

3 Aufgaben

Der Zeitraum des Praktikums deckt sich mit der ersten Hälfte der Projekt-Laufzeit (November und Dezember 2021 sowie Januar 2022). Zentrale Aufgaben in dieser Phase waren die Vorbereitung des Korpus, die Auswahl des Annotations-Schema und -Tools sowie die händische Annotation. In der zweiten Hälfte ist geplant, die Annotationsarbeit weitestgehend abzuschließen und das finale Produkt in Form des Websystems zu entwickeln. Meine Aufgaben waren dabei:

1. Vorbereitung des Korpus

- (a) Identifizierung der Seiten aus den Digitalisaten der Chronik, die die Jahresberichte des Naturkundemuseums enthalten
- (b) Crawling der Digitalisate
- (c) Simple OCR-Korrektur

2. Entwicklung des Annotations-Schema

- (a) Identifizieren der hauptsächlichen in den Jahresberichten angesprochenen Themen
- (b) Vertraut-werden mit CIDOC-CRM
- (c) Extrahieren eines Subsets von Klassen aus dem Schema, das die identifizierten Themen ausreichend modellieren kann

3. Annotation

- (a) Aufbau der Annotations-Layer in INCEpTION
- (b) Annotieren
- (c) Dokumentation des Annotations-Fortschritts und von typischen Mustern in der Annotation mit Beispielen

4. Postprocessing & Repräsentation der Annotationen

- (a) Parsen des Exports von INCEpTION und Modellierung als Python-Objekte
- (b) Bereinigen der OCR-Texte durch die Annotationen aus dem Korrektur-Layer
- (c) Generierung und Verknüpfung der „virtuellen“ impliziten Entitäten aus dem semantischen Annotations-Layer, die nicht explizit im Text genannt werden aber für das semantische Verständnis bzw. für die korrekte Anwendung von CIDOC-CRM voneinander unabhängig sind
- (d) Exportieren der Annotationen als JSON, GRAPHML und RDF-Triples
- (e) Visualisierung der einzelnen Annotationen als Graphen durch Graphviz
- (f) Erstellung einer Website mit der der semantische Graph durchsuch- und durchwanderbar ist. Drei unterschiedliche Zugänge sollen realisiert werden:
 - i. Textuell: Suchmaske, mit der eine Expert*in mit Vorwissen die annotierten Zeichenketten durchsuchen kann. Darstellung des Ergebnisses im Text der Chronik sowie tabellenartige Auflistung der jeweiligen semantischen Verknüpfungen.

- ii. Visuell: Ausschnitte aus einer gezeichneten Version des semantischen Graphen, durch die man interaktiv den Graphen durchwandern kann.
- iii. Aufbereitete Analysen: vorab erstellte Analysen des semantischen Netzes durch Grafiken und Tabellen

4 Verwendete Tools und Methoden

Im Folgenden beschreibe ich die vier oben skizzierten Mantelaufgaben ausführlicher, indem ich näher auf die einzelnen verwendeten Methoden eingehe.

4.1 Vorbereitung des Korpus

Die Chronik wurde von der Bibliothek der HU Berlin bereits digitalisiert, mit OCR maschinenlesbar gemacht und im Internet veröffentlicht [1]. Mein erster Schritt beim Aufbau des Korpus war es, die Passagen aus den Bänden der Chronik händisch zu identifizieren, die die Jahresberichte der einzelnen Sammlungen des Museums enthalten. Nach einer Diskussion, bei der entschieden wurde, dass wir mit der CC-BY-NC-SA Lizenz, unter der die Digitalisate stehen, arbeiten können, schrieb ich ein Python-Skript, dass die Texte der identifizierten Seiten gecrawlt hat. Als Vorbereitung zur Annotation lies ich die Texte durch ein simples Python-Skript bereinigen, indem ich die Zeichen normalisierte und typische Fehler bei deutschen OCR-Prozessen regelbasiert korrigierte (z.B. durch Ersetzen von `ii` durch `ü` wenn das Wort dadurch in einem deutschen Wörterbuch vorkommt).

