



Entwicklung einer automatisierten Wiki für Argumentnetzwerke

Bachelorarbeit

von

Stephan Linzbach

aus

Düsseldorf

vorgelegt am

Lehrstuhl für Rechnernetze

Prof. Dr. Martin Mauve

Heinrich-Heine-Universität Düsseldorf

September 2019

Betreuer:

Alexander Schneider, M. Sc.

Danksagung

A lot of people supported me during my work on this thesis to whom I wish to express my gratitude.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
1 Einleitung	1
1.1 Verwandte Arbeiten	1
1.2 Struktur	2
2 Genutzte <i>Frameworks</i> und <i>Libraries</i>	5
2.1 Django	5
2.2 Chart.js	5
2.3 Bootstrap	6
3 Verwendete <i>application programming interfaces</i>	7
3.1 IBM Cloud <i>Natural Language Understanding</i>	7
3.2 IBM Cloud <i>Text to Speech</i>	8
3.3 D-BAS GraphQL-API v2	9
3.4 Wikipedia REST-API v1	10
3.5 Event Registry	11
3.6 Verworfenen <i>API</i>	13
4 Berechnete Indices	15
4.1 Lesbarkeitindices	15
4.1.1 <i>The Flesch Reading Ease Index</i>	16
4.1.2 <i>Automated Readability Index</i>	17
4.1.3 Fazit	17
4.2 Positions und Argument Evaluierung	19
4.2.1 Grundlagen	19
4.2.2 Bewertung von Argumenten	21
4.2.3 Automatisierungs Entscheidungen	26

4.3	Index bzgl. der akademischen Ausrichtung des Argumentes	27
5	Implementation	29
5.1	Themenübersicht	29
5.2	Diskussionsübersicht	30
5.3	Prämissen- und Konklusionübersicht	31
5.4	Suchseite	32
5.5	Argumentübersicht	33
5.5.1	Überschrift	33
5.5.2	Inhaltsverzeichnis	33
5.5.3	Einordnungen	33
5.5.4	Darstellung im Kontext der Netzwerke	35
5.5.5	Sammlung der erzeugten Hintergrundinformationen	36
5.6	Nachrichtenübersichtsseite	37
5.7	Kategorieübersichtsseite	37
6	Evaluation	39
6.1	Bewertung der Funktionalitäten	41
7	Zusammenfassung	43
7.1	Ausblick	43
	Literaturverzeichnis	45

Abbildungsverzeichnis

3.1	Beispielhaft gewähltes Argument zur Veranschaulichung	10
3.2	Darstellung der Information zu 'pedestrian'	10
3.3	Darstellung der ambivalenten Information zu 'zone'	10
3.4	Darstellung zur zusammengesetzten Information zu 'pedestrian zone'	11
4.1	Beschreibung der Indizeausprägung des Flesch Reading Ease Index. [fre] . .	16
4.2	Beschreibung der Indizeausprägung des Automated Reading Index. [ari] . .	17
4.3	Die Auswertungsergebnisse der beiden Formeln in Bezug auf das angespro- chene Argument	18
4.4	Beispiel der Bewertung eines Argumentes	19
4.5	Skizze eines atomaren pro-Argumentes	20
4.6	Skizze eines atomaren contra-Argumentes	20
4.7	Skizze eines verbundenen pro-Argumentes.	21
4.8	Skizze eines verbundenen con-Argumentes.	21
4.9	Skizze eines vereinenden Argumentes dar.	21
4.10	Beschreibt die Entwicklung von Prämissenakzeptanz	22
4.11	Skizziert die Veränderung der Argumentsakzeptanz	23
4.12	Skizziert die Veränderung der Akzeptanz unter einem pro-konvergenten Ar- gument.	24
4.13	Skizziert Wertentwicklung der Konklusionsakzeptanz unter einem con-konvergenten Argument	25
4.14	Skizziert das Ergebnis einer Verrechnung eines verbindenden Argumentes . .	25
4.15	Skizziert das Ergebnis der Rechnung einer Undercut-Bewertung	26
5.1	Diskussionsthemen mit Information auf der Landingpage	29
5.2	Einzelne Position mit graphischer Darstellung des Akzeptanzscore	30
5.3	Erklärung zum akzeptanz Score	30
5.4	Einzelnes Argument aufgelistet unter einer Position	30

5.5	Übersichtsseite einer einzelnen Prämisse	31
5.6	Übersichtsseite einer einzelnen Konklusion	32
5.7	Darstellung einer eindeutig von Wikipedia erhaltenen Information.	36
5.8	Anzeige der Schlüsselwörter	36
6.1	Stellt Taxonomie zur Bewertung von Argumentationen dar	39
6.2	Stellt eine unpassende Wikipedia Recherche dar.	41

Kapitel 1

Einleitung

Politischer Diskurs und allgemeine gesellschaftliche Diskussionen finden immer häufiger im Internet statt. Um die Ergebnisse dieser Diskurse zu verbessern kann die Verbesserung von Diskussionsmöglichkeiten im Internet also nicht umgangen werden, da dieses die Fähigkeit besitzt, Zeit- und Raumdifferenzen zwischen den Gesprächspartnern zu kompensieren und somit Diskussionen zwischen Gesprächspartnern zu ermöglichen. Zusätzlich kann jeder Teilnehmer über das Internet Hintergrundwissen recherchieren und eine eigene Meinung zu Themen bilden. Diese beiden großen Vorteile werden durch Missinformationskampagnen, Filterblasen und generelle Strukturlosigkeit von Informationen, insbesondere in online geführter Diskussionen, angegriffen. Der Entwurf eines solchen automatisierten Wikis ist die Aufgabe der vorliegenden Arbeit. Zur Umsetzung dieser Aufgabe wurde, mit zahlreichen Methoden versucht zusätzliche Informationen zu strukturierten Argumenten zu generieren und eine übersichtliche und leicht verständliche Webseite aufzusetzen, die beim Untersuchen und Recherchieren von Argumenten hilft.

1.1 Verwandte Arbeiten

Es gibt mehrere *Mass Argumentation Tools*, bei denen zusätzliche Informationen zu den aufgelisteten Argumenten angezeigt werden. Dies sind Debatepedia , Debategraph ¹, Trut-

¹<https://debategraph.org/home>

hmapping² und ARGNET³. Debatepedia bietet zu den Diskussionen einen *Background and context* Text. Dieser wird jedoch nicht automatisch erzeugt, sondern wird von den Teilnehmern erstellt und verwaltet. Bei Debategraph bietet am rechten Rand der Seite ausführliche mit Zitaten hinterlegte Informationen zu jedem Argument. Das Verständnis der Seite und der Bedienung bedarf jedoch einem gewissen Level an *ICT skills*. Truthmapping bietet als Hintergrundinformationen nur einen Vote-Score und ein Reiter mit den angegebenen Quellen die von dem Argument verwendet wurden. ARGNET bietet die meisten automatisch angezeigten *Features*, hier wird eine Argumentbelastbarkeit berechnet, eine Erklärung die auf der Diskussion basiert wird erzeugt und ein Wert der die Widersprüchlichkeit eines Argumentes beziffert wird dargestellt [AG13]. Bei allen *Tools* werden Argumente angelegt, in Debatepedia werden nur pro und contra Argumente zu einem Thema zugelassen. Außerdem sind die erzeugten Hintergrundinformationen nur im ARGNET automatisch. Die beiden berechneten Scores benötigen jedoch zusätzliche Userangaben um automatisch errechnet zu werden, außerdem basiert die erzeugte Erklärung auch auf den Glaubwürdigkeitswert, der von Nutzern angegeben werden muss. Ist so ein Score nicht gegeben, können die hier verwendeten Werte und Hilfestellungen nicht erzeugt werden.

Des Weiteren gibt es Forschung, die sich mit dem automatischen Erzeugen von ganzen Wikipediaartikeln auseinander setzt, wie in diesem Konferenzessay [PJL18] beschrieben. Eine Wiki Umsetzung in dieser Art kann jedoch nicht das Ziel dieser Arbeit sein, da dies den Umfang der Arbeit überschreiten würde.

1.2 Struktur

Im zweiten Kapitel werden die für die Implementation genutzten Werkzeuge vorgestellt und eine kurze Einführung in ihren Nutzen und Spezifikationen gegeben. Das darauffolgende Kapitel stellt ausführlich die genutzten Programmierschnittstellen dar und diskutiert die von ihnen angefragten Daten und die zusätzlich angebotenen Funktionalitäten. Das vierte Kapitel befasst sich mit kalkulierten Indizes, die die Kategorisierung und Einschätzung der Argumente unterstützen sollen. Im anschließenden Kapitel wird die Gesamtstruktur der Webanwendung dargestellt, zusätzlich werden ihre Nutzungs- und Designentscheidung diskutiert. Das sechste Kapitel stellt eine Evaluation der Ergebnisse dar. Im letzten Kapitel wird ein Rück-

²<https://www.truthmapping.com/>

³nicht auffindbar

blick auf die Arbeit getätigt, indem diese zusammenfassend dargestellt wird. Des Weiteren werden denkbare Erweiterungen und zukünftig zu Verbessern vorgeschlagen.

Kapitel 2

Genutzte *Frameworks* und *Libraries*

2.1 Django

Django ist ein *highlevel web framework* welches für die Programmierung in Python entwickelt wurde [dja]. Sein Zweck ist es so viele generalisierbare Aufgaben im Zusammenhang mit dem Entwurf des *Back-Ends* von Internetseiten zu erleichtern wie möglich [AH13]. Der Aufbau der *Webapp* ist nach dem *Model View Template*-Pattern (kurz. MVT) aufgebaut. Django nutzt *Object-relational Mapping* (kurz. ORM), dies bietet die Funktionalitäten das die Datenbank auf Grundlage von *model* Klassen strukturiert ist. Außerdem können die genutzten Daten verschiedener Seitenansichten mithilfe der Views spezifiziert werden. Zur Darstellung der Seite ist die Einbindung von Templates gedacht, welche in der Arbeit mit HTML-5 programmiert sind.

2.2 Chart.js

Chart.js ist ein *open source librarie*, mit dessen Hilfe acht verschiedene Diagrammartent erzeugt und in HTML-5 *Canvases* eingebettet werden können [cjsb].

1. Liniendiagramm
2. Säulen-/Balkendiagramm

3. Netzdiagramm
4. Torten-/Ringdiagramm
5. Polardiagramm
6. Blasendiagramm

Chartjs kann entweder bei den *Github releases* heruntergeladen werden oder per *content delivery network* (kurz. CDN) in die Webanwendung eingebaut werden, hier wurde letztere Möglichkeit angewendet [cjsa].

2.3 Bootstrap

Bootstrap ist ein *open source toolkit*. Bootstrap bietet die Funktionalität, dass HTML *Tags* vordesign sind und so schon 'unbearbeitetes' HTML visuell ansprechender und übersichtlicher ist. Des Weiteren können durch Bootstrap kleinere Seitenstrukturierungselemente eingebaut werden wie Dropdowns oder Collapsibles, ohne zusätzlichen *Javascriptcode* zu erzeugen. Außerdem sind schon vorgefertigte CSS Designs zugänglich. So kann der Vorgang des *front-end* Designs von Webseiten beschleunigt werden. Auch Bootstrap wurde über ein CDN eingebuden.

