Bachelorarbeit am Institut für Informatik der Freien Universität Berlin

Human-Centered Computing (HCC)

Konzeptannotation von Ideen basierend auf WordNet

Ingrid Gancia Tchilibou Boudjeka

Matrikelnummer: 5020686

ganciatchilibou@yahoo.fr

Betreuer: Herr Maximilian Mackeprang

Erstgutachterin: Prof. Dr. Claudia Müller-Birn

Zweitgutachter: Prof. Dr. Lutz Prechelt

Berlin, 09.03.2020

b

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder Ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 09.03.2020

Ingrid Tchilibou

Zusammenfassung

Die Zusammenfassung und Kategorisierung einer großen Anzahl an Ideen um neue, interessante Ideen zu generieren ist ein vielversprechender Prozess, der auch Ideensynthese genannt wird. Allerdings sind die gesammelten Ideen meist sehr umfangreich, was bei der manuellen Kategorisierung sehr viel Zeit in Anspruch nimmt. Software kann Analysten dabei helfen, große Mengen zu überblicken, um schneller zu kategorisieren und besser neue Ideen zu synthetisieren. Diese Arbeit befasst sich mit der Frage, ob WordNet-Hypernyme hilfreich sind um eine Gruppe von Ideen zu kategorisieren. Um WordNet-Hypernyme zu verwenden ist es notwendig, dass wir die Ideen zunächst mit der Bedeutung (Synset) jedes in einer Idee enthaltenen Wortes annotieren (sog. Word-Sense-Disambiguierung). Zu diesem Zweck wurde in dieser Arbeit eine WordNet-Anbindung für das bestehende ICV (Interactive Concept Validation)-Tool implementiert. Zusätzlich wurden mehrere Möglichkeiten der Kategorisierung implementiert und in einem Experten-Interview getestet. Durch das Interview konnte gezeigt werden, dass Hypernyme nicht in Bezug auf die Anzahl der Ideen, die als Kategorien gruppiert werden, betrachtet werden können. Andererseits zeigte das Interview, dass Hypernyme helfen eine globale Vorstellung der Ideen zu entwickeln. Diese Erkenntnisse bieten die Grundlage für die Weiterentwicklung und Verbesserung von WordNet-basierten Kategorisierungs-Ansätzen.

Abstract

Summarizing and categorizing a large quantity of ideas to generate new and interesting ideas is a promising process, also called idea synthesis. However, the collected ideas are usually very extensive, which takes a lot of time when categorizing them manually. A Software could help analysts to oversee large quantities in order to categorize faster and better synthesize new ideas. This thesis deals with the question on whether WordNet hypernyms are helpful in categorizing a group of ideas. To use WordNet hypernyms it is necessary to first annotate the ideas with the meaning (synset) of each word contained in an idea (so-called Word-Sense-Disambiguation). For this purpose a WordNet connection for the existing ICV (Interactive Concept Validation) tool was implemented in this thesis. Furthermore, several possibilities of categorization were implemented and tested in an expert interview. The interview showed that hypernyms cannot be considered in terms of the amount of ideas that are grouped as categories. On the other hand, the interview showed that hypernyms can help in giving an orientation on the main context in the ideas. These observations provide a base for the further development and improvement of WordNet-based categorization approaches.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Vorwort

> „*Zusammenkommen ist ein Beginn, Zusammenbleiben ist ein Fortschritt, Zusammenarbeiten ist ein Erfolg.*“ Henry Ford

Die vorliegende Bachelorarbeit entstand im Rahmen meines Informatikstudiums an der Freie Universität Berlin. Während meines Studiums habe ich mich immer für Datenanalyse und Künstliche Intelligenz interessiert. Deshalb habe ich die Entscheidung getroffen, meine Arbeit in einem dieser Bereiche der Informatik zu schreiben. Mit Hilfe meines Betreuers Herrn Maximilian Mackeprang und der Erstgutachterin Prof. Dr. Claudia Müller-Birn konnten wir das Thema meiner Bachelorarbeit formulieren.

Besonders danken möchte ich meinem Betreuer Herrn Maximilian Mackeprang, der mich während dieser ganzen Phase begleitet hat, indem er mir jedes Mal Tipps und wertvolle Ratschläge für den Fortgang dieser Aufgabe gab. Ich möchte Ihnen für die Zeit und die Energie danken, die Sie in jede Lektüre und jedes erneute Durchblättern meiner Arbeit investiert haben, um mir jedes Mal Tipps zu geben und für Ihre Hilfe beim Aufbau der Struktur meiner Arbeit. Ich danke ebenfalls meiner Erstgutachterin Prof. Dr. Claudia Müller-Birn und meinem Zweitgutachter Prof. Dr. Lutz Prechelt.

Außerdem möchte ich mich bei meinen beiden Freunden Luis H. und Bernadeta C. bedanken, die sich bereit erklärt haben, meine Arbeit zu lesen und mir ein Feedback zum Aufbau der Sätze und zum Inhalt gegeben haben. Ich möchte auch Anja C. danken, die mir bei der Formatierung meiner Arbeit geholfen hat.

Mein herzlicher Dank gilt schließlich meiner Mutter und meinem Vater, die mich trotz der Distanz in dieser Phase unterstützt und ermutigt haben.

1 Einleitung

1.1 Thema und Kontext

Ideen im großen Maßstab (Engl.:*Large Scale Ideation*) beschreibt die Sammlung von großen Mengen von Ideen, mit dem Ziel innovative und einzigartige Ideen zu generieren und zu fördern. Innerhalb der Informatik wird dieses Thema im Hinblick auf die Unterstützung mit Algorithmen und Computer-Systemen in einem Mensch-Computer-Umfeld erforscht. Allerdings bringt der große Maßstab eigene Probleme mit sich, allem voran im Umgang mit großen Mengen an Ideen. Dies macht es für Beteiligte sehr schwer, die Ergebnisse eines Anwendungsfalls (Sammlung von Ideen zu einem bestimmten Thema) auszuwerten, insbesondere wenn die Auswertung manuell erfolgt. Ein Beispiel für ein Large-Scale Ideation Projekt ist das "Ideas-to-Market"-Projekt[Com20], in dem Ideen zu neuen Technologien gesammelt werden. Im Anwendungsfall "Bionic Radar" mussten die Projektbeteiligten insgesamt 581 Ideen verarbeiten, um diese zu verstehen und aus ihnen ein mentales Modell der möglichen Anwendungen zu finden. Das erfordert jedoch ein enormes Budget an Zeit und Aufwand. Ein weiteres Beispiel für ein Large-Scale Ideation Projekt ist Cambridge 2016, in dem 60 Repräsentanten drei Monate lang 20 Lösungen aus 548 Ideen erzeugt haben[Sia17]. Der längste und wichtigste Teil dieser Studie war das Verständnis der Ideen.

In der Forschung wird das Verstehen und Zusammenfassen von Ideen als *Solution Synthesis* bezeichnet, bei der die Synthesizer (Experten, Hauptakteure) alle Ideen lesen, bewerten und dann die am besten geeigneten Ideen synthetisieren, gruppieren und daraus Lösungen generieren. Eine mögliche Lösung für das Problem des Zeitaufwands wäre nach Ansicht der Forschung[Sia17], von Anfang an einen Überblick über die Ideen zu haben. Mit einem Überblick über die Ideen würde es den Synthesizern leichter fallen, Untergruppen zu erstellen und so bei sich wiederholenden Ideen Zeit zu sparen. Es existieren bereits verschiedene Ansätze, die dieses Problem zu lösen versuchen. So gibt es beispielsweise Plattformen wie OpenIdeo[1] und Idea.starbucks[2], die Bewertungsmechanismen vorschlagen, um Ideen zu kategorisieren. Die Ideen werden hier jedoch über einen Abstimmungsmechanismus gruppiert, das heißt es wird nur die Beliebtheit der Ideen betrachtet - genauer gesagt die Ideen, die die meisten Stimmen erhalten. Das hilft den Synthesizern nicht, da die beliebtesten Ideen

[1] https://openideo.com/
[2] https://ideas.starbucks.com/

nicht immer die besten oder die von Experten ausgewählten Ideen sind und durch Beliebtheit die seltensten Ideen vernachlässigt werden, die oft wichtige Elemente für Lösungen des gegebenen Projekts enthalten. Ein Überblick über alle Ideen wäre von großem Vorteil, um Entscheidungs- und letztendlich Lösungsfindungen für Synthesizer zu erleichtern und die Wahrnehmung des Benutzers zu unterstützen[Sia17].

Es gibt in der Informatik verschiedene Ansätze, um Ideen für Computer analysierbar zu machen, welche die Verarbeitung dieser Daten in Zukunft stärker automatisieren könnten. Ein Beispiel ist die Verwendung von sogenannten Satz-Einbettungen (Sentence-Embeddings), die Ideen-Texte in einen hochdimensionalen Raum einordnen. Ein Problem an diesem Ansatz ist allerdings, dass Informationen über die Strukturen der Ideen nicht mehr abrufbar sind, beispielsweise der Abstraktionsgrad einer Idee [GCN+18]. Ein weiteres Problem ist die sogenannte Word-Sense-Disambiguierung: Eine Nennung des Begriffs "Keyboard" könnte entweder ein Musik-Instrument oder ein Computer-Eingabegerät meinen. Satz-Einbettungen können diese semantische Uneindeutigkeit nicht auflösen [MMBS19].

Die sogenannte interaktive Konzeptvalidierung (Interactive Concept Validation, kurz ICV) bietet einen Ansatz, um Ideen mit semantisch eindeutigen Konzepten anzureichern[MMBS19]. Ein Beispiel für die praktische Umsetzung dieses Ansatzes ist z.B. das vom Human Computing Center (HCC) der Freien Universität Berlin entwickelte ICV-Tool[3]. Dabei wurden für das Tool bisher DBpedia als Datenquelle und Wikidata für die Superklassen verwendet. DBpedia ist eine Datenbank von strukturierten Informationen, die von der Öffentlichkeit und aus den verschiedenen Wikimedia-Projekten erstellt wurden. Diese Informationen sind in Form eines Wissensgraphen gruppiert und öffentlich zugänglich[4]. Wikimedia ist eine Reihe von Projekten wie Wikipedia, Wikidata usw., die öffentlich zugänglich sind und darauf abzielen, freie und pädagogische Inhalte in die Welt zu bringen. Das ICV-Tool benutzt die Verbindung des Textes mit Konzepten aus sogenanten "*General Knowledge Graph*"s: Graphdatenbanken, die Wissen über die Welt speichern und über Abfragesprachen verfügbar machen. Wikidata ist ein offener "*General Knowledge Graph*" auf Freiwilligen-Basis. Die Erstellung von Daten und Verbindungen auf Freiwiligen-Basis hat Vor- und Nachteile. Ein Nachteil ist, dass die Qualität schwankt. Dadurch ist die Qualität der Superklassen sehr unterschiedlich. Außerdem können Superklassen in Wikidata Zyklen bilden, was die algorithmische Analyse schwer macht. Der Ansatz dieser Arbeit ist es daher, Wikidata durch das an der Princeton University entwickelte WordNet zu ersetzen. Die Annahme ist, dass WordNet, dadurch dass es von Experten erstellt und gepflegt wird, Verbindungen höherer Qualität aufweist.

[3] https://github.com/FUB-HCC/Innovonto-ICV
[4] https://wiki.dbpedia.org/

1.2 Zielsetzung der Arbeit

Ein wichtiger Schritt im Prozess der *Solution Synthesis* ist es, gesammelte Ideen zunächst zu ordnen und zu kategorisieren. Hierbei könnten ICV-Methoden, welche die Stärken von WordNet nutzen, sinnvoll sein. Die Hauptforschungsfrage für diese Arbeit ist daher: Wie können Ideen mit Hilfe von WordNet annotiert und mit Hilfe von Hypernymen kategorisiert werden, so dass ein Benutzer einen Überblick über eine bestimmte Kategorie von Ideen gewinnen kann?

Das Hauptziel wird in drei untergeordnete Fragestellungen unterteilt:

1. Wie kann WordNet in die bestehende ICV Software integriert werden?
2. Wie können aus WordNet Hypernym-Kategorien für Ideen erstellt werden?
3. Wie hilfreich sind Hypernym-Kategorien um einen Ideen-Datensatz zu verstehen?

1.3 Vorgehen bei der Umsetzung

Der Hauptfrage von dieser Arbeit soll beantwortet werden, indem ein Beispieldatensatz manuell annotiert und dann analysiert wird. Der Ausgangsdatensatz soll aus Ideen aus dem oben erwähnten Anwendungsfall "Bionic Radar" bestehen, um einen Vergleich zwischen manueller und automatisierter Kategorisierung zu ermöglichen. Der Arbeitsprozess gliedert sich dabei in fünf Hauptphasen mit entsprechenden Unterschritten:

1. Um Ideen-Texte mit WordNet zu verknüpfen, wird ein neues Backend für das "Interactive Concept Validation" (ICV Tool) des HCC implementiert. Dieses Backend weist folgende Funktionalitäten auf:
    - Tokenization: Wörter, Sätze und Absätze werden erkannt und gruppiert.
    - Stop Word Filtering: Wörter, die nicht dazu beitragen den Text zu verstehen, werden eliminiert.
    - Bedeutungs-Kandidaten, *Synset-Candidates* gennant werden gesucht, indem die Synset-ID und die Beschreibung des Synsets aus WordNet übernommen wird.
2. Mit der neuen Implementierung werden die Synsets von Ideen gesammelt. Die Synsets werden mithilfe des bestehenden ICV-tools disambiguiert.
3. Für die annotierten Ideen werden die Hypernyme aus WordNet extrahiert.
4. Die Ergebnisse der Annotation und Extraktion werden visualisiert, um fünf Aspekte zu untersuchen:

- Wie ist die Qualität der Hypernyme in WordNet einzuschätzen?
- Wie viele Top-Level Hypernyme (direkt unter "Entity") kommen im Datensatz vor?
- Wie viele 2nd-Level Hypernyme kommen vor, und pro Hypernym, wie viele Ideen werden damit beschrieben?
- Können wir über WordNet gleichformige Unterteilungen eines Ideen-Raums als Heuristik für Kategorien erzeugen?
- Ist diese Zusammenfassung Hilfreich um einen schnellen Überblick über einen Ideen-Raum zu erlangen?

1.4 Aufbau der Arbeit

Diese Arbeit besteht aus sieben Hauptkapiteln. Nach einer allgemeinen Einführung werden in Kapitel 2 eine kurze Erläuterung der schon existierenden Methoden und Abgrenzungen, sowie eine Einführung in WordNet vorgenommen. Danach wird in Kapitel 3 der Entwurf von allen Elementen, die für die Realisierung dieses Projekts notwendig sind, sowie die zu verwendenden Werkzeuge detailliert beschrieben. Zusätzlich wird im gleichen Kapitel die API dokumentiert, die die Annotation der in den Ideen enthaltenen Wörter erlaubt, sowie die Code-Struktur mit einigen wichtigen Teilen des Codes, die die Annotation eines Textes mit WordNet erlaubt und wie die Hypernymen bzw. Kategorie extrahiert werden. Des Weiteren wird zuerst in Kapitel 4, der Anwendungsfall Bionic Radar beschrieben und danach mit Hilfe des neuen erstellten ICV-Backend ein Annotationsbeispiel aufgezeigt.

Darüber hinaus werden wir in diesem Kapitel die erhaltenen Elemente mit Hilfe des Hypernyms kategorisieren und anschließend eine Deskriptive Analyse über die erhaltenen Ergebnisse durchführen. Um festzustellen, ob unser Ziel erreicht wurde, haben wir mit einem Experten für Datenanalyse ein Interview durchgeführt und wir werden in Kapitel 5 diese Evaluation vorstellen. Danach werden wir in der Diskussion 6 einen Vergleich mit unserem Ziel durchführen und eine allgemeine Diskussion der Ergebnisse. Am Ende werden wir in Kapitel 7 unsere Arbeit zusammenfassen. Eine graphische Übersicht über einen Aufbau dieser Arbeit ist in der Abbildung 1.1 dargestellt.

Bei den nachfolgend verwendeten, persönlichen Ausdrücken ist die männliche Schreibweise gewählt worden, damit die Lesbarkeit der Arbeit erhöht wird. Darüber hinaus werden eine Reihe englischer Begriffe verwendet, um einerseits dem interessierten Leser das Studium der häufig vorhandenen englischen Original-Literatur zu erleichtern und andererseits eine Verzerrung bestehender Fachbegriffe durch die Übersetzung zu vermeiden. Vom herkömmlichen Text werden diese Begriffe durch Kursivschrift unterschieden.

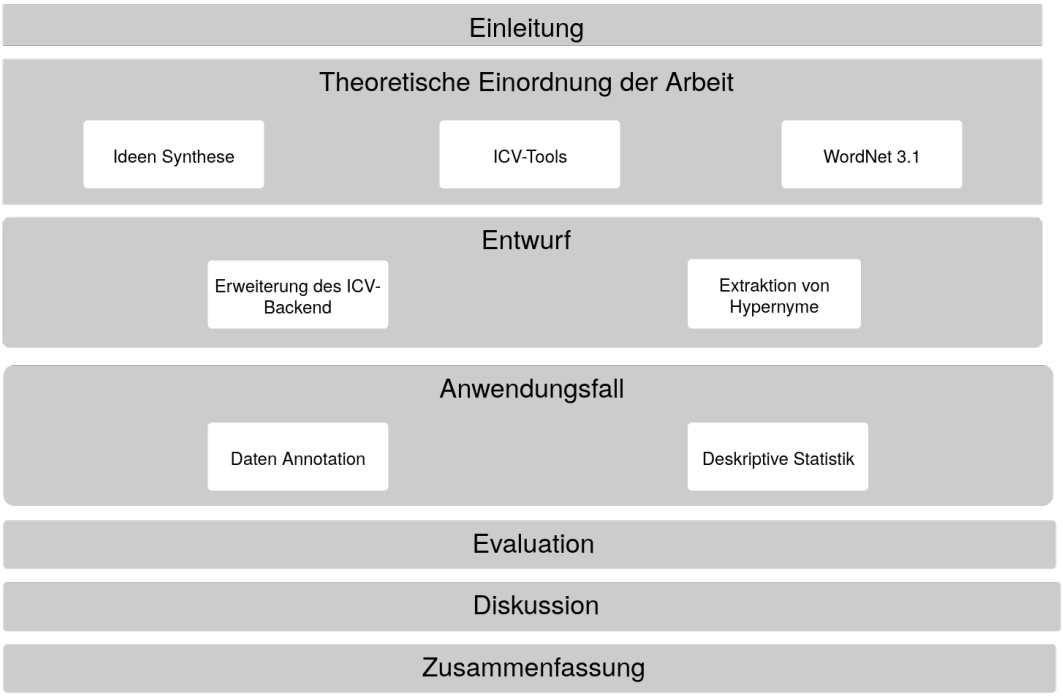


Abbildung 1.1: Aufbau der Arbeit (vgl. Beschreibung Abschnitt 1).

2 Theoretische Einordnung der Arbeit

In diesem Kapitel werden wir die verschiedenen Forschungsprojekte, die bereits zur Problematik dieser Bachelorarbeit durchgeführt wurden, und ihre Beziehung zu den Forschungen, die wir in dieser Hinsicht durchführen wollen, im Detail beschreiben. Darüber hinaus werden wir in diesem Kapitel auch die WordNet-Datenbank vorstellen und in wenigen Worten die Gründe erklären, warum wir uns für diese Datenbank entschieden haben. Am Ende werden wir die verschiedenen Analysepunkte, die wir bei dieser Bachelorarbeit verwenden werden, und ihre Quelle im Detail beschreiben.

2.1 Ideen-Analyse in Large Scale Ideation

Durch die große Anzahl an generierten Ideen in *Large Scale Ideation Challenges* (Anwendungsfälle) ist es ökonomisch nicht sinnvoll, alle Ideen im Detail zu lesen.

