Bachelorarbeit

Automatische Relationsextraktion für die semantische Volltextsuche

Christiane Schaffer

10.10.2012

Albert-Ludwigs-Universität Freiburg im Breisgau
Technische Fakultät
Institut für Informatik

Bearbeitungszeitraum
10. 07. 2012 – 10.10. 2012

Gutachter
Prof. Dr. Hannah Bast

Betreuer
Prof. Dr. Hannah Bast
Björn Buchhold
Florian Bäurle

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelorarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Bachelorarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Ort, Datum	Unterschrift

Abstract

Gegegeben seien Daten einer Ontologie. Das Ziel ist es, eine bestehende geografische Relation um zusätzliche Fakten zu erweitern. Dazu soll eine umfassendere, auf die Relation spezialisierte Datenbank verwendet werden. Die Herausforderung dabei liegt in der nicht vereinheitlichten Namensgebung von Ontologie und Datenbank. Da letztere weitaus mehr Fakten speichert als die bestehende Ontologie, müssen Lösungen für auftretende Mehrdeutigkeiten gefunden werden. Es werden verschiedene Möglichkeiten beschrieben, Zuordnungen für die Relation zu finden, sowie eine Analyse dieser vorgenommen.

Inhaltsverzeichnis

1 Einleitung

1993 beschrieb Gruber den Begriff der Ontologie als „explicit specification of a conceptualization“ [Gru93]. Vereinfacht gesagt, stellt eine Ontologie die in ihr gespeicherten Begrifflichkeiten formal strukturiert dar und wird deshalb z. B im Bereich des *Semantic Web*, in der Medizin und in der Geografie für die Organization von Wissen genutzt.

1.1 Problemdefinition

Gegeben seien Daten einer Ontologie, die als Subjekt-Prädikat-Objekt-Tripel gespeichert sind. Subjekt und Objekt sind Instanzen, die einer oder mehrerer Klassen untergeordnet sind. Diese wiederum gehören einer sogenannten Entität an. Subjekt und Objekt werden jeweils durch eine Relation in Beziehung gesetzt.

Für die in dieser Arbeit verwendete Ontologie (siehe Kapitel 2.1) soll die bereits vorhandene Relation `located-in` vorrangig mit Instanzen der Klassen `City` und `Country` erweitert werden.

Die Ontologie besitzt beispielsweise die Instanzen `Munich` und `Ingolstadt` in der Klasse `City`. Für `Munich` ist keine Relation der Form `located-in` vorhanden.
`Ingolstadt` wird dagegen als Stadt in `Bavaria`, einer `Location` gespeichert, was jedoch nicht mit `Germany` in Verbindung steht.

Das Finden weiterer `located-in`-Relationen soll unter Zuhilfename einer geografischen Datenbank und unabhängig von bereits in der Ontologie gespeicherten Beziehungen erfolgen. Ein Problem dabei ist die Zuordnung der in der Datenbank enthaltenen Namen zu den Entitäten der Ontologie. So tragen verschiedene Städte oder Regionen den gleichen Namen. Als Beispiel soll die Entität `Berlin` dienen, von der bekannt ist, dass sie der Klasse `City` angehört. Es stellt sich nun die Frage, auf welchen Ort sie sich genau bezieht. Berlin bezeichnet z. B.:

- die Hauptstadt von Deutschland
- ein deutsches Bundesland
- eine Stadt im US-Bundesstaat New York
- eine heutige Geisterstadt im US-Bundesstaat Nevada
- eine Stadt in Südafrika

Da sich die Entitäten global nur auf ein Objekt beziehen, tragen die Namen bei Mehrdeutigkeiten einen Zusatz.
So existieren z. B. `Berlin,_Nevada` und `Berlin,_New_York` in der Ontologie. Die Namensgebung folgt allerdings nur ansatzweise bestimmten Regeln: Der Ländername kann in Klammern oder durch Komma getrennt angegeben sein (`Mede_(Italy)` und `Genga,_Italy`). Für die italienische Stadt `Zocca` fehlt allerdings ein Zusatz. Zudem kann sich der Zusatz auch auf eine dem Land untergeordnete Region beziehen (`Gangi,_Sicily` und `Bema_(SO)`). Weitere Angaben und Kombinationen sind möglich.

Trotz der Zusätze, die eine Charakterisierung der Entitäten ermöglichen, kann es gerade bei nicht näher beschriebenen Namen – auch nach dem Ausschlussprinzip – zu mehreren Übereinstimmungen in der geografische Datenbank kommen, da diese umfangreicher als die verwendete Ontologie ist. Die Stadt Berlin in Südafrika ist beispielsweise nicht gelistet. Es müssen daher weitere Kriterien für eine korrekte Zuordnung gefunden werden.

1.2 Überblick

Zunächst wird kurz auf die in Zusammenhang stehenden Arbeiten und verwendeten Datenquellen eingegangen: die Suchmaschine Broccoli, zu deren Erweiterung beigetragen werden soll, YAGO, die aktuelle Ontologie hinter Broccoli und GeoNames als die verwendete geografische Datenbank.

Der Hauptteil zeigt zu Beginn, wie relevantes Wissen aus den verwendeten Datenquellen extrahiert wurde und stellt dann Ansätze zum Zusammenführen beider dar.

Kapitel 4 macht den Versuch einer ersten Evaluation der erarbeiteten Ansätze und zeigt dabei auch Herausforderungen auf.

Zum Schluss soll ein Ausblick für die Weiterführung der Arbeit gegeben werden. Verbesserungsvorschläge und Anregungen zur Problemlösung werden aufgezeigt.

2 Verwandte Arbeiten

2.1 Broccoli

Broccoli ist eine sich an der hießigen Universität in Entwicklung befindende Suchmaschine für die sogenannte semantische Volltextsuche [BBBH12]. Diese kombiniert die klassische Volltextsuche mit der Ontologie-Suche.

