Karlsruhe Institute of Technology

Extraktion und Konsolidierung von Webformularen zur Erzeugung von aktiven Ontologien

Bachelorarbeit
von

Thomas Mayer

An der Fakultät für Informatik
Institut für Programmstrukturen
und Datenorganisation (IPD)

Erstgutachter:	Prof. Dr. Walter F. Tichy
Zweitgutachter:	Prof. Dr. Ralf Reussner
Betreuender Mitarbeiter:	Dipl.-Inform. Martin Blersch

Bearbeitungszeit: 10.09.2016 – 10.01.2017

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Die Regeln zur Sicherung guter wissenschaftlicher Praxis im Karlsruher Institut für Technologie (KIT) habe ich befolgt.

Karlsruhe, 09.01.2017

. . . . . .

(**Thomas Mayer**)

Kurzfassung

Die Verwendung von Software über natürliche Sprachverarbeitungssysteme ist für viele Menschen Alltag. Sie gewährleisten eine einfache und intuitive Bedienung. Eine Möglichkeit einen Sprachassistenten zu erstellen ist die Verwendung einer aktiven Ontologie. Die manuelle Erstellung einer aktiven Ontologie gestaltet sich oft sehr aufwendig. Das Projekt „Easier“ handelt von der Automatisierung dieses Vorgangs, um diesen Prozess zu vereinfachen. Diese Arbeit befasst sich mit der Umsetzung einer Komponente des Projektes.
Ziel von „Easier“ ist die automatische Generierung von aktiven Ontologien für formularbasierte Internetdienste. Der Prozessablauf gestaltet sich wie folgt.
Zunächst werden die formularbasierten Internetdienste in Dienstkategorien eingeteilt. Im zweiten Schritt wird automatisch aus einer Dienstkategorie ein Konstruktionsplan generiert. Anhand von diesem Konstruktionsplan wird eine aktive Ontologie von der letzten Komponente generiert.
Diese Arbeit beschreibt die Umsetzung des zweiten Schrittes. Für die Verwirklichung dieses Schrittes werden automatisch semantisch gleiche Elemente zwischen den Formularen der Internetdienste bestimmt und aufeinander abgebildet. Mit verschiedenen Verfahren werden die semantischen Ähnlichkeiten zwischen den Elementen festgelegt. Anschließend werden alle semantisch gleichen Elemente über ein Cluster-Verfahren in Gruppen eingeteilt. Mithilfe von erstellten Regeln werden in dem letzten Schritt alle semantisch gleichen Elemente aufeinander abgebildet und ein Konstruktionsplan generiert.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

Die Verwendung der natürlichen Sprache ist ein immer größer werdender Bestandteil der Nutzung von Geräten und Software. Sie gewährleistet eine intuitive Bedienung und ermöglicht eine einfache und schnelle Nutzung verschiedener Anwendungen. Das Wählen einer Telefonnummer oder die Bedienung von Google Maps mithilfe eines Sprachassistenten, ist für viele Menschen, während der Nutzung ihres Handys, Alltag geworden. Bekannte Sprachassistenten sind Siri, Google Now und Cortona.

Aktive Ontologien(AOs)[Guz08] sind der Ansatz den Siri verwendet, um natürliche Sprache zu verarbeiten. Diese besitzen Sensorknoten, welche natürliche Spracheingaben erhalten und verarbeiten. Durch diesen Informationsgewinn ist es anschließend möglich Dienste aufzurufen. Das manuelle Erstellen der aktiven Ontologien ist sehr aufwendig. „Easier“[BL16] ist ein Ansatz um, dies zu vereinfachen, mit dem Ziel, die Erstellung von aktiven Ontologien zu automatisieren. Es erstellt AOs für formularbasierte Internetdienste, welche zunächst in Dienstkategorien eingeteilt werden. Eine Dienstkategorie enthält Internetdienste, welche sich in der selben Branche befinden. Beispiele hierfür sind die Dienstkategorien Verkehrsbetriebe und Hotelketten. Aus diesen Dienstkategorien wird im nächsten Schritt ein Konstruktionsplan generiert. Mithilfe von diesem Konstruktionsplan werden AOs für diese Dienstkategorien automatisch erstellt. Diese aktiven Ontologien können natürliche Sprache für alle formularbasierten Internetdienste ihrer Dienstkategorie verarbeiten.

Ein Teilbereich des Projektes handelt von der Konsolidierung einzelner formularbasierter Internetdienste derselben Dienstkategorie. Die Verwirklichung dieses Teilbereiches ist die Aufgabe dieser Bachelorarbeit.

1.1. Zielsetzung

Das Ziel dieser Arbeit ist die automatische Konsolidierung von formularbasierten Internetdiensten einer Dienstkategorie. Das Ergebnis ist ein Konstruktionsplan in XML-Format. Das Projektes „Easier“ lässt sich in drei Teilbereiche unterteilen. Diese Arbeit stellt den zweiten Teilbereich des Projektes dar.
Der erste Teilbereich befasst sich mit der Aufgabe formularbasierte Internetdienste in Dienstkategorien einzuteilen. In Abbildung 1.1 kennzeichnet der äußere gestrichelte Kasten das Projekt „Easier“. Die linke Wolke enthält Formulare verschiedener Dienstkategorien. Diese werden anschließend den Dienstkategorien der rechten Wolken zugeordnet. In den

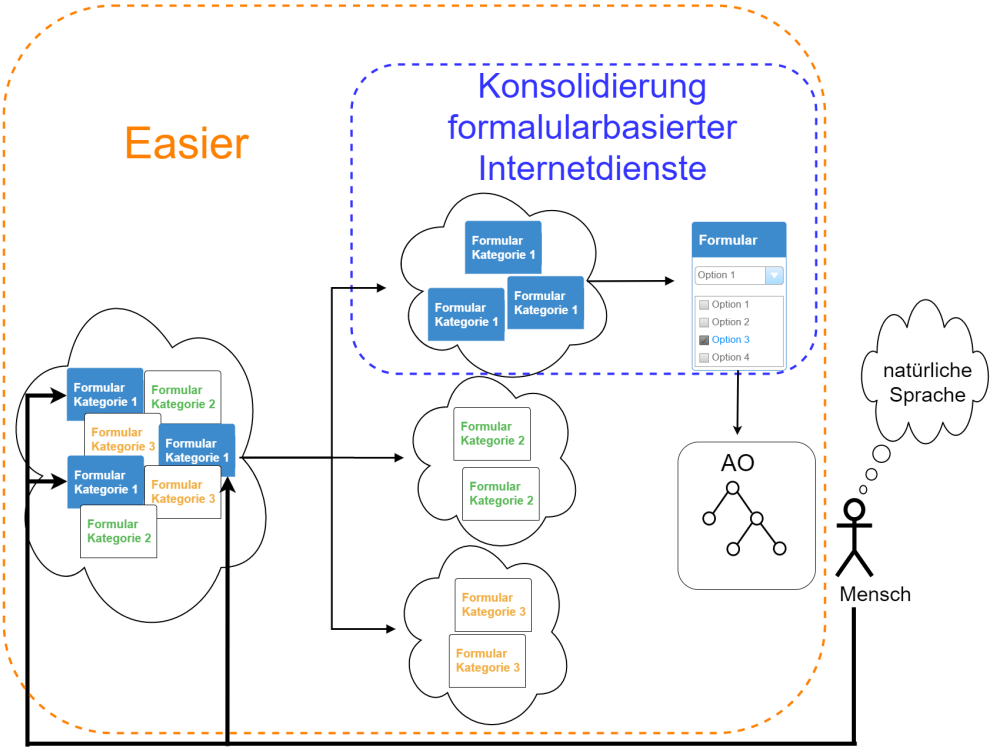

Abbildung 1.1.: Verschiedene formularbasierte Internetdienste werden in Dienstkategorien eingeteilt. Formularbasierte Internetdienste dieser Dienstkategorien werden anschließend auf einen Konstruktionsplan je Dienstkategorie abgebildet. Aus diesen Konstruktionsplänen werden im nächsten Schritt aktive Ontologien erzeugt.

Schritten zwei und drei werden für diese Dienstkategorien jeweils AOs erstellt. Zunächst werden verschiedene formularbasierte Internetdienste derselben Dienstkategorie auf einen Konstruktionsplan abgebildet. Die Automatisierung dieses Teiles ist die Aufgabe dieser Arbeit und ist in Abbildung 1.1 durch den inneren gestrichelten Kasten gekennzeichnet. Aus diesen Konstruktionsplänen werden schließlich, in dem letzten Abschnitt, automatisch aktive Ontologien erzeugt.
Die daraus entstandene AO kann natürliche Sprache für den in Schritt zwei konsolidierten Konstruktionsplan, verarbeiten. Zusätzlich können alle formularbasierten Internetdienste, aus welchen dieser Konstruktionsplan erzeugt wurde, über eine natürliche Spracheingabe genutzt werden.

Für die automatische Konsolidierung formularbasierter Internetdienste einer Dienstkategorie werden verschiedene Ansätze, wie beispielsweise Zusammenführung von Schemas oder Ontology Merging, untersucht. Diese Ansätze werden dazu verwendet Regeln aufzustellen, um semantisch gleiche Formularelemente zu erkennen. Es ist erforderlich, die Regeln ohne eine Anpassung für jede Dienstkategorie verwenden zu können. Daher muss diese semantische Gleichheit unabhängig von einer Dienstkategorie erkannt werden. Anschließend wird ein Werkzeug erstellt, welches die Umsetzung dieser Arbeit verwirklichen wird. Dieses Werkzeug verwendet die erstellten Regeln, um automatisch formularbasierte Internetdienste der gleichen Dienstkategorie zu einem Konstruktionsplan zusammenzufassen.

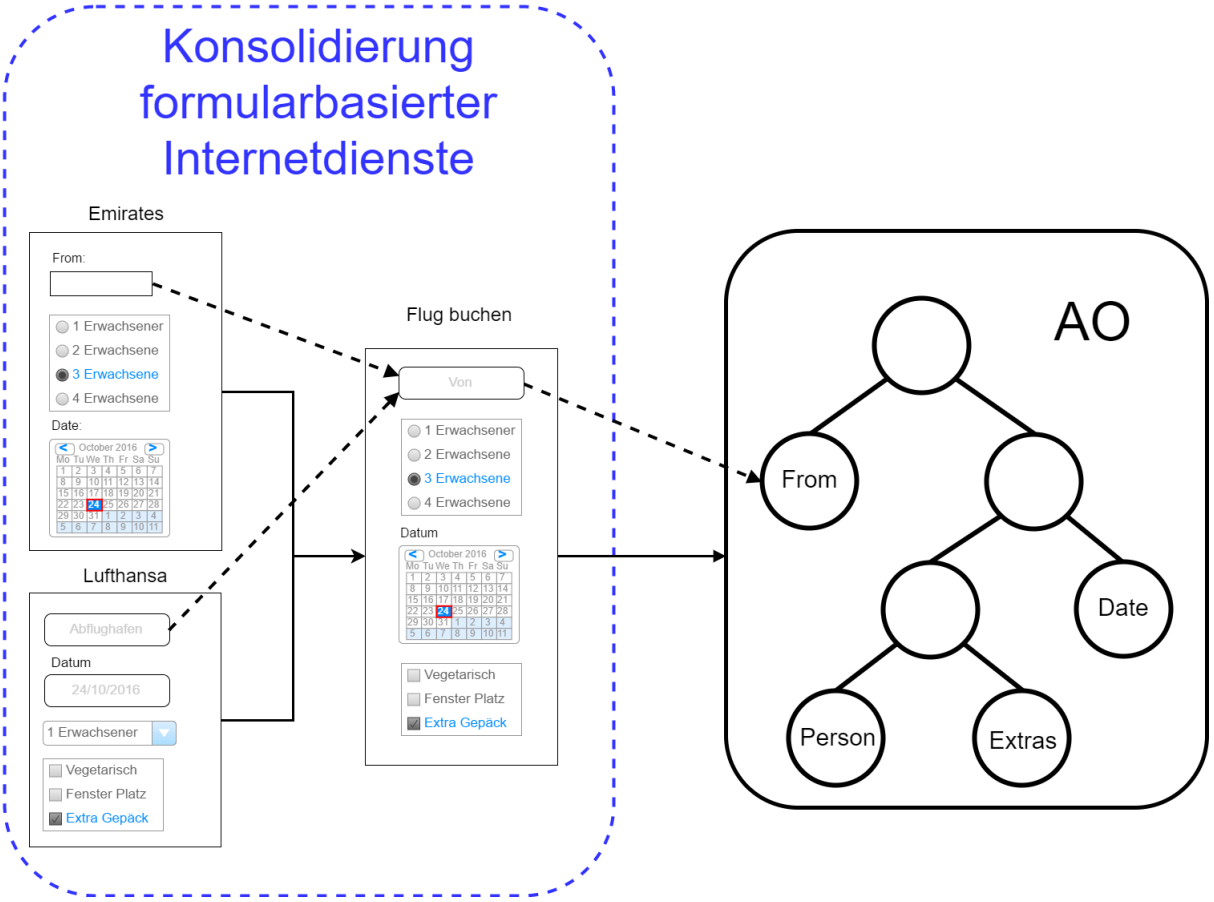

Abbildung 1.2.: Konsolidierung von zwei formularbasierten Internetdiensten derselben Dienstkategorie.

1.2. Veranschaulichung

Das Beispiel in Abbildung 1.2 zeigt, wie aus zwei formularbasierten Internetdiensten ein Konstruktionsplan erstellt und anschließend aus diesem eine aktive Ontologie generiert wird. Der Teil, welcher in dieser Arbeit umgesetzt wird, ist durch den gestrichelten Kasten gekennzeichnet. Exemplarisch wurden formularbasierte Internetdienste der Dienstkategorie „Flugbuchung“ verwendet. Der in der Mitte stehende Konstruktionsplan ist das Ergebnis der Zusammenführung der beiden formularbasierten Internetdiensten. Beispielsweise werden die beiden Formularelemente, welche für den Abflughafen zuständig sind, auf dasselbe Formularelement in dem Konstruktionsplan abgebildet. Dies findet für alle Teile, welche zugeordnet werden können, automatisch statt. Anschließend werden die Formularelemente des Konstruktionsplanes in einer weiteren Arbeit auf Knoten einer aktiven Ontologie abgebildet. Diese Vorgänge sind in Abbildung 1.2 durch die gestrichelten Pfeile gekennzeichnet.

1.3. Struktur der Arbeit

Die Arbeit ist in fünf Abschnitte unterteilt. Der erste Abschnitt befasst sich mit den Grundlagen. Hier werden alle Informationen und elementaren Begriffe, die zum Verständnis der Arbeit benötigt werden bereitgestellt. Danach werden verwandte Arbeiten vorgestellt und analysiert. In dem Analyse Kapitel wird die Problemstellung analysiert und eine Lösungsmöglichkeit entwickelt. Das Kapitel Entwurf und Implementierung enthält den verwendeten Entwurf und die umgesetzte Implementierung dieser Arbeit. Als Nächstes werden

das erstellte Werkzeug und die verwendeten Verfahren in der Evaluation bewertet. Dazu werden die Verfahren einzeln und in verschiedenen Kombinationen mit anderen Verfahren evaluiert. Zuletzt wird eine Zusammenfassung mit einem Ausblick bereitgestellt.

2. Grundlagen

In diesem Kapitel werden die Grundlagen, welche zu dem Verständnis der Arbeit benötigt werden, erläutert. Der erste Abschnitt befasst sich mit dem Thema Internetdienste. Als Nächstes wird die Sprache HTML und ihre Verwendung von Internetdiensten besprochen. Anhand dessen werden die verschiedenen HTML-Komponenten und der Aufbau eines Formulars vorgestellt. Die Sprache XML und die Unterschiede zu HTML werden in dem darauf folgenden Abschnitt erläutert. In dem letzten Teil des Kapitels werden die aktiven Ontologien und ihre Funktionsweise beschrieben.

2.1. Internetdienste

Internetdienste sind Dienste, welche über das Internet verwendet werden. Das Internet ist ein Netzwerk aus Rechnernetzen. Es gewährleistet eine Infrastruktur für einen Datenaustausch und bietet selbst keine Dienste an. Dienste, welche über das Internet verwendet werden, werden Internetdienste genannt. Das World Wide Web, E-Mail, Gopher, NET News oder FTP sind Beispiele für diese. Durch diese Dienste ist es möglich Bilder, Videos, Texte oder andere Daten über das Internet auszutauschen. In dieser Arbeit werden, für das zu erstellende Werkzeug, Formulare von HTML-basierten Internetdiensten als Eingabe verwendet. Diese Formulare werden mit der Auszeichnungssprache HTML erstellt. In den nachfolgenden Abschnitten werden ausschließlich Formulare von formularbasierten Internetdiensten behandelt.

2.2. HTML

Der Internetdienst World Wide Web und weitere verwenden für den Datenaustausch sogenannte Hypertexte. Mithilfe von Quellenverweisen innerhalb dieser Texte ist es möglich über die verschiedenen Hypertexte zu navigieren. Diese Verweise werden Hyperlinks genannt und sind ein Bestandteil des World Wide Webs. Die Auszeichnungssprache Hypertext Markup Language (HTML) ist die Hypertext Sprache, welche von dem Internetdienst World Wide Web verwendet wird. Eine Auszeichnungssprache ist eine maschinenlesbare Sprache. Diese ist für die Gliederung und Formatierung von Texte zuständig. Mit diesen ist es möglich die Abschnitte und und Elemente der Texte mit Eigenschaften, Zugehörigkeiten und Darstellungen zu beschreiben. In Folge dessen können Texte, Bilder, Videos oder andere Daten mithilfe eines Browsers dargestellt werden.
In den folgenden Abschnitten werden verschiedene HTML4 [htm99] und HTML5 [htm14] Komponenten vorgestellt.

```
<table>
  <tr>
    <td>1</td>
        <td>2</td>
        <td>3</td>
  </tr>
  <tr>
        <td>4</td>
        <td>5</td>
        <td>6</td>
  </tr>
</table>
```

Abbildung 2.1.: Darstellung einer Tabelle in HTML

1 2 3
4 5 6

Abbildung 2.2.: Darstellung einer HTML-Tabelle in einem Browsers

2.2.1. Aufbau

In der Sprache HTML bestehen keine Unterschiede zwischen der Groß- und Kleinschreibung von Zeichenfolgen. Um ein HTML-Dokument zu strukturieren und Bilder, Videos oder andere Dateien darstellen zu können verwendet HTML verschiedene Elemente, welche durch sogenannte „Tags“ dargestellt werden. Diese Elemente werden mit der unten angegebenen Syntax verwendet.

```
<Name des Tags> ... Inhalt des Elementes ... </Name des Tags>
```

Ein Element kann beispielsweise ein Knopf, eine Formatierung, ein Bild, ein Link oder auch eine Liste sein. Zusätzlich können zu einem Element auch weitere Attribute hinzugefügt werden. Diese können die Darstellung, den Eingabetyp oder andere Eigenschaften des HTML-Elementes beeinflussen. Die Attribute werden am Anfang des Elementes zwischen dem Vergleichszeichen und dem Tag Namen aufgelistet. Als Beispiel wurden die Attribute „type“ und „id“ zu dem Element „<button>“ hinzugefügt.

```
<button type="button" id="settings"> ... </button>
```

Elemente können als Inhalt wiederum Elemente beinhalten. Als Beispiel wird als Quelltextbeispiel eine Tabelle repräsentiert, welche die Zahlen von 1 bis 6 enthält und in Abbildung 2.1 zu sehen. In Abbildung 2.2 wird die Darstellung in einem Browser demonstriert. Der Tag „<table>“ gibt an, dass der Inhalt des Elementes eine Tabelle ist. In diesen wird das Element „<tr>“(table row) geschachtelt. Jedes „<tr>“ Element steht für eine Zeile in der Tabelle. Das Element „<td>“(table data) wiederum enthält jeweils den Inhalt einer Tabellenzelle.
Durch diese Verschachtlungen entstehen Bäume. Der Wurzelknoten ist das HTML-Element, welches alle anderen Elemente des Baumes enthält. Die erstellte Tabelle bildet den in Abbildung 2.3 dargestellten Baum.

Das Grundgerüst einer HTML-Seite bilden der „DOCTYPE“ und die Bereiche „<head>“ und „<body>“. Während der DOCTYPE die HTML Version angibt, umschließen die

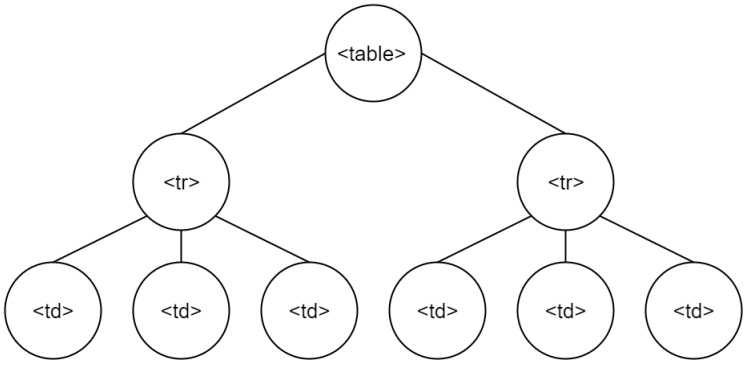

Abbildung 2.3.: Darstellung eines HTML Baumes, welcher eine Tabelle repräsentiert.

```
<!DOCTYPE html>
<html>
        <head>
                <title>Der Titel der Seite</title>
        </head>
        <body>
        </body>
</html>
```

Abbildung 2.4.: Grundgerüst einer HTML-Seite

anderen beiden Elemente jeweils den Kopf und den Inhalt der Seite. Der Kopf kann beispielsweise Verweise, Meta-Elemente und Titel enthalten. Den Aufbau und die Struktur der Seite enthält der <body>. Mit dem Element „<title>“ kann der Seite ein Titel gegeben werden. Der Quelltext in Abbildung 2.4 zeigt das Grundgerüst einer HTML-Seite mit einem Titel.

Zusätzlich existieren sogenannte Standalone-Tags. Diese Elemente besitzen keinen Inhalt und es kann auf einen abschließenden Tag des Elementes verzichtet werden.
In Tabelle 2.1 werden weitere wichtige Tags vorgestellt.

2.2.2. Attribute

Verschiedenen HTML-Elementen können verschiedene Eigenschaften mit Attributen zugewiesen werden. Ein Attribut gehört immer zu einem Element. Es ist nicht möglich, ein At-

Tag	**Definition**
<!- - Kommentar - ->	Der Inhalt des Tags ist ein Kommentar
<b>	Der Text wird fett dargestellt
 	Zeilenumbruch
<h1> bis <h6>	Unterschiedliche Überschriften
<hr>	Dient zur Unterteilung der HTML-Seite in Abschnitte
	Beschreibt ein Listenelement
	Stellt eine nummerierte Liste dar
<p>	Stellt einen Paragraphen auf der HTML-Seite dar
<style>	Beschreibt die Darstellung der gesamten HTML-Seite
	Stellt eine nicht nummerierte Liste dar

Tabelle 2.1.: Allgemeine HTML Tags

Attribut	Definition
class	Mit diesem Attribut kann einem Element eine Klasse zugewiesen werden. Diese Klassen sind in einem „style sheet“ definiert und bestimmen die Darstellung, Position, Größe und andere Eigenschaften des Elementes.
contenteditable	Das Attribut legt die Veränderbarkeit des Inhaltes eines Elementes fest.
hidden	Bestimmt, ob das Element angezeigt wird oder nicht.
id	Die ID ist ein Bezeichner. Dieser kann nur einmal einem Element zugewiesen werden.
lang	Legt die Sprache des Inhalts des Elementes fest.
spellcheck	Überprüft den Inhalt des Elementes auf Rechtschreibung und Grammatik.
style	Mit diesem Attribut kann die Darstellung und Position des Elementes verändert werden.
title	Ein Titel kann einem Element mit diesem Attribut hinzugefügt werden.

Tabelle 2.2.: HTML Attribute, welche jedem Element zugewiesen werden können.

tribut ohne zugehöriges Element zu verwenden. Unterschiedliche HTML-Elemente können unterschiedliche Attribute besitzen. Mit diesen können die Darstellung, Veränderbarkeit und andere Eigenschaften der HTML-Elemente festgelegt werden. Globale Attribute sind Attribute, welche zu allen HTML-Elementen hinzugefügt werden können. Eine Liste von verschiedenen globalen Attributen befindet sich in Tabelle 2.2.

Andere Attribute können nur bestimmten oder einem einzigen Element zugewiesen werden. Eine Liste einiger dieser Attribute befindet sich in Tabelle 2.3. In der zweiten Spalte der Tabelle werden alle Elemente angegeben, welche das jeweilige Attribut enthalten können.

2.2.3. Formulare

Mit HTML können Formulare erstellt werden. HTML-Seiten ohne die Verwendung weiterer Hilfsmittel können nach dem Laden nicht mehr verändert werden. Formulare haben diesen Nachteil nicht. Auch wenn ein Formular bereits geladen wurde, können nachträglich clientseitige Änderungen stattfinden. Formulare sind die Grundlage von formularbasierten Internetdiensten und werden in dieser Arbeit als Eingabe für das zu erstellende Softwareprogramm verwendet.

Für das Erstellen eines Formulars wird der Tag „<form>“ verwendet. Dieses Element repräsentiert das Formular. Die wichtigsten Attribute eines Formulars sind „action“ und „method“.

action

Das Attribut action bestimmt, welche Aktion nach dem Absenden des Formulars ausgeführt werden soll. Beispielsweise könnte eine URL hinterlegt sein. Über eine Uniform Ressource Locator(URL) können Ressourcen im Internet identifiziert und lokalisiert werden. In diesem Fall würde die Ressource, welche mit der URL referenziert ist, über das Internet angefordert, über welche die Daten des Formulars anschießend übertragen werden.

method

Attribut	Element	Definition
accept	<input>	Legt fest, welche Dateitypen von dem Eingabefeld angenommen werden.
checked	<input>	Ein Element mit diesem Attribut wird vorausgewählt, wenn es der Typ zulässt.
cols	<textarea>	Legt die Spalten einer Textfeldes fest.
for	<label>, <output>	Gibt an, welches andere Element dem jeweiligen Element zugewiesen ist.
headers	<td>, <th>	Gibt die Überschriften an, welche zu einer Tabellenzelle gehören.
list	<input>	Bestimmt das Listen Element, welches das Eingabefeld Element beinhaltet.
max	<input>, <meter>, <progress>	Gibt die maximale Größe an, welche in das Eingabefeld eingegeben werden kann.
maxlength	<input>, <textarea>	Legt die maximale Anzahl an Zeichen fest, welche in das Eingabefeld eingegeben werden kann.
min	<input>, <meter>	Gibt die minimale Größe an, welche in das Eingabefeld eingegeben werden kann.
multiple	<input>, <select>	Erlaubt es dem Nutzer mehr als einen Wert einzugeben.
name	<button>, <fieldset>, <form>, <iframe>, <input>, <keygen>, <map>, <meta>, <object>, <output>, <param>, <select>, <textarea>	Vergibt Namen an Elemente.
pattern	<input>	Überprüft die Eingabe des Eingabefeldes mit einem regulären Ausdruck.
placeholder	<input>, <textarea>	Der Platzhalter gibt, an welche Werte das Eingabefeld erwartet.
readonly	<input>, <textarea>	Beschreibt Schreibgeschützte Elemente.
required	<input>, <select>, <textarea>	Besitzt ein Element dieses Attribut, muss dieses Element ausgefüllt werden, bevor das Formular abgeschickt wird.
rows	<textarea>	Bestimmt die Zeilen eines Textfeldes.
selected	<option>	Ein Options Element mit diesem Attribut wird in einem Aufklappmenü vorausgewählt.
size	<input>, <select>	Legt die Anzahl der angezeigten Auswahlmöglichkeiten fest.
step	<input>	Gibt die Intervallabstände zwischen gültigen Zahlen an.
type	<button>, <embed>, <input>, <link>, <menu>, <object>, <script>, <source>, <style>	Ein Element erhält durch dieses Attribut einen Typ.
value	<button>, <input>, , <option>, <progress>, <param>	Gibt den Wert an, welcher als Eingabe erwartet wird.