Laut Siangliulue[Sia15] ist ein weiteres Problem, dass Teilnehmer an Anwendungsfällen oft simple und repetitive Ideen einreichen.

Ein Ansatz dieses Problem abzumildern ist es, Ideen schon während der Generierung zu kategorisieren. Beispielsweise wird im CrowdMuse System von Girotto et al.[GWB19] die manuelle Kategorisierung genutzt, um eine Matrix-Visualisierung des Ideen-Raumes zu erzeugen. Die Manuelle Kategorisierung in diesem Projekt passierte aber auf reiner Wort-Basis, ohne zusätzliche Kontextinformationen (wie z.b. Hypernyme oder Superklassen).

Bei der Annotation von Ideen handelt es sich um einen Ansatz, Ideen für Computer verständlich zu machen, indem Terme (einzelne Wörter wie "Keyboard" oder zusammengesetzte Wörter wie "pet food") in den Ideentexten mit eindeutigen Konzepten in einem Knowledge-Graph (KG) verbunden werden. Ein Knowledge Graph organisiert verschiedene reale Einheiten, sogenannte Konzepte, mit ihren Beziehungen in einem Graphen. Es stellt auch ein Schema zur Verfügung, das diese Konzepte in Klassen (abstrakte Konzepte) gruppiert, die auch Beziehungen zueinander haben[RP17]. Dieses beschreibt die Kontextinformationen zwischen Konzepten durch verschiedene mögliche Beziehungen (Bsp. : 'is-hypernym,' 'is-holonym,' 'is-a-part-of,' 'is-synonym') [MMBS19].

2.2 *Interactive Concept Validation* (ICV)

Die Interactive Concept Validation (ICV) ist eine von der Human-Centered Computing (HCC)-Gruppe entwickelte Software, um die Annotation einer Idee zu ermöglichen. Das ICV-Tool empfängt die Ideen des Benutzers und schlägt für jede Idee vor, diese manuell zu annotieren. Die Extraktion und Annotation von Konzepten basiert auf dem DBpedia-Wissensgraphen[MKS+19]. Mit Hilfe von DBPedia bietet ICV mehrere Annotationsmöglichkeiten für jedes Konzept. Das heißt, für jedes ausgewählte Konzept werden die verschiedenen bestehenden Bedeutungen des Konzepts aus der DBPedia Datenbank vorgeschlagen, entweder als eine Serie von Bildern, als eine Serie von Beschreibungen oder als Kombination Bildern und Beschreibungen[MKS+19]. Der Benutzer kann dann unter den verschiedenen Vorschlägen einen oder mehrere auswählen, die seiner Meinung nach das gewählte Konzept am besten beschreiben. Wenn keiner der vorgeschlagenen Kandidaten passt, bietet ICV auch die Möglichkeit, zum nächsten Konzept überzugehen. Dadurch werden fehlerhafte Daten minimiert, was die spätere Analyse vereinfacht. Obwohl die Ideen-Annotation zum Verständnis der Ideen beiträgt, hilft sie nicht, sie automatisch zu kategorisieren. Wie in der Einleitung erwähnt, wird im ICV-tool bisher DBPedia als Datenquelle und Wikdiata als Quelle für Superklassen im Backend verwendet. Durch die deutlichen Qualitätsunterschiede in den Superklassen von Wikidata sowie die Möglichkeit von Zyklen in den Superklassen, können die durch die Annotation gewonnenen Daten nur schwer analysiert und weiterverwendet werden.

2.3 WordNet 3.1 : "*A* Lexical Database for Englisch"

Um das Problem der manuellen Kategorisierung zu lösen, werden wir in dieser Arbeit die WordNet-Datenbank verwenden, um Ideen zu annotieren und zu kategorisieren. WordNet[Mil98] ist ein freies, kostenloses, semantisches Netzwerk, das aus einer umfangreichen lexikalischen Datenbank in englischer Sprache besteht. In WordNet gibt es mehrere Arten von Konzepten: Begriffe (aus dem Lexikon) und die Bedeutung der Begriffe (genannt Synset). Die Zusammenfassung von Namen, Verben, Adjektiven und Adverbien in eine Gruppe von kognitiven Synonymen (Synsets), um ein bestimmtes Konzept auszudrücken, ist sehr gut geeignet, um die semantischen und lexikalischen Beziehungen von Wörtern zu verdeutlichen[Uni09]. Diese ermöglicht es, ein Wort in einem Kontext zu klassifizieren.

Zum Beispiel kann das Wort "*Framework*" entweder eine hypothetische Beschreibung einer komplexen Einheit oder eines komplexen Prozesses[Synset 01], die zugrunde liegende Struktur[Synset 02] oder eine Struktur, die etwas trägt oder enthält[Synset 03], sein (Abbildung 2.1). Anstatt Konzepte um ihre lexikalische Form herum zu gruppieren, gruppiert WordNet durch Synsets

Konzepte nach ihrer Bedeutung im Kontext. WordNet definiert dadurch zwei Arten von Beziehungen:

- Zwischen einem Synset und den Begriffen, mit denen es im Zusammenhang bezeichnet wird.
- Zwischen einem Synset und seiner Definition im Kontext.

Darüber hinaus gibt es eine weitere Art der Beziehung zwischen Synsets:

- Hypernymie (Hyperonymie) und Hyponymie: Hypernym ist der Oberbegriff eines Wortes X. Wenn Y in X enthalten ist, dann ist Y das Hyponym von X und X ist das Hypernym von Y. Z.B. sind 'Hund', 'Katze', 'Schaf', 'Maus' Hyponyme von 'Tier' und 'Tier' ist Hypernym von 'Hund'.
- Meronymie und Holonymie: Meronym ist ein Wort X dessen Bedeutung einen Teil eines Wortes Y umfasst. Holonyme sind das Gegenteil davon. Z.B. ist 'Blatt' ein Meronym von 'Baum' und 'Baum' ist ein Holonym von 'Blatt'.

Schließlich ist unter den verschiedenen Beziehungen zwischen den von WordNet vorgeschlagenen Synsets Hypernymie angesichts seiner Definition dasjenige, das uns bei der Kategorisierung von Ideen am besten helfen kann.

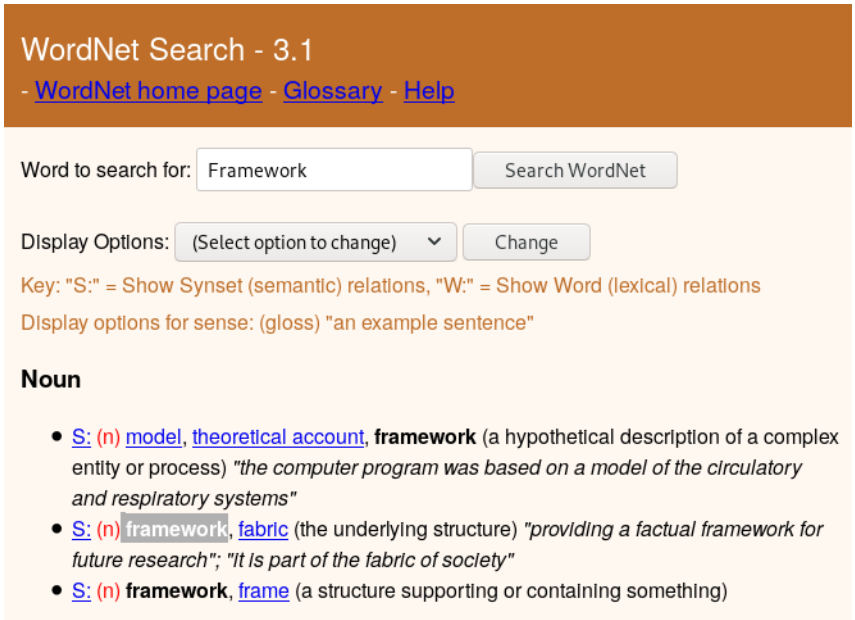


Abbildung 2.1: Ausgabe Synsets von Framework im WordNet 2

2.4 Analyse und Visualisierung

Für das Annotieren von Synsets, die in Ideen-Texten vorgefundenen wurden, liegen prinzipiell zwei Möglichkeiten der Analyse nahe. Zum einen werden Hypernyme verwendet um die Kategorisierung der Ideen zu ermöglichen und zum anderen wird der Abstraktionsgrad einer Idee über Hypernyme berechnet.

1. Die Analyse über Hypernyme ist Inspiriert durch die Arbeit von Gilon et al.[GCN+18]. In ihrer Forschung haben Gilon et al. Cyc-Superklassen (also Hypernym-ähnliche Strukturen) verwendet, um ein Suchwerkzeug von Ideen für Professionelle Designer zu implementieren. Cyc ist eine umfassende allgemeine Wissensbasis, die darauf abzielt, die Anwendung künstlicher Intelligenz auf die Logik wie ein Mensch zu ermöglichen, indem kontrollierte Begriffe mit Schlüsseleigenschaften bereitgestellt werden. Cyc bietet Superklassen von Wörtern. Zum Beispiel ı *#$DomesticatedAnimal #$CanisGenus* ȷ sind Oberklassen von "*Hund*". Die Autoren benutzen die von Cyc bereitgestellten Superklasse, um Ideen zu kategorisieren. Er versucht, ein Wort wieder auf eine abstrakte Ebene zu bringen, indem er seine Haupteigenschaften beibehält, die es ihm erlauben, es mit einem anderen Wort zu globalisieren[GCN+18]. Unsere Vorgehensweise ist ähnlich. Mit Hilfe des Hypernyms werden wir die verschiedenen Ideen kategorisieren, indem wir versuchen, jedes von einer Idee bereitgestellte Synset auf seinem höchsten Abstraktionsgrad zu bringen, um es mit anderen, in anderen Ideen enthaltenen Synsets zu gruppieren. Wir definieren eine Kategorie als ein bliebiges Hypernym außer *entity*, *physical Entity* und *abstraction* das eine genaue Anzahl von Synsets zusammenfasst. Die Filterung von den drei vorher genannten Hypernymen ergibt sich aus deren Abstraktion: Dadurch das die drei Hypernyme alle verschiedenen Wörter (Synsets) umfassen, die es gibt, und dies ist als ein sehr abstraktes Kategorie zu betrachten und für unsere Analyse nicht hilfreich. Die Abbildung (Abbildung 2.2) zeigt exemplarisch die Hypernym-Beziehung zwischen den zwei Wörtern "*Door*" und "*Window*". Laut dieser Abbildung werden diese Wörter in "*Structur Constructor*" kategorisiert.
2. Außerdem soll der Abstraktionsgrad der Ideen analysiert werden. Georgi V. Georgiev schlägt in siner Forschung vor, Abstraktionsebenen als "*Level of Abstraction*" der WordNet-Wörter zu berechnen. Der Autor beschreibt die zugrundeliegende Idee (sinngemäß übersetzt) wie folgt: " Die Abstraktionsebene steht in der Taxonomie in einem negativen Zusammenhang mit der Tiefe des Substantivs "*entity*" das abstrakteste ist, während die tiefsten Substantive in der Taxonomie am wenigsten abstrakt sind. Das Komplement der Abstraktionsebene zur Einheit ist ein Maß für die Konkretheit des Wortes." Der Grad der Abstaktion wird wird im unserem Fall als der Durchschnitt der Abstände jeder in einer Idee

enthaltenen Synsets zum Wurzel-Hypernym "*entity*" betrachtet. "*Entity*" ist der Abstrakte Synset und hat daher einer Abstraktionsgrad von 0.

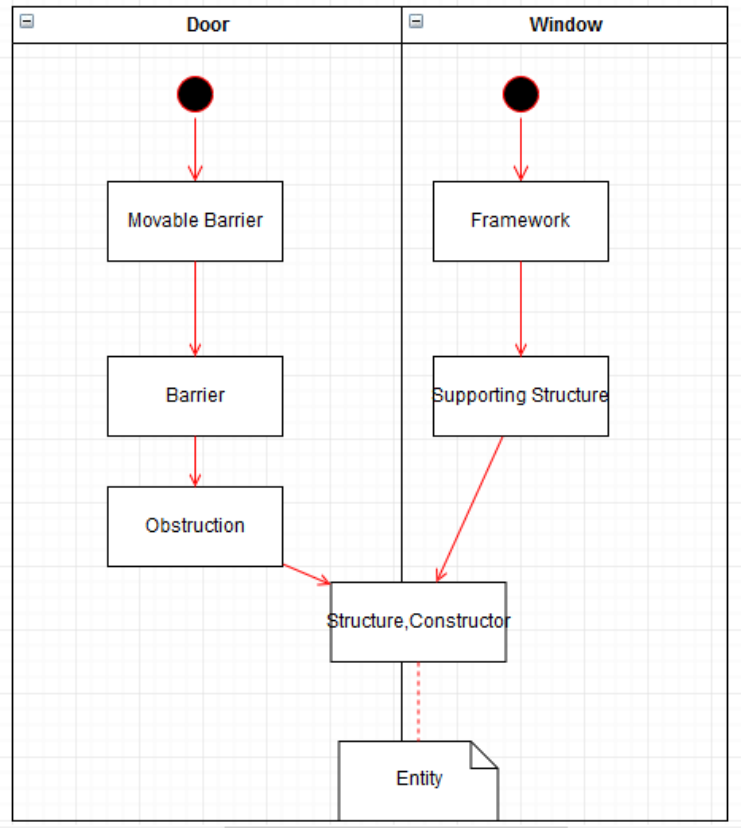


Abbildung 2.2: Beispiel einer Hypernym-Beziehung zwischen den Wörtern "*Door*" und "*Window*"

Zusätzlich zu den Analysen auf Wordnet sollen die Daten visualisiert werden. Für die Visualisierung der Daten wurden zwei Möglichkeiten ausgesucht:

- Visualisierung mit Dendogramm: die Beziehungen zwischen Hypernymen werden als Hierarchie angezeigt.
- Visualisierung durch ein Balkendiagramm: Um die Anzahl an Hypernymen auf einer bestimmten Abstraktionsebene zu visualisieren, wurden Balkendiagramme gewählt.

3 Entwurf

In diesem Kapitel werden zuerst die Software-Komponenten beschrieben, die für die Realisierung dieses Projektes verwendet wurden. Zusätzlich wird Annotation der Ideen mit Hilfe der WordNet-API detailliert beschrieben.

3.1 Erweiterung des ICV-Backend: WordNet-API

In einem ersten Schritt wurde das bestehende Backend des ICV-Tools durch eine WordNet Anbindung erweitert.

3.1.1 Werkzeuge

Für die Realisierung der WordNet API, die in das ICV Backend integriert wird, werden wir als Programmiersprache Python verwenden, genauer gesagt die Version 3.6.8. Aufgrund der zahlreichen Bibliotheken für Sprachverarbeitung wie z.B. NLTK ist Python eine geeignete Programmiersprache für die Mensch-Maschine-Interaktion und für das Textstudium. Da Python die Sprache ist, die vom Autor und Entwickler der Arbeit am Besten beherrscht wird, haben wir uns entschieden, Python für die Realisierung dieses Projekts zu verwenden. Daneben wird jupyter notebook als Editor eingesetzt und dient auch zur Visualisierung der Ergebnisse. Für die Sicherung und Verwaltung des Projekts wird der Webserver für Software-Projekte Github benutzt.

3.1.2 WordNet-API

Eine API (Application Programming Interface) ist eine Programmierschnittstelle, die den Informationsaustausch zwischen einer Anwendung und einzelnen Programmteilen ermöglicht. Die von uns entwickelte Schnittstelle ermöglicht es anderer Software, durch Anfragen transparent auf die WordNet-Daten zuzugreifen, ohne diese zu verändern. Die Antwort der Anfragen ist eine JSON-Datei. JSON (JavaScript Object Notation) ist ein kompaktes Datenformat, das Informationen in einer für den Menschen lesbaren und verständlichen Form enthält. In unserem Fall ruft das ICV-Backend die WordNet-API auf, um Annotationen von Ideen zu erhalten. Für die Realisierung dieser WordNet-API haben wir das Mini-Web-Framework flask gewählt, das durch seine vielfältigen Elemente die Entwicklung der Anwendungen erleichtert [AMM+15]. Um die

Kommunikation zwischen der WordNet-API und dem ICV-Backend zu ermöglichen wird Cross-Origin Resource Sharing (CORS) verwendet. Die Konfiguration von CORS in flask wird wie in Listing 3.1 angezeigt umgesetzt:

```
1 from flask import Flask,
2 from flask_cors import CORS
3 app = Flask(__name__)
4 CORS(app)
```

Quellcode 3.1: Anwendung von CORS und FLASK

Die Annotation der Ideen erfolgt über die in der NLTK-Bibliothek enthaltene WordNet-Bibliothek. NLTK (Natural Language Toolkit) ist eine Suite von Open-Source-Programmmodulen, Tutorials und Problemsets, die gebrauchsfertige Computerlinguistik-Materialien zur Verfügung stellen [LB02].

Die WordNet-API läuft im *localhost* unter Port 4000 und erwartet als Input einer Anfrage einen Text (eine Idee). Das Flussdiagramm in Abbildung 3.1 beschreibt den Fluss für jede empfangene Idee bis zur Erhaltung der Antwort in unserer WordNet-API. Die Antwort der API ist eine JSON-Datei mit der Struktur die in Listing 3.2 beschrieben ist.

Das in Listing 3.2 sichtbare Feld *annotation candidates* listet alle Wörter, die in der in den Parametern angegebenen Idee enthalten sind. Stopp-Wörter wie "*and*", "*or*", "*the*" und Satzzeichen sind nicht enthalten. Diese Wörter sowie Symbole, die keine nützlichen Informationen zum Verständnis der Idee enthalten, werden wie in Listing 3.3 detailliert mit Hilfe der NLTK Stopword-Bibliothek gefiltert.

