die Herkunft der Begriffe klar zu kennzeichnen. Da zu jeder Gruppe eine Spalte des Sheet-Namens eingefügt werden muss, werden 19 kontrollierte Vokabulare einzeln bearbeitet und der Gruppenname als Sheet-Name für den Konsolidierungsbereich jedes Vokabulars eingefügt.

Folgende Abbildung 11 und Listing 1 zeigen als Beispiel das Zwischenergebnis des Vokabulars von Gruppe C10 nach Eingrenzung des Konsolidierungsbereichs und Hinzufügen des Sheet-Namens.

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	C10	bundespressekonferenz	bundespressekonferenz	Ankündigungen
1	C10	flugblatt	flugblatt	Ankündigungen
2	C10	interview	interview	Ankündigungen
3	C10	jubilaumsveranstalt	jubiläumsveranstaltung	Ankündigungen
4	C10	kommenti	kommentieren	Ankündigungen

Abbildung 11: Zwischenergebnis der Vorbereitung für Gruppe C10

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 430 entries, 0 to 432
Data columns (total 4 columns):
synonym (gestemmt)      430 non-null object
synonym (original)      430 non-null object
topic                   430 non-null object
Gruppe                  430 non-null object
dtypes: object(4)
memory usage: 10.1+ KB
```

Listing 1: Zusammenfassung der Vorbereitung für Gruppe C10

Für den Konsolidierungsbereich der Datei der Gruppe C10 wird eine neue Spalte mit dem Namen „Gruppe“ hinzugefügt. Alle Zeilen in dieser Spalte haben den gleichen Inhalt „C10“ als Ursprung der Begriffe. Nach Eingrenzung des Konsolidierungsbereichs und Hinzufügen des Sheet-Namens umfasst die Datei vier Spalten und insgesamt 430 Einträge. Diese werden in einer neuen Excel-Datei gespeichert.

Alle 19 Vokabular-Dateien des Projekts werden also einzeln bearbeitet. Die ausgewählten Spalten werden dann mit der Funktion pd.ExcelWriter() in einer neuen Excel-Datei gespeichert. Somit werden 19 neue Dateien generiert, die jeweils nur die drei wichtigsten Spalten aus den ursprünglichen kontrollierten Vokabularen und eine extra neu hinzugefügte Spalte enthalten. Sie sind synonym (gestemmt), synonym (Original), topic und Gruppenname.

4.3.2 Zusammenführung und Zusammenfassung

Nach der oben beschriebenen Vorbereitung folgt die Zusammenführung der 19 neu generierten Excel-Dateien. Die Funktion *pd.concat()* ermöglicht die Zusammenführung verschiedener DataFrames. Das heißt, dass die 19 neu gespeicherten Excel-Dateien nur mit vier Spalten in dem Jupyter-Notebook eingelesen und zusammengeführt werden.

Die 19 neuen Dateien werden zunächst je nach Gruppe (A, B und C) in drei Dateien (Gruppe A, Gruppe B und Gruppe C) zusammengeführt. Zum Beispiel werden die Vokabulare aus Gruppe A01, A02, A03, A04, A06 und A10 in einer neuen Datei *Gruppe A* zusammengeführt und abgespeichert. Genau wie für Gruppe A wird eine zusammengeführte Datei für die Gruppen B und C generiert. Somit lassen sich Ähnlichkeiten und Unterschiede bei den Vokabularen der drei großen Gruppen beobachten. Anschließend wird eine Zusammenführung der kontrollierten Vokabulare aus den Gruppen A, B und C durchgeführt.

Die 19 Dateien können auch auf einmal direkt zusammengeführt werden, was das gleiche Ergebnis bei der Zusammenführung liefert. Nach der Zusammenführung der 19 Dateien lässt sich das Ergebnis mit dem Befehl *pd.ExcelWriter()* wieder in einer neuen Excel-Datei namens *Phase1.xlsx* speichern und anschließend weiterverarbeiten.

Mit der Funktion *df.info()* lässt sich die Zusammenfassung der zusammengeführten Datei anzeigen.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9642 entries, 0 to 432
Data columns (total 4 columns):
Gruppe                  9642 non-null object
synonym (gestemmt)      9513 non-null object
synonym (original)      9641 non-null object
topic                   9578 non-null object
```

```
dtypes: object(4)
memory usage: 226.0+ KB
```

Listing 2: Zusammenfassung der zusammengeführten kontrollierten Vokabulare

Die zusammengeführte Datei enthält insgesamt 9642 Zeilen und vier Spalten. Alle Elemente weisen den gleichen Datentyp *string* auf. Hierbei wird die erste Phase der Konsolidierung als fertig erachtet. Gemäß den Regelungen und Prinzipien des kontrollierten Vokabulars muss in der nächsten Phase der Konsolidierung die zusammengeführte Datei noch auf Duplikate und Eindeutigkeit der Begriffe kontrolliert werden, damit sie schließlich in ein gutes kontrolliertes Vokabular umgewandelt werden kann.

4.4 Datenaufbereitung

Nachdem gezeigt wurde, wie die kontrollierten Vokabulare zusammengeführt werden, werden in diesem Unterkapitel die Daten in der zusammengeführten Datei aufbereitet, was die zweite Phase der Konsolidierung darstellt. Folgende Abbildung zeigt die Infobox der zweiten Phase der Konsolidierung.

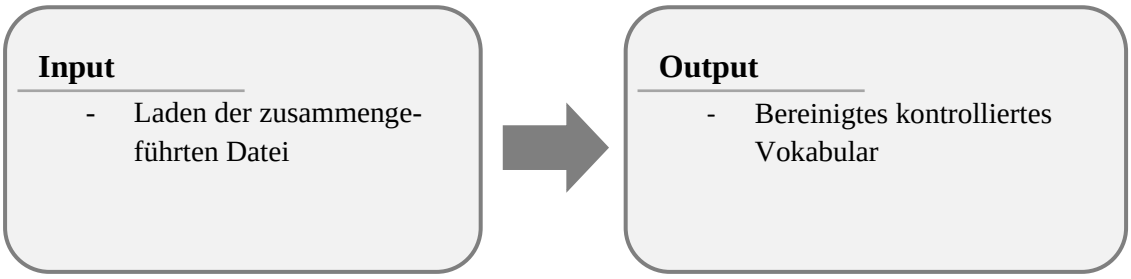

Abbildung 12: Input und Output von Phase II

Während der Eingrenzung des Konsolidierungsbereichs wurde die Datei bereits auf Duplikate in den einzelnen ursprünglichen Vokabularen kontrolliert. An dieser Stelle muss die zusammengeführte Datei auf Duplikate, nämlich zwischen den 19 ursprünglichen Vokabularen, kontrolliert werden.

Zunächst wird erklärt, wie die doppelten Einträge in einem DataFrame und leere Zeilen behandelt werden. Anschließend wird gezeigt, wie oft verschiedene Topics in der zusammengeführten Datei auftauchen. Dabei werden Mehrdeutigkeiten der kontrollierten Vokabulare herausgefiltert. Zum Schluss wird gezeigt, wie die Mehrdeutigkeit von Begriffen behoben wird.

In der folgenden Tabelle werden die Funktionen zur Anweisung der Python-Bibliothek Pandas aufgelistet, die für die weitere Verarbeitung der kontrollierten Vokabulare in dieser Arbeit sehr wichtig sind.

Funktion	**Beschreibung**
df.set_index()	Indexierung
df.duplicated	Anzeigen von Duplikaten
drop_duplicates()	Entfernung von Duplikaten
dropna	Prüfung nach fehlenden Daten
isnull	Anzeige der fehlenden Daten
value_counts()	Anzeige der Häufigkeit in einer Reihenfolge
isin()	Anzeige bestimmter Informationen

Tabelle 3: Funktion zur Aufbereitung kontrollierter Vokabulare (McKinney 2015, S. 147ff.).

Um einen bestimmten Wert in DataFrame abzurufen, muss zunächst ein einheitlicher Index festgelegt werden. Im DataFrame der Python-Bibliothek Pandas können ein oder mehrere Spalten als Index sein. Damit wird die Auffindbarkeit von Informationen versichert.

4.4.1 Duplikat und Lücken entfernen

Nach der Zusammenführung der 19 kontrollierten Vokabulare im Projekt enthält die Excel-Datei insgesamt 9642 Zeilen. Es lässt sich erkennen, dass nach dem Einlesen der Excel-Datei ins DataFrame doppelte Einträge auftauchen. Das bedeutet, dass bestimmte Begriffe in mehreren ursprünglichen kontrollierten Vokabularen gleichzeitig vorhanden sind. Um die Datei von doppelten bzw. mehrfach vorhandenen Einträgen zu bereinigen kann man die Funktion *drop_duplicated* benutzen. Während der Eingrenzung des Konsolidierungsbereichs wurden die Zeilen bereits miteinander im einzelnen Vokabular verglichen und von Duplikaten bereinigt, und zwar mit der Bedingung gleiche Begriffe in „Synonym(original)“, „Synonym(gestemmt)“ und „Topic“. Hierbei wird der Befehl *df.drop_duplicates(subset=["synonym (gestemmt)", "topic"])* angewendet. Diese Bedingung bedeutet, dass, wenn Einträge in den Spalten „Synonym(original)“ und „Topic“

zu 100% identisch sind, dann wird nur ein Eintrag beinhaltet und die Anderen werden entfernt.

Mit der Funktion *df.duplicat* werden duplizierte Inhalte in Zeilen angezeigt. Diese Funktion zeigt doppelte Zeilen an und liefert eine Series vom Typ boolean. Die Zeile wird dann gelöscht, wobei der Wert *True* liefert (McKinney 2015, S. 201).