Kapitel 3

Verwendete *application programming interfaces*

Um die Automatisierung des Wikis sicherzustellen, wurden die Dienste von fünf verschiedenen API's in Anspruch genommen. Die Nutzung einer sechsten *API* wurde verworfen. In dem folgenden Abschnitt wird vorerst genauer auf die verwendeten *APIs* eingegangen, ihre Rahmenbedingungen erläutert und ihre Nutzung dargestellt. Im letzten Teil des Kapitels wird die verworfene *API* dargestellt und die Gründe, die zum Abbruch der Arbeit mit ihr geführt haben erläutert.

3.1 IBM Cloud *Natural Language Understanding*

Die hier genutzte IBM API bietet die Möglichkeit verschiedene Merkmale eines Textes zu analysieren. Dies sind [ibma]:

- Kategorien
- Konzepte
- Emotionen
- Entitäten

- Schlüsselwörter
- Metadaten
- Relationen
- Semantische Rollen
- Meinungspolarität
- Syntax (experimentell)

Es wurde sich gegen die Verwendung der Merkmale semantische Rollen, Syntax, Entitäten und Metadaten entschieden. Bei semantischen Rollen wurde kein unbedingter Mehrwert im Bezug auf das Verstehen des Argumentes gesehen. Das Merkmal Syntax ist als experimentell markiert und bietet, deshalb kaum verlässliche Antworten. Entitäten werden von einem anderen Teil der Implementation vollständig erkannt. Metadaten finden hier keine Verwendung, da der Schnittstelle einzelne Texte zugesandt werden, die keinen Rückschluss auf relevante Metadaten zulassen. Relationen geben ebenfalls keine relevanten, weil relative unbelastbare Daten zurück um eine Verknüpfung oder Einschätzung der Argumente zu unterstützen. Bei der Implementierung wurde das *Python Software Development Kit* (kurz. SDK) verwendet [ibmc]. Eine Verbindung kann durch das Erstellen eines *NaturalLanguageUnderstandingVI* Objektes mit Übergabe des *API keys*, der URL zur Schnittstelle, abhängig vom Ort der Service-Instanz und einem Datum zur Spezifizierung der genutzten Version hergestellt werden. Die verwendete SDK verwaltet nun das *Identity and Access Management-Token* (kurz. IAM-Token), jetzt können unter Verwendung des genannten IAM-Tokens Anfragen an die Schnittstelle gestellt werden, ohne abermaliges senden des *API keys* [ibma]. Auf die empfangenden Daten und deren Verwendung wird im Laufe der Arbeit eingegangen.

3.2 IBM Cloud *Text to Speech*

Die hier genutzte *API* verwendet die IBM sprachsynthese Anwendung, und wird dafür genutzt Text in Sprache umzuwandeln. Hier wird nur die englische Sprache mit einer Stimme beispielhaft angewendet. Jedoch bietet die *API* die Möglichkeit eine Vielzahl von Sprachen, Dialekten und Stimmen zu immitieren. Für die anwendbaren Sprachen sind immer

mindestens eine Männer oder Frauen Stimme nutzbar, was für eventuelle Erweiterungen dieser Funktionalität von Interesse wäre [ibmb]. Die *API* bietet neben der Verwendung einer *websocket* auch die Möglichkeit sie nach dem *Representational State Transfer*-Paradigma (kurz. REST) zu verwenden. In diesem Fall wurde eine Verbindung mit der Schnittstelle, mithilfe der *Python SDK* [ibmc] aufgebaut. Hier wurde sich entschieden die Daten abzuspeichern, da eine kostenfreie Nutzung der Dienste nicht genug Anfragen für die angestrebte Nutzung zulässt. So kann über mehrere Monate hinweg diese Ressourcenlücke geschlossen werden. Jedes englischsprachige Argument wird zur Konvertierung, von Text zu Sprache, an die Schnittstelle gesendet und als mp3-Format im , vom *MEDIA URL* und *MEDIA ROOT* definierten, Ordner abgespeichert. Der benötigte Speicherplatz pro Tonaufnahmen liegt zwischen 408 bis 38 kb. 144 Tondateien benötigen zur Zeit einen Speicherplatz von 22.2 MB was einer durchschnittlichen Größe von ca. 155 kB entspricht. Bei heutigen Speicherkapazitäten sollte diese Menge jedoch keine Probleme darstellen.

3.3 D-BAS GraphQL-API v2

Ist die für das D-BAS implementierte *API* [TK]. Die v2 nutzt Facebooks GraphQL [gql], hier gibt es nicht wie bei einer *API*, die nach dem REST-Paradigma entworfen ist mehrere Endpunkte, sondern nur einen dem Anfragen gesendet werden können, in Form von Queries. Diese Queries müssen vom angefragten Server erst entschlüsselt werden. Über diese Schnittstelle werden alle im hier entwickelten Wiki angezeigten, Argumente, Themen und Zusatzinformationen, wie zum Beispiel die Abstimmungsergebnisse einzelner Argumente, oder die Sprache, in welcher die Argumente verfasst sind, sowie die einzelnen Infotexte zu den Diskussionsthemen entgegen genommen. Die Anfrage an die *API* wurde unter Nutzung des *Python Packages requests* [req] durchgeführt. Es wurden alle Diskussionsthemen und alle in diesen enthaltenen Argumente angefragt, aufgeteilt nach Prämisse und Konklusion. Außerdem wurden Einstellungsdatum (verwendet für die Nachrichtenanwendung) und Sprachinformationen angefragt.

3.4 Wikipedia REST-API v1

Bei dieser *API* wurde ausschließlich der Endpunkt *Page content* genutzt. Dort genauer der Zusatzparameter *summary*. Die Anfrage mit den beiden Endpunktspezifikatoren und einem Suchwort wurde unter Verwendung von *Requests* [req] an die Schnittstelle gesendet. Die an die *API* gesendeten Suchwörter werden wie folgt ermittelt. Jedes Wort aller englischen Textinhalten steht zur Anfrage bei Wikipedia aus. Um diese Menge zu verkleinern werden, die einzelnen Wörter vorher mit einem Lexikon abgeglichen, welches die 500 meist genutzten englischen Wörter enthält [fwe], so kann vermieden werden, dass Wörter wie 'a' oder 'the' gesucht werden. Kommt ein Wort eher selten vor, so wird bezüglich dieses Wortes eine Anfrage an den oben angegebenen Endpunkt gesendet. Im Arbeitsprozess ist aufgefallen, dass viele der unklaren Bezeichnungen nicht aus einzelnen Wörtern bestehen. Deshalb wird ein Wort welches nicht häufig genutzt wird mit dem Nachfolgewort zusammen an die Schnittstelle gesendet um die gemeinsame Bedeutung anzeigen zu können. Ein Beispiel für eine erfolgreiche Anwendung der Implementierung ist das Argument 3.1.

Regarding that, **then we will have more money to expand out pedestrian zone** **strengthens the claim that** **reducing the number of street festivals can save up to \$50.000 a year** 🔊

Abbildung 3.1: Beispielhaft gewähltes Argument zur Veranschaulichung

Mit der eindeutigen Information 3.2 und der ambivalenten Information zu 3.3.

for pedestrian
Pedestrian
person traveling on foot
A pedestrian is a person travelling on foot, whether walking or running. In some communities, those travelling using tiny wheels such as roller skates, skateboards, and scooters, as well as wheelchair users are also included as pedestrians. In modern times, the term usually refers to someone walking on a road or pavement, but this was not the case historically.

Abbildung 3.2: Darstellung der Information zu 'pedestrian'

for zone
Zone

Abbildung 3.3: Darstellung der ambivalenten Information zu 'zone'

Besonders wichtig ist hier, dass 'zone' auf einen uneindeutigen Wikipedia Artikel verweist. Somit ist dem Nutzer weniger geholfen, da dieser den hier passend gemeinten Sinn des Wortes 'zone' herausfiltern müsste. Es kann mithilfe des Kontextes jedoch ein eindeutiger Wikipedia Artikel gefunden werden der für den kombinierten Schlüsselbegriff 3.4 existiert.

for pedestrian zone
 Pedestrian zone
 Area of a city or town reserved for pedestrian-only use
 Pedestrian zones are areas of a city or town reserved for pedestrian-only use and in which most or all automobile traffic may be prohibited. Converting a street or an area to pedestrian-only use is called pedestrianisation. Pedestrianisation usually aims to provide better accessibility and mobility for pedestrians, to enhance the amount of shopping and other business activities in the area and/or to improve the attractiveness of the local environment in terms of aesthetics, air pollution, noise and crashes involving motor vehicle with pedestrians. However, pedestrianisation can sometimes lead to reductions in business activity, property devaluation, and displacement of economic activity to other areas. In some cases traffic in surrounding areas may increase, due to displacement, rather than substitution of car traffic. Nonetheless, pedestrianisation schemes are often associated with significant drops in local air and noise pollution, accidents, and frequently with increased retail turnover and increased property values locally. A car-free development generally implies a large scale pedestrianised area that relies on modes of transport other than the car, while pedestrian zones may vary in size from a single square to entire districts, but with highly variable degrees of dependence on cars for their broader transport links.

Abbildung 3.4: Darstellung zur zusammengesetzten Information zu 'pedestrian zone'

Zu beachten ist, dass es bei dem zweit Wort nicht geprüft wird, ob es sich um ein selten genutztes, oder ein häufig genutztes Wort handelt.