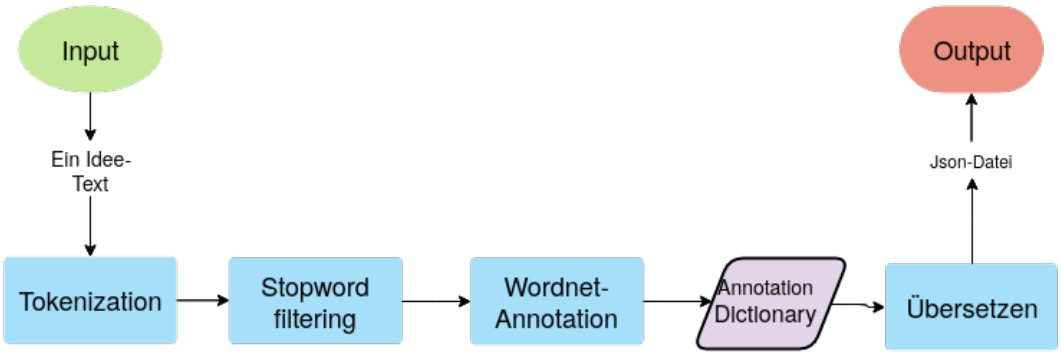


Abbildung 3.1: Flussdiagramm: Annotation-API

```
{
   "annotation_candidates":[
      {
         "offset": "Offset des Wortes in Idee Text",
         "resource_candidates":[
            {
               "description": "Beschreibung des Synset",
               "label": "Synset Name",
               "offset": "Offset des Wortes in Idee Text",
               "resource": "Id des Synset Name in WordNet",
               "source": "Synset name in WordNet",
               "text" : "Urspruengliches Wort",
               "pos": "Wortart(part of speech)"
            },
            {...},
            {...},
         ],
         "text": "Urspruengliches Wort"
      }
      {...}
      {...}
   ],
   "text": "Idee Text"
}
```

Quellcode 3.2: JSON-Ausgabestruktur der WordNet-API

```
import nltk
from nltk.corpus import stopwords
import re

def stop_words_filtering(text):
    reg_exp = r"[a-zA-Z0-9_-]+" #Regulaeare Ausdruck,der alle erlaubten Sylabeln enthaelt.
    stop_words  = set(stopwords.words('english')) #stopwoerter fuer englisches Text
    word_tokens = [ ]
    a = re.compile(reg_exp)
    word_tokens = word_tokens + a.findall(text) #loescht die Satzzeichnen im Text
    filtered_sentence = [w for w in word_tokens if not w in stop_words] #loescht die Stopp-Woerter im Text
    return filtered_sentence
```

Quellcode 3.3: Eliminieren unnötiger Wörter im Text

Danach wird jedes gefilterte Wort mit WordNet nach folgendem Verfahren annotiert:

Darüber hinaus stellt ein *resource-candidate* ein Synset-Element eines *annotation candidate* dar. Daher ist die Länge der *ressource-candidate* proportional

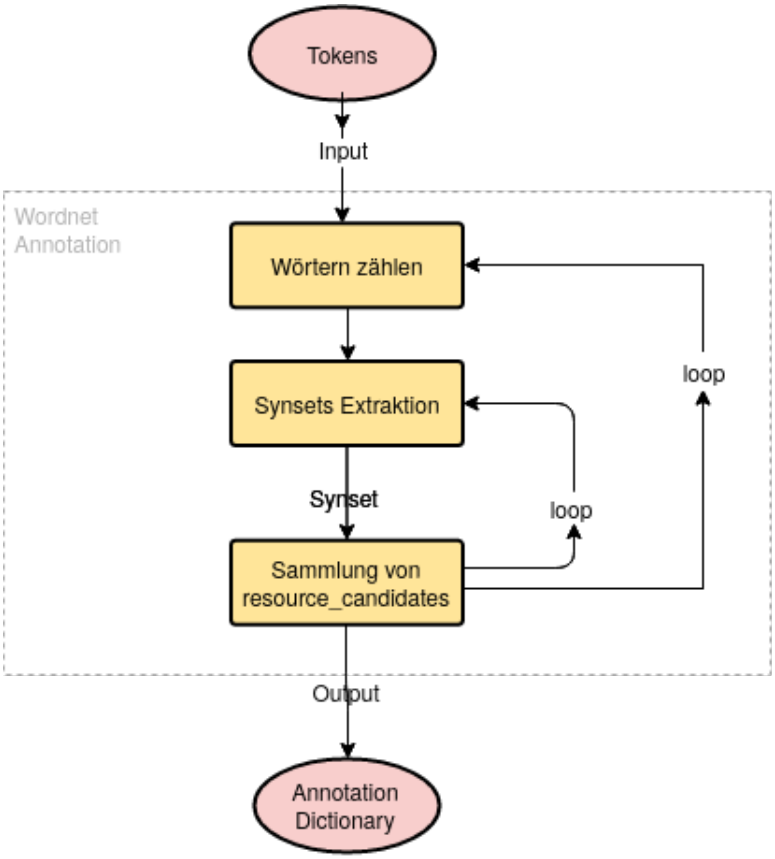


Abbildung 3.2: WordNet-Annotation

zur Anzahl der Synset-Elemente, die in einem *annotation candidate* enthalten sind. Die Funktion die im Listing 3.4 beschrieben wird, ermöglicht es, die Synsets eines Wortes von WordNet mit python zu erhalten:

```
1 from nltk.corpus import wordnet
2 def get_synsets(word):
3     synset_list = wordnet.synsets(word)
4     return synset_list
```

Quellcode 3.4: Extrahieren von Synsets aus WordNet mit Python

Nehmen wir zum Beispiel wieder das Wort "*Framework*" und extrahieren wir die Synsets über unsere API:

```
1 get_synsets('Framework')
```

Quellcode 3.5: Beispiel für Synset-Extraktion mit unserer API

Die Ausgabe ist eine Liste von Synset-Instanzen:

[Synset('model.n.01'), Synset('framework.n.02'), Synset('framework.n.03')]

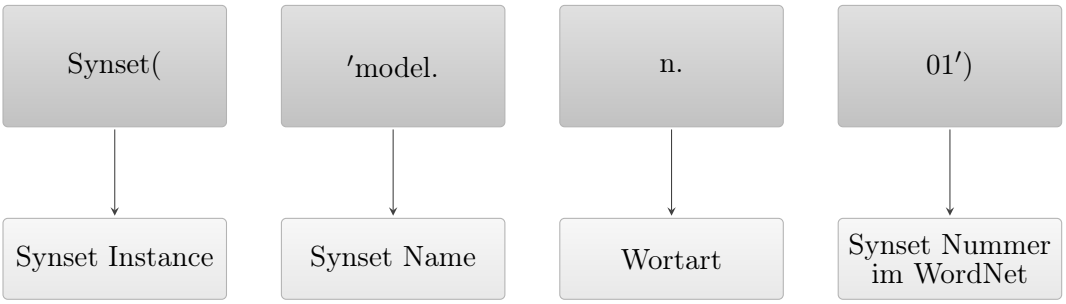


Abbildung 3.3: Synset-Eigenschaften

Bei der Ermittlung der verschiedenen Synsets eines *annotation candidate* extrahiert die API für jedes Synset die folgenden Informationen, die zum Verständnis des Synsets nützlich sind:

- Die Beschreibung des Synsets: Das entspricht die Definition dieses Synsets im WordNet.
- Der **Name** des Synsets (Label). Ein Synset hat mindesten ein Label im WordNet. Wir haben uns entschieden immer das erste Label zu nehmen. Um sicherzugehen das wir immer ein Label haben.

```
1 def get_label(syns):
2     lemmas = syns.lemmas()
3     name = lemmas[0].name()
4     return name
```

Quellcode 3.6: Extraktion des Labels aus einem Synset

- Die **Position** des aktuell betrachteten *annotation candidate* in der ursprünglich übergebenen Idee ("offset"). Dadurch kann das ICV-Tool dem Benutzer jedes Mal mitteilen, welches Wort gerade annotiert wird. Falls sich ein Wort in einer Idee wiederholt, dann ist es nicht offensichtlich, welche dieser Wiederholungen gerade annotiert wird. Aus diesem Grund speichert die API jedes Wort mit der Anzahl seiner Vorkommen in einem Wörterbuch und prüft bei jeder Behandlung eines Wortes, um welches Vorkommen es sich handelt und bestimmt so seine Position in der Idee mit Hilfe der folgenden Funktion.

```
def find_nth(text, wort, occurence):
    start = text.find(wort)
    while start >= 0 and occurence > 1:
        start = text.find(wort, start+len(wort))
        occurence -= 1
    return start
```

Quellcode 3.7: Position eines sich wiederholenden Wortes in einem Text

- Die aktuelle **Positions-Id** des Synsets in der WordNet-Datenbank (”resource”). Durch die Positions-Id kann ein Synset in der WordNet-Datenbank gefunden werden, ohne über das zum Synset zugehörige Wort abfragen zu müssen.
- Das Synset selbst (”source"”)
- Der *annotation candidate*, der gerade bearbeitet wird (”text”).
- Die Wortart(”pos”). Die verschiedenen Wortarten, die in WordNet existieren, sind: Name(n), Verb(v),Adjektiv(a), Adverb(r), Adjektives Satellite(s).

Sobald die API alle Informationen für das Annotieren eines Wortes mit WordNet gesammelt hat, werden diese in einem Wörterbuch(*Annotation Dictionary* die in Abbildung [3.2] zu sehen ist.) gespeichert. Ein Python-Wörterbuch ist eine Datentyp, der ein elektronisches Wörterbuch darstellt, das mehrere Elemente enthält. Jedes Element des Wörterbuchs besteht aus einem Key und seinem Wert. Es gibt verschiedene Möglichkeiten Datenstrukturen aus Python-Wörterbüchern aufzubauen. Wir werden für den Aufbau unserer JSON-Datei ein *Nested-dictionary* verwenden[Gee20]. Ein *Nested-dictionary* ist eine Zusammenstellung von mehreren Wörterbüchern. Jedes Element des Wörterbuchs besteht aus einem Key, der wiederum ein inneres Wörterbuch ist, und einem Wert, der die Elemente des Wörterbuchs repräsentiert und ebenfalls ein Wörterbuch ist.

In unserem Fall haben wir zwei Hauptwörterbücher definiert. Zum einen ein internes Wörterbuch, das als Key eines *ressource candidate* und als Wert eine Wörterbuchliste hat, die alle oben aufgelisteten Elemente eines *ressource candidate* enthält. Zum anderen ein externes Wörterbuch, welches das interne Wörterbuch umfasst. Der Key sind die *annotation candidates* und der Wert ist eine Liste von Wörterbüchern, wobei jedes einzelne den *annotation candidate* eines Wortes repräsentiert, das in der gegebenen Idee enthalten ist. Das *Nested Dictionary* muss noch in eine JSON-Datei umgewandelt werden, damit ein anderes Programm (in unserem Fall das ICV-Tool) durch eine Anfrage an unsere API alle von der API bereitgestellten Informationen - unabhängig von der verwendeten Programmiersprache - erhalten kann. Die Umwandlung des Wörterbuchs in eine JSON-Datei findet bei dem in der obigen Abbildung 3.1

genannten Schritt der Übersetzung statt. Die Python-Bibliothek JSON ermöglicht die Übersetzung mithilfe der Funktion dumps(). Unsere API stellt derzeit genau eine Url bereit: **GET /annotApi/**. GET-Requests gegen diese URL geben *annotation candidates* für einen bestimmten Text zurück. Genau ein Parameter **Text** ist verfügbar und wird benötigt. Dieser Parameter ist die Idee, die es zu annotieren gilt.

Beispiel: ***curl –request GET -url 'http://localhost:4000/annotApi/to find lost livestock'***.

Abschließend wird unsere WordNet-API dann mit dem ICV-Backend verbunden. Das ICV erweitert die Antwort unserer WordNet-API um einige Elemente wie die z.B. die folgenden:

1. "*id*": Id entsprechend der URL der Idee.
2. "*selected*": Gibt an, ob ein Synset-Kandidat ausgewählt wurde .
3. "*validated*": Gibt an, ob ein *annotation-candidate* annotiert worden ist.

Das ist die Antwort, die von der ICV unter Verwendung unserer WordNet-API bereitgestellt wird, nachdem der Benutzer die Ideen manuell annotiert hat. Ein Beispiel für eine JSON-Datei, welche nach der Annotation der Ideen mit dem neuen ICV-Backend erhalten wurde, ist unter Listing 3.8 aufgeführt.

```
{
   "id":"http://purl.org/innovonto/ideas/01487af4-3ed1-4746-
   a162-2379a8e1a63a",
   "content":"to find lost livestock ",
   "annotations":[
      {
         "offset":3,
         "resource_candidates":[
            {
               "description":"get something or somebody for a
   specific purpose",
               "label":"line_up",
               "offset":3,
               "resource":2213336,
               "source":"line_up.v.02",
               "text":"find",
               "pos":"v",
               "selected":true
}
{}
],
         "text":"find",
         "validated":true
}, {...}  ]
{...} }
```

Quellcode 3.8: JSON-Ausgabestruktur der ICV mit dem neuen Backend

3.2 Extraktion von Hypernymen

Um unsere Ideen mit WordNet zu kategorisieren, ist es erforderlich, zunächst das Hypernym jedes Synsets zu extrahieren, das während der Annotation der Ideen mit Hilfe dem neuen ICV ausgewählt wurde. Nach der Annotation erhalten wir die oben genannte JSON-Datei (Listing 3.8). Für unsere Analyse sind wir nun in der Lage, die Hypernyme aus den vom Benutzer gewählten Synsets zu extrahieren. Das sind die Synsets, für deren Attribute "*validated*" = "*selected*" = "*true*" ist.

Zuerst werden die ausgewählten Kandidaten extrahiert und dann für jeden ausgewählten Kandidaten die Hypernyme extrahiert. All dies speichern wir für die spätere Analyse. Zu diesem Zweck werden am Ende vier Wörterbücher zur Verfügung gestellt. Das erste verlinkt jedes Synset mit den darin enthaltenen Ideen, das zweite verlinkt jede Idee mit den darin enthaltenen Synsets, die dritte verlinkt jedes Hypernym mit den darin enthaltenen Ideen und schließlich verlinkt das vierte jede Idee mit dem darin enthaltenen Hypernym.

Die Extraktion von Hypernymen aus einem Synset erfolgt ebenfalls über die Python-WordNet-Bibliothek. Diese Bibliothek bietet drei Möglichkeiten, die Hypernyme zu extrahieren. Entweder werden für ein Synset A alle verschiedenen Hypernyme-Pfade bis zum höchsten übergeordneten Hypernyme in WordNet extrahiert (**wordnet.synset(A).hypernym_paths()**), oder nur die Hypernyme direkt aus dem Synset A, bzw. diejenigen, die es direkt zusammenfassen (**wordnet.synset(A).hypernyms()**). Die letzte Möglichkeit besteht darin, den höchsten Hypernym direkt aus Synset A zu extrahieren bzw. die Hypernymwurzel (**wordnet.synset(A).root_hypernyms()**). Nehmen wir z.B "*wheeled_vehicle*". "*wheeled_vehicle*" hat genau zwei Wege, weil zwischen "*wheeled_vehicle*" und der Hypernymwurzel-"*entity*" genau zwei Wege liegen.

Das Synset "*Wheeled_vehicle*" kann gleichzeitig als "*container*" und als "*vehicle*" klassifiziert werden (Abbildung 3.4). Zusätzlich werden alle Ideen auch in einem Wörterbuch "IdeenDict" mit einer Nummern-ID gespeichert, da uns die JSON-Datei eine ID liefert, die dem Link der Idee in der HCC-Ideendatenbank entspricht. Dieser Link ist sehr lang und für die Handhabung unserer Daten nicht praktikabel. Dieses IdeenDict-Wörterbuch hat als Schlüssel Paare $(NummerId, ID)$, wobei seine die ID jene aus der JSON-Datei ist. So kann der Benutzer nachschauen, um welche Idee es sich handelt, indem einfach die im anderen Wörterbuch angegebene NummerId aufgesucht wird.

Für unsere Analyse verwendeten wir die Extraktion aller Hypernyme. Der Algorithmus (Abbildung 3.5) beschreibt die verschiedenen Schritte, die bei der Extraktion der Hypernyme durchgeführt wurden.

Mit Hilfe des "synsetDict"-Wörterbuchs und des Direkt-Hypernyms haben wir ein "DictionaryTree"-Wörterbuch erstellt, das die verschiedenen Beziehungen

zwischen den Hypernymen der ausgewählten Synsets enthält. Das Wörterbuch enthält als Schlüssel einen Knoten, genauer gesagt ein Hypernym (das nichts anderes als ein Synset ist) und als zugehörigen Wert eine Liste, die dessen Kinder darstellt, d.h. alle Hypernyme, die als direkten Hypernyme den Schlüssel haben.

Nehmen wir zum Beispiel an, dass unser SynsetDict aus Vier Synsets besteht (*"hatchback.n.01", "compact.n.03", "truck.n.01", "wheeled_vehicle.n.01"*) und und rufen wir mit unserem SynsetsDict die Funktion analyse(SynsetsDict) auf, die das DictionaryTree erstellt. Es Werden die links angezeigenen Verbindungen herausgefunden (Abbildung 3.6) und als Ausgabe der folgenden DictionaryTree:

DictionaryTree ={ ***"hatchback.n.01"=[],***
"compact.n.03"=[],
"truck.n.01"=[],
"motocar.n.01"=["hatchback.n.01","compact.n.03"],
"motor_vehicle.n.01"=["motocar.n.01","truck.n.01"],
"self_propelled_vehicle.n.01"=["motor_vehicle.n.01"],
"wheeled_vehicle.n.01"=["self_propelled_vehicle.n.01"],
"vehicle.n.01" = ["wheeled_vehicle.n.01"]
"container.n.01" = "wheeled_vehicle.n.01"]
,...,
"entity.n.01" = [...]}

Die im aktuellen Kapitel beschriebenen Komponenten und Algorithmen wurden in zwei Jupyter-Notebooks implementiert[1]. Das erste Notebook enthält die Implementierung der WordNet-API, die im Backend der ICV-Tools aufgerufen wird und als Ergebnis die WordNet Annotations-Möglihckeiten als JSON zurückgibt. Das zweite ist für die Extraktion der Hypernyme der jeweiligen Synsets in der vom ICV-Tool erhaltenen JSON-Datei zuständig. Im Nächsten Kapitel wird dieses Jupyter-Notebook zusätzlich um die Elemente der Datenaufbereitung für die Analyse und Visualisierung erweitert.

[1]Beide Notebooks sind unter https://github.com/gancia-kiss/Bachelorarbeit verfügbar.