Dabei kann es vorkommen, dass in einer Zeile kein Wert existiert. Das heißt, dass es Lücken in der Datei gibt. Um solche fehlenden Daten zu behandeln gibt es zwei Möglichkeiten. Man kann entweder mit dem Befehl *df.fillna* die Lücken auffüllen oder die Zeile mit den fehlenden Daten herausfiltern und löschen (McKinney 2015, S. 148). In dieser Arbeit werden die Zeilen ohne Daten gelöscht.

Nach der Entfernung von Duplikaten und leeren Zeilen gibt es in der zusammengeführten Datei immer noch 6480 Zeilen.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6480 entries, 0 to 412
Data columns (total 4 columns):
Gruppe                  6480 non-null object
synonym (gestemmt)      6480 non-null object
synonym (original)      6480 non-null object
topic                   6480 non-null object
dtypes: object(4)
memory usage: 151.9+ KB
```

Listing 3: Zusammenfassung nach Entfernung von Duplikaten

4.4.2 Umbenennung der Topics

Nachdem die zusammengeführte Datei von Duplikaten und fehlenden Daten bereinigt wurde, muss geprüft werden, ob die Beziehung zwischen Begriff und Topic in der zusammengeführten Datei eindeutig ist.

Mit der Funktion *value_counts()* lässt sich die Häufigkeit der Topics in einer absteigend sortierten Reihenfolge beobachten. Diese Funktion lässt sich auch nur für die Spalten *Synonym (Original)* und *Synonym (gestemmt)* benutzen. In den folgenden Tabellen werden die 10 meisten aufgetauchten Topics aufgeführt.

Topic	***Anzahl***
Sicherheit	356
Finanzen	334
Wirtschaft	318
Soziales	291
Demokratie und Bürgerrechte	291
Verkehr und Infrastruktur	263
Umwelt	260
Arbeit	258
Inneres und Justiz	252
Bildung und Forschung	175

Tabelle 4: Häufigkeitstabelle der Topics in der zusammengeführten Datei

Das am häufigsten vorkommende Topic ist „Sicherheit“. Unter dem Topic „Sicherheit“ sind 356 Begriffe zugeordnet. Das Topic „Sicherheit“ hat umfangreiche Bedeutungen und kann sich auf viele Themengebiete beziehen. Es stellt sich die Frage, ob der Topic „Sicherheit“ für die 356 Begriffe präzise genug ist.

Sicherheit bezeichnet einen Zustand, in dem keine Gefahr für jemanden besteht (Götz, Haensch und Wellmann 2003, S. 932). Zum Beispiel haben die Begriffe „kriminell“ und „Datensatz“ unter dem Topic „Sicherheit“ in der zusammengeführten Datei zwar in einer gewissen Art und Weise mit Sicherheit zu tun, jedoch kann man noch näher darauf eingehen.

Unter dem Topic „Sicherheit“ lassen sich noch weitere Kategorien auflisten, zum Beispiel die Kategorien „Aufsicht“, „Bergrettung“, „Brandbekämpfung und Brandschutz“, „Datenschutz“, „Grenzschutz“ usw. (European Dictionary of Skills and Competences). Deswegen ist es sinnvoll, manuell zu prüfen, ob es nötig ist, im finalen kontrollierten Vokabular das Topic „Sicherheit“ für bestimmte Begriffe weiter einzugrenzen, damit die Beziehung zwischen Begriff und Topic eindeutig und präzise ist.

Schritt 1:

Die Begriffe unter dem Topic „Sicherheit“ werden extrahiert und in einer Output-Datei gespeichert, um die Begriffe manuell zu überprüfen. Im Folgenden werden die 356 Begriffe mit dem Topic „Sicherheit“ dargestellt.

Gruppe	synonym (gestemmt)	synonym (original)	topic
A06	abrust	abrüstung	Sicherheit
A06	brandschutz	brandschutz	Sicherheit
A06	brandschutzanlag	brandschutzanlage	Sicherheit
A06	datenschutz	datenschutz	Sicherheit
A06	einsatzort	einsatzort, einsatzorte	Sicherheit
A06	freiheit	freiheit, freiheiten, freiheitlichen	Sicherheit
A06	fried	frieden, friedens, friede	Sicherheit
A06	friedlich	friedlichen, friedliches, friedliche, friedlicher, friedlichere	Sicherheit
A06	polizei	polizei	Sicherheit
A06	polizeieinsatz	polizeieinsätze	Sicherheit
A06	schutz	schutz, schützen, schutzes	Sicherheit
A06	spezialeinsatzkommando	spezialeinsatzkommando	Sicherheit
A06	straftat	straftat, straftaten	Sicherheit
A06	verbot	verbot, verbote, verboten, verbotene, verbotenes, verbotenen, verbotenem	Sicherheit
A06	waffenhandel	waffenhandel	Sicherheit
A06	waffenliefer	waffenflieferungen, waffenlieferung	Sicherheit
A03	abc-waff	ABC-Waffen	Sicherheit
A03	amnesti	Amnestie	Sicherheit
A03	anarchi	Anarchie	Sicherheit
A03	angriff	Angriff, Angriffe, Angriffen	Sicherheit
A03	anschlag	anschlag	Sicherheit

Abbildung 13: Begriffe unter dem Topic „Sicherheit“

Schritt 2:

In der Excel-Datei kann man die Bedeutung der Begriffe im Einzelnen betrachten und ein passendes neues Topic definieren. Im Folgenden wird ein Thesaurus einiger Vokabulare beispielhaft dargestellt, mit dessen Hilfe man die Begriffe entsprechenden neuen Topics zuordnen kann.

Kategorie	***Unterbegriffe***	***Oberbegriff***
Attentat, Anschlag	Terroranschlag	Aktion
	Terrorattentat	Handlung
	Bombenanschlag	Operation
	Bombenattentat	
	Sprengstoffanschlag	
	Attacke	
Militär, Armee, Streitkräfte, Streitmacht, Truppe, Wehr	Landwehr	Anstalt
	Bundeswehr	Einrichtung
	Seestreitkraft	
	Flugwaffe	
	Luftstreitkraft	
	Heer	
usw.		

Tabelle 5: Ein Thesaurus-Beispiel

Schritt 3:

Nach Betrachtung der Bedeutung sowie des Themengebiets der Begriffe kann das Topic „Sicherheit“ umbenannt werden, zum Beispiel für die Begriffe „datenschutz“ und „anschlag“ wird das Topic „Sicherheit“ in die Topics „Datenschutz“ und „Anschlag“ manuell geändert. Alle 356 Begriffe unter dem Topic „Sicherheit“ werden auf diese Weise geprüft. Einige Begriffe werden verworfen, weil sie nicht wertvoll sind, zum Beispiel „Autodieb“. Während der Bearbeitung wird noch manuell geprüft, ob ähnliche Formen einiger Begriffe wie „absicher“ und „absichert“ vorhanden sind. Es soll überlegt werden, ob beide Begriffe beibehalten werden sollen.

„Sicherheit“ ist nicht das einzige Topic, das manuell kontrolliert werden soll. Man kann immer rhetorische Frage stellen, die klären sollen, ob das Topic eindeutig zu den zugeordneten Begriffen passt.

Nach der manuellen Bearbeitung des Topics „Sicherheit“ wird eine neue Zusammenführung durchgeführt. Die neu bearbeitete Datei mit neuen definierten Topics wird mit der Datei ohne Topic „Sicherheit“ zusammengeführt und dann in einer neuen Datei gespeichert.

Nach der Zusammenführung werden noch einmal Duplikate entfernt, da eine Umbenennung des Topics wieder Duplikate verursachen kann. Schließlich gibt es 6432 Zeilen

und vier Spalten in der zusammengeführten Datei. Im Folgenden zeigt die Zusammenfassung der zusammengeführten Datei. Diese Datei wird als *Phase2-2.xlsx* gespeichert, die dann weiterbearbeitet wird.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6432 entries, 0 to 232
Data columns (total 4 columns):
Gruppe                  6432 non-null object
synonym (gestemmt)      6432 non-null object
synonym (original)      6432 non-null object
topic                   6432 non-null object
dtypes: object(4)
memory usage: 251.3+ KB
```

Listing 4: Zusammenfassung konsolidierter kontrollierter Vokabulare

Im Vergleich zu dem Topic „Sicherheit“ tauchen einige Topics nur einmal auf. Das heißt, dass nur ein Begriff diesem Topic zugeordnet wurde. Entsprechende Begriffe sollte man deshalb manuell betrachten und prüfen, ob es sinnvoll ist, diese Begriffe zu behalten. Ein Beispiel davon ist der Begriff „Flüchtling“. Das Wort „Flüchtling“ als Topic ist nur einmal in der Datei vorgekommen. In dieser Arbeit werden die einmal aufgetauchten Topics möglichst beibehalten, sofern die Topics sinnvoll sind.

4.4.3 Mehrdeutigkeit

Gemäß den Prinzipien des kontrollierten Vokabulars sollen Mehrdeutigkeiten im Vokabular vermieden werden. Dieses Unterkapitel wird sich mit Mehrdeutigkeiten beschäftigen. Mehrdeutigkeit bedeutet hierbei, dass ein Begriff unter verschiedenen Topics mehrfach gelistet ist. Zunächst wird der Begriff „Flüchtling“ als Beispiel betrachtet.