4.2 Annotations-Schema/ Ontologie

Das Ziel bei der Datenmodellierung war es, die Informationen aus den Jahresberichten möglichst historisch adäquat, d.h. quellennah, abzubilden. Deshalb habe ich mich für eine semantische Annotation als Datenmodell entschieden, da dadurch die erschlossenen Informationen nicht zu stark abstrahiert werden. Das Prinzip der semantischen Annotation besteht aus der Klassifikation von Begriffen aus einer Quelle, in meinem Fall also der Jahresberichte des Naturkundemuseums, nach semantischen Klassen einer standardisierten Ontologie und der Zuweisung von klassifizierten Beziehungen zwischen Instanzen der Klassen. Unter dem Gesichtspunkt des zugrundeliegenden Datenschema soll das Produkt zwei Kriterien erfüllen. Meine semantischen Annotationen sollen eine Grundlage für weitere Erschließungen von historischen Daten des Naturkundemuseums bilden. Deshalb soll das Annotations-Schema erstens semantische Verbindungen möglichst generell definieren, sodass in das Produkt in Zukunft noch weitere, bisher unbekannte Daten einfach integriert werden können, also prinzipiell nicht zu überangepasst auf die in den Jahresberichten angesprochenen Themen sein. Zweitens soll das Schema nicht zu komplex sein, damit ich die Texte möglichst schnell und effizient annotieren kann, ohne viel in der Dokumentation nach dem richtigen Mustern suchen zu müssen.

Das International Committee for Documentation des International Council of Museums entwickelt seit über 20 Jahren das CIDOC Conceptual Reference Model (CRM) als standardisierte Ontologie, das dafür gedacht ist, vor allem Daten aus meinem Forschungsgebiet, der Museumsgeschichte, modellieren zu können [3]. Ich habe mich deshalb entschieden, mit der

neuesten Version (7.1.1) zu arbeiten. Die semantischen Klassen sind in dieser Ontologie hierarchisch geordnet, sodass eine Kindklasse einen spezifischeren Ausdruck als seine Elternklasse beschreiben kann. Eine Instanz einer semantischen Klasse kann durch fest definierte und hierarchisch vererbbarer Eigenschaften oder Properties mit anderen Instanzen von semantischen Klassen verbunden werden. Damit ich einem möglichst effizienten Annotationsprozess folgen kann, wählte ich aus dieser Ontologie die semantischen Klassen und dazugehörigen Relationen aus, die sich eignen die typischerweise in den Jahresberichten auftauchenden Themen abbilden zu können. Die Themen identifizierte ich, indem ich einzelne Stichproben der Jahresberichte grob nach den darin angesprochenen Beiträgen annotierte. Dadurch war es mir möglich, ein Subset mit den Klassen zusammenzustellen, die ich auch wirklich für die Annotationsarbeit brauche. Das Kennenlernen der Charakteristika von CIDOC-CRM sowie das Auswählen und die Dokumentation des Subsets mit typischen Beispielen aus den Jahresberichten beanspruchte etwa zwei Wochen meiner Arbeitszeit.

4.3 Annotation

Aufgrund der überschaubaren Zahl von etwa 1000 Seiten, habe ich dem Team vorgeschlagen, dass sich eine komplett händische Annotation eher lohnen würde als das Training eines eigenen Modells, das die semantischen Entitäten selbstständig vorhersagt. Meine Befürchtung war, dass ich mindestens genauso viel Zeit in die Entwicklung eines robusten Modells investieren müsste wie in die manuelle Arbeit. Für die Annotationen der Texte habe ich mich entschieden, das Tool INCEpTION zu verwenden [4]. Ausschlaggebend war die Vielzahl an Möglichkeiten zur Anpassung der Annotationsumgebung¹, die gute Darstellung von Annotationen, die viele Verbindungen mit anderen Annotationen haben, die Möglichkeit zur kollaborativen Annotation mittels der Implementation eines Webservices² und eine aktive Entwicklung mit regelmäßigen abwärtskompatiblen Updates.