Tabelle 2.3.: HTML Attibute und Definition

```
<select size="4">
  <option>Option 1</option>
  <option>Option 2</option>
  <option>Option 3</option>
  <option>Option 4</option>
  <option>Option 5</option>
</select>
```

Abbildung 2.5.: Darstellung eines Aufklappmenüs Elementes in HTML

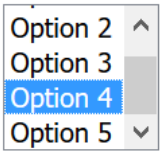

Abbildung 2.6.: Darstellung eines Aufklappmenüs in einem Browser

Als Eingabe für das Attribut method kann zwischen „post“ und „get“ gewählt werden. Die Daten, welche mit der get Methode übertragen werden, werden in der URL-Zeile des Browsers zusammen mit der angeforderten URL angezeigt. Dies ermöglicht es dem Absender des Formulars beispielsweise die URL zurückzuverfolgen, welche die verschickten Daten auswertet. Die abgesendeten Daten sind jedoch aufgrund der URL-Zeile auf eine gewisse Zeichenanzahl begrenzt. Die übertragenen Daten der post Methode besitzen keine Zeichenanzahl Grenze und werden nicht in der URL-Zeile angezeigt.

Im Folgendem werden verschiedene Formularelemente vorgestellt. Formularelemente sind die Bestandteile eines Formulars. Um die verschiedenen Formularelemente darzustellen werden Tags verwendet. Es werden in dem folgenden Abschnitt ein bezeichnendes Element , eine Auswahl, eine Option, ein Textfeld, ein Knopf, eine Liste und ein weiteres Eingabefeld vorgestellt.

Label (Bezeichnung):

Mit einem Label werden Abschnitte oder Formularelemente beschrieben. Mit dem in Tabelle 2.3 gelisteten Attribut „for“ kann ein Label einem Element zugewiesen werden. Alternativ kann für die Zuordnung das Element in das Label geschachtelt werden. Manche Formulare enthalten Bezeichnungen, welche keinem Element zugewiesen sind. In Bezug auf meine Arbeit erschweren diese Bezeichnungen den Automatisierungsprozess. Beispielsweise kann in diesem Fall nicht eindeutig bestimmt werden, ob das Label das vorhergehende Element oder das nachfolgende Element beschreibt.

Select (Auswahl) und Option:

Das select Element stellt ein Aufklappmenü dar. Innerhalb des Elementes werden option Elemente geschachtelt. Die einzelnen optionen stellen die Auswahlmöglichkeiten des Aufklappmenüs dar. Ist das multiple Attribut nicht gesetzt, ist es nicht möglich, mehr als eine option auszuwählen. Das unten stehende Quelltextbeispiel zeigt exemplarisch eine Darstellung eines solchen Aufklappmenüs. In Abbildung 2.5 wird die Darstellung des Aufklappmenüs in HTML repräsentiert. In Abbildung 2.6 ist die Ansicht des Menüs in einem Browser zu sehen.

Textarea (Textfeld):

```
<button type="button">Button-Text</button>
```

Abbildung 2.7.: Darstellung eines Knopf Elementes in HTML

Abbildung 2.8.: Darstellung eines Knopf in einem Browser

Eine Textarea ist ein Texteingabefeld. Die Größe des Feldes wird durch die Attribute „rows“ und „cols“ festgelegt, welche die Anzahl der Zeilen und Spalten angeben. Mithilfe des Attributes „maxlength“ kann eine maximale Anzahl an Zeichen festgelegt werden, welche in dieses Feld eingegeben werden dürfen.

Button (Knopf):

Mit dem Formularelement Button kann der Nutzer durch einen Klick auf diesen interagieren. Ein Button kann verschiedene Typen haben. Wird diesem kein Typ zugewiesen, erhält der Button den Typ „submit“. Dieser schickt bei einem Klick auf den Button das Formular ab. Zusätzlich können die Typen „button“ und „reset“ ausgewählt werden. Der Typ button löst eine clientseitige Aktion aus und der Typ reset setzt alle eingaben des Formulars zurück. Die Darstellung eines Knopfes in HTML wird ist in Abbildung 2.7 zu sehen. Abbildung 2.8 präsentiert diesen Knopf als Darstellung in einem Browser.

Input (Eingabe):

Das Element Input ist ein Eingabefeld. Diesem kann eine Vielzahl von Attributen und Typen zugewiesen werden. Der Defaulttyp ist der Typ „text“, welcher ein Textfeld definiert. Je nach Typ erwartet dieses Feld unterschiedliche Eingaben. Beispiele für solche Eingaben sind eine E-Mail-Adresse, ein Knopfdruck, ein Datum oder eine Datei. Die Typen dieses Eingabefeldes können in die Gruppen Texteingabe, Zeit, Button und Typen ohne passende Zuordnung unterteilt werden. Eine Auflistung der Typen befindet sich in Tabelle 2.4.

Datalist (Liste)

Ist eine vordefinierte Liste von Optionen für das Input Element. Mit dem Typ „list“ des Input-Elementes kann die Liste dem Eingabefeld hinzugefügt werden.

2.3. XML

Die Sprache Extensible Markup Language(XML) eine hierarchische Auszeichnungssprache, welche zwischen Groß- und Kleinschreibung unterscheidet [xml06]. Die beiden Sprachen sind aus der Sprache SGML entstanden und haben viele Gemeinsamkeiten. XML verwendet wie HTML ebenfalls Tags. Im Gegensatz zu HTML sind die Tags nicht vordefiniert. XML-Elemente können beliebig ineinander verschachtelt werden und eine Baumstruktur bilden.
Das Erste auf jeder XML-Datei ist eine Deklaration, welche die Version der Datei enthält. Zusätzlich kann zu dieser die Zeichencodierung hinzugefügt werden. Das Beispiel in Abbildung 2.9 zeigt die Ähnlichkeit zu HTML.

Um die Struktur von XML-Dokumenten zu beschreiben, können XML Schemata verwendet werden. Ein XML Schema(XSD) gibt die Struktur vor welche mit XML implementiert werden soll. Auch Datentypen und Instanzen von XML-Dokumenten werden mithilfe eines

Typ	Definition
Texteingabe	
email	Es wird eine E-Mail-Adresse als Eingabe erwartet.
password	Dieser Typ nimmt ein Passwort entgegen.
text	Dieser Typ erwartet einen Text als Eingabe.
search	Es handelt sich bei diesem Typen um ein Suchfeld.
url	Es wird eine URL erwartet.
Zeit	
date	Erwartet ein Datum als Eingabe.
datetime	Erwartet ein Datum und eine Uhrzeit als Eingabe und berücksichtigt die Zeitzone.
datetime-local	Erwartet ein Datum und eine Uhrzeit als Eingabe ohne die Zeitzone zu berücksichtigen.
month	Erwartet einen Monat und ein Jahr als Eingabe.
time	Erwartet eine Uhrzeit als Eingabe.
week	Erwartet eine Woche und ein Jahr als Eingabe.
Button	
button	Es wird clientseitige Aktion ausgelöst.
image	Legt einen unsichtbaren Button über ein Bild. Bei einem Klick auf diesen wird das Formular abgesendet.
reset	Das Formular wird zurückgesetzt.
submit	Das Formular wird abgesendet.
Typen ohne passende Zuordnung	
checkbox	Stellt Kontrollkästchen dar. Eingabefeld Elemente mit diesem Typ und demselben Namen gehören zu der gleichen Checkbox.
color	Ermöglicht die Auswahl von Farben.
file	Erwartet eine Datei als Eingabe.
hidden	Stellt ein Feld dar, welches nicht dargestellt wird.
number	Erwartet eine Zahl als Eingabe. Mit Attributen kann die erwartete Eingabe mit Zahlenbereichen eingeschränkt werden.
radio	Stellt Auswahlknöpfe dar. Input Elemente mit dem Typ radio und demselben Namen gehören zusammen.
range	Stellt einen Schieber dar.
tel	Erwartet eine Telefonnummer als Eingabe.

Tabelle 2.4.: Typen eines Eingabefeldes

```
<?xml ... ?>
<auto>
<karosserie>
<tür>Tür</tür>
</karosserie>
<reifen>Reifen</reifen>
<motor>Motor</motor>
</auto>
```

Abbildung 2.9.: Beispiel eines XML-Dokumentes

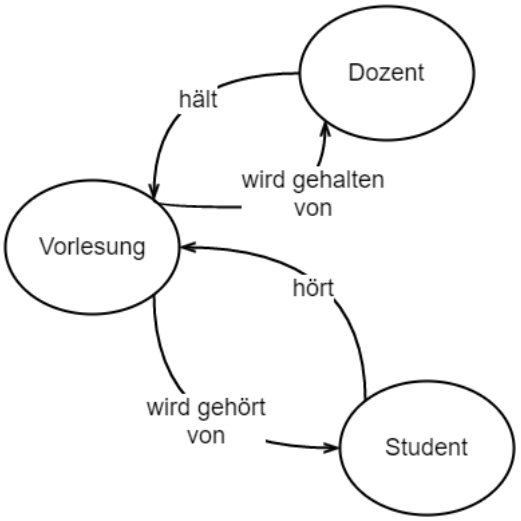

Abbildung 2.10.: Darstellung einer Vorlesungsveranstaltung als Ontologie

Schemas beschrieben.
In dieser Arbeit wird XML verwendet, um einen Konstruktionsplan, welcher die globalen Elemente der HTML-Formulare der verschiedenen Dienstkategorien repräsentiert, darzustellen. Aus diesem wird, mithilfe einer weiteren Arbeit, eine aktive Ontologie erstellt.

2.4. Ontologie

Der Begriff Ontologie kann verschiedene Bedeutungen in verschiedenen Kontexten besitzen [NG09][Tur06]. In diesem Kapitel wird der Begriff Ontologie in dem Kontext der Informatik erläutert.
Eine Ontologie beschreibt eine „Welt“ mit Objekten, welche in Beziehungen zueinanderstehen. Eine andere Beschreibung einer Ontologie ist ein formales Model, welches Objekte und deren Beziehungen zueinander in einem System beschreibt. In einem Zitat von Gruber [Gru93] wird eine Ontologie wie folgt definiert:

> „*an ontology is a specification of a conceptualization*“

In Abbildung 2.10 ist eine Beispiel Ontologie zu sehen. Das System ist eine Vorlesungsveranstaltung. Die Objekte des Systems sind in diesem Beispiel der Dozent, die Vorlesung und der Student. Diese sind über Beziehungen miteinander verbunden. Beispielsweise existiert eine Beziehung zwischen Dozenten und Vorlesung, welche angibt, dass die Vorlesung von dem Dozenten gehalten wird.

2.5. Aktive Ontologie

Eine aktive Ontologie (AO) ist sowohl eine Datenstruktur als auch eine Ausführungsumgebung. AOs können als Ontologien betrachtet werden, bei welchen eine Schicht aus Prozessen über die Datenstruktur gelegt wurde. In seiner Dissertation „*Active: A unified platform for building intelligent applications*“ beschreibt Didier Guzzoni den Aufbau und die Funktion dieser mit anschließender Umsetzung von Netzwerken mit natürlicher Sprachverarbeitung [Guz08]. Dieses Konzept wird beispielsweise von dem Sprachassistenten Siri verwendet.

2.5.1. Aufbau und Prozessablauf einer AO

Eine aktive Ontologie besteht aus Begriffen, welche Regeln beinhalten und einem Faktenspeicher. Dies wird in Abbildung 2.11 dargestellt. Die verschiedenen Begriffe sind über

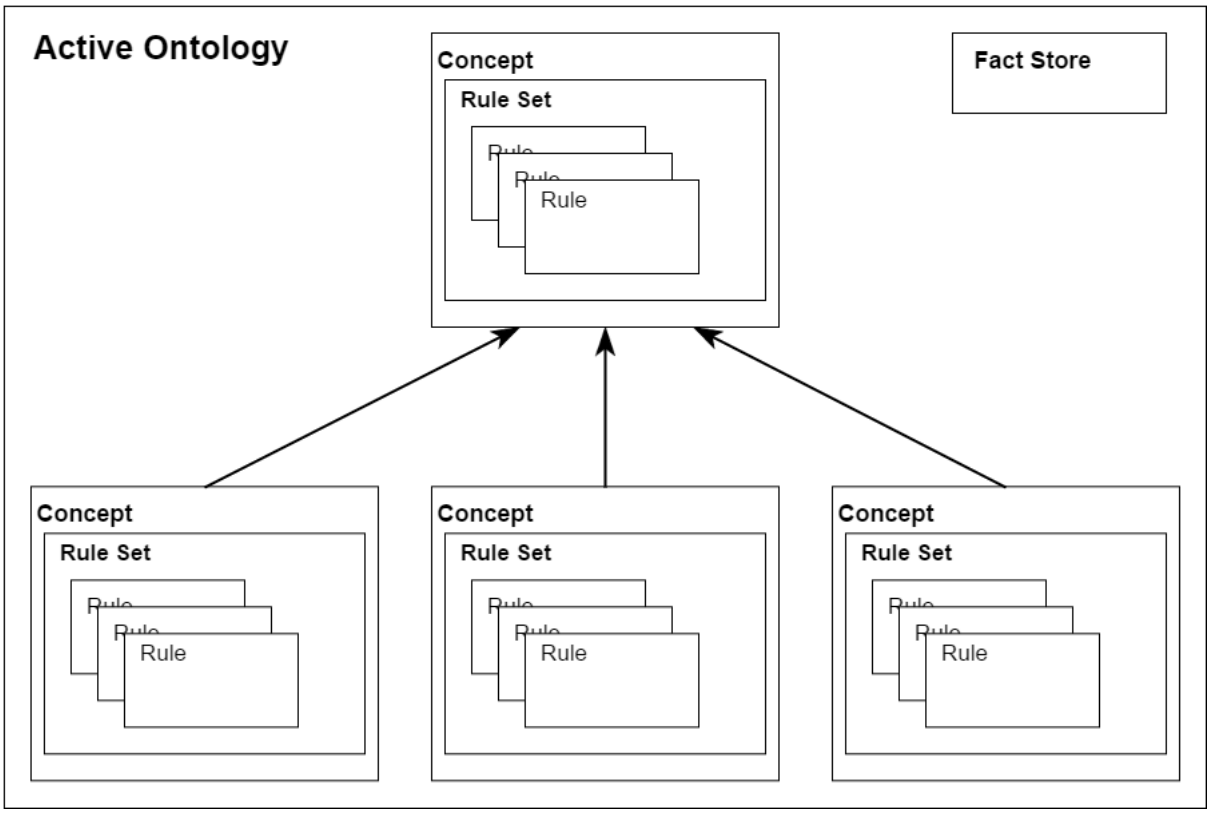

Abbildung 2.11.: Komponenten einer aktiven Ontologie [Guz08]

.

unidirektionale Beziehungen miteinander verbunden. Der Faktenspeicher beinhaltet Fakten, welche von den Begriffen mithilfe ihrer Regeln verarbeitet werden. Im Folgendem werden zunächst der Faktenspeicher und anschließend der Prozessablauf erläutert.

2.5.1.1. Faktenspeicher

Der Faktenspeicher ist der Datenspeicher einer aktiven Ontologie. Dieser stellt zugleich den aktuellen Ausführungszustand seiner AO dar. Es können Fakten in diesem Speicher gelöscht, verändert und hinzugefügt werden. Es existieren 4 verschiedene Typen von Fakten, welche im Folgendem erläutert werden.

Der einfache Fakt

Diese Fakten stellen eine Konstante dar, welche atomar ist.

Die Variablen

Diese Fakten stellen Variablen dar.

Der komplexe Fakt

Diese Fakten bestehen aus einem oder mehreren Fakten und besitzen einen Namen.

Die Fakten Liste

Diese Fakten sind eine Liste von Gruppen, welche Fakten beinhalten. Im Gegensatz zu dem komplexen Fakt besitzen diese keinen Namen.

2.5.1.2. Auswertungszyklus

Der Faktenspeicher wird regelmäßig auf Änderungen überprüft. Wurden bei der Überprüfung des Faktenspeichers Änderungen entdeckt, wird ein sogenannter Auswertungszyklus gestartet. Dieser wertet die Regeln der Begriffe aus. Jede Regel eines Begriffes besteht aus einer Bedingung und einer Aktion. Trifft eine Bedingung zu, so wird die zugehörige Aktion der Regel ausgeführt. Diese können auch den Faktenspeicher verändern und wiederum einen Auswertungszyklus verursachen.

2.5.2. Natürliche Sprachverarbeitung mit AO basierten Netzwerken

Ein Netzwerk für die Verarbeitung der natürlichen Sprache besteht aus einem Baum. Die Knoten des Baumes repräsentieren die Begriffe der AO. Die Beziehungen zwischen Kinderknoten und Elternknoten die Beziehungen der AO. Dabei sind die Beziehungen immer in die Richtung der Elternknoten gerichtet.
Die natürliche Sprache wird über ein Bottom-up Verfahren verarbeitet. Jeder Knoten überprüft, ob eine seiner Bedingungen zutrifft. Ist dies der Fall, gibt er das Ergebnis an den Elternknoten weiter. Die Ausgabe des Wurzelknotens ist das Ergebnis. Jeder Knoten überprüft lediglich die Ergebnisse seiner Kinderknoten. Die einzelnen Blätter, die sogenannten Sensorknoten des Baumes, filtern die natürliche Spracheingabe. Diese Eingabe und alle Ergebnisse repräsentieren den aktuellen Ausführungszustand und damit den Faktenspeicher des Netzwerks.
Sensorknoten können verschiedene Typen besitzen. Diese sind im Folgendem aufgelistet.

Knoten mit Wörterlisten

Dieser Knoten benötigt von dem Nutzer vordefinierte Wörterlisten. Dieser führt eine Aktion aus, wenn eines der Eingabewörter ein Wort aus der Wörterliste ist.

Prefix Knoten

Dieser Knoten benötigt von dem Benutzer vordefinierte Wörterlisten. Wird ein Wort aus der Eingabe gefunden, welches sich in der Wörterliste befindet, werden die nachfolgenden Wörter der Eingabe überprüft.
Beispielsweise könnten die Wörter in der Liste Zahlen sein. In diesem Fall könnten die nachfolgenden Wörter nach einer Bedeutung abgesucht werden. Ein Beispiel wäre die Zahl 3 und die Bedeutung Brezel für „3 Brezeln“.

Postfix Knoten

Dieser Knoten benötigt von dem Benutzer vordefinierte Wörterlisten. Wird ein Wort aus der Eingabe gefunden, welches sich in der Wörterliste befindet, werden die vorausgegangenen Wörter der Eingabe überprüft.
Beispielsweise könnte ein Wort aus der Liste „Tage“ sein. In diesem Fall könnten die nachfolgenden Knoten nach einer Anzahl abgesucht werden. Ein Beispiel wäre die Anzahl 3 für „3 Tage“.

Knoten mit regulären Ausdrücken

Diese Knoten suchen nach passenden regulären Ausdrücken in der Eingabe. Diese Ausdrücke werden von dem Nutzer vordefiniert.
Ein Beispiel wäre ein regulärer Ausdruck welcher alle Zahlen beinhaltet. Wird eine Zahl in der Eingabe gefunden, so führt dieser Knoten eine Aktion aus.

Spezialisierte Knoten

Es wird bei diesem Knoten nach bestimmten Schemas in der Eingabe, wie beispielsweise einer Zeitangabe gesucht, welche ein genaues Datum, ein Wochentag oder eine andere relative Angabe sein kann. Die Schemas erhält der Knoten über den Nutzer.

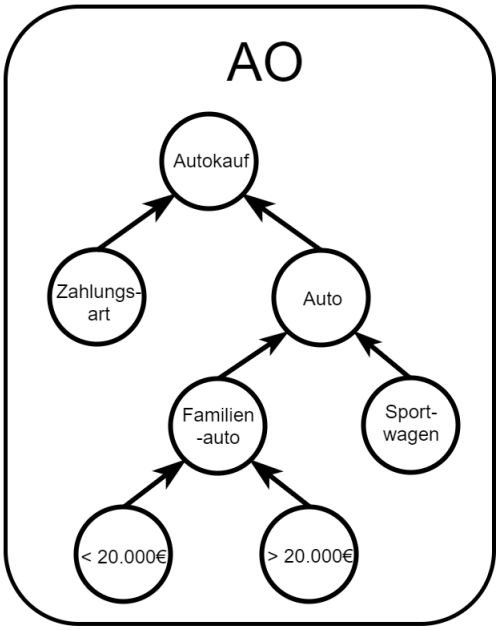

Abbildung 2.12.: Beispiel einer aktiven Ontologie

Helfer Knoten

Der Helfer Knoten sucht wichtige Informationen in der Spracheingabe. Wenn der Elternknoten ein Ergebnis aus diesem Knoten erhält, wird immer eine Aktion ausgelöst. Er kann beispielsweise eine Wörterliste von dem Nutzer erhalten. Findet dieser Knoten ein Wort aus der Liste in der Eingabe, leitet er das Ergebnis an den Elternknoten weiter, welcher anschließend eine Aktion startet.

Nicht-Sensorknoten können die Typen Sammelknoten oder Auswahlknoten besitzen. Diese Typen werden im Folgendem erläutert.

Sammelknoten

Ein Sammelknoten sammelt die Ergebnisse seiner Kinder. Jedes Mal wenn ein neues Ergebnis eintrifft, führt dieser eine semantische Bewertung durch. Das Ergebnis wird an seinen Elternknoten weitergeleitet.

Auswahlknoten

Ein Auswahlknoten führt für jedes Ergebnis eine semantische Bewertung aus. Das beste Ergebnis wird an seinen Elternknoten weiter gereicht.

Sind in der Eingabe nicht ausreichend Informationen vorhanden, um ein ausreichendes semantisches Ergebnis des Wurzelknotens für eine Ausgabe zu erzeugen, muss eine Nachfrage an den Nutzer stattfinden. Dies wird so oft wiederholt, bis das Netzwerk ausreichend Informationen erhalten hat.

2.5.3. Beispiel

In dieser Arbeit wird ein Konstruktionsplan erstellt, aus welchem mithilfe der Arbeit „*Name der Arbeit*“ von Kay Nachname [**?**] eine aktive Ontologie erstellt wird. In Abbildung 2.12 ist ein Beispiel einer aktiven Ontologie zu dem Thema Autokauf zu sehen. Für den Wurzelknoten *Autokauf*, welcher ein Sammelknoten ist, sind die Eingaben Zahlungsart und Auto notwendig. Der Knoten Zahlungsart ist Knoten mit einer Wörterliste. Dieser sucht speziell nach Wörtern wie „Ratenzahlung“ oder „Direktkauf“. Der andere Knoten ist

ein Auswahlknoten für die Ausgabe Auto. Es ist möglich, entweder ein Familienauto oder einen Sportwagen zu suchen. Das Familienauto ist wiederum ein Auswahlknoten. Es ist möglich nach Familienautos über oder unter 20.000 Euro zu suchen.
Sollte bei einer Spracheingabe beispielsweise ein Ergebnis des Knotens der Zahlungsart fehlen, kann der Sammelknoten Autokauf kein gültiges Ergebnis ausgeben. In diesem Fall müsste eine Nachfrage an den Nutzer ausgeführt werden, um alle benötigten Informationen zu erhalten.

2.6. Zusammenfassung

In diesem Kapitel wurden zunächst die Internetdienste erläutert. Dabei wurde die Wichtigkeit der formularbasierten Internetdienste, welche die Eingabe für das zu erstellende Werkzeug dieser Arbeit bereitstellen, hervorgehoben. In diesem Zusammenhang wurden der Formularaufbau und die Formularelemente demonstriert. Die Sprache XML, welche für die Erstellung des Konstruktionsplanes verwendet wird, wurde in dem nächsten Abschnitt veranschaulicht. Ontologien wurden im Anschluss als Grundlage des Kapitels der verwandten Arbeiten kurz erläutert. Zuletzt wurden aktiven Ontologien, welche aus dem Konstruktionsplan in einer weiteren Arbeit erstellt werden beschrieben.

3. Verwandte Arbeiten

Das Ziel dieser Arbeit ist die Zusammenführung verschiedener Formulare aus einer Dienstkategorie zu einem globalen Konstruktionsplan. Die Schwerpunkte liegen in der Erkennung der semantischen Gleichheit von HTML-Elementen aus verschiedenen HTML-Formularen und der Abbildung dieser auf ein globales Element. In diesem Kapitel werden verschiedene Ansätze betrachtet, welche dieses Ziel mit seinen Schwerpunkten umzusetzen. Dazu werden die Themengebiete Integrierung von HTML-Oberflächen des Deep Webs, Zusammenführung von Schemas und Ontology Merging besprochen.

3.1. Integrierung von HTML-Oberflächen des Deep Webs

Das Deep Web ist ein Teil des World Wide Webs, welcher nicht über Suchmaschinen gefunden werden kann. Durch die Zusammenführung von Oberflächen des Deep Webs wird das Ansprechen dieser Oberflächen über eine gemeinsame Schnittstelle ermöglicht.
Es existieren verschiedene Arbeiten, in denen das Thema zusammenführen von HTML-Dokumenten behandelt wurde. Diese Arbeiten müssen sich ebenfalls mit dem Grundproblem, dass finden semantisch gleicher HTML-Elemente befassen. Im Folgenden werden 2 dieser Arbeiten vorgestellt. Die erste Arbeit stellt einen Ansatz vor, welcher die Zusammenführung verschiedener Formulare behandelt. In der 2. Arbeit wird das Werkzeug Wise-Integrator präsentiert. Dieses erhält verschiedene Suchoberflächen als Eingabe und erstellt aus diesen eine globale Oberfläche, welche alle Eingaben dieser Suchoberflächen verwalten kann.

3.1.1. Hierarchisches Clustering

In der Arbeit „An Interactive Clustering-based Approach to Integrating Source Query Interfaces on the Deep Web“ [WYDM04] werden verschiedene Ansätze vorgestellt, welche das Thema Zusammenführung von HTML-Formularen behandeln und verschiedene Probleme bestehender Arbeiten gelöst, welche im Folgenden aufgelistet sind.

1. Es werden nur Abbildungen von flachen Schemas erstellt.
2. Es werden nur 1:1 Abbildungen berücksichtigt.
3. Der Prozess wird neu gestartet, wenn ein Fehler auftritt.
4. Aufwendige Verbesserungen von Parametern sind notwendig.

Der Prozessablauf dieser Arbeit wird im Folgendem erläutert. Zunächst werden die Oberflächen der HTML-Formulare in eine hierarchische Baumstruktur überführt. Anschließend werden semantisch ähnliche HTML-Felder ermittelt und die komplexen Abbildungen, welche in diesem Kapitel erläutert werden, aussortiert. Mithilfe eines interaktiven hierarchischen Cluster-Verfahrens werden die semantisch ähnlichen Ergebnisse in Cluster geordnet. Zu diesen Clustern werden im Anschluss die komplexen Abbildungen hinzugefügt. Zum Schluss werden die optimalen Parameter für die unterschiedlichen Verfahren ermittelt und Abbildungen, welche nicht ermittelt werden konnten, mithilfe des Nutzers aufgelöst.