3.5 Event Registry

Event Registry bietet unter anderem die Möglichkeit Nachrichtenartikel u.ä. sowie Events aus der gesamten Welt mit einer großen Varietät von Filterfunktionen zu suchen und anzufragen. Zur Verwendung der Event Registry Schnittstelle in die Webanwendung wurde die angebotenen *Python SDK* genutzt [erp]. Um Anfragen stellen zu dürfen, wird auch hier die Autorisierung durch Angabe eines *Api-Key* geprüft. Vorerst werden nur Nachrichtenartikel gesucht. Die Möglichkeit *Events* zu finden, was zum Verständnis von Diskussionen ebenfalls einen Mehrwert bieten kann, wird vorerst nicht verwendet. Bei der Anfrage nach Artikeln kann die Antwort sofort in Form eines leicht zu untersuchenden Iterators erhalten werden [erd]. Die Anfrage bei *Event Registry* bietet mehrere Filter um die erhaltenen Ergebnisse möglichst adequat werden zulassen. Nun werden einige der möglichen Spezifizierungen vorgestellt und in Ihrem Nutzen diskutiert. Die erste Artikelfilterung kann durch den Parameter *keywords* erfolgen. Dieser ist sehr interessant, da die von der *Natural Language Analysis* ermittelten Schlüsselwörter, für diese Anwendung geeignet sind. Auch die bei Wikipedia gefundenen Wörter könnten hier genutzt werden. Jedoch sind Schlüsselwörter, wie die von *IBM* gelieferten relevanter für den Inhalt des Argumentes, während die Wikipediawörter lediglich Wörter mit

einer bestimmten Bedeutung sind. Es können mehrere Schlüsselwörter übergeben werden, die mit UND oder ODER Verknüpfungen zusätzlich strukturiert werden können. Im Zusammenhang mit dem *keywords* Parameter steht die *keywordsLoc* Angabe, hier kann angegeben werden, wo die Schlüsselwörter gesucht werden sollen. Zur Auswahl stehen Körper, Titel und beides. Die Beschränkung auf den Titel bietet wohl die relevantesten Ergebnisse jedoch auch die wenigsten, während Körper und beides mehr Artikel zur Auswahl stellt. Welche Parametrisierung hier von Vorteil ist, wurde nicht endgültig festgestellt. Neben der Definition von Schlüsselwörtern ist für eine erfolgreiche Nachrichtenrecherche der Bezugsort relevanter Nachricht von Bedeutung. Zur Verdeutlichung dieses Konzeptes, sei das Schlüsselwort "Waffen" genannt, wenn in einer Diskussion die sich auf Deutschland bezieht das Wort "Waffen" genutzt wird, so würde dieses sich wohl am ehesten auf Rüstungsexporte beziehen. Wenn jedoch die Diskussion sich örtlich auf die Vereinigten Staaten von Amerika bezieht, handelt es sich hier wahrscheinlich eher um das Thema, Restriktion von privatem Waffenbesitz. Um solche vom örtlichen Horizont abhängigen Interpretationen schulden zutragen, kann mithilfe einer *locationUri* der örtliche Bezug der Nachricht eingeschränkt werden. Die Nutzung dieses Parameters klingt viel versprechend, ist jedoch schwer umzusetzen. Ein naiver Ansatz wäre zum Beispiel die *locationUri* auf Deutschland zusetzen, da nur Diskussionen aus Deutschland in dem Wiki eingebunden sind. Bei englisch geführten Diskussion könnte man einen internationalen Anspruch vermuten und die *locationUri* ungesetzt lassen. Neben der verwendeten Schlüsselwörter, können auch die von *Natural Language Analysis* ermittelten Kategorien und Konzepte genutzt werden um möglichst akkurate Ergebnisse zu liefern, die Einbindung in die Suche verläuft analog zum *keyword* Parameter, die Übergabewerte heißen hier jedoch *conceptUri* für Konzepte und *categoryUri* für Kategorien. Desweiteren können duplizierte Artikel ausgeschlossen werden, was sehr wichtig ist, da viele Artikel in deutschen Online-Nachrichten-Portalen bei der deutschen Presseagentur ihren gemeinsamen Ursprung haben. In der Implementierung wurden die Artikel nach Schlüsselwörtern gefiltert, zusätzlich wurden die erhaltenen Kategorien mit ODER Verknüpfung in die Anfrage eingebunden. Die *locationUri* ist auf Deutschland gesetzt und Duplikate werden übersprungen. Außerdem werden nur englischsprachige und deutschsprachige Artikel angefragt.

3.6 Verworfen*e API*

Die News *API* [new] entsprach nicht den Anforderungen und hat wenig passende Informationen geliefert. Die Arbeit mit dieser Schnittstelle wurde verworfen, da zu wenig Spezifizierung des Horizontes der Antwortdaten möglich war, was viele unpassende Nachrichten zur Folge hatte.

Kapitel 4

Berechnete Indices

4.1 Lesbarkeitindices

Aufgrund der zeitlichen Beschränkung wurde auf die Nutzung von *machine learning* Algorithmen zur Lösung dieses Problem verzichtet. Stattdessen wurden zwei von den sieben bekanntesten englischsprachigen, automatisch erzeugbaren, Lesbarkeitindices verwendet hier zu zählen [ris]:

- a) *The Flesch Reading Ease formula*
- b) *The Flesch-Kincaid Grade Level*
- c) *The Fog Scale*
- d) *The SMOG Index*
- e) *The Coleman-Liau Index*
- f) *Automated Readability Index*
- g) *Linsear Write Formula*

Hier sind drei verschiedene Arten von Methodiken zu unterscheiden. Zum einen gibt es Formeln, welche in der Berechnung die Anzahlen der Silben benötigen, dies wären hier Index

a), b), c) und d). Während Formel e) und f), zur Berechnung nur auf die Anzahlen der Buchstaben angewiesen sind. Die Formel g) kategorisiert die genutzten Wörter in Bezug auf ihre Komplexität, zur Berechnung dieses Wertes wird jedoch ein hundert Worte langer Text benötigt [lin]. Dieser Einteilung folgend wurden jeweils eine Formel, aus jeder Kategorie, die Formel g) ist außen vor, da kaum ein Argument hundert Wörter enthält, aus den sieben Formeln eingebaut. Es wurde sich dazu entschieden den *Flesch Reading Ease Score*, als Repräsentant einer Formel welche mit Silben rechnet, und den *Automated Readability Index*, als Formel, welche mit Buchstabenanzahlen kalkuliert, zu implementieren. Aufgrund der Tatsache, dass es bereits eine Implementation dieser Algorithmen gibt, wird in dieser Umsetzung das *Python package textstat* [pyT] genutzt.

4.1.1 The Flesch Reading Ease Index

Der betrachtete Index weist jedem Text einen Wert von 0-100 zu. Dies geschieht mit Berechnung der *Flesch Reading Ease formula*:

$$206,835 - 1,015 \left(\frac{\text{Gesamtzahl Wörter}}{\text{Gesamtzahl Sätze}} \right) - 84.6 \left(\frac{\text{Gesamtzahl Silben}}{\text{Gesamtzahl Wörter}} \right) \text{ [Fle]}$$

Es ist zu beachten, dass eine Steigerung der durchschnittlichen Silbenanzahl pro Wort einen stärkeren Einfluss auf das Ergebnisse der Rechnung hat, als die durchschnittliche Wortanzahl in einem Satz.

Die Ergebnisse sind wie in Tabelle 4.1 zu interpretieren:

Score	School level	Notes
100.00-90.00	5th grade	Very easy to read. Easily understood by an average 11-year-old student.
90.0-80.0	6th grade	Easy to read. Conversational English for consumers.
80.0-70.0	7th grade	Fairly easy to read.
70.0-60.0	8th & 9th grade	Plain English. Easily understood by 13- to 15-year-old students.
60.0-50.0	10th to 12th grade	Fairly difficult to read.
50.0-30.0	College	Difficult to read.
30.0-0.0	College graduate	Very difficult to read. Best understood by university graduates.

Abbildung 4.1: Beschreibung der Indizeausprägung des Flesch Reading Ease Index. [fre]

4.1.2 Automated Readability Index

Der *Automated Readability Index* (kurz. ARI) weist jedem Text einen Wert von 1-14 zu. Auch hier geschieht es durch Berechnung einer Formel [ari]:

$$4.71\left(\frac{\text{Gesamtzahl Buchstaben}}{\text{Gesamtzahl Wörter}}\right) + 0.5\left(\frac{\text{Gesamtzahl Wörter}}{\text{Gesamtzahl Sätze}}\right) - 21.43$$

Alle Ergebnisse werden auf natürliche Zahlen aufgerundet.
Die Ergebnisse sind wie in der Tabelle 4.2 zu interpretieren:

Score	Age	Grade Level
1	5-6	Kindergarten
2	6-7	First/Second Grade
3	7-9	Third Grade
4	9-10	Fourth Grade
5	10-11	Fifth Grade
6	11-12	Sixth Grade
7	12-13	Seventh Grade
8	13-14	Eighth Grade
9	14-15	Ninth Grade
10	15-16	Tenth Grade
11	16-17	Eleventh Grade
12	17-18	Twelfth grade
13	18-24	College student
14	24+	Professor

Abbildung 4.2: Beschreibung der Indizeausprägung des Automated Reading Index. [ari]

4.1.3 Fazit

ARI macht eine Einteilung auf Klassenstufen möglich, während der *Flesch Reading Ease Index* keinen direkten Bezug zu den Lesefähigkeiten einer bestimmten Klassenstufe hat. Sowohl ARI, als auch der *Flesch Reading Ease Index*, beziehen sich auf die durchschnittliche Anzahl an Wörtern pro Satz in ihrer Einschätzung. Der *Flesch Reading Ease Index* bezieht

die Silbenanzahl pro Wort ein, derweil der *ARI* die Buchstabenanzahl pro Wort für einen belastbareren Parameter hält. Die Aussagekraft der Werte ist jedoch zu hinterfragen. Es ist zubeobachten, dass die Formeln sich in Ihrer Wahrnehmung widersprechen.

Aufgrund der zwei dargestellten Unterschiede, in der Berechnung und Ergebnisdarstellung, und der Messdifferenzen ist zur besseren Einschätzung, zum Ersten der eigentlichen Lesbarkeit des Argumentes, zum Zweiten der Belastbarkeit der Formeln, beide Arten der Berechnung implementiert worden.



Abbildung 4.3: Die Auswertungsergebnisse der beiden Formeln in Bezug auf das angesprochene Argument

4.2 Positions und Argument Evaluierung

Im folgenden Abschnitt wird auf die Methodik der Evaluierung eingegangen, die hinter dem dargestellten Index, mit dem die Positionen und Argumente angezeigt werden steht.

Zur Berechnung dieses Wertes wurde die Arbeiten von Dr. Marcin Selinger [Sel14, ISSA14, Sel16] genutzt. Diese werden kurz erläutert, danach werden Entscheidungen erläutert, die eine Automatisierung im betrachteten Fall zulassen. Es handelt sich um Formeln die die Verrechnung von Argumenten zulassen, die der im entworfenen Wiki verwendeten Argumente entspricht. Des Weiteren haben die Ergebnisse entscheidende Vorteile [Sel16] gegenüber der vorgeschlagenen Verrechnung durch Robert J. Yanal in [Yan91], die im jeweils roten Graph in [Sel16] betrachtet werden können. Andere Berechnungsmethoden hätten eine zusätzliche Struktur von Nöten gemacht und somit das eigentliche Ziel der Arbeit in den Hintergrund gerückt. Bei Interesse sind die folgende Paper zu lesen [ART, LA]

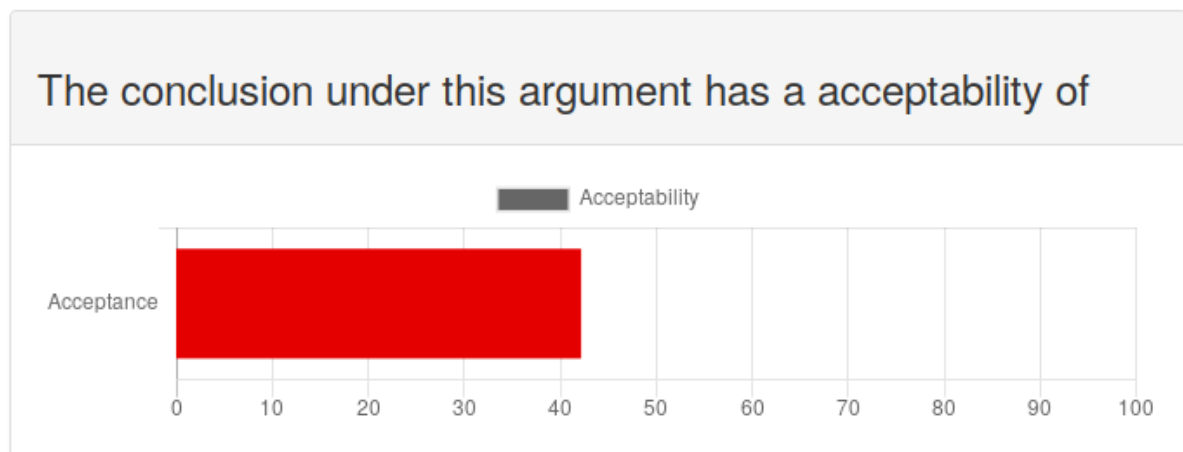


Abbildung 4.4: Beispiel der Bewertung eines Argumentes

4.2.1 Grundlagen

Zur sinnvollen Bewertung von Argumenten muss geklärt sein, welche Argumenttypen betrachtet werden und wie diese Ausprägungen aussehen. Für genauere Information können hier [Sel16] noch weitere Typen von Argumenten eingesehen werden.

