

# Bestimmung von Aktionsidentität in gesprochener Sprache

Bachelorarbeit  
von

**Christian Bitterwolf**

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

Erstgutachter:	Prof. Dr. Walter F. Tichy
Zweitgutachter:	Prof. Dr. Ralf H. Reussner
Betreuender Mitarbeiter:	Dipl.-Inform. Sebastian Weigelt
Zweiter betr. Mitarbeiter:	Tobias Hey, M.Sc.

Bearbeitungszeit: 05.01.2018 – 04.05.2018



---

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, 04.05.2018**

.....  
**(Christian Bitterwolf)**



## **Kurzfassung**

Natürliche Sprache enthält Aktionen, die ausgeführt werden können. Innerhalb eines Diskurses kommt es häufig vor, dass Menschen eine Aktion mehrmals beschreiben. Dies muss nicht immer bedeuten, dass diese Aktion auch mehrmals ausgeführt werden soll. Diese Bachelorarbeit untersucht, wie erkannt werden kann, ob sich eine Nennung einer Aktion auf eine bereits genannte Aktion bezieht. Es wird ein Vorgehen erarbeitet, das feststellt, ob sich mehrere Aktionsnennungen in gesprochener Sprache auf dieselbe Aktionsidentität beziehen. Bei diesem Vorgehen werden Aktionen paarweise verglichen. Das Vorgehen wird als Agent für die Rahmenarchitektur PARSE umgesetzt und evaluiert. Das Werkzeug erzielt ein  $F_1$ -Maß von 0,8, wenn die Aktionen richtig erkannt werden und Informationen über Korreferenz zwischen Entitäten zur Verfügung stehen.



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Zielsetzung dieser Arbeit . . . . .	1
1.2. Aufbau der Arbeit . . . . .	2
<b>2. Grundlagen</b>	<b>3</b>
2.1. Natürliche Sprache . . . . .	3
2.1.1. Syntax . . . . .	3
2.1.2. Semantik . . . . .	3
2.1.3. Pragmatik . . . . .	4
2.1.4. Wortarten . . . . .	4
2.1.5. Lemma . . . . .	4
2.1.6. Korreferenz . . . . .	4
2.1.7. Prädikat . . . . .	4
2.1.8. Semantische Relationen . . . . .	5
2.1.8.1. Synonyme . . . . .	5
2.1.8.2. Hyponymie und Hyperonymie . . . . .	5
2.1.8.3. Troponymie . . . . .	5
2.2. Natürlichen Sprachverarbeitung . . . . .	5
2.2.1. Wortart-Markierung . . . . .	5
2.2.2. Phrasenerkennung . . . . .	6
2.2.3. Semantische Rollenzuteilung . . . . .	7
2.3. WordNet . . . . .	7
<b>3. PARSE</b>	<b>9</b>
3.1. Entwurfsprinzipien . . . . .	9
3.2. Architektur . . . . .	9
3.2.1. Seichte Sprachverarbeitung . . . . .	10
3.2.2. Aktionserkennung . . . . .	10
3.2.3. Kontextanalysierer . . . . .	10
3.2.4. Korreferenzauflöser . . . . .	11
3.2.5. Bedinungsanalysierers . . . . .	11
3.2.6. Dialogmanagement . . . . .	12
<b>4. Verwandte Arbeiten</b>	<b>13</b>
4.1. Korpora . . . . .	13
4.1.1. ACE . . . . .	13
4.1.2. ECB . . . . .	14
4.1.3. IC . . . . .	14
4.2. Ansätze zur Ereigniskorreferenzauflösung . . . . .	15
4.2.1. Cross-Document Event Coreference: Annotations, Experiments, and Observations . . . . .	15
4.2.2. Graph-based Event Coreference Resolution . . . . .	16

4.2.3.	Joint Entity and Event Coreference Resolution across Documents . . .	16
4.2.4.	Detecting Subevent Structure for Event Coreference Resolution . . .	17
4.2.5.	Supervised Within-Document Event Coreference using Information Propagation . . . . .	18
4.2.6.	Event Coreference Resolution by Iteratively Unfolding Inter-dependencies among Events . . . . .	19
4.3.	Zusammenfassung . . . . .	20
<b>5.</b>	<b>Analyse</b>	<b>21</b>
5.1.	Definition von Aktionskorreferenz . . . . .	21
5.2.	Erkennung von Konzeptioneller Gleichheit . . . . .	23
5.2.1.	Prädikatengleichheit . . . . .	23
5.2.2.	Korreferenz zwischen Akteuren . . . . .	25
5.2.3.	Korreferenz zwischen Objekten . . . . .	25
5.2.4.	Gleichheit sonstiger Parameter . . . . .	27
5.2.4.1.	Parameter der Zeit . . . . .	27
5.2.4.2.	Parameter des Raums . . . . .	27
5.2.4.3.	Umstandsparameter . . . . .	28
5.3.	Erkennung von Aktionsidentität . . . . .	28
5.3.1.	Temporale, lokale, und kausale Beziehungen . . . . .	28
5.3.1.1.	Erkennung von Strukturen zur Einordnung . . . . .	28
5.3.1.2.	Betrachtung der Zeitformen . . . . .	29
5.3.1.3.	Korreferenz zwischen Aktionen in Haupt- und Nebensatz . . . . .	29
5.3.1.4.	Aktionskorreferenz in Kontrollstrukturen . . . . .	30
5.3.2.	Relativsätze . . . . .	30
5.3.3.	Spezifizierung und Generalisierung . . . . .	31
5.3.4.	Korrektur von falschen Aktionen . . . . .	32
5.3.5.	Ausschlusskriterien für Aktionskorreferenz . . . . .	32
5.4.	Spezialfälle . . . . .	33
5.4.1.	Korreferenz zwischen Aktion und Demonstrativpronomen . . . . .	33
5.4.2.	Korreferenz zwischen Aktion und Substantivierung . . . . .	33
5.4.3.	Partielle Aktionskorreferenz . . . . .	34
<b>6.</b>	<b>Entwurf und Implementierung</b>	<b>35</b>
6.1.	Ansatz zur Auflösung von Aktionskorreferenz . . . . .	35
6.2.	Paarweiser Vergleich der Aktionen . . . . .	36
6.2.1.	Erkennung von konzeptioneller Gleichheit . . . . .	36
6.2.1.1.	Vergleich der Prädikate . . . . .	37
6.2.1.2.	Vergleich der Akteure . . . . .	37
6.2.1.3.	Vergleich der Objekte . . . . .	38
6.2.1.4.	Vergleich der sonstigen Parameter . . . . .	39
6.2.1.5.	Spezialfälle . . . . .	40
6.2.2.	Erkennung von Aktionsidentität . . . . .	40
6.2.2.1.	Temporale, lokale und kausale Beziehungen . . . . .	41
6.2.2.2.	Spezifizierungen . . . . .	41
6.3.	Implementierungsdetails . . . . .	42
6.3.1.	Eingabe . . . . .	42
6.3.2.	Hauptklasse: ActionCorefResolver . . . . .	44
6.3.3.	Repräsentation von Aktionsnennungen . . . . .	45
6.3.4.	Vergleich zweier Aktionen . . . . .	45
6.3.5.	Erkennung semantischer Relationen . . . . .	46
6.3.6.	Darstellung der Ergebnisse . . . . .	46

<b>7. Evaluation</b>	<b>49</b>
7.1. Evaluationskorporus . . . . .	49
7.2. Evaluationsvorgang . . . . .	51
7.2.1. Bewertungsmetriken . . . . .	51
7.2.2. Evaluationsvarianten . . . . .	52
7.3. Evaluationsergebnisse . . . . .	52
7.3.1. Variante 1 . . . . .	52
7.3.2. Variante 2 . . . . .	54
7.3.3. Variante 3 . . . . .	55
7.3.4. Fehler durch den Agenten . . . . .	56
7.3.4.1. Zufällige semantische Beziehungen zwischen unterschiedli-	
chen Bedeutungsausprägungen . . . . .	56
7.3.4.2. Falsche Annahmen über Verwendung von Zeitformen . . . . .	57
7.3.4.3. Mehrdeutigkeit des Wortes <i>do</i> . . . . .	57
7.3.4.4. Spezifizierung und <i>do that</i> bei nicht direkt aufeinanderfol-	
genden Aktionen . . . . .	57
7.3.4.5. Fehlende Parameter in Aktionsnennungen . . . . .	57
7.4. Bewertung der Ergebnisse . . . . .	58
<b>8. Zusammenfassung und Ausblick</b>	<b>59</b>
<b>Literaturverzeichnis</b>	<b>61</b>
<b>Anhang</b>	<b>65</b>
A. Markierungssätze der Penn-Treebank . . . . .	65
A.1. Wortartmarkierungen . . . . .	65
A.2. Phrasenmarkierungen . . . . .	66
B. Strafwerte . . . . .	67
C. Klassendiagramm . . . . .	69
D. Erweiterung des Sprachkorporus . . . . .	72
D.1. Scene XI: Orange juice from the fridge . . . . .	72
D.2. Scene XII: Window and cup . . . . .	72



# Abbildungsverzeichnis

2.1. Beispiel für Korreferenz . . . . .	4
2.2. Beispiel für semantische Rollen . . . . .	7
3.1. Architektur von PARSE . . . . .	10
3.2. Darstellung von SNLP . . . . .	11
3.3. Darstellung von Aktionen . . . . .	11
4.1. Dokumentübergreifendes Korreferenzauflösungssystem . . . . .	16
4.2. Beispiel für Unterereignisbeziehungen . . . . .	18
4.3. Beispiel für Ereigniskorreferenzauflösung mit Zweistufenmodell . . . . .	19
5.1. Ereignishierarchie aus [BH10] . . . . .	22
5.2. Gleichheitshierarchie für Aktionen . . . . .	23
6.1. Vorgehen des Agenten . . . . .	36
6.2. Vergleich zweier Aktionen . . . . .	37
6.3. Vereinfachtes Klassendiagramm des Agenten . . . . .	43
6.4. Beispielhafte Eingabe des Agenten . . . . .	45
6.5. Darstellung der Ergebnisse . . . . .	47
6.6. Transitive Kanten . . . . .	47
7.1. Ergebnisse (Variante 1) . . . . .	53
7.2. Ergebnisse (Variante 2) . . . . .	54
7.3. Ergebnisse (Variante 3) . . . . .	55
D.1. Kitchen (subject view) . . . . .	72
D.2. Scenery: Fridge and table . . . . .	72
D.3. Scenery: Dishwasher . . . . .	73



# Tabellenverzeichnis

2.1. Ausschnitt aus dem Penn-Treebank Tagset . . . . .	6
2.2. Ausschnitt aus den Penn-Treebank Chunk Tags . . . . .	6
4.1. Statistiken der ACE und ECB Korpora aus [BH10] . . . . .	14
4.2. Ergebnisse verwandter Arbeiten . . . . .	20
5.1. Kombinationen von Artikeln in zwei Aktionen . . . . .	27
5.2. Unterordnende Konjunktionen . . . . .	28
5.3. Schlüsselwörter für Wiederholung . . . . .	32
6.1. Verwendbare Begriffe anstelle von <i>do</i> , <i>happen</i> und <i>that</i> . . . . .	40
7.1. Übersicht über Sprachaufnahmen . . . . .	50
7.2. Evaluationsvarianten . . . . .	52
7.3. Ergebnisse (Variante 1) . . . . .	53
7.4. Ergebnisse (Variante 2) . . . . .	54
7.5. Ergebnisse (Variante 3) . . . . .	55
A.1. Wortartmarkierungen . . . . .	65
A.2. Phrasenmarkierungen . . . . .	66
B.3. Strafwerte der Filter . . . . .	67
B.4. Strafwerte für Tempora . . . . .	68



# 1. Einleitung

In der heutigen Zeit wird unsere Gesellschaft zunehmend von technischen Systemen, wie Staubsauger- und Mährobotern, geprägt, die das Leben erleichtern sollen. Da es für Menschen relativ aufwändig ist, Programmiersprachen zu lernen, um diese Geräte zu bedienen, werden sie zunehmend durch natürliche Sprache gesteuert. Beispiele dafür sind humanoide Roboter, wie der am KIT entwickelte ARMAR-III [ARA<sup>+</sup>06] oder intelligente persönliche Assistenten, wie Amazon Echo oder Google Now [Dal16]. Wenn man einem Roboter den Ablauf einer einfachen Aufgabe, wie das Zubereiten einer Speise erklären möchte, muss man dabei alle Teilschritte beschreiben. Oftmals ist es bei einer solchen Beschreibung nötig, bestimmte Teile eines Befehls zu wiederholen, um sich auf etwas bereits Gesagtes zu beziehen. Dies ist beispielsweise nötig, wenn man einen Ablauf nicht in chronologischer Reihenfolge beschreibt oder das gleichzeitige Ausführen mehrerer Aktionen verwenden möchte. So wird in folgendem Beispiel das Öffnen des Fensters zweimal genannt:

*„Open the window and then go to the table. After you have opened the window, jump.“*

Für einen Menschen ist es hier sofort ersichtlich, dass sich die beiden Formulierungen, die das Öffnen des Fensters beschreiben, auf denselben Vorgang in der realen Welt beziehen. Das Fenster soll also nur einmal geöffnet werden. Ein Roboter, der naiv alle Anweisungen in der Eingabe nacheinander abarbeiten würde, würde dies nicht erkennen und versuchen, das Fenster zweimal zu öffnen.