3.1.1.1. Hierarchische Darstellung von HTML-Formularen

Für die Verwendung von einigen Verfahren wird die Struktur der Formulare hierarchisch untergliedert. Hierfür wird eine Baumstruktur erstellt. Die verschiedenen Beschreibungen und Elemente des HTML-Formulars repräsentieren die Knoten des Baumes.

3.1.1.2. Erkennung semantisch gleicher Formularelemente

Die semantische Ähnlichkeit der Formularelemente ist die Summe der Linguistischen und der Domänen Ähnlichkeit. Die Domäne eines Feldes ist die Menge der erwarteten Eingabetypen. Für beide Ähnlichkeiten werden Parameter eingesetzt um diese unterschiedlich zu gewichten.

Linguistische Ähnlichkeit

Bevor die linguistische Ähnlichkeit bestimmt werden kann, wird eine Normalisierung durchgeführt. Diese besteht aus einer Tokenisierung und einer Umformung. Dies wird mithilfe von Wörterbüchern durchgeführt.
In der Tokenisierung werden zusammenhängende Wörter in einzelne Wörter unterteilt, welche als Gruppe Token bilden. Die anschließende Umformung verlängert Abkürzungen zu ihren vollständigen Wörtern.
Die linguistische Ähnlichkeit ist die Summe aus namens Ähnlichkeit, Label Ähnlichkeit und dem Maximum der Ähnlichkeiten zwischen Name und Label. Für diese Ähnlichkeiten werden wieder verschiedene Parameter für die Gewichtung verwendet.

Domänen Ähnlichkeit

Die Domänen Ähnlichkeit ist die Summe der Ähnlichkeit der Typen und der Ähnlichkeit der Eingabetypen der einzelnen Elemente. Auch bei dieser Summe werden Parameter verwendet, um die einzelnen Eingaben der Summe unterschiedlich zu gewichten.

3.1.1.3. 1:1 und 1:m Abbildungen

Es gibt verschiedene Arten von Abbildungen. Die Einfachste ist einen 1:1 Abbildung. Das bedeutet, dass ein zusammengeführtes Element, aus Elementen zusammengeführt wurde, welche alle aus unterschiedlichen Formularen stammen. Bei einer 1:m Abbildung oder auch komplexe Abbildung genannt, besteht das zusammengeführte Element aus Elementen, welche nicht alle aus unterschiedlichen Oberflächen stammen.
Es existieren 2 verschiedene Abbildungen für eine 1:m Abbildung. Die aggregierte und die ist-Element-von Abbildung.

aggregierte Abbildungen

In einer aggregierten Abbildung bilden alle Elemente der m-Seite zusammen das zusammengeführte Element. Die Elemente der m-Seite erwarten einen Teil von der

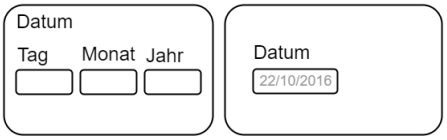

Abbildung 3.1.: aggregierte Abbildung

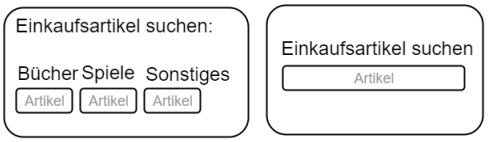

Abbildung 3.2.: ist-Element-von Abbildung

erwarteten Eingabe des zusammengeführten Elementes als Eingabe. In Abbildung 3.1 ist eine aggregierte Abbildung zu sehen. Die linke Seite dieser Abbildung ist die m-Seite. Alle 3 Eingabefelder dieser Seite bilden zusammen das auf der rechten Seite dargestellte Eingabefeld Datum. Das rechte Eingabefeld kann nur mit der Verwendung aller Eingabefelder der m-Seite verwendet werden.

ist-Element-von Abbildungen

In der ist-Element-von Abbildung, erwartet jedes der Elemente der m-Seite eine vollständige mögliche Eingabe des zusammengeführten Elementes. Die Elemente der m-Seite erwarten eine Untermenge der Menge der erwarteten Eingaben des zusammengeführten Elementes als Eingabe. Eine Eingabe der m-Seite ist auch eine Eingabe des zusammengeführten Elementes Seite. Eine Eingabe des zusammengeführten Elementes ist eine auch Eingabe eines der Elemente der m-Seite. In Abbildung 3.2 befindet sich ein Beispiel einer solchen Abbildung. Während auf der rechten Seite allgemein nach Einkaufsartikeln gesucht wird, wird auf der linken Seite speziell nach Büchern, Spielen oder sonstigen Artikeln gesucht. Alle Eingabefelder der linken Seite sind eine Untermenge des Eingabefeldes der rechten Seite.

3.1.1.4. Hierarchisches Cluster-Verfahren

Nach der Feststellung der linguistischen Ähnlichkeit werden über einen Cluster Algorithmus die semantischen Ähnlichkeiten bestimmt. Der hierarchische Cluster-Algorithmus wird über alle 1:1-Abbildungen durchgeführt. Die Ähnlichkeiten der Formularelemente werden in eine Matrix eingetragen. Die einzelnen Felder bilden jeweils ein Cluster und werden als Eingabe für den Cluster-Algorithmus verwendet. Die einzelnen Cluster werden über ein Greedy-Verfahren zusammengeführt. Das Ergebnis sind Cluster, welche alle semantisch gleichen Elemente enthalten.

3.1.1.5. Anpassungen der verwendeten Parameter und Nutzer Interaktionen

In den bisher genannten Verfahren sind viele verstellbare Parameter und Grenzwerte vorhanden. Dem Nutzer ist es möglich diese Werte anzupassen. Mithilfe von Tests und den daraus resultierenden Erkenntnissen können die Werte der Parameter verbessert werden, um die Anzahl der falsch zusammengeführten Elemente zu senken, sowie die Anzahl der richtig zusammengeführten Ergebnisse zu steigern.
Zum Schluss werden alle Abbildungen, welche nicht eindeutig bestimmt werden konnten, durch eine Nutzerinteraktion aufgelöst.

3.1.1.6. Ergebnisse

Die Testverfahren zeigen, dass die Genauigkeit des Algorithmus, für das automatische Abbilden von Feldern, zwischen 81% und 93.5% liegt.

3.1.1.7. Diskussion

In dieser Arbeit wurden einige Ansätze vorgestellt, welche HTML-Formulare und ihre Elemente zusammenführen können, mit einer zusätzlichen Betrachtung der komplexen Abbildungen. Es wird jedoch nicht erläutert, wie ein Formular oder eine andere Datei, welche die zusammengeführten Formulare beinhaltet dargestellt werden könnte. Dieser Schritt muss in dieser Bachelorarbeit zusätzlich durchgeführt werden. Zudem muss zurückverfolgt werden können, aus welchen ursprünglichen Elementen, die globalen Elemente des zusammengeführten Formulars entstanden sind.

3.1.2. Zweistufiges Cluster-Verfahren

Diese Arbeiten handeln von dem Werkzeug Wise-Integrator [HHYW05] [HHYW03]. Der Wise-Integrator kann mehrere Suchoberflächen aus derselben Kategorie, automatisch zu einer kombinierten Suchoberfläche zusammenführen. In bestehenden Arbeiten wird eine Schema Integration manuell oder semi-Automatisch durchgeführt. Im Vordergrund des Wise-Integrators steht hingegen ein maximaler Automatisierungsprozess. Der Prozessablauf wird im Folgendem erläutert.
Der WISE-Integrator besteht aus zwei Komponenten dem interface extractor und dem inteface integrator. Der interface extractor erhält als Eingabe HTML-Seiten von unterschiedlichen Quellen. Dieser unterteilt, mithilfe eines zweistufigen Cluster-Verfahrens, die einzelnen Beschreibungen und Elemente der verschiedenen Seiten in Cluster. Die Ausgabe ist ein Schema, welches diese Gruppierungen enthält. Der interface integrator erhält diese Schemas als Eingabe. Dieser erstellt für jede dieser Gruppen ein globales Objekt. Anschließend wird aus diesen globalen Objekten eine globale Suchoberfläche erstellt.

3.1.2.1. Ergebnisse

Tests zeigen, dass die Lösungen im Durchschnitt eine Genauigkeit von 95,25% und eine Vollständigkeit der Abbildungen von 97,91% besitzen.

3.1.2.2. Diskussion

Der Wise-Integrator zeigt wie aus Oberflächen von Suchwebseiten eine gemeinsame Oberfläche erstellt werden kann. Viele Ansätze, welche für die Erkennung semantischer Gleichheit und Zusammenführung verschiedener Elemente genutzt werden, können in dieser Bachelorarbeit wiederverwendet werden. Die Bachelorarbeit unterscheidet sich in der Erstellung eines Konstruktionsplanes für aktive Ontologien, welcher sich von einer gemeinsamen Suchoberfläche signifikant unterscheidet.

3.2. Zusammenführung von Schemas

In diesem Abschnitt wird eine Arbeit über Schema Abbildungen vorgestellt. In Kapitel 2.3 wurden Schemas als Beispiel von XML-Dokumenten bereits vorgestellt.
Mithilfe von Schema Abbildungen können verschiedene Schemas ineinander überführt und auf ein neues Schema abgebildet werden.

3.2.1. Generische Schema zusammenführung mit dem Werkzeug Cupid

Die Arbeit "Generic Schema Matching with Cupid" [JM01] handelt von dem Ansatz Cupid, welcher zwei Schemas generisch zusammenführen kann. Der Anpassungsalgorithmus basiert auf der Zusammenführung von hierarchischen Schemas, wie Schemabäumen. Dazu werden die einzelnen Schemas auf Bäume abgebildet. Im Folgenden wird der Anpassungsalgorithmus anhand von XML-Schemas demonstriert.
Die Zusammenführung geschieht in den Schritten linguistische Analyse, strukturelle Analyse und Generierung von Abbildungen.

3.2.1.1. Linguistische Analyse

Die linguistische Analyse erfolgt ebenfalls in 3 Schritten, Normalisierung, Kategorisierung, Vergleich.

Normalisierung

Die Normalisierung unterscheidet sich von der Normalisierung aus Kapitel 3.1.1.2 und wird durch folgende Schritte mithilfe eines Synonymwörterbuches realisiert.

- Tokenisierung: Alle Namen werden in einzelne Wörter unterteilt, welche Token bilden.
- Expansion: Abkürzungen und Kürzel werden zu vollständigen Wörtern erweitert
- Elimination: Alle Token welche Artikel, Präpositionen oder Konjunktionen sind werden markiert und während des weiteren Prozessablaufes ignoriert.
- Kennzeichnung: Alle Elemente, welche mit einem Begriff in Beziehung stehen werden mit diesem markiert, zusätzlich werden alle Token mit einer Nummer, einem speziellen Symbol, einer Präposition, einer Konjunktion, einem Begriff oder einem Inhalt markiert.

Kategorisierung

Um die Anzahl an Vergleichen zwischen Elementen zu reduzieren, werden diese in Kategorien eingeteilt. Die Vergleiche finden nur zwischen Elementen derselben Kategorien statt, jedes Element kann aber in mehreren Kategorien eingeordnet sein. Im Folgenden werden die Methoden aufgelistet, welche für die Erstellung der Kategorien verwendet werden.

- Begriffs Kennzeichnung: Jeder Begriff des Schemas erhält eine Kategorie.
- Datentypen: Jeder allgemeine Datentyp erhält eine Kategorie.
- Container: Jedes Element, welches andere Elemente enthält, erhält eine Kategorie.

Vergleich

Die Ähnlichkeiten der verschiedenen Elemente innerhalb der Kategorien werden bestimmt.

3.2.1.2. Strukturelle Analyse

In diesem Abschnitt wird der TreeMatch Algorithmus vorgestellt. Dieser kann ausschließlich hierarchische Schemas analysieren. Ein XML-Schema ist bereits ein hierarchisches Schema und benötigt keine weitere Bearbeitung.
Der TreeMatch Algorithmus vergleicht die verschiedenen Knoten des Baumes auf unterschiedliche Weise über ein Bottum-up Verfahren. Die Vergleiche werden im Folgendem vorgestellt.

Blätter

Blätter werden als ähnlich angesehen, wenn diese linguistisch gleich sind, der Datentyp gleich ist und die Nachbarelemente sich ähneln.

nicht-Blätter Elemente

Die Gleichheit wird anhand der linguistischen Ähnlichkeit und der Ähnlichkeit der Unterbäume bestimmt.

nicht-Blätter Schema Elemente

Zwei Schema Elemente sind strukturell gleich, wenn die Blätter sehr ähnlich sind.

3.2.1.3. Generierung von Abbildungen

Aus den Ergebnissen von der linguistischen und strukturellen Analyse werden Abbildungselemente erzeugt. Diese Abbildungselemente enthalten eine Liste von Elementen und Übereinstimmungen.

3.2.1.4. Diskussion

In dieser Arbeit wird ein Ansatz vorgestellt, welcher verschiedene Schemas auf ein gemeinsames Schema abbilden kann. Der Aufbau dieser Schemas kann dem von Aufbau von HTML-Formularen sehr ähnlich sein, dennoch gibt können signifikante logische Unterschiede bestehen. Ein Teil der linguistischen Analyse kann in dieser Arbeit übernommen, da sowohl in HTML-Formularen als auch in Schemata semantische Gleichheit über linguistische Verfahren ermittelt werden kann.

3.3. Ontology Merging

Der Ansatz Ontology Merging befasst sich mit dem Thema verschiedene Ontologien zu einer Ontology zusammenzuführen. Die folgenden Arbeiten befassen sich mit den Bereichen maschinelles Lernen, Zusammenführung von Ontologien mithilfe eines hierarchischen Cluster-Verfahrens und Zusammenführung von Ontologien mithilfe des Werkzeugs Ontobuilder.

3.3.1. Maschinelles Lernen

In der Arbeit „Learning to Discover Complex Mappings from Web Forms to Ontologies“ [YA12] werden Formulare automatisiert auf bestehende Ontologien abgebildet. Für die Umsetzung dieses Ziels wird ein maschinell lernender Ansatz verwendet. Der Prozessablauf wird im Folgendem erläutert.
Zunächst wird ein Formularbaum aus einem Eingabeformular erstellt. Anschließend werden die Ähnlichkeiten der einzelnen Elemente der Baumstruktur und der Ontologie ermittelt. Aus diesen Informationen kann schließlich die Abbildung des Formularbaumes auf die Ontologie bestimmt werden.

Mithilfe des Naiven Bayes Ansatzes werden die Ergebnisse über maschinelles Lernen verbessert. In mehr als 80% der Testfälle lieferte dieses Verfahren, im Vergleich zu weiteren Ansätzen, das beste Ergebnis.
Dieses Vorgehen gibt eine bestehende Ontologie als Eingabe vor. Das Erhalten einer Ontologie für das Werkzeug dieser Bachelorarbeit als Eingabe ist zu vermeiden, da dies den Automatisierungsprozess verringern würde. Aus diesem Grund ist das Verwenden von maschinellem Lernen nach dem oben genannten Vorgehen ungeeignet. Jedoch ist maschinelles Lernen für die Optimierung der Ergebnisse, mit der Verwendung andere Ansätze, als Ausblick denkbar.

3.3.2. Hierarchische Cluster-Verfahren

In der Arbeit „Automatic Ontology Merging by Hierarchical Clustering and Inference Mechanisms“ [MFBB10] werden verschiedene Ontologien zu einer globalen Ontologie zusammengeführt. Mithilfe eines hierarchischen Cluster-Verfahrens wird dieses Vorhaben umgesetzt.
Im Gegensatz zu vielen anderen Arbeiten, welche einen Grenzwert für den Abbildungsprozess verwenden und das Zusammenführen von ausschließlich 2 Ontologien ermöglichen, wird in dieser Arbeit ein anderer Ansatz verwendet. Dazu wird der Grenzwert durch eine intervenierende Variable ersetzt und ein skalierbarer Ansatz erstellt. Der Prozessablauf wird im Folgendem erläutert.
Zunächst werden Begriffsklassen mithilfe eines Cluster-Verfahrens erstellt. Als Nächstes werden die Informationen der dieser Klassen ausgewertet. Über diese Informationen werden die Klassen auf globale Begriffsklassen abgebildet. Zum Schluss wird das Ergebnis in eine hierarchische Struktur überführt, welche die neue Ontologie darstellt.
Diese Arbeit beschreibt die Abbildung von Ontologien auf eine globale Ontologie. Für diese Abbildungen werden Komponenten von Ontologien verglichen, welche sich von den Komponenten der Formulare sehr unterscheiden. Daher kann ein Großteil dieser Arbeit nicht für das Erfüllen der Bachelorarbeit in Betracht gezogen werden.

3.3.3. Ontobuilder

Der Ontobuilder ist ein Projekt, welches automatisch Ontologien aus HTML-Formularen erstellt [AG04]. Dieser kann die beste Abbildung von 2 Formularen finden. Der Ontobuilder enthält verschiedene Algorithmen, welche Terme aus verschiedenen Formularen zu einem globalen Term zusammenführen. Im Folgendem werden die Abbildungsverfahren erläutert, welche der Ontobuilder verwendet.

3.3.3.1. Syntaktische Angleichung

Bei dem syntaktischen Zusammenführen werden die syntaktischen Ähnlichkeiten der Formularelemente zweier Formulare bestimmt[AG05]. Es werden unterschiedliche Strategien für die Angleichung der Terme und Werte dieser Formulare verwendet.
Die Term Analyse ordnet die Beschreibungen ihren zugehörigen Eingabefeldern zu. Bei diesem Vorgang werden die Beschreibungen mit dem Namen der Eingabefelder verglichen. Um die Ähnlichkeit der Zeichenketten festzustellen, wird sowohl eine Wort- als auch eine Zeichenkettenanalyse angewendet. Die Werteanalyse vergleicht die Werte der Terme. Die Analysen werden im Folgenden kurz erläutert.

Wortanalyse

Bei dieser Analyse werden die einzelnen Wörter der Terme verglichen. Dieser Vorgang wird mit öffentlichen Synonymwörterbüchern unterstützt. Die Ähnlichkeit dieser Analyse entsteht aus der Schnittmenge der gemeinsamen Wörter dieses Vergleichs.

Zeichenkettenanalyse

Bei dieser Analyse werden vorerst die Leerzeichen zwischen einzelnen Wörtern entfernt. Die entstehenden Zeichenfolgen werden anschließend auf Teilwörter untersucht. Die Größe dieser Teilwörter gibt die Ähnlichkeit der beiden Zeichenfolgen an.

Werteanalyse

Die Werteanalyse vergleicht die Wertesets der Eingabefelder. Anhand dieses Vergleiches wird eine Ähnlichkeit festgelegt.

Für diese drei Analysen wird ein bestimmter Grenzwert, festgelegt. Befindet sich die Ähnlichkeit über diesem Grenzwert sind die beiden Terme, in Bezug auf diese Analyse, ähnlich.

3.3.3.2. Strukturelle Analyse

Die strukturelle Analyse betrachtet die Reihenfolge von Termen innerhalb der Formulare, sowie die Reihenfolge von Termen innerhalb hintereinander folgender Formulare und bestimmt deren Ähnlichkeit. Da semantisch gleiche Terme sich häufig an ähnlichen Positionen in deren Formularen befinden, kann durch dieses Verfahren eine Verbesserung der semantischen Analyse erzielt werden.

3.3.3.3. Diskussion

Die Analysen des Ontobuilders zeigen, wie semantisch gleiche Formularelemente zweier Formulare entdeckt werden können. Dies eignet aber sich nicht für die Verwendung in dieser Bachelorarbeit. Es müssen nicht nur 2, sondern beliebig viele Formulare ineinander überführt werden. Dennoch können Teile der Analysen für die Umsetzung eigenen Lösungen wiederverwendet werden.

3.4. Zusammenfassung

In diesem Abschnitt wurden verschiedene Abbildungsansätze aus den Bereichen Deep Web, Schema Zusammenführung und Ontology Merging vorgestellt. Innerhalb dieser Arbeiten wurden deren Ergebnisse evaluiert und in einer Diskussion für die Nützlichkeit mit dieser Bachelorarbeit verglichen und bewertet. Relevante Ansätze und Lösungen dieser Arbeiten werden den folgenden Kapiteln, für die Erstellung des Werkzeugs dieser Arbeit, wiederverwendet werden.

4. Analyse

In dieser Arbeit geht es um die Konsolidierung von Webformularen für die Erstellung von aktiven Ontologien. Das Problem dabei ist das Abbilden semantisch gleicher Formularelemente aus verschiedenen Formularen der gleichen Dienstkategorie zu globalen Elementen. Aus diesen globalen Elementen kann schließlich ein Konstruktionsplan erstellt werden.
Für den Lösungsansatz werden zunächst Objekte aus den Formularen erstellt, welche alle HTML-Elemente der Formulare beinhalten. Dies ist notwendig um die einzelnen HTML-Elemente besser miteinander vergleichen und die semantischen Ähnlichkeiten bestimmen zu können. Als Nächstes werden diese Objekte verwendet um semantische Beziehungen zwischen den einzelnen HTML-Elementen aus unterschiedlichen Formularen zu bestimmen. Die semantische Beziehungen werden dazu verwendet um alle semantisch gleiche Elemente in Gruppen zu unterteilen. Dieser Schritt wird mithilfe eines Cluster-Verfahrens umgesetzt. Aus diesen Gruppen werden anschließend die globalen Elemente generiert.
Für die Umsetzung werden die Lösungen der verwandten Arbeiten wiederverwendet und mit den eigenen Lösungen kombiniert. Das Werkzeug verwendet als Eingabe mehrere Formulare derselben Dienstkategorie. Die Ausgabe ist ein globaler Konstruktionsplan in XML-Format, aus welchem eine aktive Ontologie erstellt werden kann. In Abbildung 4.1 wird der Prozessablauf für die Extraktion und Konsolidierung der Formulare dargestellt. Dieser kann in vier große Bereiche unterteilt werden.
Der erste Abschnitt handelt von der Erstellung lokaler Objekte, welche aus der Eingabe extrahiert werden. Hierfür werden zunächst alle Eingabefelder der Formulare bestimmt und mit ihren Beschreibungen zu Termen zusammengeführt. Diese werden anschließend in eine hierarchische Baumstruktur überführt. Zum Schluss erarbeitet die Normalisierung wichtige Informationen aus den Texten der Felder, um spätere Vergleiche zu erleichtern.
Der zweiten Abschnitt ist die semantische Analyse. Diese wird in drei Schritten durchgeführt. Zuerst findet eine linguistische Analyse statt, welche die bereits behandelten Texte vergleicht und versucht semantische Ähnlichkeiten zu erkennen. Anschließend wird eine Analyse der Eingabetypen und Eingabewerte der Terme durchgeführt und deren Ähnlichkeiten bestimmt. Schließlich findet die strukturelle Analyse statt, welche semantische Ähnlichkeiten anhand der Positionen in den Formularen erkennen kann.
Das Bestimmen der semantischen Gleichheit ist der nächste Schritt. Hier werden die semantisch gleichen Elemente anhand der in der Analyse festgestellten Ähnlichkeiten, mithilfe eines Cluster Verfahrens, zusammengeführt.
In dem letzten Abschnitt wird die Ausgabe generiert. Dazu werden zuerst globale Objekte erstellt. Dies geschieht über das Festlegen eines globalen Wertebereiches, Attributen, Typen und Namen. Anschließend wird aus diesen globalen Objekten ein Konstruktionsplan

in XML-Format erstellt.

4.1. Erstellung von Termen

Die Eingabe des zu erstellenden Werkzeugs ist eine Menge von HTML-Formularen derselben Kategorie. Um aus diesem einen Konstruktionsplan zu erstellen, müssen semantisch gleiche Formularelemente aufeinander abgebildet werden. Um möglichst präzise Ähnlichkeiten bestimmen zu können, ist es wichtig sowohl die Beschreibungen als auch die Eingabefelder in die Vergleiche miteinzubeziehen. Für dieses Vorhaben wird ein in den Arbeiten [HHYW05], [AG05] vorgestelltes Konzept verwendet.
Zunächst werden die HTML-Formulare in ihre Elemente unterteilt[HHYW05]. Über das „for“ Attribut oder Verschachtelte Elemente werden die verschiedenen Eingabefelder den Beschreibungen zugeordnet[AG05][HHYW05]. Das Ergebnis sind Terme, welche jeweils ein Eingabefeld und eine Beschreibung besitzen. In dem nächsten Schritt werden Optionsfelder, Auswahlmenüs, Kontrollkästchen und andere Felder, welche Optionen darstellen zusammengeführt. Optionsfelder und Kontrollkästchen werden dazu zu einem Term verschmolzen. Die restlichen Optionsfelder werden als Optionen in ihren zugeordneten Term eingesetzt.
Zusätzlich werden alle „value“ Attribute der Terme entfernt und ebenfalls als Option eingesetzt. Die aktive Ontologie, welche aus dem Konstruktionsplan entsteht, erhält dadurch eine Verbesserung der natürlichen Sprachverarbeitung. Alle Optionen des globalen Objektes werden von der aktiven Ontologie als mögliche Eingabe für die Spracherkennung verwendet. Mithilfe dieses Vorgangs wird eine zusätzliche mögliche Eingabe für die aktive Ontologie bereitgestellt.

4.2. Erstellung einer hierarchischen Struktur

Für eine strukturelle Analyse kann es hilfreich sein die Formulare als eine hierarchisch Baumstruktur darzustellen [JM01][HHYW05]. Die Terme bilden die einzelnen Knoten des Baumes. Diese werden dazu auf verschachtelte Elemente überprüft[HHYW05]. Verschachtelte Elemente sind jeweils die Kindknoten der sie umhüllenden Elemente. Das Formular wird auf den Baum in derart abgebildet, dass die Position der Terme in dem Formular rekonstruierbar ist. Die Terme werden von links nach rechts in dem Baum positioniert, während diese äquivalent dazu in einem original Formular von oben nach unten positioniert wurden. Der Wurzelknoten ist das Formularelement, welches alle Formularelemente umgibt. In Abbildung 4.2 wird ein Beispiel für eine solche Abbildung dargestellt.
Um spätere Vergleiche zu erleichtern, werden die Terme dieser Bäume anschließend überarbeitet. Es existieren verschiedene Formularelemente-Typen, welche semantisch gleich sind. Diese differenzieren sich durch unterschiedliche Attribute und werden von einem Browser unterschiedlich dargestellt. Um die Vergleiche zwischen den Typen zu erleichtern, werden die semantisch gleichen Typen in Gruppen eingeordnet. Dazu wird ein Attribut namens „semType“ eingeführt. Der Wert des Attributes repräsentiert die Eigenschaften, den die Typen dieser Gruppe besitzen. Um zusätzlich noch die Darstellung der ursprünglichen Typen zu speichern, wird das Attribut „disType“ eingeführt. Der Wert dieses Attributes beinhaltet den Typen des HTML-Elementes.
Als Beispiel eines semantischen Typs besitzen sowohl eine Auswahlliste als auch Kontrollkästchen den Typen „list“. Anstatt für beide dieser Typen spezielle Regeln für einen Vergleich zu erstellen, werden diese Mithilfe des Attributes „semType“, als gleich angesehen. Die neu zusammengeführten Typen sind in Tabelle 4.1 aufgelistet. Die Knoten der Bäume werden mit den in Tabelle 4.2 sich befindenden Komponenten definiert, von denen nicht alle verwendet werden müssen.