Atomares Argument bei dieser Form handelt es sich um ein Argument, welches aus einer Prämisse und einer Konklusion besteht. Hier sind die pro-Argumente wie in Figure 4.5

und der con-Argumente in 4.6 zu unterscheiden.

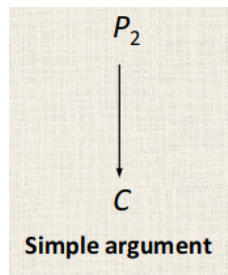


Abbildung 4.5: Skizze eines atomaren pro-Argumentes

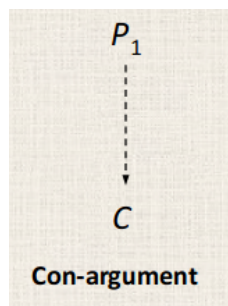


Abbildung 4.6: Skizze eines atomaren contra-Argumentes

Verbundenes Argument bei dieser Form handelt es sich um ein Argument, welches mehrere zusammenhängende Prämissen besitzt und mit gleicher Polarität eine Konklusion folgert. Diese Argumentform kann in pro und in contra Ausprägung auftreten. Figur 4.7 stellt beispielhaft ein pro-argument dar.

Konvergentes Argument bei dieser Form handelt es sich um ein Argument, welches aus mehreren unabhängige Prämissen besteht, die mit gleicher Polarität eine Konklusion folgern. Hierbei kann es sich sowohl um ein contra-konvergentes, als auch um ein pro-konvergentes Argument handeln. In Figur 4.8 ist ein pro-Argument dargestellt.

Vereinendes Argument bei dieser Form handelt es sich um ein Argument, welches aus mehreren unabhängige Prämissen besteht, die mit unterschiedlicher Polarität eine Kon-

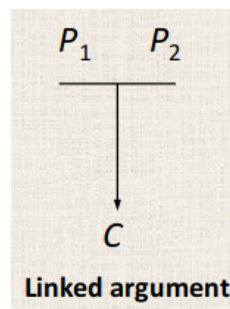


Abbildung 4.7: Skizze eines verbundenen pro-Argumentes.

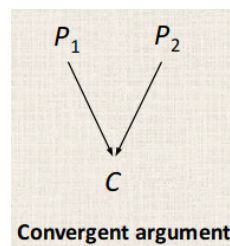


Abbildung 4.8: Skizze eines verbundenen con-Argumentes.

klusion folgern. In Figur 4.9 ist ein Argument mit einer Prämisse, welche die Konklusion angreift und einer Prämisse welche die Konklusion unterstützt.

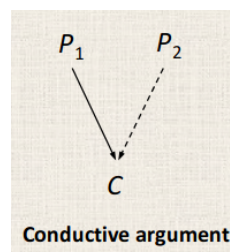


Abbildung 4.9: Skizze eines vereinenden Argumentes dar.

4.2.2 Bewertung von Argumenten

Verrechnung von Prämissenakzeptanz

Die Akzeptanz von Prämissen wird durch einfache Multiplikation der Prämissenbelastbarkeit erreicht. Die Entwicklung der Belastbarkeit eines Prämissensets beschreibt Figur 4.10.

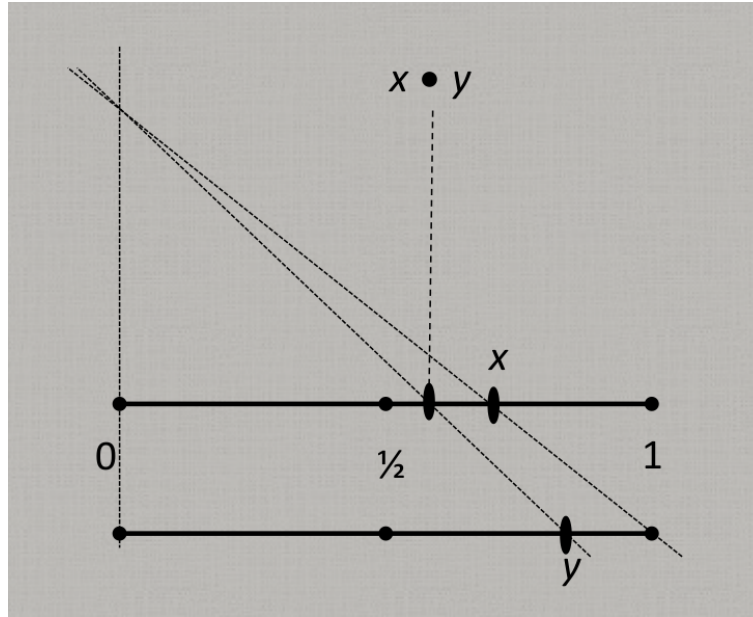


Abbildung 4.10: Beschreibt die Entwicklung von Prämissenakzeptanz

Akzeptanz eines atomaren oder verbundenen Argumentes

Zur Berechnung der Akzeptanz eines atomaren, verbundenen Argumentes muss folgende Rechnung durchgeführt werden:

$$v_a(\alpha) = v(\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n) w(\alpha | \alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n)$$

$v_a(\alpha)$, sei der Akzeptanz einer Konklusion α im Argument a .

$v(\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n)$, sei die Akzeptanz der verwendete Prämissen. Dieser Wert sollte immer $> \frac{1}{2}$ sein, da sonst von einer widerlegten Prämisse Schlüsse gezogen werden, was ein logischer Fehlschluss wäre [Sel14].

$w(\alpha | \alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n)$, sei die Akzeptanz mit welcher α unter der betrachteten Prämissen angenommen wird, wenn die Prämisse komplett richtig wäre. [Sel14] In Figur 4.11 ist das Zusammenspiel der Parameter graphisch dargestellt. Mit $x = w(\alpha | \alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n)$ und

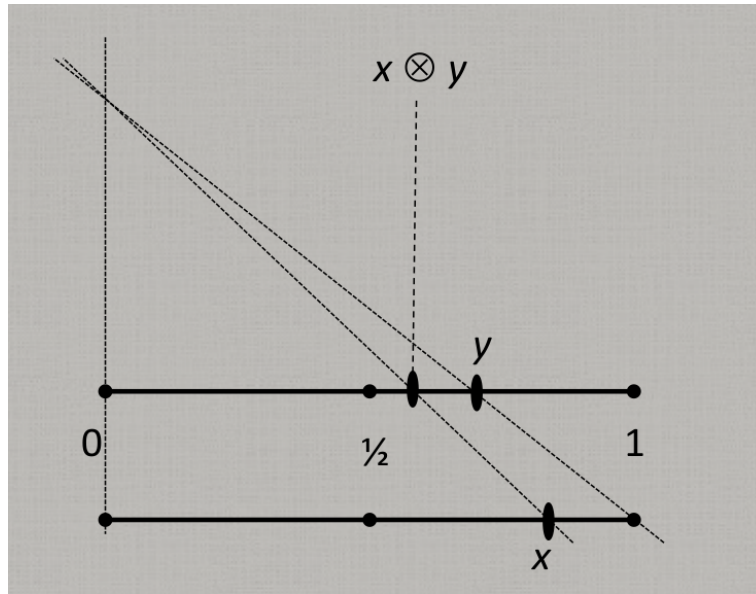


Abbildung 4.11: Skizziert die Veränderung der Argumentsakzeptanz

$$y = v(\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n)$$

Bewertung eines convergenten Argumentes

Zur Berechnung der Akzeptanz eines pro-konvergenten Argumentes wird folgende Formel genutzt, sei $v_{s_1}(\lambda) > \frac{1}{2}$ und $v_{s_2}(\lambda) > \frac{1}{2}$ und die Prämissen $p(s_1), p(s_2)$ sind unabhängig, dann:

$$v_{s_1, s_2}(\lambda) = v_{s_1}(\lambda) \oplus v_{s_2}(\lambda)$$

mit $y \oplus x = 2x + 2y - 2xy - 1$ [Sel14]. Es wird also die Ungewissheit, die das erste Argument hinterlässt hier dargestellt im Intervall $[x, 1]$, proportional verringert durch die Gewissheit gegeben vom zweiten Argument hier das Intervall $[1/2, y]$. Diese Beeinflussung ist in Figur 4.12 zuerkennen. Wobei $x = v_{s_1}(\lambda)$ und $y = v_{s_2}(\lambda)$

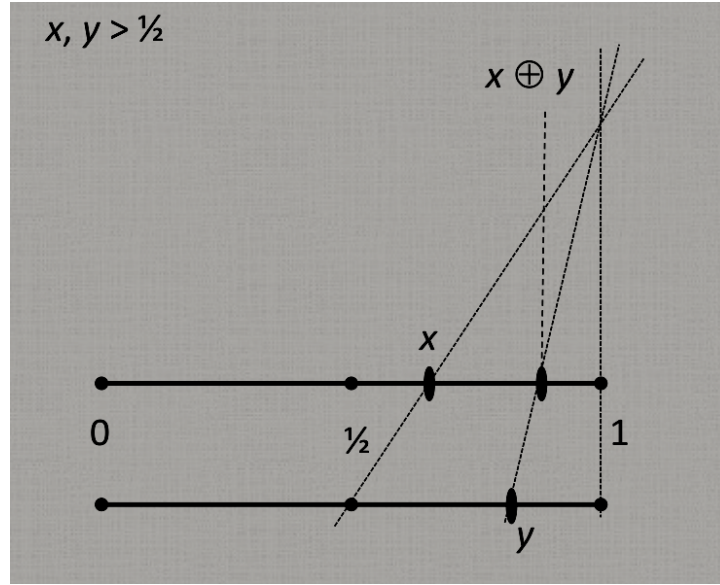


Abbildung 4.12: Skizziert die Veränderung der Akzeptanz unter einem pro-konvergenten Argument.