Mit der Methode *isin()* wird angezeigt, dass der Begriff „Flüchtling“ oder „fluchtling“ neun Mal unter dem „synonym (gestemmt)“ auftaucht. „fluchtling“ tauchte nur einmal unter dem Topic „Flüchtling“ auf. Wie folgende Abbildung zeigt, ist „fluchtling“ mehreren Topics zugeordnet. Das bedeutet, dass der Begriff „Flüchtling“ mehrdeutig ist.

```
df[df['synonym (gestemmt)'].isin(['fluchtling'])]
```

	Gruppe	synonym (gestemmt)	synonym (original)	topic
204	A06	fluchtling	flüchtlinge, flüchtlingen, flüchtling	Flucht
41	A02	fluchtling	flüchtlinge, flüchtlingen	Flüchtling
51	A02	fluchtling	flüchtlinge	Flüchtlingspolitik
180	A03	fluchtling	flüchtlinge, flüchtlingen, flüchtlings	Migration und Integration
45	A04	fluchtling	flüchtlinge, flüchtlingen, flüchtling	Integration
95	A10	fluchtling	flüchtlinge	Zuwanderung
1	B03	fluchtling	flüchtlingen, flüchtling, flüchtings,	Migration
119	C04	fluchtling	Flüchtlinge, Flüchtlingen, Flüchtlings, Flücht...	Integration, Migration und Flüchtlingspolitik
138	C06	fluchtling	Flüchtlinge, Flüchtlingen, Flüchtling	Flüchtlinge

Abbildung 14: „fluchtling“ unter verschiedenen Bezeichnungen

Durch Thesaurus kann „Flüchtling“ in einer bestimmten Reihenfolge angeordnet werden. Hierbei kann man die Bedeutungen aller Topics im Langenscheidts Großwörterbuch im Einzelnen nachschlagen.

Flucht: Man fliegt aus einer Situation (Götz, Haensch und Wellmann 2003, S. 361).

Flüchtling: Wegen des Krieges muss man seine Heimat und sein Land verlassen (Götz, Haensch und Wellmann 2003, S. 361).

Flüchtlingspolitik: Flüchtlinge betreffende Politik.

Integration: Man zieht in ein neues Gebiet, um dort weiterzuleben (Götz, Haensch und Wellmann 2003, S. 540).

Immigrant: Immigrant ist ein Synonym von Einwanderer. Person, die in ein neues Land kommt und dort weiterlebt (Götz, Haensch und Wellmann 2003, S. 533).

Emigrant: Jemand, der „wegen bedrohlicher wirtschaftlicher, politischer oder religiöser Verhältnisse sein Heimatland verlässt“ (Götz, Haensch und Wellmann 2003, S. 292).

An dieser Stelle wird das Wort *Flucht* als Oberbegriff in Thesaurus ausgewählt. Die Wörter *Flüchtling, Flüchtlingspolitik* und *Wirklichkeitsflucht* werden als Unterbegriffe erstellt.

BT: Flucht (Oberbegriff)

NT: Flüchtling (Unterbegriff)

NT: Flüchtlingspolitik, Wirklichkeitsflucht.

Da „Flucht“ als Topic den Begriff „Flüchtling“ sehr eindeutig bezeichnet und das Topic *Flüchtlingspolitik* eine Beziehung mit dem Begriff *Flüchtling* hat, lassen sich die Begriffe unter dem Topic „Flüchtlingspolitik“ auch unter dem Topic „Flucht“ auswechseln. Zu dem Topic „Flucht“ gehören auch die Begriffe „Flüchtlingsorganisation“, „Flüchtlingsausweis“, „Flüchtlingswohn“ usw. in dem kontrollierten Vokabular. Eine Änderung des Topics lässt sich nur manuell vornehmen. Schließlich ist der Begriff „Flüchtling“ nicht mehr wie in der Abbildung 13 dargestellt mehreren Topics, sondern nur dem Topic „Flucht“ zugeordnet.

Ein anderes Beispiel ist der Begriff „abendschul“. Der Begriff „abendschul“ gehört zum Topic „Schule“, gleichzeitig auch zu den Topics „Bildung und Forschung“. Gemäß der Begrifflichkeit und Häufigkeit in der Tabelle wird entschieden, dass „abendschul“ nur unter dem Topic „Bildung und Forschung“ zugeordnet ist.

> abendschul: „Bildungsstätte, an der sich besonders berufstätige Menschen im Abendunterricht weiterbilden“ (Götz, Haensch und Wellmann 2003, S. 4).

Man kann mit dem Befehl *df['synonym (gestemmt)'].value_counts().head(1000)* kontrollieren, welche Begriffe welchen Topics zugeordnet sind. Im Folgenden zeigt die Liste mehrdeutige Begriffe.

nationalsozialismus	7
rechtsstaat	6
bund	6
rechtsextremismus	6
energiew	6
freihandelsabkomm	6
rentenversicher	6
koalitionsvertrag	6
opposition	6
grundgesetz	6
abschieb	6
windenergi	5
bundestag	5
wahlkreis	5
reform	5
koalition	5

Abbildung 15: Mehrdeutigen kontrollierten Vokabulare

Es gibt insgesamt 991 Begriffe, die mindestens zwei oder mehreren unterschiedlichen Topics zugeordnet sind. Zum Beispiel taucht „rentenreform“ sieben Mal auf. Alle 991 mehrdeutigen Begriffe unter „synonymb(gestemmt)“ werden ähnlich wie der Begriff „Flüchtling“ manuell bearbeitet.

Während der Prüfung auf Mehrdeutigkeiten wird auch auf Singular- und Pluralformen sowie unterschiedliche Schreibweisen eines Begriffs oder eines Topics kontrolliert. Dabei wird eine einheitliche Schreibform für jeden Begriff und jedes Topic vorgenommen. Die folgenden Abbildungen 16 und 17 zeigen zwei Beispiele dafür.

Gruppe	synonym (gestemmt)	synonym (original)	topic
A02	ergebnis	Ergebnis	Wahl
A02	stimm	Stimmen	Wahl
A02	abstimm	Abstimmung	Wahl
A02	stimmabgab	Stimmabgabe	Wahl
A06	gewahlt	gewählt, gewählte, gewählten	Wahlen
A06	wahlkreis	wahlkreis, wahlkreisen, wahlkreise, wahlkreises	Wahlen
A06	landtagswahl	landtagswahl	Wahlen
A06	urnengang	urnengang	Wahlen
A06	stimmenabgab	stimmenabgabe	Wahlen
A06	wahlakt	wahlakt	Wahlen
A06	wahlalt	wahlalter	Wahlen
A06	kandidatenauswahl	kandidatenauswahl	Wahlen
B03	wahlplakat	wahlplakat, wahlplakate, wahlplakaten, wahlplakates	Wahl
B03	wahlhelf	wahlhelfer	Wahl
B03	wahlhelferin	wahlhelferin	Wahl
B03	kanzlerkandidatur	kanzlerkandidatur	Wahl
B05	voti	votieren	Wahlen
B05	wahlerinitiativ	wählerinitiativen	Wahlen

Abbildung 16: Singular und Plural als Topic

Gruppe	synonym (gestemmt)	synonym (original)	topic
C05	twitt	Twitter	Demokratie und Bürgerrechte
C05	direktwahl	direktwahlen	Demokratie und Bürgerrechte
C07	mau	mauer	Demokratie & Bürgerrechte
C07	demokrat	demokratischen, demokratische, demokratisch, demokratischer,	Demokratie & Bürgerrechte
C07	demokrati	demokratie	Demokratie & Bürgerrechte
C10	burgergesellschaft	bürgergesellschaft	Demokratie & Bürgerrechte
C10	mitburg	mitbürger	Demokratie & Bürgerrechte
C10	rechtsstaat	rechtsstaat, rechtsstaates, rechtsstaatlichen	Demokratie & Bürgerrechte
A02	demokrat	demokratisch, demokratisches, demokratie	Demokratie

Abbildung 17: Unterschiedliche Formen eines Topics

Es ist schnell festzustellen, dass unterschiedliche Gruppen unterschiedliche Formen für gleiche Topics verwendet haben. Gruppe A06 hat nur die Form „Wahlen“ verwendet, während Gruppe A02 sowohl die Form „Wahl“ als auch die Form „Wahlen“ verwendet

hat. Gruppe C05 schrieb „Demokratie und Bürgerrechte“, während Gruppe C10 „Demokratie & Bürgerrechte“ schrieb. Diese Topics werden in der finalen Datei einheitlich auf „Wahl“ sowie „Demokratie & Bürgerrechte“ geändert. Somit werden alle Topics sowohl auf die Schreibweise als auch auf Singular- und Pluralformen kontrolliert. In der finalen Datei steht für jedes Topic nur eine Schreibform.

4.4.4 Zusammenfassung

Nach der manuellen Bearbeitung der Mehrdeutigkeit wird zur Sicherheit geprüft, ob wieder Duplikate durch Änderungen der Topics in der Datei entstanden sind. Dazu wurde das kontrollierte Vokabular zum Schluss noch einmal in Jupyter Notebook eingelesen. Nach der nochmaligen Entfernung von Duplikaten und der Feststellung, dass es keine weiteren Duplikate in der Datei gibt, konnte die Konsolidierung als abgeschlossen betrachtet werden. Die finale Datei kann man als das finale kontrollierte Vokabular benennen.

4.5 Ergebnis

Die 19 kontrollierten Vokabulare des Projekts „Schaffung von mehr Transparenz 2017“ wurden in zwei Phasen konsolidiert. Das komplette Skript für die Konsolidierung der 19 kontrollierten Vokabulare befindet sich im Anhang. Nach der automatischen und manuellen Bearbeitung sind die Begriffe im finalen kontrollierten Vokabular unter passenden Topics zugeordnet worden. Das finale kontrollierte Vokabular enthält ausschließlich wertvolle Begriffe, die in einer finalen Excel-Datei gespeichert wurden. Da die Datei während der Bearbeitung von vielen doppelten und mehrdeutigen Begriffen bereinigt wurde, spielen die Gruppennamen im finalen kontrollierten Vokabular eigentlich keine wesentliche Rolle mehr, jedoch noch während der Konsolidierung.