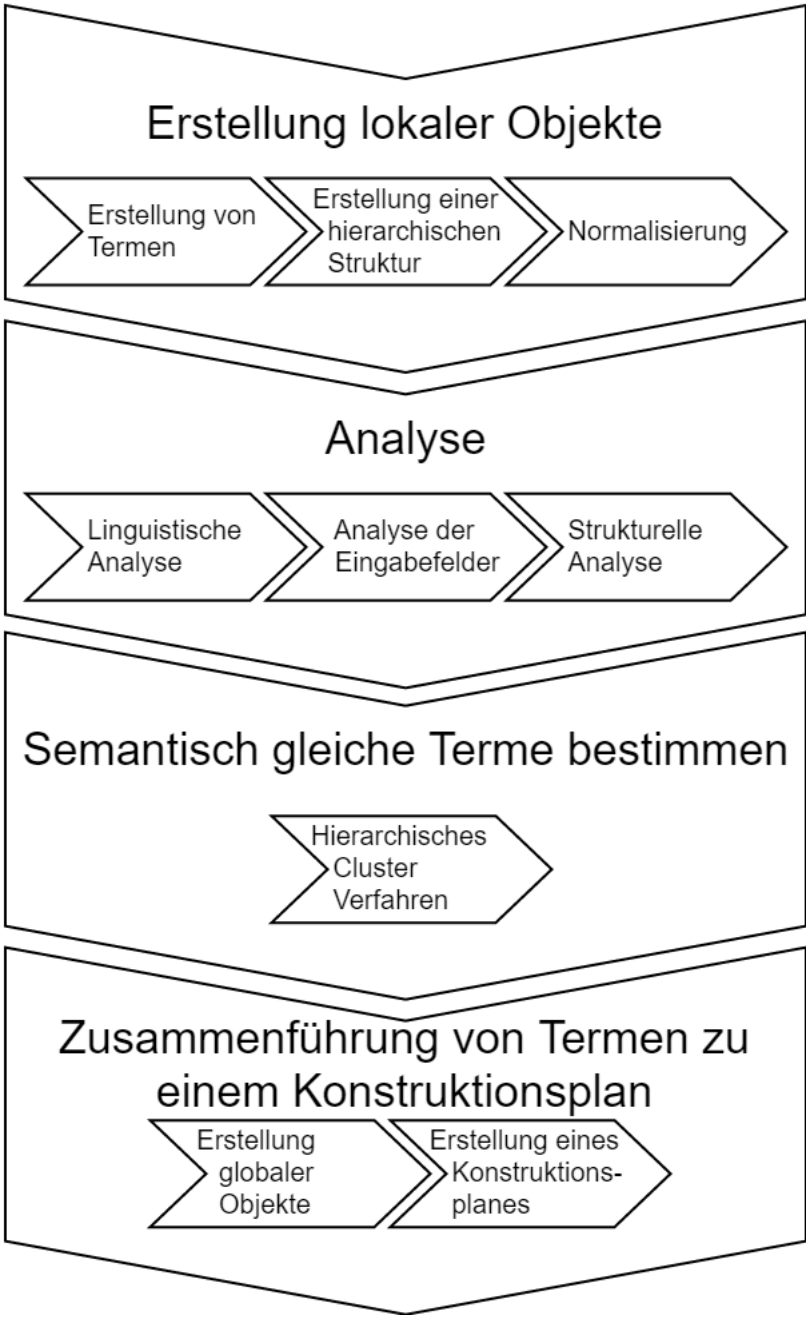

Abbildung 4.1.: Prozessablauf für das zu erstellende Werkzeug

Neuer Typ	**Formularelemente**
button	<input type=”button”>, <input type=”image”>, <button type=”button”>
list	<select> + <option>, <input type=”radio”>, <input type=”checkbox”>
number	<input type=”number”>, <input type=”range”>
text	<input type=”text”>, <textarea>, <input> + <datalist>

Tabelle 4.1.: Neue Erstellung von Typen für die Vereinfachung der Vergleiche

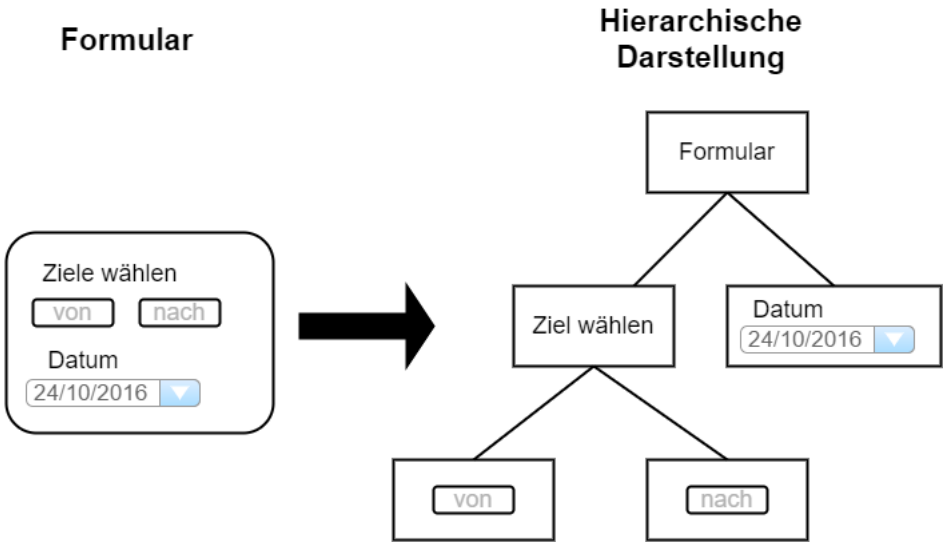

Abbildung 4.2.: Abbildung von einem Formular zu einer Baumstruktur

Komponente	**Beschreibung**
disType	Dieses Attribut gibt den Typ des HTML-Eingabefeldes an. Die Darstellung des Eingabefeldes in HTML wird berücksichtigt.
id	Ist eine Liste von ID's.
label	Gibt den Inhalt des Labels an.
max und min	Diese Komponenten legen den Wertebereich fest.
maxlength	Legt die Maximale Anzahl an Zeichen fest, welche Eingegeben werden darf.
multiple	Gibt an ob mehrere Eingaben möglich sind.
name	Gibt den Namen des Terms an.
pattern	Gibt den regulären Ausdruck an, mit welchem das Eingabefeld überprüft wird.
placeholder	Gibt den Text eines Platzhalters an.
option	Gibt die Auswahlmöglichkeiten, falls vorhanden, des Terms an.
required	Gibt an ob der Term für das abgeben des Formulares ausgefüllt werden muss.
step	Legt das Intervall fest, welches den Abstand zwischen möglichen Eingabewerten bestimmt.
semType	Gibt den semantischen Typ des Terms an. Die Darstellung des Eingabefeldes in HTML ist anhand dieses Typen nicht immer nachvollziehbar. Beispiele hierfür stehen in Tabelle 4.1.

Tabelle 4.2.: Komponenten eines Baumknotens

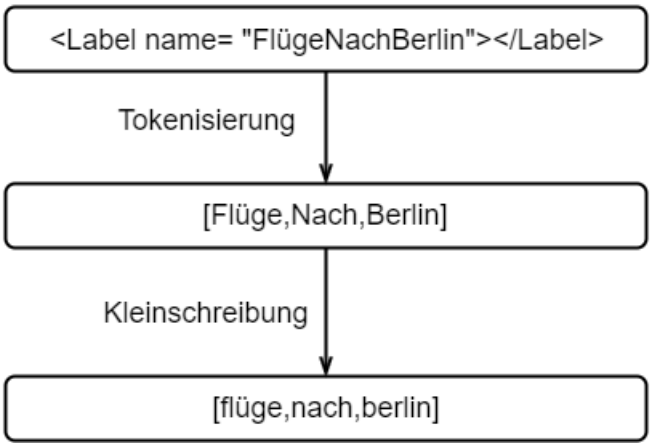

Abbildung 4.3.: Beispiel einer Normalisierung

4.3. Normalisierung

Um spätere Vergleiche zu erleichtern, werden die verschiedenen Texte, welche in den Termen enthalten, sind einer Vorbehandlung unterzogen. Mithilfe einer Normalisierung werden die Darstellungen der Inhalte dieser Texte angepasst und die Ansätze aus den Arbeiten [JM01], [WYDM04] und [AG05] verwendet.
Dazu wird eine Tokenisierung durchgeführt [JM01][WYDM04][AG05]. Ein Beispiel einer Tokenisierung ist in Abbildung 4.3 zu sehen. Bei einer Tokenisierung werden einzelne Zeichenketten, welche aus mehreren Wörtern bestehen zu Token aus einzelnen Wörtern umgeformt. Dies geschieht durch das Trennen der Zeichenketten an Großbuchstaben, nicht alphanumerischen Zeichen und Zahlen. Anschließend werden die Wörter der Token in die Kleinschreibung umgeformt, sodass die Groß- und Kleinschreibung die Vergleiche der Token nicht beeinflussen kann.
Dieser Ablauf findet für Namen, Beschreibungen, Optionsfelder, Platzhalter und IDs getrennt statt, sodass für jede dieser Zeichenketten ein Tupel entsteht. Das Ergebnis ist eine Menge von Tupeln, welche über ihre Wörter miteinander verglichen werden können.
Zusätzlich werden die Leerzeichen aus den ursprünglichen Texten der Beschreibungen entfernt und für spätere Zeichenkettenvergleiche beigefügt[AG05].

4.4. Linguistische Analyse

In der linguistischen Analyse werden Vergleiche über die linguistischen Daten der Terme durchgeführt. Es wird versucht möglichst viele Informationen aus diesen Daten zu berücksichtigen, um möglichst genaue Ergebnisse aus dieser Analyse zu erzielen. Dazu werden die Token, Zeichenketten und Eingabetypen der Terme verglichen. Es wurden Verfahren und Ansätze der Arbeiten [AG05], [WYDM04] und [JM01] für diesen Abschnitt übernommen. Im Folgenden werden die unterschiedlichen Vergleiche erläutert.

Tokengleichheit

In diesem Schritt wird das Verfahren aus [S. 4][WYDM04] verwendet. Bei dem Vergleich zweier Terme werden jeweils zwei Token aus beiden Termen verglichen. Dies wird drei Mal durchgeführt. Dazu wird jede mögliche Kombination aus ID, Beschreibung und Platzhalter der Terme als Eingabe verwendet. Das höchste Ergebnis dieser drei Vergleiche wird als Referenzwert der Tokengleichheit verwendet. Sei die Funktion Ver(a,b) die Durchführung eines Vergleiches von a und b, mit dem Ergebnis

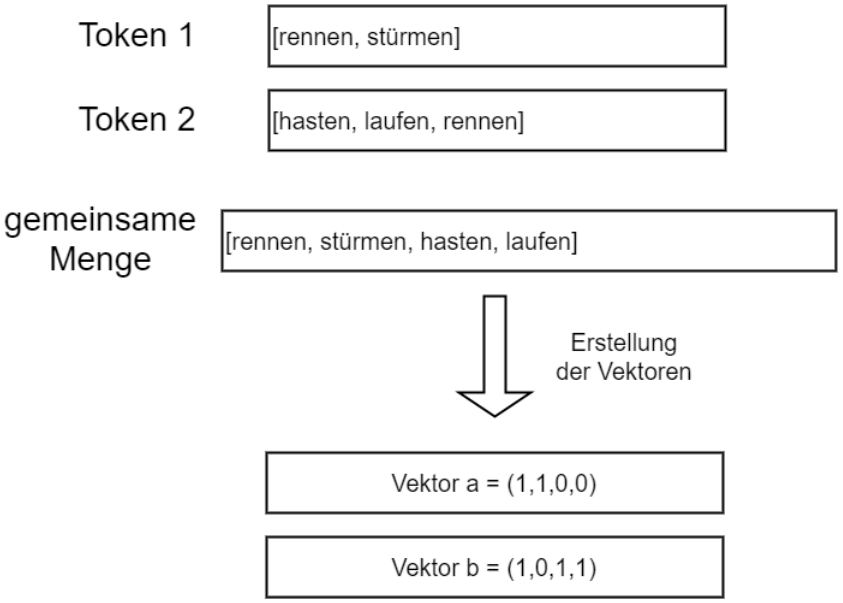

Abbildung 4.4.: Erzeugung von Vektoren aus Token

des Vergleiches als Ausgabe und Res das Ergebnis der Tokenanalyse, so gilt das im Folgenden beschriebene Verfahren.

$$\begin{aligned} \boxed{max1 = max(Ver(ID, Platzhalter), Ver(ID, Namen))} \\ \boxed{Res = max(max1, Ver(Platzhalter, Namen))} \end{aligned} \tag{4.1}$$

Wobei die Ausgabe der Funktion max(a,b) den größten Wert der Eingaben a und b liefert.
Zur Veranschaulichung wird der Vergleich zwischen ID und Namen erläutert. Dazu werden zunächst die Token der ID und Namen, der beiden Terme, getrennt verglichen. In zwei weiteren Vergleichen wird anschließend jeweils der Name des einen Terms mit der ID des anderen Terms verglichen.
Im Folgenden wird die sogenannte Cosinus Funktion(Cos(a,b)) erläutert [WYDM04]. Die Eingabe der Cosinus Funktion sind zwei Vektoren, welche aus zwei Token erstellt werden. Dazu werden die Tokeneinträge zunächst zu einer Menge zusammengefasst, die resultierende Menge wird im folgendem M genannt mit $M = (w_1, w_2, ..., w_n)$. Wobei n die Anzahl der Elemente in M ist und kein Element in M doppelt enthalten ist. Für Vektor a aus Cos(a,b) gilt:

$$a = (w_1, w_2, ..., w_n) \tag{4.2}$$

Wobei $w_i = 1$, wenn Token 1 ein Wort aus m_i in einem Tokeneintrag besitzt. Andernfalls gilt $w_i = 0$. Dieses Vorgehen wird mit der Levenshtein Distanz[1] unterstützt. Äquivalent wird der Vektor b von Token 2 erstellt. Dieser Ablauf wird an dem Beispiel von Abbildung 4.4 demonstriert. Die beiden Vektoren werden als Eingabe der Cosinus Funktion verwendet. Diese teilt das Skalarprodukt der beiden Vektoren durch das Produkt ihrer Normen. Die Formel wird im Folgendem dargestellt[WYDM04].

$$\boxed{Cos(a, b) = \frac{a \bullet b}{||a|| * ||b||}} \tag{4.3}$$

[1]Levenshtein Distanz, der verwendete Algorithmus für die Implementierung ist Online erhältlich unter https://rosettacode.org/wiki/Levenshtein_distance; abgerufen am 07. Januar 2017

Als Beispiel wird eine Berechnung der Cosinus Funktion mit den Vektoren aus Abbildung 4.4 als Eingabe im Folgendem durchgeführt.

$$\begin{aligned} a \bullet b &= 1*1+1*0+0*1+0*1=1 \\ ||a|| &= \sqrt{1^2+1^2+0^2+0^2}=\sqrt{2} \\ ||b|| &= \sqrt{1^2+0^2+1^2+1^2}=\sqrt{3} \\ Cos(a,b) &= \frac{a \bullet b}{||a|| * ||b||} = \frac{1}{\sqrt{2}*\sqrt{3}} \approx 0,3178 \end{aligned} \tag{4.4}$$

Sei Vr das Ergebnis des Vergleiches zwischen Namen und ID, Tn_i das Token für den Namen aus Term i, Tl_i das Token für das Label aus Term i und T_n, T_l und T_{nl} die Parameter für die Gewichtung, so gilt für den Vergleich zwischen zwei Termen [WYDM04]:

$$\boxed{\begin{aligned} Vr = T_n * Cos(Tn_1, Tn_2) + T_l * Cos(Tl_1, Tl_2) + \\ T_{nl} * max(Cos(Tn_1, Tn_2), Cos(Tl_1, Tl_2)) \end{aligned}} \tag{4.5}$$

Dabei haben die Parameter für die Gewichtung die Werte 0 bis 100 und bilden in der Summe den Wert 100. Die Parameterwerte T_n und T_l besitzen in dem zu erstellenden Werkzeug die gleichen Werte, um die Parameterfindung der Evaluation zu vereinfachen und die Eingabe möglicher Parameterkombinationen zu verringern. Die Ausgabe der Funktion max(a,b) ist der größte Wert der Eingaben a und b.
Die anderen beiden Vergleiche funktionieren äquivalent.

Zeichenkettengleichheit

Im Folgenden wird das Verfahren aus [AG05] verwendet. Bei diesem Vergleich wird nach möglichst langen gleichen Teilzeichenketten zweier Terme gesucht. Die Ähnlichkeit ist die längste Folge gemeinsamer Buchstaben durch die Anzahl aller Buchstaben. Sei Zkf die Zeichenkettengleichheit zweier Zeichenfolgen, Bg die längste Folge gemeinsamer Buchstaben und Ba die Anzahl aller Buchstaben zweier Zeichenketten so gilt [AG05]:

$$\boxed{Zkf = \frac{Bg}{Ba}} \tag{4.6}$$

Dieser Vergleich wird für alle möglichen Kombinationen zwischen Beschreibung, ID, Namen und Platzhalter durchgeführt. Das größte Zkf Ergebnis, welches bei diesen Vergleichen zwischen 2 Termen entsteht, wird mit 100 multipliziert und als Zeichenkettengleichheit gewählt. Dieses Vorgehen wurde gewählt, damit geringe Zkf Ergebnisse die Teilzeichenähnlichkeit nicht verschlechtern können.

Vergleich der Eingabefelder

In diesem Vergleich werden zwei Eingabefelder aus Termen unterschiedlicher Formulare verglichen. Hierfür wird ein Verfahren aus [S. 4,5][WYDM04] übernommen. Es werden sowohl die Typen der Eingabefelder als auch die gültigen Eingabewerte verglichen. Sei Wt die Ähnlichkeit des Wertevergleichs, Ts die Ähnlichkeit der Typen der Eingabefelder und Ew die Ähnlichkeit gültiger Eingabewerte, so gilt [WYDM04]:

$$\boxed{Wt = P_{Ts} * Ts + P_{Ew} * Ew} \tag{4.7}$$

Die Parameter P_{Ts} und P_{Ew} werden für die Gewichtung der Ähnlichkeiten verwendet. Ihr Wertebereich liegt zwischen 0 und 100 und die Summe der beide Werte ist 100. Der Parameter Ts ist 1, wenn die beiden Typen der Eingabefelder gleich sind, andernfalls ist Ts gleich 0. Für den Wert Ew werden verschiedene Vergleiche für die unterschiedlichen Typen verwendet. Diese werden im Folgendem erläutert.

Vergleich zwischen zwei Listen

Bei diesem Vergleich wird die Cosinus Funktion aus der Tokenanalyse verwendet [WYDM04]. Dieses Verfahren wurde aus der Arbeit [WYDM04] wiederverwendet und basiert darauf Elemente, welche sich in beiden Auswahlmenüs der zu vergleichenden Terme befinden, zu bestimmen. Alle Elemente der beiden Listen werden verglichen, wobei nur Elemente mit Elementen der anderen Liste verglichen werden dürfen. Für jeden Vergleich werden die beiden Elemente in die Cosinus Funktion eingesetzt. Wurden alle Vergleiche durchgeführt, wird eine leere Menge C erstellt. Anschließend wird jeweils das Paar mit der größten Ähnlichkeit entfernt und zu dieser Menge hinzugefügt. Dieser Vorgang wiederholt sich so oft, bis alle Ähnlichkeiten unter einem Bestimmen Grenzwert liegen. Der Grenzwert legt fest bis, zu welcher Ähnlichkeit die Listenelemente als gleich angesehen werden können. Die Kardinalität von C ist damit die Summe aller Elementpaare, welche semantisch gleich sind. Um das ungleiche Verhältnis zwischen der Summe der semantisch gleichen Elementpaare und der Summe der Elemente der beiden Listen auszugleichen, wird die Kardinalität von C in der folgenden Gleichung mit zwei multipliziert. Sei Me die Summe der Kardinalitäten der Listen der beiden Eingabefelder, so gilt [WYDM04]:

$$\boxed{Ew = \frac{2 * |C|}{Me}} \tag{4.8}$$

Ew enthält den Prozentsatz der semantisch gleichen Elemente der beiden Listen.

Vergleich zwischen zwei numerischen Wertebereichen

Dieser Vergleich bestimmt die Ähnlichkeit anhand des überschneidenden Wertebereiches, wobei Eingabeintervalle nicht berücksichtigt werden[WYDM04]. Das Ergebnis ist der prozentuale Anteil der Schnittmenge der beiden Wertebereiche. Sei E_i der Wertebereich von Eingabefeld i, so gilt[WYDM04]:

$$\boxed{Ew = \frac{min(max(E_1), max(E_2)) - max(min(E_1), min(E_2)}{max(max(E_1), max(E_2)) - min(min(E_1), min(E_2)}} \tag{4.9}$$

Ist der Nenner gleich 0 oder sinkt die Ähnlichkeit unter 0, wird Ew gleich 0 gesetzt.

Sonstige Vergleiche

Ein Vergleich zwischen zwei Feldern, welche als logischen Typen „list“, „date“, „datetime“, „datetime-local“, „month“, „time“, „week“, „email“, oder „url“ besitzen, hat einen Ew von 1, wenn der Typ der beiden zudem identisch ist[WYDM04]. Elemente mit dem logischen Typ „text“ erhalten die Werteähnlichkeit 0, da die Eingabewerte mit den Werten aller Eingabefelder übereinstimmen können[WYDM04].

Aus den vorgestellten Verfahren wird die linguistische Ähnlichkeit nach dem Verfahren aus [WYDM04] bestimmt. Seien Tk die Token Ähnlichkeit, Zk die Zeichenkettengleichheit

und Wt die Ähnlichkeit des Werte Vergleichs, wird die linguistische Ähnlichkeit Lng wie folgt erstellt:

$$\boxed{Lng = P_{Tk} * Tk + P_{Zk} * Zk + P_{Wt} * Wt} \tag{4.10}$$

Mit den Parametern P_{Tk}, P_{Zk} und P_{Wt} können die einzelnen Ähnlichkeiten unterschiedlich gewichtet werden. Die Gewichtungen liegen zwischen 0 und 1 und ergeben in der Summe 1. Das Ergebnis Lng erhält mit dieser Methode einen Wert zwischen 0 und 100.

4.5. Strukturelle Analyse

Die strukturelle Analyse findet nach der linguistischen Analyse statt [AG05][JM01][WYDM04]. Ziel dieser Analyse ist es semantische Ähnlichkeiten anhand ihrer Position in ihrem Formular und semantisch gleicher Nachbarelemente zu bestimmen [AG05][JM01][WYDM04]. In dieser Analyse werden die Nachbarterme eines jeden Terms analysiert. Sind sich diese von zwei unterschiedlichen Termen semantisch ähnlich, ist es wahrscheinlich, dass die Nachbarterme der zu vergleichenden Terme semantisch ähnlich sind.
Diese Analyse verfolgt den Ansatz, dass die Wichtigkeit der Nachbarn für die Bestimmung der Ähnlichkeit innerhalb des Formulars exponentiell abnimmt. Sei Sk die strukturelle Ähnlichkeit zwischen zwei Termen, so existieren die Mengen T1o, T1u, T2o und T2u, wobei T1o die Menge aller oberen Nachbarn von Term1, T1u die Menge aller unteren Nachbarn von Term 1, T2o die Menge aller oberen Nachbarn von Term 2 und T2u die Menge aller unteren Nachbarn von Term 2 ist. Es gilt folgende Formel:

$$\boxed{Sk = 50 * Over(T1o, T2o) + 50 * Uver(T1u, T2u)}$$

$$\begin{aligned} Over(T1o, T2o) = \frac{1}{4 * 2^0} * An(T1o_0, T2o_0) + \frac{1}{4 * 2^1} * An(T1o_1, T2o_1) + ... \\ + \frac{1}{4 * 2^{15}} * An(T1o_{15}, T2o_{15}) \end{aligned} \tag{4.11}$$

$$\begin{aligned} Uver(T1u, T2u) = \frac{1}{4 * 2^0} * An(T1u_0, T2u_0) + \frac{1}{4 * 2^1} * An(T1u_1, T2u_1) + ... \\ + \frac{1}{4 * 2^{15}} * An(T1u_{15}, T2u_{15}) \end{aligned}$$

Wobei An(a,b) 1 ausgibt, wenn die Ähnlichkeit der linguistischen Analyse der beiden Einträge über einem gewissen Grenzwert liegt. Andernfalls ist das Ergebnis von An(a,b) gleich 0. Für die Indizes der Nachbarmengen gilt, je kleiner die Zahl, desto näher befindet sich der Nachbarterm an dem zu analysierenden Term. Dieses Verfahren wird abgebrochen, wenn einer der Parameter, welcher für die Gewichtungen zuständig ist, den Wert $\frac{1}{4*2^{15}}$ erreicht hat. Die Ergebnisse der restlichen Vergleiche sind aufgrund der hinreichend geringen Gewichtungen in Hinsicht auf die strukturelle Ähnlichkeit zu vernachlässigen.
Sollten zwei Nachbarmengen nicht dieselbe Anzahl an Elementen beinhalten, wird für jede Befüllung der Funktion An(a,b) von nur einem Element der Wert 0 ausgegeben. Kann die Funktion von beiden Mengen nicht befüllt werden, wird der Wert 1 ausgegeben. Der Grund für dieses Vorgehen ist die Betrachtung der strukturellen Ähnlichkeit in Hinblick auf den Abstand zu dem Anfang und dem Ende des Formulars. Die Resultate der beiden Funktionen Over(a,b) und Uver(a,b) werden jeweils mit der Gewichtung 50 multipliziert. Die Ergebnisse dieser Analyse liegen zwischen 0 und 100.