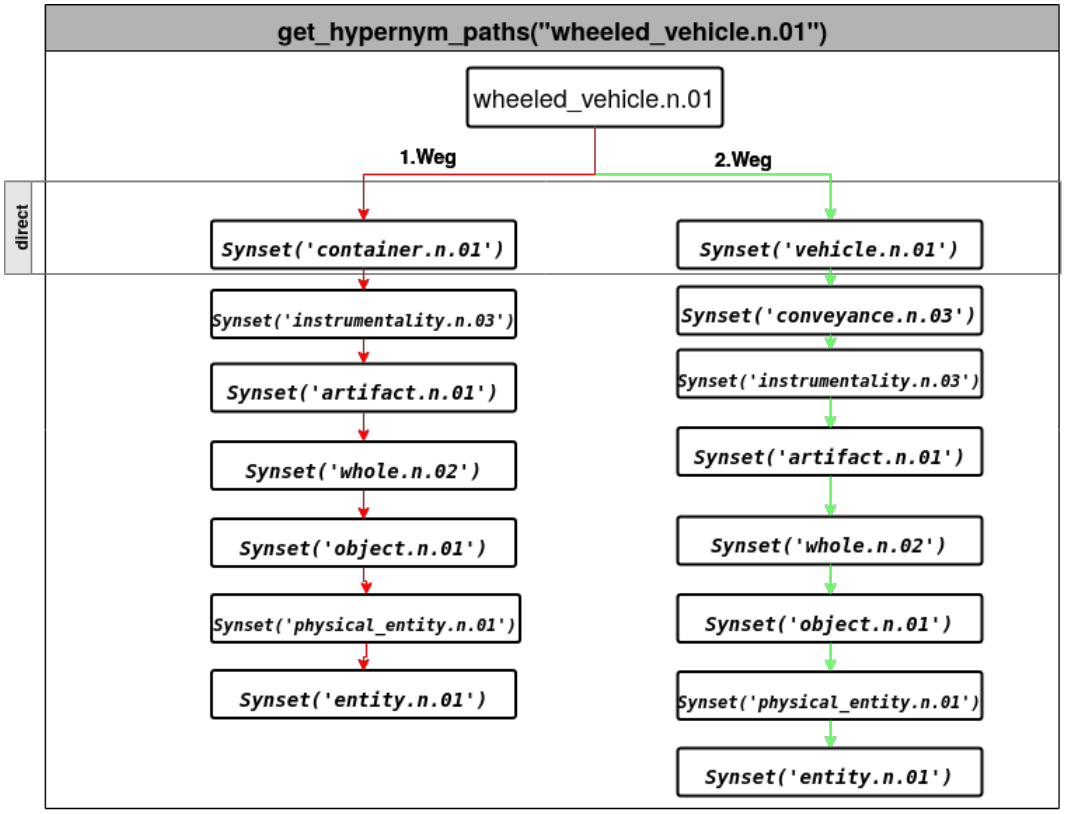


Abbildung 3.4: Extraktion eines Hypernyms aus WordNet

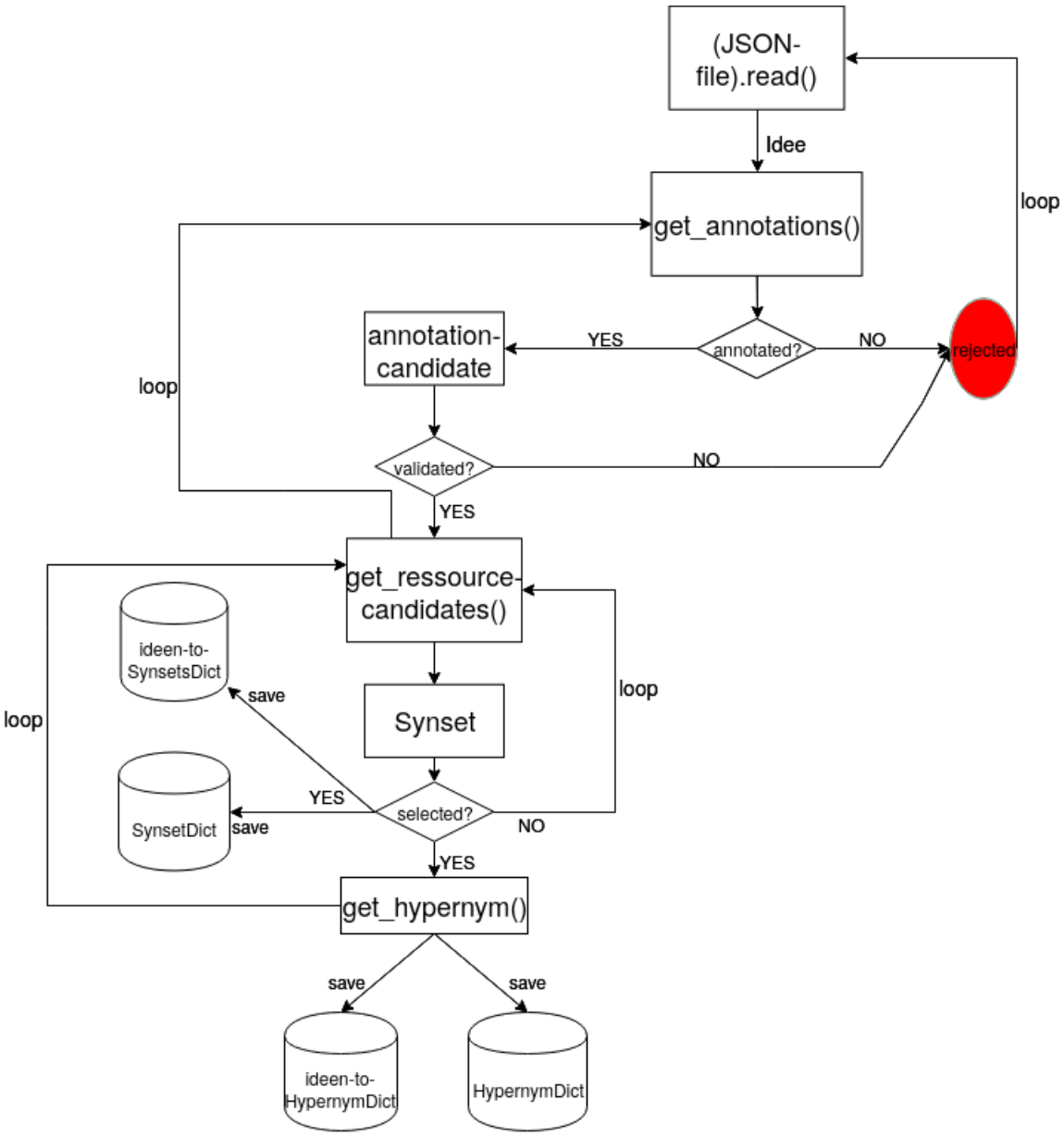


Abbildung 3.5: Hypernym-Extrahierungs-Algorithmus (Ablaufdiagram)

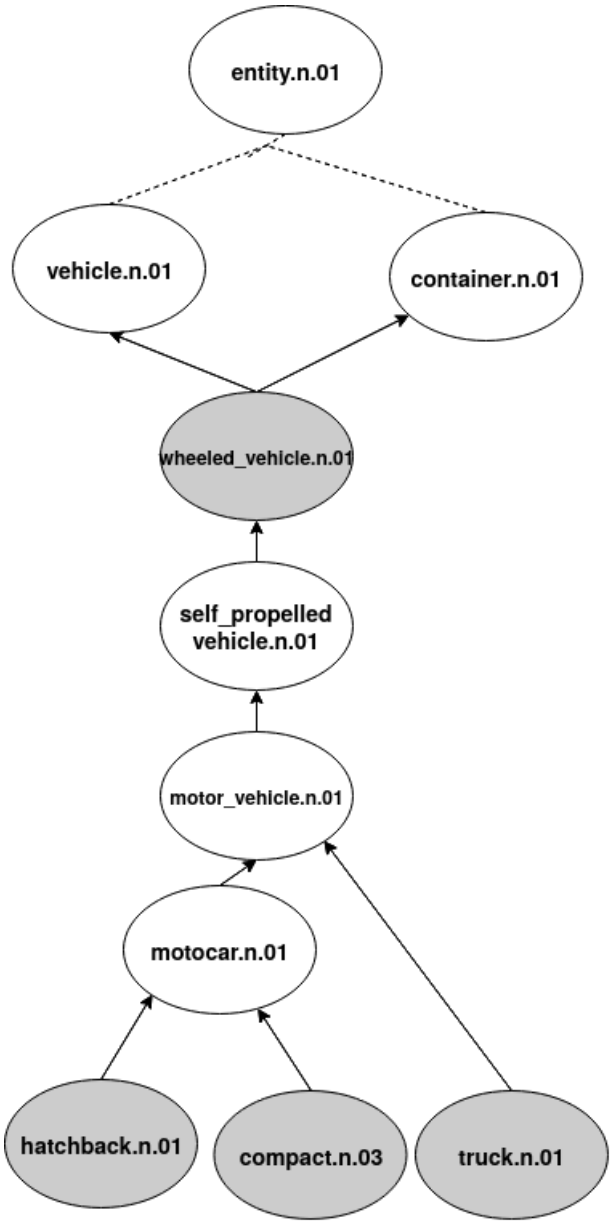


Abbildung 3.6: Darstellung der erhaltenen Beziehungen zwischen den gegebenen Synsets

4 Anwendungsfall: Bionic Radar

In diesem Kapitel werden wir zunächst den Verlauf einer Annotation von Ideen mit Hilfe der ICV-Tools und mit unserem Wordnet-API als Backend, die die HCC-Gruppe im Rahmen des "ideas to market"-Projekts erhalten hat, erläutern und dann Hypernyme aus den verschiedenen ausgewählten Synsets extrahieren und annalysieren. In der zweiten Phase des "ideas to market"-Projekts beschlossen die Organisatoren, verschiedene Ideen für den Einsatz der Bionic-Radar-Technologie zu sammeln, die derzeit vom Fraunhofer-Institut für Hochfrequenzphysik und Radartechnik (FHR) und dem Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA) entwickelt wird. "Die Technologie ermöglicht es Lebewesen anhand ihrer individuellen Bewegungsmuster zu erkennen"[1]. Das Fraunhofer Center for Responsible Research and Innovation (CeRRI) [2] ist ein Forschungszentrum, das ebenfalls in dem Projekt "ideas to market" arbeitet. Die Ideen wurden über Amazon Mechanical Turk gesammelt[3]. Die Organisatoren hatten drei verschiedene Beschreibungen der neuen "Bionic Radar"-Technologie vorbereitet.

1. Eine technische Beschreibung: unter Verwendung typischer technischer Begriffe und unter Angabe der Technik der Technologie.
2. Eine abstrakte Beschreibung: eine sehr abstrakte Beschreibung, die keine technischen Informationen enthält.
3. Eine metaphorische Beschreibung: Diese Beschreibung wurde als Beispiel dafür gegeben, wie diese Technologie auf metaphorische Weise in anderen Bereichen eingesetzt werden könnte.

Die Beschreibungen wurden zufällig aber gleichmäßig verteilt, allen 102 Teilnehmern vorgelegt. Jeder Teilnehmer hatte 30 Minuten Zeit, um seine Ideen einzureichen und mithilfe des alten ICV (ICV mit DBPedia als Backend) zu annotieren. Die 102 Teilnehmer generierten insgesamt 581 Ideen.

Für unsere Analyse haben wir 200 von diesen 581 Ideen nach dem Zufallsprinzip ausgewählt. Als Dateninput wurden nicht annotierten Ideen genommen (die bestehenden Annotation wurden gefiltert), da wir unseren Ansatz mit WordNet implementieren wollen.

[1] https://www.cerri.iao.fraunhofer.de/de/projekte/AktuelleProjekte/ideas-to-market.html
[2] https://www.cerri.iao.fraunhofer.de/
[3] https://www.mturk.com/

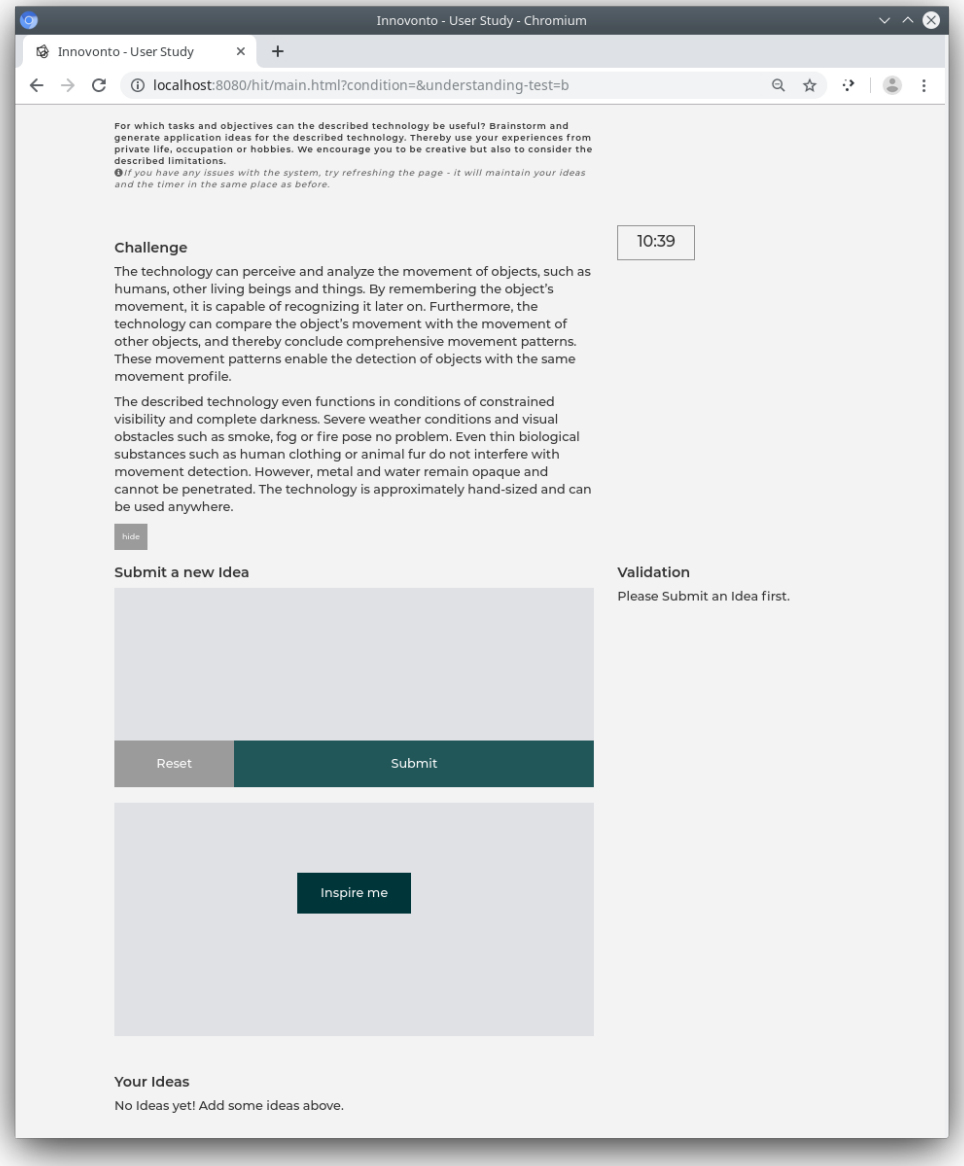


Abbildung 4.1: Beispielhaftes Interface mit einer abstrakten Beschreibung, die ein Teilnehmer erhält, um eine Idee zu generieren.

4.1 Daten-Annotation

Wegen der *Word-Sense Disambiguation* kann die Annotation der Ideen nicht automatisiert werden. Deswegen mussten wir alle 200 Ideen manuell annotieren. Die Ideen wurden im JSON-Format übergegeben. Jede Idee besteht aus einer ID, einem Type, einer Angabe, zu welchem Projekt die Idee gehört und als letztes der Inhalt der Idee (”content”). Vor der Annotation wurden zu-

nächst die WordNet-API und dann das ICV-Tool gestartet. Das ICV-Tool lief lokal auf Port 1234.

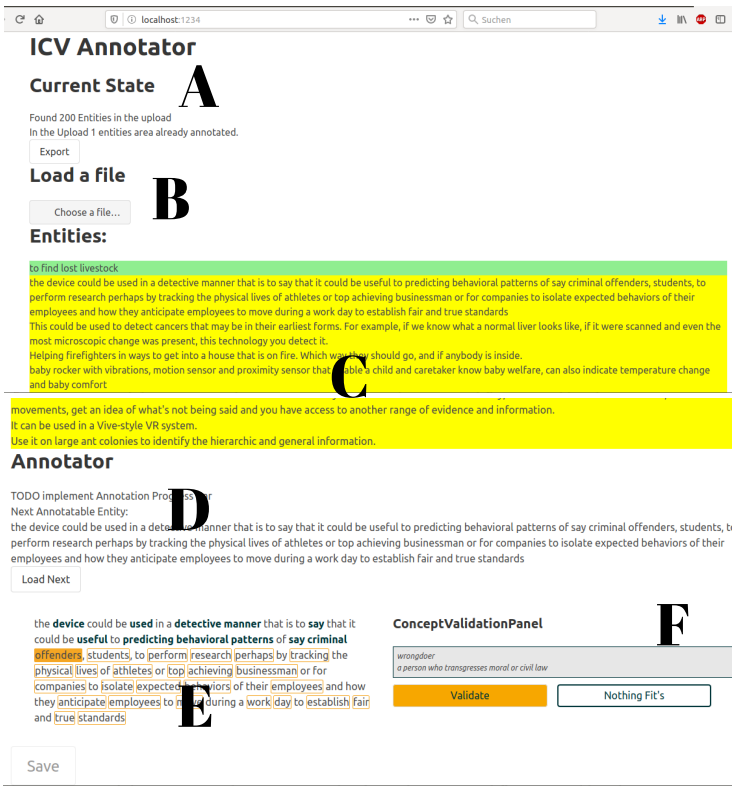


Abbildung 4.2: Beispielhaftes Interface für ICV während einer Annotation von Ideen

Das verwendete ICV-Annotations-Tool wurde von der Forschungsgruppe HCC bereitgestellt und besteht aus sechs Panels. Panel *A*, in dem die Anzahl der in der JSON-Datei enthaltenen Ideen, sowie die Anzahl der bereits vom Benutzer annotierten Ideen angezeigt wird. In demselben Panel gibt es auch einen Button "*Export*", der es dem Benutzer ermöglicht, die bereits annotierten Ideen als JSON-Datei zu exportieren. Unter dem Abschnitt "*load a file*" wird die Option "*choose a file*" angezeigt, um die Ideen in das ICV zu laden (Panel *B*). Auf Panel *C* "*Entities*" werden alle Inhalte jeder hochgeladenen Idee und der Fortschritt der Annotation angezeigt. Bereits annotierte Ideen werden in Grün und nicht annotierte Ideen in Gelb markiert. Auf dem vierten Panel *D* wird die aktuell annotierte Idee sowie die nächste Idee, die annotiert wird, angezeigt. Mit der Taste "*Load Next*" auf Panel *D* lädt der Benutzer die nächste annotierte Idee herunter (für das Backend sendet die ICV zu diesem Zeitpunkt eine Anfrage an unsere WordNet-API). Dann zeigt Panel *E* die annotierte Idee mit den verschiedenen Termen, die von der WordNet-API verwendet werden (in orangefarbenen Boxen). Die bereits annotierten Konzepte werden fett und das derzeit annotierte Konzept orange markiert. Auf dem letzten Panel *F* ("*ConceptValidationPanel*") schließlich werden alle Synsets aufgelistet und ihre Definition mit Hilfe von WordNet ermittelt. Am Ende jeder Auflistung gibt

es zwei Buttons: den Button "*Validate*", den der Benutzer nach der Auswahl der Bedeutungen, die am besten zum aktuell annotierten Wort passen drücken muss. Die Button "*Nothing Fit's*"", die es dem Benutzer erlaubt, zum nächsten Wort zu gehen, falls keine Bedeutung für ihn geeignet ist. Am Ende von Panel *E* befindet sich der Button "*Save*"", der es dem Benutzer ermöglicht, jede Annotation zu speichern, nachdem alle Konzepte einer Idee annotiert wurden.

Zur Annotation der 200 gesammelten Ideen folgten wir allen oben aufgeführten Schritten und wählten für jedes vorgeschlagene Konzept die Definition(en) aus, die wir im Zusammenhang mit der eingereichten Idee für am besten geeignet hielten.

4.2 Descriptive Statistik

Nachdem wir die 200 Ideen annotiert hatten, sammelten wir während der Extraktion der Hypernyme die verschiedenen im vorherigen Kapitel aufgelisteten Wörterbücher, um die erhaltenen Daten analysieren zu können (Abbildung 3.5).

Wir erhielten insgesamt 1289 Synsets, die während der Annotation ausgewählt wurden. Daraus haben wir insgesamt 2208 Hypernyme extrahiert. Die längste Idee enthält 90 Synsets mit 203 Hypernymen. Aber die Idee mit den meisten Synsets ist nicht unbedingt die Idee mit den meisten Hypernymes. Zum Beispiel können wir auf dem Balkendiagramm (Abbildung 4.3)(Abbildung 4.4) sehen, dass die Idee mit der Nummer ID 71, welche die Idee mit den zweimeisten Synsets ist, mit insgesamt 148 Hypernyme in der Ideenklassifizierung anhand ihrer Anzahl von Hypernymen an dritter Stelle steht.

Anhand der Daten, die wir erhalten haben, wollen wir die verschiedenen in Kapitel 2 beschriebenen Punkte analysieren.