Bei der Volltextsuche wird nach Eingabe von einem oder mehreren Schlüsselwörtern eine Liste von Dokumenten ausgegeben, die diese ganz oder teilweise enthalten. Unter Ontologie-Suche dagegen wird im Zusammenhang mit Broccoli die strukturierte Suche in einer Wissensdatenbank verstanden. Sie speichert Tripel aus zwei Entitäten, die jeweils durch eine Relation verbunden sind. Diese Informationen bilden einen gerichteten Graphen, wobei die Entitäten die Knoten darstellen und die Relationen die verbindenen Kanten symbolisieren. Die derzeitige Ontologie hinter Broccoli ist YAGO (vgl. Abschnitt 2.2).

Für eine Anfrage in der semantischen Volltextsuche müssen die Suchwörter in der Ontolologie und im Text erfasst und das Wissen daraus entsprechend dem Sinnzusammenhang der Anfrage geeignet kombiniert werden. Teilprobleme sind dabei die Erkennung der Entitäten im Text und die Zerlegung der Sätze in sogenannte *contexts*. Für genauere Ausführungen sei an dieser Stelle auf [Hau11] verwiesen. Die Textdateien für Broccoli liefert die englische Wikipedia.

Die Benutzeroberfläche von Broccoli soll bei der Erstellung eines Suchbaumes helfen, welcher sich aus den folgenden vier Objekten zusammensetzen kann:

- normale Wörter (*ordinary words*) – gelb
- Klassen (*classes*) – rot
- Instanzen (*instances*) – blau
- Relationen (*relations*) – grün

Abbildung 2.1 zeigt eine Beispielsuchanfrage in Broccoli. Gesucht wurde dabei nach Städten in Portugal, die mit Erdbeben in Zusammenhang stehen.

Bezeichnet die Kante des Baumes eine Relation aus der Ontologie, so wird sie als *ontology arc* bezeichnet (hier `located-in`). Für die Suche im Volltext gibt es die *occurs-with arcs*, an deren Zielknoten eine beliebige Menge von Wörtern, Entitäten oder sogar neue Bäume stehen kann.

Broccoli liefert zur Suchanfrage außerdem die Ausschnitte aus der Ontologie und dem Volltext, aus der die Informationen extrahiert wurden. Dies bietet dem Nutzer eine größere Transparenz und die Ergebnisse können besser nachvollzogen werden.

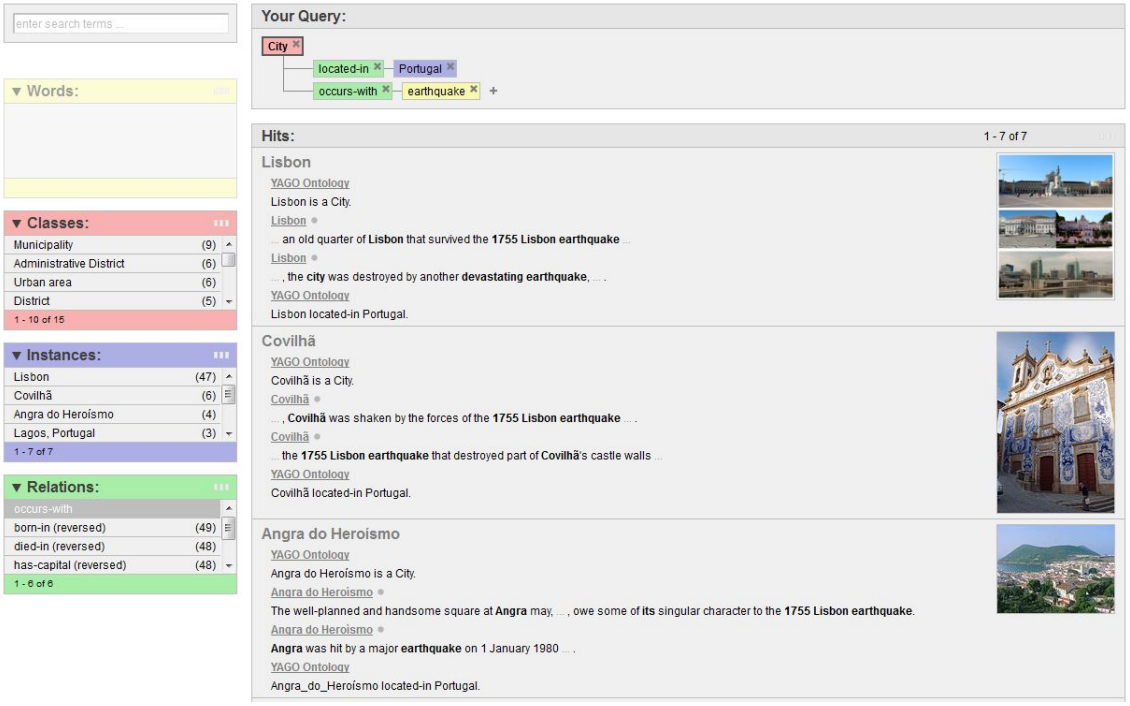

Abbildung 2.1: Screenshot einer Beispielsuchanfrage in Broccoli. Es wurde nach Städten in Portugal gesucht, die im Zusammenhang mit Erdbeben stehen.

2.2 YAGO

YAGO[1] ist eine umfassende Wissensdatenbank, die Fakten autotmatisch aus der englischen Wikipedia extrahiert und mit der Taxonomie von WordNet[2] verlinkt [SKW07, SKW08]. YAGO2 heißt die Erweiterung von YAGO, welche die Fakten zusätzlich in Zeit und Raum verankert [HSBW12].