4.6. Das hierarchische Cluster Verfahren

In diesem Kapitel wird das in [WYDM04] vorgestellte Cluster-Verfahren, welches einen Greedy Ansatz verfolgt, präsentiert. Dazu werden semantisch gleiche Elemente über dieses

Verfahren identifiziert. Alle Terme welche den Typ „button“, „color“, „search“, „password“ oder das Attribut „readonly“ besitzen werden nicht weiter für die Erstellung des Konstruktionsplanes betrachtet. Das Gleiche gilt für Blockelemente und Darstellungselemente. Diese Terme sind für die Generierung einer aktiven Ontologie nicht relevant.
Die Initialen Cluster bilden die einzelnen Terme. Zunächst werden die Ergebnisse der linguistischen und der strukturellen Analyse für die Erstellung der finalen Ähnlichkeiten zusammengeführt. Sei Lng das Ergebnis der linguistischen und Sk das Ergebnis der strukturellen Analyse, so gilt Folgendes [WYDM04]:

$$\boxed{Tr = P_{Lng} * Lng + P_{Sk} * Sk} \tag{4.12}$$

Tr ist die Ähnlichkeit zwischen zwei Termen, welche während des Cluster-Verfahrens betrachtet wird. Die Parameter P_{Lng} und P_{Sk} sind für unterschiedliche Gewichtungen der Eingaben zuständig. Diese liegen wieder zwischen 0 und 1 und ergeben in der Summe 1.
Das Cluster-Verfahren wählt jeweils die höchste nicht zugewiesene Ähnlichkeit aus und ordnet diese einem Cluster zu. Eine Ähnlichkeit wird einem Cluster zugeordnet, wenn mindestens ein Term der Ähnlichkeit bereits in dem Cluster vorhanden ist und das Cluster auch keinen Term desselben Formulars von einem der Terme der Ähnlichkeit enthält. Wurde eine Ähnlichkeit gefunden, in dem sich beide Elemente bereits in verschiedenen Clustern befinden, werden die Cluster verschmolzen, wenn sich in dem neu entstehenden Cluster nicht mehr als ein Element von jedem Anbieter befindet. Ähnlichkeiten, welche keinem Cluster zugewiesen werden können bilden neue Cluster, wenn keiner der Terme sich bereits in einem Cluster befindet, andernfalls wird die Ähnlichkeit verworfen.
Dieses Verfahren gewährleistet Transitivität. Element A hat keine Ähnlichkeit mit Element B. Beide befinden sich aber in demselben Cluster, da diese mit Element C ähnlich sind.
Besitzt mehr als ein Wert die höchste Ähnlichkeit, werden zuerst 2 zufällige Formulare, welche mindestens einen dieser Terme besitzen gewählt. Von diesen Formularen wird jeweils die Ähnlichkeit gewählt, deren Terme sich am weitesten oben in den Formularen befindet, bis alle Ähnlichkeiten der beiden Formulare abgehandelt sind. Anschließend werden wieder 2 Formulare gewählt und der Vorgang wiederholt, bis keine dieser Ähnlichkeiten mehr vorhanden ist. Der Grund für dieses Vorgehen ist die ähnliche Anordnung semantisch gleicher Elemente in verschiedenen Formularen.
Ein Grenzwert legt fest, welche Ähnlichkeiten hinreichend gering sind, um einen Abbruch des Cluster-Verfahrens zu erzwingen.
In Abbildung 4.5 aus [WYDM04] sind die Ergebnisse zwischen einem „Max Cardinality“ Verfahren und dem Greedy Verfahren, welches in dieser Arbeit verwendet wird, dargestellt. In diesem Beispiel ist zu sehen, dass der Greedy Algorithmus nicht die beste Kardinalität erreicht, im Gegensatz dazu aber der durchschnittliche Wert größer ist. Der Greedy Algorithmus erreicht einen Durchschnittswert von 0,85 und eine Kardinalität von 1,7. Das zweite Verfahren erhält einen Durchschnittswert von 0,75 und eine Kardinalität von 2,25. Das Problem dieses Beispiels ist die Wahl eines Verfahrens für die Bestimmung von geeigneten Werte von einem Cluster. In dem Zitat aus der Arbeit „An Interactive Clustering-based Approach to Integrating Source Query Interfaces on the Deep Web“ [WYDM04], aus welchem der Greedy Algorithmus entnommen wurde, wird die Frage auf die Lösung des Problems mit der Wahl des Greedy-Verfahrens beantwortet.

> „*The perfectionist egalitarian polygamy selection metric (that is, no male or female is willing to accept any partner(s) but the best) produces best results in a variety of schema matching tasks. The greedy choice step of the clustering process for the identification of 1:1 mappings can be regarded as the monogamy version of this metric.*“

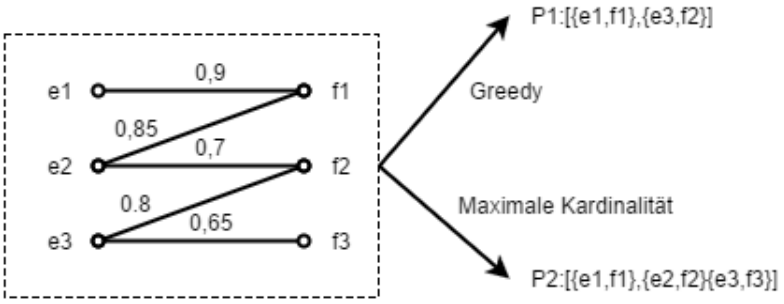

Abbildung 4.5.: Beispiel von unterschiedlichen Ergebnissen zwischen zwei unterschiedlichen Verfahren. Diese Abbildung ist aus der Arbeit von [WYDM04].

4.7. Erstellung globaler Objekte

Globale Objekte sind Objekte aus welchen die Knoten der aktiven Ontologien erstellt werden. Diese werden auch dem Konstruktionsplan beigefügt. Semantisch gleichen Elemente aus verschiedenen HTML-Formularen werden jeweils zu einem globalen Objekt zusammengeführt. Die Gruppen der semantisch gleichen Elemente wurden über das Cluster-Verfahren bestimmt. In diesem Kapitel werden die Abbildungsvorschriften für die Generierung eines globalen Elementes vorgestellt und erläutert.

4.7.1. Globale Typen

Zunächst wird der Typ des globalen Objektes bestimmt. Für dieses Vorgehen wird ein Verfahren aus [HHYW03] verwendet. Hierfür werden drei allgemeine Typen erstellt.

finite

Ein Term mit diesem allgemeinen Typ besitzt verschiedene Optionen, zwischen denen gewählt werden kann. Alle Terme, welche den logischen Typ „list“ besitzen, werden diesem zugeordnet.

infinite

Dieser allgemeine Typ kann eine beliebige Eingabe erhalten. Alle Terme mit dem logischen Typ „text“ und ohne das gesetzte „pattern“ Attribut erhalten diesen allgemeinen Typ.

range

Bei diesem allgemeinen Typ wird ein Wertebereich festgelegt. Die logischen Typen „number“, „email“, „url“, „tel“, „date“, „datetime“, „datetime-local“, „month“„week“ und der logische Typ „text“ mit einem verwendeten „pattern“ Attribut werden diesem zugeordnet.

Die allgemeinen Typen infinite und range sind ebenfalls in der Lage eine Liste zu besitzen, während der Typ finite ausschließlich aus einer Liste besteht. Es kann vorkommen, dass während der Erstellung des globalen Objektes Terme verwendet werden, welche unterschiedliche logische Typen besitzen.
In dieser Arbeit wird versucht, möglichst allgemeine Wertebereiche festzulegen, um dem Nutzer eine möglichst große Auswahl an Eingaben zu ermöglichen. Aus diesem Grund werden die allgemeineren Typen range und infinite dem Typ finite vorgezogen. Der Typ range wiederum wird anstatt des Typen infinite, aus Gründen der Datenverwertung der aktiven Ontologie, welche aus dem Konstruktionsplan erstellt wird, bevorzugt. Im Folgenden bedeutet diese Ungleichung „$a < b$“, dass b für das globale Element als Typ oder

Attribut bevorzugt wird, wenn ein Term b enthält. Die erläuterten Abbildungsregeln sind im Folgenden dargestellt [HHYW03].

$$finite < infinite < range \tag{4.13}$$

Der globale Term erhält schließlich einen der drei genannten Typen. Diese werden wiederum in ihre HTML-Elemente unterteilt. Die Abbildungsregeln werden im Folgendem erläutert.

finite Darstellungs-Typen

Der finite Typ besteht lediglich aus dem Typ „list“. Dieser ist in drei Darstellungstypen zu unterteilen. Diese sind „select“, „checkbox“ und „radio“. Die ersten beiden Typen werden dem Dritten immer bevorzugt, da dieser nur die Möglichkeit hat eine Option auszuwählen und spezifischer als die anderen beiden ist. Die anderen beiden Typen sind von ihren Eigenschaften identisch. Für eine Normierung wurde „select“ als allgemeinster Typ ausgewählt. Die Abbildungsregeln lauten wie folgt.

$$radio < checkbox < select \tag{4.14}$$

infinite Darstellungs-Typen

Der logische Typ von diesem allgemeinen Typen ist „text“. Es ist darauf zu achten, dass das „pattern“ Attribut bei diesem Term nicht gesetzt ist, da dieser andernfalls von dem Typ range wäre. Dieser logische Typ besteht aus den Darstellungstypen „text“ und „textarea“. Der Typ „text“ wird in diesem Fall immer bevorzugt, da dieser mehr Attribute ermöglicht und daher der allgemeinere Typ ist. Die Abbildungsregel lautet wie folgt.

$$textarea < text \tag{4.15}$$

range Darstellungs-Typen

Dieser Typ enthält die meisten logischen Typen. Die Abbildungen sind im Folgenden dargestellt.

$$\begin{aligned} text < url < email < number < tel < time < week < \\ month < datetime - local < datetime < date \end{aligned} \tag{4.16}$$

Wie zu sehen ist, werden in diesem Fall teilweise spezifischere Typen bevorzugt. Der Grund hierfür ist beispielsweise, dass eine Telefonnummer aussagekräftiger ist als eine Zahl und somit besser in der aktiven Ontologie verwendet werden kann. Besitzt ein Cluster eine Telefonnummer und eine Zahl als Term ist davon auszugehen, dass beide Eingabefelder der Terme eine Telefonnummer als Eingabe erwarten. Aus diesem Grund wird die Anzahl der möglichen korrekten Eingaben durch das Bevorzugen der Telefonnummer als globalen Typ nicht eingeschränkt und die Aussagekraft des globalen Objektes verbessert. Andere Typen wie „tel“ und „email“ dürfen sich nicht in einem Cluster befinden, da dies ein Fehler in einem Cluster wäre. Dennoch besitzen diese Terme Abbildungsregeln, da diese sich beispielsweise mit dem Typen „text“ in einem Cluster befinden könnten.
Zusätzlich bestehen die logischen Typen „text“ und „number“ wiederum aus mehreren Darstellungstypen. Der erste dieser Typen kann nur den Darstellungstyp „text“ enthalten, da der Typ „texterea“ das „pattern“ Attribut nicht besitzen kann. Der Typ „number“ bevorzugt den allgemeineren seiner beiden Typen, die Abbildungsregeln sind im Folgenden dargestellt.

$$range < number \tag{4.17}$$

Attribut	Abbildungsregeln
checked	gesetzt > nicht gesetzt
id	majority stategy
label	majority stategy
maxlength	kleiner Wert < hoher Wert < nicht gesetzt
max, min und step	Wahl des größten Wertebereiches
multiple	gesetzt < nicht gesetzt
name	majority stategy
option	Listen Zusammenführung
pattern	Aneinanderreihen der Attributs-Werte
placeholder	majority stategy
required	gesetzt > nicht gesetzt

Tabelle 4.3.: Abbildungsregeln der Attribute für die Erstellung eines globalen Wertebereiches

Ersetzt man die allgemeinen Typen „finit“, „infinite“ und „range“ durch die Abbildungsregeln ihrer Typen, erhält man eine Abbildungsvorschrift aller Darstellungstypen, welche in den Clustern enthalten sein können. Diese Regeln sind im Folgenden zu sehen.

$$\begin{aligned} radio < checkbox < select < textarea < text(ohnepatternAttribut) < \\ text(mitpatternAttribut) < url < email < range < number < tel < time < \\ week < month < datetime - local < datetime < date \end{aligned} \tag{4.18}$$

4.7.2. Globale Attribute

In dem nächsten Schritt werden die globalen Attribute gesetzt. Diese werden für jedes Cluster einzeln bestimmt. Unabhängig von dem Typen des globalen Terms werden alle Attribute von jedem Term innerhalb des Clusters berücksichtigt, um der aktiven Ontologie mehr Informationen über die Semantik des Elementes bereitzustellen.
In Tabelle 4.3 sind alle Attribute und deren Abbildungsregeln dargestellt. Einige dieser Regeln werden im Folgendem erläutert.

Majority strategy

Die majority strategy aus [HHYW05] ist eine Strategie, welche das am häufigsten vorkommende Element aus einer Menge als globales Element auswählt. Für alle Attribute aus Tabelle 4.3, welche diese Strategie als Abbildungsregel gewählt haben, wird die am häufigsten vorkommende Zeichenkette aller beteiligten Terme gewählt.

Wahl des größten Wertebereiches

Die Attribute „max“, „min“ und „step“ legen für ein Zahleneingabefeld einen Wertebereich fest. Um möglichst viele Eingaben zu ermöglichen, werden diese Terme auf den größten Wertebereich abgebildet. Die Abbildungsregeln lauten wie folgt.
Die Eingabewertespanne wird durch das größte „max“ und das kleinste „min“ Attribut festgelegt. Dadurch wird der größte Wertebereich der Zahleneingabefelder für das globale Objekt generiert. Ist in einem Term eines dieser Attribute nicht gesetzt, wird dieses nicht berücksichtigt, da die aktive Ontologie unendliche Werteingaben schlechter verarbeiten kann. Sollten alle Terme ein „step“ Attribut besitzen wird der kleinste Wert ausgewählt, ist dies nicht der Fall wird das „step“ Attribut in dem globalen Objekt nicht gesetzt. In Abbildung 4.6 ist ein Beispiel einer solchen Abbildung zu sehen.

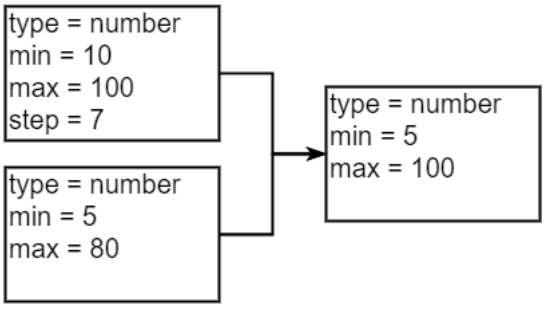

Abbildung 4.6.: Festlegen des Wertebereiches zwischen zwei range Typen.

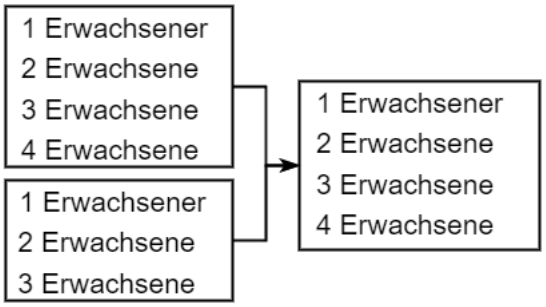

Abbildung 4.7.: Zusammenführung von Auswahlelementen.

Listen Zusammenführung

Diese Abbildungsregeln finden für alle Terme statt, welche Auswahlelemente besitzen. Gleiche Auswahlelemente werden bei diesem Vorgang zu einem globalen Auswahlelement zusammengeführt. Für jedes dieser Cluster wird im Anschluss eine globale Option erstellt. Die zuvor erläuterte majority strategy entscheidet über die Namen dieser Optionen. Diese werden anschließend als Auswahlliste dem globalen Term hinzugefügt. In Abbildung 4.7 wird beispielhaft eine Zusammenführung von zwei Listen mit Auswahlelementen demonstriert.

Aneinanderreihen der Attribut-Werte

Das „pattern“ Attribut verwendet als Abbildungsregel eine Aneinanderreihung der regulären Ausdrücke eines jeden Terms.

Durch das zusammensetzen der globalen Typen und der globalen Attribute entstehen die globalen Terme. Diese werden für die Erstellung des Konstruktionsplanes als Vorlage für die globalen Elemente verwendet, welche in dem nächsten Kapitel vorgestellt werden.

4.8. Erstellung eines Konstruktionsplanes

In diesem Kapitel wird der Konstruktionsplan erläutert, welcher für die Erstellung einer aktiven Ontologien verwendet wird.
Dieser besteht aus mehreren Komponenten, welche ineinander verschachtelt sind. Die einzelnen Komponenten, welche in Abbildung 4.8 dargestellt sind, werden im Folgenden erläutert.

Konstruktionsplan

Diese Komponente umfasst den gesamten Konstruktionsplan, welcher ein XML-Dokument ist. In diesem sind die Abbildungen der verschiedenen Dienstkategorien enthalten. Jede Dienstkategorie besitzt einen Namen. Das unten stehende Beispiel zeigt die Darstellung in XML.

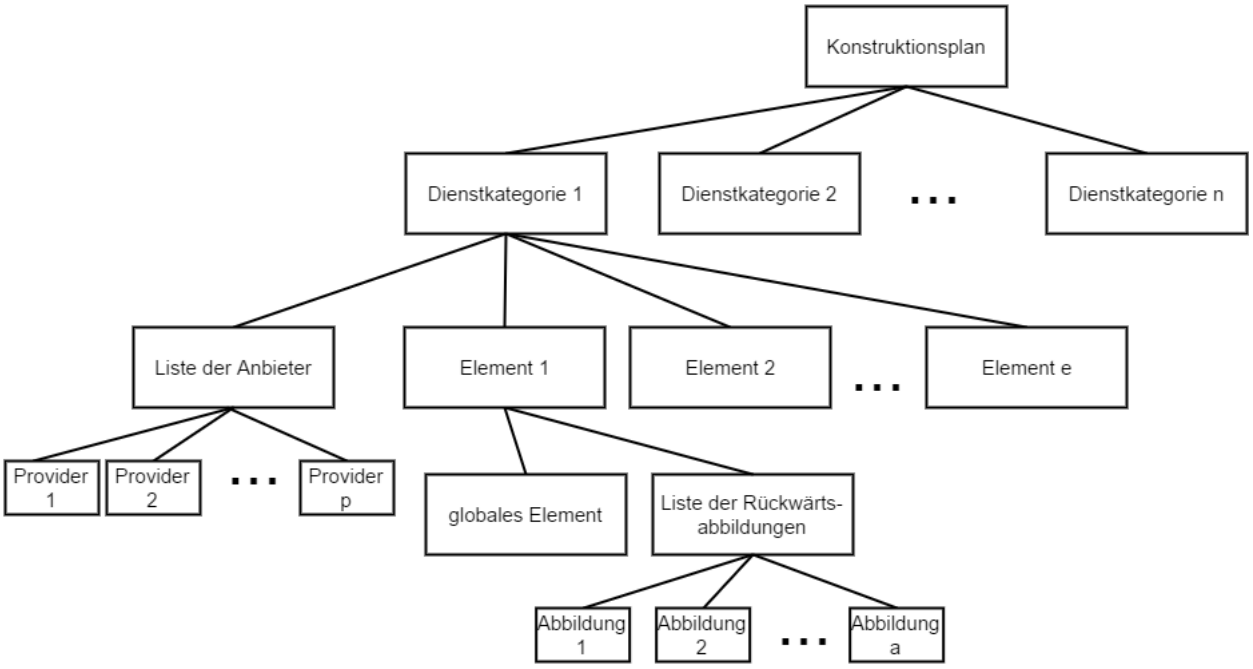

Abbildung 4.8.: Konstruktionsplan Schema

```
<categories>
   <category name="BUS">
   </category>
   <category name="FLUG">
   </category>
</categories>
```

Dienstkategorie

Diese Komponente enthält eine Liste von Anbieter und verschiedenen Elementen. Die Elemente enthalten jeweils die globalen Objekte aller Formulare, welche den Anbieter zuzuordnen sind, welche in der Anbieter-Liste der Dienstkategorie gelistet sind. In dem folgenden Codeabschnitt ist der Inhalt der Kategorie in XML dargestellt.

```
<providers>
</providers>
<formelements>
   <formelement>
   </formelement>
   <formelement>
   </formelement>
</formelements>
```

Liste der Anbieter

In dieser Liste sind sämtliche Anbieter der jeweiligen Dienstkategorien enthalten. Jeder Anbieter besitzt ein Namen, eine URL, ein Attribut mit welchem das Formular identifiziert werden kann, einen Wert mit welchem das Formular identifiziert werden kann. Formulare können aus einer oder mehreren Seiten bestehen. Aus diesem Grund besitzt der Anbieter zusätzlich eine Seitenanzahl. Diese Attribute sind für das Absenden der Formulare von der aktiven Ontologie notwendig. Zusätzlich besitzt jeder Anbieter eine ID, welche in den Rückwärtsabbildungen ebenfalls enthalten sind, um die jeweilige Abbildung dem jeweiligen Anbieter zuordnen zu können. Im unten stehenden Codeabschnitt ist ein Beispiel für ein Anbieter Element.

Komponente	Beschreibung
elementType	Diese Komponente gibt den Typ eines Feldes an.
hypernym	Das Hypernym ist eine globale Bezeichnung für das Element.
max und min	Legen den Wertebereich des Elementes fest.
maxlength	Legt die Maximale Anzahl an Zeichen fest, welche als Eingabe verwendet werden darf.
multiple	Gibt an ob mehrere Eingaben möglich sind.
name	Gibt den globalen Attributsnamen des Objektes an.
pattern	Gibt einen regulären Ausdruck an, mit welchem die Eingabe überprüft wird.
placeholder	Gibt den Inhalt des Platzhalters an, falls vorhanden.
required	Diese Komponente bestimmt ob das globale Objekt bei einer Auswertung Notwendig ist oder nicht.
step	Legt das Intervall fest, welches den Abstand zwischen möglichen Eingabewerten bestimmt.
type	Der Typ wird nur verwendet, wenn der ElementType von dem Typ input ist. In diesem Fall gibt der type den globalen Typ des Eingabefeldes an.
values	Hier sind die Optionen der Auswahlliste des globalen Objektes enthalten.

Tabelle 4.4.: Komponenten eines Elementes für den Konstruktionsplan

```
<provider id="1">
   <name>Emirates</name>
   <url>www.emirates.com/</url>
   <formIdentAttribute>id</formIdentAttribute>
   <formIdentValue>aspnetForm</formIdentValue>
   <formsteps>1</formsteps>
</provider>
```

Element

Ein Element besteht aus einem globalen Element und einer Liste von Rückwärtsabbildungen. Mithilfe des in dieser Arbeit erstellten Werkzeuges wurden globale Objekte generiert. Diese werden in diesem Konstruktionsplan als globales Element gespeichert. Die Liste von Rückwärtsabbildungen enthält Informationen über die lokalen Terme, aus welchen die globalen Objekte entstanden sind. Durch diese kann eine Rückwärtsabbildung von einem globalen Element auf das jeweilige lokale Element, in dem jeweiligen HTML-Formular, stattfinden.

Globales Element

Das globale Element besteht aus verschiedenen Komponenten, welche in Tabelle 4.4 erläutert werden. Das Attribut „value“ ist eine Option aus der Auswahlliste und besteht aus vier Komponenten. Die erste Komponente ist das „attribute“. Diese speichert den Übergabewert. Die zweite Komponente enthält die Beschreibung des Elementes. Eine weitere Komponente gibt an, ob dieses Attribut standardmäßig ausgewählt wird. Zusätzlich kann jedes globale Element Post- und Präfixe erhalten. Diese enthalten die Beschreibung der Terme als Inhalt. Befindet sich die Beschreibung des Formularelementes vor dem Element, ist diese ein Präfix, andernfalls ist sie ein Postfix.
Ein Beispiel für ein Präfix wäre ein Texteingabefeld mit einer davor stehenden Beschreibung namens „von“. Dieser Term würde in der aktiven Ontologie als Präfix

Komponente	Beschreibung
identAttribute	Enthält das Attribut mit welchem das HTML-Element eindeutig referenziert werden kann.
identValue	Enthält den Wert des identAttribute Atributes.
identType	Gibt den Eingabetyp an, falls es sich um eine komplexe Abbildung handelt.

Tabelle 4.5.: Attribute für die Identifizierung lokaler Objekte.

Knoten erstellt werden. Er würde die Sprachausgabe nach dem Wort „von“ absuchen und das nachfolgende Wort auf einen Abflughafen überprüfen.
Die letzte Komponente ist die Abbildungs-ID. Diese wird benötigt, um die Optionen des globalen Elementes auf die Optionen innerhalb der Rückwärtsabbildungen abzubilden. Dadurch ist es der aktiven Ontologie möglich, bei der Auswahl einer Option, jede korrespondierende Option der HTML-Formulare anzusprechen.

Rückwärtsabbildungen

Die Rückwärtsabbildungen stellen die Abbildungen der globalen Objekte auf die lokalen Elemente der Anbieter dar. Jede dieser Abbildungen enthält eine ID, welche auf den Anbieter des Elementes verweist. Diese Elemente enthalten wieder alle in Tabelle 4.4 enthaltenen Attribute. Zusätzlich existiert ein „page“ Attribut, welches auf die Seite hinweist, auf welcher sich das lokale HTML-Element befindet. Das „hypernym“ Attribut wird durch 3 andere Attribute ersetzt, welche das lokale Element innerhalb des Formulars eindeutig bestimmen können. Diese sind in Tabelle 4.5 aufgelistet. Als Erweiterung wird für jede Rückwärtsabbildung markiert, ob es sich um eine komplexe Abbildung handelt. Die komplexen Abbildungen wurden in Abschnitt 3.1.1 vorgestellt und erläutert. Das Behandeln dieser Abbildungen ist mithilfe dieses Werkzeugs zu dem jetzigen Standpunkt nicht möglich. Eine mögliche Behandlung dieser wird in dem nächsten Kapitel vorgestellt. Dennoch können diese Abbildungen in der Schnittstelle eingefügt werden. Handelt es sich um eine einfache Abbildung, wird dies in der Rückwärtsabbildung markiert.Ist dies nicht der Fall, enthält die Rückwärtsabbildung eine Liste von Elementen, welche das komplexe Element darstellen. Das unten stehende Beispiel demonstriert eine einfache Abbildung.

```
<reverseMapping mappingType="simple">
    <mapping>
    </mappings>
</reverseMapping>
```

In komplexen Abbildungen ist es möglich, dass alle Elemente, welche auf das globale Element abgebildet werden unterschiedliche Eingabetypen besitzen. Um dennoch eine Rückwärtsabbildung von einem globalen Element auf ein lokales Element durchführen zu können, existiert der Typ „identType“.

4.9. Komplexe Abbildungen

Komplexe Abbildungen im Folgenden auch 1:m genannt, wurden in Abschnitt 3.1.1 bereits vorgestellt. Bei diesen Abbildungen werden mehrere lokale Terme eines Formulars auf ein globales Objekt abgebildet. Diese Abbildungen werden von dem erstellten Werkzeug dieser Bachelorarbeit nicht behandelt. Der Grund hierfür ist der zu hohe Arbeits- und Zeitaufwand für die Integrierung eines Verfahrens, welches komplexe Abbildungen erkennt. Als Ausblick für dieses Thema werden die in der Arbeit [WYDM04] behandelten Verfahren und Ansätze vorgestellt.

Für die Erkennung von komplexen Abbildungen ist es notwendig, zusammenhängende Terme bestimmen zu können. Dazu werden alle Elemente des in Abschnitt 4.2 erstellten Baumes, welche kein Eingabefeld besitzen, auf deren Inhalt überprüft. Werden Ähnlichkeiten zu Elementen mit Eingabefeldern gefunden, werden Terme mit Eingabefeldern, welche Kinder dieser Terme sind gesucht. Ein Problem dieses Vorgehens ist der Aufbau der HTML-Formulare. Eingabefelder der komplexen Abbildungen besitzen oft die gleiche Tiefe der dazugehörigen Beschreibungen innerhalb eines Baumes und sind nicht deren Kinder. Diese Darstellung erschwert das finden von komplexen Abbildungen sehr, da oft keine eindeutige Zuweisung zwischen Beschreibung und Eingabefeldern stattfinden kann.
Eine weitere Hürde ist das Erkennen von ist-Element-von oder aggregierten Abbildungen. Das Lösen dieses Problems ist für das Ansprechen der Dienstanbieter über eine aktiven Ontologie notwendig und automatisiert sehr schwer lösbar.

4.10. Zusammenfassung

In diesem Kapitel wurde die Erstellung eines Konstruktionsplanes von Formularen einer Dienstkategorie bis hin zu einem Konstruktionsplan erläutert. Mithilfe verschiedener Lösungen der verwandten Arbeiten und eigenen Lösungen wurde dieses Ziel umgesetzt. In dem Folgenden Kapitel wird der Entwurf des Werkzeugs vorgestellt.

5. Entwurf und Implementierung

In diesem Kapitel werden zunächst der Entwurf und anschließend die Implementierung des Werkzeugs vorgestellt.