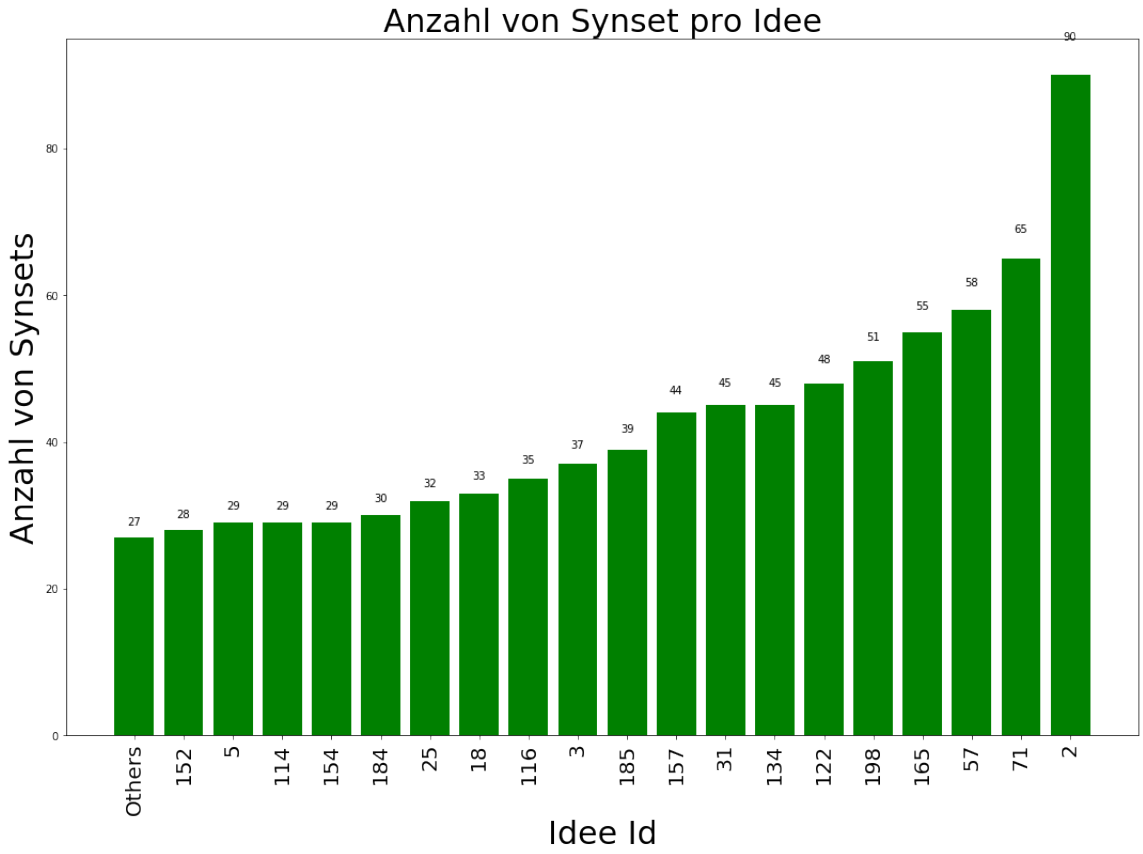


Abbildung 4.3: Balkendiagramm: Anzahl von Synsets pro Idee. Das Label **Others** sammelt Ideen mit einer Anzahl an Synsets, die weniger als 30 Prozent der gesamten Anzahl entsprechen.

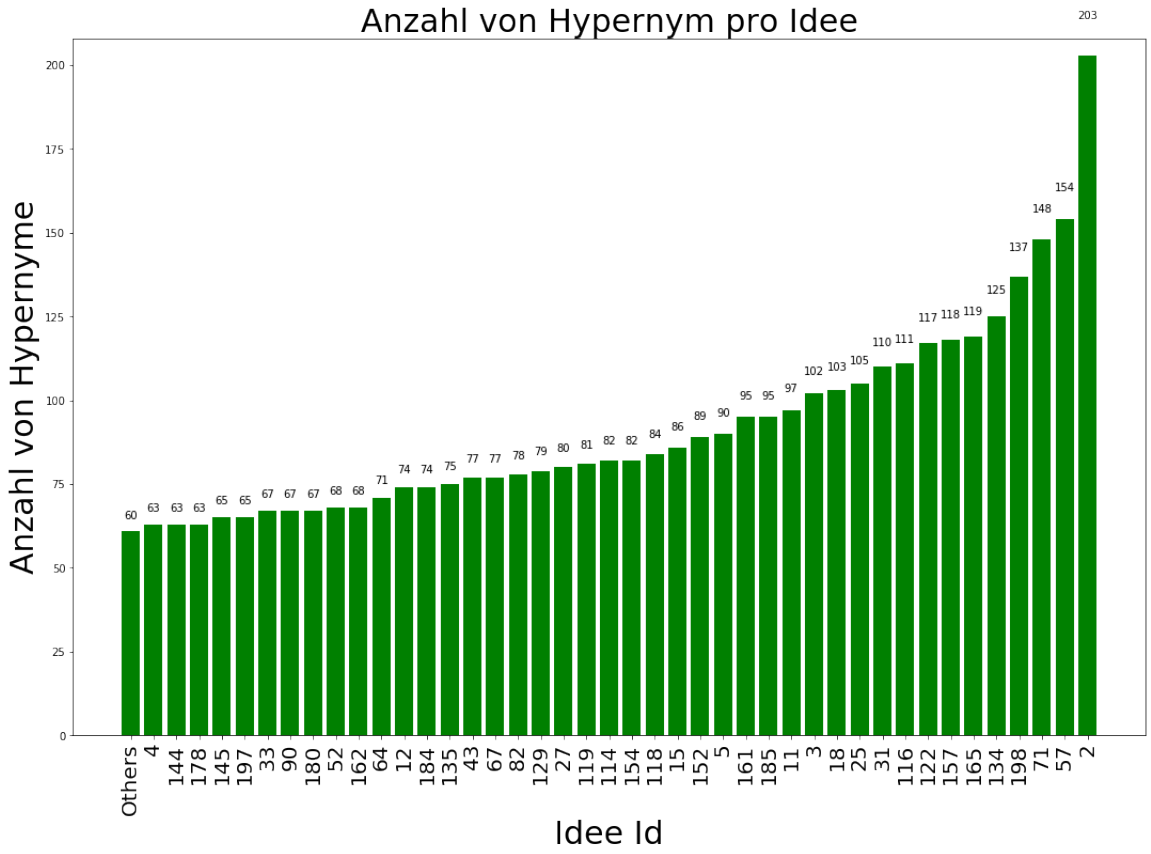


Abbildung 4.4: Balkendiagramm: Anzahl von Hypernymen pro Idee. Das Label **Others** sammelt Ideen mit einer Anzahl an Hypernymen, die weniger als 30 Prozent der gesamte Anzahl entsprechen.

4.2.1 Analyse 1 : Vorhandene Kategorien im Datensatz

Mit Hilfe des Wörterbuchs "DictionaryTree" und der Python Graphviz-Bibliothek[4] haben wir einen Baum aufgebaut, um die verschiedenen Beziehungen zwischen den Hypernyme zu visualisieren. Aber wir stellten fest, dass nicht alle Synsets "*entity*" als ihren Wurzelhypernym haben. Insgesamt gibt es 69 Synsets, die nicht "*entity*" als Wurzelhypernym haben. Diese Synsets sind Verben, Adjektive, Satelitenadjektive oder Adverbien. Wir haben sie direkt mit "*entity*" verbunden, um einen einzigen Baum zu bilden. Der Baum war so groß und unleserlich, dass wir beschlossen, uns die Teile des Baumes anzusehen bzw. die drei obersten Ebenen unseres Baums, um die verschiedenen Kategorien nach "*entity*", "*physical entity*" und *abstraction* zu finden. Unmittelbar nach "*entity*" haben wir also "*physical Entity*", *abstraction* und die 69 anderen Synsets. "*Physical entity*" wiederum gruppiert insgesamt 5 verschiedene Kategorien und *abstraction* gruppiert 6 Kategorien. Das heißt, dass

[4] https://pypi.org/project/graphviz/

die 200 Ideen des Anfangs laut WordNet in 11 Hauptkategorien unterteilt werden können, mit Ausnahme der 69 anderen Kategorien (Abbildung 4.5).

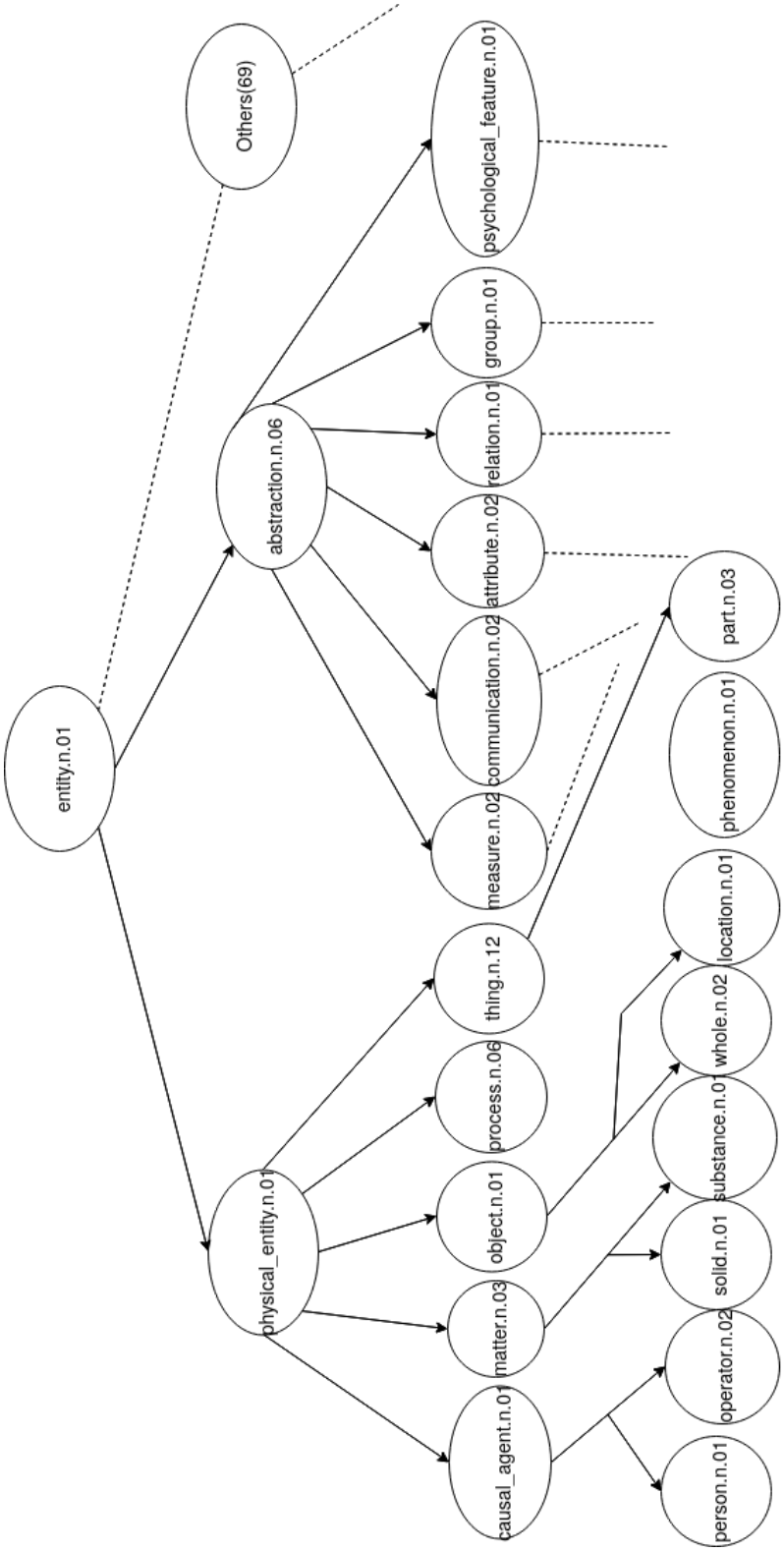


Abbildung 4.5: Teilbaum Dictionarytree mit anzeige der drei obersten Niveaus

Um die Kategorien einschließlich der Gesamtheit der gefundenen Synsets zu finden, visualisierten wir mit Hilfe eines Balkendiagramms das Wörterbuch "hypernymList", das alle Hypernyme mit ihrer jeweiligen Anzahl von Ideen enthält. Suchen wir zunächst nach den besten Kategorien. Nehmen wir an, dass Hypernyme (außer "*Entity*", "*Physical Entity*" und "*abstraction*"), die mehr als 30 Prozent der Ideen enthalten, unsere besten Kategorien sind. Wir beobachten dann auf diesem ersten Diagramm (Abbildung 4.6) genau 21 Kategorien, die fast die gleichen oben beobachteten Kategorien enthalten. Wir können jedoch feststellen, dass diese Kategorien sehr abstrakt sind und uns nicht genügend Informationen über unsere Daten geben. Nach Anzahl der Ideen geordnet folgt direkt auf "*physical entity*" zum Beispiel "*Object*", das mehr als drei Viertel der eingegangenen Ideen sammelt, aber nicht nähere Information über die *Bionic Radar*- Technologie vermittelt. Auf der anderen Seite kann das "*Phycological_feature*", "*causal agent*", interessant sein. Um zu besseren Kategorien zu gelangen, wurden die verschiedenen Hypernyme visualisiert, die zwischen 30 und 70 Ideen gruppieren. Tatsächlich ist auf dem resultierenden Balkendiagramm (Abbildung 4.7) zu sehen, dass die neuen Kategorien wie "*spy*", "*animal*" und "*profession*" mit mehr als einem Drittel der Ideen viel interessanter sind. Die Kategorie "*spy*" suggeriert zum Beispiel, dass die neue Technologie zum Spionieren verwendet werden kann und dass sie sowohl bei Tieren als auch bei der Arbeit eingesetzt werden kann.

Wenn wir nun die Kategorien analysieren, die nur eine Idee enthalten, können wir unerwartete Kategorien finden. Wir stellen fest, dass wir genau 1243 Hypernyme haben, die nur eine Idee enthalten, was mehr als der Hälfte aller gesammelten Hypernyme entspricht, weshalb wir sie in einer Liste betrachten. Beim Durchgehen dieser konnten wir feststellen, dass diese Kategorien viel präziser sind als die oben genannten. Unter den Kategorien fanden wir solche wie "*livestock*", "*malignant tumor*", "*topographic point*" oder "*search engine*". Das sagt uns, dass die Bionic-Radar-Technologie wahrscheinlich bei einem Rind, bei einem bösartigen Tumor und an einem geografischen Punkt eingesetzt werden kann.

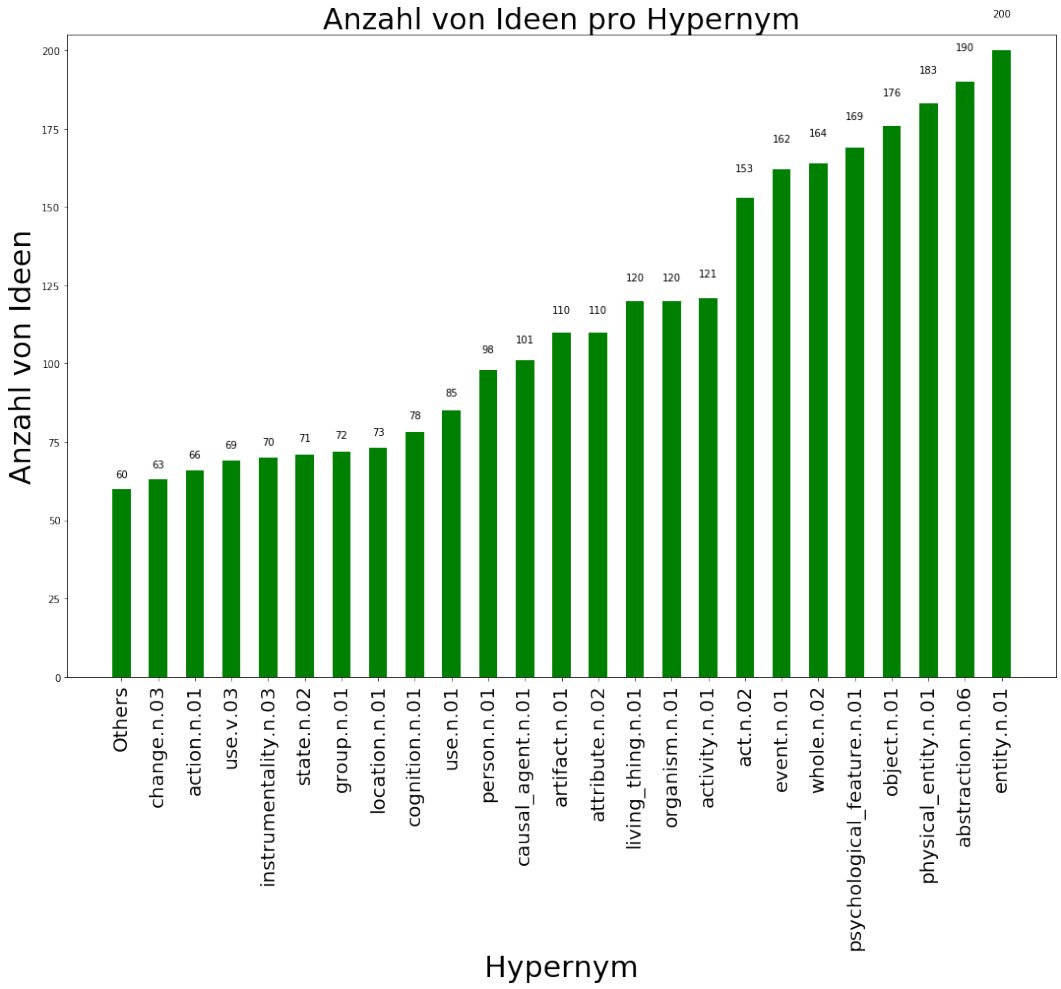


Abbildung 4.6: Balkendiagramm: Anzeige der besten Kategorie, aufsteigend geordnet nach Anzahl der Ideen. **Others** umfasst Hypernyme, deren Anzahl an Ideen kleiner als 60 ist.

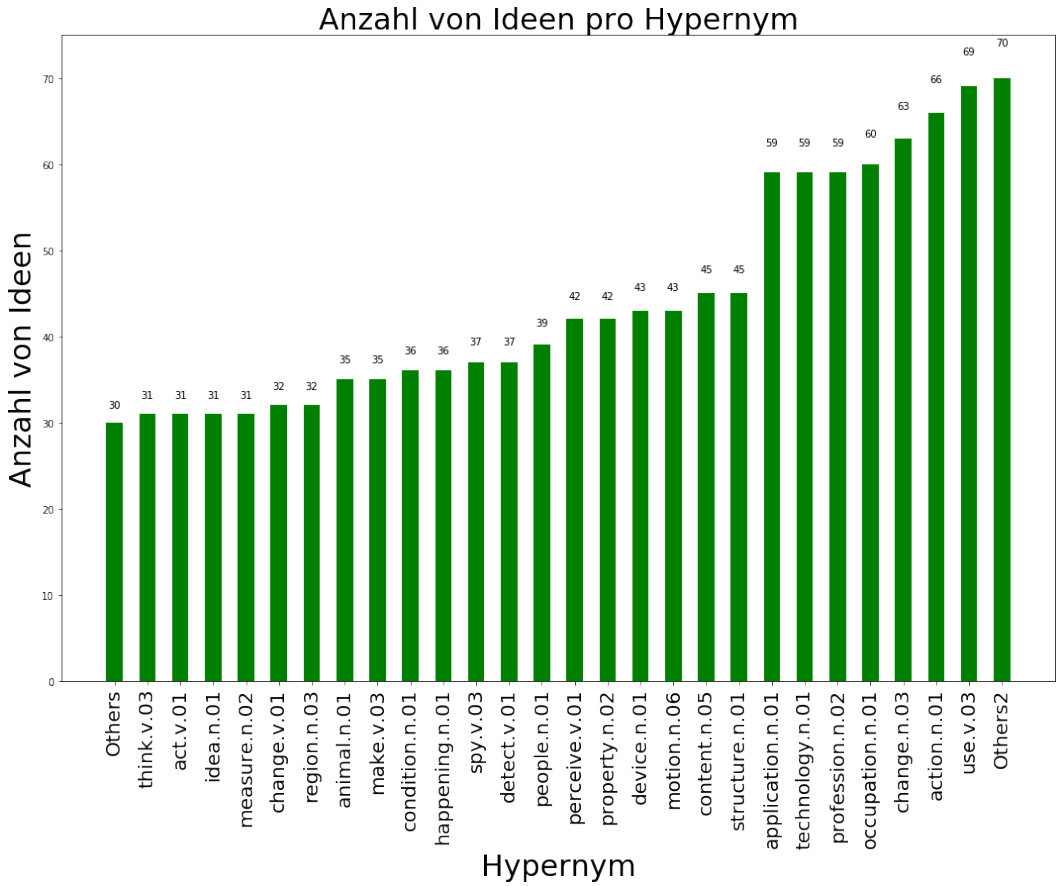


Abbildung 4.7: Balkendiagramm: Anzeige der besten Kategorie, **Others** gruppiert Hypernyme, bei denen die Anzahl an Ideen kleiner als 30 ist und **Others2** gruppiert Hypernyme, bei denen die Anzahl an Ideen größer als 60 ist.

