Hochschule Hannover
Fakultät III – Medien, Information und Design
Abteilung Information und Kommunikation
Studiengang Informations- und Wissensmanagement

Textbasierte Annotation von Abbildungen mit Kategorien von Wikimedia

Masterarbeit

Frieda Josi
E-Mail: frieda.josi@web.de

Erstprüfer: Prof. Dr. Christian Wartena
Zweitprüferin: Dr. Ina Blümel

19.01.2018, Hannover

Kurzfassung

In der vorliegenden Masterarbeit geht es um die automatische Annotation von Bildern mithilfe der Kategoriesystematik der Wikipedia[1]. Die Annotation soll anhand der Bildbeschriftungen und ihren Textreferenzen erfolgen. Hierbei wird für vorhandene Bilder eine passende Kategorie vorgeschlagen. Es handelt sich bei den Bildern um Abbildungen aus naturwissenschaftlichen Artikeln, die in Open Access Journals veröffentlicht wurden. Ziel der Arbeit ist es, ein konzeptionelles Verfahren zu erarbeiten, dieses anhand einer ausgewählten Anzahl von Bildern durchzuführen und zu evaluieren. Die Abbildungen sollen für weitere Forschungsarbeiten und für die Projekte der Wikimedia Foundation[2] zur Verfügung stehen. Das Annotationsverfahren findet im Projekt NOA - Nachnutzung von Open Access Abbildungen[3] Verwendung.

Abstract

This master thesis deals with the automatic annotation of images using the Wikipedia category system.[4] The annotation is carried out using the image's captions and their respective text references. A suitable category is suggested for existing images. The images are illustrations from scientific articles published in open access journals. The aim of the work is to develop a conceptual procedure and to carry out and evaluate it on the basis of a selected number of images. The images shall be available for further research and for projects of the Wikimedia Foundation.[5] The annotation method is used in the NOA project - reuse of open access media.[6]

[1] Das Projekt Wikipedia ist eine mehrsprachige Online-Enzyklopädie, die frei und kollektiv erstellt wird. `https://en.wikipedia.org/wiki/Wikipedia`

[2] `https://en.wikipedia.org/wiki/Wikimedia_Foundation`

[3] Projekt der Hochschule Hannover und der Technischen Informationsbibliothek Hannover: Nachnutzung von Open Access Abbildungen

[4] The Wikipedia project is a multilingual online encyclopaedia created freely and collectively.`https://en.wikipedia.org/wiki/Wikipedia`.

[5] `https://en.wikipedia.org/wiki/Wikimedia_Foundation`

[6] Project of the Hannover University of Applied Sciences and the Hannover Technical Information Library: Reuse of open access media

Inhaltsverzeichnis

Abkürzungsverzeichnis

API	Application Programming Interface
CSV	Comma-separated values
DeGEval	Deutsche Gesellschaft für Evaluation
DFG	Deutsche Forschungsgemeinschaft
DOAJ	Directory of Open Access Journals
DOI	Digital Object Identifier
ESA	Europäische Weltraumorganisation
FOSTER	Facilitate Open Science Training for European Research
GND	Gemeinsame Normdatei
HMM	Hidden Markov-Modelle
LCCN	Library of Congress Control Number
NDL	National Diet Library
NLTK	Natural Language Toolkit
NOA	Nachnutzung von Open-Access-Abbildungen
OKFN	Open Knowledge Foundation
PoS	Part of Speech
Tf-idf	Term frequency und inverse document frequency
TXT	Textdatei
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VIAF	Virtual International Authority File

Abbildungsverzeichnis

Tabellenverzeichnis

Danksagung

Mein Dank, für die freundliche Unterstützung, gilt den Mitarbeitern des NOA Projektes. Das sind in erster Linie Prof. Dr. Christian Wartena, Dr. Ina Blümel, Lucia Sohmen, Jean Charbonnier und Lambert Heller.

Hinweise

1. Dieser Arbeit liegt eine CD, mit folgenden Dateien, bei:
   - Trainingsdaten: 397 Datensätze der Abbildungen aus NOA
   - Evaluationsdaten: 58 Datensätze der Abbildungen von Wikimedia Commons
   - Alle verwendeten Jupyter Notebooks
   - Digitale Version dieser Arbeit
2. Schreibweise:
   Berufs- Gruppen- und/oder Personenbezeichnungen werden in dieser Arbeit gendersensibel verwendet. Wo es möglich ist wird eine neutrale Schreibweise genutzt.

1. Einleitung

Ein wichtiger Grundgedanke in dieser Masterarbeit ist der freie Zugang zu wissenschaftlichen Informationen und Wissen. Dies ist eine grundlegende Forderung der Wissenschaft (Berliner-Erklärung 2003) sowie der Zivilgesellschaft (Alberts 2017). In der „Berliner Erklärung“[7], die 2003 erstmals von Forschungsorganisationen unterzeichnet worden ist, geht es um die Grundannahme, dass das Internet das Medium ist, welches zunehmend zur Wissensverbreitung eingesetzt werden wird. Die Aufgabe, die sich die Institutionen stellen ist, Wissen an die Gesellschaft und Wissenschaft so weiterzugeben, dass Informationen leicht zugänglich und umfassend sind (Berliner-Erklärung 2003).[8] Unterzeichner der Erklärung sind beispielsweise die Deutsche Forschungsgemeinschaft, die Max-Planck-Gesellschaft, die Fraunhofer-Gesellschaft, der Wissenschaftsrat und auch die Hochschule Hannover. Insgesamt 592 nationale und internationale Einrichtungen (Max-Planck-Gesellschaft 2017b).[9] Umfassende Open Access Veröffentlichungen sind, laut der Berliner Erklärung, die Forschungsergebnisse mitsamt der Ursprungs- und Metadaten inkl. aller Bildmaterialien (Berliner-Erklärung 2003, S. 1).
Diese freiwilligen Verpflichtungen werden bereits von vielen wissenschaftlichen Institutionen und einzelnen Wissenschaftler*innen umgesetzt. So unterstützt beispielsweise die Max-Planck-Gesellschaft ihre Forschenden darin, die Ergebnisse ihrer Arbeit mithilfe einer Open Access Strategie zu veröffentlichen. Wissenschaftler*innen, die nach der Strategie „Goldener Weg“[10] veröffentlichen möchten, werden durch die Übernahme der Publikationskosten für Open Access Zeitschriften unterstützt. Veröffentlichungen über den „Grünen Weg“ können über das Max-Planck-Repositorium (MPG.PuRe)[11] publiziert werden (Max-Planck-Gesellschaft 2017a). Um jungen Nachwuchswissenschaftlern*innen die Möglichkeiten und Methoden einer offenen Forschung und Wissenschaft näher zu bringen werden auch Programme, wie beispielsweise das „Fellow-Programm Freies Wissen“(Wikimedia-Deutschland 2017a), angeboten. Das Fellow-Programm ist ein Gemeinschaftsprojekt des Stifterverbands,[12] der Volkswagenstiftung[13] und der Wikimedia Deutschland.[14]

[7] Vollständiger Titel: Berliner Erklärung über den offenen Zugang zu wissenschaftlichem Wissen
[8] Die Berliner Erklärung definiert Informationen als Repräsentation des Wissens (Berliner-Erklärung 2003).
[9] Stand 03.08.2017
[10] Das Publizieren, mithilfe der Open Access Strategien nach dem goldener Weg oder grünen Weg, wird im Kapitel 3.3 beschrieben.
[11] Max-Planck-Repositorium: `http://pubman.mpdl.mpg.de/pubman/`, zuletzt geprüft am 03.08.17
[12] Seite des Stifterverbandes: `https://www.stifterverband.org/`, zuletzt geprüft am 03.08.17
[13] Seite der Volkswagenstiftung: `https://www.volkswagenstiftung.de/`, zuletzt geprüft am 03.08.17
[14] Seite der Wikimedia Deutschland: `wikimedia-deutschland_wikimedia_2017`, zuletzt geprüft am 03.08.17

Dieses Programm dient dazu, die Öffnung der Wissenschaft zu fördern, die Nachwuchswissenschaftler*innen zu vernetzen und ihnen Einblicke in freie Forschungsprojekte von Expert*innen zu ermöglichen.

Auch für die zivilgesellschaftliche Zielgruppe gibt es zahlreiche Initiativen und Webseiten, die den Open Access Gedanken aufgreifen und umsetzen, beispielsweise das Projekt „Wikimedia Deutschland – Gesellschaft zur Förderung Freien Wissens e. V.“der Wikimedia Foundation (Wikimedia-Deutschland 2017c). Das Ziel der Wikimedia Foundation ist es, freies Wissen zu erstellen, zu sammeln und zu verbreiten, um es so der gesamten Gesellschaft frei zur Verfügung zu stellen (Wikimedia-Deutschland 2017b).

1.1. Ausgangssituation und Motivation

Die grundlegende Motivation für diese Masterarbeit ist die Open Access Bewegung und die Forderungen von Open Science. Beispielsweise die Forderung, Teile eines Werkes frei Verwenden und Verbreiten zu können (OKFN 2018, Abschnitt 2.1.4). Bestandteile eines Werkes sind z.B. die verwendeten Grafiken und Bilder in einer wissenschaftlichen Publikation. Publizierte Abbildungen aus naturwissenschaftlichen Artikeln können dadurch einem größeren Publikum zugänglich gemacht und beispielsweise für Wikipedia Artikel oder weitere wissenschaftlichen Forschungen nachgenutzt werden. Auch die Weiterentwicklung vieler Informationsmanagementstrategien von zwischenstaatlichen Organisationen, wie beispielsweise der Europäische Weltraumorganisation (ESA), zeigen neue Möglichkeiten auf, das vorhandene Bildmaterial einer breiteren Öffentlichkeit zur Verfügung zu stellen. Die Strategie der ESA beinhaltet beispielsweise eine umfassende und regelmäßige Nachnutzung des Bildmaterials (ESA 2017). Ab 2017 veröffentlichte die ESA weiteres Bild- und Videomaterial unter der Open Access Lizenz Creative Commons „Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 IGO“(CC BY-SA 3.0 IGO). Diese Lizenz ermöglicht u.a. zwischenstaatlichen Organisationen eine umfassende Bereitstellung von Bildmaterial auf Wikimedia Commons (Commons 2017a).

Diese Masterarbeit ist in das Projekt Nachnutzung von Open-Access-Abbildungen (NOA) eingebunden, welches von der Deutsche Forschungsgemeinschaft (DFG) gefördert wird. Das Ziel des NOA-Projektes ist es, die Nachnutzung von Abbildungen aus ausgewählten Open Access Zeitschriften zu ermöglichen. Damit reiht sich das Projekt NOA in die aktuelle Entwicklung von Open Science ein.

1.2. Forschungsstand und forschungsleitende Frage

Diese Arbeit, integriert in das Forschungsprojekt NOA, setzt eine der Forderungen des Open Science um, Forschungsdaten langfristig frei nachnutzbar und auffindbar zu halten. Die Deutsche Forschungsgemeinschaft zählt folgende Daten zu den Forschungsdaten:

> „Zu Forschungsdaten zählen u.a. Messdaten, Laborwerte, audiovisuelle Informationen, Texte, Surveydaten, Objekte aus Sammlungen oder Proben, die in der wissenschaftlichen Arbeit entstehen, entwickelt oder ausgewertet werden." (DFG 2010)

Für diese Forderung setzt sich u.a. die Deutsche Forschungsgemeinschaft engagiert ein. Sie fordert, dass Forschungsergebnisse und ihre zugrundeliegenden Daten verifizierbar sein sollen und auch produktiv nachgenutzt werden können (Neuroth 2012, S. 10). Die Helmholtz-Gemeinschaft, zu der die Arbeitsgruppe Open-Science gehört, geht davon aus, dass Forschungsdaten durch die Open-Science-Bewegung neue Potentiale für die Wissenschaft schaffen (Helmholtz-Gemeinschaft 2016, S. 1). Die Anforderungen die dabei an den Umgang mit den Forschungsdaten gestellt werden, werden in den FAIR Daten Prinzipien von der Forschungsgemeinschaft FORCE11 beschrieben (Hagstrom 2014).[15] Demnach sollen Forschungsdaten auffindbar und zugänglich, interoperabel und wiederverwendbar sein. Für die Auffindbarkeit der Forschungsdaten werden Metadaten benötigt, nach denen recherchiert werden kann. Dazu gehören auch vergebene fachliche Kategorien für die Klassifizierung in eine Systematik.

Da das NOA Projekt die Nachnutzbarkeit der vorhandenen Open Access Abbildungen durch die Integration in das Medienarchiv Wikimedia Commons gewährleisten will, wird in dieser Arbeit eine Methode entwickelt, die passende Wikipedia Kategorien für einzelne Open Access Abbildungen vorschlägt, die die Abbildungen in die Kategoriesystematik von Wikimedia möglichst passend[16] einordnen. Für die textbasierte Annotation der Abbildungen werden die Bildbeschriftungen und die Referenzstellen der Abbildungen aus dem Artikel genutzt.

Die forschungsleitende Frage dieser Arbeit lautet daher:
Kann das Kategoriensystem der Wikimedia Foundation automatisiert für die Kategorisierung der Bilder aus Open Access Journals eingesetzt werden?

[15] FAIR = **F**indable, **A**ccessible, **I**nteroperable, and **R**e-usable

[16] Die Qualität der vorgeschlagenen Kategorien wird anhand von einigen Bildern überprüft, die schon bei Wikimedia Commons kategorisiert vorhanden waren.

1.3. Theoretische Basis

Die theoretische Basis der Arbeit bilden, im Bereich Text Mining, erstellte und publizierte Lösungen der Professur Sprach- und Wissensverarbeitung der Hochschule Hannover und die Literaturreferenzen aus dem NOA Projektantrag (Blümel u. a. 2014, S. 2). Das Extrahieren der Wörter und Phrasen aus den Bildbeschriftungen und den Textreferenzstellen beruht auf Ausarbeitungen von Leong u.a. (Leong u. a. 2010) und Mihalcea und Tarau (Mihalcea u. Tarau 2004). Für das Ermitteln der Eignung von den extrahierten Schlüsselbegriffen aus den Bildbeschriftungen werden die Publikation von Frank u.a. (Frank u. a. 1999) und Turney (Turney 2000) verwendet. Für das Heranziehen der Wikipedia Kategorien dienen nachfolgende Arbeiten als Grund- und Ausgangslage. Das Mapping der extrahierten Schlüsselbegriffe auf die Wikipedia Titel wird in Mihalcea und Csomai (Mihalcea u. Csomai 2007) beschrieben. Bei Mihalcea und Csomai werden aber, anders als in dieser Arbeit, die Schlüsselbegriffe mit Links zu den Wikipedia Artikel versehen. Wie die Wikipedia Titel als Vokabular zur Verschlagwortung von Bildern genutzt werden kann, wird aufgezeigt von Medelyan u.a. (Medelyan u. a. 2008). In dieser Arbeit werden jedoch nicht die Titel der Wikipedia-Artikel, sondern deren zugewiesene Kategorien als Annotation verwendet, wie in Wartena und Brussee (Wartena u. Brussee 2008) aufgezeigt ist.

Die zweite theoretische Basis ist die Bereitstellung von explizitem Wissen. Forschungsergebnisse können, aus der Sichtweise des Wissensmanagements, als explizites Wissen eingeordnet werden. Sie liegen in einer methodischen und systematischen Form, außerhalb der Köpfe der Wissenschaftler*innen, in Mediendateien gespeichert vor. Die Forschungsergebnisse werden so, in ihrer expliziten Form, erst für andere Personen verfügbar und nachnutzbar (North 2016, S. 46). Das NOA Projekt schließt an dieser Stelle an und bietet ein Informationsangebot, bei dem explizites Wissen für ein großes Publikum verwendbar und nachnutzbar wird. Somit ist auch die vorliegende Arbeit eine Teillösung zur Bereitstellung von explizitem Wissen. Des Weiteren kann die Bereitstellung und Verteilung von Forschungsdaten als Grundvoraussetzung für Wissensaufbau gesehen werden. Explizites Wissen benötigt die sichere Speicherung und Verteilung und erfolgt in digitalisierter Form, beispielsweise in Datenbanken (North 2016, S. 287).

Für die Zusammenarbeit und kooperative Wissensnutzung bzw. -nachnutzung wird die Infrastruktur der Wikimedia Foundation[17] verwendet. Sie dient als nachhaltige Infrastruktur für das Retrieval freier Open Access Abbildungen. Aus dieser Blickrichtung kann das Informationsangebot NOA in den Bausteinen *Wissensbewahrung*, *Wissensnutzung* und *Wissensverteilung* aus den Kernprozessen des Wissensmanagements verortet werden (Raub u. Romhardt 2010, S. 28 ff).

[17] Webseite: `https://wikimediafoundation.org/`

1.4. Verwendete Methoden und Gliederung

In dieser Ausarbeitung werden neben den klassischen wissenschaftlichen Methoden, wie Literaturstudien, Analysemethoden aus der Computerlinguistik verwendet. Neben der Datenvorverarbeitung sind das Methoden aus der Informationsextraktion. Verwendet werden in dieser Arbeit die Segmentierung der Sätze, die Segmentierung auf der Wortebene, die Lemmatisierung der Wörter, das Part-of-Speech Tagging und die Syntaxanalyse (*en: Parsing*) von Phrasen. Aus dem Information Retrieval werden die Berechnungen der Termfrequenz und der inversen Dokumenthäufigkeit eingesetzt. Die dadurch erstellten Annotationsmethoden werden mit Parameteränderungen auf Testdaten angewendet und verglichen. Die besten Ergebnisse einer Annotationsvariante werden anhand von Testdatensätzen evaluiert, visualisiert und mit Empfehlungen zur Annotation an das NOA Projekt übergeben. Zur Bewertung der Evaluierung wird die Berechnungen der Genauigkeit (en: Precision) eingesetzt. Diese Methode findet im Information Retrieval und in der Klassifikation Verwendung.

Gliederung der Arbeit

Nach der Einleitung, u.a. mit der forschungsleitenden Frage im **Kapitel 1**, folgt im **Kapitel 2** die Einordnung der Inhalte dieser Arbeit in das Forschungsprojekt NOA. In dem **Kapitel 3** und den dazugehörigen Unterkapiteln, werden die Grundlagen des Open Science aufgezeigt. Die dort aufgezeigten Open Science-Verfahren finden, teilweise, in der Umsetzung dieser Arbeit Verwendung. Insbesondere der freie Zugang zu Forschungsdaten und deren Nachnutzbarkeit ist eine wichtige Ausgangslage in dieser Arbeit.

In dem entwickelten Annotationsverfahren werden die Kategorien der Wikimedia Foundation, im speziellen dem Projekt Wikipedia, genutzt. Im **Kapitel 4** wird deshalb auf die Kategoriensyntax von Wikipedia, Wikimedia Commons und Wikidata eingegangen.

Die Datensätze der Abbildungen, die das Rohmaterial dieser Arbeit bilden, werden im **Kapitel 5** vorgestellt. Für die Erstellung der Datensätze werden die Bildunterschriften der Abbildungen und die Referenzstellen im Artikel zusammengetragen. In den Kapitel **5.2.1** bis **5.2.4** wird die Extraktion der Terme beschrieben. Der Ablauf der Extraktion wird in diesem Kapitel detailliert beschrieben, in den darauf folgenden Kapiteln wird auf diese Vorgehensweisen verwiesen.

Danach folgt die Extraktion der Nominalphrasen, die Vorgehensweise dazu wird im **Kapitel 5.3** aufgezeigt. Im **Kapitel 5.4** wird die Extraktion von Termen und Nominalphrasen beschrieben, der Vergleich der drei Vorgehensweisen für die Extraktion wird im **Kapitel 5.5** gezeigt.

Die Zuordnung der Terme zu einem Wikipedia Artikel wird im **Kapitel 6** beschrieben. Hier wird auch die Schnittstelle der Wikipedia und die genaue Vorgehensweise beim Mapping dargestellt. Dazu gehört auch die Beschreibung zum Ranking der Kategorien. Im **Kapitel 6.5** wird das Mapping detailliert vorgestellt. Da der Ablauf für die Nominalphrasen und für die Kombination aus Termen und Nominalphrasen ähnlich ist, wird in den **Kapitel 7** und **Kapitel 8** (Mapping der Nominalphrasen und der Kombination zu den Wikipedia Kategorien) auf die detaillierte Beschreibung im **Kapitel 6** (Mapping der Terme und Nominalphrasen zu den Wikipedia Kategorien) verwiesen.

Die entwickelte Annotationsmethode wird anhand von Abbildungen, aus Wikimedia Commons, evaluiert. Die Evaluierung wird im **Kapitel 9** beschrieben. Verglichen werden die vorhandenen Wikimedia Commons Kategorien, für 58 Abbildungen, mit den Kategorie-Vorschlägen aus der Annotationsmethode. Eine zusätzliche manuelle Evaluation wird im **Kapitel 9.4.2** aufgezeigt.

Im **Kapitel 10** werden die Ergebnisse der Evaluation als Empfehlungen für den Einsatz im NOA Projekt erklärt. Des Weiteren werden einige Herausforderungen vorgestellt.

Die abschließende Zusammenfassung dieser Arbeit erfolgt im **Kapitel 11**.

Alle Python Programme befinden sich auszugsweise im Anhang und vollständig auf einer CD, die dieser Masterarbeit beiliegt. Zusätzlich stehen die Programme auf GitHub unter: `https://github.com/f-josi/MA` zur Verfügung.

2. Einordnung in das Forschungsprojekt NOA

In dieser Ausarbeitung werden die Bildbeschriftungen und Textreferenzen verwendet, die sich in der Bilddatenbank des NOA Projektes befinden. Das NOA Projekt entwickelt ein Verfahren, um multimediale Open Access Objekte automatisiert zu sammeln, zu erschließen und mithilfe der Wikimedia Commons bereitzustellen. Aktuell befinden sich über 5 Mio. Abbildungen in einer separaten Bilddatenbank.[18] In der Abb. 2.1 sind beispielsweise die ersten Treffer der Bilder zum Suchwort *photovoltaic* zu sehen. NOA ist ein Kooperationsprojekt der Hochschule Hannover und der Technischen Informationsbibliothek Hannover, gefördert durch die Deutsche Forschungsgemeinschaft. Das NOA Projekt wurde gestartet, um die Nachnutzbarkeit von Forschungsrohdaten, beispielsweise Abbildungen, zu erleichtern. Veröffentlichte Forschungsergebnisse erscheinen hauptsächlich als Textpublikationen angereichert mit Rohdaten. Diese Rohdaten, im NOA Projekt überwiegend Grafiken, Abbildungen und Fotos, sollen nun für weitere Forschungsarbeiten zur Verfügung gestellt werden. Es werden dafür Abbildungen aus Open Access Journals[19] verwendet, die u.a. im Hindawi Verlag veröffentlicht wurden (Blümel u. a. 2014). Eine Übersicht über die Verlage, von denen Open Access Zeitschriften genutzt werden, befindet sich im Anhang unter A.

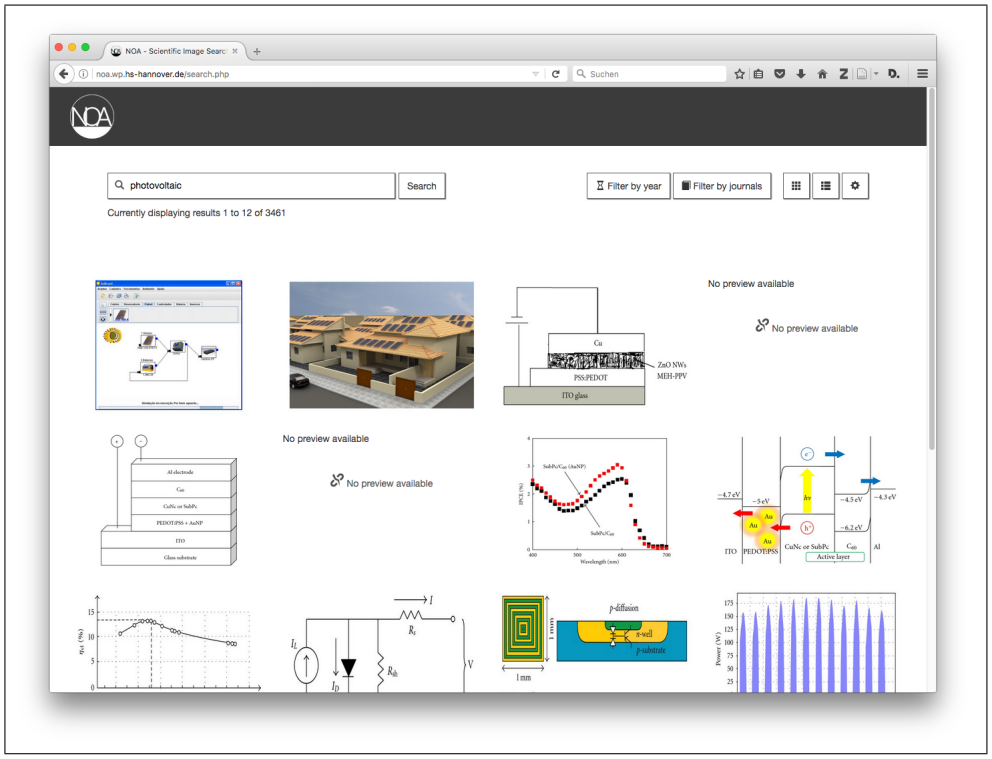


Abbildung 2.1: Oberfläche der NOA Bildersuche

[18] Webseite der NOA-Bildersuche: `http://noa.wp.hs-hannover.de/`

[19] Open Access Zeitschriften sind wissenschaftliche Fachzeitschriften, die kostenfrei zur Verfügung stehen. `https://de.wikipedia.org/wiki/Open-Access-Zeitschrift`

Eine weitere Aufgabenstellung des Projektes ist die Bereitstellung der Abbildungen für die Öffentlichkeit. Dies soll mithilfe der Wikimedia-Infrastruktur durchgeführt werden. Den Autor*innen der Wikipedia sollen zukünftig automatisch wissenschaftliche Bilder vorgeschlagen werden, die sie frei für Wikipedia-Artikel und weitere Zwecke verwenden können. Die automatische Zuordnung der Bilder soll über die vorhandenen Kategorien der Wikipedia erfolgen. Die wissenschaftlichen Forschungsrohdaten werden somit leichter recherchierbar und können weltweit von Wikipedia-Nutzer*innen verwendet werden (Niemeyer 2016). In diesem Aufgabenbereich des NOA Projektes (Arbeitspaket 3.2 Metadatengewinnung) (Blümel u. a. 2014) sind die Inhalte dieser Arbeit angesiedelt. Das Spektrum umfasst dabei den automatischen Vorschlag mehrerer Wikipedia Kategorien zu einer Abbildung aus den vorhandenen Bildern der NOA Bilddatenbank.

Im ersten Schritt sollen dazu aus den Bildbeschriftungen und Textreferenzen Terme extrahiert werden. Im nächsten Schritt werden Nominalphrasen extrahiert. Von den Termen und auch von den Nominalphrasen wird dann der Term frequency und inverse document frequency (Tf-idf)-Wert berechnet. Durch das Berechnen der Relevanz der Terme und Phrasen kann eine Auswahl getroffen werden für einen späteren Vergleich mit den vorhandenen Kategorien von Wikipedia Artikeln. Zusätzlich gibt es einen weiteren Vergleich, bei dem die Terme mit den Nominalphrasen kombiniert werden, bevor die Wikipediakategorien ermittelt werden. Diese drei Vorgehensweisen werden jeweils weiter aufgeteilt. Die Ergebnisse dieser drei Möglichkeiten (Verwendung der Terme, Phrasen, Kombination aus beiden) werden manuell gesichtet. Die Methode mit den besten Kategorievorschlägen wird im Anschluss evaluiert. Dabei werden die automatisch vorgeschlagene Kategorie einer Abbildung, mit der vorhandenen Kategorie der gleichen Abbildung bei Wikimedia Commons verglichen. Hierfür stehen 58 Open Access Abbildungen zur Verfügung, die bei Wikimedia Commons vorhanden waren und kategorisiert sind.

Angaben zum NOA Projekt:
Informationen über das Projekt können im Blog unter `http://blogs.tib.eu/wp/noa/` eingesehen werden. Das NOA-Team besteht aus Prof. Dr. Christian Wartena, Dr. Ina Blümel, Lambert Heller, Jean Charbonnier, Lucia Sohmen, John Rothman, Birte Rohden und Frieda Josi.

3. Grundlagen des Open Science

Ein veröffentlichtes Werk, das nach den Regeln von Open Science publiziert wird, beinhaltet, dass es „offen“ publiziert wird. „Offene Wissenschaft“ wird in dieser Arbeit als „offenes Wissen“ in Kombination mit den Strategien und Verfahren von Open Science, siehe Kapitel 3.2, gesehen. Die Definition von „Offenes Wissen“ wird hierfür von der Open Knowledge Foundation (OKFN)[20] übernommen. Eine offene Veröffentlichung ist, nach der OKFN, unter folgenden Voraussetzungen gegeben (Hauschke u. Herb 2017):

1. Der **Zugang** zu dem vollständigen Werk ist verfügbar und sollte zum gebührenfreien Download abrufbar sein. Des Weiteren dürfen die Kosten für den Zugang die Reproduktionskosten nicht übersteigen.

2. Die **Weiterverbreitung** muss so lizenziert sein, dass das Werk eigenständig, oder auch als Beitrag einer Sammlung verkauft oder verschenkt werden darf. Es dürfen keine Lizenzzahlungen oder andere Gebühren anfallen.

3. Die **Nachnutzung eines Werkes** ist möglich. Ebenso ist die Modifikation oder die Ableitung eines Werkes inkl. der Weiterverbreitung, unter den Lizenzbedingungen des zugrundeliegenden Werks, möglich.

4. Es dürfen **keine technischen Einschränkungen** verwendet werden, um die Nutzungen der Punkte 1-3 einzuschränken. Dies kann vorzugsweise durch die Nutzung eines offenen und frei verfügbaren Datenformats erfolgen.

5. Als Bedingung für die Weiterverbreitung oder Nachnutzung des Werkes kann eine **Namensnennung** verlangt werden. Wenn eine Namensnennung erforderlich ist, sollte dem Werk eine Liste mit den zu nennenden Personen hinzugefügt werden.

6. Um die **Integrität**[21] eines Werkes zu gewährleisten, kann als Bedingung für die Weiterverbreitung oder Nachnutzung des Werkes verlangt werden, dass für das abgeleitete Werk (Derivat) ein anderer Name bzw. eine andere Versionsnummer vergeben wird.

7. Die Lizenz, zur Nutzung und Verbreitung eines Werkes, darf **keine Person oder Personengruppe diskriminieren**.

8. Bei der Nachnutzung eines Werkes dürfen die **Einsatzzwecke nicht eingeschränkt** werden.

[20] Seite der Open Knowledge Foundatio: `http://opendefinition.org/`, zuletzt geprüft am 07.08.17

[21] Die Integrität eines Werkes ist, gemeinsam mit der Verfügbarkeit und der Vertraulichkeit eines Werkes, ein wichtiges Ziel der Informationssicherheit (BSI 2017).

9. Alle Empfangenden erhalten die gleichen rechtlichen Bedingungen denen auch das zugrundeliegende Werk unterliegt. Für die **Lizenzvergabe** müssen keine zusätzlichen Bedingungen akzeptiert werden.

10. Ein Werk, welches einer Sammlung entnommen wird, unterliegt den **Lizenzbestimmungen der ursprünglichen Sammlung**, erhält deren Rechte und kann einzeln weiterverwendet werden.

11. Die Verbreitung von Werken, die gemeinsam mit dem lizenzierten Werk veröffentlicht werden, dürfen **durch das lizenzierte Werk nicht eingeschränkt werden**. Es können, bei der Veröffentlichung mehrerer Werke auf einem Medium, unterschiedliche Lizenzen vorhanden sein.

Wenn diese Voraussetzungen erfüllt sind, kann von offenem Wissen gesprochen werden. Eine Argumentation für offenes Wissen ist auch, dass die Qualität der Forschung dadurch verbessert werden kann (Helmholtz-Gemeinschaft 2017). Offenes Wissen ist in dieser Arbeit die Basisdefinition für den Begriff Open Science.

3.1. Prinzipien des Open Science

Laut der offenen Initiative AG Open Science (Open Science Arbeitsgruppe)[22] sind es vier Grundprinzipien, auf denen Open Science basiert. Es sind *Transparenz*, *Reproduzierbarkeit*, *Wiederverwendbarkeit* und die *Offene Kommunikation* (AG 2017). Eine Übersicht zu den Grundsätzen folgt in der Abbildung 3.2.[23] Das Prinzip der Transparenz kann zur Qualitätskontrolle von Publikationen dienen, da der Zugriff auf die Veröffentlichung und auch der Zugriff auf die Forschungsdaten offen ist (Herb 2012b). Die Reproduzierbarkeit stellt sicher, dass die Forschungsergebnisse durch andere Wissenschaftler wiederholt werden können. Dies wird auch in der Denkschrift *Sicherung guter wissenschaftlicher Praxis* von der DFG empfohlen (DFG 2013, S. 17). Ein weiteres Prinzip ist die Reproduzierbarkeit, bei der mit gleichen Methoden, aber unter anderen Bedingungen, gleiche Ergebnisse erzeugt werden können (Herb 2012a, Stichwort: Repeatability).

Des Weiteren kann auch durch die Verbesserung von Kommunikationsstandards und Berichterstattung die Reproduzierbarkeit dazu dienen verifizierbare Ergebnisse zu erhalten (Herb 2012a, Stichwort: Empirical reproducibility). Darüber hinaus umfasst Open Science auch Strategien und Verfahren, diese werden im nächsten Kapitel beschrieben.

[22] Die AG Open Science wird unterstützt von open-access.net, Wikimedia Deutschland, Open Knowledge Foundation Deutschland, Bürger schaffen wissen, Alexander von Humboldt Institut für Internet und Gesellschaft und sience 2.0 Leibnitz-Forschungsverbund.

[23] Erstellt nach Open Science AG (AG 2017)

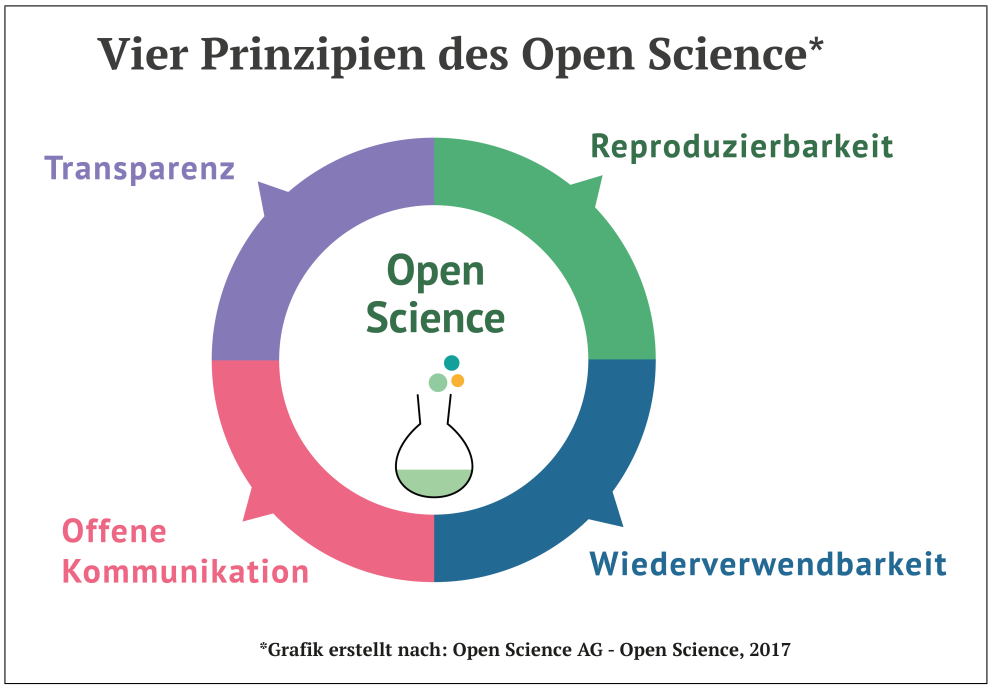


Abbildung 3.2: Prinzipien des Open Science

3.2. Strategien und Verfahren von Open Science

Im Open Science gibt es eine nicht abschließende Anzahl von Strategien und Verfahren die eingesetzt werden. In einigen Veröffentlichungen werden diese auch als Prinzipien des Open Science beschrieben (Neuhold 2016). In dieser Arbeit werden unter den Prinzipien die Grundannahmen einer Idee verstanden, die daraus erstellten Lösungen und Vorgehensweisen werden als Strategien und Verfahren bezeichnet. Der Umfang der Strategien wird aus dem Living Book[24] des Wikimedia Fellowprogramms Freies Wissen (Neuschaefer u. a. 2017) übernommen. Die Verfahren und Strategien im Open Science, die hier beschrieben werden, sind: **Open Access**, **Open Data**, **Citizen Science**, **Open Educational Resources**, **Open Methodology**, **Open Notebook Science**, **Open Source** und **Open Peer Review**. Einen ähnlichen Umfang definiert auch die e-Learning Plattform Facilitate Open Science Training for European Research (FOSTER) (Bueno de la Fuente 2017).

Auf Basis dieser Abgrenzung folgt die Aufstellung des Umfangs der Strategien in Abbildung 3.3. In dieser Abbildung ist auch der Schnittbereich der entwickelten Methode aus dieser Arbeit, der Schnittbereich mit dem gesamten NOA-Projekt und den Anforderungen von Open Science aufgezeigt.[25] Die Definitionen dieser Strategien und Verfahren werden hier in aller Kürze vorgestellt, ebenso wird die Übereinstimmung mit dieser Arbeit und dem NOA Projekt aufgezeigt. Alle erwähnten Definitionen

[24] Zuletzt abgerufen am 07.08.17
[25] Die Angaben zum NOA Projekt erfolgen in dieser Aufstellung der Strategien nur ergänzend zum Schnittbereich der Masterarbeit.

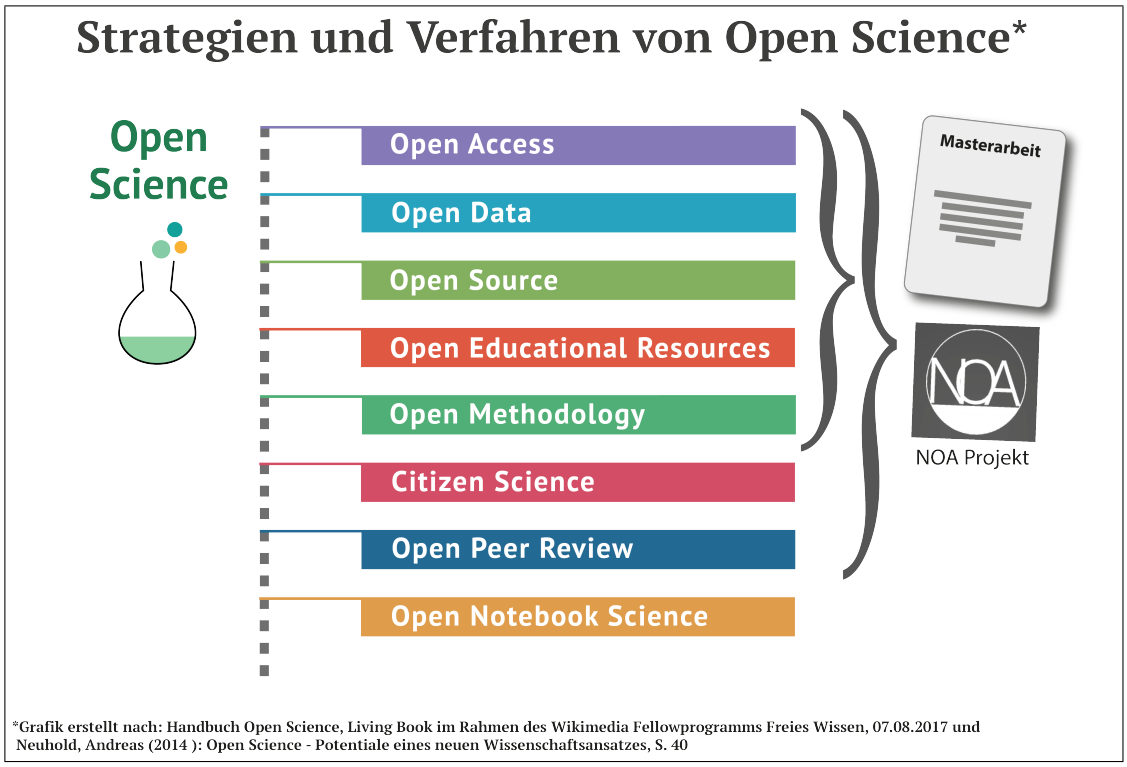


Abbildung 3.3: Schnittmenge der Strategien von Open Science, der Masterarbeit und dem NOA Projekt

sind dem Living Book Open Science (Neuschaefer u. a. 2017) entnommen.