5.1. Entwurf

In diesem Abschnitt wird der Entwurf des zu erstellenden Werkzeugs erläutert. Hierfür wird die Umsetzung des in der Analyse vorgestellten Prozessessablaufes präsentiert. Abbildung 5.1 zeigt die Architektur des Werkzeugs. Diese kann in 4 Teilbereiche *Erstellung lokaler Objekte*, *Analyse*, *Bestimmung semantisch gleicher Elemente* und *Erstellung der globalen Objekte und des Konstruktionsplans* unterteilt werden.
Die zentrale Komponente ist der *Distributor*. Dieser initiiert die Prozesse und steuert den Prozessablauf. Die Reihenfolge, in welcher die einzelnen Prozesse angesteuert werden wird in Abbildung 5.2 veranschaulicht. Die verschiedenen Komponenten und Teilbereiche werden im Folgenden erläutert.

Objectgenerator

Die erste Komponente, welche der *Distributor* anspricht, ist der *Objectgenerator*. Dieser erhält die verschiedenen HTML-Formulare und bildet diese auf Bäume ab.

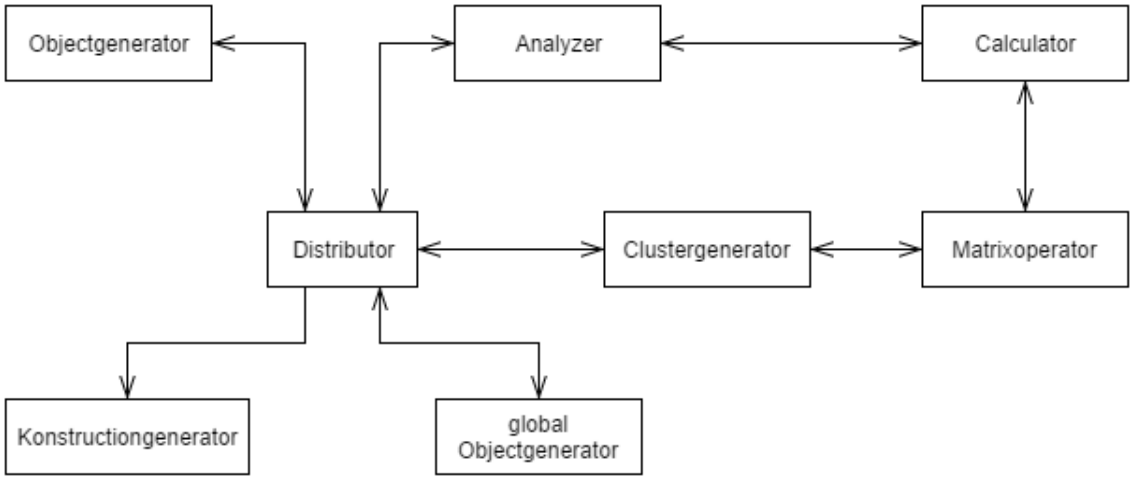

Abbildung 5.1.: Aufbau des Werkzeugs

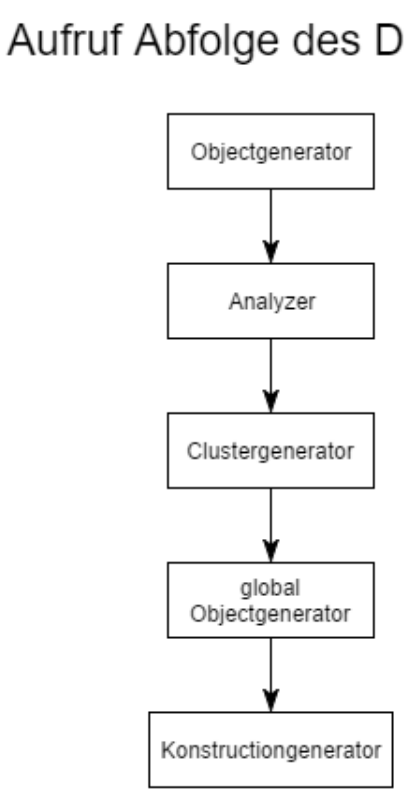

Abbildung 5.2.: Ansteuerungsreihenfolge des *Distributors*

Anschließend leitet der *Distributor* die ausgegebenen Bäume des *Objectgenerators* an den *Analyzer* weiter.

Analyzer und Calculator

Der *Analyzer* stellt zusammen mit dem *Calculator* den *Analyse* Teilbereich dar. Die *Analyzer* Komponente erhält die erstellten Bäume als Eingabe und produziert eine Ergebnismatrix als Ausgabe. Die Einträge der Ergebnismatrix stellen die Ähnlichkeiten zwischen zwei Termen dar. Während der *Analyzer* verschiedene Analyseprozesse koordiniert, führt der *Calculater* die einzelnen Analyseprozesse durch. Dieser hat die verschiedenen Analysemethoden implementiert und gibt die Ergebnisse an den *Analyzer* weiter. Während der Werte-Analyse wird ein Cluster-Verfahren angefordert, wenn ein Vergleich zwischen zwei Auswahllisten durchgeführt wird. Dieses ist in dem Matrixoperator implementiert. Bei Bedarf einer Matrixberechnung fragt der *Calculator* diesen an, um ein Ergebnis zu erhalten.

Clustergenerator und Matrixoperator

Die Komponenten *Clustergenerator* und Matrixoperator bilden zusammen den Teilbereich *Bestimmung semantisch gleicher Elemente*. Die Ergebnismatrix wird zusammen mit den HTML-Bäumen an den *Clustergenerator* weitergegeben. Der *Clustergenerator* gibt die Matrix zunächst an den Matrixoperator weiter. Dieser gibt eine geordnete Liste von allen Ähnlichkeiten, welche über einem gewissen Grenzwert liegen, zurück. Der *Clustergenerator* erstellt anschließend die Cluster mithilfe dieser Liste und gibt die Cluster-Liste an den *Distributor* weiter.

Global Objectgenerator und Constructiongenerator

Die globalen Objekte werden von dem *global Objectgenerator* erzeugt. Dieser erhält die Cluster-Liste als Eingabe und gibt eine Liste von globalen Objekten aus. Sowohl

die Cluster-Liste als auch die Liste, welche die globalen Objekte enthält, wird von dem *Distributor* an den *Constructiongenerator* weitergereicht. Dieser erstellt den Konstruktionsplan in einer XML-Datei.

5.1.1. Erstellung lokaler Objekte

Für das Zusammenführen der HTML-Formulare werden zunächst neue lokale Objekte aus den HTML-Formularen erstellt. Dazu werden die im Analyse Kapitel vorgestellten Verfahren durchgeführt. Das Ergebnis dieses Abschnitts sind Bäume, deren einzelne Knoten HTML-Elemente repräsentieren. Mithilfe der Erstellung dieser hierarchischen Struktur können Analysen durchgeführt werden, welche den Aufbau des HTML-Formulars berücksichtigen.
Die einzelnen HTML-Formulare werden dabei nacheinander abgearbeitet. Der *Objectgenerator* führt die einzelnen Prozesse durch. Zunächst wird ein Baum aus den einzelnen Formularelementen des HTML-Dokumentes erstellt. Anschließend werden die einzelnen Knoten des Baumes durch iteriert und die Normalisierung, welche in der Analyse erläutert wurde, durchgeführt. Schließlich werden die unterschiedlichen HTML-Elementobjekte zu Termen verschmolzen. Nachdem alle HTML-Formulare diesen Vorgang durchlaufen haben, werden diese über den *Distributor* an die Analyse weitergereicht.

5.1.2. Analyse

Die Analyse erhält die in Abschnitt 5.1.1 erstellten Bäume als Eingabe und besteht aus zwei Komponenten, dem *Analyzer* und dem *Calculater*. Der *Analyzer* koordiniert das Analyseverfahren, während der *Calculater* die eigentlichen Berechnungen der Analyse durchführt. Zunächst wird von dem *Analyzer* eine Matrix erstellt. Jeder Term, welcher ein Eingabefeld darstellt, erhält eine Spalte und eine Zeile. Im nächsten Schritt wird die linguistische Analyse durchgeführt. Dabei wählt der *Analyzer* einen Matrixeintrag aus, welcher Termen zuzuordnen ist, welche sich in unterschiedlichen Formularen befinden und initiiert nacheinander die Token-Analyse, Teilzeichenketten-Analyse und Werte-Analyse. Die Berechnungen der drei Analysen werden von dem *Calculator* durchgeführt. Für die Werte-Analyse wird zusätzlich ein Matrixoperator zu Hilfe gezogen, wenn ein Cluster-Verfahren benötigt wird. Die Ergebnisse führt der *Analyzer* in dem jeweiligen Matrixeintrag zusammen. Wurden alle Terme verglichen, wird mit der strukturellen Analyse fortgefahren. Zunächst wird eine neue Matrix erstellt. Anschließend wird durch alle relevanten Matrixeinträge iteriert. Die strukturelle Analyse verwendet die Ergebnisse der linguistischen Analyse als Eingabe. Dieser Vorgang wurde in Kapitel 4 erläutert. In der neu erstellten Matrix werden die Ergebnisse der strukturellen und linguistischen Analyse zusammengeführt. Das Resultat ist eine Matrix, welche die Ähnlichkeitswerte aller für den Konstruktionsplan relevanten Terme besitzt. Die Matrix wird an das Cluster-Verfahren weitergereicht.

5.1.3. Bestimmung semantisch gleicher Elemente

Das Cluster-Verfahren wird durch einen *Clustergenerator*, welcher mithilfe des Matrixoperators Cluster erzeugt, umgesetzt. Die Eingabe des *Clustergenerators* sind die Bäume und die in der Analyse erstellte Matrix. Der *Clustergenerator* gibt zunächst die Matrix an den Matrixoperator mit einem Grenzwert weiter. Dieser gibt eine Liste von allen Ähnlichkeiten aus, welche sich über diesem Grenzwert in der Matrix befinden, wobei die Ähnlichkeiten der Größe nach sortiert sind. Bei gleichen Werten sind die Ähnlichkeiten nach der Reihenfolge innerhalb der Formulare angeordnet. Der *Clustergenerator* iteriert anschließend durch die Liste und erstellt anhand dieser Ähnlichkeiten die Cluster nach den in dem Analyse Kapitel erläuterten Regeln. Anschließend werden aus allen Termen, welche für den Konstruktionsplan relevant sind, sich aber in keinem Cluster befinden, einelementige Cluster erstellt und den anderen Clustern beigefügt. Die Ausgabe ist eine Liste von Clustern.

5.1.4. Erstellung globaler Objekte und Konstruktionsplan

Für die Erstellung globaler Objekte wird ein *global Objectgenerator* verwendet. Dieser erhält die Cluster-Liste aus dem Cluster-Verfahren als Eingabe. Die einzelnen Cluster werden einzeln abgearbeitet. Für jedes Cluster, auch wenn dieses nur ein Element besitzt, wird ein globales Objekt erstellt. Dazu werden die Informationen der Elemente der Cluster und die globalen Objekte anhand der in dem Analyse Kapitel vorgestellten Regeln erstellt. Die Ausgabe des *global Objectgenerator* ist eine Liste von globalen Objekten.
Anschließend wird der Konstruktionsplan erstellt. Zunächst wird die Provider-Liste anhand der Formularbäume generiert. Anschließend werden die in dem Analyse Kapitel beschriebenen Formularelemente erstellt. Dazu werden die globalen Elemente aus den globalen Objekten generiert und die Rückwärtsabbildungen mithilfe des zugehörigen Clusters erstellt und beigefügt. Ist dieser Vorgang mit allen Cluster und globalen Objekten durchgeführt worden, werden sowohl die Cluster-Liste als auch der globalen Elemente an den *Constructiongenerator* weitergegeben. Dieser erstellt den Konstruktionsplan als XML-Datei.

5.2. Implementierung

In diesem Abschnitt wird die Umsetzung des Entwurfs anhand der Programmiersprache Java erläutert. Zusätzlich werden die verwendeten Bibliotheken vorgestellt.

5.2.1. Erstellung lokaler Objekte

Für die Erstellung von lokalen Objekten müssen zunächst die einzelnen HTML-Formulare gelesen und in Elemente unterteilt werden. Zu diesem Zweck wurde der HTML-Parser JSOUP[2] verwendet. Dieser erhält ein HTML-Formular als Eingabe und erstellt einen DOM-Baum für dieses Formular. Ein DOM-Baum (document object model) ist eine Datenstruktur, welche ein HTML-Dokument als hierarchischen Baum darstellt. Diese Datenstruktur wird anschließend in eine selbst erstellte Baumdatenstruktur überführt, aus welcher die relevanten Informationen besser abgegriffen werden können. Während der Generierung dieser Datenstruktur wird eine Normalisierung durchgeführt und die Ausgabe der Datenstruktur beigefügt.
Mithilfe eines Iterators kann über den Baum iteriert werden. Als Stemming-Algorithmus für die Normalisierung wird der Snowball-Stemmer aus der tartarus[3] verwendet. Bibliothek verwendet. Jeder Knoten eines Baumes, bis auf das Formularelement, besitzt zusätzlich eine ID, mit der dieser eindeutig identifiziert werden kann und den Namen des Providers. Die Ergebnisse werden in einer Liste eines Kategorie Objektes gespeichert.
Ein Kategorie Objekt stellt einen Iterator bereit, welcher durch alle Knoten aller Formulare iterieren kann und dabei die Formelementknoten auslässt. Dieser Iterator wird aus dem Iterator der Baumliste und den Iteratoren der einzelnen Bäume zusammengesetzt. Um die Anzahl der Terme aller Formulare zu bestimmen, wird einmal durch alle Elemente iteriert. Das Ergebnis wird ebenfalls in dem Kategorie Objekt gespeichert.

5.2.2. Analyse

Mithilfe des Iterators der Kategorie wird eine Matrix erstellt, wobei einem Term die Zeile i und die Spalte i zugewiesen wird, wenn dieser nach der iten Iteration ausgegeben wird. Anschließend wird durch die Matrix mit zwei Iteratoren iteriert. Jede Zelle der Matrix stellt einen Vergleich dar. Ein Iterator iteriert durch die Spalten, der andere durch die

[2] JSOUP, Online erhältlich unter https://jsoup.org/; abgerufen am 10. Dezember 2016
[3] Snowball-Stemmer, Online erhältlich unter http://lucene.apache.org/core/3_0_3/api/contrib-snowball/; abgerufen am 10. Dezember 2016

Zeilen. Der Vergleich von Spalte i und Zeile i ist der Vergleich eines jeden Elementes mit sich selbst. Die beiden Vergleiche Zeile i, Spalte j und Zeile j, Spalte i vergleichen jeweils die beiden gleichen Terme. Um diese irrelevanten Vergleiche zu verhindern, startet der zweite Iterator an dem Term, welcher sich nach dem Term des ersten Iterators befindet. Alle Vergleiche, welche zwei Terme desselben Formulars enthalten werden übersprungen. Dies kann mithilfe des Providernamens der Knoten festgestellt werden. Bei einem Vergleich wird zunächst eine Token-Analyse, anschließend eine Teilzeichenketten-Analyse und zum Schluss eine Werte-Analyse durchgeführt. Die einzelnen Ergebnisse werden mit ihren Gewichtungen multipliziert und als Summe in die jeweilige Zelle der Matrix eingetragen. Nachdem alle relevanten Einträge dieser Matrix eingetragen wurden, wird für die strukturelle Analyse ebenfalls eine Matrix erstellt und die Terme nach dem gleichen Prinzip abgehandelt. In der strukturellen Analyse werden bei einem Vergleich für jeden der Terme zwei Listen erstellt. Diese enthalten die Nachbarterme dieser Terme, welche sich innerhalb der Formulare befinden. Hierfür werden wieder die Iteratoren innerhalb der Formularbäume verwendet. Anschließend wird die Ähnlichkeit der strukturellen Analyse mithilfe dieser 4 Listen bestimmt. Diese Ähnlichkeit wird mit einer Gewichtung multipliziert und als Summe mit dem Eintrag derselben Matrixzelle der linguistischen Analyse, welcher ebenfalls mit einer Gewichtung multipliziert wurde, in die Ergebnismatrix eingefügt.

5.2.3. Cluster-Verfahren

Zunächst wird eine Liste mit allen Ähnlichkeiten, welche über einem gewissen Grenzwert liegen erstellt. Dazu wird mit zwei Iteratoren durch die Matrix iteriert, der größte Wert in der Liste gespeichert und der Vorgang wiederholt. Sind mehr als ein maximaler Wert in der Matrix vorhanden, wird der erste gefundene Wert ausgewählt. Durch dieses Vorgehen werden Ähnlichkeiten bevorzugt, welche weiter oben in den Formularen stehen und die Reihenfolge der Formularelemente innerhalb der Formulare berücksichtigen. Anschließend werden die Cluster erzeugt und einer Liste beigefügt. Dazu wird jede Ähnlichkeit aus der Ähnlichkeitenliste mit allen Clustern verglichen und nach den Regeln aus Kapitel 4 eingefügt.

5.2.4. Erstellung der globalen Objekte und des Konstruktionsplans

Zunächst werden die globalen Objekte aus den Attributen der lokalen Objekte erstellt. Dazu werden unterschiedliche Verfahren für das Erstellen der Attribute des globalen Objektes verwendet. Diese werden im Folgendem vorgestellt.

Auswahlelement

Alle Auswahlelemente von allen Termen eines Clusters werden in einer Liste gesammelt. Ist das Auswahlelement bereits in der Liste enthalten, wird dieses nicht hinzugefügt.

Name, Beschreibung, ID und Platzhalter

Für jedes dieser Attribute wird eine Liste erstellt. Die Werte dieser Attribute werden anschließend in den Listen gesammelt. Das am häufigsten vorkommende Element aus der Liste wird als globales Attribut gewählt.

Die Attribute max und maxLength

Für diese Attribute wird ein temporärer Wert erstellt. Dieser Wert speichert den momentanen Zustand des globalen Objektes. Die Werte der Attribute werden nacheinander ausgewertet. Die Auswertung findet jeweils zwischen dem temporären Wert und dem Wert des nächsten Attributes statt. Der höhere Wert wird als temporärer Wert gespeichert. Sind alle Attribute ausgewertet worden, wird der temporäre Wert als globaler Wert für dieses Attribut übernommen.

Die Attribute min und step

Für diese Attribute wird ein temporärer Wert erstellt. Dieser Wert speichert den momentanen Zustand des globalen Objektes. Die Werte der Attribute werden nacheinander ausgewertet. Die Auswertung findet jeweils zwischen dem temporären Wert und dem Wert des nächsten Attributes statt. Der niedrigere Wert wird als temporärer Wert gespeichert. Sind alle Attribute ausgewertet worden, wird der temporäre Wert als globaler Wert für dieses Attribut übernommen.

required und multiple

Für diese Attribute wird ein temporärer boolescher Wert erstellt, welcher als falsch initiiert wird. Dieser Wert speichert den momentanen Zustand des globalen Objektes. Die Terme werden nacheinander ausgewertet. Ist ein Attribut wahr, wird der temporäre Wert auf wahr gesetzt. Wurden alle Terme des Clusters abgearbeitet, wird der temporäre Wert als globaler Wert übernommen.

Das pattern Attribut

Es wird ein temporärer Wert erstellt. Alle Werte der pattern Attribute werden an diesen mit einem „oder“ Ausdruck beigefügt. Das Ergebnis ist eine Zeichenkette, welche die Vereinigung aller regulären Ausdrücke der Terme des jeweiligen Clusters enthält.

Der Darstellungs-Typ

Die Abbildungsregeln des Analyse Kapitels für den Darstellungs-Typen werden in ein pattern matching überführt. Es wird ein temporäres Attribut erstellt. Alle Darstellungs-Typen aller Terme werden nacheinander ausgewertet. Es wird jeweils der Typ des Terms und der temporäre Wert an das pattern matching übergeben. Das Ergebnis wird von dem temporären Wert übernommen. Sind alle Terme abgearbeitet worden, wird der temporäre Wert als Wert des globalen Attributes gesetzt.

Der semantische Typ

Dieser wird existiert in einem globalen Objekt nicht, da von dem Konstruktionsplan lediglich der Darstellungs-Typ benötigt wird.

Für die Erstellung des Konstruktionsplanes wird die „java Architecture for XML Binding“[4] (JAXB)Bibliothek verwendet. Die Werte der globalen Objekte und Cluster werden von Java Objekten übernommen, welche die JAXB Bibliotheken nutzen. Anschließend werden diese Objekte als Konstruktionsplan in XML ausgegeben.

5.3. Zusammenfassung

In diesem Kapitel wurde der Entwurf und die Implementierung des Werkzeugs vorgestellt und erläutert. Dazu wurde das Werkzeug in vier Teilbereiche unterteilt. Sowohl bei dem Entwurf als auch bei der Implementierung wurden diese nacheinander abgearbeitet. Während bei dem Entwurf insbesondere auf die Architektur des Werkzeugs eingegangen wurde, wurde bei der Implementierung auf die verwendeten Bibliotheken und den Code eingegangen. In dem nächsten Kapitel wird die Auswertung des Werkzeugs präsentiert.

[4] JAXB, Online erhältlich unter https://jaxb.java.net/; abgerufen am 10. Dezember 2016

6. Evaluation

In diesem Kapitel wird das erstellte Werkzeug anhand von verschiedenen Test- und Trainingsmengen evaluiert. Dazu werden verschiedene Testläufe mithilfe des Cluster-Verfahrens durchgeführt und die Ergebnisse dieser in diesem Kapitel besprochen.
Zunächst wird der Aufbau der Evaluation vorgestellt und erläutert. Anschließend werden die Trainings- und Testmengen präsentiert. Zum Schluss werden die Ergebnisse des Cluster-Verfahrens vorgestellt und diskutiert.

6.1. Aufbau

Der Aufbau dieser Evaluation kann in 2 Abschnitte unterteilt werden. In dem ersten Abschnitt werden alle verschiedenen Zusammensetzungen aus Verfahren untersucht. Das Werkzeug verwendet viele unterschiedlichen Verfahren, wie beispielsweise der Token-Analyse oder der strukturellen Analyse. Diese Verfahren können für die Erstellung des Konstruktionsplans alle zugleich oder auch einzeln verwendet werden. Durch diese Betrachtung kann die Effektivität der einzelnen Verfahren untersucht und wichtige oder zu vernachlässigende Verfahren oder Verfahrenskombinationen bestimmt werden.
Die meisten Verfahren besitzen Parameter für die Festlegung von Gewichtungen, Grenzwerten oder anderen Werten. In dem zweiten Abschnitt wird die Bestimmung der Parameterwerte, welche zu einem möglichst guten Ergebnis führen, besprochen.

6.1.1. Mögliche Kombinationen der Verfahren

Für die Evaluation werden die einzelnen Verfahren, welche in der Implementierung umgesetzt wurden, analysiert. In Abbildung 6.1 ist zu sehen, aus welchen Verfahren die Ergebnismatrix erstellt wird. Mithilfe dieser Matrix erstellt anschließend ein hierarchisches Cluster-Verfahren die Ausgabe für die Generierung der globalen Objekte. Wie zu sehen, ist wird das Ergebnis aus einer strukturellen Analyse und einer linguistischen Analyse erstellt. Die strukturelle Analyse verwendet die Ergebnisse der linguistischen Analyse als Eingabe. Aus diesem Grund kann die strukturelle Analyse nicht ohne linguistische Analyse evaluiert werden.
Die linguistische Analyse ist kein eigenes Verfahren, sondern wird aus drei Verfahren zusammengesetzt, welche in Abbildung 6.1 zu sehen sind. Diese Analyse führt die drei Verfahren mit unterschiedlichen Gewichtungen zusammen. Diese Zusammenführung wird in der Abbildung durch einen Kreis gekennzeichnet. Die Token-Analyse verwendet zusätzlich eine Levenshtein Distanz und ein Stemming-Verfahren.

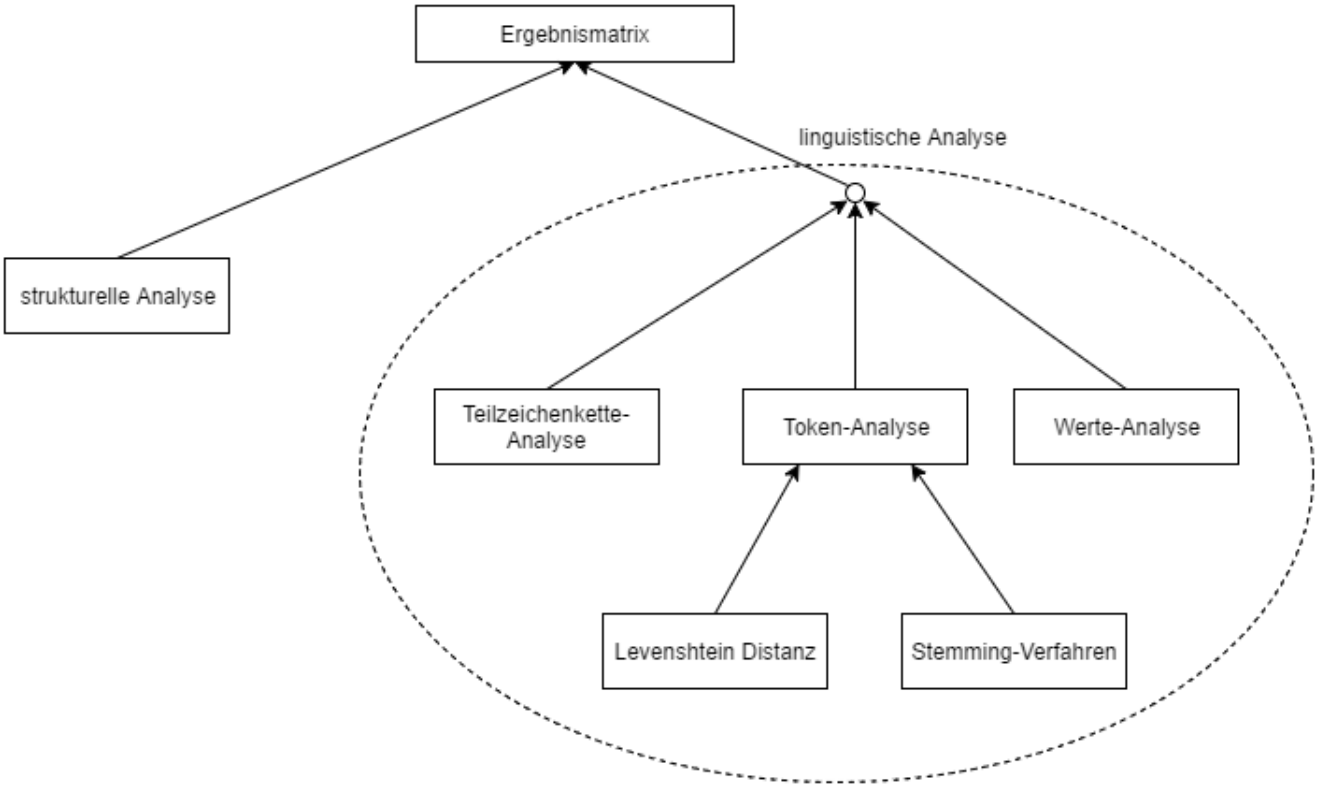

Abbildung 6.1.: Überblick der Verfahren

Unter der Berücksichtigung, dass die strukturelle Analyse nicht ohne linguistische Analyse evaluiert werden kann, sind die in Tabelle 6.1 aufgeführten Verfahrenskombinationen möglich. Jede dieser Verfahrenskombinationen wird im Folgenden an zwei Testmengen durchgeführt und ausgewertet. Die beiden Testmengen besitzen zwei unterschiedliche Dienstkategorien, um die Auswertung nicht auf eine bestimmte Dienstkategorie zu beschränken. Das Cluster-Verfahren, wurde in dieser Tabelle nicht berücksichtigt, da dieses für die Erstellung eines Konstruktionsplans notwendig ist.