Bei con-konvergenten Argumenten muss beachtet werden, dass $v_{s_1}(\lambda) < \frac{1}{2}$ und $v_{s_2}(\lambda) < \frac{1}{2}$ gegeben sein muss. Um die oben genannte Formel auf con-Konvergente Argumente anzuwenden muss also:

$$v_{s_1, s_2}(\lambda) = 1 - ((1 - v_{s_1}(\lambda)) \oplus (1 - v_{s_2}(\lambda)))$$

berechnet werden. In Figur 4.13 wird die Beeinflussung der Konklusionsakzeptanz unter einem con-konvergenten Argument dargestellt. Wobei $x = v_{s_1}(\lambda)$ und $y = v_{s_2}(\lambda)$

Bewertung verbindener Argumente

Zur Berechnung von verbindenden Argumenten werden erwartet, dass $v_{s_1}(\lambda) < \frac{1}{2}$ und $v_{s_2}(\lambda) > \frac{1}{2}$, wobei v_{s_1} die Akzeptanz aller con-konvergenten Argumente ist und v_{s_2} alle pro-konvergenten Argumente sind. Die Berechnung erfolgt durch die Formel:

$$v_{s_1, s_2}(\lambda) = v_{s_1}(\setminus) v_{s_2}$$

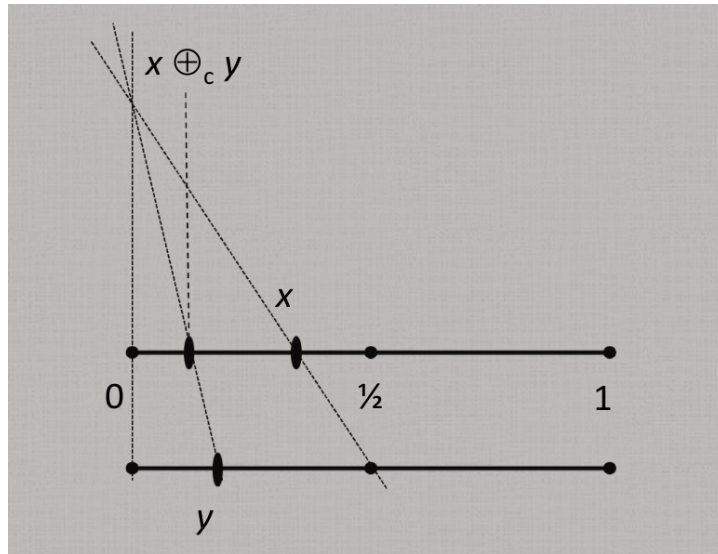


Abbildung 4.13: Skizziert Wertentwicklung der Konklusionsakzeptanz unter einem con-konvergenten Argument

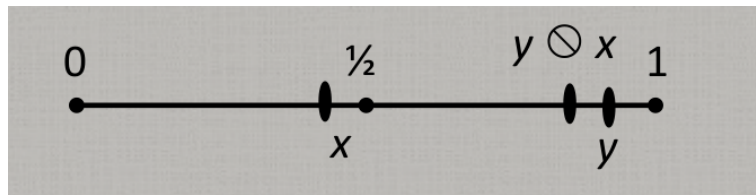


Abbildung 4.14: Skizziert das Ergebnis einer Verrechnung eines verbindenden Argumentes

wo $y \ominus x = y + x - \frac{1}{2}$ [ISSA14] ist. Die Abbildung 4.14 stellt die Beeinflussung der Konklusionsakzeptanz durch die unter x zusammengefassten con-konvergenten Argumente und die unter y zusammengefassten pro-konvergenten Argumente dar.

Bewertung von Undercuts

Um die Stärke eines Undercuts zu berechnen muss x als die abhängige Akzeptanz des angegriffenen Argumentes, das Prämisse A Konklusion B zur Folge hat. Mit der Akzeptanz des Undercutters y verrechnet werden. auch hier gilt $x, y > \frac{1}{2}$

$$v_{s_1, s_2}(\lambda) = v_{s_1} \odot v_{s_2}$$

wo $x \odot y = 2x + y - 2xy - \frac{1}{2}$. In Figur 4.15 wird die Entwicklung abhängigen Akzeptanz aufgezeigt bei einem Angriff auf y auf x .

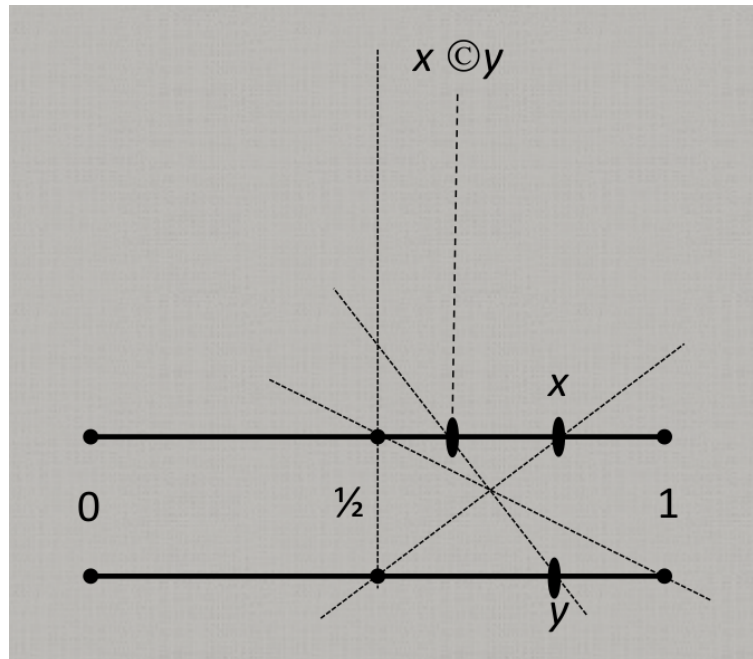


Abbildung 4.15: Skizziert das Ergebnis der Rechnung einer Undercut-Bewertung

4.2.3 Automatisierungs Entscheidungen

Es gibt folgende Probleme bei der Berechnung der Argument-, Prämissen- und Konklusionsakzeptanzen nach den oben genannten Formeln. Damit eine automatisierte Implementation gelingt muss jeder am äußersten Ende gelegenen Prämisse, also eine Prämisse welche nicht Konklusion eines anderen Argumentes ist eine pauschale Akzeptanz zugewiesen werden. Bei zyklischen Graphen wird die Berechnung unterbrochen und der zyklische Teil wird mit einer Akzeptanz von 0,5 belegt. Eine vielleicht bessere Lösung für zyklische Argumentgraphen wäre es zyklische Argumente, durch die Reviewer zu eigenen Argumenten zusammenzufassen und so diese Argumente nebeneinander aufzuführen zukönnen. Des Weiteren muss die abhängige Akzeptanz einer Konklusion berechnet werden. Zur Lösung der beiden Probleme wurde entschieden eine pauschale Bewertung der äußersten Prämisse vorzunehmen. Diese besitzt eine Akzeptanz von 0,75, dies entspricht einer Aussage die zur Hälfte richtig und zur Hälfte falsch ist. Zur Verteidigung und zur Erklärung dieser Entscheidung sei

folgendes gesagt. Eine Prämisse ist eher dazu geneigt die Realität nicht ganz darstellen zu können, deshalb kann von einer generellen Fehleranfälligkeit ausgegangen werden. Außerdem würde eine automatisierte Einschätzung der Akzeptanz von Prämissen den Horizont der Arbeit sprengen. Es bleibt die Entscheidung bzgl. der abhängigen Akzeptanz einer Konklusion. Dieses Problem wurde wie folgt angegangen. Eine Prämisse bildet nicht die Realität in ihrer ganzen Komplexität ab, ist jedoch vom Verfasser meistens dazu erdacht worden die Konklusion unumwülflich zu folgern. Deshalb wird in der Arbeit, ohne weiteres Einwirken anderer Argumente von einer abhängigen Akzeptanz von 1 ausgegangen, das heißt, wenn die Prämisse vollständig richtig wäre, dann wäre auch immer die Konklusion vollständig richtig. Diese abhängige Akzeptanz wird verringert, wenn die betrachtete Verbindung von einem Undercut angegriffen wird. Für diese Verringerung der Glaubhaftigkeit wird die in Kapitel 4.2.2 beschriebene Bewertung von Undercuts angewendet. Die so errechnete Argumentakzeptanz könnte jetzt noch durch das User-Voting nach justiert werden. Dafür müsste jedoch eine angemessene Abwägung zwischen dem Verhältniss der Pro- und Contra-Votings gefunden werden und zusätzlich, die Aussagekraft der reinen Quantität von Votings verstanden werden.

4.3 Index bzgl. der akademischen Ausrichtung des Argumentes

Als relevante Wörter stehen die Schlüsselwörter erhalten durch die *Natural Language Analysis* und die Wörter welche bei Wikipedia bekannt sind zur Verfügung. In dieser Implementierung wurden in den hier beschriebenen Score nur die von Wikipedia erkannten Wörter genutzt. Dies hat folgende Gründe, zum einen ist die Anzahl, der von der *Natural Language Analysis* ermittelten Schlüsselwörter auf drei beschränkt, da bei der im Durchschnitt gegebenen kurzen Textlänge der Argumente und der Tatsache, dass die Schlüsselwörter schon nach Relevanz geordnet übermittelt werden, mehr Schlüsselwörter kaum zusätzlichen Vorteil bringen. Zum anderen überschneiden sich die Worte teilweise, was zu einer Verzerrung des hier beschriebenen Ergebnisses führen würde. Zusätzlich bestehen die Schlüsselwörter erhalten durch *IBM Watson* oft aus mehr als einem Wort, was sie für diese Ermittlung unbrauchbar macht. Zur Kategorisierung der Wörter, im Bezug auf ihre relative Mehrnutzung, in bestimmten wissenschaftlichen Disziplinen, im Gegensatz zum alltäglichen Gebrauch wird das 3000 Worte beinhaltene Lexikon 'Academic Vocabulary List' [aca] genutzt. Nun wird, sowohl

jedes Wort, als auch jeder Titel des dem Wort zugeordneten Wikipediaartikels ¹ in dem Wörterbuch gesucht. Findet sich ein Wort welches eines der beiden Wörter beinhaltet, werden die dazugehörigen Häufigkeitswerte, die einzelnen Disziplinen betreffend gespeichert. Aus der Menge der relevanten Wörter, die in dem Lexikon gefunden wurden wird der Häufigkeitsdurchschnitt, für jede Disziplin errechnet, was Einblicke in den Gedankenhorizont des Argumentierenden gewährleisten soll.

¹ Ambivalent oder nicht wird hier nicht beachtet

Kapitel 5

Implementation

Das entworfene Wiki besitzt acht verschiedenen Übersichtsseiten, die für eine verbessertes Verständnis der zugrundeliegenden Argumentnetzwerke stark ineinander verzweigt sind.