Um noch bessere Kategorien zu finden, haben wir wieder mit Hilfe des Wörterbuchs "HypernymDict" versucht, die Überschneidungen zwischen den Hypernymen zu visualisieren, die die gleiche Anzahl von Ideen gruppieren. Das heißt, für jede Gruppe von Hypernymen haben wir die Anzahl der gemeinsamen Ideen untersucht. Eine Gruppe von Hypernymen ist eine Sammlung von Hypernymen, die die gleiche Anzahl von Ideen enthalten. Um solche Gruppen zu bestimmen haben wir die pandas-Bibliothek benutzt und die Ergebnisse in einer Tabelle gespeichert(Tabelle 4.1). Die Tabelle hat genau drei Spalten. Die erste enthält die Anzahl der Ideen, die in den verschiedenen Gruppen von Hypernymen enthalten sind, die zweite die Anzahl der Ideen in der Schnittmenge dieser Gruppen. Die letzte enthält die Anzahl der in jeder Gruppe enthaltenen Hypernyme. Die Hypernym-Gruppen, die keine gemeinsame Idee enthalten, wurden aus der Tabelle entfernt. Mit Hilfe eines Balkendiagramms haben wir die gefundenen Gruppen von Hypernymen visualisiert(Abbildung 4.8).

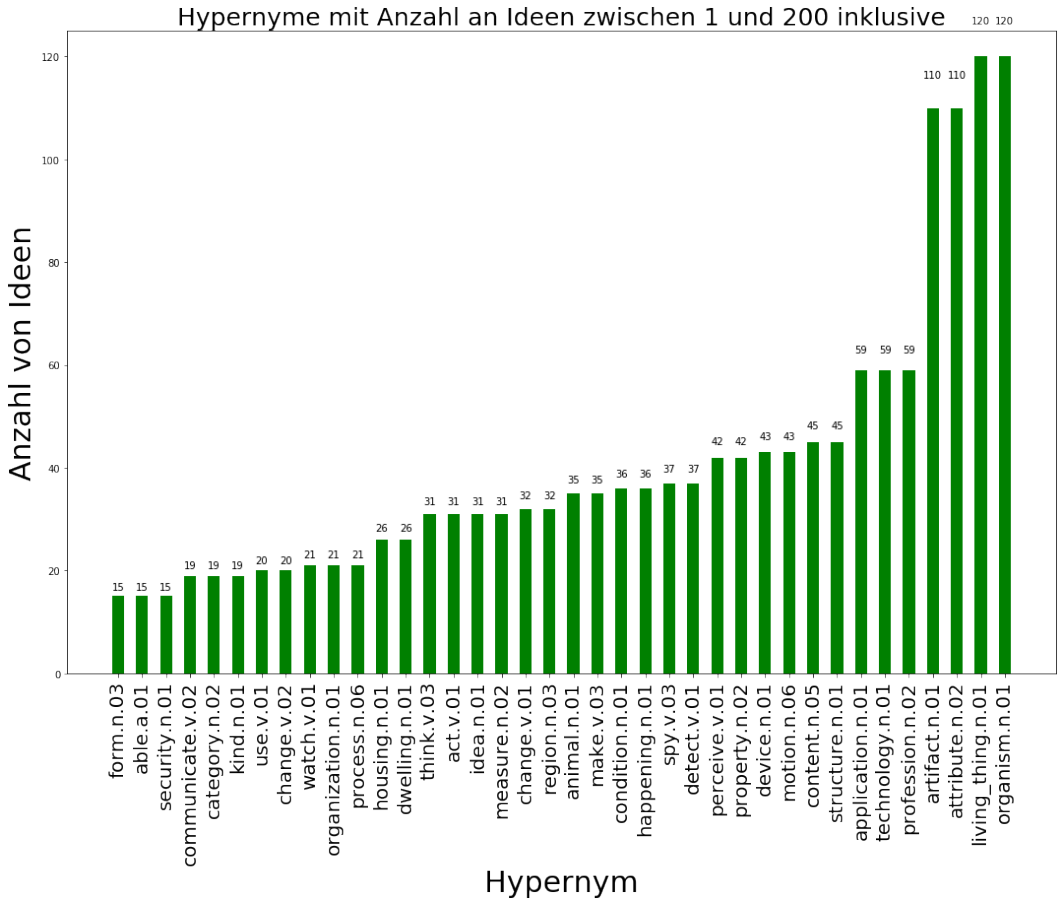


Abbildung 4.8: Balkendiagramm: Hypernym-Gruppen.

Anzahl an Ideen	Überschneidung	Anzahl Hypernyme
15	1	3
19	2	3
20	2	2
21	1	3
26	26	2
31	3	4
32	8	2
35	5	2
36	10	2
37	37	2
42	7	2
43	12	2
45	11	2
59	59	3
110	68	2
120	120	2

Tabelle 4.1: Schnittmenge von Hypernym-Gruppen.

Wenn wir die Ergebnisse in der Tabelle 4.1 betrachten, können wir feststellen, dass "*living thing*" und "*organism*" 120 Ideen gemeinsam haben. Daraus können wir schließen, dass das Bionic Radar zum Beispiel etwas im Organismus detektieren kann. Auf der anderen Seite haben wir "*technology*", "*profession*" und "*application*", was ebenfalls Gruppen von 59 Ideen ist. Diese Kategorien haben wir oben auch bekommen, und wir können zum Beispiel daraus schließen, dass diese neue Technologie für eine bestimmte Arbeit eingesetzt werden kann. Es folgen die Kategorien "*spy*" und "*detect*", die 37 gleiche Ideen umfassen. Wir finden auch neue Kategorien wie "*kind*", "*communication*" und "*category*", die 19 Ideen gruppieren, aber nicht, dass 2 Ideen gemeinsam sind, die uns mitteilen können, dass das Bionic Radar uns z.B. bei der Kommunikation helfen kann. Gleichzeitig erhalten wir auch die neuen Kategorien "*security*", "*able*" und "*form*" mit 15 Ideen. Obwohl diese nur eine Idee gemeinsam haben, lassen sie uns wissen, dass diese neue Technologie im Bereich der Sicherheit eingesetzt werden kann. Noch genauer kann man es so interpretieren, dass "Bionic Radar die Fähigkeit hat, als eine Form der Sicherheit eingesetzt zu werden".

Analyse 1b : Vorhandene Kategorien mit reduzierten Synsets

Aufgrund der großen Anzahl von Synsets haben wir beschlossen, genauer zu untersuchen, wie sich die Synsets verhalten, wie oft sie in unseren Daten erscheinen. Dafür haben wir mit Hilfe des Wörterbuchs Ideen_to_SynsetDict der Balkendiagramm (Abbildung 4.9) gebaut.

Wir können im Ergebnis-Balkendiagramm (Abbildung 4.9) sehen, dass die sich am häufigsten wiederholenden Synset wie "*use*", "*technologie*", "*detect*" für uns nicht neu sind , da sie bereits in der Beschreibung des Bionic Radar auftauchen. Wir wissen bereits, dass das Bionic Radar eine Technologie ist.

Wir haben daher versucht, unsere Daten zu reduzieren, indem wir die nur einmal vorkommenden Synsets analysieren, um wieder Kategorien zu finden. Dafür wurde jede Idee auf nur ein Synset reduziert. Das Synset auf das reduziert wurde war hierbei das Synset das die Idee am eindeutigsten beschreibt. Wir haben das Synset das eine Idee am eindeutigsten beschreibt definiert als das Synset innerhalb der Idee, das am wenigsten oft unter allen 200 Ideen vorkommt. Für den Fall, dass eine Idee mehr als ein Synset enthält, das mehr als einmal auftritt, haben wir zufällig ein Synset unter ihnen ausgewählt, um die Idee zu repräsentieren. Nachdem die anderen Synsets entfernt worden waren, wurden die Ideen, die lediglich mehr als einmal auftretenden Hypernyme enthalten, getrennt untersucht. Die Anzahl der Wiederholungen des Hypernyms wurde erhöht, um am Ende für jede Idee ein einziges Synset zu erhalten, das diese Idee repräsentiert. Insgesamt erhalten wir 200 Synsets für die 200 Ideen. Mit dieser neuen Ausgabe bauen wir unser Wörterbuch "hypernymList" wieder auf. Wir erhalten für diese 200 Synsets 524 Hypernyme. Indem wir die 524 Hypernyme mit Hilfe des Balkendiagramms in einer Weise visualisieren, die mit Hilfe des Balkendiagramms Schnitte und Details darstellt, finden wir schon ab "*psychological_feature*" interessanter Kategorie, weil diese über aussagekräftige Informationen verfügen. Außerdem können wir feststellen, dass dieses Diagramm (Abbildung 4.10) die in den beiden Diagrammen (Abbildung 4.6) und (Abbildung 4.7) gefundenen Kategorien zusammenfasst. Zum Beispiel haben wir die Kategorie "*organism*" (in Abbildung 4.6) und "*animal*" (in Abbildung 4.7), die wir auf diesem Diagramm finden können. Wir können daraus schließen, dass wir mit den reduzierten Daten in kurzer Zeit eine bessere Kategorie finden können als in einem Diagramm, weil wir die beiden anderen gruppieren können.

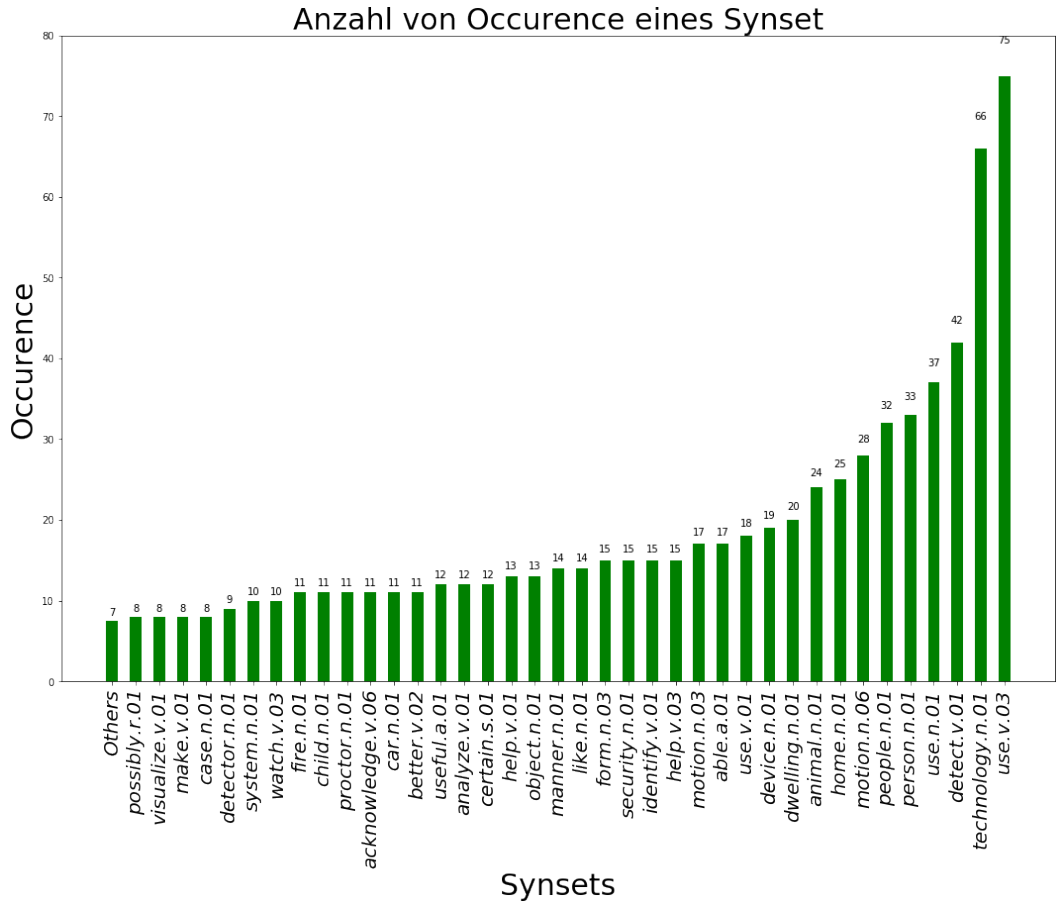


Abbildung 4.9: Balkendiagramm: Wiederholungen von Synsets in dem Datensatz. **Others** umfasst die Synsets, die weniger als sieben Mal vorkommen.

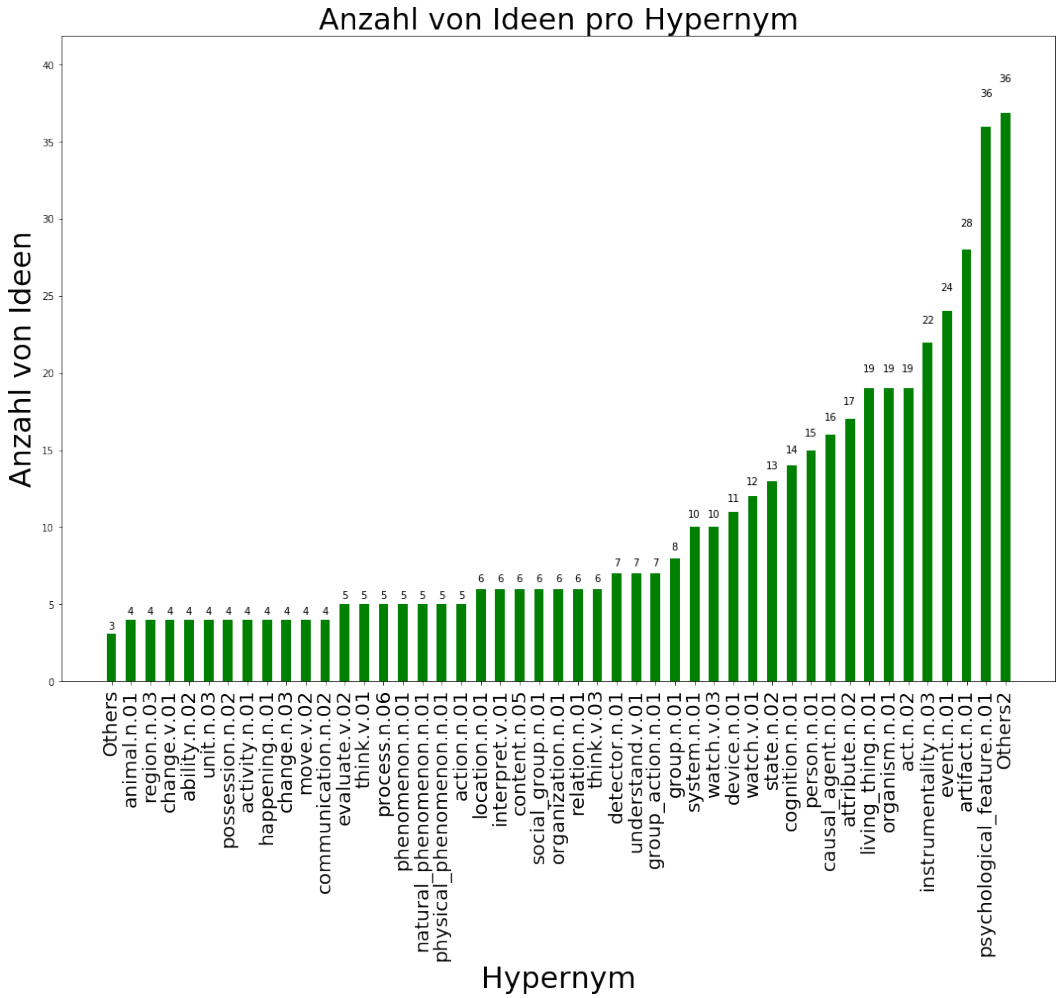


Abbildung 4.10: Balkendiagramm: Einige der gefundenen Kategorien mit den reduzierten Synsets, nach Anzahl der Ideen aufsteigend sortiert. **Others** umfasst Hypernyme, bei denen die Anzahl der Ideen kleiner als 3 ist und **Others2** umfasst Hypernyme, bei denen die Anzahl an Ideen größer als 36 ist.

4.2.2 Analyse 2 : Abstraktionsgrad einer Idee

Wir wollen nun den zweiten Punkt der in Kapitel 2 beschriebenen Analyse analysieren. Wir haben in Kapitel 3 bemerkt, dass es Synsets mit mehr Weg zum Wurzelhypernyme "*entity*" gibt. Für diese Synsets (Abbildung 3.4)) haben wir einfach die Entfernungen der verschiedenen Wege durchschnittlich berechnet, um den Abstand vom Wurzelhypernyme zu den Synsets zu ermitteln. Die Berechnung des Abstraktionsgrads a für eine Idee folgte nach der folgenden Formel (mit p_{syn} =Pfadlänge von einem konkreten synset zum Wurzelhypernym

und $||syn||$ = Anzahl der Synsets in der Idee):

$$a = (\sum_{syn} p_{syn})/||syn||$$

Nehmen wir 3 Ideen als Beispiel.
Idee 1 hat zwei Synsets: 4 ist der Abstand von Synset 1 zum Wurzel-Hypernym "*entity*" und 5 ist der Abstand von Synset 2 zum Wurzel-Hypernym "*entity*". Der Abstraktionsgrad der Idee 1 ist gleich $(4 + 5)/2 = 4,5$
Idee 2 hat drei Synsets: 4 ist der Abstand von Synset 1 zum Wurzel-Hypernym "*entity*", 5 ist der Abstand von Synset 2 zum Wurzel-Hypernym "*entity*" und das Synset 3 hat zwei Wege zum Wurzel-Hypernym "*entity*", den ersten Weg mit einem Abstand von 5 und den zweiten Weg mit einem Abstand von 7. Für dieses Synset ist die endgültige Entfernung zur Entität gleich $(5 + 7)/2 = 6$. Der Abstraktionsgrad der Idee 2 ist gleich $(4 + 5 + 6)/3 = 5$
Idee 3 hat zwei Synsets: 8 die Entfernung von Synset 1 zum Wurzel-Hypernym "*entity*" und 9 die Entfernung von Synset 2 zum Wurzel-Hypernym "*entity*". Der Abstraktionsgrad der Idee 1 ist gleich $(8 + 9)/2 = 8,5$

Die abstrakteste Idee wäre dann diejenige mit dem Kleinste Abstraktionsgrad und umgekehrt. Bzw. In unserem Beispiel wäre Idee 1 die abstrakteste und Idee 3 die konkreteste. Nach diesem Prinzip haben wir den Abstraktionsgrad aller 200 Ideen berechnet. Tabelle (Tabelle 4.2) listet die fünf abstraktesten und fünf konkretesten Ideen mit ihren unterschiedlichen Abstraktionsgraden an, die in den 200 Ideen enthalten sind. Laut WordNet ist Idee 17 mit einem Abstraktionsgrad von 9,75 am konkretesten und die Idee 151 mit einem Abstraktionsgrad von etwa 2,33 am abstraktesten.