6.1.2. Wahl der Parameter

Die meisten Verfahren besitzen Parameter für die Festlegung von Grenzwerten, Gewichtungen oder anderen Werten. Ein Verfahren, welches gute Parameterwerte für eine Verfahrenskombination findet, wird im Folgenden besprochen. Zunächst werden die verschiedenen Parameter in Tabelle 6.2 vorgestellt.
Im Folgenden werden für diese Parameter, mithilfe einer Trainingsmenge, Parameterwerte für alle in Tabelle 6.1 enthaltenen Verfahrenskombinationen ermittelt. Das Auswerten aller möglichen Parameterwerte an einer Trainingsmenge ist im Zeitrahmen dieser Bachelorarbeit nicht realisierbar.
Aus diesem Grund wird mithilfe des Teile und Herrsche Paradigmas dieses Problem in Teilprobleme unterteilt. Dazu werden die Parameter der strukturellen Analyse, Token-Analyse, Teilzeichenkette-Analyse, Werte-Analyse, Levenshtein Distanz und linguistischen Analyse getrennt ausgewertet. Für dieses Vorgehen wird eine Trainingsmenge erstellt. Anschließend wird das zu erwartende Ergebnis dieser Trainingsmenge manuell erstellt. Diese Ergebnisse dienen als Vorlage für die Bestimmung von falschen und richtigen Abbildungen von automatisch generierten Ergebnissen.
Die Festlegung der Parameterwerte für ein Verfahren findet in zwei Schritten statt. Zunächst werden verschiedene Parameterkombinationen mit der Trainingsmenge als Eingabe

Verfahrens-kombination	**Ln**	**Tk**	**Lv**	**St**	**Sb**	**Wa**	**Sa**
1	x	x	x	x	x	x	x
2	x	x	x		x	x	x
3	x	x		x	x	x	x
4	x	x			x	x	x
5	x	x	x	x	x		x
6	x	x	x		x		x
7	x	x		x	x		x
8	x	x			x		x
9	x	x	x	x		x	x
10	x	x	x			x	x
11	x	x		x		x	x
12	x	x				x	x
13	x	x	x	x			x
14	x	x	x				x
15	x	x		x			x
16	x	x					x
17	x				x	x	x
18	x				x		x
19	x					x	x
20	x	x	x	x	x	x	
21	x	x		x	x	x	
22	x	x	x		x	x	
23	x	x			x	x	
24	x	x	x	x	x		
25	x	x	x		x		
26	x	x		x	x		
27	x	x			x		
28	x	x	x	x		x	
29	x	x	x			x	
30	x	x		x		x	
31	x	x				x	
32	x	x	x	x			
33	x	x	x				
34	x	x		x			
35	x	x					
36	x				x	x	
37	x				x		
38	x					x	

Tabelle 6.1.: Kombinationsmöglichkeiten der verschiedenen Verfahren. Ein x steht für die Verwendung des Verfahrens, ein leeres Feld deutet auf die Nichtverwendung dieses Verfahrens hin. $\boxed{Ln}$ ist die linguistische Analyse, $\boxed{Tk}$ die Token-Analyse, $\boxed{Lv}$ die Levenshtein Distanz, $\boxed{St}$ der Stemming-Algorithmus, $\boxed{Sb}$ die Teilzeichenkette-Analyse, $\boxed{Wa}$ die Werte-Analyse und $\boxed{Sa}$ die strukturelle Analyse.

Parameter	**Beschreibung**
Grenzwert des Cluster-Verfahrens	Das Cluster-Verfahren fügt Terme über ihre Ähnlichkeiten zu Clustern zusammen. Sollten die zu Verfügung stehenden Werte unter diesen Grenzwert sinken, wird das Verfahren abgebrochen.
Gewichtungsparameter der strukturellen und der linguistischen Analyse	Diese beiden Parameter legen die Gewichtungen zwischen struktureller und linguistischer Analyse fest. Die Summe dieser beiden Parameter muss 100 ergeben.
Grenzwert der strukturellen Analyse	Die strukturelle Analyse verwendet die Ausgabe der linguistischen Analyse als Eingabe. Dabei wird das Ergebnis anhand von gleichen Term-Paaren festgelegt. Dieser Grenzwert legt fest ab welchen Ähnlichkeits-Werten die Paare als gleich anzusehen sind.
Gewichtungsparameter der Token-, Teilzeichenketten- und Werte-Analyse	Diese Parameter legen fest, wie die Ergebnisse der Token-, Teilzeichenketten- und Werte-Analyse gewichtet werden. Die Summe dieser Gewichtungen ist immer 100.
Parameter der Token-Analyse	Die Token-Analyse besitzt zwei weitere voneinander abhängige Parameter, welche in dem Analyse Kapitel vorgestellt wurden.
Levenshtein Distanz	Dieser Parameter wird in der Token-Analyse verwendet. Er legt die Größe der Levenshtein Distanz fest, welche die Analyse dabei unterstützt semantisch gleiche Wörter zu finden.
Werte-Analyse Parameter	Die Werte-Analyse besitzt zwei weitere voneinander abhängige Parameter und einen unabhängigen Grenzwert, welcher in dem Analyse Kapitel vorgestellt wurden.

Tabelle 6.2.: Verwendete Parameter der Verfahren

für ein Verfahren verwendet und für jede Verfahrenskombination automatisch die Ergebnisse mithilfe des Cluster-Verfahrens generiert. Dieser Vorgang kann aus zeitlichen Gründen nicht mit allen möglichen Parameterkombinationen stattfinden. Jedes dieser Ergebnisse wird auf ihre Präzision mithilfe der manuell erstellten Ergebnisse überprüft. Die Parameterwerte des automatisch generierten Ergebnisses mit der höchsten Präzision werden als beste Parameterwerte des Verfahrens gewählt.
Im Folgenden wird der Prozessablauf der Parameterfindung aller Verfahren erläutert. Dieser ist anhand von Abbildung 6.2 nachvollziehbar. Zunächst werden die Parameterwerte der linguistischen Verfahren ausgewertet. Die Parameterwerte, welche bei diesen Auswertungen entstehen, werden von den nachfolgenden Auswertungen wiederverwendet. Als Nächstes werden die Gewichtungen der linguistischen Analyse ermittelt und zum Schluss die Parameterwerte der strukturellen Analyse. Sollte eine Verfahrenskombination ein Verfahren nicht verwenden, wird die Auswertung der Parameter für dieses Verfahren übersprungen und die Gewichtung des Verfahrens auf null gesetzt. Dieser Vorgang wird mit allen Kombinationen aus Tabelle 6.1 durchgeführt.

6.1.3. Auswertung der Ergebnisse

Nachdem die Parameterwerte der Verfahrenskombinationen ermittelt wurden, werden mithilfe von zwei Testmengen die Präzision des Werkzeugs und die Präzision der einzelnen Verfahren ermittelt. Zunächst wird für eine Testmenge manuell ein Klassifikation erstellt. Für jede Verfahrenskombination wird mithilfe der Parameterwerte, welche anhand des oben genannten Verfahrens bestimmt wurden, automatisch ein Ergebnis generiert. Die Testenge ist dabei die Eingabe des Werkzeugs. Die Präzision jeder Verfahrenskombination wird bestimmt, indem das von dieser Kombination automatisch generierte Ergebnis mit dem manuell erstellten Ergebnis verglichen wird.
Die Präzision des Ergebnisclusters und der zugehörigen Verfahrenskombination setzt sich aus den Werten der gefundenen richtigen Positive und der gefundenen falschen Positive zusammen, welche im Folgendem erläutert werden.

Abbildung

Eine Abbildung entsteht, wenn eine aus Kapitel 4 verwendete Ähnlichkeit zu einem Cluster hinzugefügt wird. In diesem Fall wurden zwei Terme als semantisch gleich befunden und werden gemeinsam auf das globale Objekt abgebildet. Ein Beispiel hierfür ist, wenn zwei Terme sich in einem Cluster befinden und ein dritter Term mit einem Term aus dem Cluster semantisch ähnlich ist. Die Terme in dem Cluster bilden gemeinsam eine Abbildung. Durch das Hinzufügen des dritten Terms entsteht eine weitere Abbildung. Damit enthält das Cluster nach dem Einfügen des dritten Terms zwei Abbildungen, da die drei Terme mit zwei Abbildungen auf des globale Objekt abgebildet werden können. Sei Ac die Anzahl der Abbildungen eines Clusters und Tc die Anzahl der Terme desselben Clusters, so lassen sich die Abbildungen eines Clusters wie folgt berechnen.
$Ac = Tc - 1$

Richtig Positiv

Ein richtiger Positiv ist, wenn zwei Terme, welche sich in dem manuell erstellten Ergebnis in dem selben Cluster befinden, sich in dem selben Cluster des automatisch generierten Ergebnisses befinden. In diesem Fall wird angenommen, dass diese beiden Terme von dem Werkzeug bewusst in das selbe Cluster geführt wurden und eine richtige Abbildung stattgefunden hat.

Falsch Positiv

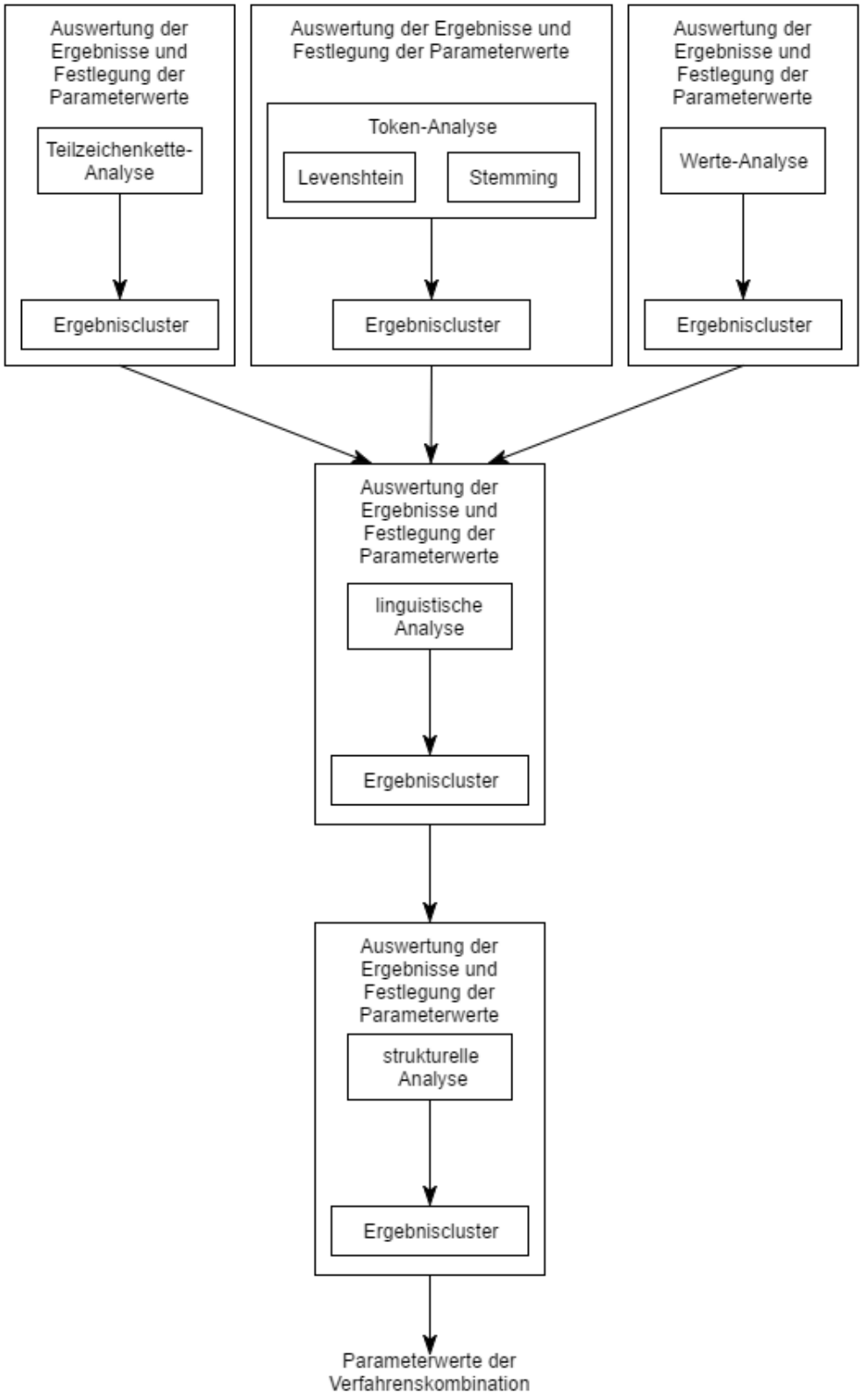

Abbildung 6.2.: Prozessablauf der Parameterfindung.

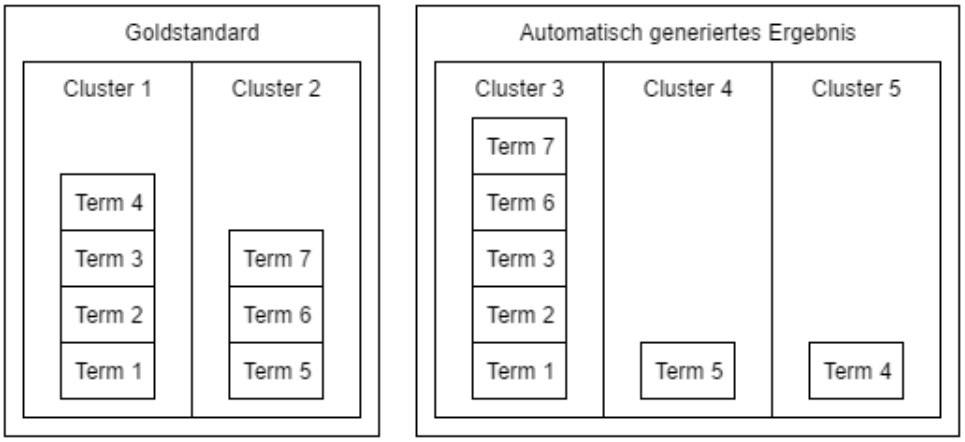

Abbildung 6.3.: Beispiel für die Bestimmung von richtigen und falschen Positiven.

> Die falschen Positive sind alle Abbildungen des automatisch generierten Ergebnisses, welche keine richtig Positive sind. Für die Bestimmung der falschen Positive werden alle richtigen Positive von den gesamten Abbildungen des automatisch erstellten Ergebnisses abgezogen.

Um die richtigen und falschen Positive zu veranschaulichen, wird ein Beispiel anhand von Abbildung 6.3 demonstriert. Der Goldstandard enthält insgesamt fünf Abbildungen, wobei drei Abbildungen in Cluster eins und zwei Abbildungen in Cluster 2 zu finden sind. Der Grund hierfür ist, dass Cluster 1 aus genau drei Ähnlichkeiten und Cluster 2 aus zwei Ähnlichkeiten zusammengeführt wird. Das automatisch generierte Ergebnis enthält im Gegensatz nur vier Abbildungen, da sowohl in Cluster 4 als auch in Cluster 5 keine Abbildungen vorhanden sind. Folglich wird ausschließlich Cluster 3 für die Auswertung der richtigen und falschen Positive betrachtet. Cluster 3 enthält insgesamt drei richtige Positive. Term 1,2 und 3 bilden die ersten beiden richtigen Positive, da diese sich auch in dem Goldstandard in demselben Cluster befinden und mithilfe von zwei Abbildungen richtig zusammengeführt wurden. Term 6 und 7 bilden einen weiteren richtigen Positiv. Zusätzlich enthält Cluster 3 ein falsch Positiv, welche die Terme 1,2 und 3 mit den Termen 6 und 7 zu einem Cluster zusammengeführt hat. Dies wird berechnet indem die drei richtigen Positive von allen Abbildungen des automatisch generierten Ergebnisses abgezogen werden.
Wurden alle richtigen und falschen Positive bestimmt, werden diese in ein Verhältnis gesetzt, durch welches die Ergebnisse ebenfalls mit anderen Testmengen verglichen werden können. Dazu wird zunächst berechnet, wie viel Prozent der Abbildungen des manuell erstellten Ergebnisses gefunden wurden. Es wird die Anzahl der richtigen Positive durch die Anzahl der Abbildungen des manuell erstellten Ergebnisses geteilt. In diesem Beispiel wird das Ergebnis wie folgt berechnet:
$3/5 = 0,6$
Dies bedeutet, dass 60% der Abbildungen in diesem Test gefunden wurden. Als Nächstes wird berechnet, wie viel Prozent der gefundenen Abbildungen falsch waren. Dazu werden die falschen Positiv durch alle Abbildungen des automatisch generierten Ergebnisses geteilt. In diesem Beispiel wird dies wie folgt berechnet berechnet:
$1/4 = 0,25$
Dies bedeutet, dass 25% der gefundenen Abbildungen dieses Tests falsch waren. Verfahren oder Verfahrenskombinationen haben eine höhere Präzision, wenn der Anteil der richtigen Positive möglichst hoch und der Anteil der falschen Positive möglichst gering ist. Im Folgenden werden die verwendeten Test- und Trainingsmengen vorgestellt und deren Ergebnisse analysiert.

Bahn	**Hotelketten**
bodo	GCH
DING	Hilton
HVV	Kempinski
KVV	nH
RMV	NOVOTEL
VBB	starwood
VBN	STEIGENBERGER
VNN	TRUMP HOTELS
VRN	Best Western

Tabelle 6.3.: Verwendete Dienstanbieter der Testmengen

Eigenschaft	**Bahn**	**Hotelketten**
Terme insgesamt	52	85
Cluster insgesamt	10	41
Cluster mit nur einem Element	3	28
Cluster mit mehr als einem Element	7	13
Abbildungen	42	44

Tabelle 6.4.: Zusätzliche Informationen der Testmengen

6.2. Trainings- und Testmengen

In diesem Abschnitt werden zunächst die Trainings-, sowie die beiden Testmengen vorgestellt. Die Trainingsmenge ist von der Dienstkategorie Fluganbieter. Für diese wurden die 3 Formulare der Dienstanbieter *South African Airways*, *Air New Zealand* und *British Airways* aus der Arbeit „*Abbildung von formularbasierten Internetdiensten auf aktive Ontologien*“ [Was16] von *Wasim Said* ausgewählt. Nachdem die Parameterwerte der Verfahrenskombinationen anhand dieser Trainingsmenge bestimmt wurden, wurden 2 Testmengen mit jeweils 10 Dienstanbietern der Dienstkategorien Bahn und Hotelketten erstellt. Die Testmengen befinden sich in einer anderen Dienstkategorie als die Trainingsmenge, um zu überprüfen, ob die Parameterwerte auch in weiteren Dienstkategorien gute Ergebnisse erzielen können. Die verschiedenen Dienstanbieter sind in Tabelle 6.3 aufgelistet. Zusätzlich sind Informationen über die beiden Testmengen in Tabelle 6.4 dargestellt. Wie zu sehen ist, enthält das manuell erstellte Ergebnis der Dienstkategorie Hotelkette wesentlich mehr Cluster als die erste Testmenge. Dennoch besitzen beide fast gleich viele Abbildungen. Dies lässt sich anhand der größeren Anzahl von einelementigen Clustern der Hotelketten Testmenge erklären. Einelementige Cluster besitzen keine Abbildungen, werden aber dennoch als Cluster gewertet.

6.3. Auswertung und Ergebnisse

Nachdem das Cluster-Verfahren mit jeder Verfahrenskombination an beiden Testmengen durchgeführt wurde, werden die Ergebnisse ausgewertet. Es werden sowohl die Ergebnisse der einzelnen Verfahrenskombinationen ausgewertet, als auch der Nutzen der einzelnen Verfahren. Dazu werden zunächst die Ergebnisse der einzelnen Testmengen und anschließend die Ergebnisse beider Testmengen betrachtet. Anschließend werden die einzelnen Verfahren durch Vergleiche bestimmter Verfahrenskombinationen ausgewertet.

6.3.1. Ergebnisse der Verfahrenskombinationen

Für die Auswertung der Ergebnisse wird der Anteil der falschen und richtigen Positive betrachtet. Wobei der Anteil der richtigen Positive im Verhältnis zu den manuell erstellten

Höchster Anteil an richtigen Positiv der Bahn Testmenge		
Kombination	**Anteil falscher Positive**	**Anteil richtiger Positive**
11	25,6%	76,2%
12	25,6%	76,2%
18	5,9 %	76,2%
24	22%	76,2%
25	22 %	76,2%
Bestes Verhältnis zwischen richtigen und falschen Positiven		
Kombination	**Anteil falsche Positive**	**Anteil richtige Positive**
18	5,9 %	76,2%

Tabelle 6.5.: Beste Ergebnisse Bahn

Höchster Anteil an richtigen Positven der Hotelketten Testmenge		
Kombination	**Anteil falscher Positive**	**Anteil richtiger Positive**
7	29,3%	65,9%
8	29,3%	65,9%
11	47,4%	65,9%
12	47,4%	65,9%
Bestes Verhältnis zwischen richtigen und falschen Positiven		
Kombination	**Anteil falscher Positive**	**Anteil richtiger Positive**
18	26,3%	63,6%
26	26,3%	63,6%
27	26,3%	63,6%

Tabelle 6.6.: Beste Ergebnisse Hotel

Ergebnissen und der Anteil der falschen Positive im Verhältnis zu den automatisch generierten Ergebnissen berechnet wird. Die Ergebnisse der Bahn Testmenge sind in Tabelle 6.5 und die Ergebnisse der Hotelketten Testmenge sind in Tabelle 6.6 aufgelistet.

Die Ergebnisse der Bahn Testmenge erzielen nicht nur einen höheren Anteil an richtigen Positiven, sondern auch einen geringeren Anteil an falschen Positiven. Das Ergebnis der Verfahrenskombination 18 aus der Bahn Testmenge stellt das beste Ergebnis dieser Evaluation dar. Die verwendeten Verfahren dieser Kombination sind eine strukturelle Analyse und eine Teilzeichenkette-Analyse. Die Testmenge der Hotelketten Kategorie enthält wesentlich schlechtere Ergebnisse. Beispielsweise entsteht bei der Verwendung von Verfahrenskombination 19 ein Anteil von über 74,2% an falschen Positiven. Den höchsten Anteil an falschen Positiven den eine Verfahrenskombination der Bahn Kategorie erzeugt sind 37,5%, welche die Kombinationen 13 und 14 erzielen. In beiden Testmengen erreicht die Verfahrenskombination 1, welche alle Verfahren verwendet, nicht das beste Ergebnis. Dies liegt daran, dass einige Verfahren in einer bestimmten Verfahrenskombination bessere Ergebnisse erzeugen, als allein oder mit anderen Verfahren zusammen. Dies wird später in diesem Kapitel weiter ausgeführt. In Tabelle 6.7 sind die besten Ergebnisse der gesamten Testmengen aufgelistet. Dazu wurde jeweils der Durchschnitt der Ergebnisse der beiden Testmengen berechnet.

Höchster Anteil an richtigen Positiven der Gesamten Menge		
Kombination	**Anteil falscher Positive**	**Anteil richtige Positive**
11	36,5%	71,1%
12	36,5%	71,1%
Bestes Verhältnis zwischen richtigen und falschen Positiven		
Kombination	**Anteil falscher Positive**	**Anteil richtiger Positive**
7	17,7%	66,5%
8	17,7%	66,5%
26	16,5%	65,2%
27	16,5%	65,2%

Tabelle 6.7.: Beste Ergebnisse gesamt

6.3.2. Diskussion der Ergebnisse

Die beiden Testmengen erzielten jeweils unterschiedliche Ergebnisse. Es geht hervor, dass bestimmte Verfahrenskombinationen in beiden Testmengen zu guten Ergebnissen führen und andere Kombination keine guten Resultate erzeugen. Die Verfahrenskombinationen 11,12 und 18 erzielen im insgesamt die besten Ergebnisse, sowohl in der Kategorie Hotel als auch in der Kategorie Bahn. Die Kombinationen 11 und 12 verwenden die Verfahren der Token-Analyse, Werte-Analyse und strukturelle Analyse. Durch das zusätzliche Verwenden des Stemming-Verfahrens in Kombination 11 wird jedoch kein verbessertes Ergebnis erzielt. Kombination 18 verwendet anstatt der Token- und Werte-Analyse eine Teilzeichenketten-Analyse. Verfahrenskombination 1 findet insgesamt durchschnittlich 56% der Abbildungen, wobei 27,1% der gefundenen Abbildungen falsche Positive sind. Dies zeigt, dass der Cluster-Algorithmus nicht durch das verwenden aller Verfahren das beste Ergebnis erzielt, sondern durch das benutzen von bestimmten Verfahrenskombinationen. Zusätzlich lässt sich sagen, dass jede Verfahrenskombination ähnliche Auswirkungen auf das Ergebnis von beiden Testmengen hat und die Ergebnisse somit auch auf weitere Testmengen übertragbar sind.