Jeder Artikel aus Wikipedia wird zu einer Entität. Um Instanzen für die manuell definierten Relationen zu finden, werden die Kategorie-Seiten und Infoboxen ausgewertet. Eine manuelle Evaluation von YAGO ergab eine Genauigkeit von 95%.

[1] Yet Another Great Ontology
[2] Lexikalische Datenbank auf Englisch, `http://wordnet.princeton.edu`

2.3 GeoNames

GeoNames[3] ist eine geografische Datenbank, die über 8 Millionen Features beinhaltet. Jedem Feature ist eine von neun Klassen zugeteilt, denen wiederum insgesamt 645 sogenannte Feature-Codes untergeordnet sind. Damit erfolgt die Charakterisierung der Ortsnamen als z. B. Länder, Städte, Gewässer oder Berge. Neben den Namen der Features in verschiedenen Sprachen sind u. a. auch Angaben zur Erhebung, Bevölkerungszahl und Koordinaten integriert.

Die Daten stammen aus über 100 verschiedenen Quellen, wie z. B. der *National Geospatial-Intelligence Agency*[4] und dem *Geographic Names Information System*[5] und sind über eine Reihe von Webdiensten zugänglich, können aber auch kostenlos heruntergeladen werden.

[3] `www.geonames.org`
[4] `http://earth-info.nga.mil/gns/html/index.html`
[5] `http://nhd.usgs.gov/gnis.html`

3 Relationsextraktion

3.1 Vorverarbeitung der Eingabedaten

3.1.1 Broccoli

Die verwendeten Ontologie-Daten von Broccoli befinden sich in einer Datei namens *ontology.txt*. Jede der knapp 25 Mio. Zeilen speichert ein Subjekt-Prädikat-Objekt-Tripel nach einem festen Schema.

Um die Ortsnamen aus dieser Datei zu filtern, müssen alle Instanzen gefunden werden, die der Klasse `Location`, `Country`, `Region` oder `City` angehören. Folgend sind beispielhaft Zeilen aus der *ontology.txt* aufgelistet, die alle möglichen Position für das Autreten der relevanten Klassen abdecken. Die zu erfassenden Instanzen sind rot hinterlegt, die Klassen sind blau markiert.

relation	*entity1*	*entity2*	*entity3*	*entity4*
:r:is-a	:e:entity:Entity	:e:class:Class	:e:berlin:Berlin	:e:city:City
:r:is-a	:e:entity:Entity	:e:class:Class	:e:germany:Germany	:e:country:Country
:r:is-a	:e:entity:Entity	:e:class:Class	:e:germany:Germany	:e:location:Location
:r:deals-with	:e:country:Country	:e:country:Country	:e:france:France	:e:germany:Germany
:r:located-in	:e:location:Location	:e:location:Location	:e:berlin:Berlin	:e:germany:Germany
:r:is-citizen-of	:e:person:Person	:e:country:Country	:e:karllagerfeld:Karl_Lagerfeld	:e:Germany
:r:has-official-language	:e:country:Country	:e:language:Language	:e:germany:Germany	:e:germanlanguage:German_language

Anhand dieser Beispiele kann der Ablauf des verwendeten Algorithmus zur Extraktion der Ortsnamen nachvollzogen werden (siehe Algorithmus 1). Am Ende wird eine Liste mit allen gefundenen Entitäten ausgegeben, die jede Instanz aber nur einmal enthält. Zusätzlich werden jeweils die Gesamtanzahl der Vorkommen in der *ontology.txt* erfasst und gespeichert. Dies soll später das Auffinden des korrekten Ländernamens vereinfachen (vgl. Kapitel 3.2).

Unter den gefundenen Instanzen befinden sich auch Namen mit numerischen Zeichen, wie z. B. `Denmark_in_the_Eurovision_Song_Contest_2010` und `H2Oasis_Indoor_Waterpark`. Diese wurden anschließend herausgefiltert, da davon ausgegangen werden kann, dass die für diese Arbeit relevanten Ortsnamen keine Zahlen enthalten.

Tabelle 3.1 stellt die Anzahl der gefundenen Entitäten mit und ohne Zahlen für verschiedene Klassen gegenüber. Da die Klassen `Country`, `Region` und `City` der

Algorithmus 1 Filtern der Ontology nach Klasse *pattern*

```
1: for all Zeilen in Ontology do
2:     if entity2 = class und entity4 = pattern then
3:         list(entity3)
4:     else
5:         if entity1 = pattern then
6:             list(entity3)
7:         end if
8:         if entity2 = pattern then
9:             list(entity4)
10:        end if
11:    end if
12: end for
```

Klasse `Location` angehören, sollte es keinen Unterschied machen, ob nur nach letzterer oder den gesamten relevanten Klassen gefiltert wird. Unterschiede treten aber zwangsweise in der Anzahl der Vorkommen der einzelnen Entitäten auf.
Die fehlende Instanz, die nur in `Country`, aber nicht in `Location` vorkommt, ist `Jamaica`.

Weiterhin fällt auf, dass die Anzahl der als `Country` gelisteten Entitäten weit die heute existierenden 250 Länder übersteigt. Dies hat verschiedene Gründe. Zum einen beinhaltet die Liste auch Regionen und Verwaltungseinheiten, wie z. B. `Baden-Württemberg` oder das nicht mehr existierende `West_Francia` und vereinzelt auch Städte (`Aachen`). Zum anderen sind aber auch Einträge zu finden, die selbst im weitesten Sinne nicht als Land bezeichnet werden können, wie `Culture_of_Canada` und `USWA_Women's_Championship`. Umgekehrt fehlen Länder, wie z.B. `Trinidad_and_Tobago` oder US-Bundesstaaten, wie `Florida` und `Arizona`, in dieser Klasse bzw. in der gesamten Ontologie. Die Liste kann also nicht uneingeschränkt als Quelle für Länder und aministrative Einheiten genutzt werden.