- **Open Access:** Der Zugang zu wissenschaftlichen Veröffentlichungen und den zugrundeliegenden Forschungsdaten soll kostenfrei und möglichst barrierefrei möglich sein.[26] Die Publikationen zu dem Forschungsprojekt NOA und auch das Ergebnis dieser Arbeit werden in die Projekte der Wikimedia Foundation integriert und bieten so einen kostenfreien Zugang. Die Publikationen dazu werden unter Open Access Bedingungen veröffentlicht.[27]

- **Open Data:** Die gesamten Daten eines Forschungsprojektes sollen offen zur Verfügung gestellt werden, nicht nur kleinere Auszüge davon in der Endpublikation, siehe dazu Kapitel 1.2 zu FAIR Daten Prinzipien von Forschungsdaten. Forschnungsdaten können beispielsweise als eigenständige Datenpublikation erfolgen. Die, in dieser Arbeit, verwendeten Informationen über die Abbildungen und auch die Abbildungen selbst, werden in der wissenschaftlichen Bildersuche unter `http://noa.wp.hs-hannover.de` kostenfrei bereitgestellt. Zusätzlich sollen die Abbildungen zu Wikimedia Commons übertragen werden und stehen somit für jeden frei nachnutzbar zur Verfügung.

[26] Siehe dazu unter Kapitel 3 die OKFN Definition zu Offenes Wissen -> Zugang zum Wissen
[27] Da das Projekt NOA von der DFG finanziert wird, sollten die Forschungsergebnisse auch Open Access publiziert werden (DFG 2016, S. 16 Veröffentlichung von Forschungsergebnissen).

- **Open Source:** Damit kann der Quellcode einer Software gemeint sein. Im Data- und Text Mining sind es die verwendeten Rohdaten. Der Quellcode der entwickelten Methoden wird ebenso wie die Beschreibung der Methoden bei GitHub `https://github.com/f-josi/MA` zur Verfügung stehen.

- **Open Educational Resources:** Das wissenschaftliche Material, das zur Ausbildung und Lehre genutzt werden kann, soll frei und öffentlich zugänglich sein. Es kann als offenes Lehrmaterial beispielsweise als Buch, als Vorlesungsmaterial oder auch als Datensammlung verfügbar sein. Die Open Access Abbildungen, die dieser Arbeit zugrunde liegen, können aus dem Medienarchiv bei Wikimedia Commons für die Einbindung in eigene Lehrmaterialien verwendet werden.

- **Open Methodology:** Eine detaillierte Beschreibung der eingesetzten Methoden ist notwendig, um die Reproduzierbarkeit eines Forschungsergebnisses zu ermöglichen. Diese Beschreibung sollte auch die Durchführung der Datenanalyse genau aufzeigen. Die Verwendung der entwickelten Methoden dieser Masterarbeit kann nach Abgabe auf SerWisS[28] und in einem Repositorium auf GitHub[29] eingesehen werden.

- **Citizen Science:** Unter Citizen Science wird die Mitarbeit von wissenschaftlichen Laien verstanden, die in einem Forschungsprojekt mitarbeiten. Dies kann beispielsweise bei der Datenerhebung erfolgen. Nach Abschluss dieser Arbeit und nach Beendigung des NOA Projektes stehen den Nutzer*innen der Wikipedia die Bilddaten und auch die Metadaten zur Weiterverwendung zur Verfügung. Sie können die vorhandenen Kategorien manuell nachbessern bzw. detailliertere Kategorien vergeben.

- **Open Peer Review:** Mit Peer Review wird die Qualitätskontrolle von Manuskripten bezeichnet, die bei einem Journal vor Veröffentlichung geprüft werden. Beim Open Peer Review soll der Prozess der Überprüfung offen geführt werden, d.h. der Autor*in sind die Reviewer bekannt.[30] Die Review-Berichte werden gemeinsam mit der Publikation veröffentlicht. Die Methoden und Ergebnisse aus dem NOA Projekt werden in Konferenz-Artikeln mit dem Peer Review Verfahren veröffentlicht. Zusätzlich wurde im Vorfeld der DFG-Antrag für das NOA-Projekt, für eine offene Diskussion des Forschungsvorhabens, veröffentlicht.[31]

[28] SerWisS: `https://serwiss.bib.hs-hannover.de/home`, zuletzt geprüft am 14.08.17
[29] GitHub-Link: `https://github.com/f-josi/MA`
[30] Dies ist beim klassischen Peer Review nicht der Fall, was zu einer unfairen Beurteilung führen kann.
[31] Der DFG-Antrag wurde auf der Webseite für wissenschaftsbezogene Berichte: `https://zenodo.org/record/12745` veröffentlicht.

- **Open Notebook Science:** In einigen Wissenschaftsdisziplinen muss ein Laborbuch geführt werden, das u.a. die Ergebnisse dokumentiert. Dieses Laborbuch kann auch öffentlich geführt werden. Ein Laborbuch wird weder in dieser Masterarbeit noch im Projekt NOA eingesetzt.

3.3. Open Access Publikationsstrategien im Open Science

Bei den Open Access Strategien gibt es zwei Möglichkeiten zu publizieren. *Goldener Open Access* beschreibt dabei die Erstveröffentlichung der wissenschaftlichen Artikel in einem OA-Journal oder als eine OA-Monografie. Für die Publikation besteht ein Rahmenvertrag mit dem Verlag. In diesem Vertrag werden die Nutzungsrechte und Nutzungsbedingungen festgelegt. Die Finanzierung kann sich aus unterschiedlichen Quellen zusammensetzen. Beispielsweise aus Werbung, Verkauf von Print-Ausgaben, Finanzierung von Institutionen oder auch durch Publikationsgebühren (Open-Access 2017).

Der *Grüne Open Access* beschreibt die Selbstarchivierung, die zusätzlich zur Veröffentlichung von einem Verlag erfolgt. Die Selbstarchivierung kann auf institutionellen oder disziplinären Repositorien[32] erfolgen, oder auf der eigenen Webseite der Wissenschaftler*in (Open-Access 2017). Institutionelle Repositorien gibt es in vielen Universitäten und Hochschulen, die Vorgaben für die Veröffentlichungen sind dabei individuell geregelt. An der Hochschule Hannover ist es beispielsweise der *SerWisS - Server für Wissenschaftliche Schriften der Hochschule Hannover*.[33] Fachrepositorien können als Ergänzung zu den institutionellen Repositorien gesehen werden, da sie nur in einigen Fachdisziplinen vorhanden sind und sich etabliert haben, beispielsweise die Plattform arXiv[34] für Physik, Mathematik und Informatik (Weingart 2016, S. 8). Forschungsergebnisse können über Repositorien wie arXiv zeitnah veröffentlicht und diskutiert werden. Auch für andere Fächer werden Repositorien aufgebaut, wie bioRxiv[35] (Biologie) und die SocArXiv[36] (Sozialwissenschaft) (Schmitz 2017, S. 305f).

Die Qualität von Publikationen in Open Access Journals wird, ähnlich wie beim klassischen Closed Access, durch ein Peer Review Prozess, Blind Peer Review oder ein Double Blind Peer Review Verfahren gesichert (Hacker 2017, S. 287f). Im Open Access wird unter Peer Review ein Gutachten verstanden, bei dem die Gutachter*in der Autor*in bekannt ist (siehe auch Kapitel 3.2 Stichwort Open Peer Review). Beim Blind Peer Review ist die Gutachter*in der Autor*in nicht bekannt und beim Double Blind Peer Review kennen beide - Autor*in und Gutachter*in - die Identität des

[32] Ein Repositorium ist ein Server zur Datenspeicherung, der beispielsweise wissenschaftliche Publikationen und/oder Forschungsdaten zugänglich hält.
[33] SerWisS: `https://serwiss.bib.hs-hannover.de/home`, zuletzt geprüft am 14.08.17
[34] arXiv: `https://arxiv.org/`, zuletzt geprüft am 14.08.17
[35] bioRxiv: `http://www.biorxiv.org/`, zuletzt geprüft am 14.08.17
[36] SocArXiv: `https://osf.io/preprints/socarxiv`, zuletzt geprüft am 14.08.17

anderen nicht. Im Open Science wird das Open Peer Review bevorzugt. So können die Publikationen öffentlich in der Community diskutiert werden. Beispielsweise werden die Peer Reviews im Open Access Verlag Frontiers gemeinsam mit dem Namen der Gutachter*in und der Publikation veröffentlicht (Hacker 2017, S. 288).

Für die Sicherung der Qualität der Publikationen im Open Science stehen den Wissenschaftler*innen Werkzeuge zur Verfügung, die bei der Suche nach geeigneten Open Access Journals Hilfestellung bieten. So stellt das Directory of Open Access Journals (DOAJ)[37] beispielsweise ein Verzeichnis über Open Access Journals zur Verfügung, bei denen die wissenschaftliche Qualität durch die Herausgeber*in oder durch ein Peer Review Verfahren gewährleistet ist (Schmitz 2017, S. 300f).

3.4. Forschungsdaten und der Einsatz von Data- und Text Mining

Open Access im Open Science beinhaltet neben dem freien Zugang zu den Publikationen auch den freien Zugang zu den zugrundeliegenden Metadaten, den Rohdaten oder dem Bildmaterial (Rosenbaum 2016, S. 46f). Seit einigen Jahren[38] können qualitätsgesicherte Datensätze als eigenständige wissenschaftliche Beiträge publiziert werden (Pfeiffenberger 2017, S. 333). Die Eigenständigkeit der Forschungsdaten beinhaltet auch, dass sie zitierfähig und auch mit dem veröffentlichten Artikel verlinkt sind. Umgesetzt wird dies mit Linked Open Data,[39] dabei werden die Forschungsdaten mit Metadaten versehen und sind über einen Digital Object Identifier (DOI)[40] eindeutig identifizierbar (Ritze u. a. 2013, S. 122). Eine große Herausforderung, bei der Nachnutzung der Forschungsdaten, ist der Umgang mit den oft umfangreichen Datenmengen. Eine weitere Herausforderung ist es Methoden zu entwickeln und einzusetzen um diese Forschungsdaten nachnutzen zu können. Für die Auswertung der großen Datenmenge kann und wird Data- und Text Mining eingesetzt (Pfeiffenberger 2017, S. 335).[41]

Der Einsatz von Data- und Text Mining bei der Auswertung von Forschungsdaten, die dadurch erleichterte Nachnutzung der Daten und die in diesem Kapitel 3 aufgenommenen Definitionen sind Motivation und Ausgangslage für die erstellten Methoden in dieser Masterarbeit.

[37] DOAJ: `https://doaj.org/`, zuletzt geprüft am 14.08.17

[38] Pfeiffenberger nennt einen Zeitraum von 2012/2013 (Pfeiffenberger 2017, S. 334).

[39] Linked Open Data meint verschiedenartige freie Daten im Internet, die über eine Uniform Resource Identifier (URI) identifiziert aufgerufen werden können.

[40] DOIs sind eindeutige Bezeichner für u.a. digitale Objekte, beispielsweise Forschungsdaten. Für das Fachgebiet Informatik vergibt die TIB DOI-Namen für Forschungsdaten. TIB: `https://www.tib.eu/de/publizieren-archivieren/doi-service/doi-registrierung/` und `https://www.tib.eu/de/publizieren-archivieren/doi-service/doi-registrierung/`

[41] Pfeiffenberger spricht hier vom Einsatz von *maschine learning*.

4. Aufbau der Kategoriensystematik der Wikimedia Foundation

Die Wikimedia Foundation fördert als gemeinnützige Gesellschaft freie Inhalte und Wissen. Diese Inhalte werden von Freiwilligen erstellt und über Wikis[42] online zugänglich gemacht. Zu den Projekten der Wikimedia Foundation gehören u.a. die Wikipedia (Lexikonartikel), die Wikimedia Commons (Mediendatenbank) und die Wikidata (mehrsprachige Datenbank für Metadaten) (Wikimedia 2017). Die Kategoriesystematik in den Projekten der Wikimedia Foundation bestehen, neben den von den freiwilligen Helfer*innen erstellten Kategorien, auch aus vorhandenen Normdaten. So können die Normdaten von Gemeinsame Normdatei (GND), Library of Congress Control Number (LCCN), National Diet Library (NDL) und Virtual International Authority File (VIAF) in die Artikel der Wikipedia eingebunden werden (Voss u. a. 2014). Ein Beispiel dazu ist der Sachbegriff Fotovoltaik[43] aus der GND. Der Sachbegriff Fotovoltaik ist ein Synonym von Photovoltaik. In der Wikipedia gibt es eine Übereinstimmung mit der Kategorie Photovoltaik, siehe dazu Abbildung 4.4.[44] Mithilfe der Normdaten in der Wikipedia können die Artikel mit den Normdaten

Abbildung 4.4: Verwendung der GND für Kategorie in der Wikipedia

einiger Nationalbibliotheken verknüpft werden. Für die Einordnung von Personen können beispielsweise die Nummern aus der GND, der LCCN oder der ID aus dem Projekt VIAF verwendet werden. Bei der VIAF werden die Normdaten aus einigen Institutionen und Nationalbibliotheken zusammengefasst.

[42] Laut dem Gabler Wirtschaftslexikon wird unter Wiki ein Autor*innen-System für Webseiten verstanden. Diese können online ohne HTML-Kenntnisse geändert und erstellt werden (Gabler 2017).

[43] Sachbegriff Fotovoltaik in der GND: `http://swb.bsz-bw.de/DB=2.104/SET=2/TTL=1/SHW?FRST=19&ADI_LND=&retrace=0`, zuletzt geprüft am 17.08.17

[44] Kategorie in der Wikipedia: `https://de.wikipedia.org/wiki/Kategorie:Photovoltaik`, zuletzt geprüft am 17.08.17

Für die Suche nach Normdaten, die für die Kategorisierung eingesetzt werden können, kann das Normdaten-Skript von Schnarks eingesetzt werden (Wikipedia 2017f).[45]

4.1. Kategorien der Wikipedia

Die Artikelseiten[46] in der Wikipedia werden, mithilfe der vergebenen Kategorien, in eine inhaltliche Syntax eingeordnet. Die Kategorien wiederum bestehen aus Themenkategorien, Objektkategorien, Strukturkategorien und Metakategorien die aufgeteilt werden können in Ober- und Unterkategorien (Wikipedia 2017i). Dies ist in der englischsprachigen Version der Wikipedia auch so umgesetzt (Wikipedia 2017h). Dabei kann eine Seite mehreren Kategorien zugewiesen werden. Die Kategoriesyntax von Wikipedia kennt zwei Prinzipien – die Facettenklassifikation und die hierarchische Klassifikation (Wikipedia 2017i). Bei der Facettenklassifikation wird ein Themengebiet in unterschiedliche, gleichwertige Wissensbereiche unterteilt, so kann ein Artikel mehrere Kategorien von unterschiedlichen Bereichen eines Themengebietes erhalten. Die Kategorie setzt sich dadurch aus mehreren Facetten zusammen (Wikipedia 2017a). Der Vorteil einer Facettenklassifikation ist, dass hochkomplexe Artikel durch die einzelnen Facetten (Ansichten) detailliert klassifiziert werden können (TerminosaurusRex 2007). Ein Beispiel für die Facettenklassifikation bei Wikipedia ist der Artikel Solarzelle aus Abbildung 4.5. Dieser Artikel hat die Kategorien Photovoltaik, Halbleiterbauelement, Solarmodul und Optoelektronik erhalten.[47] Bei der hierarchi-

Abbildung 4.5: Beispiel einer Facettenklassifikation bei Wikipedia

schen Klassifikation werden Unterbereiche eingeteilt. Diese können mehrfach weiter unterteilt werden. Die Kategorie eines Artikels setzt sich als Schnittmenge aus den

[45] Normdaten-Skript: `https://de.wikipedia.org/wiki/Benutzer:Schnark/js/personendaten/normdaten`, zuletzt geprüft am 19.08.17

[46] Die Seiten der Wikipedia, die im Artikelnamensraum erscheinen, werden auch Artikel genannt. Artikel in der Wikipedia: `https://de.wikipedia.org/wiki/Wikipedia:Artikel`, zuletzt geprüft am 17.08.17

[47] Wikipediaseite Solarzelle: `https://de.wikipedia.org/wiki/Solarzelle`, zuletzt geprüft am 17.08.17

Unterkategorien zusammen (Wikipedia 2017i). So ist der Artikel über Kopplungskonstanten in der Kategorie *Quantenfeldtheorie* eingeordnet, siehe Abbildung 4.6. Die Kategorie *Quantenfeldtheorie* wiederum ist eine Unterkategorie von *Quantenphysik*. *Quantenphysik* ist eine Unterkategorie von *Physik nach Fachgebiet*, darauf folgen die Kategorien *Physik*, *Naturwissenschaft*, *Wissenschaft nach Fachgebiet*, *Wissenschaft*, *Wissen*, *Sachsystematik*, *!Hauptkategorie*.[48] An diesem Punkt ist der Einstiegspunkt im hierarchischen Kategoriesystem von Wikipedia erreicht.

Die Kategorien werden, neben der Verwendung aus der im Kapitel 4 erwähnten

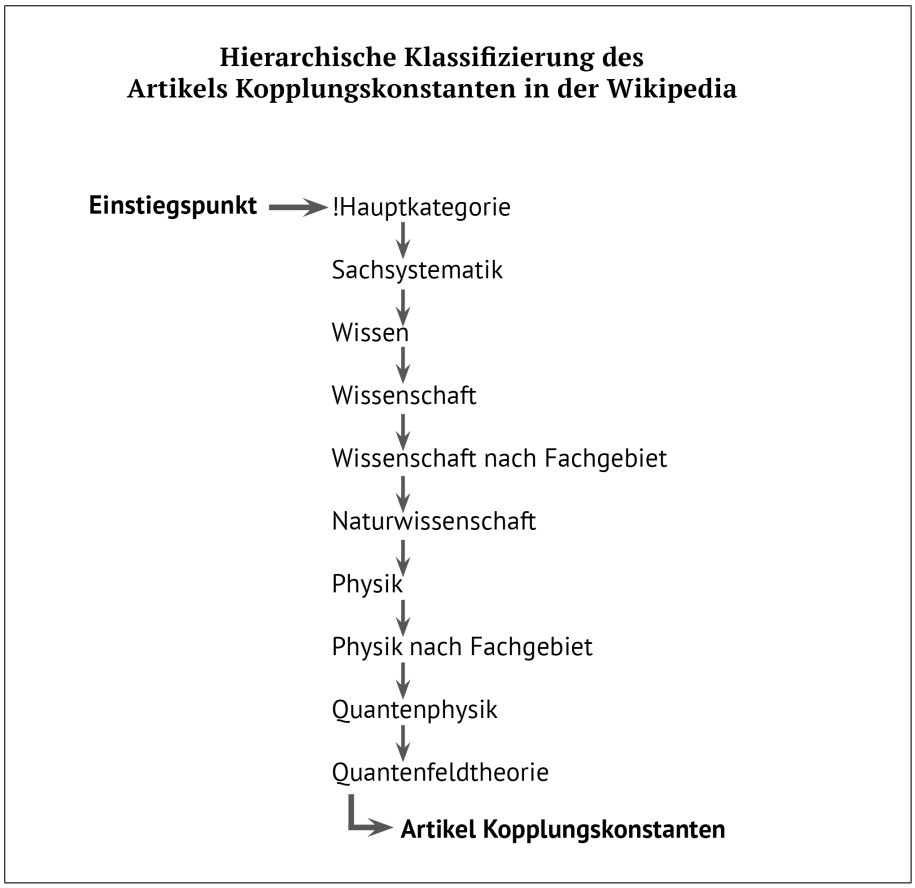


Abbildung 4.6: Beispiel einer hierarchischen Klassifizierung (Kategorienbaum) bei Wikipedia

Normdaten, zum größten Teil von den Mitgliedern aus dem *WikiProjekt Kategorien* erstellt. Diese warten die Einzelkategorien und bearbeiten die Kategorie-Vorschläge der Wikipedia Nutzer*innen (Wikipedia 2017l). In der Wikipedia sollten die Kategorien hierarchisch verbunden sein, d.h. jeder Kategorie werden Oberkategorien und/oder Unterkategorien zugeordnet. Dies kann durch die Verwendung der Facettenklassifizierung erweitert werden. Dabei werden die Artikelseiten in der Wikipedia nach drei Prinzipien eingeordnet, siehe dazu Abbildung 4.7 aus (Wikipedia 2017d).

[48] Hauptkategorie der Wikipedia: `https://de.wikipedia.org/wiki/Kategorie:!Hauptkategorie`, zuletzt geprüft am 17.08.17

Die Prinzipien, nach denen Kategorien vergeben werden können, sind *Gehört zum Fachgebiet*, *Ist ein Teil von* und der *Örtlichen oder zeitlichen Fixierung* (Wikipedia 2017d). Wenn eine Kategorie für eine Artikelseite vergeben wird, darf sie nicht auch in einer

Beispiel	Gehört zum Fachgebiet	Ist ein (Teil von)	Örtliche/zeitliche Fixierung
Charles Darwin	Kategorie:Biologe	Kategorie:Autor, Kategorie:Mann	Kategorie:Brite, Kategorie:Geboren 1809, Kategorie:Gestorben 1882
Cent (Währung)	Kategorie:Euro	Kategorie:Währungseinheit	

Abbildung 4.7: Drei Prinzipien bei der Wahl der Kategorien in der Wikipedia

der Oberkategorien und/oder Unterkategorien vorhanden sein. Im Zweifelsfall wird der Artikel in einer Oberkategorie eines Fachgebietes belassen. Klassifikator*innen aus den jeweiligen Fachbereichen ordnen die Artikel detailliert nach (Wikipedia 2017d). Die vorhandenen Kategorien, die für Artikel in der Wikipedia verwendet werden können, werden im Kategorienbaum gelistet und können interaktiv durchsucht werden (Wikipedia 2016). Des Weiteren gibt es die Bearbeitungsumgebung *VisualEditor*, mit der nachträglich Kategorien geändert oder hinzugefügt werden können (Wikipedia 2017e). In der Auflistung der einzelnen Kategorien werden auch jeweils die Artikelseiten, die mit in dieser Kategorie eingeordnet wurden, angezeigt (Wikipedia 2017d) und (Wikipedia 2017h).

4.2. Kategorisierung von Bilder in Artikeln der Wikipedia

Für Bilder, die zu einem Wikipedia-Artikel hinzugefügt werden, können Kategorien vergeben werden. Die Kategorie kann für den Artikel und für das Bild vergeben werden, oder auch nur für ein Bild. Die Kategorien einer Bilddatei sind dabei in der Regel meistens Unterkategorien der Kategorie eines Artikels und auch zusätzlich eine Unterkategorie der Kategorie für Dateien. Die Kategorien für Dateien befinden sich im Namensraum[49] *Datei*[50] (Wikipedia 2017g). Mithilfe der Namensräume werden Inhalte in der Wikipedia strukturiert. Sie beginnen mit einem Präfix, beispielsweise *Kategorie:*[51] gefolgt von der Bezeichnung für die Kategorie (Wikipedia 2017j). Beispielhaft dafür wären es dann für den Artikel *Measurement while drilling* (siehe Abbildung 4.8) die Kategorien aus der Abbildung 4.9: *Drilling technology* und *Telemetry*.

[49] In der englischsprachigen Wikipedia *Namespaces*.
[50] In der englischsprachigen Wikipedia *File*.
[51] In der englischsprachigen Wikipedia *Categories*.

Abbildung 4.8: Wikipedia Artikel *Measurement while drilling*

Abbildung 4.9: Namespace und Kategorien des Artikels Measurement while drilling

Eine Übersicht der Namensräume ist in der Abbildung 4.10 zu sehen (Wikipedia 2017c). Die markierten Namensräume sind die, die in dieser Arbeit hauptsächlich verwendet werden.[52] Das sind die Namensräume *Artikel*, *Datei* und *Kategorie*.[53] Der Namensraum Datei beginnt mit dem Präfix *File:* (Wikipedia 2017c) oder in der deutschen Sprachversion der Wikipedia mit *Datei:*.[54] Der Namensraum für die Artikel gehört zum Hauptnamensraum und hat daher keinen Präfix (Wikipedia 2017k). Der Namensraum für die Kategorien hat den Präfix *Categories:* (Wikipedia 2017b).

Die Bilder können aber auch gleich bei Wikimedia Commons, dem Medienarchiv, hochgeladen und dann in einem Artikel eingebunden werden. Bilder, die frei verwendet werden können, werden bei einem Upload in einen Wikipedia Artikel zusätzlich zu Wikimedia Commons übertragen (Wikipedia 2017g).

[52] Nicht markiert ist der Namensraum *Medien*, dieser Namensraum wird nur für Links zu den Mediendateien verwendet.

[53] Die Grafik wurde nach `https://en.wikipedia.org/wiki/Help:Files` erstellt, zuletzt geprüft am 19.08.17

[54] Beschreibung der Namensräume für die deutsche Version der Wikipedia: `https://de.wikipedia.org/wiki/Hilfe:Namensr%C3%A4ume`

Wikipedia data structure			
Namespaces			
Subject namespaces		Talk namespaces	
0	(Main/Article)	Talk	1
2	User	User talk	3
4	Wikipedia	Wikipedia talk	5
6	File	File talk	7
8	MediaWiki	MediaWiki talk	9
10	Template	Template talk	11
12	Help	Help talk	13
14	Category	Category talk	15
100	Portal	Portal talk	101
108	Book	Book talk	109
118	Draft	Draft talk	119
446	Education Program	Education Program talk	447
710	TimedText	TimedText talk	711
828	Module	Module talk	829
2300	Gadget	Gadget talk	2301
2302	Gadget definition	Gadget definition talk	2303
Virtual namespaces			
-1	Special		
-2	Media		

Abbildung 4.10: Namensräume für Artikel, Datei und Kategorie

4.3. Kategorien in Wikimedia Commons

Wikimedia Commons ist die Mediensammlung der Wikimedia Foundation, hier werden freie Medieninhalte bereitgestellt. Diese Medieninhalte können in den Projekten z.B in den Artikeln der Wikipedia sprachunabhängig eingebunden werden (Commons 2017f). Die Mediendateien, die bei Wikimedia Commons, hochgeladen werden, müssen gemeinfrei[55] und frei lizensiert sein. Des Weiteren dürfen nur freie Dateiformate hochgeladen werden.[56] Die Lizenzbestimmungen von Wikimedia Commons sehen vor, dass die Person, die die Medien hochlädt, auch die Beweispflicht hat aufzuzeigen, dass die Dateien veröffentlicht und zur Nachnutzung verwendet werden können.[57] Nach dem *vorbeugendem Prinzip* von Wikimedia Commons werden Mediendateien gelöscht wenn Zweifel bezüglich einer Datei bestehen (Commons 2017e). In der Tabelle 4.1 werden die Kategorien[58] der Wurzelkategorien *Category:CommonsRoot*

[55] Es besteht kein Urheberrechtsanspruch

[56] Liste mit erlaubten Dateiformaten: `https://commons.wikimedia.org/wiki/Commons:Project_scope/Allowable_file_types`, zuletzt geprüft am 21.08.17

[57] Dies wird im NOA Projekt für jede Abbildung unterschiedlich ausfallen, da mehrere Verlage, Journals und Wissenschaftler*innen an den Artikeln und somit auch an den verwendeten Abbildungen mitgewirkt haben. Aus der NOA Datenbank werden daher nur die frei lizensierten Abbildungen zu Wikimedia Commons übertragen.

[58] Category:Media_types `https://commons.wikimedia.org/wiki/Category:Media_types`
Category:Categories `https://commons.wikimedia.org/wiki/Category:Categories`
Category:Commons `https://commons.wikimedia.org/wiki/Category:Commons`

Tabelle 4.1: Unterkategorien von Category:CommonsRoot bei Wikimedia Commons

Kategorie	*Umfang/Funktion*
Media types	Einordnung nach Typ z.B. Audio, Bilder, Animationen.
Categories	Zuordnung über z.B. Material. Zusätzlich Unterkategorie von Topics.
Commons	Kategorien für Wartung z.B. für Medien, die gelöscht werden sollen.
Copyright statuses	Zuordnung nach Lizenz der Mediendateien.
Featured items	Kategorien für ausgewählte und besondere Mediendateien.
Gallery pages	Zuordnung u.a. nach Benutzergalerien
Media by source	Verwendung für Einordnung nach Quellen z.B. Sammlungen.
Topics	Wurzelkategorie der thematischen Zuordnung.

(Commons 2017b) von Wikimedia Commons aufgezeigt. Die Wurzelkategorie für die Abbildungen aus dem NOA Projekt wäre dann [59] *Category:Topics*, da hier die thematische Einordnung beginnt und jede Mediendatei mindestens in einer Unterkategorie vorkommen sollte (Commons 2017d). Die Mediendateien werden so bei einer thematischen Suche gefunden.

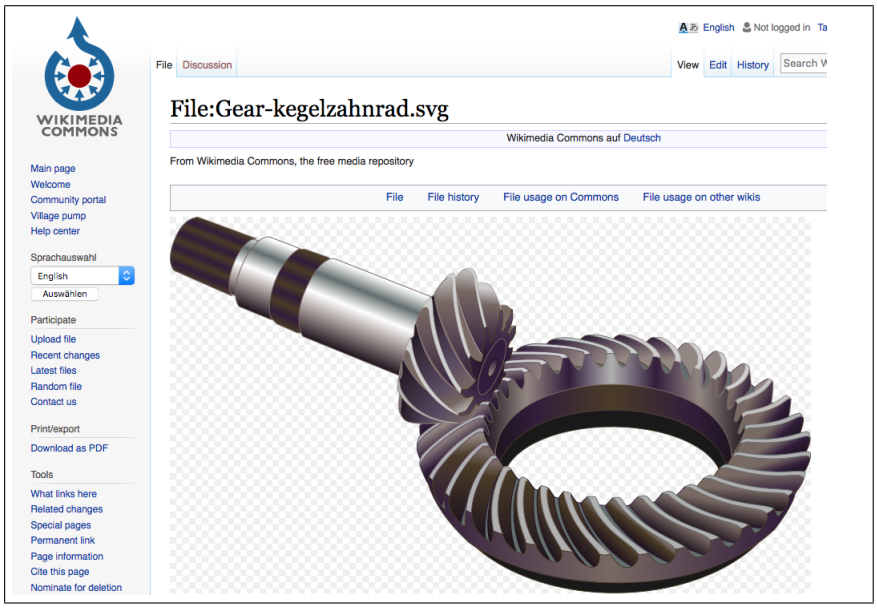


Abbildung 4.11: Mediendatei File:Gear-kegelzahnrad.svg in Wikimedia Commons

Category:Copyright_statuses https://commons.wikimedia.org/wiki/Category:Copyright_statuses
Category:Featured_items https://commons.wikimedia.org/wiki/Category:Featured_items
Category:Gallery_pages https://commons.wikimedia.org/wiki/Category:Gallery_pages
Category:Media_by_source https://commons.wikimedia.org/wiki/Category:Media_by_source
Category:Topics https://commons.wikimedia.org/wiki/Category:Topics, zuletzt geprüft am 22.08.17//

[59] Es wäre die Wurzelkategorie, wenn die Abbildungen aus NOA nach den Kategorien von Wikimedia Commons eingeordnet werden würden. Dies geschieht in dieser Arbeit jedoch anhand der Kategorien der Wikipedia.

Die Umsetzung der Kategorien wird anhand der Datei *File:Gear-kegelzahnrad.svg*[60] exemplarisch vorgestellt. Die grafische Darstellung aus Abbildung 4.11 stellt ein Kegelzahnrad dar und wurde 2007 von Myriam Thyes erstellt. Die Kategorien dieser Datei werden in Abbildung 4.12 aufgezeigt. Das sind die Kategorien *Hypoid bevel gears Computer and technical equipment with transparent background*. Zusätzlich sind weitere Kategorien vergeben, die sich nicht auf den Inhalt beziehen, beispielsweise *Pictures of the Year (2010)*. In der Abbildung 4.13 sind die Wikipedia Artikelseiten[61] verlinkt, die diese Mediendatei eingebunden haben.

Abbildung 4.12: Kategorien von File:Gear-kegelzahnrad.svg

Abbildung 4.13: Wikipedia Artikel mit File:Gear-kegelzahnrad.svg

Vergabe der Kategorien:

Die Kategorien für Medien in Wikimedia Commons sind als Polyhierarchie strukturiert, bei der jede Kategorie mehrere Elternkategorien haben kann. Die Mediendatei darf aber nicht in eine Kategorie und zusätzlich in der dazugehörigen Elternkategorie eingeordnet werden, siehe Abbildung 4.14 (Commons 2017c, Grafik). D.h. die Abbildung *File:Gear-kegelzahnrad.svg* sollte nicht in die Kategorie *Category:Equipment* eingeordnet werden, weil sie schon der Unterkategorie davon in *Computer and technical equipment with transparent background* zugeordnet ist und diese spezifischer ist (Commons 2017d).

[60] Link zur Mediendatei `https://commons.wikimedia.org/wiki/File:Gear-kegelzahnrad.svg`
[61] Die Bildschirmkopie ist ein Auszug der Artikelseiten, es sind nur die Artikelseiten aus der deutschen und englischen Wikipedia aufgenommen.

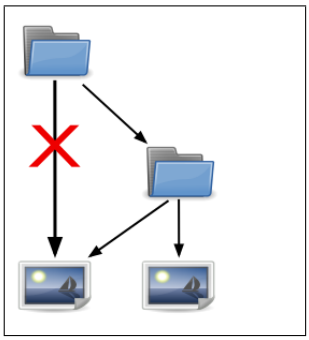


Abbildung 4.14: Kategorienstruktur Wikimedia Commons

4.4. Kategorien in Wikidata

Ein weiteres Projekt der Wikimedia Foundation ist Wikidata[62]. Wikidata ist eine freie Datenbank, an der alle mitarbeiten dürfen. Die Daten, die gesammelt werden, dienen zur Unterstützung von z.B. Wikimedia Commons und Wikipedia. In Wikidata werden Metadaten wie beispielsweise Fakten und Orte mehrsprachig gesammelt und können in den einzelnen Sprachversionen der Wikipedia-Artikel eingefügt werden (Wikidata 2017b). Themenbezoge Kategorien werden in Wikidata als *Sachbegriffe*[63] geführt. Unter (Wikidata 2017c) wird eine Liste der vorhandenen Eigenschaften u.a. auch der Sachbegriffe geführt. Am 23.08.17 waren es insgesamt nur 240 Sachbegriffe aus 20 Themengebieten. Beispielsweise werden für das Fachgebiet Chemie nur 31 Sachbegriffe verwendet. Dadurch könnten die Abbildungen aus NOA nur unzureichend kategorisiert werden, deshalb werden diese Sachbegriffe für die weitere Vorgehensweise in dieser Arbeit vernachlässigt.

Des Weiteren können auch keine Abbildungen zu Wikidata hochgeladen werden. Die Abbildungen werden bei Wikimedia Commons hochgeladen und können dann in einen Wikipedia-Artikel eingebunden werden. Bei Wikidata werden dann die Kategorien von Wikimedia Commons angezeigt (Wikidata 2017a), siehe dazu die Abbildung 4.15 aus MediaWiki (MediaWiki 2017b). Hier werden beispielsweise die Wikimedia Commons Kategorien *commons:Dactylopterus volitans* und *commons:Pseudorasbora parva* genutzt.

[62] Wikidata: `https://www.wikidata.org/wiki/Wikidata:Main_Page`

[63] Liste der Eigenschaften (Sachbegriffe): `https://www.wikidata.org/wiki/Wikidata:List_of_properties/Summary_table/de`

```
<gallery mode="packed-hover">
Image:Astronotus_ocellatus.jpg|''[[commons:Astronotus ocellatus|Astronotus ocellatus]]'' (Oscar)
Image:Salmonlarvakils.jpg|''[[commons:Salmo salar|Salmo salar]]'' (Salmon Larva)
Image:Georgia Aquarium - Giant Grouper.jpg|''[[commons:Epinephelus lanceolatus|Epinephelus lanceolatus]]'' (Giant gr
Image:Pterois volitans Manado-e.jpg|''[[commons:Pterois volitans|Pterois volitans]]'' (Red Lionfish)
Image:Macropodus opercularis - front (aka).jpg|''[[commons:Macropodus opercularis|Macropodus opercularis]]'' (Paradi
Image:Canthigaster valentini 1.jpg|''[[commons:Canthigaster valentini|Canthigaster valentini]]'' (Valentinni's sharp
Image:Flughahn.jpg|[[Image:POTY ribbon 2007.svg|25px]] ''[[commons:Dactylopterus volitans|Dactylopterus volitans]]'
gurnard)
Image:Fishmarket 01.jpg|''[[commons:Semicossyphus pulcher|Semicossyphus pulcher]]'' (California Sheephead)
Image:Pseudorasbora parva(edited version).jpg|''[[commons:Category:Pseudorasbora parva|Pseudorasbora parva]]'' (Topm
Image:MC Rotfeuerfisch.jpg|''[[commons:Category:Pterois antennata|Pterois antennata]]'' (Antennata Lionfish)
Image:Cleaning station konan.jpg|''[[commons:Novaculichthys taeniourus|Novaculichthys taeniourus]]''
Image:Synchiropus splendidus 2 Luc Viatour.jpg|''[[commons:Synchiropus splendidus|Synchiropus splendidus]]'' (Mandar
File:Psetta maxima Luc Viatour.jpg|''[[commons:Psetta maxima|Psetta maxima]]'' (Turbot)
File:Australian blenny.jpg|''[[commons:Category:Ecsenius|Ecsenius axelrodi]]''
</gallery>
```

Abbildung 4.15: Wikimedia Commons Kategorien in Wikidata

4.5. Vergleich der Kategoriensystematik der Wikimedia Foundation

Für die weitere Vorgehensweise werden im ersten Schritt die Kategorien der Wikipedia genutzt. Die Möglichkeiten, die sich dabei ergeben, sind (siehe Kapitel 4.1):

- In der Wikipedia können Artikel nach Facetten und/oder hierarchisch kategorisiert werden.
- Die Bilder können mit den gleichen Kategorien versehen werden wie die Artikel, in denen sie eingebunden sind, oder auch mit eigenen, die detaillierter ausfallen können.
- Die Bilder, die bei Wikipedia hochgeladen werden, werden zusätzlich zu Wikimedia Commons übertragen. Dabei wird die zugewiesene Kategorie mit übertragen.

Die Kategorien der Wikipedia sollen genutzt werden, um die Abbildungen automatisch einer passenden Kategorie zuweisen zu können. Dieser Schritt wird exemplarisch mit einem Datensatz von 397 Trainingsdaten durchgeführt.

Die Überprüfung der zugewiesenen Kategorien erfolgt manuell anhand schon vorhandener Open Access Abbildungen bei Wikipedia und Wikimedia Commons. Es sind 58 Abbildungen, die in die NOA Datenbank aufgenommen wurden und vorher schon bei Wikimedia Commons vorhanden waren. Die Abbildungen stammen ebenso aus Open Access Veröffentlichungen. Sie wurden automatisiert hochgeladen und nachträglich manuell in Kategorien eingeordnet. Die Bearbeitung der Kategorien erfolgte u.a. von den Benutzern: User:Thiotrix, User:Daniel Mietchen, User:Bluerasberry, User:DePlusJean, User:NeverDoING, User:Snek01, User:Ruslik0 und User:Pierpao.

Bei einigen Abbildungen sind keine Kategorien zur Evaluierung vorhanden,[64] diese Abbildungen werden nicht weiter verwendet. Die Abbildungen bei denen Kategorien vorhanden sind werden mit den automatischen Kategorievorschlägen, die mit den Methoden aus dieser Arbeit erstellt wurden, verglichen.

Die Vorgehensweise der Extraktion der Terme und Nominalphrasen, die für die Zuordnung zu den Kategorien nötig sind, werden im nächsten Kapitel beschrieben. Im Kapitel 6 folgt dann die detaillierte Beschreibung der Vorgehensweise für die Vergabe der Wikipedia Kategorien.

[64] Einige der Abbildungen, die zur Evaluation genutzt werden, haben nur die Kategorie **Media needing categories, u.ä..**

5. Beschriftungen und Textreferenzen der Abbildungen aus NOA

Die Bildunterschriften der Abbildungen sind je nach Journal, Autor und Fachgebiet unterschiedlich detailliert und aussagekräftig erstellt worden. In vielen Bildunterschriften werden Abkürzungen verwendet, die zum Teil im vollständigen Artikel aufgelöst werden. Daneben sind einige Beschriftungen sehr kurz gehalten, deshalb werden die Textstellen im Artikel, die sich auf die Abbildung beziehen, mit dazugenommen.

Beispielhaft ist der Aufbau einer Bildunterschrift aus dem Fachgebiet Ingenieurwesen und dem Journal *Mathematical Problems in Engineering* in der Abbildung 5.16[65] zu sehen. Als Suchbegriff wurde *gravitational waves* verwendet. Diese Abbildung wird von den Autoren mit drei ausführlichen Sätzen beschrieben.