Wie im folgenden Listing aufgeführt, enthält das finale kontrollierte Vokabular 4910 Zeilen und vier Spalten.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4910 entries, 1 to 232
Data columns (total 4 columns):
Gruppe                  4910 non-null object
synonym (gestemmt)      4910 non-null object
synonym (original)      4910 non-null object
```

```
topic                    4910 non-null object
dtypes: object(4)
memory usage: 191.8+ KB
```

Listing 5: Zusammenfassung des finalen kontrollierten Vokabulars

Im finalen kontrollierten Vokabular existieren keine Duplikate, mehrdeutige Begriffe oder unterschiedliche Formen des Topics mehr.

5 Zusammenfassung und Ausblick

In dieser Arbeit wurden die theoretischen Grundlagen zu Text Mining, zur Python-Bibliothek Pandas, zu einem kontrollierten Vokabular und zur Konsolidierung erläutert und das Projekt „Schaffung von mehr Transparenz in der Bundestagswahl 2017“ vorstellt. Anschließend wurde dargestellt, wie mit der Python-Bibliothek Pandas die kontrollierten Vokabulare konsolidiert werden können.

Die ursprünglichen 19 kontrollierten Vokabulare des Projekts waren gut erstellt, zwar mit einigen Mängeln, zum Beispiel einigen Duplikaten, mehrdeutigen Begriffen und Topics sowie ungeeigneten Zuordnungen. Die Struktur der 19 kontrollierten Vokabulare war aber sehr gut dargestellt und für die Zusammenführung sehr hilfreich.

Für diese Arbeit ist die Python-Bibliothek Pandas sehr gut geeignet. Die Vorgehensweise der Konsolidierung in dieser Arbeit wurde aus eigenem Verständnis für die Konsolidierung kontrollierter Vokabulare durchgeführt, weil es kaum Literatur zur Konsolidierung kontrollierter Vokabulare gibt.

Bei der Konsolidierung in dieser Arbeit wurden die Mängel der ursprünglichen 19 kontrollierten Vokabulare je nach Möglichkeit maschinell oder manuell behoben. Diese Arbeit konzentriert sich nicht auf die Verbesserung der Qualität der 19 originalen Vokabulare des Projekts, sondern auf die Konsolidierung der 19 Vokabulare. Aufgrund unbekannter Anlässe der Studierenden für manche Zuordnungen der Begriffe zu den Topics wurden die Zuordnungen der originalen Vokabulare im finalen kontrollierten Vokabular möglichst beibehalten.

In dieser Arbeit wurde die Reihenfolge der Konsolidierungsschritte festgehalten, zum Beispiel sollte eine Bearbeitung von Mehrdeutigkeiten nach der Umbenennung von Topics folgen, da eine Umbenennung der Topics Einfluss auf Mehrdeutigkeiten hat. Die meisten mehrdeutigen Begriffe und Duplikate gab es nach der Zusammenführung aller Dateien. Das finale konsolidierte kontrollierte Vokabular beinhaltet alle relevanten Daten und Informationen der 19 kontrollierten Vokabulare und weist eine gute Qualität eines kontrollierten Vokabulars auf, womit das Ziel der vorliegenden Arbeit erreicht wurde.

Eine Anwendung der in dieser Arbeit dargestellten Vorgehensweise der Konsolidierung von kontrollierten Vokabularen in anderem Projekt ist möglich. Die Python-Bibliothek Pandas wird als Tool zur Durchführung der Konsolidierung von kontrollierten Vokabularen sehr empfohlen.

In der Regel liegt der Schwerpunkt der Konsolidierung im Bereinigen der Daten in der zusammengeführten Datei. Im Rahmen dieser Arbeit wurden Zeilen mit fehlenden Daten gelöscht, in anderen Projekten jedoch könnten die Werte auch aufgefüllt werden. Einige Schritte in dieser Arbeit wie die Umbenennung der Topics werden vielleicht projektbedingt je nach Bedarf entfallen oder könnten an die Reihfolge der Schritte angepasst werden. Wenn bei Projekt diese Schritte aufgrund Hyponymen und Synonymen im finalen Vokabular jedoch erforderlich sind, könnten die Änderungen an Begriff mit hoher Wahrscheinlichkeit nur manuell durchgeführt werden.

Als wichtigste Ansatzpunkte für zukünftige Entwicklungen des Konsolidierungsverfahrens sollte man noch weitere Ergänzungsschritte versuchen bzw. neue Hilfsmitteln für Text Mining entwickeln, um die maschinellen Änderungen statt manuellen Änderungen an Begriffe in dieser Arbeit möglichst zu realisieren.

Literaturverzeichnis

BIRD, Steven, 2006. NLTK: the natural language toolkit. In: *Proceeding of COLING/ACL on interactive presentation sessions (COLING-ACL*'06*).* S. 69-72. [Zugriff am 11.10.2017]. Association for Computational Linguistics. Verfügbar unter: 10.3115/1225403.1225421

CARSTENSEN, Kai-Uwe, EBERT, Christian, EBERT, Cornelia, JEKAT, Susanne, KLABUNDE, Ralf und LANGER, Hagen. 2010. *Computerlinguistik und Sprachtechnologie: eine Einführung.,* 3. überarbeitete und erweiterte Auflage. Heidelberg: Spektrum. ISBN 978-3-8274-2023-7

DATENSCHULE, 2017. *Datenschule*. Berlin: Datenschule [Zugriff am 11.10.2017]. Verfügbar unter: https://datenschule.de/

EUROPEAN DICTIONARY OF SKILLS AND COMPETENCES. *Thesaurus Explorer*. [Zugriff am 11.10.2017]. Verfügbar unter: http://discotools.eu/disco2_portal/terms.php

INSTITUT FÜR DEUTSCHE SPRACHE (IDS), 2012. *COSMAS II* [Online] *Korpora.* Mannheim: Das Institut für Deutsche Sprache (IDS). [Zugriff am 11.10.2017], Verfügbar unter: http://www.ids-mannheim.de/cosmas2/projekt/referenz/korpora.html

GENTSCH, Peter, 1999. Business Intellegence: Aus Daten systematisch Wissen entwickeln. In: August-Wilhelm SCHEER, Hrsg. *Electronic Business und Knowledge Management-Neue Dimensionen für den Unternehmenserfolg: 20. Saarbrücker Arbeitstagung 1999 für Industrie, Dienstleitung und Verwaltung.* Berlin: Springer-Verlag, S. 167-195. ISBN 978-3-642-63686-8

GLISSMANN-HOCHSTEIN, Susanne, 2017. *Projekt: Schaffung von mehr Transparenz in der Bundestagswahl 2017* [PowerPoint-Präsentation]. Hamburg: Prof. Dr. Glissmann-Hochstein, 13.05.2017

GÖTZ, Dieter, HAENSCH, Günther und WELLMANN, Hans, 2003. *Großwörterbuch Deutsch als Fremdsprache: Langenscheidt*. Berlin: Langenscheidt KG. ISBN 978-3-468-49036-1

G.STOCK, Wolfgang, 2007. *Information Retrieval: Informationen suchen und finden.* Oldenbourg: Wissenschaftsverlag GmbH. ISBN 3-486-58172-4

HAAS, Hans-Dieter, und NEUMAIR, Simon Martin, 2010. *Internationale Wirtschaft* [Online]. *Rahmenbedingungen, Akteure, räumliche Prozesse.* Oldenbourg: Wissenschaftsverlag ISBN 978-3-486-70018-3 [Zugriff am 18.09.2017]. Verfügbar unter: https://www.degruyter.com/view/product/222481

HANS-BÖCKLER-STIFTUNG, 2011. *Der Konzernabschluss nach Handelsgesetzbuch (HGB) und International Financial Reporting Standards (IFRS). Der Konzernlagebericht nach DRS 20* [Online]. Hans-Böckler-Stiftung [Zugriff am 18.09.2017]. Verfügbar unter: https://www.boeckler.de/pdf/mbf_konzernabschluss_gesamt.pdf

HEYER, Gerhard, QUASTHOFF, Uwe und WITTIG, Thomas, 2012. *Text Mining: Wissensrohstoff Text. Konzept, Algorithmen, Ergebnisse.* Herdecke:, W3L GmbH. ISBN 3-937137-30-0

HIPPNER, Hajo und RENTZMANN, René, 2006. *Text Mining*. In: Informatik Spektrum [Online]. 29(4), S. 287-290 [Zugriff am: 17.10.2017]. ISSN 1432-122X. Verfügbar unter: DOI: 10.1007/s00287-006-0091-y

KRAPP, Michael, 2003. *Mit Text-Mining gegen die Informationsflut in der „Life Science“* [Online]. Nordrhein-Westfalen: Institut für Algorithmen und Wissenschaftliches Rechnen (SCAI) [Zugriff am: 11.10.2017]. Verfügbar unter: https://idw-online.de/de/news69535

LEWANDOWSKI, Dirk, Hrsg., 2011. Handbuch Internet-Suchmaschinen 2: Neue Entwicklungen in der Web-Suche. Heidelberg: Akademische Verlagsgesellschaft AKA GmbH. ISBN 978-3-89838-651-7

LEMKE, Matthias und WIEDEMANN, Gregor, 2016 *Text Mining in den Sozialwissenschaften*: *Grundlagen und Anwendungen zwischen qualitativer und quantitativer Diskursanalyse*. Wiesbaden: Springer VS. ISBN 978-3-658-07223-0

LEMNITZER, Lothar und ZINSMEISTER, Heike, 2015. *Korpuslinguistik: Eine Einführung. Narr Studienbücher*., 3. überarbeitete und erweiterte Auflage. Tübingen: Narr Francke Attempto Verlag. ISBN 978-3-8233-6886-1

MICROSOFT AZURE, 2016. *Jupyter Notebook in Azure*. [Zugriff am 18.09.2017] Verfügbar unter: https://github.com/Azure/azure-content-dede/blob/master/articles/virtual-machines/virtual-machines-linux-jupyter-notebook.md

MCKINNEY, Wes 2012. *Python for Data Analysis: Agile Tolls for Real World Data.* 1. Auflage. Sebastopel: O´Reilly Media, Inc. ISBN 978-1-449-31979-3

MCKINNEY, Wes, 2015. Datenanalyse mit Python: Auswertung von Daten mit Pandas, Numpy und Ipython. 1. Auflage Heidelberg: dpunkt.Verlag. ISBN 978-3-96009-000-7

NATIONAL INFORMATION STANDARDS ORGANIZATION (NISO), 2010, *Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularie* [Online]. Baltimore: National Information Standards Organization. ISBN: 978-1-937522-22-3. Verfügbar unter: http://www.niso.org/apps/group_public/download.php/12591/z39-19-2005r2010.pdf

PARLAMENTWATCH E.V., 2017. *Abgeornetenwatch.de*. Hamburg: Parlamentwatch.de [Zugriff am 11.10.2017]. Verfügbar unter: https://www.abgeordnetenwatch.de/node/7760

PARMA, David, 2016. *Installation und Konsolidierung des Bundesgrenzschutzes 1949 bis 1972 [Online]. Eine Untersuchung der Gesetzgebungsprozesse unter besonderer Betrachtung der inneradministrativen und politischen Vorgänge.* Wiesbaden: Springer [Zugriff am 11.10.2017]. ISBN 978-3-658-10927-1. Verfügbar unter: DOI 10.1007/978-3-658-10928-8

PELLEGRINI, Tassilo und BLUMAUER, Andreas 2006. *Semantic Web: Wege zur vernetzten Wissensgesellschaft.* Berlin: Springer. ISBN 3-540-29324-8

ROTH, Philipp, 2017. *Offizielle Facebook Nutzerzahlen für Deutschland* [Online]. München: Rising Media Ltd. [Zugriff am: 11.10.2017]. Verfügbar unter: https://allfacebook.de/zahlen_fakten/offiziell-facebook-nutzerzahlen-deutschland

SACHSE, Katja, 2016. *Konsolidierung eines umgekehrten Unternehmenserwerbs nach IFRS* [Online]. *Erstkonsolidierung und Änderung bestehender Beherrschungs- und Beteiligungsverhältnisse.* Wiesbaden: Springer Gabler [Zugriff am 11.10.2017]. PDF e-Book. ISBN 978-3-658-14754-9. Verfügbar unter: DOI 10.1007/978-3-658-14755-6

SEIDENFADEN, Lutz, 2007. *Ein Peer-to-Peer-basierter Ansatz zur digitalen Distribution wissenschaftlicher Informationen* [Dissertation]. Universität Göttingen. Göttingen: Cuvillier Verlag. ISBN: 978-3-8672-7321-3

SWEIGART, Al, 2016. *Routineaufgaben mit Python automatisieren: Praktische Programmierlösungen für Einsteiger*. 1. Auflage. Heidelberg: dpunkt.verlag. ISBN:978-3-86490-353-3

SULLIVAN, Dan, 2001. *Document Warehousing and Text Mining: Techniques for Improving Business Operations, Marketing, and Sales*. New York: John Wiley & Sons. ISBN: 0471399590

TWITTER, 2017. *Business* [Online]. San Francisco: Twitter, Inc. [Zugriff am: 07.12.2017]. Verfügbar unter: https://business.twitter.com

WARDATZKY, Kathrin, 2016. *Berufliche Kompetenzen in Online-Stellenausschreibungen - Entwicklung einer Methode zur automatischen Identifizierung von Kompetenzen mit der Programmiersprache Python* [Bachelorarbeit]. Hamburg: Hochschule für Angewandte Wissenschaften Hamburg

WEIGEND, Michael, 2016. *Python 3: Lernen und professionell anwenden. Das umfassende Praxisbuch.,* 6. erweiterte Auflage. Frechen: mitp Verlag. ISBN:978-3-95845-425-5

WOLLSCHLÄGER, Daniel 2014. G*rundlagen der Datenanalyse mit R: eine anwendungsorientierte Einführung*. 3. Auflage. Mainz: Springer Spektrum. ISBN 978-3-662-53669-8

WOYAND, Hans-Bernhard 2017. *Python für Ingenieure und Naturwissenschaftler: Einführung in die Programmierung, mathematisches Anwendungen und Visualisierungen*. München: Hanser. ISBN 978-3-446-45198-8

Anhang 1: Beigabe (CD)

Inhalt der CD:

1. Die Bachelorarbeit als PDF-Version
2. Excel-Dateien der Konsolidierung von kontrollierten Vokabularen
3. Die Skripte als ipynb-Datei

Anhang 2: Skripte zur Konsolidierung

Anhang 2.1: Phase I - Konsolidierungsbereich_SheetName