Nicht-chronologische Beschreibungen von Ereignissen treten besonders oft in gesprochener Sprache auf, da man dort, im Gegensatz zur geschriebenen Sprache nicht einfach das bereits Formulierte ändern kann. Wenn man also mehrere einfache Aktionen beschreibt, und nun möchte, dass eine weitere Aktion vor oder nach einer bereits beschriebenen Aktion auszuführen ist, muss man diese Aktion als Referenz erneut nennen. Es ist auch oft möglich, Kohärenz, also den inhaltlichen Zusammenhang, durch Bezugnahme auf etwas bereits Gesagtes in Nebensätzen, wie Kausal-, Attribut- oder Gliedsätzen, zu erzeugen. Eine Herausforderung ist, dass Menschen die gleiche Aktion auch nicht immer mit den gleichen Wörtern beschreiben.

## 1.1. Zielsetzung dieser Arbeit

Das Ziel dieser Bachelorarbeit ist der Entwurf und die Implementierung eines Agenten für das Rahmenwerk PARSE, der erkennt, wann zwei Nennungen von Aktionen in einem

Diskurs gleich sind. Dafür muss erst eine geeignete Definition von Gleichheit von Aktionsnennungen gefunden werden. Der Agent soll auf gesprochener natürlicher Sprache arbeiten. Er soll Gleichheit von zwei Aktionsnennungen auch dann erkennen, wenn die beiden Nennungen mit unterschiedlichen Worten und in unterschiedlichen Satzstrukturen auftreten. Der Agent erhält eine Eingabe, in der auftretende Aktionen bereits annotiert sind. Die Erkennung der Aktionen ist kein Ziel dieser Arbeit.

## **1.2. Aufbau der Arbeit**

Zunächst werden in Kapitel 2 wichtige Grundlagen zum Verständnis dieser Arbeit vorgestellt. In Kapitel 3 wird das Gesamtprojekt PARSE vorgestellt, in welches sich der hier zu entwickelnde Agent eingliedert. In Kapitel 4 werden wissenschaftliche Arbeiten vorgestellt, die ähnliche Probleme, wie das Problem in dieser Arbeit, untersuchen. In Kapitel 5 wird diskutiert, wann zwei Aktionsnennungen als gleich angesehen werden können und es wird analysiert, wie diese verschiedenen Formen von Gleichheit erkannt werden können. Mithilfe dieser Erkenntnisse wird in Kapitel 6 ein Vorgehen entworfen, um das Problem zu lösen. Im selben Kapitel wird auch die Implementierung besprochen. Letztendlich wird in Kapitel 7 der implementierte Agent anhand einer Laborstudie evaluiert. Kapitel 8 schließt diese Arbeit mit einer kurzen Zusammenfassung ab.

## 2. Grundlagen

In diesem Kapitel werden wichtige Grundlagen, die zum Verständnis dieser Arbeit dienen vorgestellt. Dies sind Begriffe, die verwendet werden, wenn über natürliche Sprache geredet wird (Abschnitt 2.1), und Aufgaben der natürlichen Sprachverarbeitung (Abschnitt 2.2). Zuletzt wird das Werkzeug WordNet vorgestellt, welches in dieser Arbeit verwendet wird (Abschnitt 2.3).

### 2.1. Natürliche Sprache

Natürliche Sprache ist die gesprochene und geschriebene Kommunikationsform der Menschen. Im Gegensatz zu formalen Sprachen, die relativ kompakt und genau definiert sind, besitzen natürliche Sprachen größere Komplexität und einen Grad von Mehrdeutigkeit [PK17]. Es ist für einen Rechner relativ einfach, formale Sprachen zu verstehen. So kann mit wenig Aufwand ein Programm zur Syntaxprüfung einer Programmiersprache erstellt werden. Die Interpretation von natürlicher Sprache ist hingegen für den Rechner weitaus schwieriger. In diesem Abschnitt werden wichtige Grundbegriffe der natürlichen Sprache erläutert.

#### 2.1.1. Syntax

Die Syntax beschreibt eine Menge von Regeln, die vorgeben wie Wörter zu Phrasen und Phrasen wiederum zu Sätzen zusammengesetzt werden können [PK17]. Syntax befasst sich nach Morris [Mor38] mit der Beziehung des Zeichens zu anderen Zeichen und ist neben der Semantik und der Pragmatik ein Teilgebiet der Semiotik. Verschiedene Sprachen haben unterschiedliche syntaktische Regeln. Beispielsweise sind im Englischen Sätze nach dem Muster Subjekt-Verb-Objekt aufgebaut. Der Satz „*ARMAR goes to the fridge*“ wäre also syntaktisch korrekt wohingegen „*To the fridge ARMAR goes*“ inkorrekt wäre.

#### 2.1.2. Semantik

Die Semantik wird auch als Bedeutungslehre bezeichnet und befasst sich mit der Beziehung des Zeichens zum Bezeichneten [Mor38]. Sie ist ein Teilgebiet der Semiotik. Sätze können syntaktisch korrekt und trotzdem semantisch inkorrekt sein, was an folgendem Beispiel verdeutlicht wird:

„*ARMAR sleeps the orange juice*“

Die syntaktischen Regeln (Subjekt-Verb-Objekt) werden eingehalten, aber dem Satz fehlt es an Sinnhaftigkeit.

### 2.1.3. Pragmatik

Pragmatik beschreibt die Beziehung des Gesprochenen zu seinem Sprecher [Mor38]. Sie untersucht die Bedeutung einer Äußerung in ihrem situativen Kontext. Beispielsweise kann der Satz „You have a green light“ abhängig vom Kontext bedeuten, dass die Ampel grün ist, wenn das Gespräch im Auto stattfindet, oder dass der Körper des Gesprächspartners grün leuchtet.

### 2.1.4. Wortarten

Wörter mit gleichen grammatikalischen Merkmalen werden in Wortarten (*engl. part of speech, POS*) eingeteilt. Die Wortart beinhaltet Informationen über das Wort und seine syntaktische Umgebung. Einige Beispiele für Wortarten sind: Substantiv (noun), Verb (verb), Adjektiv (adjective), Präposition (preposition), Adverb (adverb) und Konjunktion (conjunction) [JM09].

### 2.1.5. Lemma

Das Lemma eines Wortes ist seine Grundform [JM09]. Bei Substantiven ist im Englischen die Grundform die Singularform und bei Verben der Infinitiv. Beispielsweise ist das Lemma von *opened* das Wort *open*. *Open* ist ebenfalls das Lemma von *opens* und *opening*.

### 2.1.6. Korreferenz

Korreferenz liegt vor, wenn sich zwei Ausdrücke auf dieselbe Entität beziehen. Das Bezugsobjekt wird dabei Referent genannt. Die Referenz bezieht sich auf dieses Bezugsobjekt.



Abbildung 2.1.: Beispiel für Korreferenz

In Abbildung 2.1 bezieht sich *fridge* auf einen Kühlschrank, welcher der Referent ist. Das Pronomen *it* bezieht sich auf das Wort *fridge* und somit auf denselben Kühlschrank. Das Pronomen wird nach der Nominalphrase *the fridge* eingeführt. Man spricht in diesem Fall von einer **Anapher**. Die Wörter *Armar* und *he* sind ebenfalls korreferent, da beide die Entität ARMAR-III als Referenten haben. Das Pronomen steht hier vor dem referierten Ausdruck. Diese Konstellation wird **Katapher** genannt.

### 2.1.7. Prädikat

Das Prädikat ist der Bestandteil des Satzes, der den Inhalt und die Struktur des Satzes bestimmt [Car12]. In der traditionellen englischen Grammatik bilden alle Teile des Satzes, außer das Subjekt, das Prädikat. Das Prädikat dient also dazu, Aussagen über das Subjekt zu treffen. In der modernen Grammatik bildet die Verbalphrase das Prädikat. Für diese Arbeit wird unter dem Prädikatsbegriffs letzteres verstanden.

„Armar has opened the fridge“

Nach traditioneller Definition ist „*has opened the fridge*“ das Prädikat. Hier wird nach moderner Definition nur die Verbalphrase „*has opened*“ als Prädikat betrachtet.

### 2.1.8. Semantische Relationen

Semantische Relationen sind Beziehungen zwischen Bedeutungen von Wörtern[Mur03]. Hier werden die für diese Arbeit relevanten Relationen kurz beschrieben.

#### 2.1.8.1. Synonyme

Synonymie zwischen zwei Wörtern oder Ausdrücken liegt vor, wenn diese die gleiche, oder eine sehr ähnliche, Bedeutung haben. Es wird zwischen strikter und partieller Synonymie unterschieden. Strikte Synonymie, auch Bedeutungsgleichheit genannt, liegt dann vor, wenn zwei Wörter in allen Kontexten austauschbar sind und die gleiche Wirkung haben. Ansonsten liegt partielle Synonymie oder Bedeutungsähnlichkeit vor.

Beispiele für Synonyme sind *Kartoffel/Erdapfel*, *Hyperonym/Oberbegriff* und *essen/speisen*.

#### 2.1.8.2. Hyponymie und Hyperonymie

Hyponymie beschreibt die semantische Unterordnung eines Wortes unter ein anderes Wort. Umgekehrt beschreibt Hyperonymie eine Überordnung. Das übergeordnete Wort wird als Hyperonym oder Oberbegriff und das untergeordnete Wort als Hyponym oder Unterbegriff bezeichnet. Es besteht eine *ist-ein*-Beziehung zwischen Hypo- und Hyperonym. Die Hyponym-Hyperonym-Relation ist irreflexiv, asymmetrisch und transitiv.

Beispielsweise ist *Hund* ein Hyponym von *Tier* und *Tier* wiederum ein Hyponym von *Lebewesen*.

#### 2.1.8.3. Troponymie

Ein Verb *X* ist ein Troponym von Verb *Y*, wenn das Ausführen von *X* das Ausführen von *Y* auf eine bestimmte Art und Weise ist [CP04].

Beispielsweise ist *laufen* ein Troponym von *sich bewegen*.

## 2.2. Natürlichen Sprachverarbeitung

In diesem Abschnitt werden Aufgaben der natürlichen Sprachverarbeitung vorgestellt. Die natürliche Sprachverarbeitung beschäftigt sich mit der rechnergestützten Verarbeitung von natürlicher Sprache in Form von Text oder gesprochener Sprache [JM09]. Da dieses Themengebiet sehr umfangreich ist, werden nur die für diese Arbeit relevanten Aufgaben beschrieben.

### 2.2.1. Wortart-Markierung

Bei der Wortart-Markierung (*engl. part-of-speech tagging*) wird jedem Wort eine Wortart zugeordnet [JM09]. Üblicherweise werden auch Satzzeichen Tags zugeordnet. Tagsets enthalten die hierbei zur Verfügung stehenden Wortarten. In dieser Arbeit wird das Tagset der Penn-Treebank eingesetzt. Dieses Tagset umfasst 36 verschiedene Wortarten. Das vollständige Tagset befindet sich im Anhang Abschnitt A.1.

Im folgenden Beispiel ist der Satz „Armar please go to the fridge“ annotiert. Die Wortart steht, mit einem Schrägstrich getrennt, hinter den annotierten Wörtern.