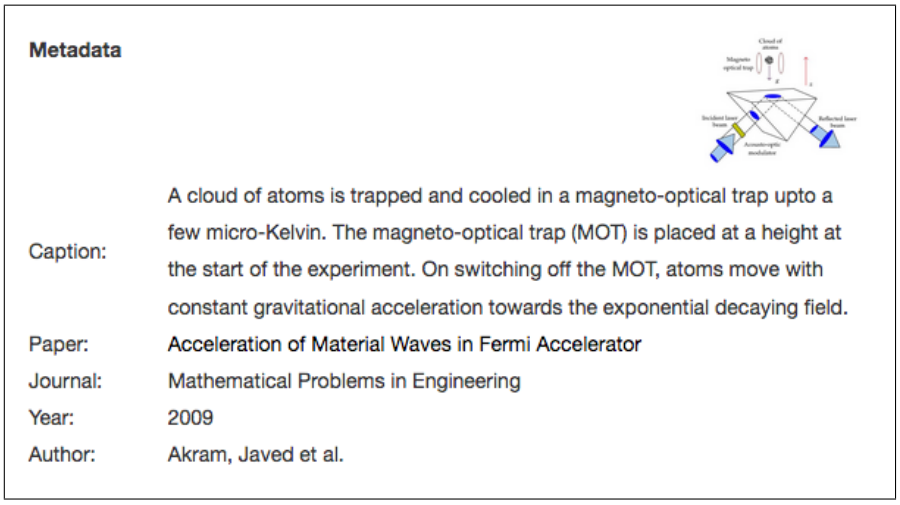


Abbildung 5.16: Bildbeschriftung einer Abbildung aus NOA

5.1. Umfang und Aufbau der Bildbeschriftungen

Die Termextraktion (*engl. keyword extraction*) der Bildbeschriftungen erfolgt beispielhaft an 397 Abbildungen, die in verschiedenen Open Access Journals von den Verlagen Copernicus, Hindawi, Frontiers und Springer Open publiziert wurden. Zusätzlich zu den Beschriftungen sind auch Texte aus dem Artikel vorhanden, in denen auf die Abbildungen eingegangen wird.[66] Die Daten der Abbildungen liegen im Dateiformat

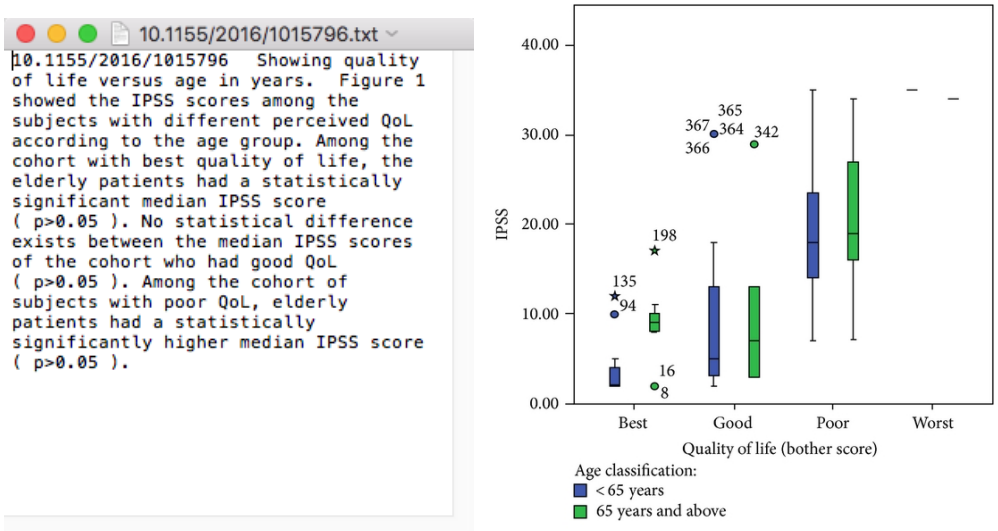
[65] Der vollständige Artikel inkl. Abbildung und Beschriftung kann unter `https://www.hindawi.com/journals/mpe/2009/246438/` aufgerufen werden. Zuletzt geprüft am 24.08.17

[66] Die 397 Bildbeschriftungen und Textreferenzen wurden von Lucia Sohmen am 15.08.17 aus der NOA Datenbank exportiert und für weitere Arbeiten dem NOA Projektteam zur Verfügung gestellt. Aus dieser Datei wurden die Daten der Artikel, die keine Angaben zu einer Abbildungen hatten, gelöscht. Für die weitere Vorgehensweise werden 397 Datensätze genutzt.

Comma-separated values (CSV) vor, eine erste Analyse zur Orientierung wurde im Programm Excel durchgeführt.

Überblick über die vorhandenen Datensätze

Im arithmetischen Mittel[67] haben die Bildunterschriften 350,78 Zeichen. In diesen Zeichen sind auch Leerzeichen und Zahlen enthalten. Die längste Beschriftung hat 3055 Zeichen[68] und die kürzeste 12 Zeichen. Insgesamt haben 95 Beschriftungen eine Zeichenlänge von über 500 Zeichen.[69] Der Umfang der Referenzstellen fällt sehr unterschiedlich aus. Im arithmetischen Mittel haben die Textreferenzen 1687 Zeichen, von 65 Abbildungen konnten keine Referenzstellen automatisiert ermittelt werden, d.h. in der Datenbank des NOA Projektes lagen keine Informationen dazu vor. Die Abbildung mit den meisten Zeichen in den Referenzstellen hat 13.756 Zeichen und insgesamt haben 100 Abbildungen über 1.900 Zeichen in den Textreferenzen.[70] Die Bildbeschriftungen und Textreferenzen der Abbildungen wurden jeweils in eine eigene Textdatei (TXT)-Datei gespeichert. Die Dateinamen der Datensätze sind die DOIs der Artikel, in denen die Abbildungen eingebunden sind. Sollten mehrere Abbildungen aus einem Artikel stammen, so wird der Dateiname durch die Bild-ID aus der Datenbank von NOA ergänzt. In der Abbildung 5.17 ist ein Beispieldatensatz mit den Rohdaten und die grafische Ansicht für eine Abbildung aus dem NOA Projekt zu sehen.



(a) Die Beschriftung und die Textreferenz (b) Die grafische Ansicht

Abbildung 5.17: Vorhandene Informationen zu einer Abbildung

[67] Für das Ermitteln des Wertes wurde die Funktion =MITTELWERT() genutzt.
[68] Verwendete Funktion: =MAX()
[69] Genutzte Funktion =KGRÖSSTE())
[70] Die Werte wurden mit den gleichen Funktionen wie bei den Bildunterschriften ermittelt.

5.2. Termextraktion aus Bildbeschriftungen und Textreferenzen

Die Umsetzung der Informationsextraktion erfolgt als Programmdokument (Notebook) mit IPython.[71] Eine schematische Darstellung der Architektur der Informationsextraktion ist in der Abbildung 5.18 zu sehen. Der vollständige Python Code dazu befindet sich im Anhang unter B.1.[72]

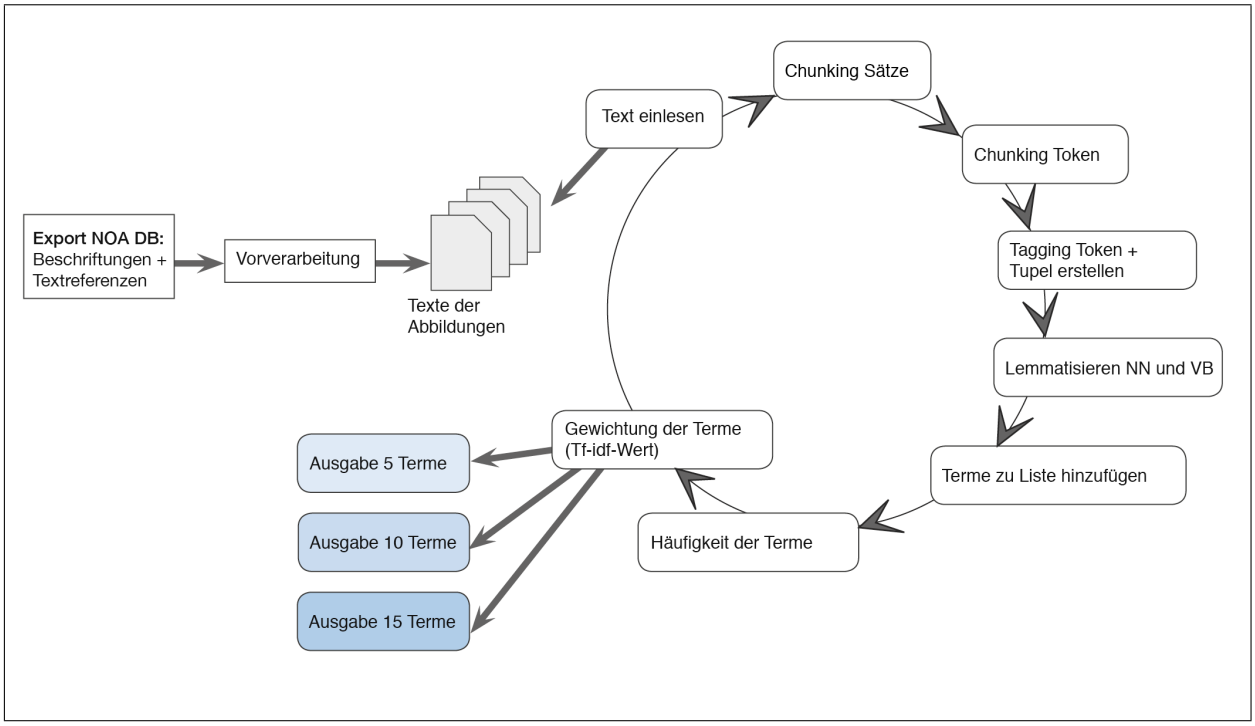


Abbildung 5.18: Darstellung der Termextraktion

Eingesetzt wird die Textverarbeitungsbibliothek Natural Language Toolkit (NLTK)[73] und daraus die Module *nltk.sent_tokenize()* für das Aufteilen von Strings in einzelne Sätze, *nltk.word_tokenize()* für das Aufteilen der Sätze in einzelne Wörter und *nltk.FreqDist()* für die Häufigkeitsverteilung der einzelnen Wörter.[74] Nach dem Aufteilen des Textes in einzelne Wörter (Token) werden diese mit der Methode *tagger.tag_text()* von TreeTagger (Schmid 1994) in eine Liste von Token und den dazu ermittelten Tags versehen. Das Wort *Structure* wird beispielsweise mit einem Tag *NP* für Eigennamen annotiert:

```
Structure\tNP\tStructure
```

[71] IPython: `https://ipython.org/`, zuletzt geprüft am 05.09.17

[72] Die Python Codes, die in dieser Arbeit verwendet werden, basieren auf die Notebooks aus dem Seminar *Text Mining* der Hochschule Hannover Wintersemester 2016 von Prof. Dr. Christian Wartena.

[73] NLTK: `http://www.nltk.org/`

[74] Modul Tokenize aus NLTK: `http://www.nltk.org/_modules/nltk/tokenize.html` und Modul FreqDist: `http://www.nltk.org/api/nltk.html`, zuletzt geprüft am 05.09.17

Danach werden diese Listen mit der Methode *treetaggerwrapper.make_tags()* in Tupel umgewandelt. Das gleiche Beispiel sieht dann als Tupel folgendermaßen aus:

```
Tag(word='Structure', pos='NP', lemma='Structure')
```

Für die Bestimmung der Grundform der Wörter (Lemmatisierung) wird das Modul *treetaggerwrapper*[75] von Laurent Pointal verwendet.[76] Im Anschluss werden die Token, die die Tags *NN* (Nomen Singular), *NNS* (Nomen Plural) oder *VB* (Verben) besitzen, zur Liste der Terme hinzugefügt, ausgezählt und ausgegeben. Für den späteren Vergleich werden hier drei Varianten erstellt. Die Ausgabe von 5 Termen für den Abgleich der Wikipedia Kategorien, die Ausgabe von 10 Termen und die Ausgabe von 15 Termen. Im Kapitel 6 erfolgt das Mapping auf die Kategorien.

5.2.1. Struktur der Terme

Mit dem Begriff *Terme* werden in dieser Arbeit die Fachausdrücke gemeint, die in den Beschriftungen der Abbildungen und in den Textreferenzen im Artikel verwendet werden. Fachterme bestehen zum größten Teil aus Nomina, selten aus Verben und manchmal aus Adverbien und Adjektiven (Heyer u. a. 2008, S. 269). Nach den Termen wird mit dem PoS Tags *NN* für die Nomen in der Singularform, *NNS* für die Nomen in der Pluralform und *VB* für die Grundform der Verben gesucht.[77] Pro Abbildung werden 5, 10 und 15 Terme ausgegeben, in einigen Datensätzen sind weniger vorhanden. Die Adverbien und Adjektive werden bei der Extraktion der Terme nicht mit ausgegeben, da sie für die nächsten Schritte nicht genutzt werden. Das Ermitteln der Nomen und Verben wird mit dem TreeTagger und daraus mit der Methode *tagger.tag_text()* durchgeführt.

5.2.2. PoS Tagging mit dem TreeTagger und dem Penn Treebank Tagset

Tagger sind Programme, die den Wortformen ihre Wortart zuordnen. Dafür wird ein entwickeltes Sprachmodell verwendet, das aus manuell getaggten Texten erstellt wurde. Mit diesem Sprachmodell können wiederum weitere Texte automatisch getaggt werden. Das Taggen wird dabei mit dem ausgewähltem Tagset durchgeführt (Heyer u. a. 2008, S. 126f). Die Funktionsweisen der Tagger basieren auf Hidden Markov-Modelle. Diese wiederum auf Markov-Modelle. Markov-Modelle sind endliche Automaten (Finite State Automatons) bei denen die Zustandsübergänge Wahrscheinlichkeiten erhalten. Bei den Markov-Modellen wird jeder Einheit[78] aus dem

[75] Der Wrapper basiert auf den TreeTagger von Helmut Schmid. `http://www.cis.uni-muenchen.de/~schmid/`, zuletzt geprüft am 05.09.17

[76] Treetaggerwrapper Documentation: `http://treetaggerwrapper.readthedocs.io/en/latest/`, zuletzt geprüft am 05.09.17

[77] PoS TagSet: `https://www.sketchengine.co.uk/english-treetagger-pipeline-2/`

[78] Einheiten können Morpheme, Buchstaben, Wörter oder Sätze sein. In dieser Arbeit sind die Einheiten Wörter.

Sprachstrom ein Label zugeordnet. Die Zustandsübergänge erhalten Wahrscheinlichkeiten um das beste Tag vergeben zu können (Jurafsky u. Martin 2000, 122). Speziell in der Sprachverarbeitung, beim Part of Speech (PoS) Tagging, werden Hidden Markov-Modelle (HMM) eingesetzt. Unter PoS Tagging wird die Anreicherung von Wörtern mit ihren Wortarten verstanden. Wortarten sind beispielsweise Nomen, Verben und Adjektive. Im Penn Treebank Tagset werden die Wortarten aufgeführt (Dengel 2012, S. 211).

Bei den Markov-Modellen werden nur die Übergangswahrscheinlichkeiten des *eindeutigen Pfades* vom Modell multipliziert, um daraus die Wahrscheinlichkeit für die Symbolfolge zu generieren. Dabei wird angenommen, dass die Übergangswahrscheinlichkeiten konstant bleiben. Im Gegensatz dazu werden bei den HMM die Wahrscheinlichkeiten an den Zustandsübergängen von *allen möglichen Pfaden* multipliziert, die Zustände selber sind dabei verborgen (en: hidden). HMM ermitteln neben der Wahrscheinlichkeit einer Symbolfolge auch den optimalen Pfad für eine Symbolfolge und erstellen aus dem Trainingskorpus das beste Sprachmodell (Heyer u. a. 2008, S. 120). Der darauf basierende TreeTagger wurde von Helmut Schmid (Schmid 1994) entwickelt. Für die englische Sprache wurde der TreeTagger mit dem **Penn Treebank Tagset** (Marcus u. a. 1993) trainiert. In der Abbildung 5.19 ist ein Auszug aus dem Penn Treebank Tagset zu sehen, markiert sind die verwendeten Tags.[79]

Penn Treebank Tagset

1.	CC	Coordinating conjunction	25.	TO	to
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential there	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund or present participle
6.	IN	Preposition or subordinating conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd person singular present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd person singular present
9.	JJS	Adjective, superlative	33.	WDT	Wh-determiner
10.	LS	List item marker	34.	WP	Wh-pronoun
11.	MD	Modal	35.	WP$	Possessive wh-pronoun
12.	NN	Noun, singular or mass	36.	WRB	Wh-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PRP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	"	Straight double quote
21.	RBR	Adverb, comparative	45.	'	Left open single quote
22.	RBS	Adverb, superlative	46.	“	Left open double quote
23.	RP	Particle	47.	’	Right close single quote
24.	SYM	Symbol	48.	”	Right close double quote

Abbildung 5.19: Auszug aus dem Penn Treebank Tagset

[79] Auflistung erstellt nach (Marcus u. a. 1993)

5.2.3. Vorkommenshäufigkeit der Terme

Bei der Vorkommenshäufigkeit von Termen werden die erkannten Terme aufaddiert, sie werden nicht in Bezug zu der gesamten Textlänge gesetzt und auch nicht in Bezug zu den anderen Datensätzen. Die Vorkommenshäufigkeit der Terme sagt lediglich aus, wie oft ein Term in einem Datensatz vorkommt. Dabei wird die Gewichtung des Terms durch die Anzahl der Vorkommen bestimmt. Bei der einfachen Ausgabe der Vorkommen der Terme werden auch häufige Substantive wie *Figure* und auch Sonderzeichen wie *Prozent* mit ausgegeben. In der Tabelle 5.2 folgt beispielhaft die Ausgabe der Terme aus drei Datensätzen. In einem nächsten Schritt wird die Vor-

Tabelle 5.2: Ausgabe der Terme und der Termfrequenz

10.1155:2016:1979348.txt	**10.1155:2016:1875357.txt**	**10.1155:2016:1962438.txt**
case, 7	stripe, 7	diagram, 2
DJD, 3,	slot, 6	condition, 2
lesion, 2	size, 6	voltage, 2
patient, 2	wavelength, 5	inverter, 2
instability, 2	time, 4	capacitance, 2
disease, 1	datum, 4	array, 1
osteochondritis, 1	switch, 3	figure, 1
%, 1	fabric, 3	bus, 1
ligament, 1	system, 3	value, 1
age, 1	bit, 3	nF/kW, 1
18–59, 1	conversion', 2),	('weather', 1),
plug, 1	computer, 2	grid, 1
inclusion, 1	control, 2	frame, 1
diagnosis, 1	switching, 2	diode, 1
dissecans, 1	be, 2	switch, 1

kommenshäufigkeit der Terme in Bezug zur Länge des Datensatzes gesetzt sowie in Bezug zu der Frequenz der Terme in allen Datensätzen. Dadurch erhält man die Relevanz eines Terms für einen Datensatz.

5.2.4. Normalisierte Termfrequenz und Tf-idf-Wert der Terme

Die Termfrequenz $\mathbf{f}_{\mathbf{i,m}}$ bezeichnet die Häufigkeit der Vorkommen eines Wortes (Term) $\mathbf{t_i}$ im Text (Dokument) $\mathbf{d_m}$. Bei der normalisierten Termfrequenz $\mathbf{nf}_{\mathbf{i,m}}$ wird die Worthäufigkeit $\mathbf{f_{i,m}}$ in Bezug zur Textlänge (Dokumentlänge) $\mathbf{max}_{\mathbf{j}}\ \mathbf{f}_{\mathbf{j,m}}$ gesetzt (Heyer u. a. 2008, S. 203).

$$nf_{\mathrm{i,m}} = \frac{f_{\mathrm{i,m}}}{max_{\mathrm{j}} f_{\mathrm{j,m}}}$$

Für die Extraktion der Terme in den Abbildungen sind es die beim Tagging erkannten Wörter der Wortformen NN (Nomen Singular), NNS (Nomen Plural) und VB (Verben) in ihrer Grundform (en: Lemma), geteilt durch die Länge der erkannten Wörter (NN,

NNS und VB in der Grundform) im Dokument. Der vollständige Code dazu befindet sich im Anhang unter B.1. Ein Auszug daraus ist in den folgenden Zeilen zu sehen.

```
fdist = nltk.FreqDist(nouns) #Anzahl Vorkommen der Terme
...
fdist[word] = float(fdist[word]) / float(len(nouns)) #Geteilt durch die Laenge des
    Datensatzes
```

Ausgabe: Normalisierte Termfrequenz

In der Tabelle 5.3 wird die normalisierte Frequenz der Terme angezeigt.[80]

Tabelle 5.3: Ausgabe der normalisierten Termfrequenz

10.1155:2016:1979348.txt	**10.1155:2016:1875357.txt**	**10.1155:2016:1962438.txt**
case, 0.23	stripe, 0.07	diagram, 0.07
DJD, 0.1	slot, 0.06	condition, 0.07
lesion, 0.06	size, 0.06	voltage, 0.07
patient, 0.06	wavelength, 0.05	inverter, 0.07
instability, 0.06	time, 0.04	capacitance, 0.07
disease, 0.03	datum, 0.04	array, 0.03
osteochondritis, 0.03	switch, 0.03	figure, 0.03
%, 0.03	fabric, 0.03	bus, 0.03
ligament, 0.03	system, 0.03	value, 0.03
age, 0.03	bit, 0.03	nF/kW, 0.03
18–59, 0.03	conversion, 0.02	weather, 0.03
plug, 0.03	computer, 0.02	grid, 0.03
inclusion, 0.03	control, 0.02	frame, 0.03
diagnosis, 0.03	switching, 0.02	diode, 0.03
dissecans, 0.03	be, 0.02	switch, 0.03

Inverse Dokumentfrequenz

Da die besten[81] Terme nicht in allen Datensätzen möglichst oft vorkommen sollen, sondern nur in dem einen speziellen Dokument der zu beschreibenden Abbildung, kann die inverse Dokumenthäufigkeit **Idf** eingesetzt werden (Heyer u. a. 2008, S. 204). **|d|** ist die Anzahl der vorhandenen Dokumente. In dieser Arbeit sind es 397 Dokumente. Die inverse Dokumenthäufigkeit wird berechnet mit:

$$idf_{\mathrm{i}} = log \frac{|d|}{|d : t_{\mathrm{i}} \in d|}$$

Bei der inversen Dokumentfrequenz $\mathbf{idf_i}$ wird die normalisierte Termfrequenz im Zusammenhang mit der Wichtigkeit eines Wortes (Terms) $\mathbf{t_i}$ für das jeweilige

[80] Die Tf-idf-Werte werden auf zwei Stellen nach dem Komma gekürzt angezeigt.
[81] *Beste Terme* meint die Terme, die den Datensatz und damit auch die Abbildung fachlich am besten beschreiben.

Dokument $\mathbf{d_m}$ berechnet (Gewichtung des Terms) (Baeza-Yates u. Ribeiro-Neto 1999, S. 25ff). Die Zeilen im Python Code, die den Tf-idf-Wert berechnen, sind im Auszug unten zu sehen. Der vollständige Code dazu befindet sich im Anhang unter B.1.

```
...
 df_n = df.get(n,0)
 df[n] = df_n + 1
...
  fdist = nltk.FreqDist(nouns)
    for word in fdist:
        idf = 1.0 + math.log(float(nr_of_docs) / float(df[word]))
        fdist[word] = float(fdist[word]) / float(len(nouns)) * idf
```

Zuerst werden alle Stammformen von *NN*, *NNS* und *VB* in das Dictionary[82] **df** geschrieben. Dabei wird jeweils immer nur das erste Vorkommen inkl. einem Schlüssel gespeichert. Im nächsten Schritt wird der Logarithmus aus der Anzahl der Dokumente durch die Keywort-Schlüssel Paare berechnet. Danach wird die Anzahl der Wörter durch die Länge aller Wörter dividiert und mit dem Wert der Variabel *idf* multipliziert. Dieser Tf-idf-Wert wird im Anschluss mit dem zugehörigem Term ausgegeben. Es werden 5, 10 und 15 Terme mit ihrem Tf-idf-Wert für jeden Datensatz (jeder Abbildung) ausgegeben.

Ausgabe: Inverse Dokumentfrequenz

In der Tabelle 5.4 werden jeweils die 15 Terme mit der größten Relevanz, für drei Datensätze, aufgezeigt.[83]

Tabelle 5.4: Ausgabe der inversen Dokumentfrequenz

10.1155:2016:1979348.txt	**10.1155:2016:1875357.txt**	**10.1155:2016:1962438.txt**
DJD, 0.69,	stripe, 0.55,	capacitance, 0.46,
case, 0.68,	slot, 0.47,	inverter, 0.46,
instability, 0.39,	wavelength, 0.31,	voltage, 0.35,
lesion, 0.32,	fabric, 0.23,	diagram, 0.27,
oat, 0.23,	bit, 0.23,	inductor, 0.25,
dissecans, 0.23,	size, 0.20,	dc, 0.25,
tear, 0.23,	switch, 0.17,	nF/kW, 0.25,
18–59, 0.23,	switching, 0.14,	condition, 0.24,
OCD, 0.23,	conversion, 0.12,	diode, 0.21,
osteochondritis, 0.23,	computer, 0.12,	bus, 0.21,
plug, 0.23,	time, 0.11,	weather, 0.19,
patient, 0.22,	advantage, 0.11,	switch, 0.19,
ligament, 0.18,	datum, 0.10,	array, 0.18,
inclusion, 0.16,	system, 0.10,	filter, 0.17,
diagnosis, 0.15	consideration, 0.10	frame, 0.17

[82] In Dictionarys werden Schlüssel-Objekt Paare gespeichert
[83] Die Tf-idf-Werte werden in dieser Arbeit auf zwei Stellen nach dem Komma verkürzt angezeigt.

Dadurch fallen Wörter wie z.B. *figure* oder Sonderzeichen und Maßeinheiten noch nicht weg, werden aber durch den Tf-idf-Wert nach unten sortiert. Diese Wörter kommen häufig in einem Text vor, aber dadurch, dass sie in allen Texten (Datensätzen) häufig vorkommen, sind sie wenig hilfreich für die Beschreibung einer einzelnen Abbildung. Für die weitere Vorgehensweise werden deswegen drei mögliche Methoden aufgezeigt, durchgeführt und miteinander verglichen. Bei der ersten Variante werden die 15 häufigsten Terme (Sortierung nach dem Tf-idf-Wert) genutzt, um den Abgleich mit den Titeln der Wikipedia Artikeln durchzuführen. Bei der zweiten Variante sind es 10 Terme und bei der dritten sind es 5 Terme, die für den Abgleich genutzt werden. Einige Datensätze der Abbildungen bestehen aus kurzen Beschriftungen und kurzen oder keinen Textreferenzen. In diesen Fällen sind die 5 besten Terme, für die weitere Vorgehensweise, ausreichend und besser geeignet. Das Mapping der Termextraktion auf die Titel der Wikipedia Artikel findet im Kapitel 6 statt. Der vollständige Python Code zu der Extraktion der Terme befindet sich im Anhang unter B.1.

5.3. Extraktion von Nominalphrasen aus Beschriftungen und Textreferenzen

Neben den Termen können auch Phrasen auftreten, die geeignet sind, die Abbildung zu beschreiben und eine Zuordnung zu den Kategorien der Wikipedia zu ermöglichen. Diese Phrasen bestehen aus Wörtern, die häufig gemeinsam auftreten. Wörter, die als Wortpaare auftreten, werden als Kollokation[84] bezeichnet. Ein Zusammenhang zwischen den Wörtern muss bei einer Kollokation nicht vorliegen. Da bei den Wortpaaren, in dieser Arbeit, aber davon ausgegangen werden kann, dass beide Wörter semantisch voneinander abhängig sind, kann die Bezeichnung Kookkurrenz verwendet werden (Heyer u. a. 2008). Bei einer Kookkurrenz besteht eine größere Abhängigkeit von zwei Termen als dies bei reinem Zufall der Fall wäre.[85] Für die

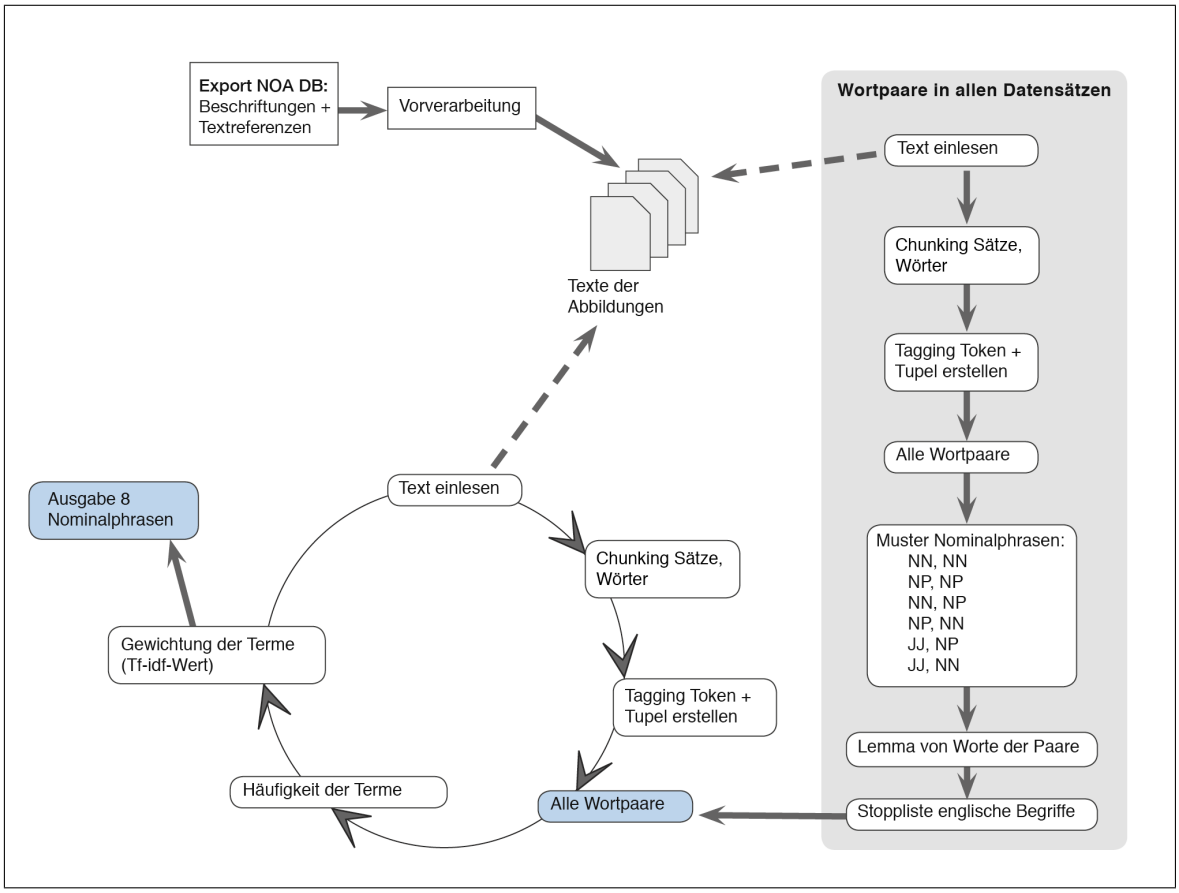


Abbildung 5.20: Modell Extraktion der Nominalphrasen

Extraktion der Nominalphrasen wird, genau wie auch bei der Termextraktion, die Textverarbeitungsbibliothek *NLTK* für das Aufteilen der Strings in Sätze, in einzelne

[84] Openthesaurus: `https://www.openthesaurus.de/synonyme/Kollokation`, zuletzt geprüft am 12.09.17

[85] Openthesaurus: `https://www.openthesaurus.de/synonyme/Kookkurrenz`, zuletzt geprüft am 12.09.17

Wörter und der TreeTagger (Schmid 1994) für die Vergabe der Tags und das Erstellen der Tupel eingesetzt (siehe dazu Kapitel 5.2 Vorgehensweise der Termextraktion).

Wie im Modell in der Abbildung 5.20 zu sehen, sind zwei Durchgänge für die Extraktion der Nominalphrasen notwendig. In einem ersten Schritt werden die Wortpaare aus den Datensätzen ermittelt, die mit dem Muster der Nominalphrasen (siehe dazu Tabelle 5.6) übereinstimmen. Die einzelnen Wörter der Wortpaare werden in ihre Grundform gebracht. Durch den Einsatz einer Stoppliste werden die englischen Stoppwörter reduziert.[86] Die Nominalphrasen werden für die Überprüfung der Wortpaare aus den einzelnen Datensätzen genutzt. In einem zweiten Durchgang werden die Wortpaare für die einzelnen Datensätze extrahiert. Dazu werden die Datensätze wieder in Sätze, dann in Wörter aufgeteilt. Eine Nominalphrase wird jetzt als ein einziges Token (Term) betrachtet. Die Relevanz der Terme (Nominalphrase) wird mithilfe des Tf-idf-Wertes berechnet. Es erfolgt eine Ausgabe von 8 Nominalphrasen für jeden Datensatz. In der Tabelle 5.5 sind die Nominalphrasen und die berechneten Tf-idf Werte von zwei Datensätzen aufgezeigt.

Tabelle 5.5: Beispiele für extrahierte Nominalphrasen

Datensatz: 10.1155:2016:2385429.txt	
Nominalphrase	**Tf-idf-Wert**
meaningful, insight	0.43
sentiment, score	0.43
word, cloud	0.43
data, gathering	0.43
sentiment, analysis	0.43
sentiment	0.43
Datensatz: 10.1155:2016:2350615.txt	
Nominalphrase	**Tf-idf-Wert**
Seribu, Island	1.74
Marine, Science	0.87
Ocean, Acoustics	0.87
Research, location	0.87
Instrumentation, Laboratory	0.87
Bogor, Agricultural	0.87

[86] Die Liste der Stoppwörter `http://www.nltk.org/book/ch02.html` sind im Porter Stemmer integriert, basieren jedoch auf C.J. Rijsbergen (Rijsbergen 1979)

5.3.1. Struktur von Nominalphrasen

Insgesamt werden in der Linguistik mögliche Phrasen unterschieden in Nominalphrase, Präpositionalphrase, Verbalphrase, Adjektivphrase und Adverbphrase (Dengel 2012, S. 212). Die Phrasen, die aus den Bildbeschriftungen und Textreferenzen extrahiert werden, sind in dieser Arbeit Nominalphrasen. Die Struktur von Nominalphrasen kann folgendermaßen aussehen:[87]

```
electrooptical/JJ switch/NN
switch/NN fabric/NN
```

In den verwendeten Datensätzen der Abbildungen werden Wortpaare erkannt, die aus Substantiven, Eigennamen und/oder Adjektiven bestehen. Die vorhandenen Wortphrasen werden in ihre Grundform gebracht, d.h. die weitere Vorgehensweise wird mit den Lemmata der Wörter durchgeführt. Damit diese Wortphrasen extrahiert werden können, werden in dem Datensatz einer Abbildung die Wortartfolgen aus Tabelle 5.6 ermittelt. Mit diesem Bildungsmuster ist die Extraktion von fachspezifischen

Tabelle 5.6: Wortartfolgen der verwendeten Kookkurrenzen

Penn Treebank Tag	**Wortart**
NN, NN	Substantive
NP, NP	Eigennamen
NN, NP	Substantiv gefolgt von Eigennamen
NP, NN	Eigennamen gefolgt von Substantiv
JJ, NP	Adjektiv gefolgt von Eigennamen
JJ, NN	Adjektiv gefolgt von Substantiv

Phrasen möglich. Durch die Prüfung der Wortfolgen und deren Häufigkeit können aussagekräftige Phrasen ermittelt werden (Heyer u. a. 2008, S. 282f).

5.3.2. Syntaktische Analyse der Nominalphrasen

Für die Extraktion der Nominalphrasen werden in dieser Arbeit musterbasierte Text Mining Verfahren angewendet. Diese musterbasierte, syntaktische Analyse wird mithilfe von PoS-Mustern durchgeführt. Die Funktionsweise des Part of Speech Tagging wird im Kapitel 5.2.2 beschrieben. Eine Aufstellung der möglichen PoS-Mustern für die Nominalphrasen ist in der Tabelle 5.6 zu sehen. Dabei wird die Kookkurenzbeziehung zum rechten Wort (Wortform) berücksichtigt (Heyer u. a. 2008, S. 275). Eine Darstellung dieser Beziehung ist in der Abbildung 5.21 markiert dargestellt. Das Wort *Sentiment* wird beim PoS-Tagging als Nomen erkannt und mit dem Tag *NN* versehen. Das Wort das auf *Sentiment* folgt ist *analysis*. *Analysis* ist ebenfalls von der Wortform *NN*, deshalb greift hier die musterbasierte Suche.

[87] Beide Beispiele sind Nominalphrasen aus dem Datensatz 10.1155:2016:1875357.txt. NN wird für die Nomen verwendet und JJ für die Adjektive.

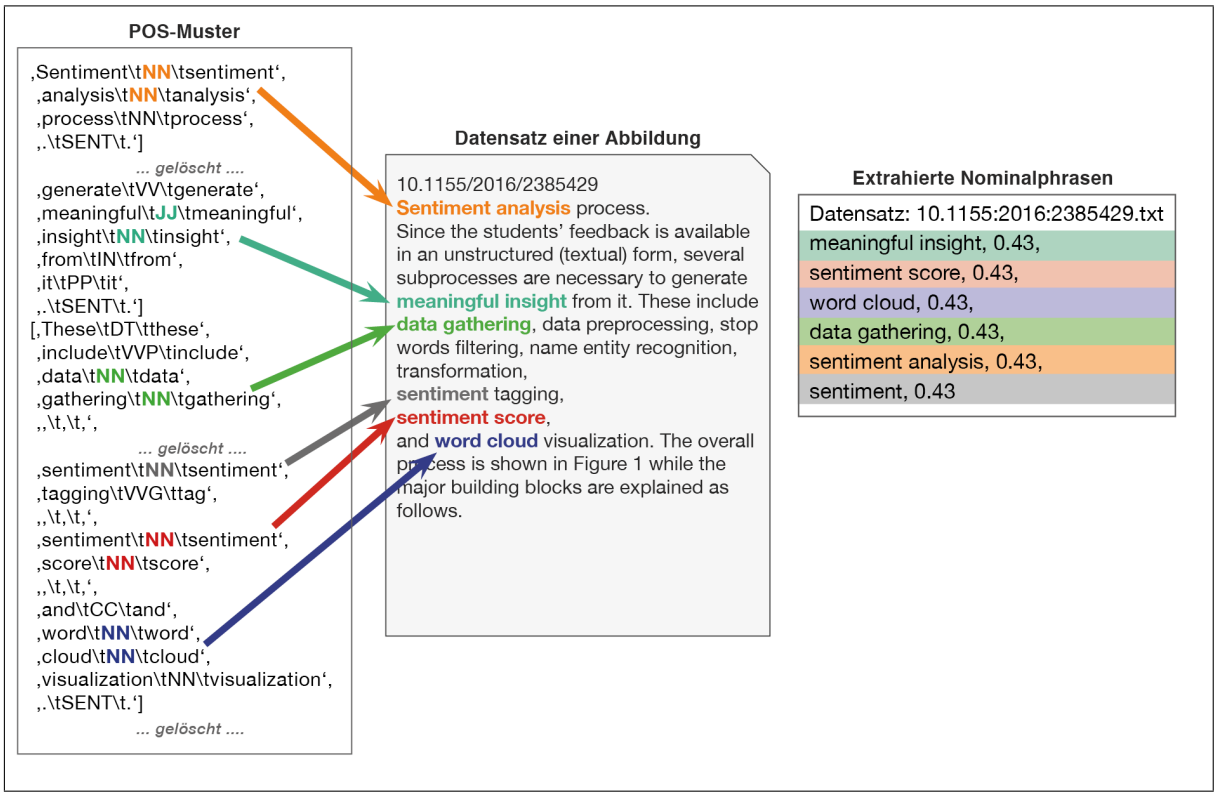


Abbildung 5.21: PoS-Muster eines Datensatzes und extrahierte Nominalphrasen

Die Nominalphrase *sentiment analysis* wurde erkannt und kann mit einem Tf-idf-Wert von *0.43* für das Mapping auf die Kategorien der Wikipedia genutzt werden.

Detaillierter Vorgang der Extraktion der Nominalphrasen:

Im folgenden Code-Ausschnitt werden für die Extraktion der Nominalphrasen alle Texte der Datensätze in Sätze und dann in Wörter aufgeteilt, mehr dazu im Kapitel 5.2. Dort ist die Vorgehensweise für die Termextraktion beschrieben, diese Vorgehensweise wird aber für die linguistische Vorverarbeitung der Extraktion von Nominalphrasen genauso angewendet. Der vollständige Python Code zu der Extraktion der Nominalphrasen befindet sich im Anhang unter B.4.

```
for datei in filelist:
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    words = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        words = [lemma for (word,pos,lemma) in tags2 if pos[0]]
        word_pairs = []
        for i in range(len(tags2)-1):
            t1 = tags2[i]
```

```
            t2 = tags2[i+1]
            if (t1.pos,t2.pos) in npmuster:
                l1 = t1.lemma
                l2 = t2.lemma
                if l1 not in swlist and l2 not in swlist and len(l1) >2 and len(l2) >2:
                    word_pairs.append((l1,l2))
        nr_of_words += len(words)
        fdist.update(words)
        fdist2.update(word_pairs)
```

Die Wörter werden wieder in die Grundform gebracht und können anhand ihrer POS-Tags mit dem Muster für Nominalphrasen überprüft werden. Wie im Kapitel 5.3.1 beschrieben, werden nur Wortartfolgen aus dem Muster der Liste *npmuster* mit den Tags z.B. ('JJ','NN') als Nominalphrase erkannt.

```
npmuster = [('NN','NN'), #Nomen
           ('NP','NP'), #Eigennamen
           ('NN','NP'),
           ('NP','NN'),
           ('JJ','NP'), #Adjektive
           ('JJ','NN')]
```

Das Ergebnis sind alle Vorkommen der Nominalphrasen aus allen Datensätzen, so kann in einem späteren Schritt die Relevanz der Nominalphrase für einen einzigen Datensatz berechnet werden. In den nächsten Schritten werden die Nominalphrasen für einen Datensatz/einer Abbildung extrahiert. Dafür werden wieder alle vorhandenen Substantive gezählt, siehe dazu die Vorgehensweise bei der Termextraktion in Kapitel 5.2.3. Mit diesen Substantiven wird die Funktion *candidates(taglist)* als Argument aufgerufen.

```
def candidates(taglist):
    cand = []
    skip = False
    for i in range(len(taglist)-1) :
        if skip:
            skip = False
            continue
        skip = False
        l1 = taglist[i].lemma
        l2 = taglist[i+1].lemma
        if len(l1) >2 and len(l2) >2:
            if (l1,l2) in NPList:
                skip = True
                cand.append((l1,l2))
    return cand
```

Die Nominalphrasen werden nun wie ein einzelner Term für einen Datensatz extrahiert. Für alle Wörter wird die Grundform des Wortes verwendet. Es erfolgt eine Überprüfung der Wortlänge, damit sie mehr als zwei Buchstaben haben. Danach wird überprüft ob beide Wörter so in der Liste aller Nominalphrasen in *NPList* vorhanden sind. *NPList* wird mit den Wortpaaren gefüllt, die eine Übereinstimmung mit dem Nominalphrasen-Muster aus *npmuster* haben. Diese Wortpaare werden dann zu *cand* hinzugefügt. Für die Gewichtung der Wortpaare wird der Tf-idf-Wert berechnet.