5.1 Themenübersicht

Die *Landingpage* stellt alle im Wiki aufgeführten Diskussionen dar, diese sind nach Erscheinungsdatum geordnet. Sie werden zur besseren Übersicht in klar getrennten Feldern angezeigt und mit zusätzlicher Kurzerklärung präsentiert. Von dieser Seite aus kann mit Auswahl einer Diskussion auf die Übersichtsseite einer komplette Diskussion gewechselt werden. In Abbildung 5.1 ist ein Ausschnitt aus der beschriebenen Themenübersichtsseite zu betrachten.

Issues

Verbesserung des Informatik-Studiengangs

Wie können der Informatik-Studiengang verbessert und die Probleme, die durch die große Anzahl der Studierenden entstanden sind, gelöst werden?

Town has to cut spending

Our town needs to cut spending. Please discuss ideas how this should be done.

Abbildung 5.1: Diskussionsthemen mit Information auf der Landingpage

Weiter kann hier ein Update durchgeführt werden, welches alle Daten aktualisiert. Oder es können alle Argumente über den Knopf mit der Aufschrift *Search* durchsucht werden.

5.2 Diskussionsübersicht

Hier sind alle Positionen mit ihren errechneten Evaluationsscores gelistet. In Abbildung 5.2 ist eine Position mit seinem berechneten Akzeptanzscore zu sehen.

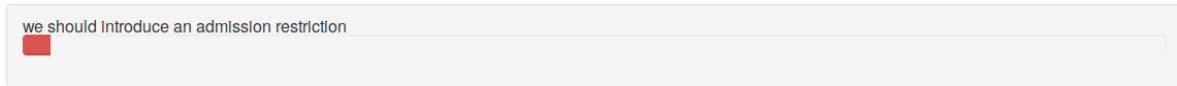


Abbildung 5.2: Einzelne Position mit graphischer Darstellung des Akzeptanzscore

Durch das Auflegen des Mauszeigers auf die angezeigte Scorevisualisierung wird die Anzeige einer Erklärung, per *tooltip*, ausgelöst. In Darstellung 5.3 ist diese Erklärung zu sehen.

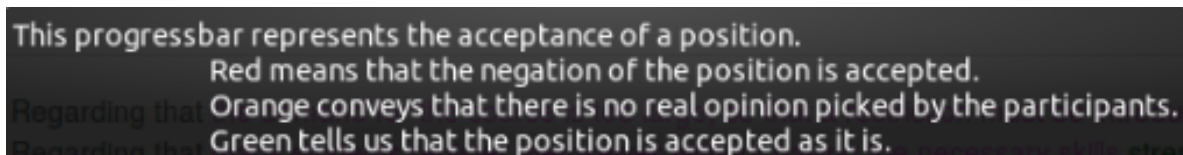


Abbildung 5.3: Erklärung zum akzeptanz Score

Per Tastendruck lassen sich die unter einer Position gelisteten Argumente auflisten. Hier ist zu beachten, dass ein Farbcode zum schnellen Erkennen der Argumentstruktur beiträgt. In Abbildung 5.4 ist ein beispielhaft ausgewähltes Argument angezeigt.

Regarding that the demand for the course is too large, so that a restriction must be introduced strengthens the claim that we should introduce an admission restriction

Abbildung 5.4: Einzelnes Argument aufgelistet unter einer Position

Die Sprechblase symbolisiert das ganze Argument. Es wird versucht den Fülltext 'Regarding that' durch ein unauffälliges grau in den Hintergrund zurücken. Die Prämisse des Argumentes ist rosa. Der Text der die Relation zwischen Prämisse und Konklusion beschreibt ist je nach Zustimmung oder Ablehnung, der Konklusion durch die Prämisse, Grün oder Rot gefärbt. Außerdem sind hier vier verschiedene Formulierungen, die dem Argumenttyp entsprechen, genutzt worden.

Support *strengthens the claim that*

Rebut *weakens the claim that*

Undermine und Undermine/Rebut *questions the validity of*

Undercut doubts the inference from

Die Konklusion ist blau angezeigt. Handelt es sich um einen Undercut, so wird das angegriffene Argument ebenfalls in einem unauffälligen grau dargestellt, da dieses Argument bereits einen eigenen Eintrag besitzt. In der beschriebenen Darstellung befinden sich drei verschiedene *Anchortags*, welche zu drei verschiedenen Seiten verlinken. Über die Sprechblase gelangt man zur Argumentübersicht, über die Prämisse und Konklusion gelangt man zu deren jeweiligen Übersichtsseiten.

5.3 Prämissen- und Konklusionübersicht

Diese beiden Übersichten sind sehr nah miteinander verwandt. Die Prämissenübersicht zeigt an, welche Konklusionen angezweifelt werden und welche Konklusionen unterstützt werden, hier in Abbildung 5.5 zu betrachten.

the demand for the course is too large, so that a restriction must be introduced

Questions:

Supports:

we should introduce an admission restriction

Abbildung 5.5: Übersichtsseite einer einzelnen Prämisse

Die Konklusionsübersicht zeigt an, welche Prämissen die Konklusion anzweifeln und welche Prämissen sie unterstützt, hier zu betrachten in Abbildung 5.6.

Hier werden dieselben Farbcodes genutzt, wie in der Diskussionsübersicht. Gegenätzlich zu dieser wird jedoch auf die Argumentübersicht verwiesen, wenn man eine der aufgeführten Prämissen oder Konklusion anklickt, diese Funktionen wird jedoch als intuitiv und zielführend eingeschätzt, da man durch besuchen der Seite sich schon für eine Konklusion bzw. Prämisse entschieden hat. Somit wählt man nicht nur eine einzelne Prämisse aus sondern die Verbindung einer Prämisse mit einer Konklusion mit der darüberaufgeführten Relation.

we should introduce an admission restriction

Attacked By:

a comparability of the high school grades is not given
the capacity of the university should be increased instead of excluding new students
only by giving the computer science studies a trial one can determine whether one is suited for these studies
the secondary school certificate refers only to a small extent to relevant fields, it is not meaningful
it only moves students, who enroll solely for the local traffic card, (if at all) to other admission-free subjects
one can achieve a good secondary school certificate also with (many) tutoring lessons, which depends on the parents' financial situation
one should first examine the factors that determine the success of computer science studies

Supported By:

the demand for the course is too large, so that a restriction must be introduced
many students register themselves without having the necessary skills
by this, the courses are not used to filter out students

Abbildung 5.6: Übersichtsseite einer einzelnen Konklusion

5.4 Suchseite

Auf dieser Seite können alle Argumente durchsucht werden. Es sind folgende Parameter zur Suche implementiert.

Search argument hier können Argumente nach Text gesucht werden.

Published From - To hier können die Erstellungsdaten der Argumente eingeschränkt werden.

Supportive oder Attacking es kann ausgewählt werden welche Beziehung ein Argument zwischen der Prämisse und der Konklusion sieht

From acceptability - To hier kann der Akzeptanzwert eingeschränkt werden

Language hier kann die Sprache spezifiziert werden

Zur Implementierung dieser Seite wurde das *Python Package django-search-views* verwendet [dsv].

5.5 Argumentübersicht

Die Argumentübersicht ist das Herzstück der Webanwendung. Die Übersichtsseite kann in 4 Bestandteile aufgeteilt werden, die Überschrift, Statistiken und andere Einordnungen, Darstellung im Kontext des Argumentnetzwerkes und die Sammlung an erzeugten Hintergrundinformationen.

5.5.1 Überschrift

Die Überschrift besteht aus dem Argument, ist in den schon erläuterten Farbcodes aufgebaut und hat dieselben Funktionalitäten, wie die Argumente in der schon beschriebenen Diskussionsübersicht. Der Zusatz, der in der Argumentationsübersicht zugänglich gemacht wird, ist das Lautsprecheresymbol, welches auf Mausklick, die, von der in Kapitel 3.2 beschriebenen *API*, Sounddatei, welche das Argument in gesprochener Sprache enthält, abspielt. Von dieser Funktion wurde sich ein höheres Potenzial versprochen, dem User das Argument zugänglich zu machen. Wie in [wik] beschrieben ist das Ziel eines *Wikis* 'Erfahrung und Wissen ... in für die Zielgruppe verständlicher Form zu dokumentieren.' [wik], deshalb ist es für empfehlenswert gehalten worden, die Möglichkeit zu schaffen die Information über verschiedene Sinneserfahrungen konsumierbar zu machen

5.5.2 Inhaltsverzeichnis

Hier sind alle Inhalte der Seite aufgeführt. Mit Mausklick kann zu der jeweiligen Information gesprungen werden.

5.5.3 Einordnungen

Zu den Statistiken und den anderen Einordnungen gehören:

Der Typ eines Argumentes Dies kann:

- Support
- Rebut
- Undermine/Rebut
- Undermine
- Undercut

sein.

Die Meinungspolarität des Arugmentes Dieser von Wert ist von der in Kapitel 3.1 beschriebenen *API* empfangen worden und kann folgende Ausprägung annehmen:

- Positive
- Neutral
- Negative

Auch hier findet eine farbliche Untermalung der Information mit den Farben Grün, Grau und Rot statt.

Die Darstellung der Zustimmungen und Gegenstimmen , diese Werte werden mithilfe des in Kapitel 2.2 beschriebenen *Frameworks* in Form eines Kreisdiagrammes dargestellt.

Die Bewertung der Konklusion unter diesem Argument , dieser Wert ist Ergebnis der Rechnung aus Kapitel 4.2.2, und wird ebenfalls mit Chart.js in Form eines Balkendiagrammes dargestellt.

Die Bewertung der Leseschwierigkeit durch ARI und durch Flesch Reading Ease Beide Ergebnisse sind durch ein Balkendiagramm visualisiert. Um ein visuellen Vergleich der Indices zu erleichtern, was durch die verschiedenen Skalen erschwert ist, lässt die Farbgebung einen direkten Vergleich zu. Je dunkler die Farbe des Balken, desto komplexer die Formulierung. die Farbgebung ist bei beiden so gestaffelt, dass gleiche Färbung gleiche Einschätzung bedeutet.

Die Einordnung der Informationen nach ihrer Nutzung in Akademischenzusammenhängen

Die in Kapitel 4.3 beschriebenen Werte werden hier in Form eines Radardiagrammes dargestellt. Per *Default* wird hier der errechnete Gesamtwert für das Argument angezeigt. Es können jedoch die einzelnen Scores der im Text gefundenen Worte aufgerufen werden und im Kontrast zu einander und zum Gesamtwert betrachtet werden.

Die durch *IBM Watson* erkannten Emotionen Der empfangene Analyse wird mithilfe eines Radardiagrammes dargestellt.

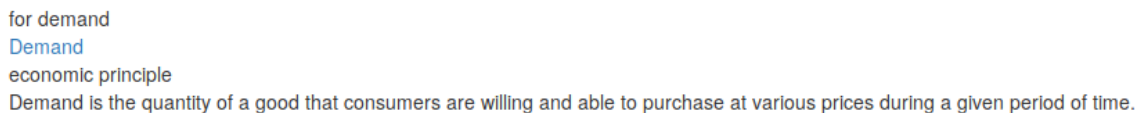
Bei allen Statistiken und Zuordnung, die einer genaueren Erklärung benötigen, kann mit dem Auflegen des Mauszeigers auf dem Namen der Information eine Hilfestellung zur Interpretation angezeigt werden.