Das Fundament bei der Annotation mit INCEpTION sind Layers, die man in den Settings eines Projektes anlegen kann. Zeichenketten aus dem zu annotierenden Text können dann pro Layer markiert und mit weiteren Informationen angereichert werden, deren Struktur man in einzelnen Features eines Layers definiert. Für die Annotation der Jahresberichte habe ich drei Layer eingerichtet:

- 1. OCR-Corrections:** zur händischen Korrektur von OCR-Fehlern. Das Schema der dazugehörigen Features ist in Tabelle 1 veranschaulicht.

Features des OCR-Correction Layers		
Feature Name	Type	Erklärung
CorrectedString	String	Zeichenkette, die die Zeichen in der Annotation ersetzen sollen
Deletion	Boolean	Zeigt an, ob annotierte Zeichen gelöscht werden können (da Artefakte durch OCR)

Tabelle 1: Schema des OCR-Correction Layers.

¹Z.B. Festlegung der Anzahl an angezeigten Zeilen, Auswählen der Farben bei der Abbildung der Annotationen sowie typengerechte Darstellung von Features des Annotationslayers (so wird ein Feature, das ein booleschen Wert erwartet durch eine binäre Box angezeigt).

²Auch wenn ich das schlussendlich nicht nutzte.

2. **SemanticEntities**: zum Klassifizieren von Zeichenketten nach semantischen Klassen aus CIDOC-CRM. Das Schema der dazugehörigen Features ist in Tabelle 2 veranschaulicht.

Features des SemanticEntities Layers		
Feature Name	Type	Erklärung
Class	Enum[String]	Semantische Klasse der Annotation aus dem CIDOC-CRM Tagset
Virtual	Boolean	Zeigt an, ob eine Entität der angegebenen semantischen Klasse nur impliziert wird, annotierte Zeichenkette wird im Postprocessing als null gesetzt
Type	String	Zum effizienten Hinzufügen eines Typs, Inhalt wird im Postprocessing modelliert zu [P2 has type] → E2 Type : String
Postprocessing	String	Möglichkeit zum Hinzufügen von weiteren Informationen, die beim Postprocessing modelliert werden, z.B. Hinzufügen weiterer Types

Tabelle 2: Schema des SemanticEntities Layers.

3. **SemanticProperties**: zur Herstellung von semantischen Verbindungen zwischen annotierten semantischen Entitäten aus dem SemanticEntities-Layer, wiederum jeweils klassifiziert nach den Definitionen und Regeln von CIDOC-CRM. Das Schema der dazugehörigen Features ist in Tabelle 3 veranschaulicht.

Features des SemanticProperties Layers		
Feature Name	Type	Erklärung
Class	Enum[String]	Semantische Klasse der Verbindung aus dem CIDOC-CRM Tagset
Source	Instanz im SemanticEntities Layer	Ursprung der semantischen Verbindung
Target	Instanz im SemanticEntities Layer	Ziel der semantischen Verbindung

Tabelle 3: Schema des SemanticProperties Layers.

Eine Schwierigkeit, die den Annotationsprozess bedeutend langwieriger gestaltete als ich ursprünglich angenommen hatte, resultierte aus einerseits der sehr dichten Beschreibung von Ereignissen in den Jahresberichten und andererseits der doch sehr starren Ontologie (wenn einem die korrekte Anwendung wichtig ist). Besonders bei Aufzählungen von gleichartigen Ereignissen, z.B. Zugängen von Objekten, wird in den Texten auf die Nennung von bestimmten Begriffen verzichtet, da die semantische Bedeutung des Ereignisses schon impliziert ist.³ Das Problem bei der textnahen Annotation ist es nun diese impliziten Begriffe trotzdem zu modellieren, sodass das einzelne Ereignis bei einer Abfrage seine Bedeutung auch ohne seinen