Die Berechnung ist im folgenden Code-Ausschnitt aus der Funktion *extract_kw()* zu sehen. Die detaillierte Vorgehensweise bei der Berechnung des Tf-idf-Wertes wird im Kapitel 5.2.4 beschrieben. Da die Nominalphrasen für die einzelnen Datensätze wie ein Term behandelt werden, ist die Vorgehensweise zu der Termextraktion identisch.

```
for word in fdist:
        idf = 1.0 + math.log(float(nr_of_docs) / float(df[word]))
        fdist[word] = float(fdist[word]) / float(len(nouns)) * idf
    return fdist.most_common(8)
```

Wie in der Rückgabeanweisung (return fdist.most_common(8)) zu sehen ist, werden 8 Nominalphrasen für jeden Datensatz und somit für jede Abbildung zurückgegeben und auch ausgegeben. In der Abbildung 5.21, am Anfang des Kapitels, ist dazu ein Beispiel, für die extrahierten Nominalphrasen einer Abbildung, visuell aufgezeigt.

Ausgabe: Inverse Dokumentfrequenz der Nominalphrasen

In der Tabelle 5.7 wird die Ausgabe der Nominalphrasen, nach Berechnung des Tf-idf-Wertes, von zwei Datensätzen gezeigt. Die Nominalphrasen bestehen aus Wortpaaren, die dem Nominalphrasen-Muster entsprechen und mehr als zwei Buchstaben lang sind.

Tabelle 5.7: Ausgabe von Nominalphrasen mit Tf-idf-Maß für drei Datensätze

Nominalphrasen	**Tf-idf-Wert**
10.1155:2016:2606453.txt	
handheld cradle,	0.54
resection plane,	0.27
iPod cradle,	0.27
distal femur,	0.24
tibial plateau,	0.13
iPod screen,	0.13
Dash Navigation,	0.13
hardware concept,	0.13
10.1155:2016:2642361.txt	
SCD crisis,	0.66
homogenous HbA,	0.33
blood transfusion,	0.33
drug administration,	0.33
simultaneous detection,	0.33
large field,	0.33
sRBC cluster,	0.33
venous blood,	0.33

Die extrahierten Nominalphrasen aus der Tabelle 5.7 werden noch weiter bereinigt und im Anschluss erfolgt das Mapping auf die Titel der Wikipedia Artikel, beschrieben wird es im Kapitel 7. Im nächsten Kapitel werden beide Verfahren, die Termextraktion und die Extraktion der Nominalphrasen, kombiniert. Durch die Nominalphrasen werden sehr genaue Kategorien, beim Mapping mit den Artikel Titel bei Wikipedia, generiert. Da viele extrahierte Nominalphrasen nicht mit einem Titel übereinstimmen, werden die Terme mit berücksichtigt.

5.4. Kombination von Term- und Nominalphrasen-Extraktion

In der Abbildung 5.22 ist das Modell der Kombination Term- und Nominalphrasen-Extraktion aufgezeigt. Es werden erst, wie im Kapitel 5.3.2 bei der Extraktion der Nominalphrasen beschrieben, die Wortpaare auf die vorgegebenen Muster von Nominalphrasen überprüft. Die Nominalphrasen aus allen Datensätzen werden für den Abgleich der Wortpaare aus den einzelnen Datensätzen genutzt. Wie im Modell zu

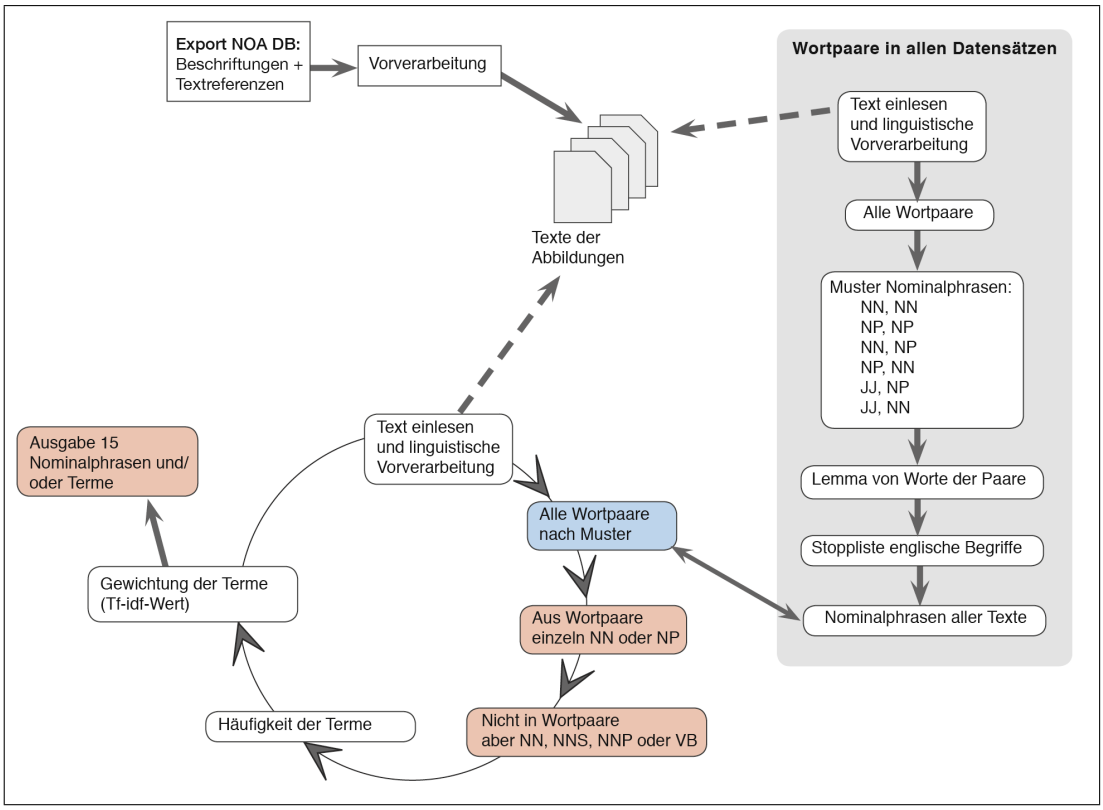


Abbildung 5.22: Modell der Kombination Term- und Nominalphrasen-Extraktion

sehen, werden die Wortpaare nicht nur auf das Vorhandensein in den gesammelten Nominalphrasen hin überprüft, es werden zusätzlich alle einzelnen Wörter dieser Wortpaare auf die PoS-Tags *NN* (Nomen) und *NP* (Eigennamen) durchsucht. Diese werden zur Ergebnismenge dazu genommen. Die Adjektive, die in dem Muster der Nominalphrasen vorkommen können, werden einzeln nicht weiter verwendet. Die Wörter, die als Wortpaar gar nicht vorkommen, werden auf die Wortformen *NN* (Nomen Singular), *NNS* (Nomen Plural), *NNP* (Eigennamen Plural) und *VB* (Verben) überprüft. Dadurch sind die einzelnen Terme, die den Datensatz gut beschreiben, in der Ergebnismenge vorhanden. Dieser Vorgang erfolgt in der Funktion *def candidates(taglist)* (siehe folgender Codeausschnitt). Der vollständige Python Code dazu befindet sich im Anhang unter B.6.

```
def candidates(taglist):
    cand = []
    skip = False
```

```
for i in range(len(taglist)-1) :
    if skip:
        skip = False
        continue
    skip = False
    l1 = taglist[i].lemma
    l2 = taglist[i+1].lemma
    if len(l1) >2 and len(l2) >2:
        if (l1,l2) in NPList:
            skip = True
            cand.append((l1,l2))
            if taglist[i].pos == 'NN' or taglist[i].pos == 'NP':
                cand.append((l1))
        else:
            if taglist[i].pos == 'NN' or taglist[i].pos == 'VB' or taglist[i].pos == '
                NNS':
                cand.append(l1)
w1 = taglist[-1]
if w1.pos == 'NN'or w1.pos == 'VB' or w1.pos == 'NNS':
    cand.append(w1)
return cand
```

Nach dem Auszählen der Häufigkeit und dem Berechnen des Tf-idf-Wertes werden für jeden Datensatz 15 Nominalphrasen und/oder Terme ausgegeben. Mit diesen extrahierten Termen und Wortpaaren wird das Mapping auf den Wikipedia Titeln durchgeführt.

Tabelle 5.8: Ausgabe von Kombination aus Nominalphrasen und Terme

NP oder Term	**Tf-idf-Wert**
10.1155:2016:2175896.txt'	
time, curve	0.82
prazosin	0.82
elimination	0.41
Plasma, concentration	0.41
prazosin, pretreatment	0.41
Plasma	0.41
elimination, half-life	0.41
maximum, concentration	0.36
time	0.34
significant, effect	0.31
concentration	0.25
area	0.20
Table	0.16
Table @card@	0.16

Die Tabelle 5.8 zeigt, für die Abbildung aus dem Artikel mit der DOI: 10.1155/2016/2175896 und der Bild-ID 0, die extrahierten Nominalphrasen und Terme auf. Hier ist zu sehen, dass *time* einmal Bestandteil der Nominalphrase *time curve* ist und einmal als eigenständiger Term vorkommt. Als Term hat *time* eine geringere Gewichtung (Tf-idf-Wert von 0.34) als in Kombination mit *curve* in der Nominalphrase (Tf-idf-Wert von 0.82).

5.5. Vergleich von Term- und Nominalphrasen-Extraktion und der Kombination

Die drei Verfahren, Termextraktion, Extraktion der Nominalphrasen und deren Kombination haben Stärken und Schwächen, die je nach den vorhandenen Rohdaten auftreten. Ist nur eine sehr kurze Bildbeschriftung und kurze Referenzen mit vielen Abkürzungen zum Abbild vorhanden, dann sind die extrahierten Terme vorteilhafter für ein Mapping zur Kategoriegewinnung. Für den Vergleich aus der Abbildung

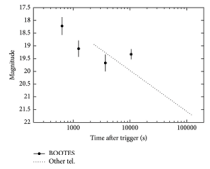
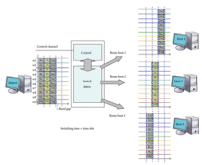
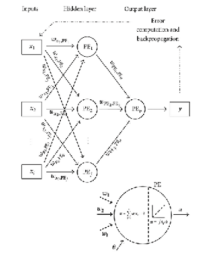
Rohtexte	Termextraktion	Extraktion Nominal-phrase	Kombination NP + Term	Bildbeschriftung	Abbildung
			Datensatz: 10.1155/2016:1928465.txt		
Wenig	curve light S telescope end hour datum transient weather focus trigger behaviour observation	optical light BOOTES observation first telescope optical transient Table @card@	optical light BOOTES BOOTES observation optical transient trigger first telescope‘ behaviour focus curve Table @card@ Table	The optical light curve of GRB 050824; the optical light curve represents the behaviour seen by Sollerman et al. [null].	
			Datensatz: 10.1155/2016:1875357.txt		
Mittel	stripe slot size wavelength time datum switch fabric system bit conversion computer control switching be	time slot different wavelength stripe size switch fabric guard band slot size electrooptical switch communication link	stripe time slot different wavelength stripe size switch fabric slot system switch time datum semisynchronous time optimum stripe electrooptical switch communication link advantage	A ten-wavelength striping system with semisynchronous time slots	
			Datensatz: 10.1155/2016:2303181.txt		
Viel	output weight variable input /math math layer network \\\\mathrm value y exemplar predictor datum contribution	input variable synaptic weight trained weight multiple processing multiple PEs1 multilayer perceptron unbiased estimation relative share	predictor synaptic weight input variable weight output function network input datum quotient relative contribution absolute contribution hidden-output output variable input-HL-output	Representations of artificial neural networks. (a) A multilayer perceptron depicting interaction/ influences among input variables (x 1 , … , i), hidden/output layers with processing elements (P E 1 , … , j and P E o), synaptic weights (w), and modeled output (y). (b) A representative network interpretation diagram for a representative simulation within Case Study. Line thickness portrays the relative magnitude of the weight.	

Abbildung 5.23: Vergleich von Menge des Rohmaterials und Extraktionsverfahren

5.23[88] werden jeweils 15 extrahierte Terme, 8 extrahierte Nominalphrasen und 15

[88] Bei diesem Vergleich sind nur die Terme und Nominalphrasen aufgenommen, die Tf-idf-Werte wurden aus Gründen der Lesbarkeit aus der Ansicht entfernt. Auch die Referenztexte sind nicht in

kombinierte Terme/Nominalphrasen miteinander verglichen. Bei dem ersten Datensatz, eingeordnet unter *Rohtext Wenig*, stehen 85 Wörter für die Extraktion zur Verfügung. Für den Datensatz mit *Rohtext Mittel* sind es 278 Wörter und bei dem letzten Datensatz, unter *Rohtext Viel*, sind es 502 Wörter. In diesem Datensatz sind in den 502 Wörter sehr viele Abkürzungen und einzelne Buchstaben und auch Auszüge aus Formeln vorhanden, die extrahierten Terme haben hier keine ausreichende Aussagekraft. Die Kombination aus Nominalphrasen und den Termen ist hier deutlich besser geeignet, die Abbildung im fachlichen Kontext zu beschreiben. Bei dem Datensatz mit der mittleren Anzahl der Rohtexte sind die Nominalphrasen aussagekräftiger, beispielsweise *different wavelength* und *semisynchronus time*. Wenn die Rohtexte der Datensätze dagegen kurz ausfallen, dann werden sowohl mit der Kombination aus Termen und Nominalphrasen, als auch nur durch die Nominalphrasen gute Begriffe zur Beschreibung der Abbildungen erreicht.

Die Extraktion der Terme und Nominalphrasen aus allen Datensätzen dient in erster Linie nicht dazu, die Abbildungen in Form einer Informationsextraktion zu beschreiben, sie werden eingesetzt, um den Abbildungen Kategorien zuzuordnen. Das Mapping für die Kategorien erfolgt auf die Titel der Wikipedia Artikel. Das Mapping für die extrahierten Termen wird im nächsten Kapitel im Kapitel 6 aufgezeigt. Im Kapitel 7 erfolgt dann das Mapping mit den Nominalphrasen und im Kapitel 8 das Mapping mit der Kombination aus beidem.

der Ansicht vorhanden, wurden aber in die Wortanzahl mit berechnet. Die vollständigen Rohtexte dazu befinden sich auf der CD, die dieser Masterarbeit beiliegt.

6. Zuordnung der Terme zu einem Wikipedia Artikel

Da die Abbildungen aus dem NOA Projekt in die Projekte der Wikimedia Foundation aufgenommen werden sollen, werden in diesem Kapitel die Vorgehensweisen und Abläufe aufgezeigt, die dazu führen, dass die Abbildungen Kategorien von Wikipedia erhalten und dadurch integriert werden können. Die Gründe für das Verwenden der Kategorien der Wikipedia werden im Kapitel 4.5 aufgezeigt.

6.1. Modell des Mappings auf die Kategorien der Wikipedia

In der Abbildung 6.24 werden die Schritte nach der Extraktion der Terme aufgezeigt. Wie schon im Kapitel 5.2.4 erwähnt, werden jeweils drei Verfahren miteinander verglichen. Die Nutzung von 5, von 10 und von 15 Termen. Der Ablauf ist identisch, aufgrund der Anzahl der Terme, werden aber unterschiedlich viele Kategorien ermittelt. Beispielhaft wird in diesem Kapitel der Vorgang für 10 extrahierte Terme aufgezeigt.

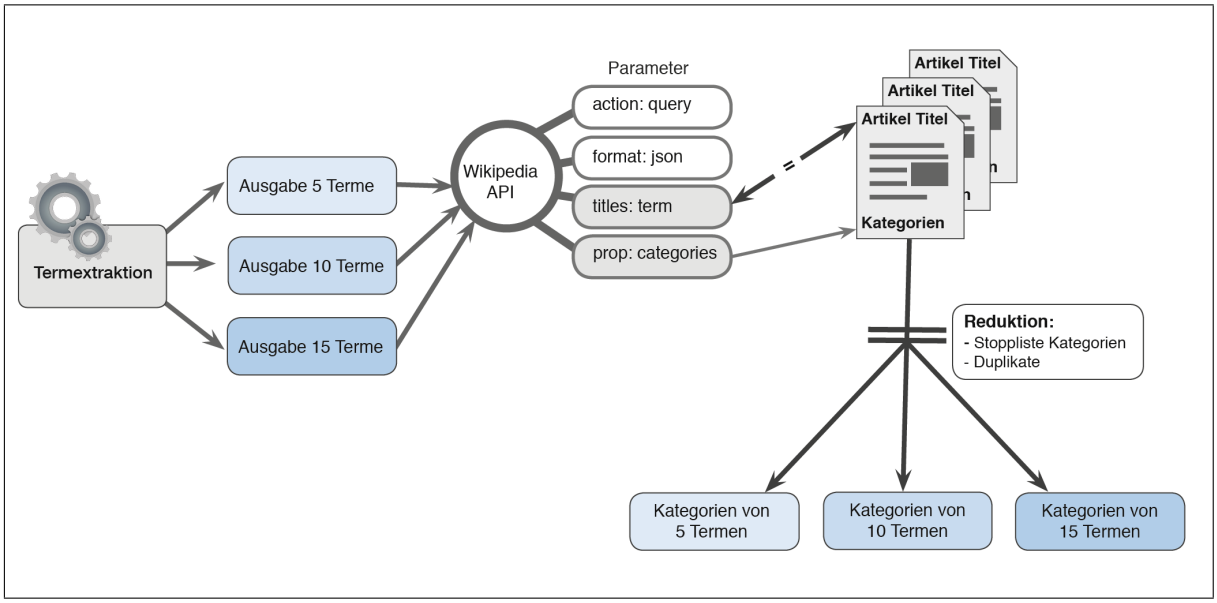


Abbildung 6.24: Modell vom Abgleich mit den Titeln der Wikipedia Artikel

Nach erfolgter Extraktion der Terme (siehe Kapitel 5.2) werden die 10 Terme mit den höchsten Tf-idf-Werten mit den Wörtern verglichen, die in den Titeln der Artikel in der Wikipedia vorhanden sind. Da dies mithilfe der Wikipedia Application Programming Interface (API) (*dt:* Schnittstelle für Anwendungsprogrammierung) geschieht, sind die Vergleichsinhalte aktuell. Stimmt der Term eines Datensatzes vollständig mit dem Titel des Wikipedia Artikels überein, dann werden die Kategorien zu diesem Artikel zurückgegeben. Diese Kategorien werden anhand einer manuell erstellten Stoppliste geprüft. Die Stoppliste enthält Kategorien wie *Category:Nothing*, *Category:Self* oder

auch *Category:Goal*.[89] Die Kategorien, die mit *Category:Disambiguation* beginnen, werden ebenfalls entfernt. Vor der Ausgabe der Kategorien werden noch die Duplikate entfernt.

6.2. Schnittstelle der Wikipedia

Mithilfe der MediaWiki Internetservice API kann aus der entwickelten Annotationsmethode heraus u.a. auf die Titel der Artikel zugegriffen werden.[90] Dabei werden mit dem Parameter *action* Informationen abgefragt. Für die Kategorienzuweisung der extrahierten Terme ist es der Parameter *action=query*. Hierbei werden Informationen abgefragt, es werden keine Inhalte zur Wikipedia hinzugefügt. Der Endpunkt, der dafür genutzt wird, ist `https://en.wikipedia.org/w/api.php`. Diese Uniform Resource Locator (URL) ist die API der englischsprachigen Wikipedia. Die Informationen über die Funktionsweise der MediaWiki API entstammt der API Dokumentation (MediaWiki 2017a).

In den nächsten Zeilen ist der Code dargestellt, der die Parameter der Abfrage aufzeigt. Der vollständige Code dazu befindet sich im Anhang unter B.2.

```
def wiki_cats(term):
    cats = []
    response = requests.get(
        'https://en.wikipedia.org/w/api.php',
        params={
            'action': 'query',
            'format': 'json',
            'titles': term,
            'prop': 'categories',
            'clshow':'!hidden',
            'cllimit':'50'
        }
    ).json()
```

Die wichtigste Aktion der MediaWiki Web Service API ist *action=query*. Diese Aktion wird genutzt, um Information aus der Wikipedia zu erhalten.[91] Das Query Modul hat drei Arten von Submodulen:

- Metainformationen (Informationen über das Wiki)
- Eigenschaften (Seiteninhalte und -informationen)
- Listen (Leistung von Artikelseiten nach Kriterien)

In der entwickelten Annotationsmethode dieser Arbeit wird das Submodul *Eigenschaften* verwendet. Mit *format=json* wird der API das gewünschte Rückgabeformat

[89] Die vollständige Stoppliste der Kategorien befindet sich auf der beiliegenden CD oder unter GitHub `https://github.com/f-josi/MA`.
[90] MediaWiki API: `https://en.wikipedia.org/w/api.php`
[91] Dokumentation der API Aktion Query: https://www.mediawiki.org/wiki/API:Query

mitgeteilt. Es wird das Format JSON[92] genutzt, weil es die Informationen strukturiert aufnehmen kann und mit Python wieder gut ausgelesen werden kann. Die einzelnen Artikelseiten werden mit dem *titles* Parameter ausgegeben. Dabei wird der Term, der mit dem Titel übereinstimme soll, der Anfragefunktion mit übergeben. Das wird folgendermaßen umgesetzt. Die Funktion *def wiki_cats(term)* wird mit dem Parameter *term*, für die extrahieren Termen aus dem Datensatz, aufgerufen. Der Wert für den Parameter *term* wird beim Funktionsaufruf mit übergeben. Die Terme werden in den Titeln der Artikelseiten gesucht, um diese auszuwählen. Aufgrund der Begrenzung der MediaWiki API auf 50 Titel pro Aufruf, benötigt die Abfrage etwas Zeit. Die Aktion *query* hat u.a. die Eigenschaft *categories*. Diese Eigenschaft soll für die abgefragten Artikelseiten, die eine Übereinstimmung von übermitteltem Term und Titel haben, abgerufen werden. Die Eigenschaft *prop:categories* listet alle Kategorien auf, in die der Artikel eingeordnet wurde. Diese Kategorien sind der Grund der ganzen Abfrage. Sie werden den Datensätzen der Abbildung aus NOA zugewiesen und stellen eine erste grobe Kategorisierung dar.

Zu diesem Zeitpunkt befinden sich aber noch viele wenig aussagekräftige Kategorien wie beispielsweise *Category:Systems* darunter. Eine erste Reduktion dieser Kategorien erfolgt direkt bei der Abfrage. Dabei werden Kategorien gelöscht, die Metainformationen beinhalten z.B. *Category:ISO basic Latin letters*, oder zu allgemein sind wie *Category:Goal*. Einige Kategorien aus diesen Bereichen wurden in eine manuelle Stoppliste für unerwünschte Kategorien aufgenommen (siehe Kapitel 6.1), diese Kategorien werden nicht zur Ergebnismenge hinzugefügt. Zusätzlich werden alle Kategorien, die mit *Category:Disambiguation* beginnen, nicht weiter berücksichtigt. Sie werden für die Wikipedia Artikel häufig vergeben, tragen aber nicht zum gewünschten Zweck, der Annotation der Abbildungen in NOA, bei. Mit der Mengenfunktion *set()* in der Rückgabeanweisung werden Duplikate in der Ergebnismenge der Kategorien entfernt. Die Eigenschaft *categories* hat u.a. die Parameter *clshow* (Art der Kategorie versteckte oder sichtbare) und *cllimit* (Anzahl der zurückgegebenen Kategorien). Verwendet werden hier nur die sichtbaren Kategorien einer Artikelseite und die Anzahl der Kategorien ist beschränkt auf 50 pro Term.

Aufruf der MediaWiki API:

```
def collect_nouns(flist):
    global noun2cat
    for f in flist:
        nouns = extract_term(f, nr_of_docs)
        for (n,f) in nouns:
            noun2cat[n] = wiki_cats(n)

collect_nouns(filelist[150:170])
```

[92] JavaScript Object Notation

Der Funktion *collect_nouns* wird beim Aufruf das Argument *filelist[150:170]* mit übergeben.[93] In dem Dictionary *noun2cat* werden alle Kategorien zu einem Datensatz gespeichert. Für jeden Term in der Variablen *nouns* wird die Funktion *wiki_cats()* mit dem Argument *n* aufgerufen. Dabei wird in der for-Schleife über die Elemente von *nouns* iteriert. In *nouns* sind wiederum die extrahierten Terme enthalten, die als Rückgabewert aus der Funktion *extract_term(f, nr_of_docs)* übergeben wurden. Die Funktion *extract_term(f, nr_of_docs)* wird mit den Argumenten f für jeden Datensatz aus der Dateiliste und der Anzahl der Dateien der Dateiliste aufgerufen.

Ausgabe der Kategorien vom Termmapping:

```
for f in filelist[150:170]:
    terme = extract_term(f,nr_of_docs)
    cats = set([c for (kw,f) in terme for c in noun2cat.get(kw)])
    pprint.pprint(f)
    pprint.pprint(cats)
```

Für die Datensätze von 150 bis 170 werden die Terme extrahiert. Die Kategorien werden für jeden extrahierten Term, der in *terme* und als Schlüssel im Dictionary *noun2cat* vorkommt, zurückgegeben. Mit der *get()* Methode wird für jeden gegebenen Schlüssel der Wert zurückgegeben. Danach erfolgt die Ausgabe des Dateinamens des Datensatzes und die Ausgabe der Kategorien.

Tabelle 6.9: Ermittelte Kategorien für zwei Terme

pyridine:
Category:Amine solvents, Category:Aromatic bases, Category:Aromatic solvents, Category:Foul-smelling chemicals, Category:Functional groups, Category:Pyridines, Category:Simple aromatic rings
prazosin:
Category:Alpha-1 blockers, Category:Antihypertensive agents, Category:Anxiolytics, Category:Carboxamides, Category:Furans, Category:Phenol ethers, Category:Piperazines, Category:Quinazolines, Category:Vasodilators

In der Tabelle 6.9 sind beispielhaft die Kategorien für die Terme *pyridine* und *prazosin* aufgezeigt. Diese Terme wurden aus den Datensätzen 10.1155:2016:2380540.txt

[93] Für die Zuordnung zu den Kategorien werden beispielhaft 20 von den 397 Datensätzen angezeigt.

und 10.1155:2016:2175896.txt extrahiert.[94]

(a) Wikipedia Artikelseite für den extrahierten Term *pyridine*

(b) Wikipedia Artikelseite für den extrahierten Term *prazosin*

Abbildung 6.25: Anzeige der Kategorien in den Artikelseiten der Wikipedia

Parallel dazu befinden sich in der Abbildung 6.25 die Wikipedia Artikelseiten der jeweiligen Terme, inkl. den dazugehörigen Kategorien. Für die weitere Vorgehensweise werden aber nur noch die Kategorien zu den einzelnen Datensätzen ausgegeben, nicht mehr die dazugehörigen Terme.

Argumentation für die gewählte Vorgehensweise:
Der Gedanke hinter dieser Lösungsstrategie ist, wenn ein extrahierter Term aus der Bildunterschrift einer Abbildung oder aus der Textreferenz Teil des Titels eines Wikipedia Artikels ist, dann ist die Wahrscheinlichkeit hoch, dass es im Artikel um die gleiche Thematik geht und die Abbildung mit den Kategorien des Artikels passend annotiert ist.

[94] Die Datensätze haben, je nach angegebener Termenanzahl für den Rückgabewert, mehrere extrahierte Terme. Für einen ersten Überblick wurde beispielhaft je einer ausgewählt.

6.3. Mapping der Terme auf Wikipedia Kategorien

Für das Mapping werden, wie im Modell im Kapitel 6.1 dargestellt, drei Verfahren durchgeführt und miteinander verglichen. In der Abbildung 6.26 sind die Kategorien aufgezeigt, aus denen 5, 10 und 15 extrahierten Termen ermittelt wurden. Wie in

Kategorien von 5 Termen pro Datensatz	Kategorien von 10 Termen pro Datensatz	Kategorien von 15 Termen pro Datensatz
10.1155:2016:1875357.txt	10.1155:2016:1875357.txt	10.1155:2016:1875357.txt
Category:Electrical components,	Category:Binary arithmetic,	Category:Binary arithmetic,
Category:Human–machine interaction,	Category:Concepts in metaphysics,	Category:Concepts in metaphysics,
Category:Size,	Category:Concepts in physics,	Category:Concepts in physics,
Category:Switches	Category:Data types,	Category:Contract law,
	Category:Electrical components,	Category:Data types,
	Category:Human–machine interaction,	**Category:Electrical components,**
	Category:Physical quantities,	**Category:Human–machine interaction,**
	Category:Primitive types,	Category:Legal doctrines and principles,
	Category:SI base quantities,	Category:Physical quantities,
	Category:Size,	Category:Primitive types,
	Category:Switches,	Category:SI base quantities,
	Category:Units of information,	**Category:Size,**
	Category:Waves	**Category:Switches,**
		Category:Systems,
		Category:Units of information,
		Category:Waves
10.1155:2016:1901493.txt	10.1155:2016:1901493.txt	10.1155:2016:1901493.txt
Category:Balls	**Category:Balls,**	**Category:Balls,**
	Category:Lubricants,	Category:Classical mechanics,
	Category:Lubrication,	Category:Combustion,
	Category:Mechanical engineering,	Category:Force,
	Category:Petroleum products,	Category:Friction,
	Category:Tribology	Category:Lubricants,
		Category:Lubrication,
		Category:Mechanical engineering,
		Category:Petroleum products,
		Category:Tribology
10.1155:2016:1925827.txt	10.1155:2016:1925827.txt	10.1155:2016:1925827.txt
Category:Elementary shapes,	**Category:Elementary shapes,**	**Category:Elementary shapes,**
Category:Quadrilaterals	Category:Geometry,	Category:Geometry,
	Category:Quadrilaterals,	**Category:Quadrilaterals,**
	Category:Space,	Category:Space,
	Category:Topology	Category:Topology

Abbildung 6.26: Ermittelte Kategorien von 5, 10 und 15 Termen im Vergleich

der Gegenüberstellung zu sehen ist, werden bei der Verwendung von 5 Termen sehr wenige Kategorien ermittelt und bei dem Abgleich mit den 15 Termen sind es viele Kategorien,[95] mit z.T. aussageschwachen Kategorien wie *Category:Systems* oder *Category:Force*. Der vollständige Code des Mappings mit 15 Termen befindet sich im Anhang unter B.2. Für die Optimierung der Kategorie-Ergebnismenge werden die Beziehungen der Kategorien ausgezählt und sortiert.

[95] Bei den verwendeten drei Datensätzen wurden weniger Kategorien ermittelt, als im Beispiel aus Kapitel 6.2 mit dem Datensatz *10.1155:2016:2175896.txt*, wo für den einen Term *prazosin* alleine neun Kategorien erhalten wurden.

6.4. Ranking der Kategorien aus den Termen

Für die Berechnung der Beziehungen zwischen den Kategorien werden alle ermittelten Kategorien genutzt. In der Abbildung 6.27 ist eine Übersicht über die Vorgehensweise des Rankings dargestellt. Nach der Extraktion der Terme und dem Mapping

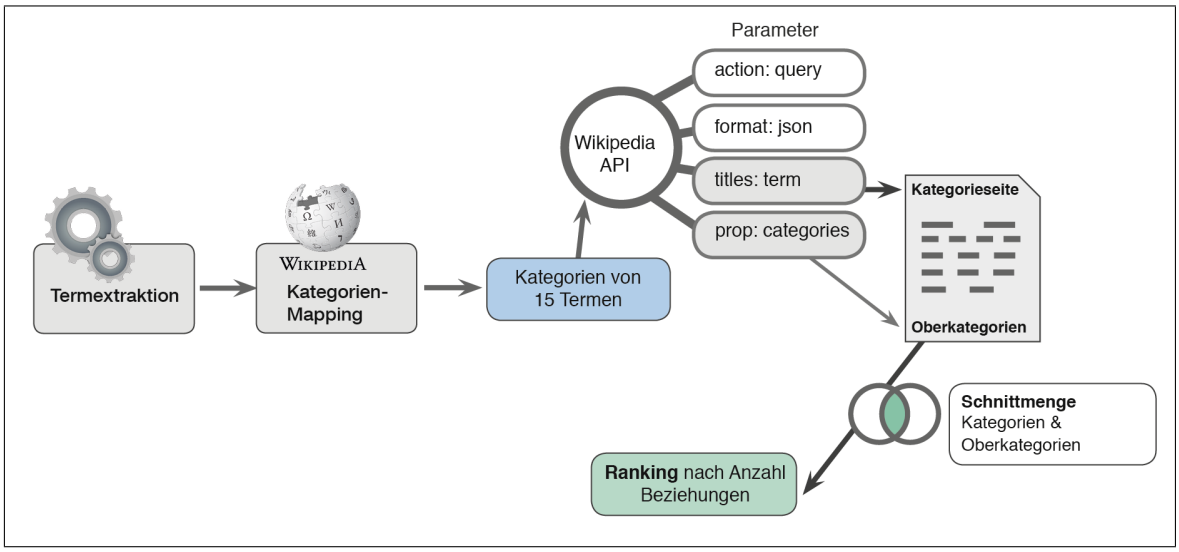


Abbildung 6.27: Modell vom Ranking der Kategorien der Terme

auf die Titel der Artikel, werden die einzelnen Kategorien erneut über die Media Wiki API aufgerufen. Dazu werden die Kategorien im Titel gesucht, d.h. statt der extrahierten Terme werden die daraus schon zugeordneten Kategorien in den Titeln gesucht. Eine Kategorieseite beginnt bei Wikipedia mit dem Präfix Category:, beispielsweise *Category:Mechanical engineering*. Auf jeder Kategorieseite sind am Ende der Seite die Oberkategorien aufgelistet. Für die eben erwähnte Kategorieseite sind das die Oberkategorien *Applied and interdisciplinary physics Engineering disciplines* und *Manufacturing*. Die Kategorie, die für die Abfrage übergeben wurde und die Oberkategorien werden gemeinsam mit den Kategorien der anderen extrahierten Terme verglichen. Dieser Vergleich bezieht sich immer auf die Kategorien eines Datensatzes. In der Abbildung 6.28 sind die Kategorien aus dem Datensatz *10.1155:2016:1901493.txt* und ihre Beziehung aufgezeigt. Die Kategorien *Tribology*, *Friction* und *Force* haben jeweils zwei Beziehungen zu anderen Kategorien. Die Kategorien *Lubricants*, *Lubrication*, *Classical mechanics* und *Mechanical engineering* haben jeweils eine Beziehung zu einer anderen Kategorie. Die anderen Kategorien haben keine Übereinstimmung. Die Kategorien werden nach ihrer Anzahl von Übereinstimmungen sortiert ausgegeben. Durch diese Vorgehensweise können aussagekräftigere Kategorien zur Annotation der Abbildungen genutzt werden. Es kann davon ausgegangen werden, dass die fachliche Nähe der Kategorien steigt, wenn sie mehrere gemeinsame Oberkategorien haben (Gazendam u. a. 2009). Dadurch können die Kategorien der Fachgebiete, die hierarchisch

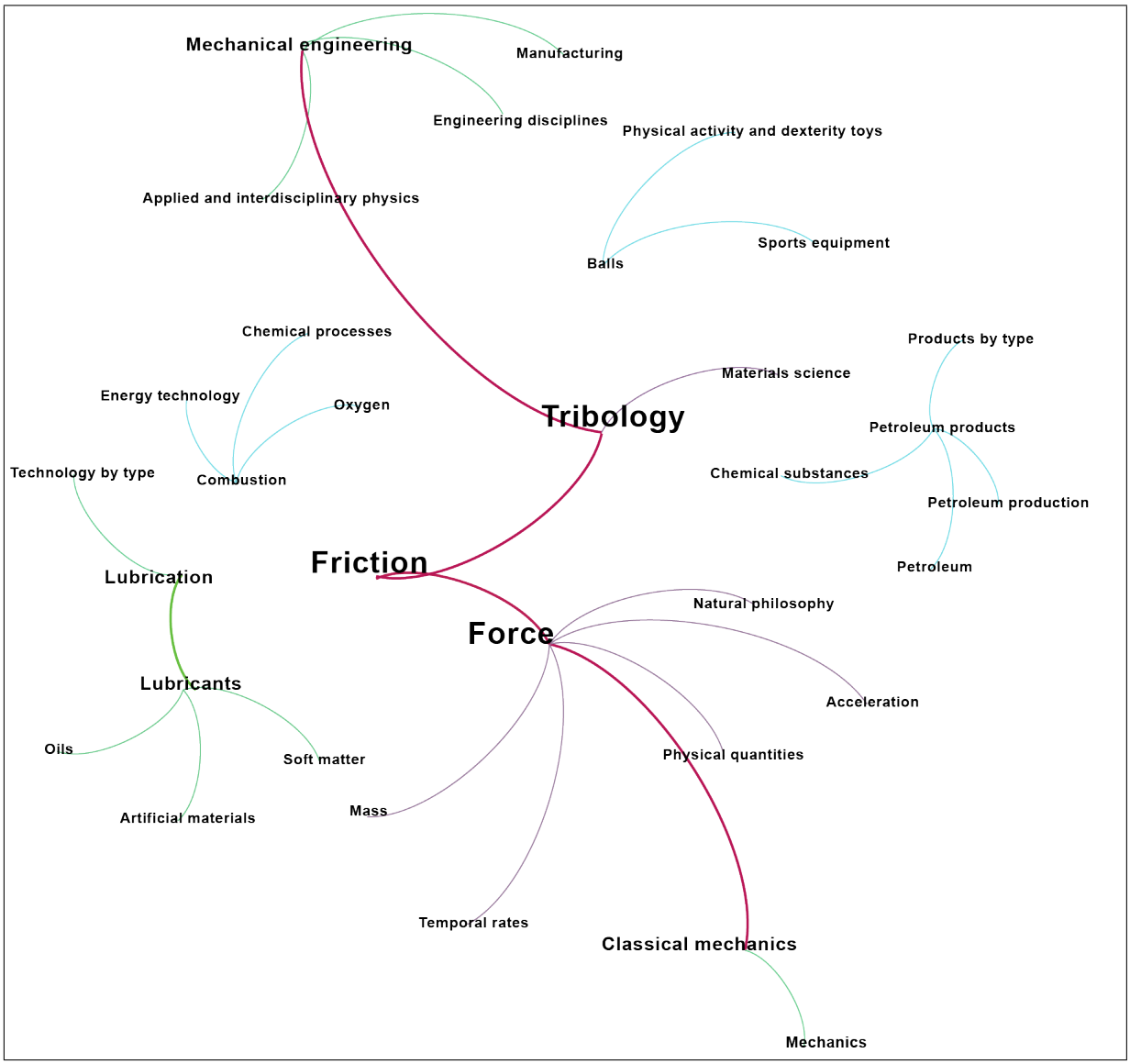


Abbildung 6.28: Ranking der Kategorien anhand der Anzahl der Beziehungen

klassifiziert sind, effizienter zur Gewichtung der Kategorien genutzt werden. Eine weitere Motivation ist, dass Kategorien die von einem Term erhalten werden niedriger gewichtet werden können, als Kategorien, die für viele Terme gefunden wurden. Diese Kategorien können die Abbildung detaillierter klassifizieren. Die Kategorien der Facettenklassifikation werden bei dieser Vorgehensweise nur bedingt berücksichtigt, da sie nicht zwingend die gleichen Oberkategorien haben.

Nach dem Ranking anhand der Anzahl der Übereinstimmungen, können die Abbildungen aus NOA mit fachlich zutreffenden Kategorien versehen werden. Die Tabelle 6.10 zeigt den gleichen Datensatz wie in der Abbildung 6.28, das Ranking erfolgt dabei nach der Anzahl der Übereinstimmungen in den Oberkategorien.