5.5.4 Darstellung im Kontext der Netzwerke

Es wurde sich für die Darstellung von 7 verschiedenen Einordnungen in das Argumentnetzwerk entschieden. Als erste Information wird unter dem Stichpunkt *Under Issues* in dieser Kategorie wird dargestellt, in welchen Diskussionsthemen das Argument Anwendung gefunden hat. Die zweite Einordnung unter Stichpunkt *Under Positions*, zeigt an auf welche Positionen sich das Argument direkt oder indirekt bezieht. Weiter werden unter Stichpunkt *Rebuts of conclusion*, alle diejenigen Prämissen angezeigt welche zur Entkräftung des gefolgerten Ergebnisses, genutzt werden. Auch hier gilt wieder es wird nicht auf eine Prämissenübersichtsseite verwiesen sondern auf eine Argumentsübersichtsseite, da sowohl Prämisse, Konklusion und Relation implizit im Kontext gegeben sind. Als nächstes wird unter der Überschrift *Supports for Inference*, alle diejenigen Prämissen gelistet, welche in Argumenten genutzt werden, mit der selben Konklusion und zu dieser die selbe Polarität ausdrücken. Unter *Undermine by*, befinden sich alle Prämissen welche die Prämisse vom betrachteten Argument als Konklusion haben und diese versuchen anzugreifen. Im Gegenteil dazu sind unter *Supports for premise*, alle Prämissen der Argumente gelistet, welche die Prämisse, des betrachteten Argumentes, unterstützend als Konklusion enthalten. Zuletzt werden unter *Attacked by*, alle diejenigen Prämissen angezeigt welche einen Undercut auf das betrachtete Argument zur Folge haben.

5.5.5 Sammlung der erzeugten Hintergrundinformationen

Die Sammlung der erzeugten Zusatzinformationen enthält die Informationen, welche durch die mit in 3.1, 3.4 und 3.5 genutzten *APIs* erhaltenen Informationen, erstellt werden konnten. Der Abschnitt wird in drei Hauptbestandteile unterteilt. Zum einen die Wörter oder Bezeichnungen die vermeintlich zweifellos von der Wikipedia *API* erkannt wurden. Zum anderen diese die auf mehrere Wikipediaeinträge anwendbar sind. Und letztlich wird auf die Verwendung und Darstellung derer Daten eingegangen, die von der in 3.1 beschriebenen *API* erhalten wurden. Die Wörter oder Wortpaare welche zweifelsfreie Antworten von der Wikipedia *Api* erhalten, werden Form dargestellt wie in Abbildung 5.7.

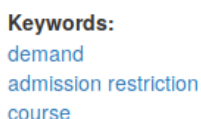


for demand
Demand
economic principle
Demand is the quantity of a good that consumers are willing and able to purchase at various prices during a given period of time.

Abbildung 5.7: Darstellung einer eindeutig von Wikipedia erhaltenen Information.

In der ersten Zeile befindet sich das genutzte Wort/Wortpaar welches an die *API* gesendet wurde. Darunter wird der Titel der Wikipediaseite genannt mit der Möglichkeit über einen *Hyperlink* direkt auf die entsprechende Wikipediaseite zu wechseln. Es folgen nun eine kurze Beschreibung und eine Zusammenfassung des Artikels. Bei ambivalenten Suchbegriffen wird auf das Anzeigen der erhaltenen Beschreibung und der Zusammenfassung verzichtet. Bei den von der *Natural Language Analysis* zurück gegebenen Schlüsselwörtern

Kann auf eine Seite zugegriffen werden die Nachrichten bzgl. dieser Schlüsselwörter prä-



Keywords:
demand
admission restriction
course

Abbildung 5.8: Anzeige der Schlüsselwörter

sentiert. Die Konzepte die ermittelt werden verfügen über einen eigenen Informationsartikel, welcher mit anklicken des Konzeptes erreicht werden kann. Diese ähnliche Funktionalität zu der ,der Wikipedia Einbindung, findet seine Berechtigung in der Tatsache, dass die vom IBM Watson ermittelten Konzepten nicht namentlich in dem Text genannt sein müssen. Zum Schluss gibt es für die angegebenen Emotionen und Kategorien die Möglichkeit eine Übersichtsseite zu besuchen.

5.6 Nachrichtenübersichtsseite

Die Nachrichtenübersichtsseite ist über die Schlüsselwörter, der Argumente zu erreichen. Auf dieser Ansicht werden alle Artikel aufgeführt, die zu dem betrachteten Schlüsselwort im Zusammenhang mit den gesammelten Kategorien gefunden wurden. Die Artikel werden komplett zitiert. Die Titel der Artikel sind mit einer Verlinkung zum Original versehen. Es wird neben Titel und Inhalt, die Nachrichtenquelle, die Art der Internetquelle und das Datum der Entstehung angezeigt.

5.7 Kategorieübersichtsseite

Auf dieser Seite werden alle Argumente aufgeführt, die einer bestimmten Kategorie zugeordnet werden. Hier wird bewusst darauf verzichtet, diese Menge durch einer bestimmten Zugehörigkeit, beispielsweise zu einem Diskussionsthema oder zu einer Position, zu beschränken. Durch diesen Verzicht und der Tatsachen, dass jedem Argument 3 Kategorien zugeordnet sind, wird die angezeigte Menge an Argumenten sehr unübersichtlich. Damit die Seite für den Nutzer einen Navigationsvorteil darstellen kann, wird der von der *API* mitgelieferte Überzeugungsparameter zur Strukturierung der Seite eingebaut. Es wird eine Aufteilung der Argumente vorgenommen nach der Überzeugung mit welcher *IBM Watson* die Texte der jeweiligen Kategorie zugeordnet hat, diese Aufteilung besteht aus 3 Gruppen:

Hohe Überzeugung Relevanzwert $\geq 0,9$

Mittlere Überzeugung Relevanzwert $< 0,9$ und $> 0,7$

Niedrige Überzeugung Relevanzwert $\leq 0,7$

Kapitel 6

Evaluation

In der Bewertung des Wikis wird sich auf die Taxonomie für Argumentationen von Wachsmuth et al. [war17] bezogen 6.1. Es werden im Gegensatz zu Wachsmuth, der nur die Beweis-

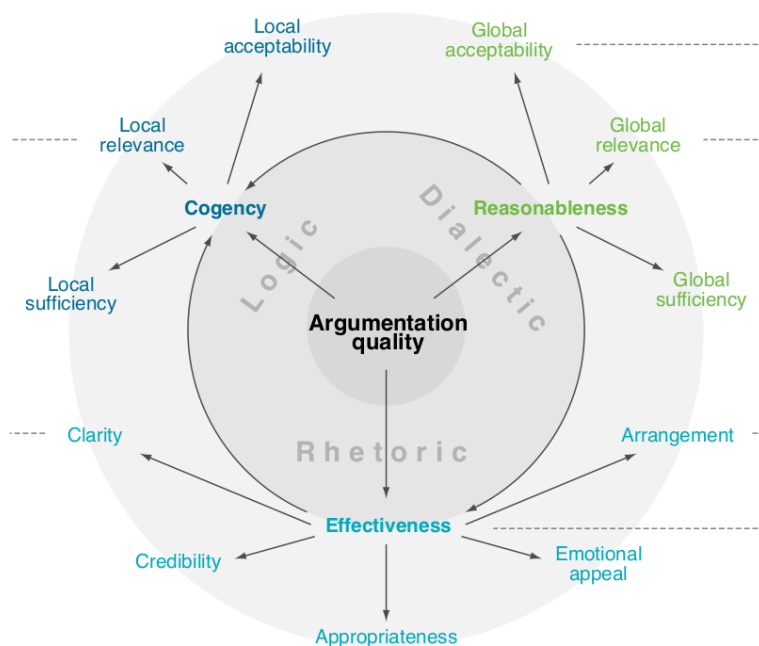


Abbildung 6.1: Stellt Taxonomie zur Bewertung von Argumentationen dar

kraft als Eigenschaft eines Argumentes sieht, die Effektivität die meistens als Eigenschaft der gesamten Diskussion betrachtet wird ([war17] Kapitel 3.2). Auch auf einzelne Argumente bezogen. Die Dimension *Reasonableness* ist analog zur Cogency, nur auf Argumena-

tionsebene. Ob ein Argument Beweiskraft besitzt wird in drei verschiedene Unterkategorien aufgeteilt. Die lokale Akzeptanz, eine Prämisse ist akzeptable, wenn sie rational als wahr angesehen werden kann. Die lokale Relevanz ist gegeben wenn die Prämisse zur Annahme oder Ablehnung der Konklusion beiträgt. Die lokale Zulänglichkeit beschreibt die Fähigkeit der Prämissen, gemeinsam genug Wissen anzubieten, sodass die Konklusion gefolgert werden kann. Die lokale Akzeptanz eines Argumentes kann vom Nutzer besser verstanden werden, wenn er sich zum einen den Wert der Argumentationsevaluation anschaut, ist dieser bei 0,5 so ist die Prämisse von seinen Vorgängern erfolgreich angegriffen worden. Wie gut jedoch diese Prämissen der Angreifer sind, kann der Nutzer in der jeweiligen Übersicht ansehen und ist so befähigt eine verlässliche Aussage zur lokalen Akzeptanz des Argumentes treffen zu können. Die lokale Relevanz ist nur fehlerfrei zu beobachten, wenn der Nutzer mit den Argumentationsergebnissen übereinstimmt. Andernfalls bildet die Einschätzung aus Kapitel 4.2 keine richtige Auswertung an. Bei der lokalen Zulänglichkeit ist der User darauf angewiesen das Argument mithilfe der Informationen richtig zu bewerten und so selber zu entscheiden, ob diese Eigenschaft eines Argumentes gegeben ist, für diese komplexeren Fälle wird jedoch eine Vielzahl von recherchierten Daten bereitgestellt, die dem Nutzer die Möglichkeit geben sich ein genaueres Bild zu schaffen.

Nun zur Effektivität. Ein Argument ist effektiv, wenn es die Zielgruppe überzeugt. Auch diese Dimension ist in mehrere Unterkategorien aufgeteilt. Hier unterscheidet man Kreditibilität, kann dem Autor geglaubt werden. Die emotionale Wirkung, wird die Zielgruppe ergriffen und öffnet sich so besser dem Argument. Klarheit, ist das Argument eindeutig geschrieben, vermeidet es unnötige Komplexität. Die Eignung des Arguments ist gegeben, wenn die gewählte Sprache zur Bildung von Kreditibilität und emotionaler Wirkung beiträgt und sie zum Thema passt. Anordnung wird ausgelassen, da ein Bezug auf ein Argument keinen Mehrwert bietet, da strukturierte Argumente betrachtet werden. Die Glaubwürdigkeit des Argumentes kann anhand des Abstimmungsdiagrammes eingesehen werden. Die emotionale Wirkung kann durch abgleich der empfundenen Emotionen, mit den ausgewerteten Emotionen geschehen, die Relevanz dieser Ergebnisse ist jedoch schwer einzuschätzen. Ob das Argument klar formuliert ist kann mithilfe der Lesbarkeit-Indices abgelesen werden. Die Eignung des Argumentes kann teilweise mit den schon genannten Funktionalitäten erkannt werden, jedoch bietet hier der in Kapitel 4.3 beschriebene Score zusätzliche Einsicht, ob das Argument sich einer geeigneten Sprache bedient.