**Beispiel: Markierung des Satzes „Armar please go to the fridge“**

Armar/NNP get/VB me/PRP a/DT glass/NN

Hier ist *Armar* ein Eigenname, *get* ein Verb, *me* ein Personalpronomen *a* ein Artikel und *glass* ein Substantiv im Singular. Die für das Beispiel relevanten Markierungen werden in Tabelle 2.1 aufgelistet.

Tabelle 2.1.: Ausschnitt aus dem Penn-Treebank Tagset

Markierung	Beschreibung
NNP	Eigenname, Singular
VB	Verb, Basisform
PRP	Personalpronomen
DT	Artikel
NN	Substantiv, Singular

### 2.2.2. Phrasenerkennung

Die Aufgabe der Phrasenerkennung (*engl. chunking*) ist es, den Text in abgeschlossene syntaktische Einheiten, sogenannte Phrasen, zu unterteilen und diese mit einem Typ zu markieren. Phrasen überlappen sich dabei nicht. Nicht alle Wörter gehören zu einer Phrase. Beispielsweise gehört das Wort „and“, das zwei Sätze verbindet zu keiner Phrase. Die wichtigsten Phrasen sind die Nominal- und Verbalphrasen.

Im folgenden Beispiel sind Wörter, die zu einer Phrase gehören in eckigen Klammern zusammengefasst. Die tiefgestellten Abkürzungen geben den Typ der Phrase an.

#### Beispiel: Phrasenerkennung

[NP Armar] [VP get] [NP me] [NP the orange juice] [SBAR after] [NP you] [VP have brought] [NP me] [NP the popcorn]

Das Beispiel enthält 6 Nominalphrasen, 2 Verbalphrasen und die unterordnende Konjunktion „after“, die den Temporalsatz einleitet. Tabelle 2.2 beinhaltet die für das Beispiel relevanten Phrasentypen aus Penn-Treebank. Insgesamt unterscheidet Penn-Treebank zwischen 8 verschiedenen Phrasentypen, die im Anhang Abschnitt A.2 aufgelistet sind.

Phrasen werden in PARSE im IOB-Format (*inside, outside, beginning*) [RM99] gekennzeichnet. In diesem Format wird für jedes Wort markiert, zu welchem Phrasentyp es gehört und ob es am Anfang (B) oder im Inneren (I) der Phrase steht oder ob es keiner Phrase angehört (O). Das Ende einer Phrase kann am Beginn der nächsten Phrase erkannt werden. Der obige Satz sieht im IOB Format wie folgt aus:

Tabelle 2.2.: Ausschnitt aus den Penn-Treebank Chunk Tags

Markierung	Beschreibung
NP	Nominalphrase
VP	Verbalphrase
SBAR	unterordnende Konjunktion

**Beispiel: IOB-Format**

Armar	get	me	the	orange	juice	
B-NP	B-VP	B-NP	B-NP	I-NP	I-NP	
after	you	have	brought	me	the	popcorn
B-SBAR	B-NP	B-VP	I-VP	B-NP	B-NP	I-NP

**2.2.3. Semantische Rollenzuteilung**

Die Aufgabe der semantischen Rollenzuteilung (*engl. semantic role labeling*) ist die Erkennung von Argumenten eines Prädikats in einem Satz und das Versehen von diesen mit semantischen Rollen [CM05]. Diese Aufgabe hilft dabei, den Sinn von Sätzen zu verstehen. Semantische Rollen geben an, in welcher semantischen Beziehung die Phrasen mit dem Prädikat stehen. Einige Beispiele für semantische Rollen sind der Agens, welcher die Aktion ausführt, der Patiens, dem etwas widerfährt, Recipient, der etwas erhält usw.

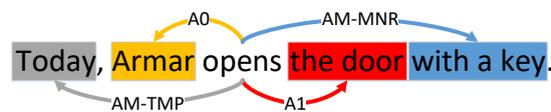


Abbildung 2.2.: Beispiel für semantische Rollen

In Abbildung 2.2 werden im Satz „Today, Armar opens the door with a key.“ die semantischen Rollen markiert. Es werden die semantischen Rollen der CoNLL-2005 [CM05] verwendet. Das Prädikat des Satzes ist „opens“. „Armar“ wurde die Rolle *A0* zugeteilt, welche meist den Argens angibt. „The door“ ist das direkte Objekt, welches geöffnet wird und erhält die Rolle *A1*. „With a key“ gibt das Hilfsmittel und „Today“ die Zeit an.

**2.3. WordNet**

WordNet [Fel05] ist ein lexikalisch-semantisches Netz, das 1985 an der Princeton University entwickelt wurde. Es beinhaltet Einträge für Substantive, Verben, Adjektive und Adverbien. WordNet enthält 155.287 englischsprachige Wörter, die in 117.659 sogenannte Synsets eingeteilt sind. Ein Synset fasst dabei alle Wörter gleicher Wortart mit gleicher Bedeutung zusammen. Synsets enthalten neben den Wörtern eine Beschreibung und ggf. Beispielsätze. Wörter können in mehreren Synsets enthalten sein, wenn sie mehrere Bedeutungen haben. Das Substantiv *dog* ist zum Beispiel unter anderem in folgenden Synsets:

1. S: (n) **dog**, domestic dog, *Canis familiaris* (a member of the genus *Canis* (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds) *“the dog barked all night”*
2. S: (n) cad, bounder, blackguard, **dog**, hound, heel (someone who is morally reprehensible) *“you dirty dog”*

Mit *dog* kann also der Hund als Haustier, aber auch ein gemeiner Kerl gemeint sein.

WordNet enthält semantische Beziehungen zwischen diesen Synsets. Unter anderem sind dies Hyperonym- und Hyponymbeziehungen und Meronymiebeziehungen (Teil-Ganzes-Beziehungen) zwischen Substantivsynsets und Troponymbeziehungen zwischen Verbsynsets. Durch die Hyperonym- und Hyponymbeziehungen wird so eine Taxonomie aufgebaut. Alle Substantive sind dabei (indirekte) Hyponyme des Synsets *entity* (außer *entity* selber).



## 3. PARSE

Das in dieser Arbeit entwickelte System soll als Agent in das Rahmenwerk PARSE (Programming ARchitecture for Spoken Explanations) [WT15] eingebunden werden. Das Projekt PARSE hat das Ziel, einem System, wie dem am KIT entwickelten humanoiden Haushaltsroboter ARMAR-III [ARA<sup>+</sup>06] mittels natürlicher Sprache neue Funktionen beizubringen.

### 3.1. Entwurfsprinzipien

PARSE verfolgt folgende Entwurfsprinzipien:

**Agentenbasiert** Die einzelnen Verarbeitungsschritte von PARSE werden als Agenten realisiert. Diese arbeiten unabhängig und transparent voneinander. Dadurch ist das System einfach erweiterbar, da bestehende Agenten für das Hinzufügen von neuen Agenten nicht modifiziert werden müssen. Außerdem können die Agenten unabhängig voneinander evaluiert werden.

**Testgetrieben** Die Funktionalitäten von PARSE werden während der Entwicklung evaluiert, um die Qualität zu überprüfen. Dies geschieht mit Sprachkorpora, die Befehle an den Haushaltsroboter ARMAR-III enthalten.

**Wissensbasiert** Das System verwendet Werkzeuge, wie die lexikalische Datenbank WordNet [Fel05] und SENNA [CWB<sup>+</sup>11], um tieferes Sprachverständnis zu ermöglichen. Damit ist beispielsweise die Auflösung von Mehrdeutigkeiten, Erkennung von Eigennamen und Zuordnung von semantischen Rollen möglich.

**Domänenunabhängigkeit der Agenten** Die Agenten arbeiten unabhängig von der Zieldomäne. Das Domänenwissen ist in einer Ontologie mit fester Struktur dargestellt. Bei einer Änderung der Zieldomäne müssen die Agenten nicht angepasst werden.

### 3.2. Architektur

Die Architektur gliedert sich in mehrere Teilprojekte. Aus den gesprochenen Anweisungen wird zuerst mithilfe eines automatischen Spracherkenners (*engl. automatic speech recognition*) Text gewonnen. Aus diesem Text wird mithilfe von seichter Sprachverarbeitung (*engl. shallow natural language processing*) ein Graph gebaut. Dieser Graph dient als Repräsentation des Textes und gemeinsame Datenstruktur der Agenten. Diese Agenten für Sprachverständnis (*engl. natural language understanding*) modifizieren den Graphen anschließend in

unabhängigen Analyseschritten. Der letzte Schritt ist die Generierung von ausführbarem Quellcode in der Zieldomäne. Der Aufbau der Architektur ist in Abbildung 3.1 dargestellt. Der in dieser Arbeit zu implementierende Agent ist rot eingefärbt. Im Folgenden werden einige bereits vorhandene Vorverarbeitungsschritte und Agenten näher beschrieben.

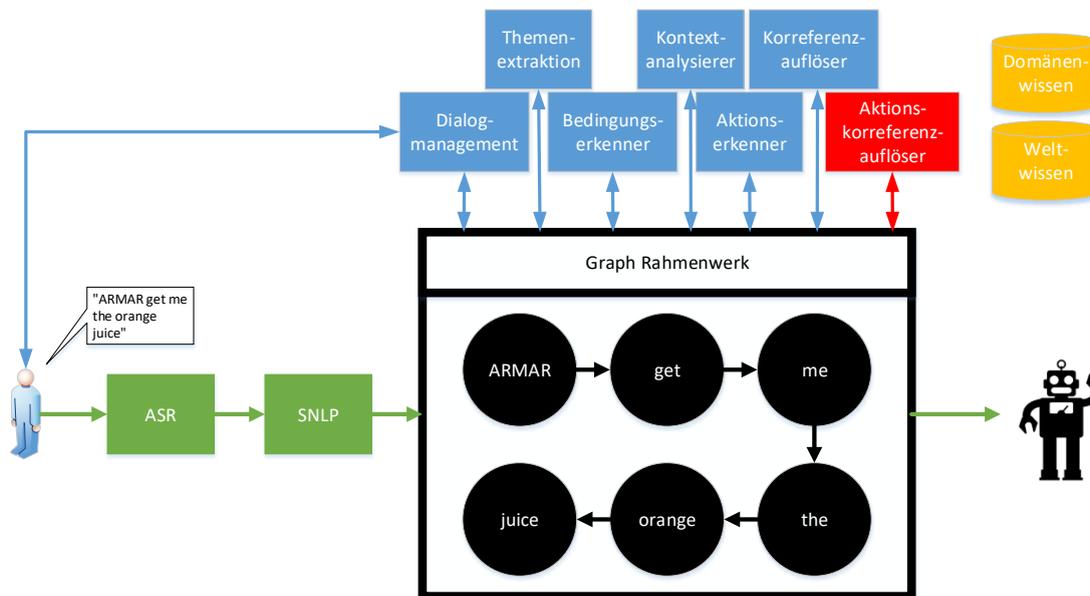


Abbildung 3.1.: Architektur von PARSE

### 3.2.1. Seichte Sprachverarbeitung

Die seichte Sprachverarbeitung [Koc15] nimmt den Text, der von dem automatischen Spracherkennung erzeugt worden ist, entgegen und generiert aus diesem eine initiale Handlungsrepräsentation. Diese Handlungsrepräsentation wird als Graph dargestellt. Die Knoten des Graphen sind die Tokens (Wörter) der Eingabe, die mit Kanten verbunden sind. In jedem Knoten werden neben dem Wort unter anderem auch die Wortart (Abschnitt 2.2.1), Phrase im IOB-Format (Abschnitt 2.2.2) und die Befehlsnummer gespeichert.

Die Graphdarstellung des Befehls „Armar get me the orange juice after you have brought me popcorn“ wird in Abbildung 3.2 gezeigt.

### 3.2.2. Aktionserkennung

Die Aktionserkennung [Ou16] extrahiert aus dieser initialen Handlungsrepräsentation Aktionen. Aktionen bestehen aus einem Prädikat, einem Akteur und Parametern, die die Fragen „was?“, „wer?“, „wo?“, „wann?“, „wie?“ und „warum?“ beantworten. Aktionen werden als Graph dargestellt. Das Prädikat wird mit dem jeweils ersten Token der Akteure und Parameter verbunden. Mehrere hintereinander auftretende Aktionen werden durch ihre Prädikate verbunden. In den Knoten stehen neben den Wörtern auch deren Rollen.