```
In [1]: # Pandas und entsprechende Objekte des Pandas importieren
        import os as os
        import pandas as pd
        from pandas import Series, DataFrame
        import numpy as np
        import matplotlib.pyplot as plt

In [2]: os.getcwd()

Out[2]: 'C:\\Users\\Elisa\\Desktop\\Kontrollierte_Vokabulare_und_Skripte\\politicalTopics'

In [3]: # die Dateien im Einzelnen einlesen, geleichen Schritt auch für Gruppe B und Gruppe C
        df=pd.DataFrame(pd.read_excel('C10_Vokabular.xlsx'))

In [4]: df.head()

Out[4]:
```

	synonym (gestemmt)	synonym (original)	topic	topic_art	\
0	bundespressekonferenz	bundespressekonferenz	Ankündigungen	Others	
1	flugblatt	flugblatt	Ankündigungen	Others	
2	interview	interview	Ankündigungen	Others	
3	jubilaumsveranstalt	jubiläumsveranstaltung	Ankündigungen	Others	
4	kommenti	kommentieren	Ankündigungen	Others	

	Quelle	Quellentyp	Quellen Erklärung	Quelle (Abschnitt/URL)	Bemerkung
0	wortliste	sortiert nach	häufigkeit	abschnitt C.6	NaN
1	wortliste	sortiert nach	häufigkeit	abschnitt C.6	NaN
2	wortliste	sortiert nach	häufigkeit	abschnitt C.6	NaN
3	wortliste	sortiert nach	häufigkeit	abschnitt C.6	NaN
4	wortliste	sortiert nach	häufigkeit	abschnitt C.6	NaN

```
In [5]: # gleiche Einträge in "synonym (gestemmt)","synonym (original)","topic" entfernen
        new_df=df.drop_duplicates(subset=["synonym (gestemmt)","synonym (original)",\
                                          "topic"]).ix[:,[0,1,2]]

C:\Users\Elisa\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: DeprecationWarning:
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing
```

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#deprecate_ix

```
In [6]: new_df.head()
```

Out[6]:

	synonym (gestemmt)	synonym (original)	topic
0	bundespressekonferenz	bundespressekonferenz	Ankündigungen
1	flugblatt	flugblatt	Ankündigungen
2	interview	interview	Ankündigungen
3	jubilaumsveranstalt	jubiläumsveranstaltung	Ankündigungen
4	kommenti	kommentieren	Ankündigungen

```
In [7]: # neuen Spalt einfügen
        new_df["total"] = new_df["synonym (gestemmt)"] + new_df["synonym (original)"]\
                          + new_df["topic"]
        new_df.head()
```

Out[7]:

	synonym (gestemmt)	synonym (original)	topic	\
0	bundespressekonferenz	bundespressekonferenz	Ankündigungen	
1	flugblatt	flugblatt	Ankündigungen	
2	interview	interview	Ankündigungen	
3	jubilaumsveranstalt	jubiläumsveranstaltung	Ankündigungen	
4	kommenti	kommentieren	Ankündigungen	

	total
0	bundespressekonferenzbundespressekonferenzAnkü...
1	flugblattflugblattAnkündigungen
2	interviewinterviewAnkündigungen
3	jubilaumsveranstaltjubiläumsveranstaltungAnkün...
4	kommentikommentierenAnkündigungen

```
In [8]: # neuen eingefügten Spalt mit dem Sheet-Name definieren
        new_df['total'] = 'C10'
```

```
In [9]: new_df.head()
```

Out[9]:

	synonym (gestemmt)	synonym (original)	topic	total
0	bundespressekonferenz	bundespressekonferenz	Ankündigungen	C10
1	flugblatt	flugblatt	Ankündigungen	C10
2	interview	interview	Ankündigungen	C10
3	jubilaumsveranstalt	jubiläumsveranstaltung	Ankündigungen	C10
4	kommenti	kommentieren	Ankündigungen	C10

```
In [10]: # Sheet-Name umnennen
         new_df.rename(columns={'total':'Gruppe'}, inplace=True)
```

```
In [11]: new_df.head()
```

Out[11]:

	synonym (gestemmt)	synonym (original)	topic	Gruppe
0	bundespressekonferenz	bundespressekonferenz	Ankündigungen	C10
1	flugblatt	flugblatt	Ankündigungen	C10
2	interview	interview	Ankündigungen	C10
3	jubilaumsveranstalt	jubiläumsveranstaltung	Ankündigungen	C10
4	kommenti	kommentieren	Ankündigungen	C10

In [12]:
```
# die Reihnfolge der 4 Spalte festlegen
cols = list(new_df)
cols.insert(0, cols.pop(cols.index('Gruppe')))
cols
```

Out[12]: ['Gruppe', 'synonym (gestemmt)', 'synonym (original)', 'topic']

In [13]:
```
new_df.ix[:, cols].head()
```