³z.B. „Die Sammlung erhielt Geschenke durch Dr. Reinhardt aus dem Altai-Gebiet, Oberleutnant Friedrich aus Deutsch-Südwestafrika, Prof. Dr. Matschie aus China und Herzberg von der zoologischen Station Neapel.“ Nach der Ontologie von CIDOC-CRM ist dabei jedes Geschenk ein unabhängiges Ereignis, das korrekt so zu modellieren ist: E21 Person ← [P23 transferred title from] ← E8 Acquisition → [P24 transferred title of] → E19 Physical Object/ E20 Biological Object → [P53 has former location] → E53 Place . Das Problem hierbei ist, dass „Geschenk“ als E8 Acquisition nur einmal und E19 Physical Object/ E20 Biological Object gar nicht explizit für alle vier Ereignisse genannt wird, dies aber implizit wird.

textuellen Kontext nicht verloren. Ich musste dadurch teilweise mehr semantische Entitäten in einer Textpassage konstruieren als darin Wörter vorkommen. Dafür habe ich das Feature der „virtuellen“ Entität angelegt, deren Inhalt im Postprocessing zu semantischen Instanzen ohne textuellen Inhalt aufgelöst wird. Der Annotationsprozess lief über die gesamten restlichen etwa sieben Wochen der Praktikumsphase und zieht sich aktuell immer noch hin, parallel habe ich aber auch immer mal wieder an den folgenden Teilaufgaben Postprocessing und Präsentation der Annotationen gearbeitet.

4.4 Postprocessing & Repräsentation

Um in den wöchentlichen Sitzungen einen Zwischenstand des Annotationsprozesses geben zu können und auch in Vorbereitung auf das finale Produkt des Projektes, die Webseite, habe ich während dem händischen Annotieren an Skripten gearbeitet, die die bereits fertiggestellten Annotationen konsolidiert, rudimentär analysiert und visuell aufbereitet haben. Dazu exportierte ich regelmäßig die annotierten Texte aus INCEpTION als UIMA-XMI-Dateien [2], lies die Texte durch die Annotationen aus dem OCR-Corrections-Layer mit Python bereinigen und parste die Informationen aus den beiden anderen Layer als Python-Objekte. Dadurch war es mir durch das Programmieren von eigenen Methoden möglich, die Daten in verschiedenen anderen Formaten zu repräsentieren, z.B. als GRAPHML zur Darstellung des gesamten semantischen Netzes oder als JSON für den Datenverkehr der noch zu entwerfenden Webseite. Im gleichen Parsing-Schritt generierte ich die oben angesprochenen „virtuellen“ impliziten semantischen Entitäten sowie die zusätzlichen Verbindungen im Type- und Postprocessing-Feature des SemanticEntities-Layer und konsolidierte dort wo es sinnvoll ist mehrere semantische Instanzen zu einer einzigartigen Instanz, z.B. Instanzen von `E28 Conceptual Object`, die denselben taxonomischen Begriff darstellen.⁴ Neben einfachen Übersichten in Tabellenform, die die Häufigkeit von verwendeten semantischen Klassen und Eigenschaften anzeigen, habe ich für die Präsentation der Annotationen ein Skript geschrieben, dass mithilfe einer Übersetzung der Daten in die DOT-Sprache und anschließendem Rendern durch Graphviz die Nachbarschaft einer semantischen Entität durch einen Graph darstellt [7]. Die Nachbarschaft, die ich durch eine Breitensuche zusammenstellte, bildet sich dabei durch die Entitäten, die mit der gewünschten Entität bis zu einer standardmäßigen Tiefe von vier verbunden sind. Insgesamt habe ich an dieser Aufgabe nicht in einem Block gearbeitet, sondern parallel zur Annotation der Texte, deshalb ist eine zeitliche Aufwandsschätzung schwierig. Ich denke aber, dass ich etwa eine bis zwei Arbeitswochen dafür investiert habe.

Aufgrund des unterschätzten Aufwandes des händischen Annotierens war es mir zeitlich nicht mehr möglich, mich mit weitergehenden Analysen des Datensatzes durch Methoden aus den DH zu beschäftigen.