Tabelle 3.1: Anzahl der gefundenen Entitäten aus *ontology.txt* abhängig von den jeweiligen Klassen, nach denen gefiltert wurde.

Klasse	Anzahl der Entitäten	
	mit Zahlen	ohne Zahlen
`Country`	5.693	4.936
`City`	38.685	38.673
`Location`	414.035	408.262
`Location, Country, Region, City`	414.036	408.263

3.1.2 GeoNames

Aus GeoNames wurden die in Tabelle 3.2 gelisteten Textdateien zur Datenextraktion verwendet.

Alle Dateien sind UTF-8-kodiert und enthalten ein geografisches Feature pro Zeile. Die jeweils dazugehörigen Informationen werden in Spalten (durch Tabulatoren getrennt) gelistet. Siehe dazu Tabelle 3.3.

Tabelle 3.2: Verwendete Dateien aus GeoNames.

Name	Inhalt	Anzahl
allCountries.txt	alle geografischen Features	8.314.274
cities15000.txt	alle Städte mit über 15000 Einwohnern oder Hauptstädte	22.929
cities1000.txt	alle Städte mit über 1000 Einwohnern oder Sitze administrativer Einheiten	128.127

Tabelle 3.3: Spalten der GeoNames-Dateien.

Spalte	Beschreibung	Datentyp / Format
1	ID des Eintrages in der GeoNames-Datenbank	int
2	Name des geografischen Ortes, UTF 8-codiert	varchar(200)
3	Name des geografischen Ortes, ASCII-codiert	varchar(200)
4	alternative Namen, durch Komma getrennt	varchar(5000)
5	Breitengrad	Dezimalgrad (WGS 84[a])
6	Längengrad	Dezimalgrad (WGS 84)
7	Feature-Klasse	char(1)
8	Feature-Code	varchar(10)
9	Ländercode nach ISO-3166-1	char(2)
10	alternative Ländercodes, Komma getrennt	char(60)
11	Code für die erste Verwaltungseinheit	varchar(20)
12	Code für die zweite Verwaltungseinheit	varchar(80)
13	Code für die dritte Verwaltungseinheit	varchar(20)
14	Code für die vierte Verwaltungseinheit	varchar(20)
15	Einwohnerzahl	8 byte int
17	Erhebung in Metern	int
18	Digitales Höhenmodell	int
19	ID für die Zeitzone	varchar(40)
20	Datum der letzten Änderung	yyyy-mm-dd

[a] World Geodetic System 1984

Aus der Datei *allCountries.txt* wurden zunächste alle Zeilen herausgefiltert, die eine relevante Feature-Klasse oder einen bestimmten Feature-Code[1] aufweisen. Tabelle 3.4 listet diese auf. Anschließend wurden für diese Zeilen jeweils die für wichtig empfundenen Spalten extrahiert.

Tabelle 3.4: Anzahl der aus *allcountries.txt* extrahierten Zeilen für die benötigten Feature-Klassen oder Feature-Codes, welche an der angegebenen Spaltenposition zu finden sind. Weiterhin sind die anschließend benötigten Spalten gelistet (siehe dazu Tabelle 3.3).

Feature-Klasse/Code		Spalte	Anzahl Zeilen	extrahierte Spalten
Länder	PCL, PCLD, PCLF, PCLI, PCLIX, PCLS, TERR	8	253	2, 4, 9
Verwaltungs-einheiten	ADM1	8	3.830	2, 4, 9, 11
	ADM1, ADM2		36.123	
Orte	P	7	3.124.046	1, 2, 4, 9, 11, 15

Länder Länder sind unter der Feature-Klasse A (Länder, Staaten, Regionen etc.) zu finden. Die extrahierten Feature-Codes umfassen

- Staaten (*political entity*, PCL),
  z.B. *Isle of Man*
- abhängige Staaten (*dependent political entity*, PCLD),
  z.B. *Guadeloupe*
- assozierte Staaten (*freely associated state*, PCLF),
  z.B. *Federated States of Micronesia*
- unabhängige Staaten (*independent political entity*, PCLI),
  z.B. *Republic of Poland*
- Sektionen unabhängiger Staaten (*section of independent political entity*, PCLIX),
  z.B. *Saint-Martin*
- teilweise unabhängige Staaten (*semi-independent political entity*, PCLS),
  z.B. *Hong Kong Special Administrative Region*
- Territorien (*territory*, TERR),
  z.B. *Antarctica*

[1] siehe dazu auch `www.geonames.org/export/codes.html`

Gefunden wurden 253 anstatt der 250 erwarteten Länder, da unter AU nicht nur *Commonwealth of Australia*, sondern auch die drei dazugehörigen Territorien *Territory of Ashmore and Cartier Islands*, *Coral Sea Islands Territory* und *Jervis Bay Territory* gelistet sind, die später, falls nötig, ignoriert werden können.

Verwaltungseinheiten Ebenfalls in der Feature-Klasse A sind die Verwaltungseinheiten zu finden. ADM1 bis ADM4 bezeichen die einem Land untergeordneten Verwaltungseinheiten erster bis vierter Ordnung. Weiteren Einheiten wurde der Feature-Code ADMD zugewiesen.

Folgend wurden nur Features mit den Codes ADM1 und ADM2 einbezogen. Zu ADM1 gehören u. a. alle US-Bundesstaaten sowie die Bundesländer Deutschlands. Ein Beispiel für ein Feature mit dem Code ADM2 ist der *Regierungsbezirk Freiburg*.