```
C:\Users\Elisa\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: DeprecationWarning:
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#deprecate_ix
  """Entry point for launching an IPython kernel.
```

Out[13]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	C10	bundespressekonferenz	bundespressekonferenz	Ankündigungen
1	C10	flugblatt	flugblatt	Ankündigungen
2	C10	interview	interview	Ankündigungen
3	C10	jubilaumsveranstalt	jubiläumsveranstaltung	Ankündigungen
4	C10	kommenti	kommentieren	Ankündigungen

In [14]:
```
new_df.shape
```

Out[14]: (430, 4)

In [15]:
```
new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 430 entries, 0 to 432
Data columns (total 4 columns):
synonym (gestemmt)    430 non-null object
synonym (original)    430 non-null object
topic                 430 non-null object
Gruppe                430 non-null object
dtypes: object(4)
memory usage: 10.1+ KB
```

```
In [16]: new_df.dtypes

Out[16]: synonym (gestemmt)     object
         synonym (original)     object
         topic                  object
         Gruppe                 object
         dtype: object

In [17]: # ausgewählte Splate in einer neuen Datei speichern.
         out = pd.ExcelWriter('C10.xlsx')
         new_df.to_excel(out)
         out.save()
```

Anhang 2.2: Phase I – Zusammenführung der Dateien je nach Gruppe A, B und C

```
In [1]: # Pandas und entsprechende Objekte importieren
        import os as os
        import pandas as pd
        from pandas import Series, DataFrame
        import numpy as np
        import matplotlib.pyplot as plt

In [2]: os.getcwd()

Out[2]: 'C:\\Users\\Elisa\\Desktop\\Kontrollierte_Vokabulare_und_Skripte\\politicalTopics'

In [3]: # Die neue gespeicherte Datei wie A01, A02, A03, A04, A06
        # und A10 werden im Einzelnen eingelesen
        df=pd.DataFrame(pd.read_excel('A01.xlsx'))
        df.head()
```

Out[3]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	A06	abendschul	abendschule	Schule
1	A06	abgasaffar	abgasaffäre	Wirtschaft
2	A06	abgasbetrug	abgasbetrug	Wirtschaft
3	A06	abgasnorm	abgasnorm, abgasnormen	Umwelt
4	A06	abgasskandal	abgasskandal	Wirtschaft

```
In [4]: df.shape

Out[4]: (721, 4)

In [5]: df1=pd.DataFrame(pd.read_excel('A02.xlsx'))
        df1.head()
```

Out[5]:

	synonym (gestemmt)	synonym (original)	topic	Gruppe
0	leistung	Leistung	Arbeit	A02
1	gewerb	Gewerbe	wirtschaftlicheTätigkeit	A02
2	geschaft	Geschäft	Arbeit	A02
3	handwerk	Handwerk	Handwerk	A02
4	dienstleist	Dienstleistung	Dienstleistung	A02

```
In [6]: df1.shape
```

Out[6]: (247, 4)

In [7]:
```
df2=pd.DataFrame(pd.read_excel('A03.xlsx'))
df2.head()
```

Out[7]:

	synonym (gestemmt)	synonym (original)	topic	Gruppe
0	abc-waff	ABC-Waffen	Sicherheit	A03
1	abgeordnet	Abgeordneter, Abgeordnete	Abgeordnete	A03
2	abschieb	Abschiebung	Integration	A03
3	abschlusspressekonferenz	Abschlusspressekonferenz	Ankündigung	A03
4	absolutismus	Absolutismus	Staatsform	A03

In [8]:
```
df2.shape
```

Out[8]: (765, 4)

In [9]:
```
df3=pd.DataFrame(pd.read_excel('A04.xlsx'))
df3.head()
```

Out[9]:

	synonym (gestemmt)	synonym (original)	topic	Gruppe
0	chef	chef	Arbeit	A04
1	beruf	beruf, berufe, berufung, beruflich, berufen, b...	Arbeit	A04
2	arbeitslos	arbeitslos, arbeitslosen	Arbeit	A04
3	jobcent	jobcenters, jobcenter	Arbeit	A04
4	schul	schule, schulen, schüler, schülern	Bildung und Forschung	A04

In [10]:
```
df3.shape
```

Out[10]: (201, 4)

In [11]:
```
df4=pd.DataFrame(pd.read_excel('A06.xlsx'))
df4.head()
```

Out[11]:

	synonym (gestemmt)	synonym (original)	topic	Gruppe
0	pensionarin	pensionärin	Arbeit	A06
1	pensionar	pensionär	Arbeit	A06
2	pensioniert	pensionierte, pensionierter, pensioniert	Arbeit	A06
3	beruf	beruf, beruflich, beruflichen, beruflicher, be...	Arbeit	A06
4	bewerb	bewerben, bewerber, bewerbern, bewerbung, bewe...	Arbeit	A06

```
1   Arbeit      A06
2   Arbeit      A06
3   Arbeit      A06
4   Arbeit      A06
```

In [12]:
```
df4.shape
```

Out[12]:
```
(748, 4)
```

In [13]:
```
df5=pd.DataFrame(pd.read_excel('A10.xlsx'))
df5.head()
```

Out[13]:

	synonym (gestemmt)	synonym (original)	topic	Gruppe
0	alterssicherungssys	alterssicherungssystem	Soziales	A10
1	amtsvorstand	amtsvorstand	Politisches Amt	A10
2	amtsvorsteh	amtsvorsteher	Politisches Amt	A10
3	amtsweg	amtsweg	Regierungsstruktur	A10
4	amtszeit	amtszeit	Regierungsstruktur	A10

In [14]:
```
df5.shape
```

Out[14]:
```
(390, 4)
```

In [15]:
```
# Die Dateien werden je nach Gruppe A, B und C zusammengeführt.
df_t=pd.concat([df,df1,df2,df3,df4,df5])
df_t.head()
```

Out[15]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	A06	abendschul	abendschule	Schule
1	A06	abgasaffar	abgasaffäre	Wirtschaft
2	A06	abgasbetrug	abgasbetrug	Wirtschaft
3	A06	abgasnorm	abgasnorm, abgasnormen	Umwelt
4	A06	abgasskandal	abgasskandal	Wirtschaft

In [16]:
```
df_t.shape
```

Out[16]:
```
(3072, 4)
```

In [17]:
```
# zusammengeführte Datei für Gruppe A wird gespeichert,
out = pd.ExcelWriter('GruppeA.xlsx')
df_t.to_excel(out)
out.save()
```

In [18]:
```
# gleichen Schritt für Gruppe B und Gruppe C durchführen
```

Anhang 2.3: Phase I – Zusammenführung der Gruppe A, B und C

```
In [1]: # Pandas und entsprechende Objekte importieren
        import os as os
        import pandas as pd
        from pandas import Series, DataFrame
        import numpy as np
        import matplotlib.pyplot as plt

In [2]: os.getcwd()

Out[2]: 'C:\\Users\\Elisa\\Desktop\\Kontrollierte_Vokabulare_und_Skripte\\politicalTopics'

In [3]: # die Dateien von der Gruppe A. B und C einlesen und zusammenführen.

In [4]: df=pd.DataFrame(pd.read_excel('GruppeA.xlsx'))
        df.head()

Out[4]:   Gruppe synonym (gestemmt)        synonym (original)        topic
        0    A06          abendschul                abendschule       Schule
        1    A06           abgasaffar                abgasaffäre  Wirtschaft
        2    A06          abgasbetrug                abgasbetrug  Wirtschaft
        3    A06           abgasnorm  abgasnorm, abgasnormen        Umwelt
        4    A06         abgasskandal              abgasskandal  Wirtschaft

In [5]: df.shape

Out[5]: (3072, 4)

In [6]: df1=pd.DataFrame(pd.read_excel('GruppeB.xlsx'))
        df1.head()

Out[6]:   synonym (gestemmt)                                synonym (original)  \
        0             euro                                              euro
        1       fluchtling                 flüchtlingen, flüchtling, flüchtings,
        2           europa  europäischen, europäische, europäischer, europ...
        3           arbeit  arbeit, arbeiten, arbeiter, arbeitenden, arbei...
        4            schul                schule, schüler, schulen, schülern

                topic  Gruppe
```

```
0                    Europa     B03
1                 Migration     B03
2                    Europa     B03
3                    Arbeit     B03
4  Bildung und Forschung        B03
```

In [7]:
```
df1.shape
```

Out[7]:
```
(3840, 4)
```

In [8]:
```
df2=pd.DataFrame(pd.read_excel('GruppeC.xlsx'))
df2.head()
```

Out[8]:
```
   synonym (gestemmt)                                  synonym (original)  \
0          abgeordnet          abgeordneter, abgeordneten, abgeordnete
1   abgeordnetenhaus            abgeordnetenhaus, abgeordnetenhauses
2             abstimm                          abstimmung, abstimmen
3            amerikan  amerikaner, amerikanisch, amerikanische, ameri...
4              arbeit  arbeit, arbeiten, arbeiter, arbeite, arbeitern...