5 Ergebnisse

Das angestrebte fertige Produkt in Form der Webseite zum systematischen Abfragen der Informationen und die vollständige semantische Struktur der Jahresberichte kann ich wohl frühes-

⁴erkennbar durch `E28 Conceptual Object` → `[P2 has type]` → `E55 Type` :„Taxon“. Darüber hinaus habe ich Instanzen der Klassen `E55 Type`, `E78 Curated Holding`, `E21 Person`, `E53 Place` konsolidiert, wenn die annotierten Zeichenketten dieselben waren.

tens in drei Monaten präsentieren, aber zumindest auf die Zwischenergebnisse der einzelnen Teilaufgaben werde ich im Folgenden eingehen.

Das Ziel, dass ich bei der Vorbereitung des Korpus verfolgte, war es für die Annotation optimale Texteinheiten zu erstellen. Nachdem ich die einzelnen Seiten der Jahresberichte von der HU-Webseite gecrawlt und die Texte normalisiert & korrigiert hatte, stellte ich die Texte so zusammen, dass ich pro Jahr vier rohe Textdateien bekam (jeweils eine pro Sammlung und eine für die Generalverwaltung), damit ich auch über Seitengrenzen semantische Beziehungen einfach annotieren kann. Die originalen Seitenumbrüche markierte ich im Textfluss, sodass eine Identifikation der Seitennummern möglich bleibt.

Zugänge: Objekte	17	28	33	36	49	10	11	7	35
Publikationen	2	23	22	24	18	10	1	37	20
Präparationsarbeit/Sammlungsorganisation	45	12	19	13	11	9	4	1	14
Öffentlichkeit/Publikum	12	22	9	11	9	10	2	12	11
Lehre	1	4	5	5	5	15	40	18	6
Personalia	5	8	5	4	3	33	23	13	6
Zugänge: Bibliothek	3	2	6	6	2	7	16	4	4
Gebäude	10	1	1	1	2	5	0	5	2
Zugänge: Instrumente	4	1	1	2	1	1	5	3	2
	1889	1894	1899	1904	1909	1914	1928	1938	Gesamt

Abbildung 1: Übersicht über die angesprochenen Themen in einer Stichprobe aus den Jahresberichten, die Zahl ist der Anteil eines Themas an allen Zeichen im jeweiligen Jahresbericht in Prozent.

Auch bei der Vorbereitung des auf CIDOC-CRM-basierenden Annotations-Schemas war es die Intention, einen möglichst effizienten Annotationsprozess zu gewährleisten. Meine Idee war es die ursprünglich 81 Klassen und 160 Eigenschaften der Ontologie soweit zu reduzieren, dass ich damit die in den Jahresberichten angesprochenen Themen ausreichend modellieren kann, ohne dass ich während dem Annotieren die meiste Zeit die Dokumentation von CIDOC-CRM nach dem richtigen Muster durchforsten muss. Als Zwischenschritt habe ich dementsprechend die Themen in einer Stichprobe der Berichte grob annotiert, das Ergebnis ist in Abbildung 1 visualisiert. Bei der anschließenden Auswahl der benötigten semantischen Klassen und Eigenschaften auf Basis dieser Themen konnte ich die viel zu umfassende Ontologie auf 28 Klassen und 32 Eigenschaften beschränken.

Anfang Dezember, am Ende dieser ersten beiden Aufgaben, stellte ich meinen Ansatz im institutsweiten Kolloquium vor, bei dem wir in der Diskussion vor allem die Möglichkeiten erörtert haben, die Daten zukünftig in externe Repositorien wie Wikidata oder der Biodiversity Heritage Library einzubetten.

Bis zum jetzigen Zeitpunkt habe ich in den Jahresberichten des Naturkundemuseums 6603 semantische Entitäten annotiert, die untereinander durch 8534 gelabelte semantische Eigenschaften verbunden sind. Eine Übersicht über die Verteilung nach Klassen aus CIDOC-CRM

gibt Tabelle 4, bzw. Tabelle 5. Als abzusehen war, dass ich die Annotation bis zum Ende der Projektlaufzeit nicht vollkommen fertigstellen kann, haben wir uns darauf geeinigt, dass ich mich zunächst auf Berichte der Zugänge von neuen Objekten, insbesondere von Säugetieren, fokussieren soll. Vollständig annotiert sind die Jahresberichte deshalb nur bis 1892, 1892–1896 sind die Passagen zu allen Objekt-Zugängen annotiert und die Passagen zu Objekt-Zugängen der Säugetiersammlung 1896–1915. Die in den Tabellen aufgeführten Zahlen repräsentieren also nicht die semantische Struktur der kompletten Texte, sondern die Struktur in ausgewählten Ausschnitten, die vor allem über Eingänge von Ausstellungsstücken berichten.