Tabelle 6.10: Ranking der Kategorien für den Datensatz aus Abbildung 6.28

10.1155:2016:1901493.txt
Category:Tribology, 2
Category:Friction, 2
Category:Force, 2
Category:Classical mechanics, 1
Category:Lubrication, 1
Category:Lubricants, 1
Category:Mechanical engineering, 1
Category:Petroleum products, 0
Category:Combustion, 0
Category:Balls, 0

6.5. Detaillierte Umsetzung des Kategorien Rankings

Die folgende Beschreibung der Methode beginnt nach dem Mapping der extrahierten Terme auf die Kategorien der Artikelseiten bei Wikipedia. Um eine größere Anzahl von Kategorienüberschneidungen zu bekommen, werden die Kategorien von den 15 Termen übernommen. Die Funktion *wiki_cats(cat)* wird aus der Funktion *wiki_cats_sort_rel()* mit dem Argument *cat* aufgerufen. Der Parameter *term*, mit der übergebenen Kategorie, wird dann in dem Titel der Kategorienseite in der Wikipedia gesucht. Die, auf der Kategorieseite, vorhandenen Oberkategorien werden als Liste *cats* zurückgegeben.

```
def wiki_cats(term):
    cats = []
    response = requests.get(
        'https://en.wikipedia.org/w/api.php',
        params={
            'action': 'query',
            'format': 'json',{Detaillierte Umsetzung des Kategorien Rankings}
            'titles': term,
            'prop': 'categories',
            'clshow':'!hidden',
            'cllimit':'50' }
    ).json()

    for pageid in  response['query']['pages']:
        cats.extend([cat['title'] for cat in response['query']['pages'][pageid]['categories
            '] if cat['ns'] == 14])
    return cats
```

Der Vorgang bis zu diesem Punkt ist auch in der Abbildung 6.28 zu sehen. Es werden alle Oberkategorien zurückgegeben, es findet keine Reduzierung statt. [96]
Die Funktion *wiki_cats_sort_rel(f)* wird für jeden Datensatz mit dem Argument *f* (einzelne Datei) aufgerufen.[97] Nach dem Extrahieren der Terme für jeden Datensatz

[96] Anders als bei der Ermittlung der Kategorien der Artikelseiten wird an dieser Stelle keine Stoppliste von Kategorien eingesetzt.
[97] Hier im Code sind es die Dateien der Datensätze 150 bis 170.

(Funktionsaufruf *extract_term(f,nr_of_docs)*) und dem Mapping auf die Kategorien der Artikel von Wikipedia, wird die Funktion für das Ermitteln der Oberkategorien aufgerufen *family.extend(wiki_cats(cat))*. Alle Oberkategorien (*extended[cat] = family*) werden mit den Kategorien der Artikelkategorien auf Gleichheit überprüft (siehe zweite und dritte for-Schleife).

```
def intersection(a,b):
    i = 0
    for e in a:
        if e in b:
            i += 1
    return i

def wiki_cats_sort_rel(f):
    terme = extract_term(f,nr_of_docs)
    cats = set([c for (kw,f) in terme for c in noun2cat.get(kw)])
    extended = {}
    relnumber = {}
    for cat in cats:
        family = [cat]
        family.extend(wiki_cats(cat))
        extended[cat] = family
    for cat in cats:
        relsize = 0
        for cat2 in cats:
            if cat == cat2:
                continue
            if intersection(extended[cat],extended[cat2]) > 0:
                relsize +=1
        relnumber[cat] = relsize
    return sorted(relnumber.items(),key = lambda x:x[1],reverse=True)
```

Wenn die Kategorien in *cat* und *cat2* gleich sind, wird die Beziehungsangabe (*relsize*) erhöht. Dafür wird die Funktion *intersection(a,b)* mit den Argumenten *extended[cat]* und *extended[cat2]* aufgerufen. Aus den Schlüssel-Wert Paaren in *relnumber* wird mithilfe der anonymen Lambda-Funktion auf die Anzahl der Beziehungen zugegriffen und danach sortiert. Im letzten Abschnitt erfolgt der Aufruf der Funktion für das Ermitteln der Oberkategorien *wiki_cats_sort_rel(f)*. Dadurch ergibt sich für den Datensatz *10.1155:2016:1901493.txt* ein Ranking der Kategorien, wie in der Tabelle 6.11 aufgezeigt. Der vollständige Python Code zum Ranking der Kategorien befindet sich im Anhang unter B.3.

Tabelle 6.11: Ranking der Kategorien für einen Datensatz

Kategorie	**Anzahl**
Category:Friction	2
Category:Force	2
Category:Tribology	2
Category:Mechanical engineering	1
Category:Lubricants	1
Category:Classical mechanics	1
Category:Lubrication	1
Category:Combustion	0
Category:Balls	0
Category:Petroleum products	0

6.6. Abschlussbetrachtung Kategorien der Termextraktion

In der Abbildung 6.29 sind die verschiedenen Verfahren dargestellt, die mit den extrahierten Termen durchgeführt wurden. Die Vorgehensweise mit nur 5 extrahierten Termen ist für Abbildungen mit kurzen Beschriftungen anwendbar, durch die Hinzunahme der Referenzen im Text können aber mehrere passende Terme für jede Abbildung gefunden werden, die eine genauere Einordnung in die Kategoriesystematik von Wikipedia ermöglichen. Für die Gewichtung der Kategorien ist es nützlich, viele Kategorien zur Verfügung zu haben. Dadurch können Kategorien, die eine höhere Schnittmenge mit anderen aufweisen, höher gewichtet werden.
Für die Evaluierung der entwickelten Methoden werden im Kapitel 9 die Verfahren mit den 5, 10 und mit den 15 extrahierten Termen eingesetzt. Die Überschneidungen in den Kategorien werden für das Ranking der Kategorien eingesetzt.

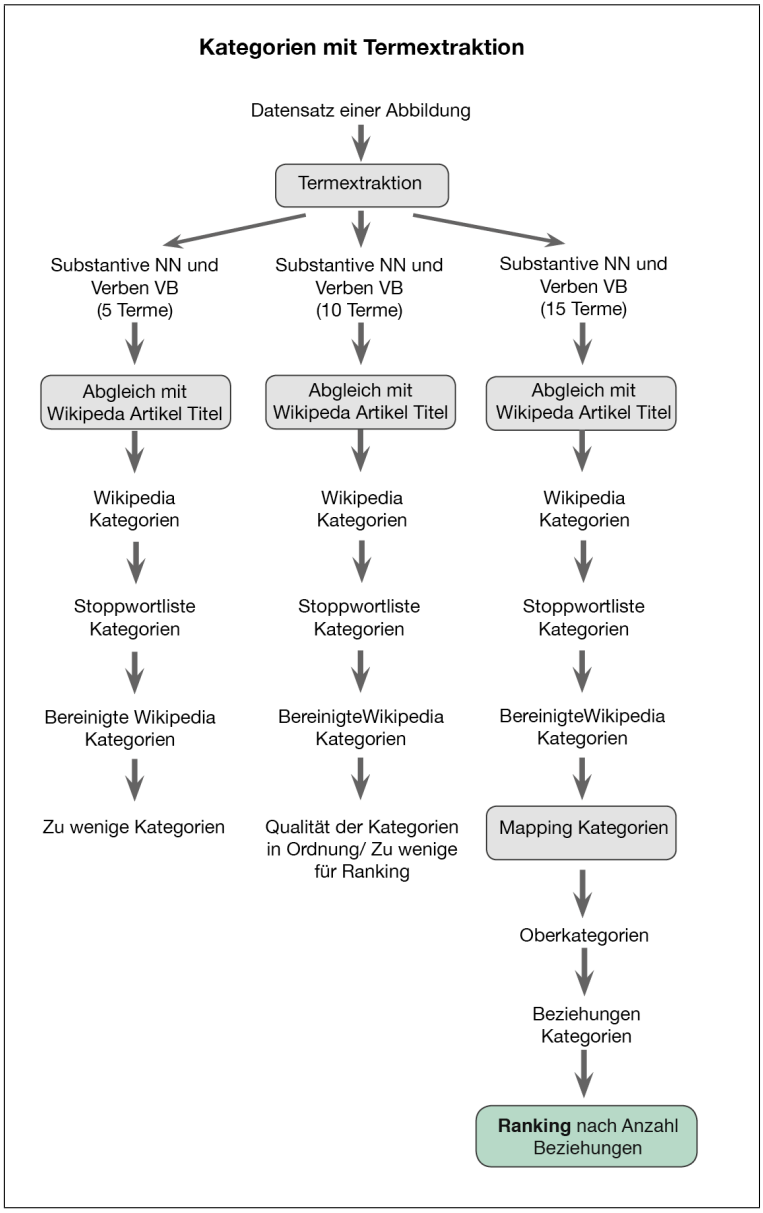


Abbildung 6.29: Durchgeführte Varianten für Kategorien Mapping durch extrahierte Terme

7. Mapping von Nominalphrasen zu Wikipedia Kategorien

Das Mapping der Nominalphrasen zu den Kategorien der Wikipedia wird in der Abbildung 7.30 aufgezeigt. Wie schon bei dem Mapping der Terme in Kapitel 6.3, werden hierfür die Terme aus der Extraktion genutzt. In diesem Fall die Terme, die die Nominalphrasen beinhalten. Das Mapping wurde mit 8 Nominalphrasen pro Datensatz durchgeführt, siehe auch Kapitel 5.5. Es werden Nominalphrasen mit einem hohen Tf-idf-Wert eingesetzt. Die Nominalphrasen werden für die weitere Verwendung bereinigt, d.h. die vorkommenden Zeichen der Tokenisierung werden entfernt. In der Tabelle 7.12 sind die Nominalphrasen vor der Bereinigung und nach der Entfernung der, für das Wikipedia-Mapping, wichtigen Zeichen zu sehen. Zusätzlich werden beide Wörter der Nominalphrase in Kleinbuchstaben umgewandelt. Das Mapping ist nur erfolgreich, wenn das zweite Wort mit einem Kleinbuchstaben beginnt. Die Wörter der Nominalphrase, die aus weniger als zwei Buchstaben bestehen, werden nicht weiterverwendet. Beim Mapping werden die Titel der Wikipedia Artikel auf die

Tabelle 7.12: Nominalphrasen vor und nach ihrer Bereinigung

Nomialphrasen	**Bereinigte Nomialphrasen**
('successful', 'utilization')	successful utilization
('Seribu', 'Island')	seribu island
('Instrumentation', 'Laboratory')	instrumentation laboratory
('Research', 'location')	research location
('Bogor', 'Agricultural')	bogor agricultural
('Ocean', 'Acoustics')	ocean acoustics
('Marine', 'Science')	marine science

extrahierten und bereinigten Nominalphrasen aus den Datensätzen durchsucht. Die Durchführung des Mappings wird detailliert im Kapitel 6.1 beim Mapping der Terme zu den Kategorien beschrieben. Im Unterschied zu den Termen, werden die Titel hier auf Nominalphrasen hin abgeglichen. Die Nominalphrase kommt dabei in genau der gleichen Reihenfolge im Titel vor. Für das Mapping wird die MediaWiki Internetservice API `https://en.wikipedia.org/w/api.php` verwendet. Die Funktionsweise und die Verwendung der API bei dem Mapping wird im Kapitel 6.2 aufgezeigt. Bei den Parametern der API-Abfrage werden, im Gegensatz zu dem Mapping mit den extrahierten Termen, mit dem Parameter *titles* die bereinigte Nominalphrase beim Funktionsaufruf mit übergeben, siehe dazu den folgenden Code unter *'titles': term_clear*.

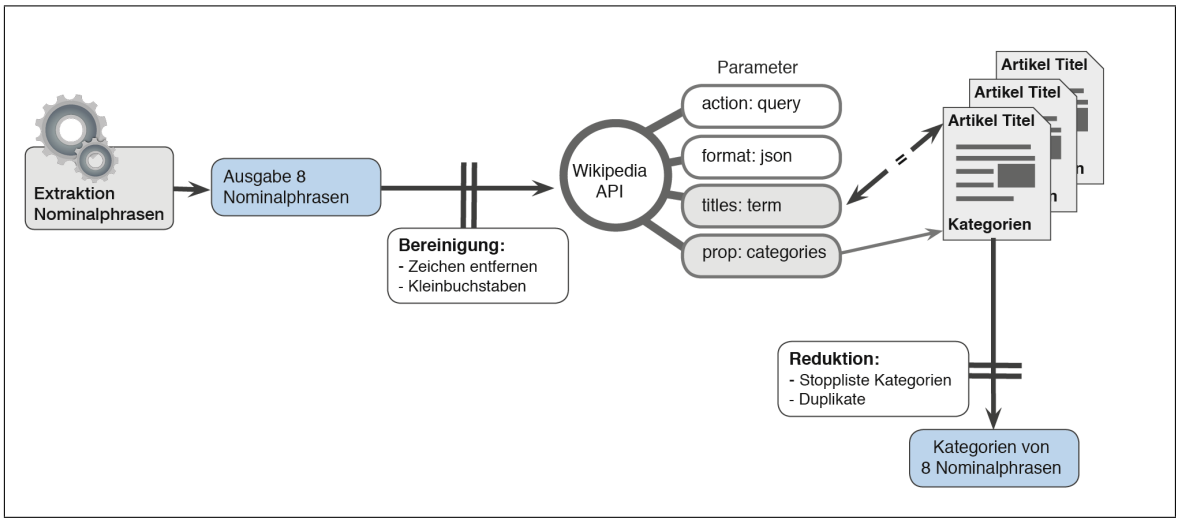


Abbildung 7.30: Modell Mapping der Nominalphrasen auf Wikipedia Kategorien

```
def wiki_cats(term):
    cats = []
    term_string = str(term)
    term_lower = term_string.lower()
    rechar = re.compile(r"['(),()]")
    term_clear = rechar.sub("", term_lower)

    response = requests.get(
        'https://en.wikipedia.org/w/api.php',
        params={
            'action': 'query',
            'format': 'json',
            'titles': term_clear,
            'prop': 'categories',
            'clshow':'!hidden',
            'cllimit':'50' }
    ).json()

    for pageid in response['query']['pages']:
        if 'categories' in response['query']['pages'][pageid]:
            cats.extend([cat['title'] for cat in response['query']['pages'][pageid]['
                categories'] if cat['ns'] == 14 and not cat['title'].startswith('Category:
                Disambiguation') and not cat['title'].startswith('Category:Wikipedia
                articles incorporating') and cat['title'] not in catStoplist])
        return set(cats)
```

Eine Besonderheit bei dem Mapping der Nominalphrasen ist, dass das zweite Wort der Nominalphrasen mit einem Kleinbuchstaben beginnen muss. Da die Nominalphrase als vollständiger String mit einem Titel der Wikipedia übereinstimmen muss, führen auch die Klammern, die einfachen Anführungszeichen und die Kommata zu keiner Übereinstimmung und müssen entfernt werden.

Dies wird mit einem regulären Ausdruck[98] durchgeführt, siehe folgenden Ausschnitt aus der Funktion *def wiki_cats(term)*:

```
term_string = str(term)  #Strings
term_lower = term_string.lower() #Kleinbuchstaben
rechar = re.compile(r"[',()]") #Ersetzen der Zeichen
term_clear = rechar.sub("", term_lower)
```

Die genaue Vorgehensweise des Mappings wird anhand eines Beispiels aus der Abbildung 7.31 und der Tabelle 7.13 verdeutlicht. Der vollständige Python Code zum Mapping mit den Nominalphrasen befindet sich im Anhang unter B.5.
Abbildung: DOI=10.1155/2016/2385429 ID=0
Datensatz: 10.1155/2016/2385429.txt
Die Abbildung 7.31 zeigt das Bild, das zum Datensatz gehört, aus dem die Nominal-

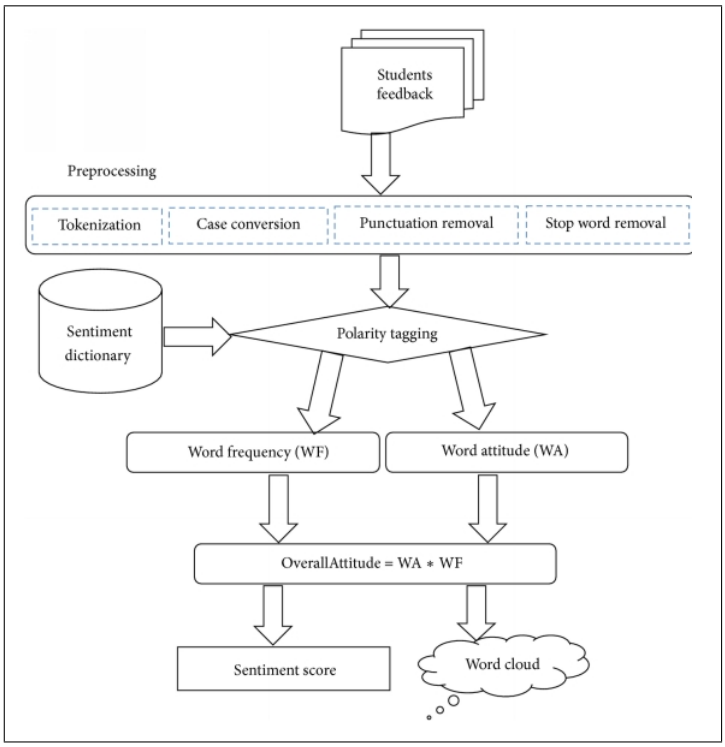


Abbildung 7.31: Abbildung aus der die Nominalphrase *sentiment analysis* extrahiert wurde

phrase *sentiment analysis* extrahiert wurde. Die Nominalphrase muss eine vollständige Übereinstimmung mit einem Titel eines Wikipedia Artikels haben. Die Kategorien, die diesem Artikel zugeordnet wurden, werden zur Kategorisierung der Abbildung 7.31 genutzt. Die Tabelle 7.13 zeigt die extrahierte Nominalphrase, den übereinstimmenden Titel eines Wikipedia Artikels und die dazugehörigen Kategorien. Die einzelnen bereinigten Nominalphrasen werden beim Aufruf der MediaWiki API übergeben und mit den Titeln der Artikel überprüft. Im Fall einer Übereinstimmung werden

[98] In Python werden reguläre Ausdrücke verwendet, um Strings auf das Vorkommen eines anderen String hin zu untersuchen (Klein 2014, S. 319). Die Entwicklung der regulären Ausdrücke geht auf den Mathematiker Stephen Cole Kleene um 1950 zurück.

Tabelle 7.13: Beispielabbildung für das Kategorienmapping mit extrahierten Nominalphrasen

Nominalphrase: sentiment analysis
Wikipedia Artikel: Sentiment analysis
Kategorien: Category:Affective computing, Category:Natural language processing, Category:Polling, Category:Social media

die Kategorien des Artikels an die Funktion *def wiki_cats(term):* zurückgegeben, siehe dazu das Modell in der Abbildung 7.30. Die Kategorien werden, wie auch bei dem Mapping mit den extrahierten Termen, mithilfe einer Stoppliste reduziert. Die Dubletten werden entfernt, indem die Kategorien als Menge mit dem Datentyp *set()*, erfasst werden. In einer Menge werden Elemente ungeordnet und jeweils nur einmal aufgenommen (Klein 2014, S. 53). Die Kategorien, die jetzt ausgegeben werden, sind die Kategorien der Wikipedia Artikel, die eine vollständige Übereinstimmung mit einem Titel eines Artikels haben, siehe dazu Tabelle 7.13 mit der Nominalphrase *sentiment analysis*.

7.1. Ranking der Kategorien aus den Nominalphrasen

Die Kategorien, die vom Mapping mit den Nominalphrasen generiert wurden, werden erneut an die MediaWiki API übergeben, um die Oberkategorien zu ermitteln. Wie auch im Kapitel 6.5 bei dem Ranking der Kategorien aus der Termextraktion, werden dazu die einzelnen Kategorieseiten genutzt. Der vollständige Python Code dazu befindet sich im Anhang unter B.3.

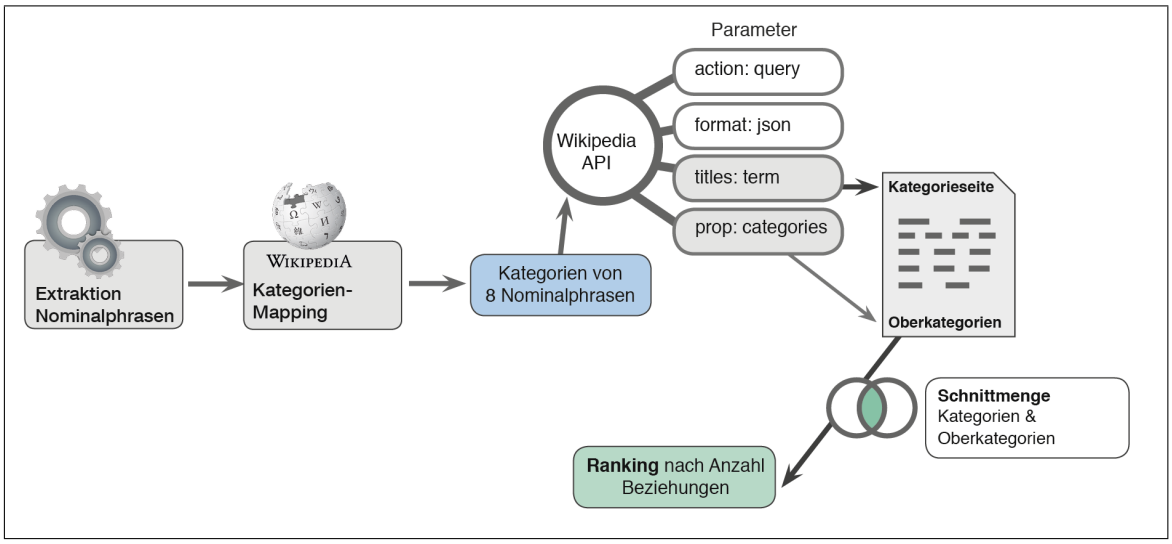


Abbildung 7.32: Modell vom Ranking der Nominalphrasen Kategorien

Die Vorgehensweise für das Ermitteln der Beziehungen der Kategorien, wird anhand der Abbildung 7.33 mit der DOI *10.1155/2016/2653915* und der Bild-ID *0* mit dem Dateinamen *10.1155:2016:2653915.txt*, vorgestellt.[99] Der Vorgang, für die Ge-

Abbildung 7.33: Abbildung des Beispieldatensatzes

wichtung der Kategorien, wird anhand der Nominalphrase *gait analysis* im Modell in der Abbildung 7.34 aufgezeigt. Die Nominalphrase *gait analysis* wird als String in den Artikel Titeln gesucht. Die Kategorien, die diesem Artikel zugeordnet wurden, sind *Terrestrial locomotion, Orthopedic surgical procedures, Rehabilitation medicine, Neurology procedures, Biometrics, Forensic disciplines und Forensic techniques*. Zusätzlich werden auch alle anderen Nominalphrasen des Datensatzes mit Titeln abgeglichen. Nach dem Überprüfen durch die Stoppliste und der Reduzierung der Duplikate, werden

[99] Im Beispiel, aus dem vorangegangenem Kapitel 7, haben die Kategorien keine Beziehungen untereinander. Da die Artikel bei Wikipedia auch nach Facetten kategorisiert werden können, ist das nicht zwingend ein Kriterium dafür, dass die Kategorien fachlich keine Überschneidungen haben (siehe Kapitel 4.1 Facettenklassifizierung).

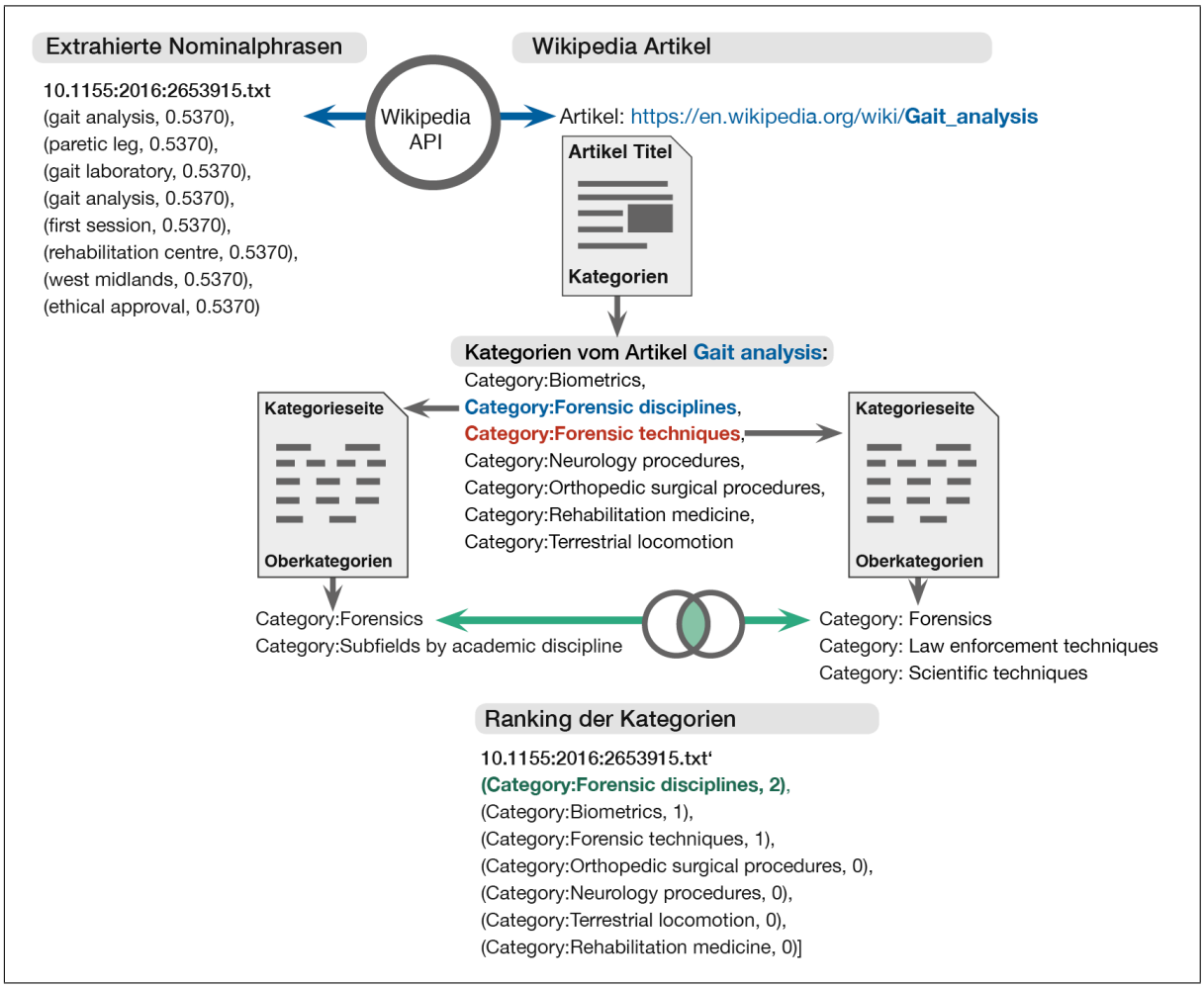


Abbildung 7.34: Verwendung der Beziehungen der Kategorien für Gewichtung

diese Kategorien für die Berechnung der Gewichtung genutzt. Dafür werden die Kategorien selber als Term für die Titel-Abgleichung übergeben.

Die Seiten, die dann gefunden werden, sind Kategorieseiten, siehe dazu in der Abbildung 7.34 die Kategorie *Category:Forensic disciplines*. Die Kategorie *Category:Forensic disciplines* wurde den Oberkategorien *Category:Forensics* und *Category:Subfields by academic discipline* zugeordnet. Die Kategorie *Category:Forensic techniques* hat auch die Oberkategorie *Category:Forensics*. Somit besteht für die Kategorien *Category:Forensic disciplines* und *Category:Forensic techniques* eine Übereinstimmung und eine gemeinsame Schnittmenge in den Oberkategorien. Da die Kategorie *Category:Forensic disciplines* auch eine Oberkategorie von der Kategorie *Category:Biometrics* ist, hat sie zwei Übereinstimmungen.

In diesem Beispiel haben die anderen Kategorien weniger Übereinstimmungen, deshalb beginnt das Ranking der Kategorien mit *Category:Forensic disciplines*, danach mit *Category:Forensic techniques* und *Category:Biometrics*, weil hier jeweils eine Übereinstimmung vorliegt.

7.2. Abschlussbetrachtung Kategorien aus Nominalphrasen Extraktion

In der Abbildung 7.35 ist das Verfahren zum Mapping mit den Kategorien der Wikipedia mithilfe der Nominalphrasen dargestellt. Dafür wurden für jeden Datensatz 8

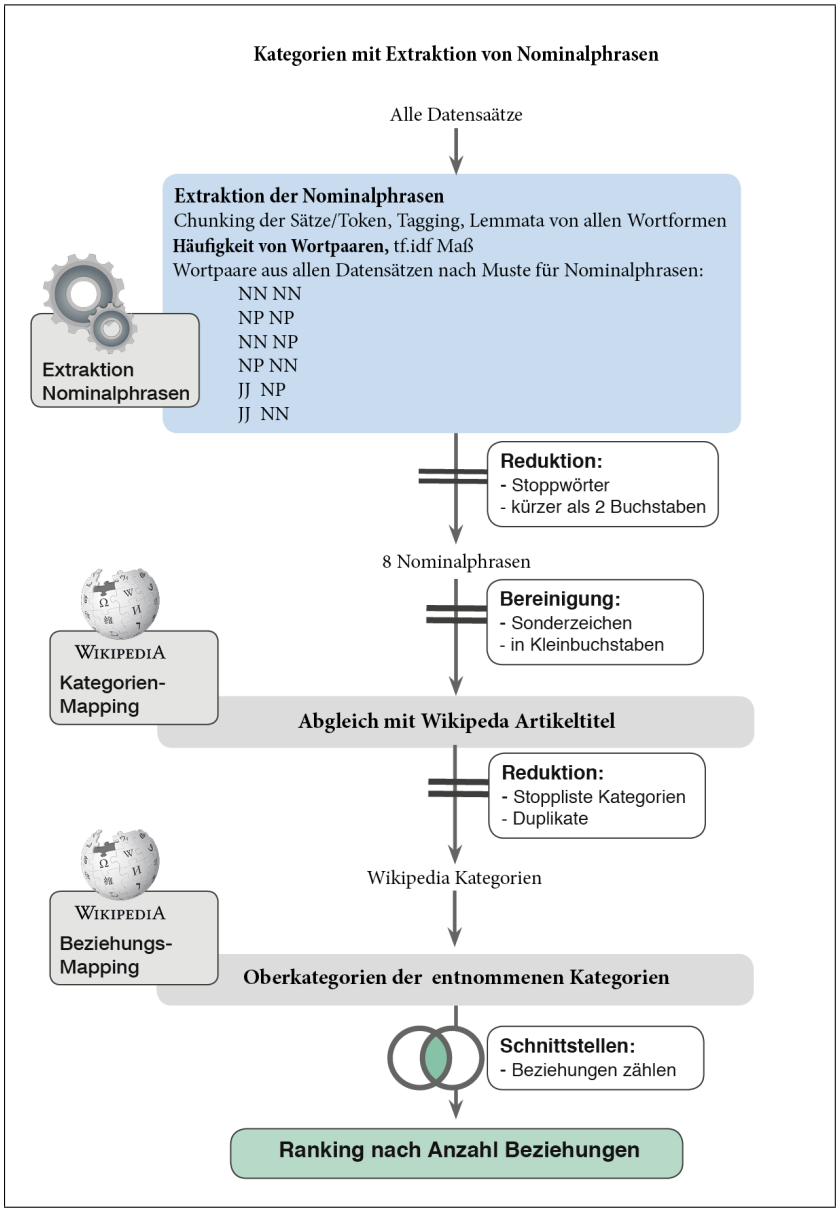


Abbildung 7.35: Verfahren des Mappings mithilfe der Nominalphrasen

Nominalphrasen verwendet. Um die Anfrage an die MediaWiki API gering zu halten

und auch die Qualität der Kategorien zu verbessern, wurden alle Nominalphrasen mit weniger als zwei Buchstaben pro Wort, nicht verwendet. Die Nominalphrasen wurden vollständig in Kleinbuchstaben umgewandelt, da das zweite Wort im Wikipedia Artikeltitel klein beginnen muss. Da die Nominalphrase mit beiden Wörtern im Titel vorkommen muss, ist die Ergebnismenge der Kategorien kleiner als bei dem Mapping mit den extrahierten Termen.

Die Kategorien der Nominalphrasen sind dafür genauer, weil durch die Verwendung von einem Wortpaar das Fachgebiet begrenzter ist. Damit für möglichst alle Abbildungen Kategorien zur Verfügung stehen, werden die Verfahren der Termextraktion und der Extraktion der Nominalphrasen kombiniert. Im Kapitel 5.4 wird die Kombination der extrahierten Terme und Nominalphrasen beschrieben und im nächsten Kapitel wird die Kombination aus beidem für das Mapping mit den Kategorien aufgezeigt.

8. Kategorienmapping mit Kombination aus Nominalphrasen und Termen

Mit den Nominalphrasen, ergänzt durch die extrahierten Terme, aus dem Kapitel 5.4 werden die Titel der Wikipedia Artikel erneut abgeglichen. Das Mapping mit dieser Kombination erfolgt fast identisch wie das Mapping mit den Nominalphrasen aus dem Kapitel 7. Der Unterschied besteht dabei in den Begriffen, die an die MediaWiki API übergeben werden. Wie in der Abbildung 8.36 zu sehen, werden 15 Nominalphrasen und/oder Terme an die Schnittstelle übermittelt. Wie im folgenden

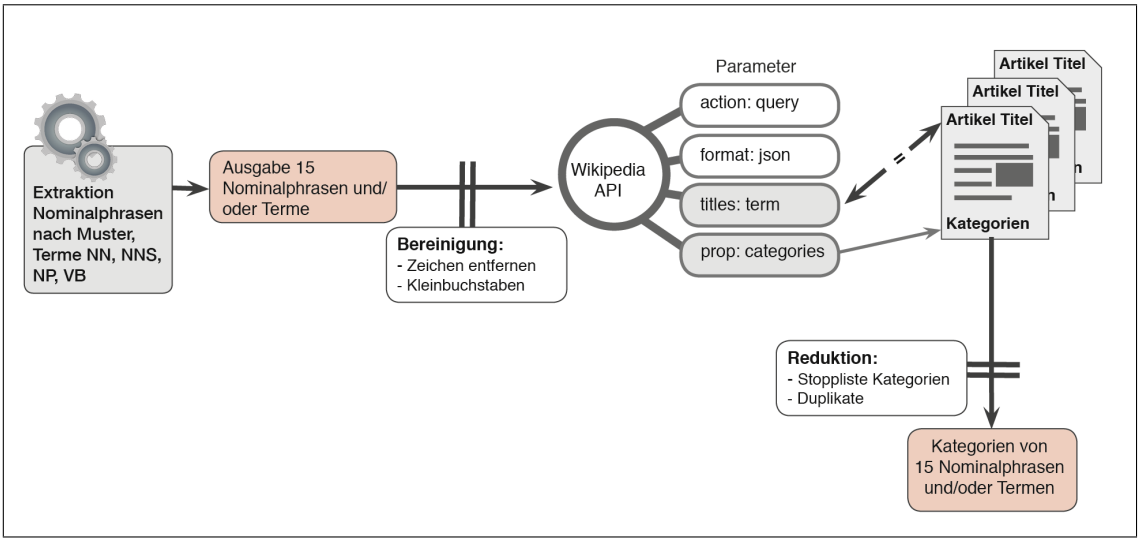


Abbildung 8.36: Modell Mapping der Kombination aus extrahierten Termen und Nominalphrasen

Code-Ausschnitt unter *if (l1,l2) in NPList:* aufgezeigt ist, werden die Wortpaare zuerst auf ein Vorhandensein in der Gesamtliste der Nominalphrasen aller Datensätze überprüft. Danach werden die einzelnen Wörter dieses Paares, die ein Nomen *NN*, oder ein Eigennamen *NP* sind, zur Abgleichliste hinzugefügt. Die Wörter, die nicht in den Wortpaaren vorkommen, aber von der Wortform Nomen Singular *NN*, Nomen Plural *NNS* oder Verben *VB)* sind, werden ebenfalls genutzt. Aus diesen Nominalphrasen und Termen werden, nach der Tf-Idf-Berechnung, die 15 besten an die Funktion, für den Aufruf der Wikipedia Schnittstelle, übergeben. Der vollständige Python Code dazu befindet sich im Anhang unter B.7.

```
def candidates(taglist):
    cand = []
    skip = False
    for i in range(len(taglist)-1) :
        if skip:
            skip = False
            continue
        skip = False
```

```
        l1 = taglist[i].lemma
        l2 = taglist[i+1].lemma
        if len(l1) >2 and len(l2) >2:
            if (l1,l2) in NPList:
                skip = True
                cand.append((l1,l2))
                if taglist[i].pos == 'NN' or taglist[i].pos == 'NP':
                    cand.append((l1))
            else:
                if taglist[i].pos == 'NN' or taglist[i].pos == 'VB' or taglist[i].pos == '
                    NNS':
                    cand.append(l1)
    w1 = taglist[-1]
    if w1.pos == 'NN'or w1.pos == 'VB' or w1.pos == 'NNS' or w1.pos == 'NP' or w1.pos == '
        NNP':
        cand.append(w1)

    return cand
```

Wie auch bei dem Mapping mit den Nominalphrasen, werden die Terme und Nominalphrasen vor dem Abgleich von Zeichen bereinigt und in Kleinbuchstaben umgewandelt. Die Kategorien, die so generiert werden, sind zum einen sehr detailliert, weil die Titel der Wikipedia Artikel aus den gleichen Wortpaaren bestehen und zum anderen kann durch das Mapping mit den Termen sichergestellt werden, dass fast jeder Datensatz und somit auch die dazugehörige Abbildung mit Kategorien versehen werden kann. Es gibt vereinzelt auch Datensätze, die keine Bildbeschriftung haben, oder in denen die Bildbeschriftung nur aus Abkürzungen bestehen. Wenn dann zusätzlich die Referenzstellen zu knapp ausfallen, können keine Nominalphrasen und auch keine Terme extrahiert werden.

In der Tabelle 8.14 sind zwei Beispiele aufgezeigt, die Kategorien aus der Kombination von Nominalphrasen und Termen erhalten haben.[100] Im ersten Datensatz wird beispielsweise für den extrahierten Term *osteochondritis*[101] die Kategorie *Chondropathies*[102] zurückgegeben. Diese Kategorie ist eine gute Einordnung für den Datensatz. Kombiniert mit den Kategorien die aus den anderen Termen und Nominalphrasen ermittelt wurden, kann der Datensatz *10.1155:2016:1979348.txt* nachvollziehbar in die Kategorie-Systematik von Wikipedia eingeordnet werden. Bei dem zweiten Datensatz, aus dem Beispiel in der Tabelle 8.14, werden die meisten Kategorien direkt aus dem Artikel *Synaptic weight* entnommen.

[100] Die Darstellung der Kategorien in der einen und die Terme und Nominalphrasen in der anderen Spalte zeigt keine 1:1 Gegenüberstellung. Es ist lediglich eine Aufzählung aller Terme und Nominalphrasen und auf der anderen Seite die dadurch generierten Kategorien.

[101] Osteochondritis dissecans ist eine Knorpelerkrankung, siehe dazu: `https://de.wikipedia.org/wiki/Osteochondrosis_dissecans`

[102] Chondropathie ist ein medizinischer Oberbegriff für pathologische Veränderungen im Gelenkknorpel, siehe dazu: `https://en.wikipedia.org/wiki/Osteochondritis` `https://en.wikipedia.org/wiki/Chondropathy`

Tabelle 8.14: Kategorien von Kombination aus Termen und Nominalphrasen

Terme und NPs	**Kategorien**
Datensatz: 10.1155:2016:1979348.txt	
osteochondritis	Category:Chondropathies
instability	Category:Concepts in physics
Chondrofix	Category:Inflammations
range 18–59	Category:Ligaments
Chondrofix plug	Category:Orthopedics stubs
MFC	Category:Plasma physics
tear	Category:Stability theory
resultant instability	Category:Systems theory
MFC OCD	
joint disease	
cruciate ligament	
average age	
Datensatz: 10.1155:2016:2303181.txt	
synaptic weight	Category:Artificial neural networks
predictor	Category:Computer science stubs
input variable	Category:Evaluation methods
output	Category:Evidence
input	Category:Machine learning algorithms
network	Category:Neural networks
representative simulation	Category:Neuroplasticity
holistic behavior	Category:Neuroscience stubs
backpropagation	Category:Scientific method
trained weight	
Case study	
linear transfer	

8.1. Ranking der Kategorien aus der Kombination

Das Ranking der Kategorien erfolgt bei der Kombination von Termen und Nominalphrasen ähnlich wie bei dem Ranking der Kategorien von den Nominalphrasen. In der Abbildung 8.37 wird die Vorgehensweise für die Berechnung der Beziehungen aufgezeigt. Die Kategorien der Ergebnismenge aus dem Abgleich mit den Wikipedia Artikel-Titel werden, wie im Kapitel 6.4 für die Kategorien der extrahierten Terme und auch im Kapitel 7.1 für die Kategorien der Nominalphrasen, erneut an die MediaWiki API übermittelt, um die Oberkategorien davon zu erhalten. Die Anzahl der Übereinstimmungen in den Oberkategorien ergeben jeweils die Anzahl der Beziehung. Alle Kategorien werden wieder sortiert für jeden Datensatz aufgezeigt. Der vollständige Python Code, zum Ranking der Kategorien aus den extrahierten Termen und Nominalphrasen, befindet sich im Anhang unter B.3.