Abbildung 3.3 zeigt die Darstellung des Satzes „Armar came and he brought me an orange juice“. Dieser Satz enthält zwei Aktionen mit den Prädikaten *came* und *brought*. Prädikate werden rot, Akteure blau und Parameter grün dargestellt.

### 3.2.3. Kontextanalysierer

Der Kontextanalysierer [Hey16] ergänzt den Graphen um Kontextinformationen. Er erkennt Entitäten und weist ihnen ein übergeordnetes Konzept zu. Den Entitäten werden Eigenschaften zugeteilt. Der Kontextanalysierer erkennt Kontextinformationen, wie Teil-Ganzes-Beziehungen, Zustände, mögliche Zustandsübergänge und Ortsbeziehungen.

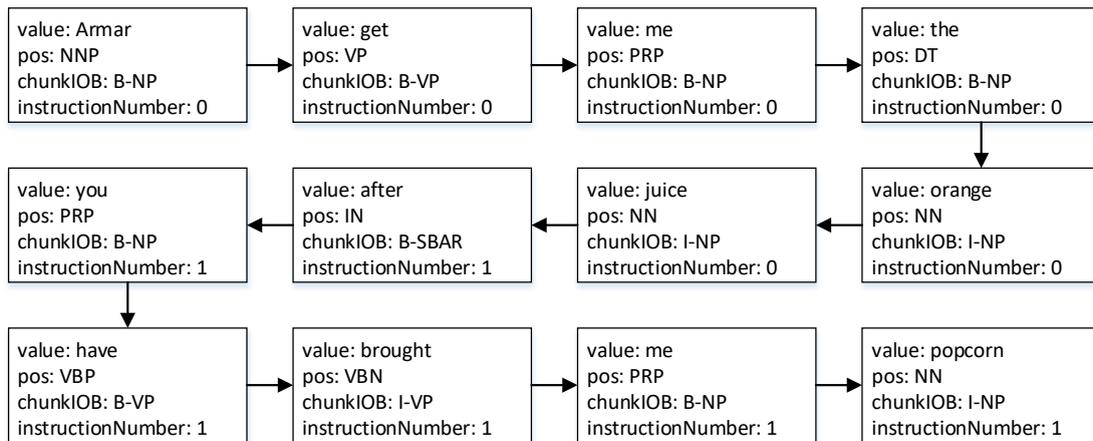


Abbildung 3.2.: Darstellung von „Armar get me the orange juice after you have brought me popcorn“

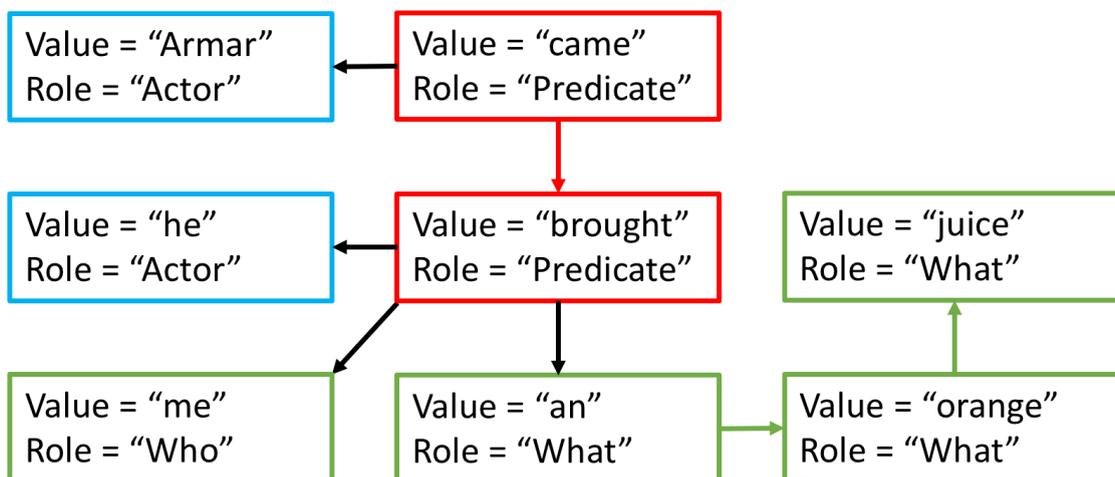


Abbildung 3.3.: Darstellung von Aktionen in „Armar came and he brought me an orange juice.“ aus [Ou16]

### 3.2.4. Korreferenzauflöser

Dieser Agent verwendet die Ergebnisse des Kontextanalysierers, um Korreferenzen Abschnitt 2.1.6 aufzulösen [Hey16]. Im Satz „Armar, please go to the dishwasher and open it“ beziehen sich *dishwasher* und *it* auf dieselbe Entität. Korreferenzen werden durch Kanten dargestellt, die unter Anderem die Art der Korreferenz und einen Konfidenzwert enthalten.

### 3.2.5. Bedingungsanalysierers

Der Bedingungsanalysierers [Ste16] identifiziert Bedingungssätze der Form *if-then-else* in der als Graph vorliegenden Eingabe. Er markiert, welche Teile zu den *if*-, *then*- und *else*-Blöcken einer Bedingung gehören. Das folgende Beispiel veranschaulicht dies:

„ARMAR please go to the table *if the dishes are dirty then put them into the dishwasher else put them into the cupboard*“

Der *if*-Teil ist grün, der *then*-Teil blau und der *else*-Teil gelb eingefärbt. Der schwarze Teil wird unabhängig ausgeführt.

### 3.2.6. Dialogmanagement

Der Dialogagent [Sch17] hilft dabei, Fehler des Spracherkenners, des Korreferenzauflösers und des Bedinungsanalyserers zu beheben, indem er den Nutzer nach der Intention seiner Eingabe fragt.

## 4. Verwandte Arbeiten

Im Gegensatz zur Entitätskorreferenz (*entity coreference*) wird Korreferenz zwischen Aktionen in der Forschung kaum betrachtet. Die meisten Arbeiten in diesem Bereich untersuchen Ereigniskorreferenz (*event coreference*) in Zeitungsartikeln aus den in Abschnitt 4.1 vorgestellten Korpora. Dabei wird zwischen Korreferenz von Ereignissen innerhalb eines Dokuments (*within-document*) und Ereignissen aus mehreren Dokumenten (*cross-document*) unterschieden. Es geht darum, bestimmte Ereignisnennungen aus den Dokumenten zu extrahieren und alle Nennungen, die sich auf dasselbe Ereignis in der realen Welt beziehen, in Äquivalenzklassen zusammenzufassen. Bei den untersuchten Zeitungsartikeln handelt es sich nicht um die, für diese Arbeit vor allem relevanten Imperativsätze in gesprochener Sprache, sondern um geschriebene beschreibende Sätze. Trotzdem können die hier vorgestellten Arbeiten zur Ereigniskorreferenz interessante Ansätze haben, die teilweise für die Ziele dieser Arbeit von Nutzen sind.

In diesem Kapitel werden wichtige Korpora mit Annotationen zu Ereigniskorreferenz vorgestellt. Anschließend werden einige für diese Bachelorarbeit relevanten Arbeiten über Ereigniskorreferenz vorgestellt und deren Stärken und Schwächen diskutiert.

### 4.1. Korpora

Die folgenden Korpora werden im Bereich Ereigniskorreferenzauflösung verwendet. Ein Korpus ist in diesem Fall eine Sammlung von Schriften, in der die Musterlösungen für zu findenden Korreferenzbeziehungen annotiert sind.

#### 4.1.1. ACE

Das ACE-Korpus ist ein oft verwendetes Korpus im Bereich Ereigniskorreferenz.

Die Ziele des Forschungsprojekts **Automatic Content Extraction (ACE)** [DMP<sup>+</sup>04] ist die Entwicklung von Technologien zur Erkennung von genannten Entitäten, Beziehungen zwischen diesen und Ereignissen an denen diese Entitäten teilnehmen. ACE stellt eine Reihe von Korpora mit englischen, chinesischen und arabischen annotierten Texten bereit. Diese Korpora beinhalten Texte aus Nachrichtensendungen und Zeitungsartikeln. Im Rahmen des Projektes wurden Evaluationswerkzeuge entwickelt und Workshops veranstaltet. Das Projekt ist der Nachfolger von **Message Understanding Conferences (MUC)** [GS96]. Im Gegensatz zu MUC ist das Ziel von ACE, Objekte und nicht deren wörtlichen Namen zu identifizieren. Somit ist die Aufgabe abstrakter.

Die Texte werden im XML-Format annotiert. Es gibt drei Annotationsaufgaben:

Tabelle 4.1.: Statistiken der ACE und ECB Korpora aus [BH10]

	ACE	ECB
Anzahl Dokumente	745	482
Anzahl dokumentinterner Ereignisse	4946	1302
Anzahl dokumentübergreifender Ereignisse	-	208
Anzahl Themen	-	43

**Entity Detection and Tracking (EDT)** Es müssen alle Nennungen von Entitäten verschiedener Typen gefunden werden. Die sieben Typen sind: *Person*, *Organisation*, *Ort*, *Einrichtung*, *Waffe*, *Fahrzeug* und *geopolitische Entität*. Zu jedem Typ existieren Subtypen. Die gefundenen Entitäten müssen in Äquivalenzklassen eingeordnet werden. Dafür muss Korreferenzauflösung stattfinden. Für jede Nennung muss die größte Zeichenkette, die die Entität beschreibt zusammen mit dem Kopf annotiert werden. Es muss außerdem annotiert werden, ob die Entität spezifisch, generisch, als Attribut, negativ, quantifiziert oder unterspezifiziert genannt wird.

**Relation Detection and Characterization (RDC)** Es müssen Beziehungen zwischen den gefundenen Entitäten identifiziert und charakterisiert werden. Hierbei gibt es fünf Beziehungstypen. Die Argumente einer Beziehung sind die Entitäten, zwischen denen die Beziehung besteht.

**Event Detection and Characterization (VDC)** Es müssen Ereignisse von fünf verschiedenen Typen gefunden werden, an denen Entitäten teilnehmen. Diese Typen sind: *Interaktion*, *Bewegung*, *Transfer*, *Erstellung* und *Zerstörung*. Neben Typ und Subtyp müssen Teilnehmer und Attribute von Ereignissen identifiziert werden.

#### 4.1.2. ECB

Im ACE-Korpus ist keine Korreferenz zwischen dokumentübergreifenden Ereignissen annotiert. Die Ereignisse sind auf wenige verschiedene Typen beschränkt. Aus diesem Grund habe Bejan und Harabagiu 2010 [BH10] das EventCorefBank-Korpus erstellt. Das Korpus beinhaltet mehr Ereignistypen und ermöglicht die Evaluation von Systemen zur dokumentübergreifenden Ereigniskorreferenzauflösung. Die Texte stammen aus dem GoogleNews-Archiv. Die Dokumente, die das gleiche grundlegende Ereignis beschreiben sind in Themen (*engl. topics*) organisiert.

Tabelle 4.1 zeigt Statistiken zu den beiden Korpora.

#### 4.1.3. IC

Das Intelligence Community Corpus [HMV<sup>+</sup>13] unterscheidet zwischen vollständiger und partieller Identität. Zwei Ereignisnennungen sind vollständig korreferent, wenn die beschriebenen Ereignisse in allen Punkten identisch sind, also auch dieselben Teilnehmer haben und zur selben Zeit am selben Ort stattfinden. Zwei Ereignisnennungen sind partiell korreferent, wenn die beschriebenen Ereignisse quasi-identisch sind. Ereignisse sind quasi-identisch, wenn sie in den meisten Aspekten übereinstimmen, aber ein Ereignis zusätzliche Informationen beinhaltet. Es gibt zwei Arten partieller Korreferenz:

- **Membership:** Die erste Ereignisnennung beschreibt mehrere ähnliche Ereignisse. Die zweite Ereignisnennung beschreibt eine Teilmenge aus diesen. *Beispiel:* „I attended three parties(*E1*) last month. The first one(*E2*) was the best.“

- Subevent: Die erste Ereignisnennung beschreibt ein kompliziertes Ereignis, das aus mehreren Schritten zusammengesetzt ist. Die zweite Ereignisnennung ist einer dieser Schritte. *Beispiel:* „I was eating in a restaurant(E1).First I was ordering food(E2)...“

Das Korpus beinhaltet 100 Texte über Gewalttaten. Es sind Ereignisse, die vollständig oder partiell miteinander in Korreferenzbeziehungen stehen, und die Art der Beziehungen annotiert.