Semantische Klasse	Instanzen	Semantische Property	Instanzen
E21 Person	1332	P23 transferred title from	1368
E53 Place	1036	P53 has former or current location	1359
E19 Physical Object	726	P2 has type	1351
E8 Acquisition	638	P24 transferred title of	790
E78 Curated Holding	449	P22 transferred title to	708
E28 Conceptual Object	391	P43 has dimension	563
E20 Biological Object	334	P46 is composed of	540
E60 Number	288	P4 has time-span	362
E54 Dimension	279	P130 shows features of	350
E55 Type	164	P128 carries	205
E35 Title	157	P94 has created	176
E74 Group	134	P1 is identified by	110
E96 Purchase	112	P147 curated	99
E41 Appellation	103	P25 moved	82
E3 Condition State	79	P44 has condition	81
E87 Curation Activity	78	P16 used specific object	38
E52 Time-Span	65	P26 moved to	31
E73 Information Object	54	P2 has Type	22
E9 Move	36	P109 has current or former curator	21
E7 Activity	28	P143 joined	20
E85 Joining	17	P108 has produced	19
E12 Production	14	P51 has former or current owner	14
E86 Leaving	13	P67 refers to	14
E11 Modification	13	P31 has modified	14
E29 Design or Procedure	12	P92 brought into existence	13
E14 Condition Assessment	10	P145 separated	12
...	<10	P129 is about	12
		P144 joined with	9
		P15 was influenced by	9
		P11 had participant	9
		P107 has current or former member	8
		...	<8
<i>Insgesamt</i>	6603	<i>Insgesamt</i>	8534

Tabelle 4: Übersicht des aktuellen Standes an annotierten semantischen Entitäten nach Klassen.

Tabelle 5: Übersicht des aktuellen Standes an annotierten semantischen Verbindungen nach Klassen.

Durch den letzten Schritt, dem Postprocessing der Annotationen, habe ich einerseits die durch händische Annotation korrigierten Textteile und anderseits die semantische Struktur der Textteile mithilfe meiner semantischen Annotationen zusammengestellt. Die Daten habe ich durch meine Skripte in verschiedene Formate exportiert:

1. als UIMA-XMI pro Jahr und Sammlung, dabei jedoch ohne mein oben genanntes Postprocessing
2. als eine Pickle-Datei für alle annotierten Jahre, die man mithilfe meines Python-Moduls einlesen kann und die für jede Semantische Entität die semantische Klasse, die markierte Zeichenkette, den Beginn und das Ende der Zeichenkette im Text, das Jahr und die Institution (also Sammlung) des Jahresberichts aus dem die Annotation stammt, die Seiten- und Zeilennummer sowie die ihr durch semantische Eigenschaften verbundenen Entitäten speichert
3. als eine JSON-Datei pro Jahr und Sammlung mit den gleichen Informationen wie in der Pickle-Datei
4. als eine GRAPHML sowie RDF-Triples in XML für alle annotierten Jahre, Knoten sind die semantischen Entitäten und Kanten sind die semantischen Eigenschaften.

Zur Analyse der Annotationen bin ich aufgrund der Unterschätzung des Aufwandes für die händische Annotationen noch nicht gekommen. Aber in Vorbereitung dessen – vor allem aber für die noch zu erstellende Webseite, mit der Interessierte die Annotationen abfragen können – habe ich mithilfe meines Skriptes und Graphviz für jede einzelne der 6603 semantischen Entitäten eine SVG-Datei erstellt, die die jeweils ihr durch semantische Properties verbundenen Nachbar-Entitäten bis zu einer Tiefe von vier in Form eines Graphen darstellt (ein Beispiel ist die Abbildung 2). Mein Ziel damit ist es, den Kontext in dem eine Entität auftaucht, die man z.B. über eine Suchmaske auf der Webseite ausfindig gemacht hat, nutzerfreundlich aufbereitet zu präsentieren. Das gesamte semantische Netz ist durch Gephi in Abbildung 3 gezeichnet.