6.3.3. Bewertung der Verfahren

Um den Einfluss eines Verfahrens auf das Ergebnis bestimmen zu können, werden alle Verfahrenskombinationen-Paare verglichen, von denen eine Kombination das Verfahren verwendet und die andere nicht. Beispielsweise kann mithilfe von Verfahrenskombination 1 und 2 aus Tabelle 6.1 der Einfluss des Stemming-Verfahrens auf das Ergebnis betrachtet werden. Der prozentuale Zuwachs an richtigen und falschen Positiven ist der Zuwachs, den dieses Verfahren erbracht hat. Nimmt man den Durchschnitt der Resultate aller Verfahrenskombinations-Paare eines Verfahrens, so erhält man die durchschnittliche Zunahme an richtigen und falschen Positiven, den das Cluster-Verfahren erhält, wenn es dieses Verfahren zu einer Kombination hinzufügt. In Abbildung 6.4 sind die Ergebnisse aller Verfahren zu sehen. Dazu wurde jeweils der durchschnittliche Zuwachs, welchen dieses Verfahren an richtigen und falschen Positiven erbracht hat durch die ersten beiden Säulen dargestellt. Zusätzlich sind zwei weitere Säulen zu sehen, welche den höchsten gemessenen Zuwachs darstellen, welchen dieses Verfahren in einem Verfahrenskombinations-Paar erzielt hat.
Besonders auffällig sind die Werte des Stemming-Verfahrens. Es geht hervor, dass dieses Verfahren keinen Einfluss auf das Endergebnis des Cluster-Verfahrens hat.
Die Levenshtein Distanz und Teilzeichenkette-Analyse besitzen einen negativen Zuwachs

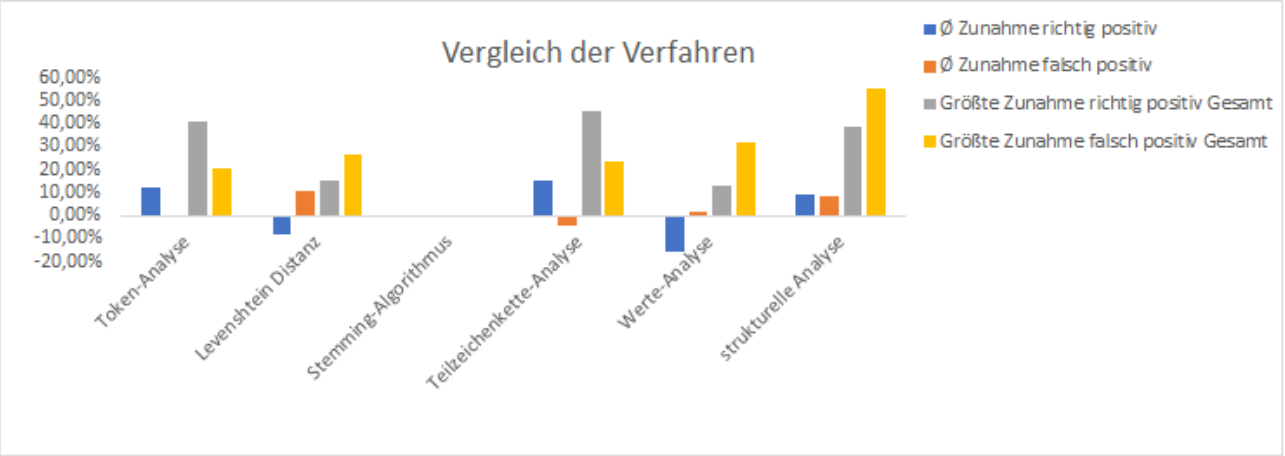

Abbildung 6.4.: Ein Säulendiagramm, welches die Zunahme der richtigen und falschen Positive der Verwendung von den einzelnen Verfahren darstellt.

an richtigen Positiven. Da auch die falschen Positive der Ergebnisse einen Zuwachs erhalten, wird das Ergebnis im Durchschnitt schlechter, wenn diese Verfahren verwendet werden.
Der Grund hierfür könnte sein, dass beide Verfahren dazu neigen nicht nur viele richtige Abbildungen zu finden, sondern auch viele falsche. Dies kann dazu führen, dass falsche Abbildungen über das Cluster-Verfahren zuerst einem Cluster zugeordnet werden. Dadurch ist es möglich, dass das Cluster-Verfahren eine richtige Abbildung dem Cluster nicht mehr zuweisen kann, weil die falsche Abbildung dies verhindert. Dies führt zu einer Verringerung der richtigen Abbildungen.
Die Token-Analyse, welche weder eine Levenshtein Distanz noch den Stemming-Algorithmus verwendet erzielt mit der Teilzeichenketten-Analyse die besten Ergebnisse. Diese Verfahren erhöhen nicht nur die richtigen Positive, sondern verringern im Durchschnitt auch den Anteil der falschen Positive. Das letzte Verfahren, die strukturelle Analyse, erreicht ausgeglichene Werte.
Auch wenn einige Verfahren zu einem verschlechterten Ergebnis führen können, ist an den Ergebnissen der Testmengen zu sehen, dass jedes Verfahren zu einem verbesserten Ergebnis beitragen kann. Das Cluster-Verfahren erzielt nicht durch einzelne Verfahren gute Ergebnisse, sondern mit bestimmten Kombinationen aus Verfahren. Bestimmte Verfahren können in bestimmten Kombinationen sehr gute Ergebnisse erreichen, während diese in einer anderen Kombination ein verschlechterndes Ergebnis erzielen können.

6.4. Zusammenfassung

Für die Evaluation des Werkzeugs wurden anhand von einer Trainingsmenge und 2 Testmengen Ergebnisse automatisch generiert. Dabei wurde die Präzision anhand von den gefundenen richtigen Abbildungen und den gefundenen falschen Abbildungen, mithilfe von manuell erstellten Ergebnissen, bestimmt. Um die einzelnen Verfahren zu testen, wurden die Tests anhand von 38 verschiedenen Kombinationen dieser durchgeführt. Das Werkzeug erzielte bei der Kombination, welche alle Verfahren verwendet, nicht das beste Ergebnis. Durch das geschickte Kombinieren von mehreren Verfahren ist es möglich die besten Ergebnisse zu erzielen. Das Werkzeug findet bei Wahl der richtigen Verfahrenskombination 60 - 70% der Abbildungen. Dabei sind zwischen 5% und 50% der gefundenen Abbildungen falsch. Der Stemming-Algorithmus hat keinen Einfluss auf das Ergebnis der Testmengen. Dies könnte jedoch an der Testmenge liegen. Beispielsweise, weil nicht genug Wörter von dem Algorithmus verändert werden konnten, um einen Einfluss auf das Ergebnis zu haben.

7. Zusammenfassung und Ausblick

Das manuelle Erstellen aktiver Ontologien ist ein komplexer und aufwendiger Prozess. Das Projekt Easier ist ein Ansatz diesen Prozess zu automatisieren und zu vereinfachen. Eine Komponente des Projektes ist das in dieser Arbeit erstellte Werkzeug.
Es erhält verschiedene Webformulare einer Dienstkategorie als Eingabe und erstellt aus diesen einen Konstruktionsplan. Dieser Konstruktionsplan dient als Vorlage der zu erstellenden aktiven Ontologie.
Im Folgenden werden zunächst eine Zusammenfassung und anschließend einige Ausblicke der Arbeit präsentiert.

7.1. Zusammenfassung

Bis zu der Erstellung des Konstruktionsplanes werden mehrere Schritte durchlaufen. Zunächst werden die HTML-Formulare in eine Baumstruktur überführt. Anschließend werden die Informationen der Formularelemente aufbereitet und Terme erstellt. Als Nächstes werden die semantischen Ähnlichkeiten der Terme bestimmt. Basierend auf diesen Ähnlichkeiten werden semantisch gleiche Terme mithilfe eines Cluster-Verfahrens zusammengeführt. Für jedes Cluster wird ein globales Element gebildet. Zum Schluss wird aus den Clustern und den globalen Elementen ein Konstruktionsplan für diese Dienstkategorie erstellt und als XML-Datei ausgegeben.
In der Evaluation wurden zunächst Parameterwerte bestimmt. Dabei wurde erkannt, dass das Werkzeug über 76,2% der Abbildungen finden kann, während 5,9% der Abbildungen falsch abgebildet wurden. Das Cluster-Verfahren kann verschiedene Analysemethoden verwenden, um ein Ergebnis zu erzeugen. Dabei ging hervor, dass bestimmte Kombinationen der Analysemethoden die besten Ergebnisse erzeugen. Schließlich lässt sich sagen, das es für beide Testmengen möglich war über 60% der richtigen Abbildungen zu finden und dabei weniger als 30% falsche Abbildungen zu generieren.
Zudem ist das Werkzeug auch für andere Dienstkategorien anwendbar. Obwohl das Werkzeug mit den gefundenen Parameterwerten für die getesteten Dienstkategorien gut funktioniert, könnten andere Parameterwerte für eine andere Dienstkategorie besser geeignet sein. In diesem Fall müsste eine erneute Durchführung der Parameterfindung für diese Dienstkategorie durchgeführt werden.

7.2. Ausblick: Komplexe Abbildungen

Das erstellte Werkzeug betrachtet lediglich einfache Abbildungen. Die Entwicklung von Methoden, welche auch komplexe Abbildungen erstellen können, würde das Cluster-Verfah-

ren signifikant verbessern. Dazu müsste auch der Typ der komplexen Abbildung bestimmt werden, um eine Rückwärtsabbildung zu ermöglichen. Die komplexen Abbildungen können vor dem Start des Cluster-Verfahrens erkannt und aussortiert werden. Anschließend werden diese zu den bestehenden Clustern hinzugefügt. Der Konstruktionsplan ermöglicht bereits das Einsetzen von komplexen Abbildungen.

7.3. Ausblick: Wörterbücher und Synonymerkennung

Ein Verfahren des Werkzeugs ist die Token-Analyse. Diese vergleicht die verschiedenen Wörter der Terme. Bisher werden die Wörter immer in ihrer Ausgangsform verglichen. Durch das einsetzten von Wörterbüchern ist es möglich gleiche Wörter zu erkennen, welche in verschiedenen Sprachen dargestellt werden. Zusätzlich können Synonyme hinzugefügt werden, um auch unterschiedliche Beschreibungen als gleich anzusehen. Dazu müsste zusätzlich die Cosinus Funktion des Verfahrens angepasst werden.

7.4. Ausblick: Verbesserung der Parametersuche

Ein wichtiges Kriterium für das Erzielen guter Ergebnisse dieses Werkzeugs ist eine gute Wahl der Parameterwerte. Aufgrund des Zeitrahmens dieser Arbeit konnten nur beschränkt viele Parameterkombinationen anhand einer Trainingsmenge getestet werden. Mithilfe von Parallelisierung und einer höheren Rechenleistung könnte die Parametersuche mit einigen Parameterkombinationen erweitert werden. Zusätzlich könnten weitere Trainingsmengen dazu verwendet werden, um die Parameterwerte möglichst unabhängig von einer Dienstkategorie bestimmen zu können.

Literaturverzeichnis

[AG04] Avigdor Gal, Hasan J. Giovanni Modica M. Giovanni Modica: OntoBuilder: Fully Automatic Extraction and Consolidation of Ontologies from Web Sources. ICDE Conference : IEEE, 2004 (zitiert auf Seite 25).

[AG05] Avigdor Gal, Hasan Jamil Ami E. Giovanni Modica M. Giovanni Modica: Automatic Ontology Matching Using Application Semantics. In: *AI MAGAZINE* 26 (2005), Nr. 1 (zitiert auf den Seiten 25, 28, 31, 33 und 35).

[BL16] Blersch, M. ; Landhäuser, M.: *EASIER: An Approach to Automatically Generate Active Ontologies for Intelligent Assistants*. The 20th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2016) Orlando, FL, USA 05.07.2016 DOI: 10.13140/RG.2.1.2586.9043, Juli 2016 (zitiert auf Seite 1).

[Gru93] Gruber, Thomas R.: A Translation Approach to Portable Ontology Specification / Stanford University. 1993. – Forschungsbericht (zitiert auf Seite 13).

[Guz08] Guzzoni, Didier: *Active: A Unified Platform for Building intelligent Applications*, École polytechnique fédérale de Lausanne, dissertation, 2008. `http://biblion.epfl.ch/EPFL/theses/2008/3990/3990_abs.pdf` (zitiert auf den Seiten xi, 1, 13 und 14).

[HHYW03] Hai He, Weiyi M. (Hrsg.) ; Yu, Clement (Hrsg.) ; Wu, Zonghuan (Hrsg.): *WISE-Integrator: An Automatic Integrator of Web Search Interfaces for E-Commerce*. Bd. *29*. VLDB Endowment, 09 2003 (zitiert auf den Seiten 22, 37 und 38).

[HHYW05] Hai He, Weiyi M. (Hrsg.) ; Yu, Clement (Hrsg.) ; Wu, Zonghuan (Hrsg.): *WISE-Integrator: A System for Extracting and Integrating Complex Web Search Interfaces of the Deep Web*. VLDB Endowment, 08 2005 (zitiert auf den Seiten 22, 28 und 39).

[htm99] *HTML 4.01 Specification*. 12 1999. – Online erhältlich unter `https://www.w3.org/TR/html4/`; abgerufen am 10. Oktober 2016. (zitiert auf Seite 5).

[htm14] *A vocabulary and associated APIs for HTML and XHTML*. 10 2014. – Online erhältlich unter `https://www.w3.org/TR/html5/`; abgerufen am 10. Oktober 2016. (zitiert auf Seite 5).

[JM01] Jayant Madhavan, Erhard R. Phil Bernstein B. Phil Bernstein: Generic Schema Matching With Cupid / Microsoft Research. 2001 (MSR-TR-2001-58). – TechReport (zitiert auf den Seiten 23, 28, 31 und 35).

[MFBB10] Maiz, Nora (Hrsg.) ; Fahad, Muhammad (Hrsg.) ; Boussaid, Omar (Hrsg.) ; Bentayeb, Fadila (Hrsg.): *Automatic Ontology Merging by Hierarchical Clustering and Inference Mechanisms*. 2010 (zitiert auf Seite 25).

[NG09] Nicola Guarino, Steffen S. Daniel Oberle O. Daniel Oberle: *What Is an Ontology?* 2009. – Online erhältlich unter `http://iaoa.org/isc2012/docs/Guarino2009_What_is_an_Ontology.pdf`; abgerufen am 10. Oktober 2016. (zitiert auf Seite 13).

[Tur06] Turk Žiga: Construction informatics: Definition and ontology. In: *Advanced Engineering Informatics* 20 (2006), Nr. 2, S. 187–199 (zitiert auf Seite 13).

[Was16] Wasim, Said: *Abbildung von Webformularen auf aktive Ontologien.* Germany, Karlsruher Institut für Technologie, Lehrstuhl IPD Tichy, Masterarbeit, April 2016 (zitiert auf Seite 58).

[WYDM04] Wu, Wensheng ; Yu, Clement ; Doan, AnHai ; Meng, Weiyi: An Interactive Clustering-based Approach to Integrating Source Query Interfaces on the Deep Web. In: *Proceedings of the 2004 ACM SIGMOD international conference on Management of data Table of Contents*, ACM New York, NY, USA 2004, JUL 2004, S. 95–106 (zitiert auf den Seiten xi, 19, 31, 32, 33, 34, 35, 36, 37 und 43).

[xml06] *Extensible Markup Language (XML) 1.1 (Second Edition).* 08 2006. – Online erhältlich unter `https://www.w3.org/TR/xml11/`; abgerufen am 10. Oktober 2016. (zitiert auf Seite 11).

[YA12] Yuan An, Il-Yeol S. Xiaohua Hu H. Xiaohua Hu (Hrsg.): *Learning to Discover Complex Mappings from Web Forms to Ontologies.* CIKM '12 Proceedings of the 21st ACM international conference on Information and knowledge management, 10 2012 (zitiert auf Seite 24).

Anhang

A. First Appendix Section

In den folgenden Tabellen sind alle Ergebnisse der Evaluation aufgelistet. Dabei steht die Spalte *negative* für den prozentualen Anteil der falschen Positive und die Spalte *positive* für den prozentualen Anteil der richtigen Positive eines Testdurchlaufs.

Bahn			Hotel		
Kombination	negative	positive	Kombination	negative	positive
1	0,188	0,619	1	0,353	0,5
2	0,188	0,619	2	0,353	0,5
3	0,286	0,714	3	0,627	0,5
4	0,286	0,714	4	0,627	0,5
5	0,262	0,738	5	0,469	0,591
6	0,262	0,738	6	0,469	0,591
7	0,061	0,738	7	0,293	0,659
8	0,061	0,738	8	0,293	0,659
9	0,222	0,333	9	0,4	0,136
10	0,222	0,333	10	0,4	0,136
11	0,256	0,762	11	0,473	0,659
12	0,256	0,762	12	0,473	0,659
13	0,375	0,595	13	0,528	0,386
14	0,375	0,595	14	0,528	0,386
15	0,148	0,548	15	0,289	0,614
16	0,148	0,548	16	0,289	0,614
17	0,293	0,69	17	0,667	0,432
18	0,059	0,762	18	0,263	0,636
19	0,357	0,429	19	0,742	0,182
20	0,071	0,619	20	0,31	0,455
21	0,059	0,318	21	0,32	0,386
22	0,071	0,619	22	0,31	0,455
23	0,059	0,381	23	0,32	0,386
24	0,22	0,762	24	0,468	0,568
25	0,22	0,762	25	0,468	0,568
26	0,067	0,667	26	0,263	0,636
27	0,067	0,667	27	0,263	0,636
28	0,267	0,262	28	0,545	0,227
29	0,267	0,262	29	0,545	0,227
30	0,143	0,286	30	0,273	0,364
31	0,143	0,286	31	0,273	0,364
32	0,341	0,643	32	0,63	0,386
33	0,341	0,643	33	0,63	0,386
34	0,071	0,619	34	0,372	0,614
35	0,071	0,619	35	0,372	0,614
36	0	0,5	36	0,471	0,409
37	0,067	0,667	37	0,257	0,591
38	0	0	38	0	0

Abbildung A.1.: Positive und negative Abbildungen der Dienstkategorien Bahn und Hotel

Gesamt			
Kombination	negative	positive	Verhältnis
1	0,2705	0,5595	2,06839187
2	0,2705	0,5595	2,06839187
3	0,4565	0,607	1,32968237
4	0,4565	0,607	1,32968237
5	0,3655	0,6645	1,81805746
6	0,3655	0,6645	1,81805746
7	0,177	0,6985	3,94632768
8	0,177	0,6985	3,94632768
9	0,311	0,2345	0,75401929
10	0,311	0,2345	0,75401929
11	0,3645	0,7105	1,94924554
12	0,3645	0,7105	1,94924554
13	0,4515	0,4905	1,08637874
14	0,4515	0,4905	1,08637874
15	0,2185	0,581	2,6590389
16	0,2185	0,581	2,6590389
17	0,48	0,561	1,16875
18	0,161	0,699	4,34161491
19	0,5495	0,3055	0,55595996
20	0,1905	0,537	2,81889764
21	0,1895	0,352	1,85751979
22	0,1905	0,537	2,81889764
23	0,1895	0,3835	2,0237467
24	0,344	0,665	1,93313953
25	0,344	0,665	1,93313953
26	0,165	0,6515	3,94848485
27	0,165	0,6515	3,94848485
28	0,406	0,2445	0,60221675
29	0,406	0,2445	0,60221675
30	0,208	0,325	1,5625
31	0,208	0,325	1,5625
32	0,4855	0,5145	1,05973223
33	0,4855	0,5145	1,05973223
34	0,2215	0,6165	2,78329571
35	0,2215	0,6165	2,78329571
36	0,2355	0,4545	1,92993631
37	0,162	0,629	3,88271605
38	0	0	#DIV/0!
Summe	11,238	19,5845	
Durchschnitt	0,29573684	0,51538158	

Abbildung A.2.: Positive und negative Abbildungen beider Testmengen im Durchschnitt

	Bahn	Hotel
Kombination	Verhältnis	Verhältnis
1	3,29255319	1,41643059
2	3,29255319	1,41643059
3	2,4965035	0,79744817
4	2,4965035	0,79744817
5	2,81679389	1,26012793
6	2,81679389	1,26012793
7	12,0983607	2,24914676
8	12,0983607	2,24914676
9	1,5	0,34
10	1,5	0,34
11	2,9765625	1,39323467
12	2,9765625	1,39323467
13	1,58666667	0,73106061
14	1,58666667	0,73106061
15	3,7027027	2,12456747
16	3,7027027	2,12456747
17	2,35494881	0,64767616
18	12,9152542	2,41825095
19	1,20168067	0,24528302
20	8,71830986	1,46774194
21	5,38983051	1,20625
22	8,71830986	1,46774194
23	6,45762712	1,20625
24	3,46363636	1,21367521
25	3,46363636	1,21367521
26	9,95522388	2,41825095
27	9,95522388	2,41825095
28	0,98127341	0,41651376
29	0,98127341	0,41651376
30	2	1,33333333
31	2	1,33333333
32	1,8856305	0,61269841
33	1,8856305	0,61269841
34	8,71830986	1,65053763
35	8,71830986	1,65053763
36	#DIV/0!	0,86836518
37	9,95522388	2,29961089
38	#DIV/0!	#DIV/0!

Abbildung A.3.: Verhältnisse zwischen den positiven und negativen Abbildungen der Dienstkategorien Bahn und Hotel

Token mit Levenshtein		Bahn		Hotel		Gesamt	
an	aus	positive	negative	positive	negative	positve	negative
1	3	-0,095	-0,098	0	-0,274	-0,0475	-0,186
2	4	-0,095	-0,098	0	-0,274	-0,0475	-0,186
5	7	0	0,201	-0,068	0,176	-0,034	0,1885
6	8	0	0,201	-0,068	0,176	-0,034	0,1885
9	11	-0,429	-0,034	-0,523	-0,073	-0,476	-0,0535
10	12	-0,429	-0,034	-0,523	-0,073	-0,476	-0,0535
13	15	0,047	0,227	-0,228	0,239	-0,0905	0,233
14	16	0,047	0,227	-0,228	0,239	-0,0905	0,233
20	21	0,301	0,012	0,069	-0,01	0,185	0,001
22	23	0,238	0,012	0,069	-0,01	0,1535	0,001
24	26	0,095	0,153	-0,068	0,205	0,0135	0,179
25	27	0,095	0,153	-0,068	0,205	0,0135	0,179
28	30	-0,024	0,124	-0,137	0,272	-0,0805	0,198
29	31	-0,024	0,124	-0,137	0,272	-0,0805	0,198
32	34	0,024	0,27	-0,228	0,258	-0,102	0,264
33	35	0,024	0,27	-0,228	0,258	-0,102	0,264
	Summe	-0,225	1,71	-2,366	1,586	0,705	1,635
	Durchscnitt	-0,0140625	0,106875	-0,147875	0,099125	-0,08096875	0,103

Abbildung A.4.: Zunahme der Abbildungen der Levenshtein Distanz

Stemming-Verfahren		Bahn		Hotel		Gesamt	
an	aus	positive	negative	positive	negative	positive	negative
1	2	0	0	0	0	0	0
3	4	0	0	0	0	0	0
5	6	0	0	0	0	0	0
7	8	0	0	0	0	0	0
9	10	0	0	0	0	0	0
11	12	0	0	0	0	0	0
13	14	0	0	0	0	0	0
15	16	0	0	0	0	0	0
20	22	0	0	0	0	0	0
21	23	0	0	0	0	0	0
24	25	0	0	0	0	0	0
26	27	0	0	0	0	0	0
28	29	0	0	0	0	0	0
30	31	0	0	0	0	0	0
32	33	0	0	0	0	0	0
34	35	0	0	0	0	0	0
	Summe	0	0	0	0	0	0

Abbildung A.5.: Zunahme der Abbildungen des Stemming-Verfahrens

Teilzeichenkette		Bahn		Hotel		Gesamt	
an	aus	positive	negative	positive	negative	positive	negative
1	9	0,286	-0,034	0,364	-0,047	0,325	-0,0405
2	10	0,286	-0,034	0,364	-0,047	0,325	-0,0405
3	11	-0,048	0,03	-0,159	0,154	-0,1035	0,092
4	12	-0,048	0,03	-0,159	0,154	-0,1035	0,092
5	13	0,143	-0,113	0,205	-0,059	0,174	-0,086
6	14	0,143	-0,113	0,205	-0,059	0,174	-0,086
7	15	0,19	-0,087	0,045	0,004	0,1175	-0,0415
8	16	0,19	-0,087	0,045	0,004	0,1175	-0,0415
17	19	0,261	-0,064	0,25	-0,075	0,2555	-0,0695
20	28	0,357	-0,196	0,228	-0,235	0,2925	-0,2155
21	30	0,032	-0,084	0,022	0,047	0,027	-0,0185
22	29	0,357	-0,196	0,228	-0,235	0,2925	-0,2155
23	31	0,095	-0,084	0,022	0,047	0,0585	-0,0185
24	32	0,119	-0,121	0,182	-0,162	0,1505	-0,1415
25	33	0,119	-0,121	0,182	-0,162	0,1505	-0,1415
26	34	0,048	-0,004	0,022	-0,109	0,035	-0,0565
27	35	0,048	-0,004	0,022	-0,109	0,035	-0,0565
36	38	0,5	0	0,409	0,471	0,4545	0,2355
	Summe	3,078	-1,282	2,477	-0,418	3,855	4,632
	Durchschnitt	0,171	-0,0712222	0,13761111	-0,0232222	0,15430556	-0,0472222

Abbildung A.6.: Zunahme der Abbildungen der Teilzeichenketten-Analyse

Werte-Analyse		Bahn		Hotel		Gesamt	
an	aus	positive	negative	positive	negative	positive	negative
1	5	-0,119	-0,074	-0,091	-0,116	-0,105	-0,095
2	6	-0,119	-0,074	-0,091	-0,116	-0,105	-0,095
3	7	-0,024	0,225	-0,159	0,334	-0,0915	0,2795
4	8	-0,024	0,225	-0,159	0,334	-0,0915	0,2795
9	13	-0,262	-0,153	-0,25	-0,128	-0,256	-0,1405
10	14	-0,262	-0,153	-0,25	-0,128	-0,256	-0,1405
11	15	0,214	0,108	0,045	0,184	0,1295	0,146
12	16	0,214	0,108	0,045	0,184	0,1295	0,146
17	18	-0,072	0,234	-0,204	0,404	-0,138	0,319
20	24	-0,143	-0,149	-0,113	-0,158	-0,128	-0,1535
21	26	-0,349	-0,008	-0,25	0,057	-0,2995	0,0245
22	25	-0,143	-0,149	-0,113	-0,158	-0,128	-0,1535
23	27	-0,286	-0,008	-0,25	0,057	-0,268	0,0245
28	32	-0,381	-0,074	-0,159	-0,085	-0,27	-0,0795
29	33	-0,381	-0,074	-0,159	-0,085	-0,27	-0,0795
30	34	-0,333	0,072	-0,25	-0,099	-0,2915	-0,0135
31	35	-0,333	0,072	-0,25	-0,099	-0,2915	-0,0135
36	37	-0,167	-0,067	-0,182	0,214	-0,1745	0,0735
	Summe	-2,97	0,061	-2,84	0,596	-5,153	-7,336
	Durchschnitt	-0,165	0,00338889	-0,1577778	0,03311111	-0,1613889	0,01825

Abbildung A.7.: Zunahme der Abbildungen der Werte-Analyse

strukturelle Analyse		Bahn		Hotel		Gesamt	
an	aus	positive	negative	positive	negative	positive	negative
1	20	0	0,117	0,045	0,043	0,0225	0,08
2	22	0	0,117	0,045	0,043	0,0225	0,08
3	21	0,396	0,227	0,114	0,307	0,255	0,267
4	23	0,333	0,227	0,114	0,307	0,2235	0,267
5	24	-0,024	0,042	0,023	0,001	-0,0005	0,0215
6	25	-0,024	0,042	0,023	0,001	-0,0005	0,0215
7	26	0,071	-0,006	0,023	0,03	0,047	0,012
8	27	0,071	-0,006	0,023	0,03	0,047	0,012
9	28	0,071	-0,045	-0,091	-0,145	-0,01	-0,095
10	29	0,071	-0,045	-0,091	-0,145	-0,01	-0,095
11	30	0,476	0,113	0,295	0,2	0,3855	0,1565
12	31	0,476	0,113	0,295	0,2	0,3855	0,1565
13	32	-0,048	0,034	0	-0,102	-0,024	-0,034
14	33	-0,048	0,034	0	-0,102	-0,024	-0,034
15	34	-0,071	0,077	0	-0,083	-0,0355	-0,003
16	35	-0,071	0,077	0	-0,083	-0,0355	-0,003
17	36	0,19	0,293	0,023	0,196	0,1065	0,2445
18	37	0,095	-0,008	0,045	0,006	0,07	-0,001
19	38	0,429	0,357	0,182	0,742	0,3055	0,5495
	Summe	2,393	1,76	1,068	1,446	6,667	10,941
	Durchschnitt	0,12594737	0,09263158	0,05621053	0,07610526	0,09107895	0,08436842

Abbildung A.8.: Zunahme der Abbildungen der strukturellen Analyse

Token-Analyse		Bahn		Hotel		Gesamt	
an	aus	positive	negative	positive	negative	positive	negative
4	17	0,024	-0,007	0,068	-0,04	0,046	-0,0235
8	18	-0,024	0,002	0,023	0,03	-0,0005	0,016
12	19	0,333	-0,101	0,477	-0,269	0,405	-0,185
23	36	-0,119	0,059	-0,023	-0,151	-0,071	-0,046
27	37	0	0	0,045	0,006	0,0225	0,003
31	38	0,286	0,143	0,364	0,273	0,325	0,208
	Summe	0,5	0,096	0,954	-0,151	0,727	-0,0275
	Durchschnitt	0,08333333	0,016	0,159	-0,0251667	0,12116667	-0,0045833

Abbildung A.9.: Zunahme der Abbildungen der Token-Analyse