## 4.2. Ansätze zur Ereigniskorreferenzauflösung

Im Folgenden werden wichtige Arbeiten, die sich mit der Auflösung von Ereigniskorreferenz beschäftigen vorgestellt. Am Ende dieses Kapitels wird ein Überblick über alle diskutierten Ansätze gegeben.

### 4.2.1. Cross-Document Event Coreference: Annotations, Experiments, and Observations

Der erste Ansatz zur Ereigniskorreferenzauflösung stammt von Bagga und Baldwin und wird in **Cross-Document Event Coreference: Annotations, Experiments, and Observations** [BB99] beschrieben. Das System erhält Dokumente (Zeitungsartikel der New York Times) zu einem bestimmten Ereignistyp, beispielsweise „Rücktritt“ und fasst alle Dokumente, in denen es um das selbe Ereignis in der echten Welt geht, in Äquivalenzklassen zusammen. Zwei Rücktrittsereignisse stehen beispielsweise in einer Korreferenzbeziehung, wenn jeweils dieselbe Person zur selben Zeit von derselben Organisation zurücktritt. Die Software zur Erkennung von Ereigniskorreferenz baut auf einer Software zur Erkennung von Personenkorreferenz auf. Dieses erste System wird in Abbildung 4.1 dargestellt und funktioniert wie folgt: Zuerst werden Korreferenzketten für die einzelnen Dokumente gebildet. Dies geschieht mit dem Within-Document-Korreferenzmodul des CAMP-Systems der University of Pennsylvania [BB98]. Korreferenzketten enthalten jeweils alle zueinander korreferenten Entitäten. Anschließend werden alle Sätze extrahiert, die eine Nominalphrase aus der relevanten Korreferenzkette enthalten. Diese Sätze bilden sogenannten „Zusammenfassungen“ für jedes Dokument. Das VSM-Disambiguierungsmodul betrachtet diese Zusammenfassungen und berechnet anhand der darin vorkommenden Wörtern die Ähnlichkeiten zwischen diesen. Wenn die Ähnlichkeit zweier Dokumente einen gewissen Grenzwert überschreitet, wird davon ausgegangen, dass es in diesen Dokumenten um dieselbe Entität geht.

Für die Betrachtung von Ereigniskorreferenz wurde das System abgeändert, weil CAMP keine Korreferenzketten für Ereignisse bildet. Für die Zusammenfassung werden alle Sätze verwendet, die das Verb (z.B. „wählen“), das das Ereignis beschreibt, oder eine Substantivierung davon (z.B. „Wahl“) enthalten. Sätze mit Synonymen des Verbs werden nicht betrachtet.

Experimente auf verschiedenen Datensätzen zeigen, dass der optimale Grenzwert, ab dem zwei Ereignisse als korreferent angesehen werden sollen, sinkt, je mehr Wörter sich die verschiedenen Zusammenfassungen teilen. Das System erzielt F-Maße zwischen 0,45 und 0,9. Besonders schlecht fielen die Ergebnisse bei der Untersuchung von Dokumenten, die politische Wahlen beschreiben aus. Wahlen finden oft zwischen den gleichen Parteien statt und unterscheiden sich nur durch das Datum. Dieses Datum wird in den meisten Artikeln nicht explizit erwähnt. Stattdessen ist beispielsweise nur von der „nächsten Wahl“ die Rede. Bei weiteren Experimenten wurden auch vordefinierte Synonyme eines Verbs betrachtet. Die Betrachtung von Synonymen konnte die Ergebnisse um 0,03 verbessern.

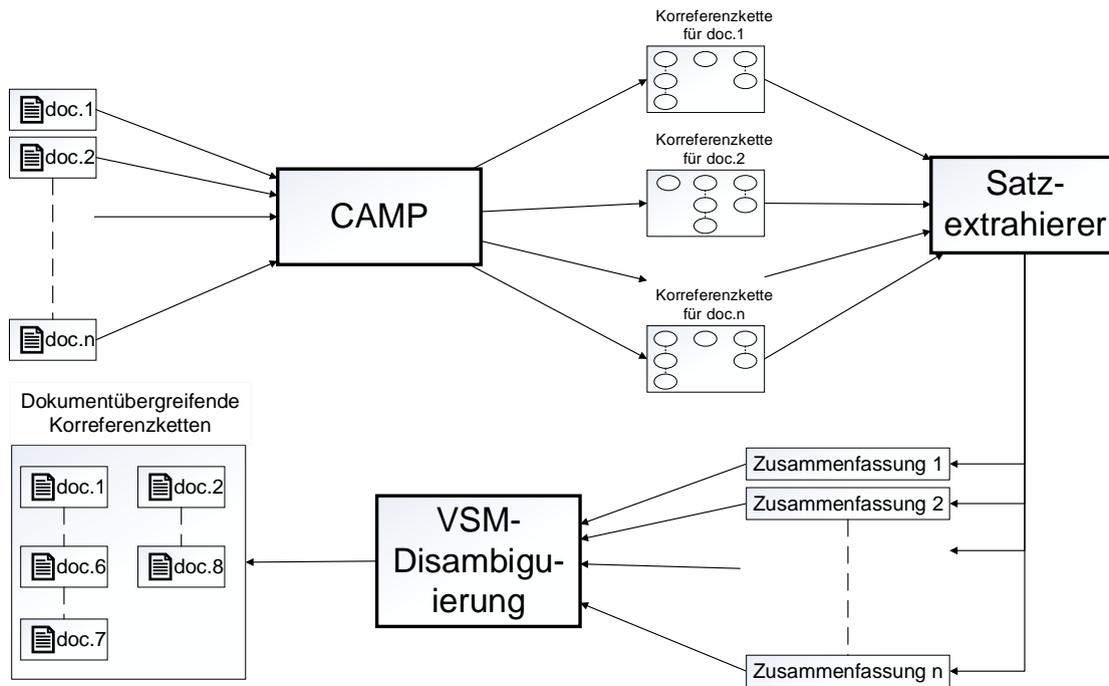


Abbildung 4.1.: Architektur des dokumentübergreifenden Korreferenzauflösungssystems aus [BB99]

#### 4.2.2. Graph-based Event Coreference Resolution

Chen und Ji betrachten in der Arbeit **Graph-based Event Coreference Resolution** [CJ09] die Aufgabe der Ereigniskorreferenzauflösung als Clustering-Problem.

Dabei wird für jeden ACE-Ereignissubtypen ein ungerichteter gewichteter Graph aufgebaut. Die Knoten entsprechen den Ereignisnennungen des jeweiligen Typen. Das Gewichte einer Kante zwischen zwei Knoten gibt an, wie wahrscheinlich es ist, dass sich beide Nennungen auf dasselbe Ereignis beziehen. Die Bestimmung dieser Kantengewichte ist ein Spectral Graph Clustering Problem, welches das normalisierte Spectral Clustering nach Shi und Malik [SM00] optimiert. Für diese Optimierung wird der Hauptvektor 2. Stufe der Korreferenzmatrix berechnet. Bei der Berechnung der Korreferenzmatrix werden die Ereignisauslöser (das Wort, das am deutlichsten das Auftreten eines Ereignisses ausdrückt) und die Ereignisargumente betrachtet.

Chen und Ji erreichen mit ihrem Vorgehen bei Experimenten auf dem ACE-2005-Korpus ein CEAF-Maß [Luo05] von 0,8363.

#### 4.2.3. Joint Entity and Event Coreference Resolution across Documents

Lee et al. stellen in der Arbeit **Joint Entity and Event Coreference Resolution across Documents** [LRC<sup>+</sup>12] einen neuen Ansatz zur Korreferenzauflösung vor, in dem sie Entitäts- und -Ereigniskorreferenz als gemeinsame Aufgabe betrachten. Das System untersucht sowohl Korreferenz zwischen Entitäten und Ereignissen in verschiedenen Dokumenten als auch innerhalb eines Dokuments. Dabei fließen Informationen zwischen Entitäts- und Ereignisclustern und helfen so, weitere Cluster zu verbinden. Nominalphrasen können dabei auch Ereignisse beschreiben, was bei einer getrennten Betrachtung von Entitäten als Nominalphrasen und Ereignissen als Verbalphrasen nicht berücksichtigt werden würde. Ereignisse werden neben ihrer Zeit und ihrem Ort auch durch ihre Teilnehmer und Argumente näher beschrieben. Eine Korreferenzauflösung zwischen diesen Entitäten hilft weiter bei der Korreferenzauflösung von Ereignissen.

Das folgende Beispiel aus [LRC<sup>+</sup>12] soll zeigen, wie Korreferenz zwischen Ereignissen als Verbal- und Nominalphrasen aussehen und eine wechselseitige Betrachtung von Entitäten und Ereignissen bei der Korreferenzauflösung helfen kann:

#### Beispiel: Korreferenz zwischen Verbal- und Nominalphrasen

- (1) **One of the key suspected Mafia bosses arrested yesterday** has hanged himself.
- (2) Police said **Lo Presti** had hanged himself.
- (3) His suicide appeared to be related to clan feuds.

In (3) beschreibt die Nominalphrase *His suicide* ein Ereignis. In (1) und (2) beschreiben die Verbalphrasen *has/had hanged* dasselbe Ereignis. Es besteht also Korreferenz zwischen Nominal- und Verbalphrasen. Die Korreferenz zwischen den Entitäten *One of the key suspected Mafia bosses arrested yesterday* in (1) und *Lo Presti* in (2) kann gefunden werden, da sie Akteure von Ereignissen sind, zwischen denen Korreferenz besteht.

Das System arbeitet mit einem Algorithmus, der iterativ Cluster aus Entitäts- und Ereignisnennungen bildet.

Zuerst werden die Eingabedokumente in Dokumentencluster eingeteilt, um den Suchraum zu reduzieren und Fehler durch Mehrdeutigkeiten zu vermeiden. Es werden nur Entitäts- und Ereignisnennungen in einem Cluster zusammengefasst, welche im selben Dokumentencluster liegen.

Anschließend werden nominale, verbale und pronomiale Nennungen extrahiert und bilden jeweils einelementige Cluster. Das System unterscheidet dabei nicht zwischen Entitäts- und Ereignisnennungen. Auf diese Cluster werden Filter mmit hoher Präzision aus dem Stanford Coreference Resolution System [LPC<sup>+</sup>11] angewendet, um den Suchraum weiter zu reduzieren. Diese Filter betrachten unter anderem lexikalische, syntaktische und semantische Informationen.

Die Cluster werden mithilfe linearer Regression verbunden. Es wird dabei die Qualität einer Verschmelzung mit einer Punktzahl modelliert. Eine Punktzahl, die größer als 0,5 ist bedeutet, dass mehr als die Hälfte der durch die Verschmelzung der Cluster entstehenden Nennungspaare korrekt ist. In jedem Schritt wird das Clusterpaar mit der höchsten Punktzahl verschmolzen.

Das System erreicht bei Experimenten auf dem ECB-Korpus [BH10] ein MUC-F-Maß von 0,678.

#### 4.2.4. Detecting Subevent Structure for Event Coreference Resolution

Araki et al. untersuchen in **Detecting Subevent Structure for Event Coreference Resolution** [ALHM14] die Beziehung zwischen Ereignissen und ihren Unterereignissen. Abbildung 4.2 zeigt beispielsweise die Unterereignisstruktur für folgenden Text:

*Ismail said the fighting, which lasted several days, intensified when forces loyal to Egal's Ha-bar Awal sub-clan of the Issak **attacked**(E12) a militia stronghold of his main opposition rival,...*

*Egal militia, claiming to be the national defence force, said they had **captured**(E15) two opposition posts, **killing**(E16) and **wounding**(E17) many of the fighters, **destroying**(E18) three technicals (armed pick-up trucks) and **confiscating**(E19) artillery guns and assorted ammunition.*

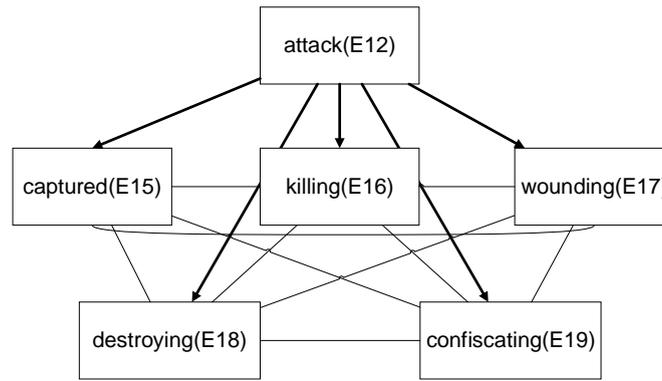


Abbildung 4.2.: Beispiel für Unterereignisbeziehungen aus [ALHM14]. Pfeile sind Unterereignisbeziehungen vom Eltern- zum Unterereignis.