Neighbourhood for Entity No. 18701 in Zoologisches Institut und zoologische Sammlung (1903) with depth 3

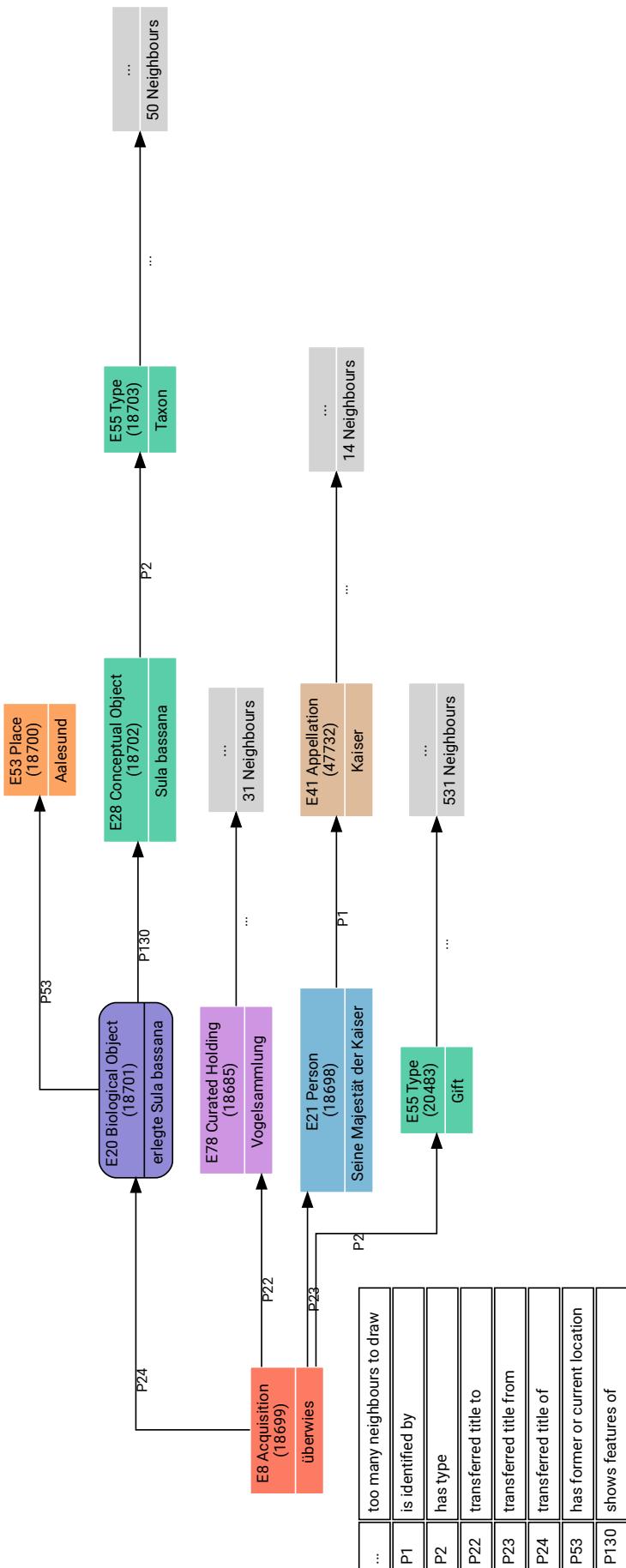


Abbildung 2: Beispiel für eine automatisch generierte Visualisierung der Nachbarschaft der semantischen Entität „erlegte Sula bassana“ mit Tiefe 3 im Jahresbericht von 1903.