                          topic Gruppe
0       Institutionelle Fragen     C03
1       Institutionelle Fragen     C03
2  Demokratie und Bürgerrechte     C03
3                Internationales     C03
4                        Arbeit     C03
```

In [9]:
```
df2.shape
```

Out[9]:
```
(2201, 4)
```

In [10]:
```
# Zusammenführung von kontrollierten Vokabularen aus Gruppe A,
# Gruppe B und Gruppe C
df_t=pd.concat([df,df1,df2])
df_t.head()
```

Out[10]:
```
  Gruppe synonym (gestemmt)       synonym (original)      topic
0    A06          abendschul              abendschule     Schule
1    A06          abgasaffar              abgasaffäre  Wirtschaft
2    A06        abgasbetrug              abgasbetrug  Wirtschaft
3    A06          abgasnorm  abgasnorm, abgasnormen      Umwelt
4    A06       abgasskandal             abgasskandal  Wirtschaft
```

In [11]:
```
df_t.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9113 entries, 0 to 432
Data columns (total 4 columns):
Gruppe                  9113 non-null object
```

x

```
synonym (gestemmt)      8984 non-null object
synonym (original)      9112 non-null object
topic                   9050 non-null object
dtypes: object(4)
memory usage: 213.6+ KB
```

```
In [12]: # zusammengeführte Datei wird in Out-Datei "Phase1" abgespeichert.
         out = pd.ExcelWriter('Phase1.xlsx')
         df_t.to_excel(out)
         out.save()
```

Anhang 2.4: Phase II – Entfernung der Duplikate und Lücken

```
In [1]: # Pandas und entsprechende Objekte importieren
        import pandas as pd
        from pandas import Series, DataFrame
        import numpy as np
        import os as os

In [2]: # die zusammengeführte Datei "Phase1" einlesen
        df=pd.DataFrame(pd.read_excel('Phase1.xlsx'))

In [3]: df.head()

Out[3]:
   Gruppe  synonym (gestemmt)        synonym (original)        topic
0     A06          abendschul                abendschule       Schule
1     A06          abgasaffar                abgasaffäre    Wirtschaft
2     A06         abgasbetrug                abgasbetrug    Wirtschaft
3     A06           abgasnorm  abgasnorm, abgasnormen         Umwelt
4     A06       abgasskandal              abgasskandal    Wirtschaft

In [4]: # Zusammenfassung der Datei
        df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 9113 entries, 0 to 432
Data columns (total 4 columns):
Gruppe                  9113 non-null object
synonym (gestemmt)      8984 non-null object
synonym (original)      9112 non-null object
topic                   9050 non-null object
dtypes: object(4)
memory usage: 213.6+ KB

In [5]: # doppelte Einträge vom Spalt "synonym (gestemmt)" ansehen.
        df.duplicated("synonym (gestemmt)").head()

Out[5]: 0    False
        1    False
        2    False
```

```
3     False
4     False
dtype: bool
```

```
In [6]: # Löschen doppelter Begriffe von "synonym (gestemmt)" und "topic",
        # "br" bedeutet in dieser Phase für Bereinigung
        br=df.drop_duplicates(subset=["synonym (gestemmt)","topic"])
```

```
In [7]: br.head()
```

Out[7]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	A06	abendschul	abendschule	Schule
1	A06	abgasaffar	abgasaffäre	Wirtschaft
2	A06	abgasbetrug	abgasbetrug	Wirtschaft
3	A06	abgasnorm	abgasnorm, abgasnormen	Umwelt
4	A06	abgasskandal	abgasskandal	Wirtschaft

```
In [8]: br.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6568 entries, 0 to 412
Data columns (total 4 columns):
Gruppe                  6568 non-null object
synonym (gestemmt)      6542 non-null object
synonym (original)      6567 non-null object
topic                   6505 non-null object
dtypes: object(4)
memory usage: 153.9+ KB
```

```
In [9]: # Ausfiltern fehlender Daten
        br['synonym (gestemmt)'].isnull().value_counts()
```

```
Out[9]: False    6542
        True       26
        Name: synonym (gestemmt), dtype: int64
```

```
In [10]: # Ausfiltern fehlender Daten
         br['topic'].isnull().value_counts()
```

```
Out[10]: False    6505
         True       63
         Name: topic, dtype: int64
```

```
In [11]: # die leeren Zeilen werden entfernt.
         Vokabular=br.dropna()
```

```
In [12]: Vokabular.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6480 entries, 0 to 412
Data columns (total 4 columns):
Gruppe                  6480 non-null object
synonym (gestemmt)      6480 non-null object
synonym (original)      6480 non-null object
topic                   6480 non-null object
dtypes: object(4)
memory usage: 151.9+ KB
```

```
In [13]: Vokabular.head()
```

Out[13]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	A06	abendschul	abendschule	Schule
1	A06	abgasaffar	abgasaffäre	Wirtschaft
2	A06	abgasbetrug	abgasbetrug	Wirtschaft
3	A06	abgasnorm	abgasnorm, abgasnormen	Umwelt
4	A06	abgasskandal	abgasskandal	Wirtschaft

```
In [14]: # bereinigete Datei in Excel speichern.
         out=pd.ExcelWriter('Phase2_1.xlsx')
         Vokabular.to_excel(out)
         out.save()
```

Anhang 2.5: Phase II – Umbenennung der Topics

```
In [1]: # Pandas und entsprechende Objekte importieren
        import pandas as pd
        from pandas import Series, DataFrame
        import numpy as np
        import os as os

In [2]: # die bereinigte Datei "Phase2_1" einlesen
        df=pd.DataFrame(pd.read_excel('Phase2_1.xlsx'))

In [3]: df.head()
```

Out[3]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	A06	abendschul	abendschule	Schule
1	A06	abgasaffar	abgasaffäre	Wirtschaft
2	A06	abgasbetrug	abgasbetrug	Wirtschaft
3	A06	abgasnorm	abgasnorm, abgasnormen	Umwelt
4	A06	abgasskandal	abgasskandal	Wirtschaft

```
In [4]: # Häufigkeiten von Topic
        df['topic'].value_counts().head(20)
```

Out[4]:

Sicherheit	356
Finanzen	334
Wirtschaft	318
Demokratie und Bürgerrechte	291
Soziales	291
Verkehr und Infrastruktur	263
Umwelt	260
Arbeit	258
Inneres und Justiz	252
Bildung und Forschung	175
Integration	162
Kultur	131
Gesundheit	129
Internationales	111
Städtebau und Stadtentwicklung	109

```
Wahlen                    109
Kinder und Jugend         106
Wahl                       86
Familie                    82
Schulen                    81
Name: topic, dtype: int64
```

```
In [5]: # Begriffe unter dem Topic "Sicherheit" anfragen und exportieren.
        TopicSicherheit=df[df['topic'].isin(['Sicherheit'])]
```

```
In [6]: TopicSicherheit.head()
```

Out[6]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
6	A06	abrust	abrüstung	Sicherheit
112	A06	brandschutz	brandschutz	Sicherheit
113	A06	brandschutzanlag	brandschutzanlage	Sicherheit
142	A06	datenschutz	datenschutz	Sicherheit
155	A06	einsatzort	einsatzort, einsatzorte	Sicherheit

```
In [7]: # Die Begriffe unter dem Topic "Sicherheit" werden in einer neuen Datei gespeichert.
        out=pd.ExcelWriter('TopicSicherheit.xlsx')
        TopicSicherheit.to_excel(out)
        out.save()
```

```
In [8]: # Zeilen mit dem Topic "Sicherheit" werden in der Datei "Phase2_1.xlsx" entfernt.
        OhneTopicSicherheit=df[(True-df['topic'].isin(['Sicherheit']))]
```

```
C:\Users\Elisa\Anaconda3\lib\site-packages\pandas\core\computation\expressions.py:183: UserWar
  unsupported[op_str]))
```

```
In [9]: OhneTopicSicherheit.head()
```

Out[9]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	A06	abendschul	abendschule	Schule
1	A06	abgasaffar	abgasaffäre	Wirtschaft
2	A06	abgasbetrug	abgasbetrug	Wirtschaft
3	A06	abgasnorm	abgasnorm, abgasnormen	Umwelt
4	A06	abgasskandal	abgasskandal	Wirtschaft

```
In [10]: # Die Datei wurden ohne dem Topic "Sicherheit" gespeichert.
         out=pd.ExcelWriter('OhneTopicSicherheit.xlsx')
         OhneTopicSicherheit.to_excel(out)
         out.save()
```

Anhang 2.6: Phase II – Bearbeitung auf Mehrdeutigkeit und Zusammenfassung

```
In [1]: # Pandas und entsprechende Objekte importieren
        import pandas as pd
        from pandas import Series, DataFrame
        import numpy as np
        import os as os

In [2]: # die Excel-Datei "OhneTopicSicherheit" einlesen.
        df1=pd.DataFrame(pd.read_excel('OhneTopicSicherheit.xlsx'))

In [3]: df1.head()

Out[3]:   Gruppe  synonym (gestemmt)        synonym (original)        topic
0     A06              abendschul                abendschule       Schule
1     A06              abgasaffar                abgasaffäre    Wirtschaft
2     A06            abgasbetrug                 abgasbetrug    Wirtschaft
3     A06              abgasnorm  abgasnorm, abgasnormen          Umwelt
4     A06           abgasskandal                abgasskandal    Wirtschaft

In [4]: # die Excel-Datei "manuell_umbenannt_Sicherheit" einlesen.
        df2=pd.DataFrame(pd.read_excel('manuell_umbenannt_Sicherheit.xlsx'))

In [5]: df2.head()

Out[5]:      Gruppe  synonym (gestemmt)                    synonym (original)  \
6       A01               abrust                              abrüstung
112     A01          datenschutz                             datenschutz
113     A01           einsatzort                  einsatzort, einsatzorte
142     A01             freiheit  freiheit, freiheiten, freiheitlichen
155     A01                fried              frieden, friedens, friede

                topic
6           Militär
112     Datenschutz
113         Militär
142      Sicherheit
155       Regierung

In [6]: df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 310 entries, 6 to 232
Data columns (total 4 columns):
Gruppe                  310 non-null object
synonym (gestemmt)      310 non-null object
synonym (original)      310 non-null object
topic                   310 non-null object
dtypes: object(4)
memory usage: 7.3+ KB
```