ID von Ideen	Abstraktionsgrad	Inhalt
151	2,333	*It can be used for next generation Microsoft Kinect.*
24	2,8125	*Could be used by the Space Station to detect and analyze oncoming objects.*
174	3,25	*Astronomical observations: detecting objects in otherwise poorly monitored sections of the sky.*
83	3,5	*Look through forests to see all of the different animals and if there are any that have not been discovered*
184	3,716	*the device would be useful for predictive behaviors as an app for example, say someone wants to create a better lifestyle for themselves, they want to go to the gym more often or avoid fast food, they will be triggered by the app to make clear they've breached their range and must revert,perhaps it could also notify another person to prevent them from doing so. You could set parameters such as local fast food places and when the movement is predicted to be in that direction you're notified and made to stop.*
95	8,25	*I will see where an intruder is in my home at night.*
163	8,5	*I will know if a stranger is in my home.*
94	8,583	*Doorbell security systems that ID what's on the porch*
74	9	*Airplane navigation in low visibility*
17	9,75	*as a sensor on a car*

Tabelle 4.2: Die fünf abstraktesten und konkretesten Ideen aus dem Datensatz (nach Berechnung über WordNet Hypernyme)

5 Evaluation

Wir wollen nun die verschiedenen Ergebnisse des vorherigen Kapitels auswerten. Um die verschiedenen Informationen, zu bewerten, führten wir ein Interview mit einem Datenanalyse-Experten der HCC-Gruppe. Dieser Experte hatte die Daten bereits einmal analysiert und gehört zu den Forschern, deren Aufgabe es war, alle 581 gesammelten Ideen zu lesen, um sie besser zu verstehen und dann manuell Kategorien festzulegen. Mit Hilfe dieser Auswertung wollen wir drei wesentliche Punkte überprüfen.

- Zuallererst wollen wir wissen, ob Hypernyme als Kategorien betrachtet werden können.
- Zweitens wollen wir wissen, ob wir WordNet zur Berechnung des Abstraktionsgrades einer Idee verwenden können.
- Schließlich wollen wir wissen, ob es möglich ist, sich mit Hilfe von Hypernymen eine allgemeine Vorstellung von einer Gruppe von Ideen zu machen.

Zur Durchführung dieses Interviews haben wir die Methode des *Thinking Aloud*[1] angewandt, die es uns ermöglicht, einen *Usability-Test* durchzuführen, indem ein Benutzer laut nachdenkt. Zu diesem Zweck haben wir einen Leitfaden erstellt, in dem wir zunächst die verschiedenen Punkte und Ziele unserer Arbeit, die Art der eingegangenen Ideen und die verschiedenen Aspekte, die mit Hilfe dieser Ideen bewertet werden sollen, erläutert haben. Wir haben auch ein Jupyter-Notebook vorbereitet, in dem wir zunächst die annotierten Ideen, die verschiedenen Hypernyme, die wir extrahiert hatten, und die verschiedenen Wörterbücher, die im Unterkapitel zu Hypernymextraktion in Kapitel 3 erwähnt wurden, abrufen konnten. Diese Wörterbücher und ihre Funktionalitäten wurden dann im *Jupyter-Notebook* abgebildet. Wir haben dem Experten auch einige Funktionen zur Verfügung gestellt, die es ihm ermöglichen sollten, die Hypernyme und Ideen aus verschiedenen Perspektiven zu visualisieren. Um das Ziel dieser Evaluation zu erreichen, haben wir fünf Übungen formuliert, die der Experte mit Hilfe des Jupyter-Notebooks bearbeiten musste. Die verschiedenen Aufgaben waren:

1. Welche Themenbereiche gibt es in dem Datensatz?
2. Finde unerwartete Ideen oder Outlier.
3. Mit dem neuen Daten (reduzierte Daten mit genau einem Synset pro Idee): Was sind die Themenbereiche?

[1] https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/

4. Was sind die abstraktesten Ideen in dem Datensatz? Was sind die konkretesten Ideen in dem Datensatz? (5 Stück)
5. Welche wichtigen Informationen über die Daten könne wir mit Hilfe der bereitsgestellten Wörterbücher noch erhalten?

Nachdem wir uns mit den verschiedenen Übungen befasst hatten, führten wir ein 30-minütiges Interview mit dem Experten. Im Verlaufe dessen haben wir ihm einige Fragen gestellt. Während des gesamten Prozesses dieser Evaluation wurden Audioaufnahmen gemacht, ebenso wie eine Bildschirmaufnahme des Computers, mit dem der Experte die Übungen bearbeitet hat. Der Leitfaden und das für die Auswertung verwendete *Jupyter-Notebook* liegen dieser Arbeit am Ende bei 7.

Am Ende der Evaluierung konnten wir unsere Fragestellungen wie folgt beantworten:

- Die erste Frage konnten wir mit Hilfe der ersten, zweiten und dritten Aufgabe beantworten. Um die Kategorien zu finden, visualisierte der Experte das Diagramm, das die verschiedenen Hypernyme und die Anzahl der Ideen, die es zusammenfasst, zeigt. Dann visualisierte er auch dasjenige, das die verschiedenen Überschneidungen zwischen den Gruppen von Hypernymen zeigt. Der Experte konnte fast die gleichen Themen finden, die er mit Hilfe der manuellen Kategorisierung gefunden hatte. Dazu war es jedoch notwendig, die von WordNet bereitgestellten Hypernyme einzeln zu betrachten und zu bewerten, ob das jeweilige Hypernym informativ ist oder nicht. Außerdem stellte der Experte fest, dass es auch viele nutzlose Kategorien (Hypernyme) gab, die nicht oder bereits in der Beschreibung des Bionic Radar vorhanden waren. Darüber hinaus konnte der Experte die Kategorie "*livestock*" finden, die nur eine Idee sammelte und die auf unerwarteten Kategorien basierte. Der Experte gab uns auch zu verstehen, dass diese Kategorie von anderen Datenanalyse-Experten, die an dem Projekt teilnahmen, separat bewertet wurde, weil es sich um eine wirklich unerwartete Kategorie handelte. Deshalb sagte der Experte, dass er mit Hilfe unseres Ansatzes leicht die Kategorien einer einzigen Idee oder nur drei Prozent der Ideen in den unerwarteten Kategorien betrachten könne und er fand es sehr interessant. Daher meinte der Experten zum Schluss, dass Hypernyme als Kategorie zu betrachten sehr schwierig sei, da nicht alle vorhandenen Hypernyme auch Kategorien seien.
- Die zweite Fragestellung der Evaluation konnten wir mit Hilfe der vierten Aufgabe beantworten. Der Experte nach der Lektüre der fünf abstraktesten und konkretesten Ideen, die mit Hilfe unseres Ansatzes mit WordNet gefunden wurden. Er bemerkte schnell, dass nicht alle als sehr abstrakt klassifizierte Ideen überhaupt abstrakt sind. Dies gilt zum Beispiel für Idee 24, die seiner Meinung nach eher sehr detailliert ist, weil sie uns klar darauf hinweist, dass das Bionic Radar Objekte im Weltraum erkennen kann. Auch Idee 17 ist seiner Meinung nach abstrakt, obwohl sie von den

konkretesten Ideen ausgeht. Aber er fand einigen Ideen gut klassifiziert. Wir können daraus schlussfolgern, dass unser Ansatz für den Abstraktionsgrad einer Idee nach Ansicht des Experten Mängel aufweist.

- Die dritte Fragestellung der Evaluation konnten wir dank der 30-minütigen Befragung und der verschiedenen Anmerkungen des Experten während der Evaluation beantworten. Obwohl die Hypernyme laut Experte nur Konzepte sind, sind sie hilfreich, um einen groben Überblick über alle Ideen zu haben. Er konnte zum Beispiel leicht die unerwarteten Kategorien und auch interessante Konzepte finden, die sich auch beim Lesen der Ideen finden lassen.

Die oben erwähnten Punkte waren die Ergebnisse von der Evaluation. Im nächsten Kapitel werden die verschiedenen Ergebnisse diskutiert und Schlussfolgerungen für die gesamte Bachelorarbeit gezogen.

6 Diskussion

Um bewerten zu können, ob WordNet Annotationen helfen können, Ideenräume zu kategoriesieren und besser verständlich zu machen, führten wir ein Interview sowie einen Think-Aloud Test mit einem Datenanalyse-Experten der HCC-Gruppe durch. Nachfolgend werden die Ergebnisse der Evaluation erklärt, interpretiert und mögliche Lösungsansätze angeboten.

6.1 Qualität der WordNet Annotationen

Erstens haben wir in unserer Bachelorarbeit zunächst eine API mit WordNet erstellt, um die verschiedenen Ideen zu annotieren. Die Annotation war ein Zwischenschritt, der es uns ermöglichte, die Ideen zu kategorisieren. Nach der Annotation erwarteten wir konkrete und explizite Konzepte, die die Kernpunkte der Ideen beschreiben um den Forschern ein besseres Verständnis der Ideen zu ermöglichen. Allerdings konnten wir in den gesammelten Daten Begriffe wie *natural*, *use*, *kind* usw. finden, die für uns überhaupt nicht aussagekräftig sind. Der von uns befragte Experte war auch der gleichen Meinung: " *concept ,relation, communication oder thing* bringt mir alles nichts, damit kann ich nichts beschreiben. Sie sind mir zu abstrakt". Aus diesem Grund können wir eine gute Kategorie nicht auf der Grundlage der Anzahl der darin enthaltenen Ideen definieren. Tatsächlich sagt der Experte: "So was wie *natural, object...* sind nicht viel informativer würde ich sagen. Deswegen würde ich durchgehen und mir halt die ganzen Hypernyme anschauen und sagen okay hier es ist wie informativ das könnte ein topic sein und das könnte kein topic sein..." Das verrät uns, dass es unter den verschiedenen Synsets, die dank der Annotation gesammelt wurden, einige gibt, die für uns nicht von Nutzen sind. Es wäre vielleicht interessanter, die Synsets herauszufiltern, die keine Stoppwörter sind, aber uns nicht viele Informationen geben, indem Methoden von *Machine Learning* über den Informationsgehalt von jedem ausgewählten Synset berücksichtigen werden. Das könnte helfen, bessere Kategorien zu erzeugen.

Außerdem wäre es interessant, die Ideen von mehreren Personen annotieren zu lassen, da unsere gesamte Analyse auf der Annotation einer einzelnen Person basiert. Dazu sollten auch alle 581 Ideen, die über die Bionic Radar-Technologie gesammelt wurden, annotiert werden, um vielleicht auf genau dieselben Kategorien zu verweisen. Denn in unseren Daten gab es mehrere Kategorien, die nur eine Idee beinhalten. Wenn wir mehr Ideen hätten, könnten wir vielleicht bessere Ergebnisse erhalten.

6.2 WordNet als automatisches Maß für den Abstraktionsgrad einer Idee

Zweitens sollte die Abstraktion von Ideen es den Forschern ermöglichen, mehr Zeit beim Verstehen von Ideen zu sparen. Denn sehr abstrakte Ideen sollten inhaltsleer sein. Sie könnten die sehr konkreten Ideen priorisieren, um ein Maximum an Informationen zu sammeln und am Ende die weniger konkreten zu betrachten. Ebenso wie der Experte stellten auch wir fest, dass der mit Hilfe von WordNet definierte Abstraktionsgrad es uns nicht erlaubt, abstrakte und konkrete Ideen zu finden. Möglicherweise liegt es an der Methode, die bei der Definition des Abstraktionsgrades verwendet wird. Nach Ansicht des Experten können wir nicht davon ausgehen, dass zwei Synsets, die den gleichen Abstand zum Wurzelhypernym "*entity*" haben, auch gleich abstrakt sind. Bei der Datenanalyse konnte festgestellt werden, dass die Anzahl der Synsets von einer Idee zur anderen variiert. Darüber hinaus konnte im Hinblick auf die gesammelten Hypernymefestgestellt werden, dass einige Hypernyme mit der Mehrheit der Ideen manchmal sehr abstrakt waren, während Hypernyme mit wenigen Ideen meist sehr präzise waren. Ein besserer Ansatz könnte sein, den Abstraktionsgrad einer Idee mit einer Methode zu berechnen, die die Anzahl der Synsets, die sie umfasst, sowie den Informationsgehalt der Idee berücksichtigt.

6.3 Hypernyme als Kategorisierungs-Methode

Was den dritten Punkt betrifft, so ermöglichen Hypernyme effektiv das globale Verständnis einer Menge von Ideen. Aus den manuell entstandenen Kategorien (Abbildung 6.1) konnte man erkennen, dass die Mehrzahl der in diesen Kategorien vergebenen Punkte in den Hypernymen zu sehen war. Darüber hinaus konnte das Verständnis durch die Verringerung der Anzahl der Hypernyme verbessert werden. Unsere Übersichten ermöglichten es dem Wissenschaftler, mehr Zeit bei der Synthese der Ideen und deren Kategorisierung zu sparen.

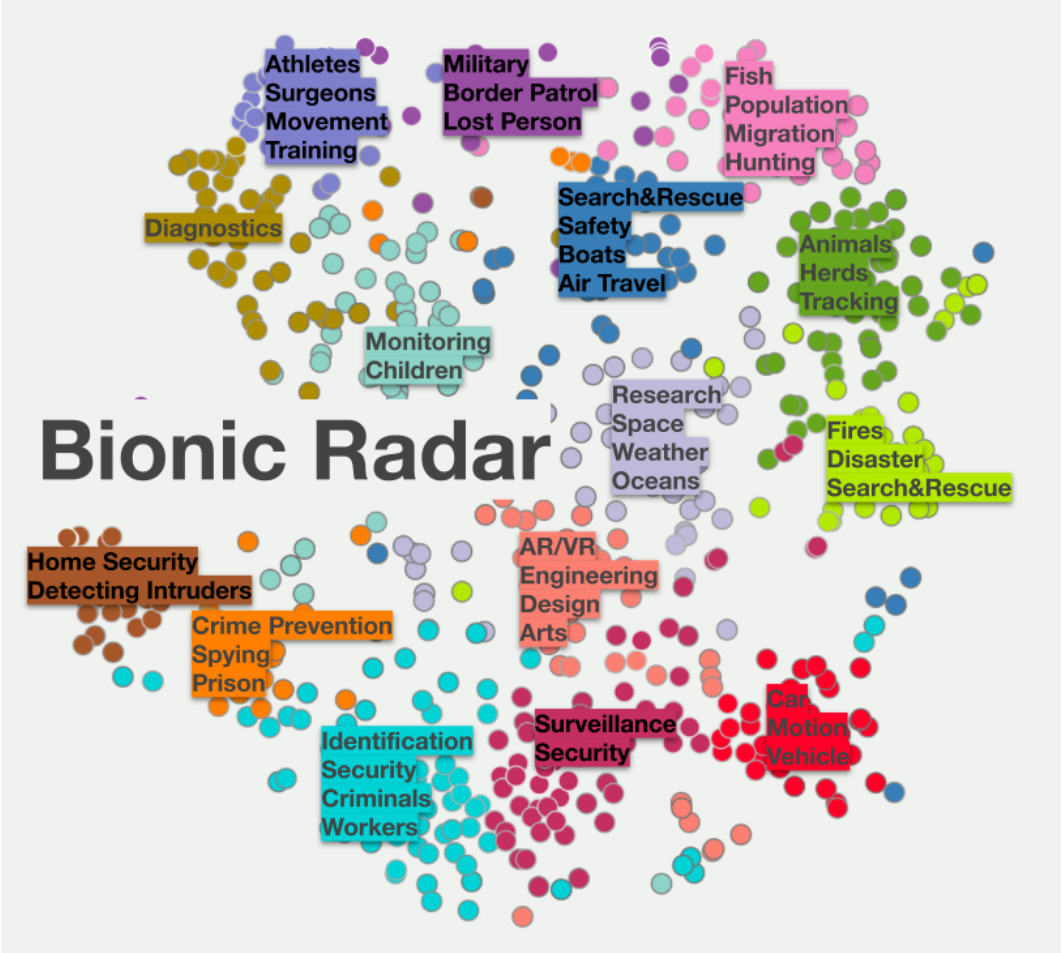


Abbildung 6.1: Manuelle Kategorien, die im Laufe des Innovonto-Projekts anhand der 581 Ideen gefunden wurden.

7 Zusammenfassung und Ausblick

Gegenstand der vorliegenden Arbeit war es, herauszufinden, inwieweit sich Ideen mit Hilfe von WordNet annotieren und mit Hilfe der von WordNet zur Verfügung gestellten Hypernymes kategorisieren lassen, um Wissenschaftlern das Verständnis von Ideen zu erleichtern und ihnen einen Überblick über die Idee zu ermöglichen um mehr Zeit zu sparen. Das Hauptziel wurde in drei Teilaufgaben unterteilt. In diesem Kapitel werden wir zunächst die Maßnahmen zur Beantwortung dieser Zwischenziele im Detail beschreiben und dann einen kurzen Überblick über die Ergebnisse dieser Einzelziele geben.

Das erste Teilziel bestand darin zu evaluieren, inwieweit wir WordNet in die bestehende ICV-Software integrieren können, die bisher die Annotation von Ideen nur über die Abfrage von DBpedia ermöglichte. Da WordNet uns eine Bibliothek zur Verfügung stellt, die es uns erlaubt, die verschiedenen darin enthaltenen Informationen zu sammeln, haben wir eine WordNet-API erstellt, die es uns erlaubt, eine Idee zu annontieren. Die API erhält die Idee über eine URL. Für jede erhaltene Idee filtert die API zunächst die Stoppwörter heraus und extrahiert dann für jedes verbleibende Wort die verschiedenen Bedeutungen dieses Wortes, die von WordNet Synsets genannt werden. Diese verschiedenen Synsets werden dem Benutzer dann während des Annotationsprozesses vorgeschlagen. Die in der Arbeit entwickelte Software-Komponente ließ sich über die Bereitstellung einer API in die bestehende ICV-Software einbinden.

Das zweite Teilziel bestand darin, herauszufinden, wie aus Ideen hypernyme Kategorien geschaffen werden können. Zu diesem Zweck haben wir das Ergebnis der Idee-Annotation verwendet, um die Hypernyme zu extrahieren. Für jedes in der Annotation ausgewählte Synset haben wir alle verschiedenen Hypernyme aus der Gruppe extrahiert. Dann gruppierten wir die Ideen und Synsets nach den gemeinsamen Hypernymen, die jeder von ihnen besitzt. Auf diese Weise wurden Kategorien von Hypernymen gebildet.

Das dritte Teilziel bestand darin, herauszufinden, wie Hypernym-Kategorien zum Verständnis einer Gruppe von Ideen beitragen können. Zu diesem Zweck haben wir 200 der im Innovonto-Projekt gesammelten Ideen für den Einsatz der neuen Bionic-Radar-Technologie verwendet. Zunächst wurden die 200 Ideen mit Hilfe der ICV-Software mit dem neuen WordNet-Backend annotiert. Dann wurden aus den empfangenen Synsets die Hypernyme extrahiert und auf dieser Datengrundlage eine deskriptive Statistik. Während dieser Deskriptiven Statistik wurden zwei wesentliche Aspekte analysiert.

Die erste war die von WordNet zur Verfügung gestellte Hypernym-Kategorie. Der zweite war der Abstraktionsgrad einer Idee. Wir haben anschließend die mit Hilfe eines Experten-Interviews mit der "*Thinkind Aloud*"-Methode erzielten Ergebnisse evaluieren. Die Ergebnisse der Analyse und Evaluation zeigten, dass die von WordNet bereitgestellten Hypernyme-Kategorien keine guten Kategorien waren und dass die Anzahl der Ideen, die ein Hypernym enthält, nicht als beste Kategorie gelten kann. Die Ergebnisse haben jedoch gezeigt, dass es möglich ist, einen globalen Überblick über eine Gruppe von Ideen zu erhalten, wenn Hypernyme verwendet werden. Die Ergebnisse haben jedoch, dass der Abstraktionsgrad nicht funktioniert. Denn zu den abstraktesten Ideen gehörten die ganz konkreten.

Ursprünglich war geplant, ein Dendrogramm zur Visualisierung der empfangenen Hypernyme zu verwenden, um die in den Daten enthaltenen Kategorien besser erkennen zu können. Da aber ein Hypernym gleichzeitig ein Oberbegriff eines in einer Idee enthaltenen Synsets sein kann, kann es auch ein in einer anderen Idee enthaltenes Synset sein. Dies machte den Aufbau des Dendrogramms sehr schwierig. Schließlich wird es interessant sein, andere Analyse- und Visualisierungsmethoden zu verwenden, um die von WordNet erhaltenen Synsets und Hypernym-Kategorien untersuchen zu können, um besser beurteilen zu können, in welchem der Perspektive diese Daten zum Verständnis der Ideen beitragen können, denn da es sich um einen Bachelorarbeit handelt, haben wir nicht genug Zeit, um nach anderen Methoden zu suchen.