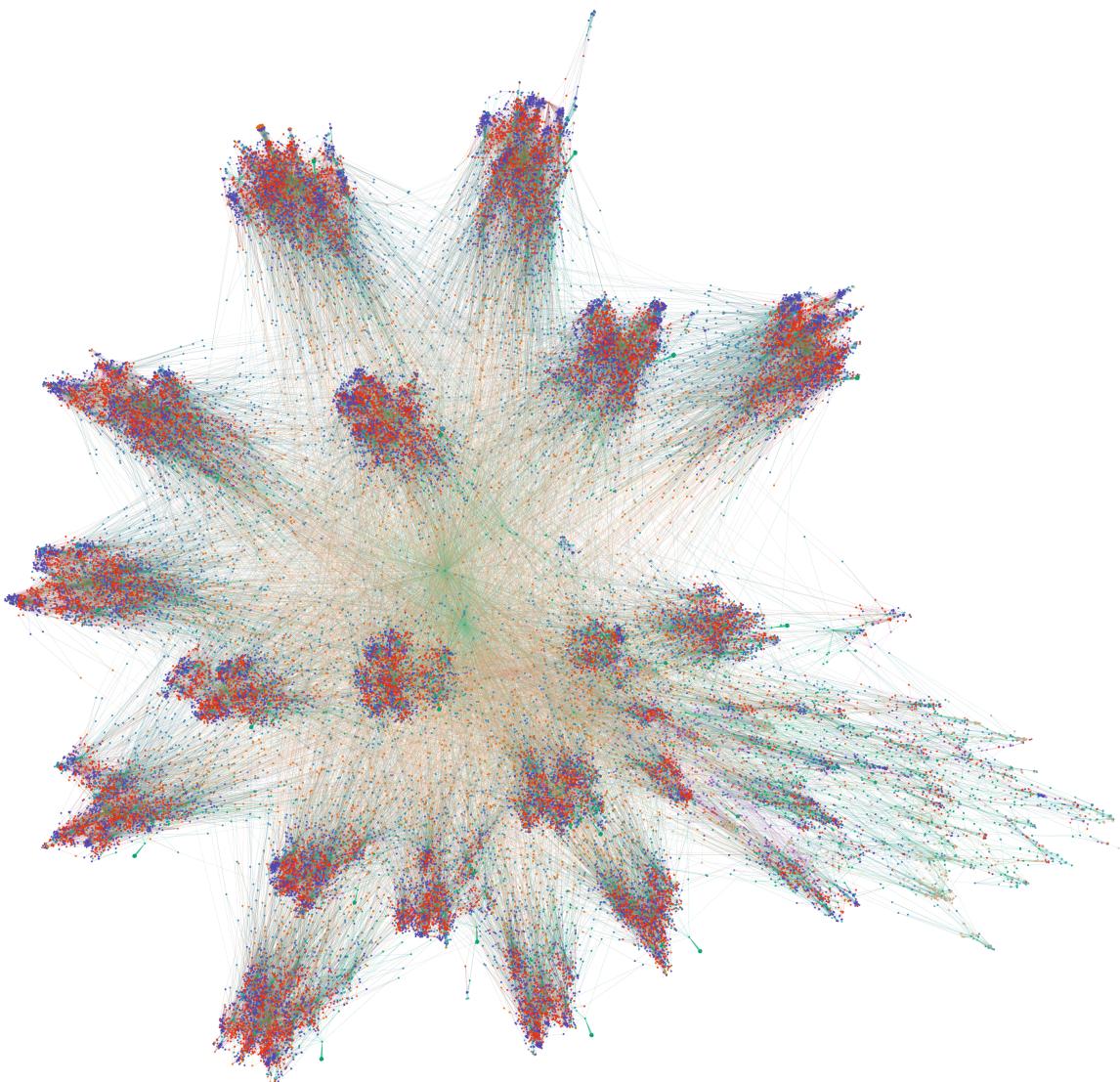


Abbildung 3: Eine durch Gephi gerenderte Darstellung des gesamten semantischen Graphen. Die semantischen Entitäten gruppieren sich darin erkennbar nach Jahren.