Orte Extrahiert man alle Zeilen mit der Feature-Klasse P (Städte, Dörfer etc.) aus der Datei *allCountries.txt*, so erhält man auch eine Vielzahl von Orten mit einer sehr geringen Einwohnerzahl. Daher sollten weitere Schritte zur Extraktion relevanter Orte vorgenommmen werden. Da GeoNames auch Dateien mit Städten für eine bestimmte Mindesteinwohnerzahl zur Verfügung stellt (siehe Tabelle 3.2), wurden diese im Rahmen der vorliegenden Arbeit verwendet. Generell sollte aber über die Einbeziehung auch von Orten mit wenigen Einwohnern nachgedacht werden, da sonst verlassene, aber z. B. historisch signifikante Orte wie *Chernobyl* verloren gehen (besitz in GeoNames den Feature-Code für *abandoned populated place*, PPLQ und eine Einwohnerzahl von 15).

3.2 Vereinheitlichung der Ländernamen

Da für diese Arbeit Relationen der Form `City located-in Country` gefunden werden sollen, wurden die Ländernamen separat vereinheitlicht. Aus GeoNames wurden dazu alle Ländernamen (regulärer Name und alternative Namen) für die jeweiligen Ländercodes in einer HashMap gespeichert (vgl. Tabelle 3.4)

Für jede relevante Entität der Ontologie (vorrangig Instanzen der Klasse `Country`) wurde anschließend durch alle Namen der Map iteriert, um direkte Übereinstimmungen zu extrahieren. Dabei wurde auch die zu jeder Entität erfasste Anzahl der Vorkommen mit gespeichert.

Eine gesonderte Behandlung wurde vorgenommen, wenn die Instanz den Zusatz `(country)` beinhaltete. Das beste und momentan einzige Beispiel hierfür ist `Georgia_(country)`. Eine Abgrenzung zum US-Bundesstaat des gleichen Namens ist zwar nicht unbedingt notwendig, da auch dieser einen charakteristischen Zusatz in der Ontologie trägt, jedoch muss zumindest das Suchmuster abgeändert werden, da Ergänzungen dieser Art für GeoNames nicht üblich sind. Gesucht wurde demnach

nach `Georgia`. Gespeichert wurde aber der Entitätsname aus der Ontologie, welcher zudem mit einem maximalen Score versehen wurde, da dieser eindeutig als Land klassifiziert werden kann.

Nach diesen Schritten erhält man eine Map folgender Form:

```
DE  {Gyaaman=1, Germany=963, Federal_Republic_of_Germany=1}
GE  {Georgian_Soviet_Socialist_Republic=1, Georgia_(country)=4936}
HU  {Hungary=300}
ZM  {Zambia=4, Northern_Rhodesia=1}
```

Bei mehreren gefundenen Entitäten wurde anschließend diejenige mit den meisten Vorkommen gewählt, da davon ausgegangen wird, dass offizielle und aktuelle bzw. regulär verwendete Ländernamen viel häufiger in der Ontologie verwendet werden. Vor allem Relationen wie `deals-with` und `citizen-of` nutzen wiederkehrend Instanzen der Klasse `Country`. Wiesen die Entitäten die gleiche Anzahl an Vorkommen auf, wurde die erste gespeicherte Entität gewählt. Wurden für einen Ländercode keine direkten Übereinstimmungen gefunden, so wurde ihm der reguläre Name aus GeoNames zugewiesen.

Zur Auswertung wurden vergleichend die Ontolgie-Daten für die Klassen `Country`, `Location` und einer Kombination beider Klassen genutzt (siehe Kapitel 4.1).

3.3 Relationsfindung

3.3.1 Zuordnung mit Hilfe von Namenszusätzen

Da sich jede geografische Entität in der Ontologie nur auf einen Ort beziehen kann, muss bei Mehrdeutigkeiten eine Möglichkeit der Differenzierung gefunden werden. Wie bereits in den einleitenden Worten erwähnt, wird dies bei der verwendeten Ontologie durch Namenszusätze realisiert.

Folgend soll eine Möglichkeit vorgestellt werden, Zusätze aus den Entitätsnamen zu extrahieren und zu analysieren, um die Entität mit deren Hilfe direkt einem Land zuordnen zu können. Der „Umweg“ über den Städtenamen wird also vermieden. So können beispielsweise die Entitäten `Erp_(Germany)`, `Erp_(Netherlands)` und `Erp,_Ariège` sofort den korrekten Ländern Deutschland, Niederlande und dem französischen Département Ariège zugewiesen werden. GeoNames liefert hierbei die Daten sowohl für die Verwaltungseinheiten, als auch für die Länder.

Mögliche Erweiterungen sind beispielsweise:

1	`Georgia_(U.S._state)`	US-Bundesstaat
2	`Aachen_(district)`	Landkreis in Deutschland
3	`Taki_(India)`	Stadt in `India`
4	`Klinkenberg_(Gelderland)`	Dorf in `Gelderland` (NL)
5	`Bharatpur,_India`	Stadt in `India`
6	`Atlanta,_Texas`	Stadt in `Texas` (US-Bundesstaat)
7	`Lakin,_Burma`	Dorf in `Burma` (MM)
8	`Kyontawa,_Ayeyarwady,_Myanmar`	Ort in `Burma` (MM)
9	`Richmond_County,_Georgia`	County in `Georgia_(U.S._state)`

Bei der Extraktion wird nach Erweiterungen in Form von Klammerausdrücken oder durch Komma abgetrennten Teile gesucht.

Die Zusätze in Zeilen 1 und 2 entsprechen keinem genauen geografischen Feature und werden daher nicht durch den Algorithmus gefunden.

Zeilen 3 und 4 enthalten den Ländernamen bzw. den Namen der Verwaltungseinheit in Klammern.

Zeilen 5 bis 9 zeigen, dass diese auch durch Komma abgetrennt angegeben sein können.

Die Entität in Zeile 8 besteht sogar aus drei Teilen. Der Algorithmus muss aber auch hier nur den hintersten Teil auswerten.