6.1 Bewertung der Funktionalitäten

Ob das implementierte Wiki mit seinen heutigen Funktionalitäten nützlich ist, kann erst ein längerer Gebrauch zeigen. Es können jedoch schon nützliche und weniger nützliche Versuche sinnvolle Funktionalitäten zu implementieren ausgemacht werden. Die Struktur ist sicher von Vorteil um Argumentnetzwerke in Ruhe zu untersuchen. Außerdem gibt der errechnete Akzeptanzscore eine schnelle und akkurate Übersicht über die vertretene Meinung der Diskussionsteilnehmer. Auch die Einordnung des Argumentes in den Kontext des Argumentnetzwerkes ist zielführend. Die Nützlichkeit der Lesbarkeit-Indices ist fragwürdig, es müsste klar gestellt werden wie oft die angegebenen Indices sich widersprechen und ob eine Einbindung aufgrund der Erkenntnisse gerechtfertigt bleiben kann. Der in Kapitel 4.3 beschriebene Index, bietet ebenfalls Anhaltspunkte für das bessere Verständnis. Die Anwendbarkeit findet seine Grenze darin, dass die im Wiki aufgeführten Diskussionen in der Mehrheit von Menschen geführt wurden deren Muttersprache nicht Englisch ist, deswegen büßt ein Index, der auf einem Wort-Corpus sich mit akademischen amerikanischem Englisch basiert, ein wenig von seiner Aussagekraft ein. Die durch Wikipedia empfangene Recherche bietet viele gute Ergebnisse nur selten wurden unpassende Beiträge gefunden. Ein unpassender Beitrag ist in Abbildung 6.2 zu erkennen, welcher zu dem Argument *"Regarding that, one can determine the suitability for the study generally only during the study. Many students fail because they lack self-motivation, which is secondary in during school weakens the claim that I do not see any other way to determine whether one is able to be successfull at computer science studies."* gehört. Offensichtlich ist hier nicht ein reinrassiges Rennpferd gemeint. Diese Fehl-

for determine
 Determine
 American-bred Thoroughbred racehorse
 Determine, was an American Thoroughbred race horse. In a racing career which lasted from 1953 through 1955, the California-trained colt ran forty-four times and won eighteen races. His best season was 1954 when he became the first gray horse to win the Kentucky Derby.

Abbildung 6.2: Stellt eine unpassende Wikipedia Recherche dar.

einschätzung fallen jedoch sehr selten auf. Da sonst meistens ambivalente Artikel bestehen. Schlüsselwörter, die von der *IBM* Schnittstelle kommen sind sehr gut und seltens fehlerhaft. Die Kategorien sind nur wirklich überzeugend wenn eine Relevanz von über 0.9 angegeben wird, dies gilt auch für die Konzepte. Alles in Allem ist es gelungen eine Vielzahl von guten Ansätzen zu finden und umzusetzen, die ein Verstehen von Argumenten unterstützen.

Kapitel 7

Zusammenfassung

Diese Arbeit setzt sich mit der Erstellung einer automatisierten Wiki für Argumentnetzwerke auseinander. Dazu wurde eine *Web Application* in der Programmiersprache Python entwickelt, mit zur Hilfenahme des *web frameworks Django*. Diese ermöglicht es, Argumentnetzwerke, im Speziellen diese die von D-BAS zur Verfügung gestellt wurden. In einer sinnvollen Ordnung darzustellen. Durch farbliche Unterstützung leicht konsumierbar zumachen. Durch errechnete Indizes, Möglichkeiten zur schnellen Einschätzung des Ausgangs einer Diskussion herzustellen. Darüber hinaus vielfältige automatisch generierte Daten mit sinnstiftenden Diagrammen darzustellen. Außerdem wird eine hohe Verständlichkeit der Argumente sichergestellt durch das Recherchieren bei Informationsquellen wie zum Beispiel Wikipedia und durch die Möglichkeit Argumente vorlesen zu lassen. Des Weiteren bieten die vielen implementierten Übersichtsseiten neue Einblicke in den Zusammenhang von Argumenten, die ohne die Darstellung nicht möglich gewesen wäre. Einschränkend muss festgestellt werden, dass sich viele der bisher implementierten Funktionen lediglich auf eine Nutzung mit englischsprachigen Argumenten beziehen.

7.1 Ausblick

Die Funktionalitäten sollten auch für die deutsche Sprache zugänglich sein. Des Weiteren wäre eine Weiterentwicklung der Lesbarkeitseinschätzungen wie in Quelle [rea] diskutiert zu empfehlen. Der örtliche Bezug der Nachrichten stellt eine Verbesserung dar, die nicht nur relevantere Nachrichten ermöglichen würde, sondern auch das Erkennen von relevanten

Events. Zusätzlich würde eine Verbindung zu EDEN [CM] sinnvoll sein, da dann alle durch EDEN gesammelten Argumente direkt mit Wikieinträgen recherchierbar gemacht werden. Als weitere Aufgabe könnte man sich das automatische Erstellen einer Art Zusammenfassung vornehmen, sodass Argumente, in Argumentsystemen direkt eine Kurzübersicht bieten könnten. Außerdem könnte der Bewertungsalgorithmus angepasst werden, sodass er mehr Belastbarkeit erlangt, ein Beispiel wäre das in Betrachtziehen der *User Voting*s um die finale Stärke eines Argumentes zusätzlich anzupassen.

Literaturverzeichnis

- [aca] *Academic Vocabulary List*. <https://www.academicwords.info/>, Abruf: 18.08.2019.
- [AG13] ADRIAN GROZA, Sergiu I.: Enacting Social Argumentative Machines in Sematic Wikipedia. (2013).
- [AH13] ADRIAN HOLOVATY, et a. Jacob Kaplan-Moss: *The Django Book Release 2.0*. 2013
- [ari] *Wikipedia Automated Readability Index Tabelle*. https://en.wikipedia.org/wiki/Automated_readability_index, Abruf: 18.08.2019.
- [ART] ANTONIO RAGO, Kristijonas C.; TONI, Francesca: Adapting the DF-QuAD Algorithm to Bipolar Argumentation.
- [cjsa] *Chart.js Dokumentation*. <https://www.chartjs.org/docs/latest/>, Abruf: 18.08.2019.
- [cjsb] *Chart.js Homepage*. <https://www.chartjs.org/>, Abruf: 18.08.2019.
- [CM] CHRISTIAN METER, Martin M. Alexander Schneider S. Alexander Schneider: EDEN: Extensible Discussion Entity Network.
- [dja] *Django Homepage*. <https://www.djangoproject.com/>, Abruf: 18.08.2019.
- [dsv] *Python package django-search-views*. <https://github.com/inmagik/django-search-views>, Abruf: 27.08.2019.

- [erd] *Beispiele aus der SDK Dokumentation.* <https://github.com/EventRegistry/event-registry-python/blob/master/eventregistry/examples/QueryArticlesExamples.py>, Abruf: 18.08.2019.
- [erp] *Python SDK zur Nutzung von Event Registry.* <https://github.com/EventRegistry/event-registry-python>, Abruf: 18.08.2019.
- [Fle] FLESCHE, Rudolf: *How to Write Plain English.* https://web.archive.org/web/20160712094308/http://www.mang.canterbury.ac.nz/writing_guide/writing/flesch.shtml, Abruf: 18.08.2019.
- [fre] *Reading Ease Wikipedia.* https://en.wikipedia.org/wiki/Flesch_T1_textendashKincaid_readability_tests, Abruf: 18.08.2019.
- [fwe] *Liste der 500 meistgenutzten englischen Wörtern.* <https://www.sketchengine.eu/wp-content/uploads/word-list/english/english-word-list-total.csv>, Abruf: 18.08.2019.
- [gql] *Graph QL Homepage.* <https://graphql.org/>, Abruf: 18.08.2019.
- [ibma] *IBM Cloud Natural Language Understanding.* <https://cloud.ibm.com/apidocs/natural-language-understanding>, Abruf: 18.08.2019.
- [ibmb] *IBM Cloud Text to Speech.* <https://cloud.ibm.com/apidocs/text-to-speech>, Abruf: 18.08.2019.
- [ibmc] *IBM Python SDK.* <https://github.com/watson-developer-cloud/python-sdk>, Abruf: 18.08.2019.
- [ISSA14] ISSA (Veranst.): *A Formal Model Of Conductive Reasoning.* 2014
- [LA] LEILA AMGOUD, Dragan D.: *Gradual Semantics Accounting for Varied-Strength Attacks.*
- [lin] *Linsear Write Wikipedia.* https://en.wikipedia.org/wiki/Linsear_Write, Abruf: 18.08.2019.


- [new] *News API*. <https://newsapi.org/docs/endpoints/sources>, Abruf: 27.08.2019.
- [PJL18] PETER J. LIU, Mohammad Saleh et al.: *Generating Wikipedia By Summarizing Long Sequences*, 2018.
- [pyT] *textstat 0.5.6*. <https://pypi.org/project/textstat/>, Abruf: 18.08.2019.
- [rea] *Flesch Reading Ease*. <http://www.readabilityformulas.com/flesch-reading-ease-readability-formula.php>, Abruf: 04.08.2019.
- [req] *Python Package Requests*. <http://2.python-requests.org/en/master/>, Abruf: 18.08.2019.
- [ris] *Readability Formulas*. <http://www.readabilityformulas.com/>, Abruf: 04.08.2019.
- [Sel14] SELINGE, Marcin: *Towards Formal Representation and Evaluation of Arguments*. (2014).
- [Sel16] SELINGER, Marcin: *Argument Evaluation Based on Proportionality*. (2016).
- [TK] TOBIAS KRAUTHOFF, Christian M.: *Dokumentation der DBAS API v2*. <https://dbas.cs.uni-duesseldorf.de/docs/api/v2.html>, Abruf: 18.08.2019.
- [war17] *Computational Argumentation Quality Assessment in Natural Language*. 2017
- [wik] *Wikipedia Wiki*. <https://de.wikipedia.org/wiki/Wiki>, Abruf: 18.08.2019.
- [Yan91] YANAL, Robert J.: *Independent Reasons*. 1991

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Düsseldorf, 02. September 2019

Stephan Linzbach



Hier die Hülle
mit der CD/DVD einkleben

Diese CD enthält:

- eine *pdf*-Version der vorliegenden Bachelorarbeit
- die \LaTeX - und Grafik-Quelldateien der vorliegenden Bachelorarbeit samt aller verwendeten Skripte
- **[anpassen]** die Quelldateien der im Rahmen der Bachelorarbeit erstellten Software XYZ
- **[anpassen]** den zur Auswertung verwendeten Datensatz
- die Websites der verwendeten Internetquellen