Die Ereignisse E15, E16, E17, E18, und E19 sind Unterereignisse des Elternereignisses E12. Ereignispaare, die sich ein Elternereignis teilen werden *subevent sisters* genannt.

Araki et al. entwickelten ein System zur Auflösung von Ereigniskorreferenz, das Beziehungen zwischen zwei Ereignisnennungen in folgende Klassen einteilt:

- full coreference (FC): Beide Nennungen beziehen sich auf dasselbe Ereignis
- subevent parent-child (SP): Ein Ereignis ist ein Unterereignis des anderen Ereignisses.
- subevent sister (SS): Beide Ereignisse haben dasselbe Elternereignis.
- no coreference (NC): Es liegt keine Korreferenz vor.

Das System arbeitet mit einem multinomialen logistischen Regressionsmodell.

Araki et al. erzielen bei der Erkennung von Eltern-Kind-Beziehungen einen BLANC-F1-Wert von 0,5619. Sie verwenden das Korpus der Intelligence Community [HMV<sup>+</sup>13].

#### 4.2.5. Supervised Within-Document Event Coreference using Information Propagation

Liu et al. beschreiben in **Supervised Within-Document Event Coreference using Information Propagation** [LAHM14] eine Methode zur Erkennung von Ereigniskorreferenz innerhalb eines Dokuments.

Für Ereignisnennungen wird von einem überwachten Klassifikator paarweise die Wahrscheinlichkeit berechnet, dass sie korreferent sind. In dem paarweisen Modell werden 105 verschiedene semantische Features betrachtet. Alle Paare mit hoher Wahrscheinlichkeit werden als korreferent angesehen und geclustert. Dabei werden Informationen zwischen den beiden Ereignisnennungen übertragen. So ist es möglich, fehlende Argumente in einer Ereignisnennung von einer mit ihr korreferenten Nennung zu kopieren. Durch diese Anreicherung kann dann möglicherweise Korreferenz mit einer dritten Nennung erkannt werden.

In diesem Vorgehen werden auch Informationen über den Handlungsort von Ereignissen betrachtet. Es ist zum Beispiel möglich, die Nennung „Istanbul“ mit dem Land „Türkei“ in Verbindung zu bringen. Die Ortsnennungen werden vom *Stanford Named Entity Recognizer* [FGM05] erkannt. Die Informationen über Orte stammen aus *DBpedia* [LIJ<sup>+</sup>15] und *Geonames* [geo]. Liu et al. erreichen mit ihrem Vorgehen ein F-Maß von 0,5311 auf dem *Intelligence Community (IC)*-Korpus und 0,4697 auf dem *ACE*-Korpus.

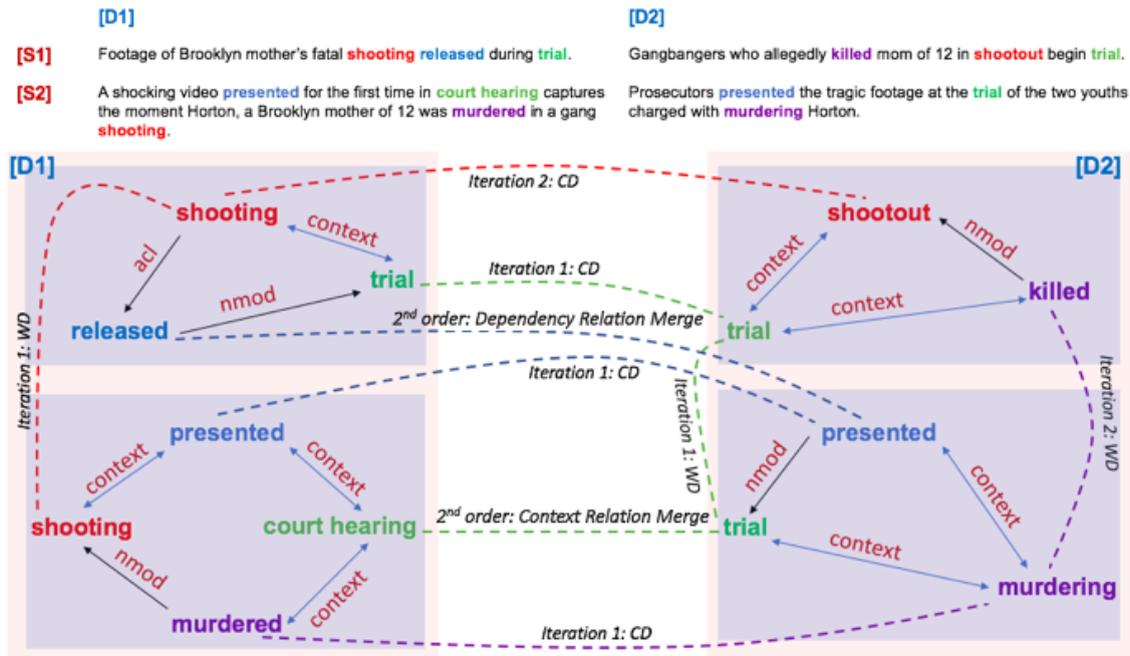


Abbildung 4.3.: Beispiel für Ereigniskorreferenzauflösung mit Zweistufenmodell. Ereignisnennungen sind fettgedruckt; durchgezogene Pfeile sind Beziehungen zweiter Ordnung; gestrichelte Linien verbinden korreferente Ereignisse.

#### 4.2.6. Event Coreference Resolution by Iteratively Unfolding Inter-dependencies among Events

Choubey und Huang stellen in **Event Coreference Resolution by Iteratively Unfolding Inter-dependencies among Events** [CH17] einen Ansatz zur dokumentübergreifenden (*engl. cross-document*) Ereigniskorreferenzauflösung vor. Im Gegensatz zu früheren Ansätzen, verwenden Choubey und Huang zwei verschiedene Klassifikatoren für dokumentinterne (WD)- und dokumentübergreifende (CD) Ereigniskorreferenz. Dabei betrachtet der WD-Klassifikator eher Argumente von Ereignisnennungen, während für den CD-Klassifikator eher der Kontext relevant ist. Der Ansatz clustert im ersten Schritt Ereignisse, indem abwechselnd WD- und CD-basierende Verschmelzoperationen auf ähnlichen Clustern angewendet werden. Dies wird solange wiederholt, bis es keine Cluster mehr gibt, die sich ähnlich genug sind, um verschmolzen zu werden. Mit diesem Vorgehen werden Wechselwirkungen zwischen Ereignisnennungen aus verschiedenen Ereignisketten ausgenutzt. Im zweiten Schritt werden Beziehungen zweiter Ordnung zwischen Ereignisclustern betrachtet. Wenn zwei Ereignisse gleiche Beziehungen (z.B. Kontextbeziehungen) zu anderen Ereignissen haben, ist es wahrscheinlich, dass es sich dabei um dasselbe Ereignis handelt.

Abbildung 4.3 zeigt ein Beispiel, wie Ereigniskorreferenz mit diesem Ansatz funktioniert: Es soll Ereigniskorreferenz zwischen zwei Dokumenten mit je zwei Sätzen aufgelöst werden. In der ersten Iteration werden einfache Korreferenzen innerhalb der Dokumente (*shooting* und *trial*) und zwischen den Dokumenten (*presented*, *trial* und *murdered*/*murdering*) aufgelöst. Dabei werden Argumente zwischen korreferenten Ereignissen ausgetauscht. In der zweiten Iteration werden weitere Korreferenzen aufgelöst (*shooting*/*shootout* und *killed*/*murdering*). Danach werden weitere Korreferenzen mithilfe Beziehungen zweiter Ordnung aufgelöst: *released* aus dem ersten und *presented* aus dem zweiten Dokument stehen in beide in einer *nmod*-Beziehung zum *trial*-Cluster, was darauf hindeutet, dass es sich um das gleiche Ereignis handelt. Es besteht Korreferenz zwischen *court hearing* und

Tabelle 4.2.: Ergebnisse verwandter Arbeiten

Autoren	Jahr	CD/WD	Korpus	Metrik	F-Maß (in %)
Bagga, Baldwin	1999	CD	nicht verfügbar	MUC	71-90
				$B^3$	43-88,2
Chen, Ji	2009	WD	ACE 2005	CEAF	83,63
Lee et al.	2012	CD	ECB	MUC	67,8
				$B^3$	64,1
				CEAF	35,8
				BLANC	72,3
				CoNLL	55,9
Araki et al.	2014	WD	IC	BLANC	56,19
Liu et al.	2014	WD	IC	Paarweise	53,11
				MUC	61,44
				$B^3$	84,33
				CEAF	78,83
				BLANC	75,06
Choubey, Huang	2017	CD+WD	ECB+	$B^3$	61 (CD), 72,4 (WD)
				MUC	73,4 (CD), 62,6 (WD)
				CEAF	56,5 (CD), 71,8 (WD)
				CoNLL	63,6 (CD), 68,9 (WD)

*trial*, weil diese korreferente Ereignisse in ihrem Kontext haben.

Das System mit zwei verschiedenen Klassifikatoren und der Betrachtung von Beziehungen zweiter Ordnung, erreicht bei Experimenten auf dem ECB+ Korpus [CV14] ein CoNLL-F-Maß [PLR<sup>+</sup>14] von 0,636 bei CD- und 0,689 bei WD-Ereigniskorreferenzauflösung. Damit erzielt das System bessere Ergebnisse als Systeme, die nur einen gemeinsamen Klassifikator für die beiden Aufgaben verwenden.

### 4.3. Zusammenfassung

Die in diesem Kapitel vorgestellten Arbeiten untersuchten Korreferenz zwischen Ereignissen. Diese Aufgabe unterscheidet sich von der Erkennung von Korreferenz zwischen Aktionen, wie sie in Kapitel 5 definiert werden. Die hier genannten Arbeiten verfolgten unterschiedliche Ziele (dokumentübergreifende und dokumentinterne Korreferenz, Untereigniserkennung) und arbeiten auf verschiedenen Korpora mit unterschiedlichen Bewertungsalgorithmen. Die Ergebnisse sind deshalb schlecht vergleichbar. Tabelle 4.2 gibt einen Überblick über die Arbeiten; ob sie dokumentübergreifende (CD) oder dokumentinterne (WD) Ereigniskorreferenz behandeln, welche Korpora sie benutzen und welche Ergebnisse die Bewertungsalgorithmen liefern.

## 5. Analyse

Das Ziel dieser Bachelorarbeit ist die Entwicklung eines Agenten für PARSE, der die Erkennung von Aktionskorreferenz in gesprochener Sprache ermöglicht. Dazu werden Aktionen und verschiedene Formen von Aktionkorreferenz definiert und die Aufgabe von der Erkennung von Ereigniskorreferenz, wie sie von den verwandten Arbeiten in Kapitel 4 untersucht wird abgegrenzt (Abschnitt 5.1). Anschließend wird in Abschnitt 5.2 untersucht, an welchen Merkmalen zweier Aktionen erkannt werden kann, ob diese gleich sind, und wie diese Merkmale bei dem Erreichen des Ziels genutzt werden können. Danach wird analysiert, aus welchen Gründen und unter welchen Umständen dieselbe Aktion mehrmals in einem Diskurs genannt werden kann (Abschnitt 5.3). In Abschnitt 5.4 werden Randfälle betrachtet und diskutiert, ob die Implementierung der Erkennung dieser Fälle im Rahmen dieser Bachelorarbeit umgesetzt werden kann. Mit den in diesem Kapitel gewonnenen Erkenntnissen wird im darauffolgenden Kapitel 6 eine Lösung für das Problem der Aktionskorreferenzauflösung entworfen.