Zeile 9 bezeichnet keine Stadt, sondern einen administrativen Bereich, würde aber unter Umständen (wenn sich die Entität z. B fälschlicherweise in der Klasse `City` befindet oder die Klasse `Location` für die Suche verwendet wird) auch der Relation `City located-in Country` zugeordnet werden, da der vorderste Teil nicht durch den Algorithmus ausgewertet wird.

Die Zusätze werden sowohl in den als Verwaltungseinheiten gespeicherten GeoNames-Features, als auch in den Ländernamen gesucht. Bei letzterem wird dabei erneut durch die Liste aller Namen (reguläre und alternative Namen) und nicht nur durch die bereits den Ländercodes zugeordneten Namen iteriert. Damit wird sichergestellt, das z. B auch die Entität aus Zeile 8 gefunden wird, die als Zusatz einen alternativen, aber relevanten Ländernamen aufweist.

Algorithmus 2 stellt grob den Ablauf zur Extraktion aller Zusätze (*suffix*) und zum Abgleich mit relevanten Namen aus GeoNames dar. Am Ende steht eine Map, die jedem Zusatz gefundene Codes (Ländercode und/oder Code der Verwaltungseinheit) zuteilt. Der Ausschluss von Zusätzen aus zwei Großbuchstaben (siehe Algorithmus 2, Zeile 4) wird als notwendig erachtet, da italiensche Gemeinden oft Zusätze dieser Form tragen, die sich aber gleichzeitig auf einen US-Bundesstaat beziehen können und in GeoNames meist auch nur unter diesem gefunden werden. Beispiele sind

hier `Capranica_(VT)` und `Siliqua_(CA)`, die `California` bzw. `Vermont` zugeordnet werden.

Algorithmus 2 Extraktion von Zusätzen und Speichern zugehöriger Codes

```
1: for all entity in Ontology do
2:     if entity has suffixSign then                    ▷ suffixSign: ',' or '()'
3:         extract suffix
4:         if suffix.length = 2 AND suffix is upperCase then
5:             continue
6:         end if
7:         if suffix is adminDiv OR country then
8:             map(suffix, code)
9:         end if
10:    end if
11: end for
```

Wurden mehrere Übereinstimmungen gefunden, kann unter bestimmten Bedingungen trotzdem eine Zuordnung zu einem Land erfolgen:

1. Ländercode ist für alle Elemente gleich → wähle diesen Ländercode
   z. B. `Sapporo [JP.00, JP.12]`
2. Ländercode US ist enthalten → wähle US
   z. B. `Virginia [HN.13, US.VA]`

Ansonsten erfolgt keine Zuordnung, wie bei `Cambridge [GB.ENG, AU.08]`.

3.3.2 Direkte Zuordnung

Wurde der Schritt der Zuordnung mit Hilfe von Namenszusätzen (vgl. Kapitel 3.3.1) durchgeführt, müssen nun die Entitäten betrachtet werden, die keinen Zusatz tragen oder deren Zusatz nicht zugeordnet werden konnte. Bei letzteren muss der Zusatz entfernt werden, um folgenden Algorithmus auch auf diese anwenden zu können.

Da der Entitätsname nun keine genaueren Informationen mehr liefert, muss direkt in GeoNames nach diesem gesucht werden. Wie bereits erwähnt, wurden als Referenz nur GeoNames-Dateien mit einer Mindesteinwohnerzahl genutzt (vgl. Abschnitt „Orte“ in Kapitel 3.1.2). Aus diesen wurde eine Map erstellt, die zu jeder GeoNames-ID den regulären Orstnamen sowie alle alternativen Namen speichert. Weitere relevante Informationen zu den GeoNames-Features wie Ländercode und Einwohnerzahl wurden in einer zweiten Map unter der jeweiligen ID ausgelagert.

Für jede Entität wird nun nach direkten Namensübereinstimmungen in der GeoNames-Map gesucht und die ID für jeden Treffer gespeichert.

Abschließend muss die Zuordnung zu einem Land aufgrund der gespeicherten IDs und den zusätzlichen Informationen gefunden werden. Dabei können folgende Fälle auftreten:

- nur eine zugeordnete Übereinstimmung vorhanden → wähle deren Ländercode
- gespeicherte IDs haben alle den gleichen Ländercode → wähle diesen Ländercode
- gespeicherte IDs haben unterschiedlichen Ländercode → werte die Anzahl der Einwohner aus und wähle Ländercode der ID mit den meisten Einwohnern

Ansonsten erfolgt keine endgültige Zuordnung.

Für die Ontologie-Daten wurden zunächst nur Instanzen der Klasse `City` verwendet.

4 Evaluation

4.1 Vereinheitlichung der Ländernamen

Bei der Überprüfung der Korrektheit der Ländernamen wurde die englische Wikipedia zur Hilfe genommen. Ein gefundener Ländername gilt unter anderem als nicht korrekt, wenn er einen historischen Begriff für das Land bzw. das Gebiet des heutigen Landes darstellt oder nur einen geografischen Teil des Landes abdeckt. Beispielsweise würde die ehemalige französische Kolonie *French Guinea* nicht dem heutigen Land *Guinea* entsprechen und *Saint Helena* nicht *Saint Helena, Ascension and Tristan da Cunha*.

Tabelle 4.1 enthält die Gesamtzahl der Ländercodes, für die mindestens eine Übereinstimmung (Hit) zwischen Ländernamen in GeoNames und jeweils einer Ontologie-Entität der gegebenen Klasse(n) gefunden wurde.

In der Klasse `Country` gibt es für 24 von 250 Ländercodes keine Zuordnung. Für weitere drei Ländercodes wurden Kandidaten in der Klasse `Location` gefunden, jedoch fehlt hier, wie bereits erwähnt, die Instanz `Jamaica`. Die fehlenden Codes beinhalten zu einem großen Teil eher unbekannte Länder wie *Bonaire, Sint Eustatius and Saba* oder *Saint Pierre and Miquelon*, aber auch als wichtig empfundene Staaten wie *Japan* oder *Turkey*.