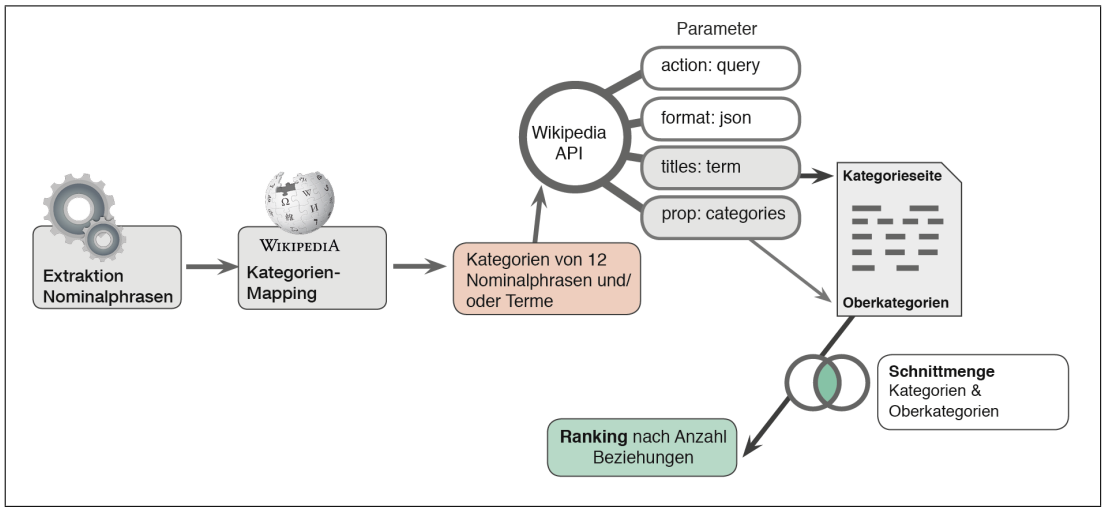


Abbildung 8.37: Modell Ranking der Kategorien für Mapping der kombinierten Terme und Nominalphrasen

8.2. Abschlussbetrachtung Kategorien aus der Kombination

Bei der Nutzung der Kombination aus Termen und Nominalphrasen kann ein Problem entstehen, wenn es zu viele Wortpaare gibt. Zu viele sind es, wenn es mehr als 15 sind, denn dann werden keine Terme mehr genutzt.[103] Wenn die Nominalphrasen so aber nicht in einem Titel von Wikipedia Artikel vorkommen, werden keine Kategorien ermittelt. Die Wörter der Nominalphrasen können dabei einzeln aussagekräftig sein, werden aber nicht berücksichtigt. Die Kategorien, die auf Grundlage der kombinierten Terme und Nominalphrasen ermittelt wurden, ordnen die meisten Abbildungen dennoch gut in die Kategoriesystematik der Wikipedia ein, siehe Tabelle 8.14.

Um die Qualität der erarbeiteten Text Mining-Verfahren testen und überprüfen zu können, wird im nächsten Kapitel eine Evaluation mit neuen Datensätzen durchgeführt. Diese Datensätze beschreiben Abbildungen, die schon vor dem Start dieser Arbeit bei Wikimedia Commons vorhanden waren und mit Kategorien versehen worden sind.

[103] Wie im Kapitel 5.4 aufgezeigt, werden 15 Terme und/oder Nominalphrasen, für jeden Datensatz, genutzt.

9. Evaluierung des Annotationsverfahrens

Das Ziel der durchgeführten Evaluation ist die Gewinnung von Erkenntnissen aus der ausgearbeiteten Annotationsmethode. Die Durchführung der Evaluation wird nach den empfohlenen Methoden und Vorgehensweisen von Stockmann (Stockmann 2004, S. 3) und der Deutsche Gesellschaft für Evaluation (DeGEval) (Gesellschaft für Evaluation 2016) umgesetzt. Mithilfe der Evaluation sollen Änderungsmaßnahmen, für die Annotation der Abbildungen im Projekt NOA, herausgestellt werden. Die Aufgabe dieser Evaluation ist die Verbesserung der Annotationsmethode, sie ist daher eine *ex-ante Evaluation* (Stockmann 2004, S. 5).[104] Nach der DeGEval sollte eine Evaluation die Eigenschaften *Nützlichkeit*, *Durchführbarkeit*, *Fairness* und *Genauigkeit* haben (Gesellschaft für Evaluation 2016, S. 14). Für die Evaluation der Annotationsmethode werden die Eigenschaften *Durchführbarkeit* und *Genauigkeit* aufgezeigt.

Durchführbarkeit:
Der Aufwand für die Evaluation wurde so gewählt, dass er im angemessenen Verhältnis zum Ziel der Evaluation steht (Gesellschaft für Evaluation 2016, S. 19).

Genauigkeit:
Der Umfang der Evaluation für die Annotationsmethode und die zugrundeliegenden Evaluationsobjekte werden ausreichend beschrieben und dokumentiert. Die eingesetzten Methoden werden aufgezeigt und können nachvollzogen werden. Die Schlussfolgerungen der Evaluation werden mithilfe der analysierten Daten begründet und können bewertet werden (Gesellschaft für Evaluation 2016, S. 20f).

Im Kapitel 9.4 folgt die Durchführung der Evaluation, anhand vorhandener Bilder aus Wikimedia Commons. Für die Bewertung der Qualität der Kategorien wird die Bewertungsmethode *Genauigkeit* (en: Precision) eingesetzt. Dies erfolgt im Kapitel 9.4.1. Um die fachliche Qualität der ermittelten Kategorien aufzuzeigen, wird im Kapitel 9.4.2 eine manuelle Evaluierung der Kategorien von einem Beispielbild durchgeführt. Eine ergänzende Evaluierung findet im Kapitel 9.5 statt. Für diese Evaluierung wurden 100 Abbildungen aus der NOA-Datenbank zu Wikimedia Commons übertragen und manuell mit vorhandenen Kategorien der Wikimedia Foundation kategorisiert.[105]

[104] Eine ex-ante Evaluation meint eine Evaluation, die vor dem Einsatz der Methode, das Konzept der Methode bewertet (Glosar 2012).
[105] Das Hochladen der 100 Abbildungen wurde von Lucia Sohmen durchgeführt. Die Kategorien wurden von Mitarbeiter*innen aus dem NOA-Projekt manuell vergeben.

Aufbau der Datensätze für die Evaluation

Für die Evaluation der entwickelten Mapping-Methode werden extrahierte Terme und auch Nominalphrasen verwendet. Die Testdaten für die Evaluierung befinden sich sowohl in der Datenbank von NOA, als auch bei Wikimedia Commons. Bei Wikimedia Commons wurden diese 58 Abbildungen schon mit Kategorien versehen, für die gleichen Abbildungen in der NOA-Datenbank wird das kombinierte Verfahren dieser Arbeit durchgeführt. Die ermittelten Kategorien werden mit den vorhandenen Kategorien bei Wikimedia Commons verglichen. Für die Extraktion der Terme und Nominalphrasen stehen wieder die Bildbeschriftungen und die Referenzstellen aus dem Text zur Verfügung.[106] Die Datensätze für die Evaluation stammen aus 14 wissenschaftlichen Open Access Artikeln. Der Aufbau und Umfang der Datensätze, mit denen die Verfahren erstellt wurden, werden im Kapitel 5.1 beschrieben. In der Tabelle 9.15 wird der Aufbau von beiden Datensets gegenüber gestellt. Aus der Tabelle wird deutlich, dass für die Entwicklung der Verfahren in vielen Datensätzen wenige Wörter vorhanden waren. Für die Evaluierung hat der umfangreichste Datensatz 11.945 Zeichen,[107] der kürzeste Datensatz hat 213 Zeichen. Im arithmetischen Mittel haben die Datensätze 2091,08 Zeichen.[108] Die meisten Datensätze haben über 500 Zeichen. Es sind nur 5 Datensätze dabei, die weniger als 500 Zeichen haben.

Tabelle 9.15: Vergleich der Trainingsdatensätze und der Testdatensätze für die Entwicklung und Evaluation der Methode

Trainingsdaten für die Entwicklung	**Testdaten für die Evaluation**
397 Datensätze	58 Datensätze
350,78 Zeichen im Mittel	2091,08 Zeichen im Mittel
12 Zeichen Min	213 Zeichen Min
3055 Zeichen Max	11945 Zeichen Max
95 Datensätze mehr als 500 Zeichen	53 Datensätze mehr als 500 Zeichen

9.1. Extrahierte Terme und Nominalphrasen aus den Testdatensätzen

Für die Extraktion der Terme und Nominalphrasen wird die gleiche Vorgehensweise wie in Kapitel 5.4 (Kombination von Nominalphrasen und Terme) gewählt.[109] Für die 58 Abbildungen wurden Datensätze mit den Bildbeschriftungen und den Referenzstellen aus dem Artikel angelegt. Da diese Abbildungen auch in der NOA Bildersuche verfügbar sind, konnten die Inhalte aus der Datenbank abgerufen werden.

[106] Im Anhang, unter C, befindet sich dazu die vollständige Liste der Links zu den Abbildungen bei Wikimedia Commons.
[107] Verwendete Excel-Funktion: =MAX()
[108] Excel-Funktion: =MITTELWERT()
[109] Der vollständige Python Code dazu befindet sich im Anhang unter B.8.

Für einzelne Abbildungen wurden die Referenzstellen manuell aus den Artikeln hinzugefügt.[110] Diese Datensätze bilden die Grundlage für die Extraktion der Terme und Nominalphrasen. In der Tabelle 9.16 werden die Ergebnisse der Extraktion für einen Testdatensatz aufgezeigt. Hier ist zu sehen, dass in den Referenzstellen und in der Bildbeschriftung viele Abkürzungen verwendet wurden. Die Herausforderung, bei

Tabelle 9.16: Extrahierte Terme und Nominalphrasen für einen Evaluierungs-Datensatz

Terme und/oder Nominalphrasen	**Tf-idf-Wert**
doi-10.1186:1471-2148-5-26-id0.txt	
sample	0.28
lineage	0.28
new lineage	0.28
phylogenetic tree	0.22
shaded region	0.14
T12477C mutation	0.14
haplogroup	0.14
M18	0.14
M18 lineage	0.14
HVSI	0.14
unclassified lineage	0.14
Whole	0.14
number	0.14
similar HVSI	0.14
sub-types	0.14

den Testdatensätzen für die Evaluation, sind die Datensätze mit wenig Rohmaterial, beispielsweise der Datensatz *doi-10.1186/1471-2148-5-59-id5.txt* mit nur 213 Zeichen.

10.1186/1471−2148−5−59
Active T. villosus foraging on a leaf . Note that the water−film on the leaf is adhering to the shell. This water film is usually in contact with the shell during locomotion (Figure 6).

Die Terme und die Nominalphrase, die aus diesem Rohtext extrahiert werden, sind: *villosus*, *shell*, *water film*, *contact note* und *water*. Beim Mapping, auf die Kategorien der Wikipedia, werden für diese Begriffe zwar Kategorien gefunden, diese beschreiben die Abbildung des Datensatzes jedoch nicht optimal, siehe Kategorien im Kapitel 9.3. Bei den Testdatensätzen mit über 500 Zeichen ist die Wahrscheinlichkeit höher aussagekräftigere Nominalphrasen und Termen zu erhalten. Für den Datensatz *doi-10.1186/1471-2148-9-210-id3.txt* werden die Begriffe aus der Tabelle 9.17 extrahiert. Dieser Datensatz besteht aus 790 Zeichen. Mit diesen Termen und vor allem der Nominalphrase *Lottia digitalis* können beim Mapping fachlich passende Kategorien ermittelt werden. Der Wikipedia Artikel *Lottia digitalis* hat die Kategorien: *Lottiidae*,

[110] Einige Autoren verwenden in ihren Artikeln Abbilder, die wiederum über Zusatzabbildungen verfügen, die sich teilweise in Bildshows verbergen. Die Referenzstellen dazu konnten nicht fehlerfrei in die NOA-Datenbank übertragen werden.

Tabelle 9.17: Terme und Nominalphrasen für Beispieldatensatz mit 790 Zeichen

doi-10.1186:1471-2148-9-210-id3.txt
statistical support
monophyly
Lottia digitalis
Lottia
reconstructed tree
amino acid
phylogram
inset
posterior probability
topology
bootstrap
amino

Molluscs of the Pacific Ocean, *Taxa named by Martin Rathke* und *Animals described in 1833*.[111] Weitere Kategorien, die für die Datensätze ermittelt wurden, werden im Kapitel 9.3 aufgezeigt.

9.2. Kategorien der Testdaten-Abbildungen aus Wikimedia Commons

Der Upload der Abbildungen bei Wikimedia Commons wurde automatisiert von einer Benutzer*in der Wikipedia durchgeführt. Die Kategorien der Abbildungen sind von den Benutzern Thiotrix, Daniel Mietchen, Bluerasberry, DePlusJean, NeverDoING, Snek01, Ruslik0 und Pierpao manuell ergänzt und bearbeitet worden. Für die Wikimedia Foundation Projekte *Wikipedia*, *Wikimedia Commons* und *Wikidata* wird der Aufbau der Kategoriensyntax in den Kapiteln 4.2, 4.3 und 4.4 beschrieben. Die Kategorien, die die Evaluations-Abbildungen bei Wikimedia Commons erhalten haben, setzen sich daraus zusammen. Die Abbildungen wurden am 19. Mai 2017 bei Wikimedia Commons abgerufen und wurden ursprünglich in Open Access Journals veröffentlicht.[112] In der Tabelle 9.18 sind die Wikimedia Commons Kategorien von drei Beispielabbildungen aufgezeigt. Beispiel 1 und 2 werden auch in dem vorherigen Kapiteln 9.1 (Extraktion der Terme und Nominalphrasen) und aus dem nächsten Kapitel 9.3 (Kategorienmapping) verwendet. Das dritte Beispiel ist eine Abbildung mit mehreren fachlichen Kategorien. Wie vor allem in den ersten beiden Beispielen zu sehen ist, werden in Wikimdia Commons auch Platzhalter-Kategorien und nicht fachliche Kategorien (*en: Non-topical*) oder versteckte Kategorien verwendet, beispielsweise die Kategorie *Uploaded with Open Access Media Importer* oder Unterkategorien

[111] Wikipedia Seite: `https://en.wikipedia.org/wiki/Lottia_digitalis`

[112] Die Abfrage aller Abbildungen aus Open Access Journals aus Wikimedia Commons wurde von Prof. Dr. Christian Wartena durchgeführt.

Tabelle 9.18: Wikimedia Commons Kategorien der Evaluations-Abbildungen

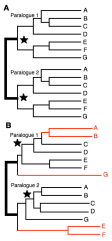
Kategorien bei Wikimedia Commons
1. File:Trochulus villosus.jpg
Media from BMC Evolutionary Biology
Open Access File of the Day
Trochulus villosus
2. File:Neogastropod-phylogenetic-relationships-based-on-entire-mitochondrial-genomes-1471-2148-9-210-2.jpg
All media needing categories as of 2015
Media from BMC Evolutionary Biology
Media needing categories as of 20 January 2015
Uploaded with Open Access Media Importer
Uploaded with reCitation Bot
Uploaded with reCitation Bot and needing category review
3. File:Representatives of ceratioid families.jpg
Bufoceratias shaoi
Bufoceratias wedli
Centrophryne spinulosa
Cryptopsaras couesii
Lasiognathus amphirhamphus
Melanocetus eustalus
Thaumatichthys binghami
Media from BMC Evolutionary Biology
Open Access File of the Day

der Kategorie *Media needing categories*. Von den 58 Abbildungen in Wikimedia Commons hat die Hälfte mehr als 5 zugewiesene Kategorien, darunter befinden sich aber, wie in der Tabelle 9.18 aufgeführt, viele versteckte und nicht-fachliche Kategorien. 10 Abbildungen haben bei Wikimedia Commons gar keine fachlichen Kategorien erhalten. Diese werden zur Evaluierung nicht weiter verwendet.
Des Weiteren haben 21 Bilder nur eine einzige fachliche Kategorie, die Kategorie *RNA polymerase*. Der Umgang mit diesen Kategorien wird am Ende des nächsten Kapitels aufgezeigt.

9.3. Kategoriemapping für Testdatensätzen aus NOA

Das Kategorienmapping wird, wie beim Mapping mit der Kombination von Termen und Nominalphrasen im Kapitel 8 beschrieben, durchgeführt. Die Begriffe werden, vor der Übergabe an die MediaWiki API, bereinigt und in Kleinbuchstaben umgewandelt. Alle Schritte der Vorgehensweise für das Mapping, werden in den Kapiteln 6.3 (Mapping für extrahierte Terme), Kapitel 7 (Mapping für extrahierte Nominalphrasen) und Kapitel 8 (Mapping für kombinierte Variante) detailliert aufgeführt. Der vollständige Python Code zum Mapping mit den Evaluationsdatensätzen befindet sich im Anhang unter B.9. In der Tabelle 9.19 werden die Kategorien für die Beispiele aus dem Kapitel 9.1 aufgezeigt. Hierbei hat der Datensatz mit den 213 Zeichen (35 Wörter) nur die extrahierten Terme und Nominalphrasen *villosus*, *shell*, *water film*, *contact note* und *water*. In diesem Beispiel sind alle ermittelten Kategorien vom extrahierten Term *water*. Wie schon im Kapitel 9.1 erwähnt, sind 35 Wörter für die Extrahierung von

Tabelle 9.19: Kategorien für zwei Evaluations-Datensätze

Datensatz mit 35 Wörtern *doi-10.1186/1471-2148-5-59-id5.txt*	**Datensatz mit 431 Wörtern** *doi-10.1186/1745-6150-1-19-id9.txt*
Extrahierte Terme und NPs: contact, water film, villosus, shell, water, note	**Extrahierte Terme und NPs:** murein, reverse direction, bar, posibacteria, murein wall, nuclear pore, change, crucial role, Simplified summary, murein peptidoglycan, intracellular enslavement, place, first time, mitochondrion, single burst
Kategorien: Category:Hydrogen compounds Category:Inorganic solvents Category:Liquids Category:Oxides Category:Oxygen compounds Category:Water	**Kategorien:** Category:Bartending Category:Cell nucleus Category:Cellular respiration Category:Endosymbiotic events Category:Membrane biology Category:Mitochondria Category:Nuclear pore complex Category:Types of drinking establishment Category:Types of restaurants
Abbildung: 	**Abbildung:** 

brauchbaren Termen, auch aufgrund der oft verwendeten Abkürzungen, zu wenig. Ab 500 Zeichen (ca. 95 Wörter) waren in den meisten der 397 Trainingsdaten und auch

in den 48 Testdatensätzen für die Evaluierung ausreichende und fachlich sinnvolle Terme vorhanden.[113]

In den Datensätzen für die Evaluierung befinden sich auch 21 Dateien, die aus einem medizinischen Artikel stammen und in denen viele Abkürzungen verwendet werden. Die Autoren verwenden in diesem Artikel Abkürzungen, beispielsweise *RNAP*. *RNAP* ist die Abkürzung für die aufgelöste Form von *RNA polymerase*. Bei der Verwendung der Abkürzung wird keine Übereinstimmung mit einem Wikipedia Artikel-Titel gefunden. Anders mit *RNA polymerase*, hier können die Kategorien *Gene expression*, *RNA*, *Enzymes* und *EC 2.7.7* ermittelt werden.

Deshalb wird im Kapitel 9.4 die Evaluierungen mit der kombinierten Methode aus Kapitel 8 durchgeführt, bei der die Abkürzungen durch die ausgeschriebene Form ersetzt werden. [114]

Abkürzungen auflösen:

Damit eine Evaluation mit den 48[115] verbliebenen Abbildungen durchgeführt werden kann, werden die Abkürzungen aufgelöst. Die Ergänzung der aufgelösten Abkürzungen werden vor dem Aufruf der MediaWiki API durchgeführt und werden mit den Titeln der Wikipedia-Artikel abgeglichen. Im folgenden Code ist ein Ausschnitt der aufgelösten Abkürzungen für die Abbildungen aufgeführt. Der vollständige Python Code dazu befindet sich im Anhang unter B.9.

```
acronym ={"rifsv" : "rifamycin sv", "ge" : "GE23077", "rnap" : "rna polymerase", "rif" : "
    rifamycin", "ic50" : "half maximal inhibitory concentrations", "sor" : "sorangicin", "
    rnap-sor" : "rna polymerase sorangicin", "rpo" : "rnap promoter open complex", "rna": "
    ribonucleic acid", "rnap-ge" : "rna polymerase GE23077",
....#Aus Platzgruenden entfernt}
```

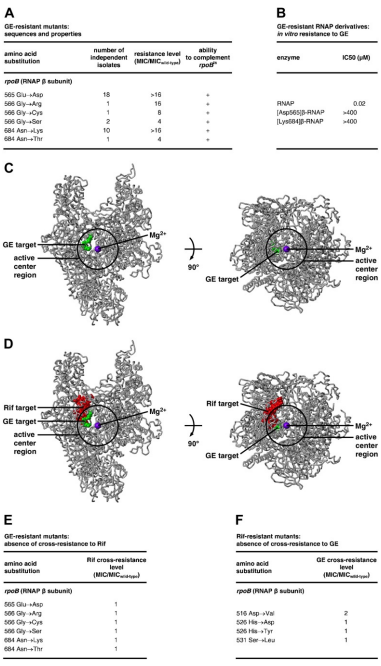
In der Tabelle 9.20 sind die Kategorien aus dem Datensatz *doi-10.7554:eLife.02450-id3.txt* aufgezeigt. In der linken Spalte werden die extrahierten Abkürzungen aus dem Datensatz nicht durch ihre ausgeschriebene Form ersetzt, bevor das Kategorien-Mapping durchgeführt wird. Hier sind jetzt auch aus dem Artikel *Rif* 8 Kategorien, beispielsweise *Category:Rif* vorhanden. Die rechte Spalte zeigt die Kategorien, bei denen die extrahierten Abkürzungen *RNAP* und *Rif* aufgelöst wurden in *RNA polymerase* und *rifamycin*. Durch das Auflösen der Abkürzungen kann die Anzahl der zutreffenden Kategorien erhöht werden. Die ausgeschriebenen Abkürzungen wurden für die Evaluierung manuell hinzugefügt und ersetzen vor der Übergabe an die

[113] Dies ist für die, in dieser Arbeit, verwendeten Datensätze der Fall. Da es sich um wissenschaftliche Artikel handelt, werden in den Bildbeschriftungen häufig Abkürzungen verwendet. Mit diesen Abkürzungen ist ein Abgleich mit den Wikipedia Artikel-Titeln selten erfolgreich. Für Datensätze mit anderen strukturellen und fachlichen Gegebenheiten kann diese Einschätzung unzutreffend sein.

[114] Der vollständige Python Code dazu befindet sich im Anhang unter B.9.

[115] 10 der 58 Abbildungen haben bei Wikimedia Commons keine fachlichen Kategorien erhalten, deshalb ist hier eine Überprüfung der Kategorien hinfällig.

Tabelle 9.20: Optimierte Kategorien durch Auflösung von Abkürzungen

Datensatz: doi-10.7554:eLife.02450-id3.txt	
Mit Abkürzungen:	**Aufgelöste Abkürzungen:**
Category:Rif	Category:Biochemistry
Category:Geography of Fès-Meknès	Category:Elementary mathematics
Category:Geography of Oriental (Morocco)	Category:Immunology stubs
Category:Geography of Tanger-Tetouan-Al Hoceima	Category:Laboratory techniques
Category:Toxicology stubs	Category:Otologicals
Category:Mountain ranges of Morocco	Category:Rifamycin antibiotics
Category:Laboratory techniques	Category:Sequences and series
Category:Toxicology	Category:Titration
Category:Immunology stubs	Category:Toxicology
Category:Biochemistry	Category:Toxicology stubs
Category:Sequences and series	
Category:History of Morocco	
Category:Titration	
Category:Elementary mathematics	
Category:Berber history	
Category:Al Hoceïma Province	
Aufgelöste Abkürzung:	**Abbildung:**
RNA polymerase (RNAP)	
rifamycin (Rif)	

MediaWiki API die Abkürzungen. Für alle Abbildungen im Projekt NOA wird die Auflösung der Abkürzungen später jedoch automatisiert umgesetzt werden.

Die Gewichtung der Kategorien erfolgt, wie auch für die Kategorien mit den Termen, als Ausgangslage im Kapitel 6.4 und auch für die Nominalphrasen im Kapitel 7.1 und wie in der kombinierten Version im Kapitel 8.1. Das Ranking wird bestimmt, indem die Oberkategorien jeder ermittelten Kategorie verglichen wird, dabei wird jede Übereinstimmungen als Beziehung addiert. Der vollständige Python Code dazu befindet sich im Anhang unter B.10.[116]

9.4. Durchführung der Evaluierung

Für die Evaluierung werden die Kategorien der Testdatensätze aus Kapitel 9.3 mit den vorhandenen Kategorien der Abbildungen aus Wikimedia Commons, siehe Kapitel 9.2 verglichen. Es sind 48 Bilder, die bei Wikimedia Commons fachliche Kategorien haben und deshalb für die Evaluierung genutzt werden. Die Evaluation prüft die Übereinstimmung der ermittelten Kategorien mit den vorhanden Kategorien der gleichen Abbildungen. Es wird keine intellektuelle Bewertung der fachlichen Qualität der Kategorien von Wikimedia Commons vorgenommen. Die Evaluation wird anhand der folgenden Kriterien durchgeführt:

- Terminologische Übereinstimmung der ermittelten Kategorien
- Semantische Übereinstimmung der ermittelten Kategorien
- Inkorrekte Kategorien

Die Bewertung erfolgt dabei für das Kriterium *Terminologische Übereinstimmung der ermittelten Kategorien* in Form der Überprüfung der terminologischen Konsistenz (Iivonen 1995).[117] Der Umfang der terminologischen Konsistenz beinhaltet u.a. die Singular- und Pluralform eines Begriffes (einer Kategorie) (Iivonen 1995, S. 6). Das Kriterium *Terminologische Übereinstimmung der ermittelten Kategorien* beschreibt in dieser Arbeit aber nicht nur die Kategorien, die terminologisch konsistent sind, sondern auch die Kategorien, die mit einem Teilwort einer Kategorie übereinstimmen. Ein Beispiel hierfür ist, aus dem Datensatz *doi-10.1186:1471-2148-5-26-id0.txt*, die Kategorie *Phylogenetics*. Die Abbildung des Datensatzes hat bei Wikimedia Commons die Kategorie *Phylogenetic trees*. Die Kategorie *Phylogenetics* wurde deshalb dem Kriterium *Terminologische Übereinstimmung der ermittelten Kategorien* zugeordnet. Die Bewertungskriterien für die ermittelten Kategorien wurden so festgelegt, weil die Kategorien von Wikipedia und Wikimedia Commons identisch sein können, aber

[116] Die Rohdaten der Evaluations-Datensätze stehen auf GitHub unter: `https://github.com/f-josi/MA` zur Verfügung.

[117] Mit der terminologischen Konsistenz ist die Überschneidung zweier Begriffe gemeint.

nicht zwingend identisch sein müssen. Der Aufbau der Kategoriensystematik von Wikipedia, Wikimedia Commons und Wikidata wird im Kapitel 4 beschrieben.

Das Kriterium *Semantische Übereinstimmung der ermittelten Kategorien* wird für Kategorien vergeben, die konzeptuell konsistent sind. Bei der konzeptuellen Konsistenz wird die semantische Überschneidung bewertet (Gazendam u. a. 2009). Ein Beispiel für diese Bewertung ist dem Datensatz *doi-10.1186:1741-7007-6-33-id1.txt* entnommen. Die für diesen Datensatz ermittelte Kategorie ist *Echinoderms*, die Kategorie aus Wikimedia Commons ist *Echinoidea anatomy*. Beide Kategorien stehen sich semantisch sehr nahe, sind aber nicht terminologisch übereinstimmend. Die Kategorien, die weder terminologisch noch semantisch übereinstimmen, werden mit *Inkorrekte Kategorien* bewertet. Hier wurden keine Übereinstimmungen mit den gegebenen Kategorien bei Wikimedia Commons gefunden. In dieser Evaluation werden die ermittelten Kategorien nur anhand der gegebenen Kategorien von Wikimedia Commons bewertet, es wird keine Aussage über die fachliche Qualität getroffen.

In der Abbildung 9.38 sind die Ergebnisse der Evaluation aufgezeigt. Für das Kategoriemapping wurden jeweils 15 extrahierten Termen und/oder Nominalphrasen genutzt. In dem Diagramm ist zu sehen, dass der Datensatz mit der Nummer 39, keine Kategorien erhalten hat. Die extrahierten Terme und Nominalphrasen werden dazu in der Tabelle 9.21 aufgezeigt. Für diesen Datensatz wurden keine Kategorien ermittelt, weil die Abkürzungen, als Teil einer Nominalphrase, nicht aufgelöst werden konnten. Diese Herausforderung wird in den Empfehlungen im Kapitel 10, unter dem ersten Punkt, näher beschrieben.

Tabelle 9.21: Terme und Nominalphrasen für den Datensatz 39

doi-10.7554:eLife.02450-id19.txt	Tf-idf-Werte
bipartite inhibitor	0.54
compound	0.13
7C–E	0.12
high potency	0.12
proof-of-concept	0.12
wild-type	0.11
wild-type RNAP	0.11
figure	0.08
wildtype	0.07
structural characterization	0.07
wildtype RNAP,	0.07
10.7554/eLife.02450 Synthesis	0.07
method	0.07
linker	0.07
step	0.07

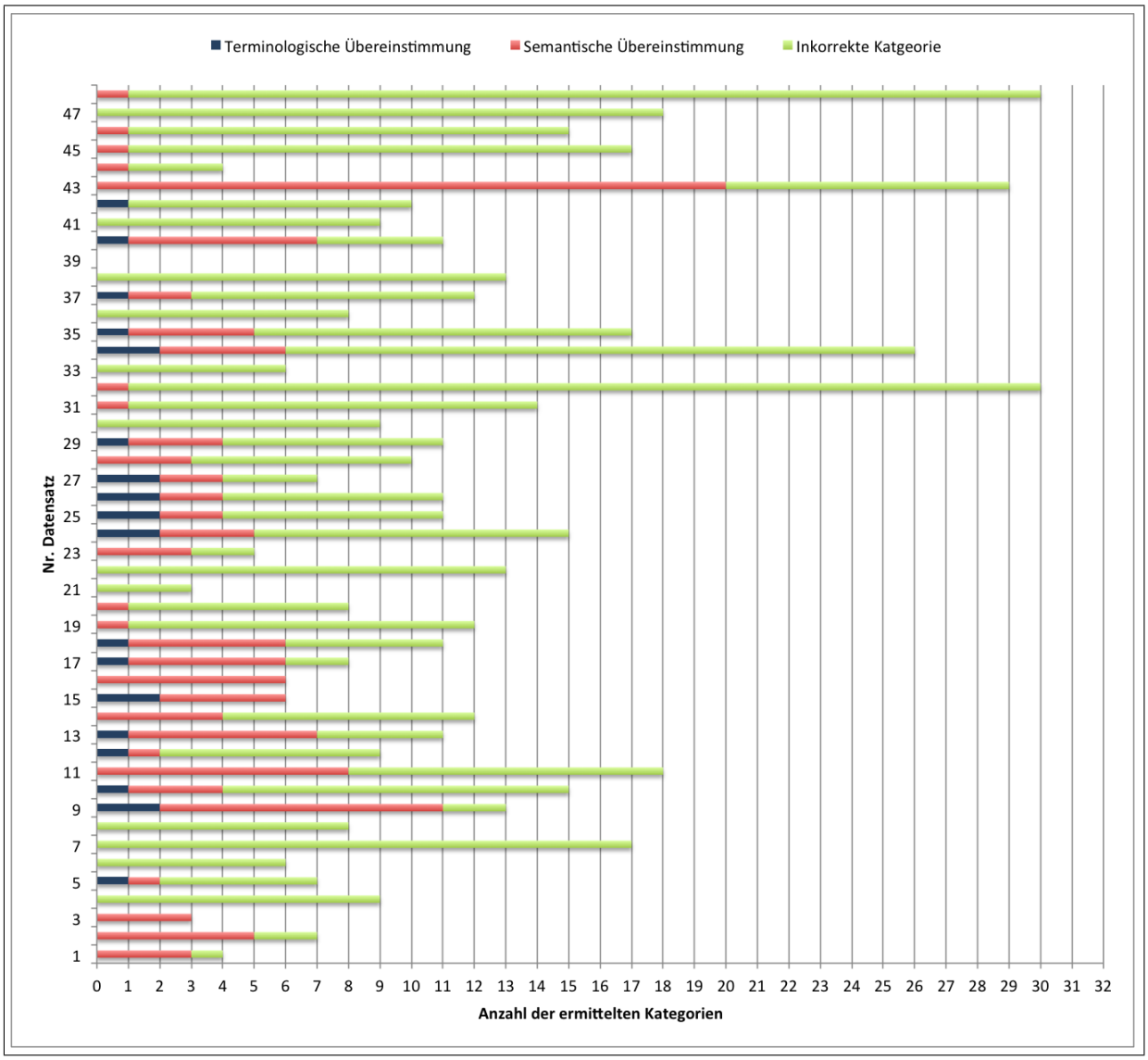


Abbildung 9.38: Ergebnisse der Kategorien-Evaluation mit 15 Termen und/oder Nominalphrasen. Es sind automatisiert ermittelte Wikipedia-Kategorien im Vergleich zu vorhandenen Wikimedia Commons Kategorien.

Des Weiteren werden für 12 Datensätze keine semantischen oder terminologischen Übereinstimmungen in den Kategorien gefunden. In 18 Datensätzen gibt es mindestens eine Übereinstimmung aus dem Kriterium *Terminologische Übereinstimmung der ermittelten Kategorien*. Die meisten übereinstimmenden Kategorien sind semantisch konsistent. Zu dem Kriterium *Semantische Übereinstimmung der ermittelten Kategorien* konnten 34 Datensätze zugeordnet werden. Die meisten Übereinstimmungen konnten für den Datensatz 9 (*doi-10.1186:1741-7007-6-33-id1.txt*) erreicht werden. Zu diesem Datensatz sind in der Tabelle 9.22 die extrahierten Terme und Nominalphrasen, die ermittelten Kategorien, die Wikimedia Commons-Kategorien und die Übereinstimmungen aufgezeigt. In dem letzten Abschnitt der Tabelle 9.22 sind die Kategorien aufgeführt, die eine terminologische oder eine semantische Konsistenz darstellen.

Da die terminologisch konsistenten Kategorien, im Kontext eines Datensatzes, auch semantisch konsistent sind, werden beide Kriterien für die weitere Evaluierung zusammengefasst.

Tabelle 9.22: Datensatz mit der höchsten terminologischen oder semantischen Konsistenz der Kategorien

Datensatz: *doi-10.1186:1741-7007-6-33-id1.txt*	
Terme und Nominalphrasen aus Extraktion:	**Kategorien aus Extraktion Terme und Nominalphrasen:**
lantern, 0.29	Category:Stomach
lantern muscle, 0.29	Category:Marine animals
Eucidaris metularia, 0.23	Category:Echinoderms
Echinocyamus pusillus, 0.23	Category:Aquatic deuterostomes
Echinoneus, 0.23	Category:Organs (anatomy)
gastric caecum, 0.23	Category:Digestive system
Echinoneus cyclostomus, 0.23	Category:Cidaridae
file, 0.23	Category:Abdomen
Echinocyamus, 0.23	Category:Cidaroida genera
Eucidaris, 0.23	Category:Extant Cambrian first appearances
bilateral, 'symmetry'), 0.11	Category:Light fixtures
datasets, 0.11	Category:Taxa named by Jean-Baptiste Lamarck
stomach, 0.11	Category:Animals described in 1816
echinoderm, 0.11	
urchin, 0.11	
Wikimedia Commons Kategorien:	
Echinocyamus pusillus	
Echinoidea anatomy	
Echinoneus cyclostomus	
Eucidaris metularia	
Psammechinus miliaris	
Kategorien mit terminologischer oder semantischer Konsistenz:	
Category:Stomach	
Category:Marine animals	
Category:Echinoderms	
Category:Aquatic deuterostomes	
Category:Organs (anatomy)	
Category:Digestive system	
Category:Cidaridae	
Category:Abdomen	
Category:Cidaroida genera	
Category:Extant Cambrian first appearances	
Category:Taxa named by Jean-Baptiste Lamarck	

9.4.1. Beurteilung der relevanten Kategorien

Die Beurteilung des Annotationsverfahrens wird anhand der Berechnung von der Genauigkeit (en: Precision) durchgeführt. Dieses Maß kann zur Bewertung von Klassifikationssystemen eingesetzt werden (Dengel 2012, S. 155).[118]
Die Genauigkeit wird berechnet mit:

$$\text{Genauigkeit} = \frac{\text{relevante Kategorien} \cap \text{gefundene Kategorien}}{\text{gefundene Kategorien}}$$

In der Abbildung 9.39 werden die Maße für die Genauigkeit der semantischen Konsistenz der Kategorien für jeden Datensatz aufgezeigt. Der Datensatz mit der Nummer 39 hat mit der Annotationsmethode keine Kategorien erhalten. Der Mittel-

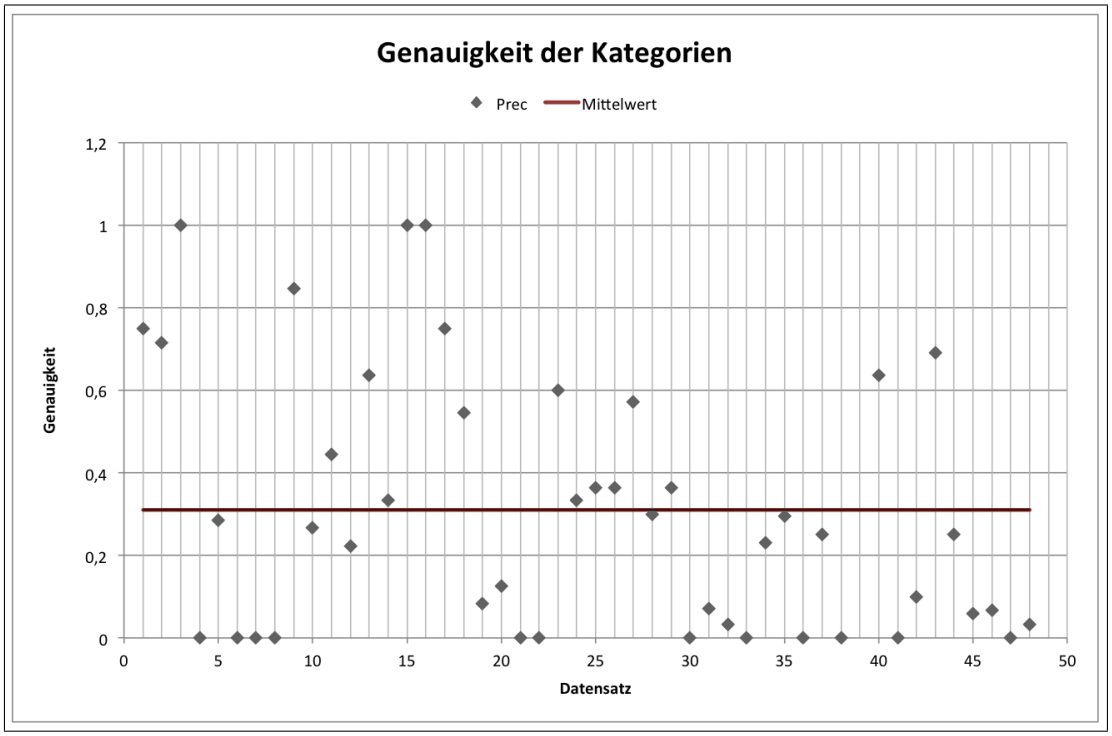


Abbildung 9.39: Werte für die Genauigkeit der semantischen Konsistenz für jeden Datensatz, mit 15 Termen und/oder Nominalphrasen für das Kategoriemapping.

wert für die Genauigkeit in der semantischen Übereinstimmung der Kategorien liegt bei 0,31. D.h. die Datensätze haben im Mittel eine semantische Übereinstimmung von 31%. Dies liegt z.T. daran, dass einige Datensätze viele inkorrekte Kategorien erhalten haben. Ein Grund dafür sind die verwendeten Abkürzungen in den Bildbeschriftungen und Textreferenzen. Da ein Mapping mit der Abkürzung zu einem Wikipedia Artikeltitel nicht möglich ist und die Auflösung innerhalb einer Nominalphrase nicht

[118] Die Trefferquote (en: Recall) wird nicht berechnet. Die Abbildungen bei Wikimedia Commons haben wenige Kategorien und in der Annotationsmethode werden oft mehrere semantisch übereinstimmende Kategorien gefunden. Dazu wird eine weitere Evaluierung vorbereitet, siehe Kapitel 10.

möglich war, siehe Kapitel 9.3, werden für das Mapping Terme eingesetzt, die einen niedrigeren Tf-idf-Wert haben.