Am Ende dieses Bachelorarbeit haben wir ein funktionales ICV-Backend entwickelt, das es ermöglicht, Ideen mit WordNet zu annotieren. Im Hinblick auf die Ideenkategorisierung mit Hypernymen haben wir Funktionen wie die Hypernymextraktion aus einer Gruppe von Ideen und deren Speicherung in verschiedenen Arten von Wörterbüchern bereitgestellt und damit eine solide Grundlage für die weitere Analyse und Visualisierung geschaffen, um die von WordNet bereitgestellten Hypernyme besser zur Kategorisierung von Ideen zu nutzen. Die beiden verwendeten Analysekriterien können auch als Grundlage für Datenanalytiker dazu beitragen, andere Analysepunkte zu definieren, um Hypernyme zur Kategorisierung von Ideen zu verwenden.

Literaturverzeichnis

[AMM+15] Fankar Armash Aslam, Hawa Nabeel Mohammed, Jummal Musab Mohd, Murade Aaraf Gulamgaus, and PS Lok. Efficient way of web development using python and flask. *International Journal of Advanced Research in Computer Science*, 6(2), 2015.

[Com20] Human-Centered Computing(HCC). Innovonto/Ideas-to-Market. https://www.mi.fu-berlin.de/en/inf/groups/hcc/research/projects/innovonto/index.html, 2020. [Online; accessed 12-Januar-2020].

[GCN+18] Karni Gilon, Joel Chan, Felicia Y Ng, Hila Liifshitz-Assaf, Aniket Kittur, and Dafna Shahaf. Analogy mining for specific design needs. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 121. ACM, 2018.

[Gee20] GeeksforGeeks. Nested-dictionary. https://www.geeksforgeeks.org/python-nested-dictionary/, 2020. [Online; accessed 12-Januar-2020].

[GWB19] Victor Girotto, Erin Walker, and Winslow Burleson. Crowdmuse: Supporting crowd idea generation through user modeling and adaptation. In *Proceedings of the 2019 on Creativity and Cognition*, pages 95–106. ACM, 2019.

[LB02] Edward Loper and Steven Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[Mil98] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[MKS+19] Maximilian Mackeprang, Abderrahmane Khiat, Maximilian Stauss, Tjark Sascha Müller, and Claudia Müller-Birn. The impact of concept representation in interactive concept validation (icv). 2019.

[MMBS19] Maximilian Mackeprang, Claudia Müller-Birn, and Maximilian Timo Stauss. Discovering the sweet spot of human-computer configurations: A case study in information extraction. *arXiv preprint arXiv:1909.07065*, 2019.

[RP17] Daniel Ringler and Heiko Paulheim. One knowledge graph to rule them all? analyzing the differences between dbpedia, yago, wikidata & co. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 366–372. Springer, 2017.

[Sia15] Pao Siangliulue. Supporting collaborative innovation at scale. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 9–12. ACM, 2015.

[Sia17] Kanya Pao Siangliulue. *Supporting Effective Collective Ideation at Scale*. PhD thesis, 2017.

[Uni09] Princeton University. Wordnet: a lexical database for the english language, 2009.

Appendix

7.1 Leitfaden

Interviewleitfaden

Ingrid Tchilibou

5. März 2020

Inhaltsverzeichnis

1 Chek-In

Bevor wir beginnen, lesen und unterschreiben Sie bitte das bereitgestellte Einverständniserklärung.

2 Einleitung

An diesem Tag wollen wir unsere Jupyter Notebook mit verschiedenen Ergebnisse, die wir im Rahmen von Projekt " Konzept Annotation basierend auf WordNet " erzielt haben, auswerten. Das Ziel ist es, den Forschern zu helfen, eine Menge von Ideen zu verstehen, indem sie diese in Kategorien gruppieren. Genauer dann ein Überblick über einer Menge an Ideen zu bekommen. Eine Kategorie ist in unserem Fall ein Wort (synset), das mehrere andere Wörter (synsets) gruppiert, genauer gesagt ein Hypernyme.

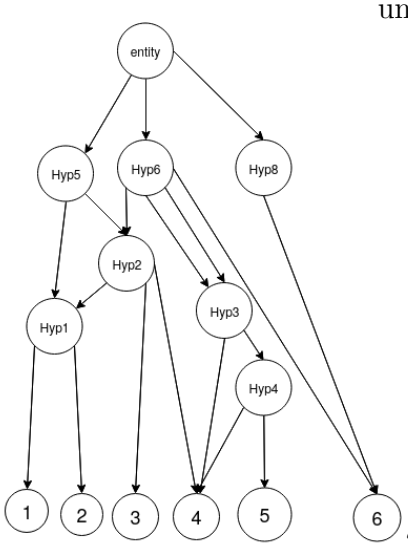
Im Laufe dieses Projekts erhielten wir 200 der 581 Ideen aus dem Bionic Radar Project. Diese Ideen wurden erstmal mit Hilfe des Wordnet-API-Backends von ICV annotiert. Für jede Idee, die wir erhielten, wählten wir das/die Synset(s) aus, das/die die einzelnen Wörter der zu annotierenden Idee am besten beschreibt/beschreiben. Am Ende erhielten wir eine JSON-Datei mit all unseren Entscheidungen. Dann haben wir aus diesem Datensatz die Hypernyme jedes ausgewählten Synset extrahiert und in verschiedenen Wörterbücher gespeichert.

Bevor wir fortfahren, haben Sie Fragen?

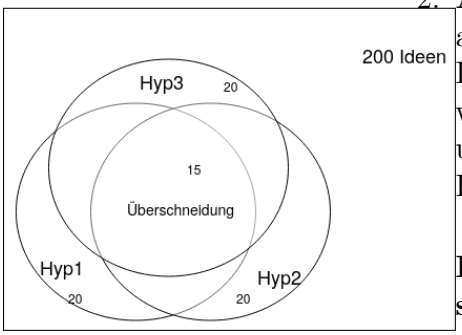
3 Evaluation

Wir werden Ihnen zunächst die verschiedenen Punkte erklären, die wir bewerten wollen.

- Zuerst wollen wir mit dieser Evaluation, die Themen(Hypernyme) finden, die die maximale und minimale Ideen beschreibt.



1. **Ansatz 1:** Nehmen wir zum Beispiel sechs Ideen, jede Idee enthält mehrere Hypernymes und als top hypernyme ist *entity*. Das Ziel ist es, den Hypernym zu finden, der möglichst viele Ideen enthält. Auf unserer Grafik enthalten Hypernym 6 und Hypernym 5 jeweils 4 Ideen. Dies sind für uns also die besten Themen, die diese 6 Ideen am besten beschreiben. Andererseits enthält Hypernyme 8 nur eine Idee (Idee 6), die als unerwartetes Thema betrachtet wird.



2. **Ansatz 2:** Nehmen wir zum Beispiel an, dass wir 3 Hypernyme mit jeweils 20 Ideen (Fenster größe) haben. Analysieren wir die Schnittmenge dieser 3 Hypernyme, um herauszufinden, wie viele Ideen die Hypernyme 1, 2 und 3 gemeinsam haben.

   Frage: Wie groß muss dass Fenster sein damit man eine ” gute ” Liste an Hypernymen bekommt?
   Was heißt ” gute ” ?
   Überschneidung $> 95\%$

- Weiterhin haben wir jede Idee auf ein einziges Synset reduzieren, so dass wir die Anzahl der Hypermymien reduzieren und sie besser analysieren können. Dazu haben wir für jede Idee das beste Synset gefunden, das diese Idee am besten beschreibt.
  Was ist das beste Synset?
  Das Synset, das genau einmal in unseren Daten erscheint. Nachdem wir die Synsets, die mehr als einmal in unseren Ideen erscheinen, eliminiert haben. Wir stellten fest, dass es Ideen gab, die völlig verschwunden waren, weil sie nur bereits verwendete Synsets enthielten. Für diejeni-

gen, die wir sie getrennt untersucht haben, haben wir die Anzahl der Wiederholungen jedes Mal erhöht, um die beste Synset zu finden.
Für die Ideen von mehrfach vorkommenden Synsets, die nur einmal in unseren Daten vorkommen, haben wir nur das erste Synset genommen.

Frage : Wird es dadurch leichter, die besten Themen zu bekommen?

- Wir wollen unsere Daten auch nutzen, um den Abstraktionsgrad einer Idee zu bewerten.
  Aber was ist der Abstraktionsgrad einer Idee?
  Nehmen wir 3 Ideen als Beispiel.
  Idee 1 hat zwei Synsets: 4 ist der Abstand von Synset 1 zum Wurzel-Hypernym "*entity*" und 5 ist der Abstand von Synset 2 zum Wurzel-Hypernym "*entity*". Der Abstraktionsgrad der Idee 1 ist gleich (4+5)/2 = 4,5

  Idee 2 hat drei Synsets: 4 ist der Abstand von Synset 1 zum Wurzel-Hypernym "*entity*", 5 ist der Abstand von Synset 2 zum Wurzel-Hypernym "*entity*" und das Synset 3 hat zwei Wege zum Wurzel-Hypernym "*entity*", den ersten Weg mit einem Abstand von 5 und den zweiten Weg mit einem Abstand von 7. Für dieses Synset ist die endgültige Entfernung zur Entität gleich (5+7)/2 = 6. Der Abstraktionsgrad der Idee 2 ist gleich (4+5+6)/3 = 5

  Idee 3 hat zwei Synsets: 8 die Entfernung von Synset 1 zum Wurzel-Hypernym "*entity*" und 9 die Entfernung von Synset 2 zum Wurzel-Hypernym "*entity*". Der Abstraktionsgrad der Idee 1 ist gleich (8+9)/2 = 8,5

  Die abstrakteste Idee wäre dann diejenige mit dem Kleinste Abstraktionsgrad und umgekehrt. Bzw. In unserem Beispiel wäre Idee 1 die abstrakteste und Idee 3 die konkreteste.

Bevor wir fortfahren, haben Sie Fragen?
Wir haben Ihnen einen Jupyter-Notebook mit verschiedenen Wörterbüchern

zur Verfügung gestellt, die Informationen enthalten, die während unserer Analyse der Daten gesammelt wurden. Wir haben in diesem Jupyter-Notebook jedes Wörterbuch und seine Funktionalität sowie die verschiedenen Funktionen erklärt, die wir Ihnen bei Ihrer Analyse zur Verfügung gestellt haben.

Um fortzufahren, haben Sie 10 Minuten Zeit, das Jupyter-Notebook zu lesen, um zu verstehen, wie diese verschiedenen Punkte umgesetzt wurden und welche verschiedenen Funktionen Ihnen zur Beantwortung der verschiedenen Fragen der Evaluation zur Verfügung stehen.
Bevor wir fortfahren, haben Sie Fragen?

4 Aufgabe

Sie haben nun 20 Minuten Zeit, um die verschiedenen gestellten Fragen zu beantworten.

1. Welche Themenbereiche gibt es in dem Datensatz?
2. Finde unerwartete Ideen oder Outlier (5 Stück)
3. Was sind die abstraktesten Ideen in dem Datensatz? Was sind die konkretesten Ideen in dem Datensatz? (5 Stück)

Bevor wir fortfahren, haben Sie Fragen?

Sie haben nun 10 Minuten Zeit, um ein wenig mit dem Jupyter-Notebook zu spielen, um weitere Informationen zu finden, die wir mit unseren Daten noch erhalten könnten.

Frage: Welche wichtigen Informationen über die Daten können wir mit Hilfe der bereitgestellten Wörterbücher noch erhalten?

5 Interview

Nun werden wir mit einem 30-minütigen Interview fortfahren, um Ihre verschiedenen Eindrücke von dem, was wir gerade getan haben, zu bewerten.

1. Wie konntest du die Aufgaben bearbeiten? Hattest du Probleme bei einer der Aufgaben?

2. Wie schätzt du den Ansatz mit Wordnet ein? Was hat dir gut gefallen, was hat dir nicht gefallen? Hat dir der Ansatz geholfen einen Überblick über die Ideen zu bekommen?

3. Was könnten andere Data-Scientists mit diesem Jupyter notebokk machen?

4. Was könnten die Forscher damit anfangen?

5. Was für Aufgaben hast du mit den Bionic Radar Daten bisher gemacht?

7.2 Bewertung

Bewertung

March 4, 2020

1 Bewertung für Bachelorarbeit

Autorin: Frau Ingrid Tchilibou

Bewerter: Herr Michael Tebbe

1.0.1 1.1 Datensatz von Annotation laden

[ ]:
```
import json
import import_ipynb
import extract_hypernyms as EH
# read file
if __name__ == '__main__':

    #data loaded
    with open('export200.json', 'r') as myfile:
        data = myfile.read()
    ideen_list   = json.loads(data)
```

1.0.2 1.2 Ideen Wörterbuch (key = (nummerId , Id) , Value = Content)

Alle 200 Ideen werden in der ideenDict mit entsprechen nummerID und Id gespeichert

[ ]:
```
ideenDict    = EH.get_dictionaryIdee(ideen_list)
```

1.0.3 1.3 DictionaryTree: Beziehungen zwischen Hypernym

Key = ein Hypernym(Eltern Knoten)

Value = List von Hypernyme (Kinder Knoten die als Direkthypernym der Eltern Knoten haben)

i) Verfügbare Funktionen

   a) EH.build_tree(dictionaryTree) Zeigt der gesamte Baum

   b) Ein Spezifisches Knoten mit Kinder als unterbaum anzeigen lassen EH.build_part_of_tree_(Tree,knoten) wobei knoten den form synset.pos.id hat Beispiel: EH.build_part_of_tree(dictionaryTree, "entity.n.01")

[ ]:
```
synsetsList    = EH.get_synsetsList(ideen_list)   #key = Synset, Value = [list␣
 ↪von ID]
dictionaryTree = EH.analyse(synsetsList)
```

1.0.4 1.4 SynsetsList,hypernymList, ideen_synset, ideen_hypernym Wörterbücher

a) Verfügbare Funktionen:

i) EH.plot_balken:hypernym(hypernymList,prozent): Balkendiagramm Darstellung: Anzahl von Ideen pro Hypernyme Hypernyme mit Anzahl an Ideen weniger als "prozent" werden in Others gruppiert. Und "Others" wird als eine geordnete Liste von der kleinsten bis zur größten angezeigt. Beispiel: EH.plot_balken_hypernym(hypernymList,20)

ii) EH.plot_hypernyme_fenster(hypernymList,minimum,maximum) Hypernym mit Anzahl an Ideen Zwischen minimum and maximum Tabelle mit Anzahl an Ideen, Überschneidung und Anzahl an Hypernyme Beispiel: EH.plot_hypernyme_fenster(hypernymList,26,40)

iii) EH.plot_mapping_idee_Synstet(ideen_synset,prozent) Balkendiagramm Darstellung: Anzahl von Synset pro Idee IdeenId mit Anzahl an Synsets weniger als "prozent" werden in Others gruppiert. Und "Others" wird als eine geordnete Liste von der kleinsten bis zur größten angezeigt.

iv) EH.plot_mapping_idee_Hypernym(ideen_hypernym,prozent) Balkendiagramm Darstellung: Anzahl an Hypernyme pro Idee IdeenId mit Anzahl an Hypernyme weniger als "prozent" werden in Others gruppiert. Und "Others" wird als eine geordnete Liste von der kleinsten bis zur größten angezeigt.

[ ]:
```
hypernymList      = EH.get_hypernymDict(synsetsList) #Key = hypernym ,      Value␣
 ↪= [list von ID]
ideen_synset      = EH.mapping_idee_to_synsets(ideen_list) #key = (nummerId,Id)␣
 ↪Value = [Listsynsets]
ideen_hypernym    = EH.mapping_idee_to_hypernym(ideen_synset) # key =␣
 ↪(nummerId,Id)  Value = [Listhypernymen]
print("Anzahl Synsets", len(synsetsList.keys()))
print("Anzahl Hypernym", len(hypernymList.keys()))
```

Frage 1: Welche Themenbereiche gibt es in dem Datensatz ?

[ ]:
```
EH.plot_hypernyme_fenster(hypernymList,5,7)
```

59 ideas: application, technology, profession -> technology can be used to do a job rest maybe: tech can be used at home, not on the job (e.g. sports)

[ ]:
```

```

[ ]:
```
#Antwort:
#print("Synsets", synsetsList.keys())
```

```
#print("Hypernyms", hypernymList.keys())
EH.plot_balken_hypernym(hypernymList,10)
```

EH.plot_balken_hypernym(hypernymList,5) Topics: bad person invesigation building container mammal.... 11 - 16 occurences

EH.plot_balken_hypernym(hypernymList,10) Topics: residence home ... - technology

Frage 2: Finde unerwartete Ideen bzw. unerwartete Themenbereiche oder Outlier (5 Stück)

[ ]:
```
#Antwort:
Livestock, Student
```

1.0.5 1.5 Daten Reduktion: Pro Ideen genau 1 Synset

Synset die am Wenigstens in Datensatz vorkommt bzw. genau ein mal!

a) verfügbare Funktionnen:

i) EH.plot_occurence_synset(synsets_occurence,prozent)
Balkendiagramm Darstellung: Pro Synset wird die Anzahl an wiederholung in der Datensatz ange
Synsets mit Anzahl an Occurence weniger als "prozent" werden in Others gruppiert.
Und "Others" wird als eine geordnete Liste von der kleinsten bis zur größten angezeigt.

ii) Alle in 1.3 genanten Tree Funktionnen

[ ]:
```
synsets_occurence        = EH.occurence_of_synset(ideen_synset) #key = Synset,␣
 ↪Value = Anzahl von vorkommen
ideen_synsetList_clearly = EH.ideen_synsetDict_clearly(ideen_synset,␣
 ↪synsets_occurence)#key = (nummerId,Id), Value = [ein Synset]
synsetList_clearly       = EH.synsetDict_clearly(ideen_synsetList_clearly,␣
 ↪synsetsList )#key = Synset, Value = [list von ID]
dictionaryTree_clearly   = EH.analyse(synsetList_clearly)
```

Frage 3: Mit dem neuen Daten was sind Die Themenbereiche?

[ ]:
```
#Antwort

hypernymList_clearly      = EH.get_hypernymDict(synsetList_clearly)
EH.plot_balken_hypernym(hypernymList_clearly,1)

#filter through hypernyms and choose tzhe ones that might be topics
```

1.0.6 1.6 Abstraktionsgrad eine Idee

Verfügbare Funktionnen:

i) EH.Top_Abstract_and_concret(Abstractionsgrad,ideenDict,top)
gibt ein Tupel mit top Abstrakte und Konkrete Ideen

```
beispiel: listTop2_Abstrakt,listTop2_KonKret = EH.Top_Abstract_and_concret(Abstractionsgrad,id
```

[ ]:
```
Abstractionsgrad = EH.Abstractionsgradfunc(ideen_synset)  #key = (numId,Id)␣
↪value = distance
```

Frage 4: Was sind die abstraktesten Ideen in dem Datensatz? Was sind die konkretesten Ideen in dem Datensatz? (5 Stück)

[ ]:
```
#Antwort
listTop5_Abstrakt,listTop5_KonKret = EH.
↪Top_Abstract_and_concret(Abstractionsgrad,ideenDict,5)

print('abstract:', listTop5_Abstrakt)
print()
print('konkret: ', listTop5_KonKret)
```

Frage 5: Welche wichtigen Informationen über die Daten können wir mit Hilfe der bereitgestellten Wörterbücher noch erhalten?

2 Antwort

find rare ideas by looking at overlaps between synsets. Hypothesis: rare ideas produce 'strange' overlaps

synsets per idea k tree of one idea

compare two trees

use corpus-based methods to filter out uninformative topics