Außerdem wurden zwei *false-negative*-Elemente gefunden, also Länder, die in der Ontologie enthalten sind, aber denen kein Ländernamen aus GeoNames zugeordnet werden konnten:

Code	Name in Ontologie	regulärer Name in Geonames
`VI`	`United_States_Virgin_Islands`	`Virgin_Islands_of_the_United_States`
`SS`	`Southern_Sudan`	`South_Sudan`

Die aus den Hits resultierenden, entgültig gespeicherten Ländernamen wurden weiterhin auf ihre Korrektheit geprüft. Dabei wurde ein besonderes Augenmerk darauf gelegt, ob bei mehreren Hits der richtige Ländername gewählt wurde.

Aufgrund der Anzahl der Vorkommen wurde in der Klasse `Country` nur eine Fehlentscheidung getroffen. Diese bezieht sich auf die Instanz `North_Korea`, die nicht

Tabelle 4.1: Gesamtzahl der Ländercodes, denen mindestens eine Entität aus der Ontologie zugeordnet werden konnte sowie die Korrektheit der Zuordnung bei unterschiedlichen Arten der Entscheidungsfindung und für verschiedene Klassen.

	Country	Location	Location, Country, Region, City
Hits gesamt	226 (90,4 %)	228 (91,2 %)	229 (91,6 %)
- davon korrekt	219 (96,9 %)	211 (92,5 %)	218 (95,2 %)
mehrere Hits	67	108	108
Entscheidung nach Anzahl	65	101	106
- davon korrekt	64 (98,5 %)	95 (94,1 %)	100 (94,3 %)
Entscheidung nach Reihenfolge	2	7	2
- davon korrekt	1 (50 %)	2 (28,6 %)	1 (50 %)
ein Hit	159	120	121
- davon korrekt	154 (96,9 %)	116 (96,7 %)	117 (96,7 %)

gespeichert wird, da für `Korea` ein höherer Score existiert. Letztere Instanz wird allerdings meist synonym für das ebenfalls gespeicherte `South_Korea` verwendet. Die Ontologie ist hier nicht eindeutig.

Wird die Klasse `Location` mit einbezogen, so verringert sich die Genauigkeit insgesamt, da Elemente wie Städte oder Regionen das Ergebnis verfälschen.

Überdacht werden sollte das Vorgehen, bei gleicher Anzahl an Vorkommen, pauschal das erste Element der Liste zu wählen. Dabei ergibt sich höchstens eine Genauigkeit von 50 %. Die erfassten Begriffe können meist nicht als gleichbedeutend angesehen werden. Als Beispiel sollen folgende zwei Länder der Klasse `Country` dienen, für die neben dem heutigen Namen auch der Name aus britischer Kolonialzeit gespeichert ist. Die richtige Entscheidung wurde nur für `BZ` getroffen:

```
BZ  {Belize=1, British_Honduras=1}
MW  {Nyasaland=1, Malawi=1}
```

Die Fehler bei einem Hit von jeweils knapp 97 % sind meist darauf zurückzuführen, dass die zugeordneten Länder nach Definition nicht korrekt sind (siehe oben).

Zusammenfassend ergeben sich bei der Suche mit Entitäten der Klasse `Country` die besten Resultate für die gefundenen Hits, allerdings gibt es von diesen insgesamt weniger als in der Klasse `Location`.

Auch sollte erwähnt werden, dass mit dieser Überprüfung nicht komplett ausgeschlossen werden kann, dass ein Ländername unter einem weniger offensichtlichen Namen in der Ontologie geführt wird.

4.2 Relationsfindung

4.2.1 Zuordnung mit Hilfe von Namenszusätzen

Als Referenzdaten für die Verwaltungseinheiten wurden zunächst alle GeoNames-Features aus der Gruppe ADM1 verwendet. In Tabelle 4.2 sind allgemeine Werte für die Zuordnung angegeben ohne Aufschluss über die Genauigkeit der gefundenen Paare zu liefern. Nach ersten Stichproben wurde erkannt, dass die Menge der Referenzen zu gering ist und mehrere eindeutige, aber inkorrekte Zuordnungen gemacht werden. Als Beispiel kann hier der Zusatz `Jura` dienen, der lediglich einem Kanton in der Schweiz zugewiesen wird, sich aber auf eine Region in Frankreich bezieht, wie bei `Montrond,_Jura` und `Vadans,_Jura`. Erst bei der Hinzunahme der GeoNames-Gruppe ADM2 wird auch diese Relation erkannt.

Tabelle 4.2: Anzahl der gefunden Zusätze, Zusätze für die eine eindeutige Zuordnung gemacht werden konnte sowie Anzahl der Entitäten, die dadurch einem Land zugeordnet wurden.

	ADM1	ADM1 und ADM2
Zusätze gesamt	1.473	2.601
Zusätze mit eindeutiger Zuordnung	1423	2.423
zuordenbare Entitäten	97.806	108.572

Tabelle 4.3 macht nähere Angaben zu gefundenen Zusätzen und der Anzahl der getroffenen Entscheidungen, sowie deren Genauigkeit bei der kombinierten Suche in den Gruppen ADM1 und ADM2.

Waren einem Zusatz nur gleiche Ländercodes zugewiesen, konnte kein offensichtlicher Fehler in der Zuordnung erkannt werden. Die Entscheidung für `United_States` bei mehrfach zugeordneten Ländercodes inklusive US war in 90 % der Fällen korrekt. Immerhin 68,3 % der Zusätze können nicht zugeordnet werden, da es hierfür noch keine Regel gibt.