Die Evaluierung wurde zusätzlich mit 5 und mit 10 Terme und/oder Nominalphrasen durchgeführt. In der Tabelle 9.23 werden zu allen drei Varianten die Mittelwerte der Genauigkeit für die semantische Übereinstimmung der Kategorien aufgezeigt.

Tabelle 9.23: Übersicht der Mittelwerte für die Genauigkeit der Kategorien mit 5, 10 und 15 Terme und/oder Nominalphrasen für das Kategoriemapping

Anzahl Terme und Nominalphrasen	**Mittelwert semantischer Konsistenz**
5	0,34
10	0,30
15	0,31

Der Mittelwert, für die semantische Übereinstimmung der Kategorien, verbessert sich zwar bei der Verwendung von 5 Terme und/oder Nominalphrasen um 0,03%, dafür werden aber für 14 von 48 Datensätzen gar keine Kategorien ermittelt. Im Gegensatz dazu wird bei der Verwendung von 15 Termen und/oder Nominalphrasen nur für einen Datensatz keine Kategorie ermittelt.

9.4.2. Manuelle Evaluierung

Zusätzlich, zu der Bewertung der Genauigkeit der semantischen Übereinstimmung der Kategorien aus dem Annotationsverfahren und Wikimedia Commons, werden die Kategorien von einem einzigen Datensatz manuell auf ihre semantische Übereinstimmung evaluiert.[119] Der verwendete Datensatz dafür ist *10.1016:j.humpath.2016.07.013.txt*. Dieser Datensatz beschreibt die erste Abbildung des Artikels mit der DOI: 10.1016/j.humpath.2016.07.013. In der Abbildung 9.40 ist eine visuelle Ansicht von dem beschriebenen Bild zu sehen. In dem wissenschaftlichen Artikel geht es um Minichromosomale Maintenance (MCM)-Proteine, die als Marker zur Bestimmung von Speiseröhrenkrebs genutzt werden können. Dazu wird Gewebe auf das Vorhandensein der Proteine mikroskopisch untersucht.[120] In der Tabelle 9.24 sind die Terme und Wortpaare, die manuell auf Grundlage des Abstrakts vergeben wurden, dargestellt. Diese vergebenen Begriffe wurden in Wikipedia als Artikel manuell gesucht. Die Kategorien, in die diese Artikeln eingeordnet wurden, werden für den Vergleich genutzt.

[119] Der Datensatz für diese Evaluierung wurde nach dem Zufallsprinzip aus dem Bereich 50:100 der 397 Datensätze gewählt. Das Annotationsverfahren wurde anhand der Ergebnisse der Datensätze aus dem Bereich 150:170 entwickelt. Die Abbildungen aus Wikimedia Commons (Evaluierung aus den vorangegangenen Kapiteln) werden dafür nicht genutzt, weil sie fast nur über 1 bis 3 fachliche Kategorien verfügen.

[120] Die Zusammenfassung des Inhaltes erfolgt auf die Angaben im Abstrakt des Artikels.

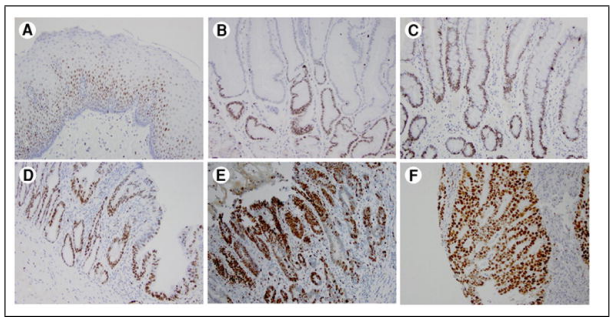


Abbildung 9.40: Abbildung des Datensatzes für die manuelle Evaluation

Die manuell beschreibenden Begriffe sind:

1. **Immunohistochemistry** *Deutsche Beschreibung: Immunhistochemie ist eine Methode mit der Proteine sichtbar gemacht werden*
2. **Minichromosomale Maintenance Proteine** *Deutsche Beschreibung: Spezielle Proteine als Marker für Tumoren*
3. **Esophageal carcinoma** *Deutsche Beschreibung: Speiseröhrenkrebs*

Tabelle 9.24: Manuelle Kategorien für einen Datensatz

Terme und Wortpaare	**Wikipedia Artikel**	**Kategorien der Artikel**
Immunohistochemistry	Immunohistochemistry	Category:Immunohistochemistry, Category:Histology, Category:Immunologic tests, Category:Protein methods, Category:Anatomical pathology, Category:Staining, Category:Laboratory techniques, Category:Pathology
Minichromosomale Maintenance Proteine	Minichromosomale Maintenance	Category:DNA replication, Category:Protein stubs
Esophageal carcinoma	Esophageal cancer	Category:Gastrointestinal cancer

Die Kategorien, die mit dem entwickelten Annotationsverfahren aus dieser Arbeit ermittelt wurden sind in der Tabelle 9.25 zu sehen. Wie in den Tabellen 9.24 und 9.25 zu sehen ist, gibt es eine vollständige terminologische Konsistenz für die Kategorie *Category:Anatomical pathology* und eine weitere terminologische Konsistenz für ein Teilwort der Kategorie *Category:Pathology* und *Category:Anatomical pathology*. Für die Kategorie *Category:Histology* (Gewebelehre) und *Category:Histopathology* (Lehre von krankhaften Gewebeveränderungen) besteht eine semantische Konsistenzen, die auf einer Oberkategorie beruht. Des Weiteren gibt es eine semantische Konsistenz für die Kategorie *Gastrointestinal cancer* zu den automatisch ermittelten Kategorien *Category:Oncology* und *Category:Glands*. Die automatisch ermittelten Kategorien *Category:Glands*, *Category:Medical signs* und *Category:Induced stem cells* finden

Tabelle 9.25: Kategorien aus Annotationsverfahren für Datensatz aus Abbildung 9.40

Kategorien aus Annotationsverfahren *10.1016:j.humpath.2016.07.013.txt*
Category:Anatomical pathology
Category:Histopathology
Category:Oncology
Category:Glands
Category:Medical signs
Category:Induced stem cells

keine Übereinstimmung. Zusammengefasst zeigt diese manuelle Evaluation zwei terminologische und zwei semantische Übereinstimmungen der Kategorien auf. Drei Kategorien, die mit dem Annotationsverfahren ermittelt wurden, besitzen keine Übereinstimmung.

9.5. Ergänzende Evaluierung

Wie im Kapitel 9 erwähnt folgt eine erweiterte Evaluierung, die auf 100 Abbildungen beruht, die aus der NOA-Datenbank zu Wikimedia Commons übertragen wurden.[121] Die Kategorien für diese Abbildungen wurden von Mitarbeiter*innen aus dem NOA-Projekt vergeben. Dabei wurden nur Kategorien vergeben, die von Wikimedia Commons zur Verfügung gestellt wurden.[122] Die erweiterte Evaluierung wurde durchgeführt, weil für die Evaluierung aus Kapitel 9.4 nur 48 Bilder aus Wikimedia Commons verwendet werden konnten. Die dafür genutzten Bilder waren sowohl in der NOA-Datenbank als auch bei Wikimedia Commons vorhanden.

Für die ergänzende Evaluation, mit den zugrundeliegenden 100 Bildern, werden 825 Kategorien aus der Annotationsmethode und 264 Kategorien von Wikimedia Commons verglichen. Für die Extraktion der Kategorien wird das Verfahren aus Kapitel 5.4 eingesetzt. Dabei werden für jeden Datensatz insgesamt 15 Terme und/oder Nominalphrasen ermittelt. Mit diesen Begriffen wird das Mapping mit den Wikipedia Artikeltiteln durchgeführt. Für die Begriffe, die mit einem Titel übereinstimmen, werden die Kategorien des Wikipediaartikels übernommen. Diese Kategorien werden mit den vorhandenen Wikimedia Commons Kategorien der 100 Abbildungen verglichen. Die Gründe für die gemeinsame Nutzung der Wikipedia- und Wikimedia Commons-Kategorien werden im Kapitel 4.5 aufgezeigt. Für diese Evaluierung werden die Bewertungsmethoden *Genauigkeit* (en: Precision) und *Trefferquote* (en: Recall) eingesetzt.

Tabelle 9.26: Gesamte Übereinstimmungen der Kategorien aus der Annotationsmethode und Wikimedia Commons

Übereinstimmung	**Prozent von Wikimedia Commons Kategorien**
Terminologisch	12,88%
Semantisch	60,23%
Gesamt fachlich	73,11%

In der Tabelle 9.26 ist der Prozentsatz der Übereinstimmung zwischen den Kategorien aus der Annotationsmethode und den Kategorien der Wikimedia Commons für alle Abbildungen gemeinsam gelistet. Insgesamt gibt es für 12,88% aller Kategorien von Wikimedia Commons eine terminologische Übereinstimmung in den Kategorien aus der Annotationsmethode. Bei der terminologischen Übereinstimmung wird die Singular- und Pluralform der Kategorien als Übereinstimmung gewertet. Die

[121] Die Bilder sind auf Wikimedia Commons unter folgendem Link erreichbar: `https://commons.wikimedia.org/w/index.php?title=Special:ListFiles/Sohmen&ilshowall=1`.

[122] Es ist bei den Projekten der Wikimedia Foundation jederzeit möglich neue Kategorien zu erstellen, siehe `https://commons.wikimedia.org/wiki/Commons:Categories` für das Erstellen neuer Kategorien in Wikimedia Commons.

semantische Übereinstimmung beträgt 60,23%. Die terminologischen Übereinstimmungen werden in den semantischen Übereinstimmungen nicht mit aufgenommen. Demnach ist eine gesamte fachliche Übereinstimmung der Kategorien von 73,11% vorhanden. Aufgeteilt nach den einzelnen Abbildungen sind die Übereinstimmungen wie in der Tabelle 9.27 aufgezeigt verteilt. Für 7 Abbildungen konnten mit der Annotationsmethode keine Kategorien ermittelt werden und 20 Abbildungen hatten keine Übereinstimmung der Kategorien. Für 3 Abbildungen gab es nur terminologische Übereinstimmungen, 24 Abbildungen hatten sowohl terminologische, als auch semantische Übereinstimmungen und 46 Abbildungen hatten nur semantische Übereinstimmungen der Kategorien.

Tabelle 9.27: Übersicht über die ermittelten Kategorien aus der Annotationsmethode für die 100 Abbildungen der ergänzenden Evaluierung

Anzahl Abbildungen	**Übereinstimmung der Kategorien**
7	Keine ermittelten Kategorien
3	Terminologisch
46	Semantisch
24	Terminologisch & Semantisch
20	Keine

Den 825 Kategorien aus mit der Annotationsmethode stehen 264 Kategorien aus Wikimedia Commons gegenüber. Wie in der Tabelle 9.28 zu sehen, sind davon für 34 eine terminologische Übereinstimmung und für 159 eine semantische Übereinstimmung vorhanden. Für 632 ermittelte Kategorien gibt es keine Übereinstimmung bei den Kategorien aus Wikimedia Commons, dies bedeutet jedoch nicht, dass die Kategorien fachlich inkorrekt sind. Die Kategorien der Annotationsmethode klassifizieren die Abbildungen im Kontext des wissenschaftlichen Artikels. Die Kategorien bei Wikimedia Commens wurden für die sichtbare Inhalte des Bildes vergeben, der Kontext des ursprünglichen Artikels wurde nicht berücksichtigt. Des Weiteren wurden für die Bilder bei Wikimedia Commons meist zwei bis drei Kategorien vergeben, aus der Annotationsmethode dagegen im Durchschnitt acht Kategorien. Die Genauigkeit der Kategorien wird berechnet mit der Anzahl der gefundenen und relevanten Kategorien geteilt durch die Anzahl aller gefundenen Kategorien. Die genaue Vorgehensweise dazu wird im Kapitel 9.4.1 aufgezeigt. Der Mittelwert der Genauigkeit (en: Precision) liegt hier bei der ergänzenden Evaluierung bei 0,3.

Des Weiteren wird die Trefferquote der ermittelten Kategorien für jede Abbildung betrachtet. Die Beurteilung der Trefferquote (en: Recall) der Kategorien aus der Annotationsmethode wird berechnet mit:

$$\text{Trefferquote} = \frac{\text{relevante Kategorien} \cap \text{gefundene Kategorien}}{\text{relevante Kategorien}}$$

Tabelle 9.28: Genauigkeit der Kategorien

Übereinstimmung	**Anzahl der Kategorien**
Terminologische	34
Semantische	159
Keine	632

Für die terminologische und semantische Übereinstimmung haben die Kategorien einen Recallwert von 0,42. Wird für die Berechnung des Recalls nur mit den terminologischen Übereinstimmungen der Kategorien aus der Annotationsmethode und den Wikimedia Commons Kategorien durchgeführt beträgt er 0,32, siehe Tabelle 9.29.

Tabelle 9.29: Trefferquote der Kategorien

Übereinstimmung	**Recallwert**
Terminologisch und Semantische	0,42
Terminologisch	0,32

Die ergänzende Evaluierung zeigt, dass die 100 Abbildungen, die zu Wikimedia Commons übertragen und manuell mit Kategorien versehen wurden, mit den ermittelten Kategorien aus der Annotationsmethode eine Trefferquote von 0,42 bei einer Genauigkeit von 0,3 haben. Durch diese Evaluierung kann begründet werden, dass die Kategorien der Annotationsmethode eine gute Grundlage für eine automatisierte Kategorisierung mithilfe der Kategorien der Wikimedia Foundation darstellen.

10. Diskussion und Verwendung für Projekt NOA

Das Annotationsverfahren wurde im Rahmen dieser Masterarbeit erstellt, um Parameter für die Vorgehensweise im DFG-Projekt NOA zu prüfen. Die Ergebnisse der Evaluation fließen in die Planung des Annotationsverfahren für die Abbildungen aus NOA ein.

Empfehlungen aus der Evaluation:

1. Abkürzungen in den Bildbeschriftungen und Textreferenzen
2. Länge der Bildbeschriftung
3. Unterschiede in Wikipedia Kategorien und Wikimedia Commons Kategorien
4. Besonderheit der Facettenklassifikation
5. Reduzierung der Kategorienmenge
6. Wikimedia API

Zu 1. Da es sich bei den Abbildungen in dem Projekt NOA um wissenschaftliche Abbildungen handelt, werden sehr häufig Abkürzungen verwendet. Diese Abkürzungen werden bei der Extraktion der Nominalphrasen als ein Bestandteil der Phrase erkannt und können so beim Mapping mit den Artikeln der Wikipedia, zu keiner Übereinstimmung führen. Abkürzungen, die vor der Extraktion aufgelöst werden, können eigenständige Nominalphrasen bilden und für das Mapping genutzt werden. Die Abkürzungen, die als einzelne Terme extrahiert werden, können aber auch in einem späteren Schritt aufgelöst werden. So bleibt der berechnete Tf-Idf-Wert der Abkürzung erhalten. Ein Gegenargument für das Auflösen der Abkürzungen vor der Extraktion ist jedoch, dass Abkürzungen auch aus mehr als zwei Wortbestandteilen bestehen können. Diese Phrasen können bei dem Annotationsverfahren nicht vollständig als Nominalphrase für das Mapping genutzt werden. Eine weitere Schwierigkeit stellen zusammengesetzte Abkürzungen dar, also zwei oder mehrere Abkürzungen, die mit einem Bindestrich verbunden sind. Die Vorgehensweise für den Umgang mit diesen kombinierten Abkürzungen sollte im Vorfeld geprüft werden.

Zu 2. Die Bildbeschriftungen sollten eine ausreichende Länge haben, ansonsten können, wie in dieser Arbeit, die Textreferenzstellen für die Extraktion der Terme und Nominalphrasen mit dazugenommen werden.

Zu 3. Wie im Kapitel 4.3 beschrieben, gibt es Unterschiede in der Kategoriensyntax von Wikipedia und Wikimedia Commons. Die Kategorien aus diesem Annotationsverfahren bestehen aus Wikipedia-Kategorien. In Wikimedia Commons können zusätzlich Kategorien vergeben werden, beispielsweise aus den Oberkategorien *Media types*, *Copyright statuses* und *Media by source*. Die fachlichen Kategorien werden bei Wikimedia Commons jedoch überwiegend aus Wikipedia entnommen. Die Verwendung der hier genannten Wikimedia Commons Kategorien ist nicht Teil dieser Masterarbeit.

Zu 4. Für das Ranking der Kategorien wird die Schnittmenge der gemeinsamen Oberkategorien ausgewertet, siehe z.B. Kapitel 6.4. Bei der, neben der hierarchischen Klassifikation, möglichen Facettenklassifikation bei Wikipedia, kann es vorkommen, dass die Facettenkategorien keine gemeinsamen Oberkategorien haben. Sie werden dadurch beim Ranking nicht berücksichtigt.

Zu 5. In diesem entwickelten Annotationsverfahren wird die Kategoriemenge anhand einer manuell erstellten Stoppliste reduziert. Die Kategorien, die mit der Stoppliste entfernt werden, sind beispielsweise *Category:10 (number)* oder *Category:Numbers*. Fachliche Kategorien, die jedoch die Abbildung falsch einordnen, sind zum Teil vorhanden. Der Umgang mit diesen Kategorien muss noch erarbeitet werden.

Zu 6. Die Schnittstelle der Wikipedia (MediaWiki Internetservice API) wird für jeden extrahierten Term und jede Nominalphrase genutzt. Das sind für jede der 5. Mio Abbildungen jeweils 15 Terme und Nominalphrasen. Das Kategorien-Mapping für alle Abbildungen aus NOA sollte vorzugsweise auf einem Wikipedia Dump[123] stattfinden.

Verwendung in der NOA-Bildersuche:

Die Kategorien, die mit diesem Annotationsverfahren ermittelt werden, können in der Oberfläche der NOA Bildersuche als Kategorien zur Verschlagwortung eingesetzt werden.[124] Dadurch können die Kategorien als zusätzliche Suchmöglichkeit eingesetzt werden.

Nachnutzbarkeit in der Wikipedia:

Des Weiteren soll das Annotationsverfahren dazu beitragen, Grundlagen zu schaffen, um Open Access Abbildungen auf Wikimedia Commons bereitstellen zu können. Die Autoren der Wikipedia sollen dadurch ihre Artikel bequemer visualisieren können.

[123] Wikipedia Dumps stehen unter `https://dumps.wikimedia.org/` zur Verfügung
[124] NOA Scientific Image Search: `http://noa.wp.hs-hannover.de/`

In der Abbildung 10.41 ist die Nutzung der Bilder vom Projekt NOA beispielhaft für den Wikipedia Artikel *Hip* zu sehen.[125] Durch das Hochladen der Bilder zu Wikimedia Commons stehen die Abbildungen einer großen Zielgruppe zur Verfügung. Die Metadaten der Abbildungen sollen maschinenlesbar und sprachneutral als Datengrundlage für die Schwestern-Projekte, beispielsweise Wikidata, verwendet werden.

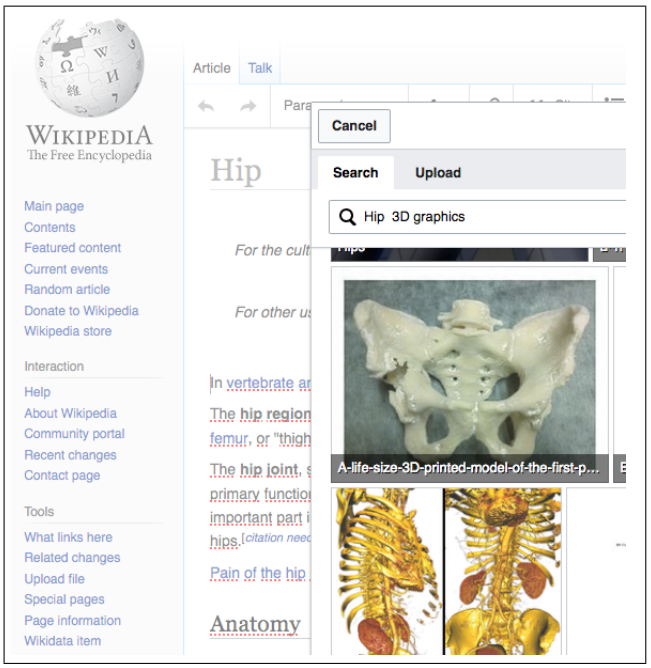


Abbildung 10.41: Beispiel für eine Nachnutzung der Abbildungen in der Wikipedia

[125] Das Bild des gedruckten 3D-Modells einer Hüfte wurde zu Testzwecken schon zu Wikimedia Commons übertragen und stammt aus der Datenbank des NOA-Projektes.

11. Zusammenfassung

Ziel dieser Arbeit war es, die Nachnutzbarkeit von wissenschaftlichen Abbildungen im Sinne von Open Science zu verbessern. Eingesetzt wurde dafür eine entwickelte Annotationsmethode, die dazu dient, den Abbildungen Kategorien aus Wikipedia zu vergeben. Die Abbildungen sollen im Anschluss kategorisiert bei Wikimedia Commons zur freien Nachnutzung, für Wikipedia-Artikel und Forschungsarbeiten, zur Verfügung stehen. Die Annotationsmethode, die anhand der Testdaten erstellt und evaluiert wurde, dient zur Entwicklung einer übergreifenden Annotation für den gesamten Bestand der Abbildungen im Projekt NOA. Die Anzahl der Abbildungen beträgt derzeit 5 Mio.[126]

Die Ausgangsfrage dieser Arbeit war: Kann das Kategoriensystem der Wikimedia Foundation automatisiert für die Kategorisierung der Bilder aus Open Access Journals eingesetzt werden? Das Ergebnis der Evaluation aus dieser Annotationsmethode zeigt, dass die Kategorien der Wikimedia Foundation für eine Kategorisierung geeignet sind, um die Abbildungen passend zu kategorisieren. Die Herausforderungen, die zu klären sind, werden im vorangegangenem Kapitel 10 beschrieben.

[126]Stand: Dezember 2017

Literatur

AG 2017

AG, Open S.: *Open Science AG - Definition Open Science*. https://www.ag-openscience.de/open-science/. Version: 2017

Alberts 2017

Alberts, Anna: *Offene Wissenschaft > Open Knowledge Foundation Deutschland*. https://okfn.de/themen/offene-wissenschaft/. Version: 2017

Baeza-Yates u. Ribeiro-Neto 1999

Baeza-Yates, Ricardo A. ; Ribeiro-Neto, Berthier: *Modern Information Retrieval*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1999. – ISBN 978–0–201–39829–8

Berliner-Erklärung 2003

Berliner-Erklärung: *Berliner Erklärung über offenen Zugang zu wissenschaftlichem Wissen*. http://openaccess.mpg.de/Berliner-Erklaerung. Version: 2003

Blümel u. a. 2014

Blümel, Ina ; Cartellieri, Simone ; Heller, Lambert ; Wartena, Christian: Entwicklung eines Verfahrens zur automatischen Sammlung, Erschließung und Bereitstellung multimedialer Open-Access-Objekte mittels der Infrastruktur von Wikimedia Commons und Wikidata. (2014). https://serwiss.bib.hs-hannover.de/frontdoor/index/index/docId/675

BSI 2017

BSI: *BSI Bundesamtes für Sicherheit in der Informationstechnik - Glossar - IT-Grundschutz*. https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/Glossar/glossar_node.html. Version: 2017. – Stand: 13. EL Stand 2013

Commons 2017a

Commons, Creative: *Creative Commons — Attribution-ShareAlike 3.0 IGO — CC BY-SA 3.0 IGO*. https://creativecommons.org/licenses/by-sa/3.0/igo/. Version: Juni 2017

Commons 2017b

Commons, Wikimedia: *Category:CommonsRoot - Wikimedia Commons*. https://commons.wikimedia.org/wiki/Category:CommonsRoot. Version: Juni 2017

Commons 2017c

Commons, Wikimedia: *Commons:Categories - Wikimedia Commons*. https://commons.wikimedia.org/wiki/Commons:Categories. Version: August 2017

Commons 2017d

Commons, Wikimedia: *Commons:Kategorien - Wikimedia Commons*. https://commons.wikimedia.org/wiki/Commons:Kategorien. Version: März 2017

Commons 2017e

Commons, Wikimedia: *Commons:Project scope - Wikimedia Commons*. https://commons.wikimedia.org/wiki/Commons:Project_scope/en. Version: August 2017

Commons 2017f

Commons, Wikimedia: *Commons:Welcome - Wikimedia Commons*. https://commons.wikimedia.org/wiki/Commons:Welcome. Version: August 2017

Dengel 2012

Dengel, Andreas: *Semantische Technologien: Grundlagen - Konzepte - Anwendungen*. Heidelberg : Springer Berlin Heidelberg, 2012. – ISBN 978–3–8274–2664–2

DFG 2010

DFG: *Umgang mit Forschungsdaten - DFG-Leitlinien zum Umgang mit Forschungsdaten*. http://www.dfg.de/foerderung/antrag_gutachter_gremien/antragstellende/antragstellung/nachnutzung_forschungsdaten/. Version: 2010. – Aktualisierungsdatum: 29.10.2015

DFG 2013

DFG (Hrsg.): *Sicherung guter wissenschaftlicher Praxis, Deutsche Forschungsgemeinschaft*. Wiley-VCH Verlag GmbH & Co. KGaA, 2013 http://onlinelibrary.wiley.com/doi/10.1002/9783527679188.oth1/summary. – ISBN 978–3–527–67918–8. – DOI: 10.1002/9783527679188.oth1

DFG 2016

DFG: *Verwendungsrichtlinien - Deutsche Forschungsgemeinschaft*. http://www.dfg.de/formulare/2_10/. Version: 2016

ESA 2017

ESA: *Open Access: ESA bekräftigt Strategie des offenen Zugangs für Bilder, Videos und Daten*. http://www.esa.int/ger/ESA_in_your_country/Germany/Open_Access_ESA_bekraeftigt_Strategie_des_offenen_Zugangs_fuer_Bilder_Videos_und_Daten. Version: Februar 2017

Frank u. a. 1999

Frank, Eibe ; Paynter, Gordon W. ; Witten, Ian H. ; Gutwin, Carl ; Nevill-Manning, Craig G.: Domain-Specific Keyphrase Extraction. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. San Francisco, CA, USA :

Morgan Kaufmann Publishers Inc., 1999 (IJCAI '99). – ISBN 978–1–55860–613–5, 668–673

Bueno de la Fuente 2017

Fuente, Gema Bueno de l.: *What is Open Science? Introduction*. https://www.fosteropenscience.eu/content/what-open-science-introduction. Version: 2017

Gabler 2017

Gabler, Springer: *Definition » Wiki « | Gabler Wirtschaftslexikon*. http://wirtschaftslexikon.gabler.de/Definition/wiki.html. Version: 2017

Gazendam u. a. 2009

Gazendam, Luit ; Wartena, Christian ; Malaisé, Véronique ; Schreiber, Guus ; Jong, Annemieke d. ; Brugman, Hennie: Automatic Annotation Suggestions for Audiovisual Archives: Evaluation Aspects. In: *Interdisciplinary Science Reviews* 34 (2009), September, Nr. 2-3, 172–188. http://dx.doi.org/10.1179/174327909X441090. – DOI 10.1179/174327909X441090. – ISSN 0308–0188

Gesellschaft für Evaluation 2016

Gesellschaft für Evaluation (Hrsg.): *Standards für Evaluation*. Köln : Geschäftsstelle DeGEval, 2016. – ISBN 978–3–941569–06–5. – Erste Revision 2016

Glosar 2012

Glosar, Wiki: *Ex-ante-Evaluation – Eval-Wiki: Glossar der Evaluation*. https://eval-wiki.org/glossar/Ex-ante-Evaluation. Version: Oktober 2012

Hacker 2017

Hacker, Andrea: Software für den Publikationsworkflow und den Peer-Review-Prozess (6a) : Praxishandbuch Open Access. Version: Mai 2017. https://www.degruyter.com/view/books/9783110494068/9783110494068-033/9783110494068-033.xml. In: *Praxisbuch Open Access*. Berlin, Boston : De Gruyter Saur, Mai 2017. – ISBN 978–3–11–049203–3

Hagstrom 2014

Hagstrom, Stephanie: *The FAIR Data Principles*. https://www.force11.org/group/fairgroup/fairprinciples. Version: September 2014

Hauschke u. Herb 2017

Hauschke, Christian ; Herb, Ulrich: *Open Knowledge Foundation - Definition: Offenes Wissen - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge*. http://opendefinition.org/od/1.1/de/. Version: 2017. – Version v.1.1

Helmholtz-Gemeinschaft 2016

Helmholtz-Gemeinschaft: Die Ressource Information besser nutzbar machen! (2016), Oktober. https://www.helmholtz.de/fileadmin/user_upload/01_forschung/Open_Access/DE_AKOS_TG-Forschungsdatenleitlinie_Positionspapier.pdf

Helmholtz-Gemeinschaft 2017

Helmholtz-Gemeinschaft: *Helmholtz Open Science: Helmholtz Open Science*. http://os.helmholtz.de/. Version: 2017

Herb 2012a

Herb: *Open Research Glossary*. https://figshare.com/articles/Open_Research_Glossary/1482094. Version: 2012

Herb 2012b

Herb, Ulrich (Hrsg.): *Open initiatives: Offenheit in der digitalen Welt und Wissenschaft*. Saarbrücken : Universaar, 2012 (Saarbrücker Schriften zur Informationswissenschaft). – ISBN 978–3–86223–062–4

Heyer u. a. 2008

Heyer, Gerhard ; Quasthoff, Uwe ; Wittig, Thomas: *Text mining: Wissensrohstoff Text: Konzepte, Algorithmen, Ergebnisse*. Korrigierter Nachdr. Herdecke [u.a.] : W3L-Verl., 2008 (Informatik). – ISBN 978–3–937137–30–8

Iivonen 1995

Iivonen, Mirja: Consistency in the selection of search concepts and search terms. In: *Information Processing & Management* 31 (1995), März, Nr. 2, 173–190. http://dx.doi.org/10.1016/0306-4573(95)80034-Q. – DOI 10.1016/0306–4573(95)80034–Q. – ISSN 03064573

Jurafsky u. Martin 2000

Jurafsky, Daniel ; Martin, James H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 1st. Upper Saddle River, NJ, USA : Prentice Hall PTR, 2000. – ISBN 978–0–13–095069–7

Klein 2014

Klein, Bernd: *Einführung in Python 3: für Ein- und Umsteiger*. 2., überarbeitete und erweiterte Auflage. München : Hanser, 2014. – ISBN 978–3–446–44133–0 978–3–446–44151–4

Leong u. a. 2010

Leong, Chee W. ; Mihalcea, Rada ; Hassan, Samer: Text Mining for Automatic

Image Tagging. In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Stroudsburg, PA, USA : Association for Computational Linguistics, 2010 (COLING '10), 647–655

Marcus u. a. 1993
Marcus, Mitchell P. ; Marcinkiewicz, Mary A. ; Santorini, Beatrice: Building a Large Annotated Corpus of English: The Penn Treebank. In: *Comput. Linguist.* 19 (1993), Juni, Nr. 2, 313–330. http://dl.acm.org/citation.cfm?id=972470.972475. – ISSN 0891–2017

Max-Planck-Gesellschaft 2017a
Max-Planck-Gesellschaft: *Open Access-Aktivitäten der Max-Planck-Gesellschaft*. https://openaccess.mpg.de/3570/Aktivitaeten. Version: 2017

Max-Planck-Gesellschaft 2017b
Max-Planck-Gesellschaft: *Signatoren Berliner Erklärung*. https://openaccess.mpg.de/3883/Signatories. Version: 2017

Medelyan u. a. 2008
Medelyan, Olena ; Witten, Ian H. ; Milne, David N.: Topic indexing with Wikipedia. AAAI Technical Report WS-08-15 (2008), 19–24. http://researchcommons.waikato.ac.nz/handle/10289/1776

MediaWiki 2017a
MediaWiki: *API-Main page - MediaWiki*. https://www.mediawiki.org/wiki/API:Main_page. Version: Oktober 2017

MediaWiki 2017b
MediaWiki: *Help:Images - MediaWiki*. https://www.mediawiki.org/wiki/Help:Images#Rendering_a_single_image. Version: Februar 2017

Mihalcea u. Csomai 2007
Mihalcea, Rada ; Csomai, Andras: Wikify!: Linking Documents to Encyclopedic Knowledge. In: *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*. New York, NY, USA : ACM, 2007 (CIKM '07). – ISBN 978–1–59593–803–9, 233–242

Mihalcea u. Tarau 2004
Mihalcea, Rada ; Tarau, Paul: Textrank : Bringing order into text. In: *Proc. 9th Conference on Empirical Methods in Natural Language Processing (EMNLP'04)* (2004). http://ci.nii.ac.jp/naid/20001460576/

Neuhold 2016

Neuhold, Andreas ; Schön, Sandra (Hrsg.) ; Ebner, Martin (Hrsg.): *Open Science: Potentiale eines neuen Wissenschaftsansatzes*. 1. Norderstedt : Books on Demand, 2016 (Beiträge zu offenen Bildungsressourcen (O3R) 12). – ISBN 978–3–7412–2610–6

Neuroth 2012

Neuroth, Heike: *Langzeitarchivierung von Forschungsdaten: eine Bestandsaufnahme*. Boizenburg : vwh, Hülsbusch, 2012. – ISBN 978–3–86488–008–7

Neuschaefer u. a. 2017

Neuschaefer, M ; Hamann, Hanjo ; Heftberger, Adelheid ; Goller, Marion ; Arslan, Ruben: *Handbuch Open Science – Wikibooks, Sammlung freier Lehr-, Sach- und Fachbücher*. `https://de.wikibooks.org/wiki/Handbuch_Open_Science#Strategien_und_Verfahren_von_Open_Science`. Version: 2017. – Living Book vom 07.08.17

Niemeyer 2016

Niemeyer, Sandra: *Projekt von TIB und Hochschule Hannover bewilligt: Nachnutzung von Open-Access-Abbildungen*. Version: Juni 2016. `https://idw-online.de/de/news654774`

North 2016

North, Klaus: *Wissensorientierte Unternehmensführung: Wissensmanagement gestalten*. 6., akt. und erw. Aufl. 2016. Wiesbaden : Springer Fachmedien Wiesbaden, 2016. – ISBN 978–3–658–11643–9

OKFN 2018

OKFN: *Open Definition 2.1 - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge*. `http://opendefinition.org/od/2.1/en/`. Version: 2018. – Version 2.1

Open-Access 2017

Open-Access: *Informationsplattform Open Access: Open-Access-Strategien*. `https://open-access.net/informationen-zu-open-access/open-access-strategien/`. Version: 2017

Pfeiffenberger 2017

Pfeiffenberger, Hans: Data Publishing und Open Access (8) : Praxishandbuch Open Access. Version: Mai 2017. `https://www.degruyter.com/view/books/9783110494068/9783110494068-038/9783110494068-038.xml`. In: *Praxisbuch Open Access*. Berlin; Boston : De Gruyter Saur, Mai 2017. – ISBN 978–3–11–049203–3

Raub u. Romhardt 2010

Raub, S. ; Romhardt, K.: *Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen*. 6., überarbeitete und erweiterte Auflage. Wiesbaden : Gabler Verlag / GWV Fachverlage GmbH, Wiesbaden, 2010. – ISBN 978–3–8349–8597–2

Rijsbergen 1979

Rijsbergen, C. J. V.: *Information Retrieval*. 2nd. Newton, MA, USA : Butterworth-Heinemann, 1979. – ISBN 978–0–408–70929–3

Ritze u. a. 2013

Ritze, Dominique ; Eckert, Kai ; Pfeffer, Magnus: Forschungsdaten. Version: 2013. `https://www.degruyter.com/downloadpdf/books/9783110278736/9783110278736.122/9783110278736.122.xml`. In: *(Open) Linked Data in Bibliotheken*. Berlin [u.a.] : De Gruyter Saur, 2013 (Bibliotheks- und Informationspraxis / Gantert, Klaus. - Berlin : de Gruyter Saur, 2010- 50). – ISBN 978–3–11–027873–6 978–3–11–027634–3, 122–138

Rosenbaum 2016

Rosenbaum, Konstanze: Von Fach zu Fach verschieden. Diversität im wissenschaftlichen Publikationssystem. Version: 2016. `https://edoc.bbaw.de/frontdoor/index/index/docId/2651`. In: *Weingart, Peter / Taubert, Niels (Hrsg.): Wissenschaftliches Publizieren : zwischen Digitalisierung, Leistungsmessung, Ökonomisierung und medialer Beobachtung.* Berlin, Boston : De Gruyter Akademie Forschung, 2016 (Forschungsberichte / Interdisziplinäre Arbeitsgruppen, Berlin-Brandenburgische Akademie der Wissenschaften 38). – ISBN 978–3–11–044810–8, 41–74. – BBAW / Interdisziplinäre Arbeitsgruppe Zukunft des wissenschaftlichen Kommunikationssystems

Schmid 1994

Schmid, Helmut: Probabilistic Part-of-Speech Tagging Using Decision Trees. In: *International Conference on New Methods in Language Processing*. Manchester, UK, 1994, S. 44–49

Schmitz 2017

Schmitz, Jasmin: Informations- und Qualitätssicherungswerkzeuge (6c–f) : Praxishandbuch Open Access. Version: Mai 2017. `https://www.degruyter.com/view/books/9783110494068/9783110494068-035/9783110494068-035.xml`. In: *Praxisbuch Open Access*. Berlin, Boston : De Gruyter Saur, Mai 2017. – ISBN 978–3–11–049203–3

Stockmann 2004

Stockmann, Reinhard: *Was ist eine gute Evaluation? Einführung zu Funktionen und*

Methoden von Evaluationsverfahren. Saarbrücken, 2004 (CEval-Arbeitspapier 9). http://nbn-resolving.de/urn:nbn:de:0168-ssoar-118018

TerminosaurusRex 2007

TerminosaurusRex: *TerminosaurusRex - Trex - Die Informationswissenschaft in Begriffen – Fachrichtung Informationswissenschaft Saarbrücken*. https://trex.infowiss.net/index.php?id=2.3.2.1.1.5. Version: August 2007

Turney 2000

Turney, Peter D.: Learning Algorithms for Keyphrase Extraction. In: *Inf. Retr.* 2 (2000), Mai, Nr. 4, 303–336. http://dx.doi.org/10.1023/A:1009976227802. – DOI 10.1023/A:1009976227802. – ISSN 1386–4564

Voss u. a. 2014

Voss, Jakob ; Bausch, Susanna ; Schmitt, Julian ; Bogner, Jasmin ; Berkelmann, Viktoria ; Ludemann, Franziska ; Löffel, Oliver ; Kitroschat, Janna ; Bartoshevska, Maiia ; Seljuzki, Katharina: Normdaten in Wikidata. (2014), Mai. https://serwiss.bib.hs-hannover.de/frontdoor/index/index/docId/438

Wartena u. Brussee 2008

Wartena, Christian ; Brussee, Rogier: Instanced-Based Mapping between Thesauri and Folksonomies. In: Sheth, Amit (Hrsg.) ; Staab, Steffen (Hrsg.) ; Dean, Mike (Hrsg.) ; Paolucci, Massimo (Hrsg.) ; Maynard, Diana (Hrsg.) ; Finin, Timothy (Hrsg.) ; Thirunarayan, Krishnaprasad (Hrsg.): *The Semantic Web - ISWC 2008*, Springer Berlin Heidelberg, Oktober 2008 (Lecture Notes in Computer Science). – ISBN 978–3–540–88563–4 978–3–540–88564–1, 356–370

Weingart 2016

Weingart, Peter: Zur Situation und Entwicklung wissenschaftlicher Bibliotheken. Version: 2016. https://edoc.bbaw.de/frontdoor/index/index/docId/2651. In: *Weingart, Peter / Taubert, Niels (Hrsg.): Wissenschaftliches Publizieren : zwischen Digitalisierung, Leistungsmessung, Ökonomisierung und medialer Beobachtung.* Berlin, Boston : De Gruyter Akademie Forschung, 2016 (Forschungsberichte / Interdisziplinäre Arbeitsgruppen, Berlin-Brandenburgische Akademie der Wissenschaften 38). – ISBN 978–3–11–044810–8, 103–121. – BBAW / Interdisziplinäre Arbeitsgruppe Zukunft des wissenschaftlichen Kommunikationssystems

Wikidata 2017a

Wikidata: *Help:Navigation in Wikidata - Wikidata*. https://www.wikidata.org/wiki/Help:Navigating_Wikidata/de. Version: Juni 2017

Wikidata 2017b

Wikidata: *Wikidata:Einführung - Wikidata*. https://www.wikidata.org/wiki/Wikidata:Introduction/de. Version: Juni 2017

Wikidata 2017c

Wikidata: *Wikidata:Liste der Eigenschaften/zusammenfassende Tabelle - Wikidata*. https://www.wikidata.org/wiki/Wikidata:List_of_properties/Summary_table/de. Version: Juli 2017

Wikimedia 2017

Wikimedia: *Wikimedia Foundation*. https://de.wikipedia.org/w/index.php?title=Wikimedia_Foundation&oldid=168082515. Version: August 2017. – Page Version ID: 168082515

Wikimedia-Deutschland 2017a

Wikimedia-Deutschland: *Fellow-Programm Freies Wissen. Wissenschaft offen gestalten. – Wikimedia Deutschland*. https://www.wikimedia.de/wiki/BildungWissenschaftKultur/Fellowprogramm. Version: 2017