### 5.1. Definition von Aktionskorreferenz

Um zu untersuchen, ob sich mehrere Nennungen von Aktionen auf dieselbe Aktionsidentität beziehen, muss zunächst definiert werden, wann Aktionen gleich sind. In der Forschung wurde Aktionskorreferenz noch nicht untersucht, Ereigniskorreferenz (*engl. event coreference*) hingegen schon. Ereignisse sind im Gegensatz zu den hier untersuchten Aktionen komplexe Geschehnisse, an denen mehrere Akteure beteiligt sein können. Aktionen im Sinne dieser Bachelorarbeit sind Handlungen, welche von einem Menschen oder System, wie dem ARMAR-III, ausgeführt werden. Aktionen werden meist durch ein Prädikat gebildet. Diesem Prädikat können der ausführende Akteur und Parameter hinzugefügt werden. Die Parameter fügen der Aktion zusätzliche Informationen hinzu. Diese sind vor allem Objekte, mit denen interagiert wird und Informationen zu Raum und Zeit. Sie beantworten die W-Fragen „was“, „wer“, „wo“, „wann“, „wie“ und „warum“ [Ou16]. Die Informationen, welche Aktionen im Diskurs vorkommen und welche Phrasen welche Rollen realisieren, stehen durch den Aktionserkenner aus [Ou16] zur Verfügung. Die Erkennung von Aktionen ist somit keine Aufgabe dieser Bachelorarbeit.

#### Beispiel: Aktionen und Ereignisse

Donald Trump won the *election*.

So ist das Wort „election“ im obigen Beispiel keine Aktion, wird im Forschungsgebiet der Ereigniskorreferenz aber als Ereignis betrachtet. Das Gewinnen der Wahl ist allerdings eine Aktion. Das Prädikat ist „won“, der Akteur ist „Donald Trump“ und der einzige Parameter ist „the election“, welcher angibt, was gewonnen wurde.

Um nun Gleichheit von Aktionen zu definieren werden zunächst bestehende Definitionen von Ereigniskorreferenz betrachtet. Eines der ältesten Kriterien für Ereignisidentität stammt vom Philosophen Donald Davidson:

*„[...] events are identical if and only if they have exactly the same causes and effects. Events have a unique position in the framework of causal relations between events in somewhat the way objects have a unique position in the spatial framework of objects.“[Dav69]*

Demnach sind zwei Ereignisse genau dann identisch, wenn sie die selbe Ursache und Wirkung haben. Ereignisse bilden aneinandergereiht eine Kausalitätskette, in die sie sich eingliedern. Diese Definition ist nicht auf die hier diskutierten Aktionen übertragbar. Es wird in einer Beschreibung von Aktionen nie die komplette Menge an Ursachen und Wirkungen von Aktionen angegeben und diese kann auch nicht abgeleitet werden.

Bagga und Baldwin [BB99] grenzen die Aufgabe der Ereigniskorreferenz von Informationsextraktion (*engl. Information extraction*) und Themenentdeckung und -verfolgung (*engl. Topic Detection and Tracking, TDT*) ab. Es besteht Korreferenz zwischen zwei Ereignisnennungen, wenn diese das selbe Ereignis in der realen Welt beschreiben. Bei der Informationsextraktion hingegen geht es darum, aus Freitext Informationen zu extrahieren und diese strukturiert abzuspeichern. Das Ziel von TDT ist es, Dokumenten in ein übergeordnetes Themengebiet einzuordnen. Bei dieser Aufgabe würden beispielsweise Wahlen von verschiedenen Ämtern zusammengefasst werden. In der Ereigniskorreferenzanalyse würden nur Wahlen zusammengefasst werden, in denen es um dasselbe Amt zur selben Zeit geht.

Laut Bejan und Harabagiu [BH10] sind zwei Ereignisnennungen korreferent, wenn sie dieselben Eigenschaften und Teilnehmer haben. Sie schlagen eine Ereignishierarchie vor, auf derer ersten Stufe die einzelnen Ereignisnennungen, auf der zweiten Stufe die einzelnen realen Ereignisse und auf der dritten Stufe generische Ereignisse stehen. In Abbildung 5.1 ist ein Ausschnitt aus der Hierarchie zu sehen. Es gibt sieben Ereignisnennungen  $em_i$ , die drei Ereignisse  $e_i$  von verschiedenen Käufen beschreiben.

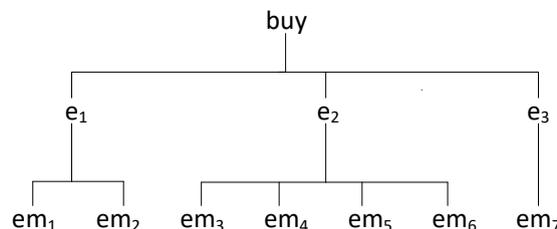


Abbildung 5.1.: Ereignishierarchie aus [BH10]

Im Gegensatz zu realen Ereignissen muss es bei Aktionen wie sie in dieser Arbeit betrachtet werden, keine reale Ausführung dieser existieren. Aktionen können im Imperativ vorliegen und somit Befehle beschreiben, zu denen noch keine Ausführung in der realen Welt existiert. Um nun eine Definition für Aktionskorreferenz herzuleiten, orientieren wir uns an der Hierarchie von Bejan und Harabagiu. Aktionskorreferenz sei für den Zweck dieser Arbeit wie folgt definiert:

**Definition 5.1.1.** Zwischen zwei Nennungen von Aktionen  $a_1$  und  $a_2$  mit  $a_1 \neq a_2$  liegt genau dann **Aktionskorreferenz** vor, wenn sie dieselbe Aktionsidentität referenzieren.

Das bedeutet, dass sich nicht nur die Prädikate auf die gleiche Handlung beziehen müssen, sondern dass die Aktion vom selben Akteur an denselben Objekten ausgeführt werden muss. Die Aktion muss auch zur gleichen Zeit am gleichen Ort ausgeführt werden. Zwei korreferente Beschreibungen von Aktionen im Diskurs resultieren in höchstens einer ausgeführten Aktion in der realen Welt. Neben der echten Aktionsidentität, bei der sich die Aktionsnennungen auf dieselbe Identität beziehen, können zwei Aktionen auch konzeptionelle Gleichheit haben. Diese sei wie folgt definiert:

**Definition 5.1.2.** Zwischen zwei Nennungen von Aktionen  $a_1$  und  $a_2$  mit  $a_1 \neq a_2$  liegt genau dann **konzeptionelle Gleichheit** vor, wenn sie gleichartige Interaktionen durch denselben Akteur mit denselben Objekten beschreiben.

So sind mehrfache Ausführungen einer Aktion, wie das Holen desselben Orangensafts aus demselben Kühlschrank durch denselben Holenden, konzeptionell gleich. Wenn hingegen zwei verschiedene Orangensäfte aus dem Kühlschrank geholt werden liegt keine konzeptionelle Gleichheit vor. Die schwächste Form von Gleichheit ist Prädikatengleichheit:

**Definition 5.1.3.** Zwischen zwei Nennungen von Aktionen  $a_1$  und  $a_2$  mit  $a_1 \neq a_2$  liegt genau dann **Prädikatengleichheit** vor, wenn sich ihre Prädikate auf das gleiche Konzept beziehen.

Beispielsweise sind das Bringen von Wasser und Bringen von Popcorn prädikatengleich. Prädikatengleichheit ist Voraussetzung für konzeptionelle Gleichheit und konzeptionelle Gleichheit ist wiederum Voraussetzung für volle Aktionskorreferenz. Die drei Stufen von Gleichheit werden in Abbildung 5.2 dargestellt.



Abbildung 5.2.: Gleichheitshierarchie für Aktionen

## 5.2. Erkennung von Konzeptioneller Gleichheit

Damit konzeptionelle Gleichheit zwischen zwei Aktionen vorliegt, müssen sich die Prädikate auf das gleiche Konzept beziehen. Zusätzlich zur Prädikatengleichheit muss die Aktion vom selben Akteur mit denselben Objekten durchgeführt werden. Das bedeutet, dass Korreferenz zwischen den Akteuren herrschen muss und dass auch Korreferenz zwischen den Objekten, die jeweils in der gleichen Beziehung zum Prädikat stehen, herrschen muss. Eine Herausforderung hierbei ist, dass diese Akteure und Objekte nicht nur mit Substantive, sondern auch mithilfe von (Personal-)Pronomen beschrieben werden können.

### 5.2.1. Prädikatengleichheit

Die Grundvoraussetzung für das Vorliegen von Aktionskorreferenz zwischen zwei Aktionen ist die Gleichheit ihrer Prädikate (siehe Abschnitt 2.1.7). Im Gegensatz zu Akteuren und Parametern liegt bei der Beschreibung von Aktionen immer ein Prädikat in Form eines Verbes vor (Ausnahmen siehe Abschnitt 5.4.2). Wie in 5.1 definiert, sollen zwei Aktionsnennungen als prädikatengleich betrachtet werden, wenn sich ihre Prädikate auf das gleiche

Konzept beziehen. Es ist dabei nicht erforderlich, dass die Prädikate durch das gleiche Wort beschrieben sind. Beispielsweise können die Wörter *move* und *go* sich auf die gleiche Art der Fortbewegung beziehen. Das Verb kann aufgrund von Konjugation in verschiedenen Verbformen vorliegen. So ändert sich die Endung der Verben in Abhängigkeit von Person, Numerus, Modus, Tempus und Genus Verbi (aktiv/passiv). Im Englischen enden Verben in der 3. Person mit einem *s*. Vergangenheitsformen werden bei regelmäßigen Verben mit *-ed* gebildet und von den Wörtern *have/has* bzw. *had* im present perfect bzw. past perfect begleitet. In den Sätzen „*The men have opened a window*“ und „*Armar open the fridge*“ sind die Prädikate „*have opened*“ und „*open*“ nach der Definition in dieser Bachelorarbeit gleich. Um von diesen Verbformen zu abstrahieren werden nur die Lemmata (siehe Abschnitt 2.1.5) der Verben betrachtet, um Prädikatengleichheit festzustellen. Hilfsverben wie *have* werden dabei nicht betrachtet.

Neben der Verwendung verschiedener Verbformen, soll es auch möglich sein, gleichartige Aktionen durch Verben mit verschiedenen Grundformen zu beschreiben. Insbesondere können gleiche Konzepte mithilfe von Synonymen und Troponymen (siehe Abschnitt 2.1.8) beschrieben werden.

Die Synonymbeziehung zweier Wörter ist symmetrisch. Wenn das Prädikat  $p_1$  ein Synonym von  $p_2$  ist, so ist auch  $p_2$  ein Synonym von  $p_1$ . Bei der Troponymbeziehung gibt es hingegen einen allgemeinen Oberbegriff und einen Unterbegriff, der das Ausführen der Aktion auf eine bestimmte Art beschreiben kann. Wenn eine Aktion zwei mal genannt wird, werden typischerweise eher in der ersten Nennung zusätzliche Informationen genannt, da diese in der zweiten Nennung redundant wären. Troponyme beinhalten mehr Informationen als ihre Oberbegriffe, da sie spezifischer sind. Deshalb ist anzunehmen, dass es häufiger vorkommt, dass in der ersten Nennung der Unter- und in der zweiten Nennung der Oberbegriff verwendet wird. Zur Veranschaulichung dient folgendes Beispiel:

#### Beispiel: Troponyme in erster und zweiter Nennung

- (1) Armar, unlock the door!<sup>a</sup> After you have opened the door...
- (2) Armar, open the door! After you have unlocked the door...

<sup>a</sup>Satzzeichen sind in der Eingabe von PARSE nicht vorhanden. Sie dienen in Beispielen lediglich der besseren Lesbarkeit.

In diesem Beispiel ist *unlock* ein Troponym von *open*. In (1) wird zuerst das Troponym verwendet. Dieses beinhaltet zusätzliche Informationen, nämlich dass die Tür zugeschlossen ist. Wenn diese Aktion beispielsweise zur zeitlichen Einordnung referenziert wird, müssen die genauen Informationen über die Art und Weise des Öffnens nicht nochmals wiederholt werden, da diese bereits in der ersten Beschreibung der referenzierten Aktion enthalten sind. Der Fall in (2) tritt hingegen in der Regel nicht auf. Bei der Referenz würden zusätzliche Informationen hinzukommen, die bei der eigentlichen Nennung der Aktion noch nicht bekannt gewesen sind.