Tabelle 4.3: Aufteilung der gefundenen 2.601 Übereinstimmungen für ADM1 und ADM2 in einfache und mehrfache Hits sowie die Anzahl der getroffenen Entscheidungen mit Angabe der dabei ermittelten Genauigkeit.

ein Hit	2.339		
mehrfache Hits	262		
	gleicher Ländercode	73	(100 %)
	US enthalten	10	(90 %)
	keine Zuordnung	179	

An dieser Stelle wurde auch noch nicht umfangreich überprüft, inwieweit Orte in der Ontologie existieren, die zwar den gleichen Zusatz tragen, dieser sich aber auf unterschiedliche Länder bezieht. Zum jetztigen Zeitpunkt wurde nur ein Beispiel gefunden. So liegen die Instanzen `Offen_(Bergen)` in Deutschland und `Salhus,_Bergen` in Norwegen.

Generell sollten weitere Stichproben gemacht werden, um mehr über die Genauigkeit der Zuordnungen aussagen zu können.

Außerdem sollte in Betracht gezogen werden, auch GeoNames-Features der Gruppe ADM3 und ADM4 mit einzubeziehen, da die Ontologie durchaus noch Zusätze dieser Verwaltungseinheiten führt (z. B. `Mahottari` aus ADM3).

4.2.2 Direkte Zuordnung

Für diesen Ansatz existieren noch keine relevanten Datensätze. An dieser Stelle soll daher auf eine Auswertung verzichtet werden.

5 Zusammenfassung

Das Ziel dieser Arbeit war die Erweiterung einer Ontologie durch geografische Beziehungen. Nach einer kurzen Vorstellung der zugrundeliegenden Systeme Broccoli, YAGO und GeoNames wurden zunächst Mittel vorgestellt um relevante Daten aus den gegebenen Wissensdatenbanken zu extrahieren. Anschließend wurden mehrere Ansätze für eine effiziente Nutzung des umfangreichen Wissens vorgestellt und teilweise analysiert. Herausforderungen werden vor allem im Zusammenführen der beiden unabhängigen Datenquellen gesehen.

5.1 Ausblick

5.1.1 Laufzeit

Aufgrund der stetig wachsenden Datenmenge der Ontologie müssen Möglichkeiten gefunden werden, Informationen schneller und zielgerichteter extrahieren zu können, ohne dabei an Qualität zu verlieren. Erste Schritte wurden durch die Einschränkung der Eingabedaten auf relevante Kategorien gemacht. Weiterhin könnte z. B. darüber nachgedacht werden, nicht alle alternativen Namen mit einzubeziehen.

5.1.2 Qualität

Eine Ontologie sollte nicht nur auf das Speichern großen Wissens setzen, sondern auch der Qualität der eingetragenen Fakten einen hohen Stellenwert beimessen. Erste Analysen wurden mit verschieden Datensätzen gemacht. Um konkretere Ergebnisse erzielen zu können, müssen diese aber ausgeweitet werden.

5.1.3 Verbesserung der Ontologie

Eine Ontologie kann weder als vollständig noch als fehlerfrei betrachtet werden. Da sie allerdings die Grundlage für ihre eigene Erweiterung darstellt, ist ihre Qualität auch maßgeblich für die Qualität des Endergebnisses verantwortlich. So würde eine einheitliche Namensgebung das Auslesen von Daten bzw. die Vereinheitlichung mit anderen Datenbanken vereinfachen, wie in dieser Arbeit besonders an den Namenszusätzen deutlich wurde.

5.1.4 Relevanz

Einhergehend mit Laufzeit und Qualität sollten Möglichkeiten gefunden werden, Fakten eine Relevanz hinzuzufügen. In dieser Arbeit wurden Versuche mit der Anzahl der Vorkommen gemacht. Auch die Miteinbeziehung von Einwohnerzahlen wurde angedacht. Welche Leistungen die Ergebnisse tatsächlich bringen, muss in weiteren Untersuchungen ausgewertet werden. Die neuste Version von YAGO wurde außerdem um Koordinaten erweitert, welche in Zukunft mit einbezogen werden könnten.

Danksagung

Als erstes möchte ich mich bei meiner Gutachterin und Betreuerin Prof. Dr. Hannah Bast bedanken, die mir das Thema für diese Bachelorarbeit gab und mich stets ermutigte, über meinen eigenen Schatten zu springen.
Danke auch an Björn Buchhold und Florian Bäurle, die mir gerade in den letzten Wochen und Tagen hilfreiche Tipps geben konnten.
Einen besonderen Dank möchte ich an David Klein für seinen moralischen Beistand richten.

Literaturverzeichnis

[BBBH12] Bast, H. ; Bäurle, F. ; Buchhold, B. ; Haussmann, E.: Broccoli: Semantic Full-Text Search at your Fingertips. In: *CoRR* abs/1207.2615 (2012)

[Gru93] Gruber, T. R.: A Translation Approach to Portable Ontologies. In: *Knowledge Acquisition* 5 (1993), Nr. 2, S. 199–220

[Hau11] Haussmann, E.: *Contextual Sentence Decomposition with Applications to Semantic Full-Text Search*, Albert-Ludwigs-Universität Freiburg, Masterarbeit, 2011

[HSBW12] Hoffart, J. ; Suchanek, F. M. ; Berberich, K. ; Weikum, G.: YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. In: *Artificial Intelligence Journal – Special Issue* (2012)

[SKW07] Suchanek, F. M. ; Kasneci, G. ; Weikum, G.: YAGO - A Core of Semantic Knowledge. In: *WWW 2007*, 2007

[SKW08] Suchanek, F. ; Kasneci, G. ; Weikum, G.: YAGO: A Large Ontology from Wikipedia and WordNet. In: *J. Web Sem.* 6 (2008), Nr. 3, S. 203–207