Wikimedia-Deutschland 2017b

Wikimedia-Deutschland: *Wikimedia Deutschland*. https://www.wikimedia.de/wiki/Hauptseite. Version: 2017

Wikimedia-Deutschland 2017c

Wikimedia-Deutschland: *Wikipedia: Wikimedia Deutschland*. https://de.wikipedia.org/w/index.php?title=Wikimedia_Deutschland&oldid=166675007. Version: Juni 2017. – Page Version ID: 166675007

Wikipedia 2016

Wikipedia: *Hilfe:Kategoriebaum*. https://de.wikipedia.org/w/index.php?title=Hilfe:Kategoriebaum&oldid=158994079. Version: Oktober 2016. – Page Version ID: 158994079

Wikipedia 2017a

Wikipedia: *Facettenklassifikation*. https://de.wikipedia.org/w/index.php?title=Facettenklassifikation&oldid=167102125. Version: Juli 2017. – Page Version ID: 167102125

Wikipedia 2017b

Wikipedia: *Help:Category*. https://en.wikipedia.org/w/index.php?title=Help:Category&oldid=795334035. Version: August 2017. – Page Version ID: 795334035

Wikipedia 2017c

Wikipedia: *Help:Files*. https://en.wikipedia.org/w/index.php?title=Help:Files&oldid=795468561. Version: August 2017. – Page Version ID: 795468561

Wikipedia 2017d

Wikipedia: *Hilfe:Kategorien*. https://de.wikipedia.org/w/index.php?title=Hilfe:Kategorien&oldid=166768419. Version: Juni 2017. – Page Version ID: 166768419

Wikipedia 2017e

Wikipedia: *Hilfe:Kategorien/VisualEditor*. https://de.wikipedia.org/w/index.php?title=Hilfe:Kategorien/VisualEditor&oldid=167917150. Version: August 2017. – Page Version ID: 167917150

Wikipedia 2017f

Wikipedia: *Hilfe:Normdaten*. https://de.wikipedia.org/w/index.php?title=Hilfe:Normdaten&oldid=165973255. Version: Mai 2017. – Page Version ID: 165973255

Wikipedia 2017g

Wikipedia: *Wikipedia:Categorization*. https://en.wikipedia.org/w/index.php?title=Wikipedia:Categorization&oldid=795794163. Version: August 2017. – Page Version ID: 795794163

Wikipedia 2017h

Wikipedia: *Wikipedia:Categorization - Wikipedia*. https://en.wikipedia.org/wiki/Wikipedia:Categorization. Version: 2017

Wikipedia 2017i

Wikipedia: *Wikipedia:Kategorien*. https://de.wikipedia.org/w/index.php?title=Wikipedia:Kategorien&oldid=162086918. Version: Januar 2017. – Page Version ID: 162086918

Wikipedia 2017j

Wikipedia: *Wikipedia:Namespace*. https://en.wikipedia.org/w/index.php?title=Wikipedia:Namespace&oldid=789974277. Version: Juli 2017. – Page Version ID: 789974277

Wikipedia 2017k

Wikipedia: *Wikipedia:What is an article?* https://en.wikipedia.org/w/index.php?title=Wikipedia:What_is_an_article%3F&oldid=786583897. Version: Juni 2017. – Page Version ID: 786583897

Wikipedia 2017l

Wikipedia: *Wikipedia:WikiProjekt Kategorien*. `https://de.wikipedia.org/w/index.php?title=Wikipedia:WikiProjekt_Kategorien&oldid=168269375`. Version: August 2017. – Page Version ID: 168269375

A. Verlage mit Open Access Journals für die Analyse der Beschriftungen. Referenz aus Kapitel 2.

Tabelle A.30: Verlage mit Open Access Journals für das NOA-Projekt

Verlagsname
Academic Press
American Physiological Society
American Psychological Association
BioMed Central
Blackwell Publishing Ltd
Copernicus GmbH
Copernicus Publications
Dove Medical Press
Elsevier Ltd
Faculdade de Odontologia de Bauru da Universidade de São Paulo
Frontiers Media S.A.
Frontiers Research Foundation
Gustav Fischer Verlag
Hindawi Publishing Corporation
Informa Healthcare
International Union of Crystallography
JMIR Publications Inc.
John Wiley and Sons Inc.
Mary Ann Liebert, Inc.
Medknow Publications
Nature Publishing Group
North-Holland Pub. Co
Oxford University Press
Pergamon Press
Publisher
Routledge
Royal Society of Chemistry
SAGE Publications
Seismological Society of China
Springer Berlin Heidelberg
Springer International Publishing
Springer Milan
Springer Netherlands
Springer Singapore
Springer US
Taylor & Francis
Wiley Subscription Services, Inc., A Wiley Company

B. Verwendete Codes

B.1. Code: Termextrakion

Referenz aus Kapitel 5.2.

```
import glob
import nltk
import codecs
import pprint
import treetaggerwrapper
import math

df = {}
tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')

def substantive_zaehlen(datei):
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    nouns_in_text = []
    sentences = nltk.sent_tokenize(text)
    sentences_tok = [nltk.word_tokenize(sent) for sent in sentences]
    nouns = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = [lemma for (word,pos,lemma) in tags2 if pos == "NN" or pos == "
            NNS" or pos == "VB"]
        for substantiv in nouns_from_sent:
            if substantiv not in nouns_in_text:
                nouns_in_text.append(substantiv)

    for n in nouns_in_text:
        df_n = df.get(n,0)
        df[n] = df_n + 1

for f in glob.glob("doi-cap-con/*.txt"):
    substantive_zaehlen(f)

def extract_term(datei,nr_of_docs):
    tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    sentences = nltk.sent_tokenize(text)
    sentences_tok = [nltk.word_tokenize(sent) for sent in sentences]
    nouns = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = [lemma for (word,pos,lemma) in tags2 if pos == "NN" or pos == "
            NNS" or pos == "VB"]
        nouns.extend(nouns_from_sent)
    fdist = nltk.FreqDist(nouns)

    for word in fdist:
        idf = 1.0 + math.log(float(nr_of_docs) / float(df[word]))
        fdist[word] = float(fdist[word]) / float(len(nouns)) * idf
```

```
    return fdist.most_common(15)

filelist = glob.glob("doi-cap-con/*.txt")
nr_of_docs = len(filelist)
for f in filelist[150:170]:
    terme = extract_term(f,nr_of_docs)
    pprint.pprint(f)
    pprint.pprint(terme)
    pprint.pprint("-----------------------------------")
```

B.2. Code: Termextrakion mit Kategoriemapping

Referenz aus den Kapiteln 6.2 und 6.3.

```
import glob
import nltk
import codecs
import pprint
import treetaggerwrapper
import math
import requests
from nltk.corpus import stopwords

filelist = glob.glob("doi-cap-con/*.txt")
nr_of_docs = len(filelist)
noun2cat = {}
catStoplist = stopwords.words('categories')
df = {}
tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')

def substantive_zaehlen(datei):
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    nouns_in_text = []
    sentences = nltk.sent_tokenize(text)
    sentences_tok = [nltk.word_tokenize(sent) for sent in sentences]
    nouns = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = [lemma for (word,pos,lemma) in tags2 if pos == "NN" or pos == "
            NNS" or pos == "VB"]
        for substantiv in nouns_from_sent:
            if substantiv not in nouns_in_text:
                nouns_in_text.append(substantiv)

    for n in nouns_in_text:
        df_n = df.get(n,0)
        df[n] = df_n + 1

for f in glob.glob("doi-cap-con/*.txt"):
    substantive_zaehlen(f)

def extract_term(datei,nr_of_docs):
    tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
    textfile = codecs.open(datei, "r", "utf8")
    text = textfile.read()
    textfile.close()
    sentences = nltk.sent_tokenize(text)
    sentences_tok = [nltk.word_tokenize(sent) for sent in sentences]
    nouns = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = [lemma for (word,pos,lemma) in tags2 if pos == "NN" or pos == "
            NNS" or pos == "VB"]
        nouns.extend(nouns_from_sent)#Te
    fdist = nltk.FreqDist(nouns)
```

```
    for word in fdist:
        idf = 1.0 + math.log(float(nr_of_docs) / float(df[word]))
        fdist[word] = float(fdist[word]) / float(len(nouns)) * idf

    return fdist.most_common(15)

def wiki_cats(term):
    cats = []

    response = requests.get(
        'https://en.wikipedia.org/w/api.php',
        params={
            'action': 'query',
            'format': 'json',
            'titles': term,
            'prop': 'categories',
            'clshow':'!hidden',
            'cllimit':'50'
        }
    ).json()

    for pageid in  response['query']['pages']:
        if len(term) >2 :
            if 'categories' in response['query']['pages'][pageid]:
                cats.extend([cat['title'] for cat in response['query']['pages'][pageid]['
                    categories'] if cat['ns'] == 14 and not cat['title'].startswith('
                    Category:Disambiguation') and cat['title'] not in catStoplist])

    return set(cats)

def collect_nouns(flist):
    global noun2cat
    for f in flist:
        nouns = extract_term(f, nr_of_docs)
        for (n,f) in nouns:
            noun2cat[n] = wiki_cats(n)

collect_nouns(filelist[150:170])

for f in filelist[150:170]:
    terme = extract_term(f,nr_of_docs)
    cats = set([c for (kw,f) in terme for c in noun2cat.get(kw)])
    pprint.pprint(f)
    #pprint.pprint(noun2cat)
    pprint.pprint(cats)
    pprint.pprint("----------------------------------------")
```

B.3. Code: Ranking der Kategorien

Referenz aus den Kapiteln 6.5, 7.1 und 8.1.

```
def intersection(a,b):
    i = 0
    for e in a:
        if e in b:
            i += 1
    return i

def wiki_cats_sort_rel(f):
    terme = extract_term(f,nr_of_docs)
    cats = set([c for (kw,f) in terme for c in noun2cat.get(kw)])
    extended = {}
    relnumber = {}
    for cat in cats:
        family = [cat]
        family.extend(wiki_cats(cat))
        extended[cat] = family
    #pprint.pprint(extended)
    for cat in cats:
        relsize = 0
        for cat2 in cats:
            if cat == cat2:
                continue
            if intersection(extended[cat],extended[cat2]) > 0:
                relsize +=1
        relnumber[cat] = relsize
    return sorted(relnumber.items(),key = lambda x:x[1],reverse=True)

for f in filelist[150:170]:
    pprint.pprint(f)
    cts = wiki_cats_sort_rel(f)
    pprint.pprint(cts)
    pprint.pprint("————————————————————————————————")
```

B.4. Code: Extraktion von Nominalphrasen

Referenz aus Kapitel 5.3.2.

```
import glob
import nltk
import codecs
import re
import treetaggerwrapper
import pprint
import math
from nltk.corpus import stopwords

swlist = stopwords.words('english')
tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
filelist = glob.glob("doi-cap-con/*.txt")
nr_of_words = 0
fdist = nltk.FreqDist()
fdist2 = nltk.FreqDist()

npmuster = [('NN','NN'),
            ('NP','NP'),
            ('NN','NP'),
            ('NP','NN'),
            ('JJ','NP'),
            ('JJ','NN')
           ]

for datei in filelist:
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    words = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        words = [lemma for (word,pos,lemma) in tags2 if pos[0]]
        word_pairs = []
        for i in range(len(tags2)-1) :
            t1 = tags2[i]
            t2 = tags2[i+1]
            if (t1.pos,t2.pos) in npmuster:
                l1 = t1.lemma
                l2 = t2.lemma
                if l1 not in swlist and l2 not in swlist  and len(l1) >2 and len(l2) >2:
                    word_pairs.append((l1,l2))

        nr_of_words += len(words)
        fdist.update(words)
        fdist2.update(word_pairs)
print("Anzahl Wortvorkommen (tokens): ", nr_of_words)
pprint.pprint(fdist2.most_common(100))
```

Nach Mustererkennung die Extraktion der Nominalphrasen

```
import glob
import nltk
import codecs
import pprint
```

```
import treetaggerwrapper

df = {}
tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
NPList = fdist2.keys()

def candidates(taglist):
    cand = []
    skip = False
    for i in range(len(taglist)-1) :
        if skip:
            skip = False
            continue
        skip = False
        l1 = taglist[i].lemma
        l2 = taglist[i+1].lemma
        if len(l1) >2 and len(l2) >2:
            if (l1,l2) in NPList:
                skip = True
                cand.append((l1,l2))

    return cand

def substantive_zaehlen(datei):
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    nouns_in_text = []
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = candidates(tags2)
        for substantiv in nouns_from_sent:
            if substantiv not in nouns_in_text:
                nouns_in_text.append(substantiv)

    for n in nouns_in_text:
        df_n = df.get(n,0)
        df[n] = df_n + 1

for f in glob.glob("doi-cap-con/*.txt"):
    substantive_zaehlen(f)

def extract_kw(datei,nr_of_docs):
    tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    nouns = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = candidates(tags2)
        nouns.extend(nouns_from_sent)
    fdist = nltk.FreqDist(nouns)
```

```
    for word in fdist:
        idf = 1.0 + math.log(float(nr_of_docs) / float(df[word]))
        fdist[word] = float(fdist[word]) / float(len(nouns)) * idf

    return fdist.most_common(8)

filelist = glob.glob("doi-cap-con/*.txt")
nr_of_docs = len(filelist)
for f in filelist[150:170]:
    keywords = extract_kw(f,nr_of_docs)
    pprint.pprint(f)
    pprint.pprint(keywords)
    pprint.pprint("-----------------------------------")
```

B.5. Code: Extraktion von Nominalphrasen mit Kategoriemapping.

Referenz aus Kapitel 7.

```
import glob
import nltk
import codecs
import pprint
import treetaggerwrapper
import math
import requests
from nltk.corpus import stopwords
import string
import re

filelist = glob.glob("doi-cap-con/*.txt")
nr_of_docs = len(filelist)
noun2cat = {}
catStoplist = stopwords.words('categories')
df = {}
tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
NPList = fdist2.keys()

def candidates(taglist):
    cand = []
    skip = False
    for i in range(len(taglist)-1) :
        if skip:
            skip = False
            continue
        skip = False
        l1 = taglist[i].lemma
        l2 = taglist[i+1].lemma
        if len(l1) >2 and len(l2) >2:
            if (l1,l2) in NPList:
                skip = True
                cand.append((l1,l2))
    return cand

def substantive_zaehlen(datei):
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    nouns_in_text = []
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = candidates(tags2)
        for substantiv in nouns_from_sent:
            if substantiv not in nouns_in_text:
                nouns_in_text.append(substantiv)

    for n in nouns_in_text:
        df_n = df.get(n,0)
        df[n] = df_n + 1


for f in glob.glob("doi-cap-con/*.txt"):
```

```
    substantive_zaehlen(f)

def extract_kw(datei,nr_of_docs):
    tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    nouns = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = candidates(tags2)
        nouns.extend(nouns_from_sent)
    fdist = nltk.FreqDist(nouns)

    for word in fdist:
        idf = 1.0 + math.log(float(nr_of_docs) / float(df[word]))
        fdist[word] = float(fdist[word]) / float(len(nouns)) * idf

    return fdist.most_common(8)

def wiki_cats(term):
    cats = []
    term_string = str(term)
    term_lower = term_string.lower()
    rechar = re.compile(r"['(),]")
    term_clear = rechar.sub("", term_lower)
    #pprint.pprint(term_clear)

    response = requests.get(
        'https://en.wikipedia.org/w/api.php',
        params={
            'action': 'query',
            'format': 'json',
            'titles': term_clear,
            'prop': 'categories',
            'clshow':'!hidden',
            'cllimit':'50'
        }
    ).json()

    for pageid in response['query']['pages']:
        if 'categories' in response['query']['pages'][pageid]:
            cats.extend([cat['title'] for cat in response['query']['pages'][pageid]['
                categories'] if cat['ns'] == 14 and not cat['title'].startswith('Category:
                Disambiguation') and not cat['title'].startswith('Category:Wikipedia
                articles incorporating') and cat['title'] not in catStoplist])
        return set(cats)

def collect_nouns(flist):
    global noun2cat
    for f in flist:
        nouns = extract_kw(f, nr_of_docs)
        for (n,f) in nouns:
            noun2cat[n] = wiki_cats(n)

collect_nouns(filelist[150:170])
```

```
filelist = glob.glob("doi-cap-con/*.txt")
nr_of_docs = len(filelist)
for f in filelist[150:170]:
    keywords = extract_kw(f,nr_of_docs)
    catsPair = set([c for (kw,f) in keywords for c in noun2cat.get(kw)])
    pprint.pprint(f)
    pprint.pprint(catsPair)
    pprint.pprint("-----------------------------------")
```

B.6. Code: Extraktion von Nominalphrasen und Termen

Referenz aus Kapitel 5.4.

```
import glob
import nltk
import codecs
import re
import treetaggerwrapper
import pprint
import math
from nltk.corpus import stopwords

swlist = stopwords.words('english')
tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
filelist = glob.glob("doi-cap-con/*.txt")
nr_of_words = 0
fdist = nltk.FreqDist()
fdist2 = nltk.FreqDist()

npmuster = [('NN','NN'),
            ('NP','NP'),
            ('NN','NP'),
            ('NP','NN'),
            ('JJ','NP'),
            ('JJ','NN')
           ]

for datei in filelist:
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    words = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        words = [lemma for (word,pos,lemma) in tags2 if pos[0]]
        word_pairs = []
        for i in range(len(tags2)-1) :
            t1 = tags2[i]
            t2 = tags2[i+1]
            if (t1.pos,t2.pos) in npmuster:
                l1 = t1.lemma
                l2 = t2.lemma
                if l1 not in swlist and l2 not in swlist  and len(l1) >2 and len(l2) >2:
                    word_pairs.append((l1,l2))

        nr_of_words += len(words)
        fdist.update(words)
        fdist2.update(word_pairs)
print("Anzahl Wortvorkommen (tokens): ", nr_of_words)
pprint.pprint(fdist2.most_common(100))
```

Nach Mustererkennung die Extraktion der Nominalphrasen und Terme

```
import glob
import nltk
import codecs
import pprint
```

```
import treetaggerwrapper

df = {}
tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
NPList = fdist2.keys()

def candidates(taglist):
    cand = []
    skip = False
    for i in range(len(taglist)-1) :
        if skip:
            skip = False
            continue
        skip = False
        l1 = taglist[i].lemma
        l2 = taglist[i+1].lemma
        if len(l1) >2 and len(l2) >2:
            if (l1,l2) in NPList:
                skip = True
                cand.append((l1,l2))
                if taglist[i].pos == 'NN' or taglist[i].pos == 'NP':
                    cand.append((l1))
            else:
                if taglist[i].pos == 'NN' or taglist[i].pos == 'VB' or taglist[i].pos == '
                    NNS':
                    cand.append(l1)
    w1 = taglist[-1]
    if w1.pos == 'NN'or w1.pos == 'VB' or w1.pos == 'NNS' or w1.pos == 'NP' or w1.pos == '
        NNP':
        cand.append(w1)

    return cand

def substantive_zaehlen(datei):
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    nouns_in_text = []
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = candidates(tags2)
        for substantiv in nouns_from_sent:
            if substantiv not in nouns_in_text:
                nouns_in_text.append(substantiv)

    for n in nouns_in_text:
        df_n = df.get(n,0)
        df[n] = df_n + 1

for f in glob.glob("doi-cap-con/*.txt"):
    substantive_zaehlen(f)

def extract_kw(datei,nr_of_docs):
    tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
```

```
    textfile.close()
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    nouns = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = candidates(tags2)
        nouns.extend(nouns_from_sent)
    fdist = nltk.FreqDist(nouns)

    for word in fdist:
        idf = 1.0 + math.log(float(nr_of_docs) / float(df[word]))
        fdist[word] = float(fdist[word]) / float(len(nouns)) * idf

    return fdist.most_common(15)

filelist = glob.glob("doi-cap-con/*.txt")
nr_of_docs = len(filelist)
for f in filelist[150:170]:
    keywords = extract_kw(f,nr_of_docs)
    pprint.pprint(f)
    pprint.pprint(keywords)
    pprint.pprint("----------")
```

B.7. Code: Extraktion von Nominalphrasen und Termen mit Kategoriemapping

Referenz aus Kapitel 8.

```
import glob
import nltk
import codecs
import pprint
import treetaggerwrapper
import math
import requests
from nltk.corpus import stopwords
import string
import re

filelist = glob.glob("doi-cap-con/*.txt")
nr_of_docs = len(filelist)
noun2cat = {}
catStoplist = stopwords.words('categories')
df = {}
tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
NPList = fdist2.keys()

def candidates(taglist):
    cand = []
    skip = False
    for i in range(len(taglist)-1) :
        if skip:
            skip = False
            continue
        skip = False
        l1 = taglist[i].lemma
        l2 = taglist[i+1].lemma
        if len(l1) >2 and len(l2) >2:
            if (l1,l2) in NPList:
                skip = True
                cand.append((l1,l2))
                if taglist[i].pos == 'NN' or taglist[i].pos == 'NP':
                    cand.append((l1))
            else:
                if taglist[i].pos == 'NN' or taglist[i].pos == 'VB' or taglist[i].pos == '
                    NNS':
                    cand.append(l1)
    w1 = taglist[-1]
    if w1.pos == 'NN'or w1.pos == 'VB' or w1.pos == 'NNS' or w1.pos == 'NP' or w1.pos == '
        NNP':
        cand.append(w1)

    return cand

def substantive_zaehlen(datei):
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    nouns_in_text = []
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    for sent in sentences_tok:
```

```
            tags = tagger.tag_text(sent,tagonly=True)
            tags2 = treetaggerwrapper.make_tags(tags);
            nouns_from_sent = candidates(tags2)
            for substantiv in nouns_from_sent:
                if substantiv not in nouns_in_text:
                    nouns_in_text.append(substantiv)

        for n in nouns_in_text:
            df_n = df.get(n,0)
            df[n] = df_n + 1

    for f in glob.glob("doi-cap-con/*.txt"):
        substantive_zaehlen(f)

    def extract_kw(datei,nr_of_docs):
        tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')

        textfile = codecs.open(datei, "r", "utf-8")
        text = textfile.read()
        textfile.close()

        sentences = nltk.sent_tokenize(text,language='english')
        sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
        nouns = []
        for sent in sentences_tok:
            tags = tagger.tag_text(sent,tagonly=True)
            tags2 = treetaggerwrapper.make_tags(tags);
            nouns_from_sent = candidates(tags2)
            nouns.extend(nouns_from_sent)
        fdist = nltk.FreqDist(nouns)

        for word in fdist:
            idf = 1.0 + math.log(float(nr_of_docs) / float(df[word]))
            fdist[word] = float(fdist[word]) / float(len(nouns)) * idf

        return fdist.most_common(15)

    def wiki_cats(term):
        cats = []
        term_string = str(term)
        term_lower = term_string.lower()
        rechar = re.compile(r"['()]")
        term_clear = rechar.sub("", term_lower)

        response = requests.get(
            'https://en.wikipedia.org/w/api.php',
            params={
                'action': 'query',
                'format': 'json',
                'titles': term_clear,
                'prop': 'categories',
                'clshow':'!hidden',
                'cllimit':'50'
            }
        ).json()

        for pageid in response['query']['pages']:
            if 'categories' in response['query']['pages'][pageid]:
```

```
            cats.extend([cat['title'] for cat in response['query']['pages'][pageid]['
                categories'] if cat['ns'] == 14 and not cat['title'].startswith('Category:
                Disambiguation') and not cat['title'].startswith('Category:Wikipedia
                articles incorporating') and cat['title'] not in catStoplist])
        return set(cats)

def collect_nouns(flist):
    global noun2cat
    for f in flist:
        nouns = extract_kw(f, nr_of_docs)
        for (n,f) in nouns:
            noun2cat[n] = wiki_cats(n)

collect_nouns(filelist[150:170])

filelist = glob.glob("doi-cap-con/*.txt")
nr_of_docs = len(filelist)
for f in filelist[150:170]:
    keywords = extract_kw(f,nr_of_docs)
    catsPair = set([c for (kw,f) in keywords for c in noun2cat.get(kw)])
    pprint.pprint(f)
    pprint.pprint(catsPair)
    pprint.pprint("-----------------------------------")
```

B.8. Code: Extraktion für Evaluierung

Referenz aus Kapitel 9.1.

```
import glob
import nltk
import codecs
import re
import treetaggerwrapper
import pprint
import math
from nltk.corpus import stopwords

swlist = stopwords.words('english')
tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
filelist = glob.glob("evalu-commons/*.txt")
nr_of_words = 0
fdist = nltk.FreqDist()
fdist2 = nltk.FreqDist()

npmuster = [('NN','NN'),
            ('NP','NP'),
            ('NN','NP'),
            ('NP','NN'),
            ('JJ','NP'),
            ('JJ','NN')
           ]

for datei in filelist:
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    words = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        words = [lemma for (word,pos,lemma) in tags2 if pos[0]]
        word_pairs = []
        for i in range(len(tags2)-1) :
            t1 = tags2[i]
            t2 = tags2[i+1]
            if (t1.pos,t2.pos) in npmuster:
                l1 = t1.lemma
                l2 = t2.lemma
                if l1 not in swlist and l2 not in swlist  and len(l1) >2 and len(l2) >2:
                    word_pairs.append((l1,l2))

        nr_of_words += len(words)
        fdist.update(words)
        fdist2.update(word_pairs)
print("Anzahl Wortvorkommen (tokens): ", nr_of_words)
pprint.pprint(fdist2.most_common(100))
```

Extraktion der Nominalphrasen und Terme aus den Evaluationsdatensätze

```
import glob
import nltk
import codecs
import pprint
```

```
import treetaggerwrapper

df = {}
tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
NPList = fdist2.keys()

def candidates(taglist):
    cand = []
    skip = False
    for i in range(len(taglist)-1) :
        if skip:
            skip = False
            continue
        skip = False
        l1 = taglist[i].lemma
        l2 = taglist[i+1].lemma
        if len(l1) >2 and len(l2) >2:
            if (l1,l2) in NPList:
                skip = True
                cand.append((l1,l2))
                if taglist[i].pos == 'NN' or taglist[i].pos == 'NP':
                    cand.append((l1))
            else:
                if taglist[i].pos == 'NN' or taglist[i].pos == 'VB' or taglist[i].pos == '
                    NNS':
                    cand.append(l1)
    w1 = taglist[-1]
    if w1.pos == 'NN'or w1.pos == 'VB' or w1.pos == 'NNS' or w1.pos == 'NP' or w1.pos == '
        NNP' :
        cand.append(w1)

    return cand

def substantive_zaehlen(datei):
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    nouns_in_text = []
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = candidates(tags2)
        for substantiv in nouns_from_sent:
            if substantiv not in nouns_in_text:
                nouns_in_text.append(substantiv)

    for n in nouns_in_text:
        df_n = df.get(n,0)
        df[n] = df_n + 1

for f in glob.glob("evalu-commons/*.txt"):
    substantive_zaehlen(f)

def extract_kw(datei,nr_of_docs):
    tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
```

```
    textfile.close()
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    nouns = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = candidates(tags2)
        nouns.extend(nouns_from_sent)
    fdist = nltk.FreqDist(nouns)

    for word in fdist:
        idf = 1.0 + math.log(float(nr_of_docs) / float(df[word]))
        fdist[word] = float(fdist[word]) / float(len(nouns)) * idf

    return fdist.most_common(15)

filelist = glob.glob("evalu-commons/*.txt")
nr_of_docs = len(filelist)
for f in filelist[0:48]:
    keywords = extract_kw(f,nr_of_docs)
    pprint.pprint(f)
    pprint.pprint(keywords)
    pprint.pprint("-----------------------------------")
```

B.9. Code: Extraktion für Evaluierung mit Kategoriemapping

Referenz aus Kapitel 9.3.

```
import glob
import nltk
import codecs
import pprint
import treetaggerwrapper
import math
import requests
from nltk.corpus import stopwords
import string
import re

filelist = glob.glob("evalu-commons/*.txt")
nr_of_docs = len(filelist)
noun2cat = {}
catStoplist = stopwords.words('categories')
df = {}
tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
NPList = fdist2.keys()
acronym ={"rifsv" : "rifamycin sv", "ge" : "GE23077", "rnap" : "rna polymerase", "rif" : "
    rifamycin", "ic50" : "half maximal inhibitory concentrations", "sor" : "sorangicin", "
    rnap-sor" : "rna polymerase sorangicin", "rif-resistant" : "rifamycin resistant",
          "rpo" : "rnap promoter open complex", "rna": "ribonucleic acid", "rnap-ge" : "rna
               polymerase GE23077", "rnap-rif" : "RNA polymerase", "mp" : "maximum
              parsimony", "ghr" : "growth hormone receptor", "sro" : "spermatophore
              receiving organ", "rrna" : "ribosomal rna", "ml" : "maximum likelihood", "tbr
              " : "tree bisection and reconnection", "pca" : "principle component analysis
              ", "pc" : "principal components", "bi" : "Bayesian inference", "pic" : "
              phylogenetically independent contrasts", "sro" : "spermatophore receiving
              organ", "mtdna" : "mitochondrial DNA", "rcrs" : "revised cambridge reference
              sequence", "pcr" : "polymerase chain reaction", "gtr" : "general time
              reversible","bpp" : "bayesian posterior probability", "mri" : "magnetic
              resonance imaging", "tet" : "tetrahedral", "om" : "outer membrane", "piv" : "
              particle image velocimetry ", "sem" : "scanning electron microscope", "rsegfp
              " : "reversibly switchable enhanced GFP", "gfp" : "green fluorescent protein
              ", "egfp" : "enhanced green fluorescent protein", "paa" : "polyacrylamide
              layers", "er" : "endoplasmic reticulum", "bar" : "Scale Bar" }

def candidates(taglist):
    cand = []
    skip = False
    for i in range(len(taglist)-1) :
        if skip:
            skip = False
            continue
        skip = False
        l1 = taglist[i].lemma
        l2 = taglist[i+1].lemma
        if len(l1) >2 and len(l2) >2:
            if (l1,l2) in NPList:
                skip = True
                cand.append((l1,l2))
                if taglist[i].pos == 'NN' or taglist[i].pos == 'NP':
                    cand.append((l1))
            else:
                if taglist[i].pos == 'NN' or taglist[i].pos == 'VB' or taglist[i].pos == '
                    NNS':
```

```
                    cand.append(l1)
        w1 = taglist[-1]
        if w1.pos == 'NN' or w1.pos == 'VB' or w1.pos == 'NNS' or w1.pos == 'NP':
            cand.append(w1)

        return cand

def substantive_zaehlen(datei):
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    nouns_in_text = []
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = candidates(tags2)
        for substantiv in nouns_from_sent:
            if substantiv not in nouns_in_text:
                nouns_in_text.append(substantiv)

    for n in nouns_in_text:
        df_n = df.get(n,0)
        df[n] = df_n + 1

for f in glob.glob("evalu-commons/*.txt"):
    substantive_zaehlen(f)

def extract_kw(datei,nr_of_docs):
    tagger = treetaggerwrapper.TreeTagger(TAGLANG='en')
    textfile = codecs.open(datei, "r", "utf-8")
    text = textfile.read()
    textfile.close()
    sentences = nltk.sent_tokenize(text,language='english')
    sentences_tok = [nltk.word_tokenize(sent,language='english') for sent in sentences]
    nouns = []
    for sent in sentences_tok:
        tags = tagger.tag_text(sent,tagonly=True)
        tags2 = treetaggerwrapper.make_tags(tags);
        nouns_from_sent = candidates(tags2)
        nouns.extend(nouns_from_sent)
    fdist = nltk.FreqDist(nouns)

    for word in fdist:
        idf = 1.0 + math.log(float(nr_of_docs) / float(df[word]))
        fdist[word] = float(fdist[word]) / float(len(nouns)) * idf

    return fdist.most_common(15)

def wiki_cats(term):
    cats = []
    term_string = str(term)
    term_lower = term_string.lower()
    rechar = re.compile(r"['(),]")
    term_clear = rechar.sub("", term_lower)

    if term_clear in acronym:
        term_clear = acronym.get(term_clear)
```

```
    response = requests.get(
        'https://en.wikipedia.org/w/api.php',
        params={
            'action': 'query',
            'format': 'json',
            'titles': term_clear,
            'prop': 'categories',
            'clshow':'!hidden',
            'cllimit':'50'
        }
    ).json()

    for pageid in response['query']['pages']:
        if 'categories' in response['query']['pages'][pageid]:
            cats.extend([cat['title'] for cat in response['query']['pages'][pageid]['
                categories'] if cat['ns'] == 14 and not cat['title'].startswith('Category:
                Disambiguation') and not cat['title'].startswith('Category:Wikipedia
                articles incorporating') and cat['title'] not in catStoplist])
        return set(cats)

def collect_nouns(flist):
    global noun2cat
    for f in flist:
        nouns = extract_kw(f, nr_of_docs)
        for (n,f) in nouns:
            noun2cat[n] = wiki_cats(n)

collect_nouns(filelist[0:48])

filelist = glob.glob("evalu-commons/*.txt")
nr_of_docs = len(filelist)
for f in filelist[0:48]:
    keywords = extract_kw(f,nr_of_docs)
    catsPair = set([c for (kw,f) in keywords for c in noun2cat.get(kw)])
    pprint.pprint(f)
    pprint.pprint(catsPair)
    pprint.pprint("-----------------------------------")
```

B.10. Code: Ranking der Kategorien für Evaluierung

Referenz aus Kapitel 9.3.

```
import requests

def wiki_cats(term):
    cats = []
    response = requests.get(
        'https://en.wikipedia.org/w/api.php',
        params={
            'action': 'query',
            'format': 'json',
            'titles': term,
            'prop': 'categories',
            'clshow':'!hidden',
            'cllimit':'50'
        }
    ).json()

    for pageid in  response['query']['pages']:
        cats.extend([cat['title'] for cat in response['query']['pages'][pageid]['categories
            '] if cat['ns'] == 14])

    return cats

def intersection(a,b):
    i = 0
    for e in a:
        if e in b:
            i += 1
    return i

def wiki_cats_sort_rel(f):
    keywords = extract_kw(f,nr_of_docs)
    cats = set([c for (kw,f) in keywords for c in noun2cat.get(kw)])
    extended = {}
    relnumber = {}
    for cat in cats:
        family = [cat]
        family.extend(wiki_cats(cat))
        extended[cat] = family
    #pprint.pprint(extended)
    for cat in cats:
        relsize = 0
        for cat2 in cats:
            if cat == cat2:
                continue
            if intersection(extended[cat],extended[cat2]) > 0:
                relsize +=1
        relnumber[cat] = relsize
    return sorted(relnumber.items(),key = lambda x:x[1],reverse=True)

for f in filelist[0:48]:
    pprint.pprint(f)
    cts = wiki_cats_sort_rel(f)
    pprint.pprint(cts)
    pprint.pprint("----------")
```

C. Wikimedia Commons-Links der Evaluationsabbildungen

Links aller Abbildungen für Evaluation aus Kapitel 9.

Link bei Wikimedia Commons:

1. https://commons.wikimedia.org/wiki/File:Representatives_of_ceratioid_families.jpg
2. https://commons.wikimedia.org/wiki/File:Malleus_of_Golden_moles.jpg
3. https://commons.wikimedia.org/wiki/File:Mandibles_of_Golden_moles.jpg
4. https://commons.wikimedia.org/wiki/File:Skulls_of_Golden_moles.jpg
5. https://commons.wikimedia.org/wiki/File:Love-darts.png
6. https://commons.wikimedia.org/wiki/File:Phylogeny-and-antiquity-of-M-macrohaplogroup-inferred-from-complete-mt-DNA-sequence-of-Indian-1471-2148-5-26-1.jpg
7. https://commons.wikimedia.org/wiki/File:Trochulus_villosus.jpg
8. https://commons.wikimedia.org/wiki/File:Pomacea_paludosa_eggs_maturation.jpg
9. https://commons.wikimedia.org/wiki/File:Neogastropod-phylogenetic-relationships-based-on-entire-mitochondrial-genomes-1471-2148-9-210-1.jpg
10. https://commons.wikimedia.org/wiki/File:Neogastropod-phylogenetic-relationships-based-on-entire-mitochondrial-genomes-1471-2148-9-210-2.jpg
11. https://commons.wikimedia.org/wiki/File:Neogastropod-phylogenetic-relationships-based-on-entire-mitochondrial-genomes-1471-2148-9-210-3.jpg
12. https://commons.wikimedia.org/wiki/File:Neogastropod-phylogenetic-relationships-based-on-entire-mitochondrial-genomes-1471-2148-9-210-4.jpg
13. https://commons.wikimedia.org/wiki/File:Neogastropod-phylogenetic-relationships-based-on-entire-mitochondrial-genomes-1471-2148-9-210-5.jpg
14. https://commons.wikimedia.org/wiki/File:Systematic-comparison-and-reconstruction-of-sea-urchin-(Echinoidea)-internal-anatomy-a-novel-1741-7007-6-33-1.jpg
15. https://commons.wikimedia.org/wiki/File:Systematic-comparison-and-reconstruction-of-sea-urchin-(Echinoidea)-internal-anatomy-a-novel-1741-7007-6-33-2.jpg
16. https://commons.wikimedia.org/wiki/File:Evolution-and-diversity-of-Rickettsia-bacteria-1741-7007-7-6-1.jpg
17. https://commons.wikimedia.org/wiki/File:Evolution-and-diversity-of-Rickettsia-bacteria-1741-7007-7-6-2.jpg
18. https://commons.wikimedia.org/wiki/File:Rooting-the-tree-of-life-by-transition-analyses-1745-6150-1-19-10.jpg
19. https://commons.wikimedia.org/wiki/File:Rooting-the-tree-of-life-by-transition-analyses-1745-6150-1-19-11.jpg
20. https://commons.wikimedia.org/wiki/File:Rooting-the-tree-of-life-by-transition-analyses-1745-6150-1-19-2.jpg
21. https://commons.wikimedia.org/wiki/File:Rooting-the-tree-of-life-by-transition-analyses-1745-6150-1-19-3.jpg
22. https://commons.wikimedia.org/wiki/File:Rooting-the-tree-of-life-by-transition-analyses-1745-6150-1-19-4.jpg
23. https://commons.wikimedia.org/wiki/File:Rooting-the-tree-of-life-by-transition-analyses-1745-6150-1-19-5.jpg
24. https://commons.wikimedia.org/wiki/File:Rooting-the-tree-of-life-by-transition-analyses-1745-6150-1-19-6.jpg
25. https://commons.wikimedia.org/wiki/File:Rooting-the-tree-of-life-by-transition-analyses-1745-6150-1-19-7.jpg
26. https://commons.wikimedia.org/wiki/File:Rooting-the-tree-of-life-by-transition-analyses-1745-6150-1-19-8.jpg
27. https://commons.wikimedia.org/wiki/File:Rooting-the-tree-of-life-by-transition-analyses-1745-6150-1-19-9.jpg
28. https://commons.wikimedia.org/wiki/File:Notonecta_glauca_Skin_SEM_01.jpg
29. https://commons.wikimedia.org/wiki/File:RsEGFP2-enables-fast-RESOLFT-nanoscopy-of-living-cells-elife00248f001.jpg

30. https://commons.wikimedia.org/wiki/File:RsEGFP2-enables-fast-RESOLFT-nanoscopy-of-living-cells-elife00248f002.jpg
31. https://commons.wikimedia.org/wiki/File:RsEGFP2-enables-fast-RESOLFT-nanoscopy-of-living-cells-elife00248f003.jpg
32. https://commons.wikimedia.org/wiki/File:RsEGFP2-enables-fast-RESOLFT-nanoscopy-of-living-cells-elife00248fs001.jpg
33. https://commons.wikimedia.org/wiki/File:RsEGFP2-enables-fast-RESOLFT-nanoscopy-of-living-cells-elife00248fs002.jpg
34. https://commons.wikimedia.org/wiki/File:RsEGFP2-enables-fast-RESOLFT-nanoscopy-of-living-cells-elife00248fs003.jpg
35. https://commons.wikimedia.org/wiki/File:RsEGFP2-enables-fast-RESOLFT-nanoscopy-of-living-cells-elife00248fs004.jpg
36. https://commons.wikimedia.org/wiki/File:RsEGFP2-enables-fast-RESOLFT-nanoscopy-of-living-cells-elife00248fs005.jpg
37. https://commons.wikimedia.org/wiki/File:RsEGFP2-enables-fast-RESOLFT-nanoscopy-of-living-cells-elife00248fs006.jpg
38. https://commons.wikimedia.org/wiki/File:RsEGFP2-enables-fast-RESOLFT-nanoscopy-of-living-cells-elife00248fs007.jpg
39. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450f001.jpg
40. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450f002.jpg
41. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450f003.jpg
42. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450f004.jpg
43. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450f005.jpg
44. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450f006.jpg
45. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450f007.jpg
46. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450f008.jpg
47. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450f009.jpg
48. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450f010.jpg
49. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450fs001.jpg
50. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450fs002.jpg
51. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450fs003.jpg
52. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450fs004.jpg
53. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450fs005.jpg
54. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450fs006.jpg
55. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450fs007.jpg
56. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450fs008.jpg
57. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450fs009.jpg
58. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450fs010.jpg
59. https://commons.wikimedia.org/wiki/File:GE23077-binds-to-the-RNA-polymerase-i-and-i%2B1-sites-and-prevents-the-binding-of-initiating-elife02450fs011.jpg

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die eingereichte Masterarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzen Werken wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

_____

Ort, Datum, Frieda Josi