```
In [7]: # Zusammenführung der Datei "OhneTopicSicherheit" und "manuell_umbenennt_Sicherheit"
        Zusammenfuehrung=pd.concat([df1,df2])
        Zusammenfuehrung.head()
```

Out[7]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	A06	abendschul	abendschule	Schule
1	A06	abgasaffar	abgasaffäre	Wirtschaft
2	A06	abgasbetrug	abgasbetrug	Wirtschaft
3	A06	abgasnorm	abgasnorm, abgasnormen	Umwelt
4	A06	abgasskandal	abgasskandal	Wirtschaft

```
In [8]: Zusammenfuehrung.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6434 entries, 0 to 232
Data columns (total 4 columns):
Gruppe                  6434 non-null object
synonym (gestemmt)      6434 non-null object
synonym (original)      6434 non-null object
topic                   6434 non-null object
dtypes: object(4)
memory usage: 150.8+ KB
```

```
In [9]: # zusammengeführte Datei speichern.
        out=pd.ExcelWriter('zusammengefuehrte.xlsx')
        Zusammenfuehrung.to_excel(out)
        out.save()
```

```
In [10]: # oben zusammengeführt Datei wieder einlesen
         df=pd.DataFrame(pd.read_excel('zusammengefuehrte.xlsx'))
```

```
In [11]: df.head()
```

Out[11]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	A06	abendschul	abendschule	Schule
1	A06	abgasaffar	abgasaffäre	Wirtschaft
2	A06	abgasbetrug	abgasbetrug	Wirtschaft
3	A06	abgasnorm	abgasnorm, abgasnormen	Umwelt
4	A06	abgasskandal	abgasskandal	Wirtschaft

```
In [12]: # Duplikate entfernen
         Bereinigung=df.drop_duplicates(subset=["synonym (gestemmt)","topic"])

In [13]: Bereinigung.head()
```

Out[13]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	A06	abendschul	abendschule	Schule
1	A06	abgasaffar	abgasaffäre	Wirtschaft
2	A06	abgasbetrug	abgasbetrug	Wirtschaft
3	A06	abgasnorm	abgasnorm, abgasnormen	Umwelt
4	A06	abgasskandal	abgasskandal	Wirtschaft

```
In [14]: # Zusammenfassung der Datei "zusammengefuehrte.xlsx"
         Bereinigung.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 6432 entries, 0 to 232
Data columns (total 4 columns):
Gruppe                  6432 non-null object
synonym (gestemmt)      6432 non-null object
synonym (original)      6432 non-null object
topic                   6432 non-null object
dtypes: object(4)
memory usage: 150.8+ KB

In [15]: # Nach der Entfernung der Duplikate in einer neuen Datei speichern
         out=pd.ExcelWriter('Phase2_2.xlsx')
         Bereinigung.to_excel(out)
         out.save()

In [16]: # Datei wieder einlesen
         df=pd.DataFrame(pd.read_excel('Phase2_2.xlsx'))

In [17]: df.head()
```

Out[17]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
0	A06	abendschul	abendschule	Schule
1	A06	abgasaffar	abgasaffäre	Wirtschaft
2	A06	abgasbetrug	abgasbetrug	Wirtschaft
3	A06	abgasnorm	abgasnorm, abgasnormen	Umwelt
4	A06	abgasskandal	abgasskandal	Wirtschaft

```
In [18]: # Begriff "Flüchtling" betrachten
         # df.loc[df["synonym (gestemmt)"] == "fluchtling"]
         df[df['synonym (gestemmt)'].isin(['fluchtling'])]
```

Out[18]:

	Gruppe	synonym (gestemmt)	\
204	A06	fluchtling	

	Gruppe	synonym (gestemmt)
41	A02	fluchtling
51	A02	fluchtling
180	A03	fluchtling
45	A04	fluchtling
95	A10	fluchtling
1	B03	fluchtling
119	C04	fluchtling
138	C06	fluchtling

	synonym (original)
204	flüchtlinge, flüchtlingen, flüchtling
41	flüchtlinge, flüchtlingen
51	flüchtlinge
180	flüchtlinge, flüchtlingen, flüchtlings
45	flüchtlinge, flüchtlingen, flüchtling
95	flüchtlinge
1	flüchtlingen, flüchtling, flüchtings,
119	Flüchtlinge, Flüchtlingen, Flüchtlings, Flücht...
138	Flüchtlinge, Flüchtlingen, Flüchtling

	topic
204	Flucht
41	Flüchtling
51	Flüchtlingspolitik
180	Migration und Integration
45	Integration
95	Zuwanderung
1	Migration
119	Integration, Migration und Flüchtlingspolitik
138	Flüchtlinge

In [19]:
```
# Topic"Flücht" betrachten.
df[df['topic'].isin(['Flucht'])]
```

Out[19]:

	Gruppe	synonym (gestemmt)
192	A06	familiennachzug
203	A06	fluchthelf
204	A06	fluchtling
205	A06	fluchtlingsdrama
206	A06	fluchtlingsheim
207	A06	fluchtlingshilf
208	A06	fluchtlingslag
209	A06	fluchtlingspolit
210	A06	fluchtlingsstrom
211	A06	fluchtlingsunterkunft
539	A06	subsidiar
669	A06	willkommenskultur
710	A06	zugewandert

715	A06	zuwand
716	A06	zuwander
717	A06	zuwandererfamili
718	A06	zuwanderin
719	A06	zuwanderungsgesetz
720	A06	zuwanderungswell
651	A03	umweltfluchtling
672	A03	verfolg
675	A03	verhetz
682	A03	vertrieb

	synonym (original)	topic
192	familiennachzug	Flucht
203	fluchthelfer	Flucht
204	flüchtlinge, flüchtlingen, flüchtling	Flucht
205	flüchtlingsdrama	Flucht
206	flüchtlingsheim, flüchtlingsheime	Flucht
207	flüchtlingshilfe	Flucht
208	flüchtlingslager	Flucht
209	flüchtlingspolitik, flüchtlingspolitischen	Flucht
210	flüchtlingsstrom	Flucht
211	flüchtlingsunterkünfte, flüchtlingsunterkünften	Flucht
539	subsidiär, subsidiäre, subsidiärer, subsidiäre...	Flucht
669	willkommenskultur	Flucht
710	zugewanderte, zugewandert, zugewanderten	Flucht
715	zuwanderer, zuwandern	Flucht
716	zuwanderung	Flucht
717	zuwandererfamilie	Flucht
718	zuwanderin	Flucht
719	zuwanderungsgesetz	Flucht
720	zuwanderungswelle	Flucht
651	Umweltflüchtling	Flucht
672	verfolgen, Verfolgung	Flucht
675	Verhetzung	Flucht
682	Vertriebene	Flucht

In [20]:
```
# prüfen, ob ein Begriff unter unterschiedenen Topics zugeordnet ist.
Mehrdeutig=df['synonym (gestemmt)'].value_counts()
```

In [21]:
```
Mehrdeutig.head(20)
```

Out[21]:

fluchtling	9
rassismus	7
nationalsozialismus	7
rentenreform	7
grundgesetz	6
rentenversicher	6
opposition	6

```
freihandelsabkomm      6
bund                   6
energiew               6
rechtsstaat            6
rechtsextremismus      6
abschieb               6
koalitionsvertrag      6
reform                 5
ttip                   5
gerecht                5
anschlag               5
auslanderfeind         5
energi                 5
Name: synonym (gestemmt), dtype: int64
```

In [22]:
```
# Mehrdeutige Begriffe in einer neuen Datei speichern
out=pd.ExcelWriter('Mehrdeutig.xlsx')
Mehrdeutig.to_excel(out)
out.save()
```

In [23]:
```
df[df['synonym (gestemmt)'].isin(['energiew'])]
```

Out[23]:

	Gruppe	synonym (gestemmt)	synonym (original)	topic
162	A06	energiew	energiewende	Umwelt
145	B03	energiew	energiewende	Verkehr und Infrastruktur
157	B05	energiew	energiewende	Energie
244	B06	energiew	energiewende	Umweltschutz
227	B10	energiew	energiewende	Umelt
158	C04	energiew	Energiewende	Energie, Energieversorgung und Energiewende

In [ ]:
```
# Die Datei "Phase2_2.xlsx" wird manuell nach der Mehrdeutigkeit bearbeitet.
# Nach der Bearbeitung wird die Datei "Phase2_2.xlsx" zu
# "Die Final_Kontrollierte_Vokabulare-Datei" geändert.
```

In [24]:
```
# Die "Final_Kontrollierte_Vokabulare-Datei" wird eingelesen.
df=pd.DataFrame(pd.read_excel('Final_Kontrolliertes_Vokabular.xlsx'))
```

In [25]:
```
# Prüfen, ob Jede Begriff nur unter einem Topic zugeordent wurde.
df['synonym (gestemmt)'].value_counts().head(10)
```

Out[25]:
```
nazissindnazis    1
krisengebiet      1
```

```
steuervergeh                  1
frauenwahlrecht               1
personenverkehr               1
kartell                       1
strafregisterbeschein         1
kitastattbetreuungsgeld       1
atomar                        1
islamist                      1
Name: synonym (gestemmt), dtype: int64
```

In [26]:
```
# Zusammenfassung von endgültige Datei.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4910 entries, 1 to 232
Data columns (total 4 columns):
Gruppe                    4910 non-null object
synonym (gestemmt)        4910 non-null object
synonym (original)        4910 non-null object
topic                     4910 non-null object
dtypes: object(4)
memory usage: 115.1+ KB
```

Eidesstattliche Versicherung

Ich versichere, die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt zu haben. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quelle kenntlich gemacht.

Hamburg, 10.12.2017

_____

Xiaoyu Shi