In gesprochener Sprache verwendet ein Sprecher, nach eigener Beobachtung, meist das gleiche Verb, um das gleiche Konzept zu beschreiben. Allerdings können wegen Homonymie und Polysemie gleiche Verben unterschiedliche Bedeutungen haben. So kann das englische Verb *to run* rennen bedeuten, aber auch das Führen eines Geschäftes. Verben, zwischen denen eine Synonym- oder Troponymbeziehung vorliegt, müssen sich nicht auf das gleiche Konzept beziehen, da die Beziehung zwischen verschiedenen Bedeutungsausprägungen der Wörter herrschen kann. Daher kann selbst bei zwei gleichen Prädikaten nicht sicher gesagt werden, dass diese dasselbe Konzept beschreiben.

### 5.2.2. Korreferenz zwischen Akteuren

Akteure, welche Aktionen ausführen können, sind in der Regel Personen oder Systeme, wie der ARMAR-III. Der Akteur einer Aktion ist in Aktivsätzen das Subjekt und wird durch eine Nominalphrase gebildet. Er kann über die semantische Rolle identifiziert werden.

Es gibt mehrere Möglichkeiten, Referenz zum Akteur herzustellen. Zum einen kann der Akteur durch eine bestimmte oder unbestimmte Nominalphrase oder Eigennamen beschrieben werden. Zum anderen können auch Personalpronomen verwendet werden. In imperativen Sätzen ist der Akteur der Empfänger der Äußerung und muss in einem grammatikalisch korrekten Satz nicht aufgeführt werden. In Passivsätzen kann der Akteur ebenfalls ausgelassen oder mit der Präposition *by* aufgeführt werden.

#### Beispiel: Formen von Akteuren

- (1) The man opens the door.
- (2) A man opens the door.
- (3) John opens the door.
- (4) He opens the door.
- (5) John, open the door!
- (6) Open the door!
- (7) The door is opened by John.

In diesem Beispiel wird ein Akteur auf verschiedene Weisen beschrieben. In (1) wird eine bestimmte, in (2) eine unbestimmte Nominalphrase, in (3) ein Eigenname und in (4) ein Personalpronomen für den Akteur verwendet. Bei (5) und (6) handelt es sich bei *Open* um einen Imperativ. Der Akteur wird in (6) nicht explizit genannt. Bei (7) handelt es sich um einen Passivsatz mit Akteur.

Da derselbe Akteur auf verschiedene Weisen beschrieben werden kann, muss eine Korreferenzauflösung stattfinden. Dabei stellt vor allem die Verwendung von Personalpronomen als Anaphern eine Herausforderung dar. Die Korreferenzauflösung in PARSE wird bereits durch einen Agenten in einer anderen Arbeit [Hey16] realisiert. Die Informationen aus dieser Korreferenzauflösung kann verwendet werden, um festzustellen, ob Akteure gleich sind. Der hier zu entwickelnde Agent soll auch dann Ergebnisse liefern, wenn ihm diese Informationen nicht zur Verfügung stehen. Eine ausführliche Korreferenzauflösung von Entitäten ist jedoch nicht Teil dieser Arbeit. Daher wird bei Abwesenheit der Korreferenzinformationen nur eine grundlegende Annahmen über Akteure getroffen: Bei Verwendung von zwei Pronomen muss das Geschlecht und der Numerus übereinstimmen. Ein *he* kann sich nicht auf ein *she* beziehen. Werden zwei Eigennamen verwendet, sind die Akteure korreferent, wenn die Eigennamen gleich sind. Ist ein Akteur ein Personalpronomen und der andere Akteur ein Substantiv ist es möglich, dass beide Akteure korreferent sind.

In imperativen Sätzen, in denen der Akteur nicht explizit genannt wird ist dieser der Empfänger der Äußerung und wird bei einer Referenz auf die Aktion in einem nicht-imperativen Satz mit *you* referenziert. In Passivsätzen, in denen der Akteur ausgelassen wird, kann er nicht bestimmt werden.

### 5.2.3. Korreferenz zwischen Objekten

Neben dem Prädikat und dem Akteur können Aktionen Parameter enthalten, die die Aktion weiter beschreiben. Der Aktionserkennung weißt diesen Parametern verschiedene Rollen

zu. Objekte können Sachen oder Personen sein, mit denen der Akteur interagiert. Sie werden wie der Akteur durch Nominalphrasen gebildet und sind nach der Ausführung des Aktionserkenners mit dem Prädikat verbunden und erhalten vom Aktionserkennung die Rolle *WHAT* bzw. *WHO*. In der englischen Sprache gibt es drei verschiedene Arten von Objekten: Das **direkte** Objekt ist das Objekt mit dem interagiert wird. Das **indirekte** Objekt ist indirekt von der Aktion betroffen. Ein Prädikat kann nur dann ein indirektes Objekt besitzen, wenn es auch ein direktes Objekt besitzt. Das indirekte Objekt ist dann der Empfänger der Handlung. **Präpositionalobjekte** sind Objekte, die mithilfe einer Präposition wie *from* oder *in* eingeführt werden.

Im Gegensatz zu Akteuren können Objekte obligatorisch oder optional sein. Wenn eine Aktion bereits genauer beschrieben worden ist, ist es bei einer Wiederholung zur Bezugnahme auf diese Aktion nicht immer nötig, alle Objekte zu wiederholen, sofern der Referent durch sein Prädikat und seinen Akteur bereits eindeutig ist.

#### Beispiel: Auslassen von Parametern

Alice is waiting for Bob. She missed the bus because she was waiting.

In diesem Beispiel ist *for Bob* ein Präpositionalobjekt. Es muss in der zweiten Nennung der Aktion nicht wiederholt werden, da bereits bekannt ist, auf wen Alice gewartet hat. Trotz des fehlenden Objektes in der zweiten Nennung von *waiting* beziehen sich beide Nennungen auf dieselbe Aktion und sind somit korreferent. Würde Alice auf mehrere Personen warten, müsste es wiederholt werden, um deutlich zu machen, auf welches Warten Bezug genommen wird.

Falls die jeweiligen Objekte in beiden Aktionen vorhanden sind und Korreferenzinformationen über diese vorliegen, können diese Informationen verwendet werden. Falls die Informationen nicht vorliegen, können einige Eigenschaften der Objekte verwendet werden, um zu entscheiden, ob Korreferenz zwischen ihnen besteht. Es ist anzunehmen, dass Objekte gleich sind, wenn sie durch die gleichen Wörter gebildet werden. Es ist ebenfalls möglich, dass Objekte zwar wiederholt, aber bei der Wiederholung durch eine generellere Beschreibung Bezug auf sie genommen wird. Es ist nicht nötig, alle Details, in Form von Adjektiven, über ein Objekt zu wiederholen, wenn nur ein Objekt als möglicher Kandidat für einen Referenten in Frage kommt. Beispielsweise kann ein bereits im Diskurs eingeführter roter Becher einfach mit *cup* referenziert werden, sofern es nur diesen einen Becher gibt. Eine weitere Möglichkeit Objekte allgemeiner zu beschreiben ist die Verwendung von Hyperonymen, da ein Hyperonym ein Objekt genereller als das Hyponym beschreiben. So könnte der rote Becher auch einfach mit *tableware* referenziert werden.

Bei der Referenzierung auf ein im Diskurs bereits eingeführtes Objekt wird meist ein bestimmter Artikel (*the*) als Determinierer verwendet [JM09]. Bei neu eingeführten Objekten, die noch nicht bekannt sind, wird ein unbestimmter Artikel (*a/an*) verwendet. Bei Objekten im Plural und bei unzählbaren Dingen (z.B. Wasser) tritt kein unbestimmter Artikel auf. Wenn auf eine bereits genannte Aktion, die ein unbestimmtes Objekt beinhaltet Bezug genommen wird, kann dies auch durch erneute Nutzung eines unbestimmten Artikels im Objekt realisiert werden. Beispielsweise werden in Befehlen wie „Armar get me a glass. After you have got me a glass do...“ die beiden Nennungen des Glases vom indefiniten Artikel *a* begleitet. Konzeptionelle Gleichheit zweier Aktionen kann durch Betrachtung der Artikel nur ausgeschlossen werden, wenn ein bestimmter Artikel in der ersten und ein unbestimmter Artikel in der zweiten Nennung vorliegt, wie durch die Beispiele in Tabelle Tabelle 5.1 deutlich wird. Ein Problem was hier auftritt ist, dass der Korreferenzauflöser annimmt, dass keine Korreferenz zwischen zwei Nominalphrasen mit unbestimmten Artikeln herrschen kann. Deshalb muss auch bei Verfügbarkeit der Korreferenzinformationen dieser Fall gesondert betrachtet werden.

Tabelle 5.1.: Kombinationen von Artikeln in zwei Aktionen

1. Art.	2. Art.	möglich	Beispiel
the	the	ja	Get me <b>the</b> glass. After you have got me <b>the</b> glass...
a	the	ja	Get me <b>a</b> glass. After you have got me <b>the</b> glass...
the	a	nein	Get me <b>the</b> glass. After you have got me <b>a</b> glass...
a	a	ja	Get me <b>a</b> glass. After you have got me <b>a</b> glass...

Objekte können auch mithilfe von Personalpronomen gebildet werden. Hier gelten die gleichen Annahmen wie bei dem Vergleich der Akteure, wenn die Korreferenzinformationen nicht verfügbar sind.

#### 5.2.4. Gleichheit sonstiger Parameter

Weitere Parameter, die vom Aktionserkennung annotiert werden, beantworten die Fragen „wann?“, „wo?“, „wie?“ und „warum?“. Sie werden durch Präpositionalphrasen und Adverbialphrasen gebildet. Sie sind immer optional. Da sie die Aktion weiter spezifizieren, wird angenommen, dass sie sie vorwiegend in der ersten Nennung der Aktion verwendet werden. In der zweiten Nennung ist es nicht nötig, diese Parameter nochmals anzuführen, sofern die Referentenaktion bei Weglassen dieser eindeutig ist. Abschnitt 5.2.4 führt auf, welchen Teilen der Aktionserkennung welche Parameterrollen zuweist.

##### Beispiel: Sonstige Parameter

Alice is eating slowly in the kitchen on Tuesday

Hier gibt *slowly* die Art, *in the kitchen* den Ort und *on Tuesday* die Zeit an.

Damit zwei Aktionen konzeptionell gleich sein können, müssen sich die Parameter, die die Frage „wie?“ beantworten, in beiden Aktionen auf das jeweils Selbe beziehen. Die Parameter, die Raum und Zeit beschreiben müssen nur für die echte Aktionskorreferenz gleich sein, da die gleiche Aktion zu unterschiedlichen Zeiten an unterschiedlichen Orten mehrmals ausgeführt werden kann.

##### 5.2.4.1. Parameter der Zeit

Parameter, die mit der Rolle *WHEN* versehen werden, spezifizieren die Zeit, zu der die Aktion geschieht. Sie werden meist durch eine Präpositionalphrase eingeleitet und haben die semantische Rolle *AM-TMP*. In der Einsatzumgebung von PARSE wird die Zeit eher selten spezifiziert, da Anweisungen wie „Get me the oranges“ in der Regel sofort ausgeführt werden sollen. Bei einer erneuten Nennung der Aktion zur Referenzierung wird die Zeit noch seltener genannt.

Wenn zwei verschiedene Ausdrücke zur Bestimmung der Zeit verwendet werden, können sich diese trotzdem auf denselben Zeitpunkt beziehen. So können sich „*the third of July*“ und „*Tuesday*“ auf denselben Tag beziehen. Zeitpunkte können auch mehr oder weniger spezifisch beschrieben werden, wie „*yesterday*“ und „*yesterday afternoon*“.

##### 5.2.4.2. Parameter des Raums

Parameter, die mit der Rolle *WHERE* versehen werden, geben an, wo die Aktion ausgeführt wird. Sie werden ebenfalls meist durch Präpositionalphrasen eingeleitet und haben

Tabelle 5.2.: Unterordnende Konjunktionen

Konjunktionstyp	unterordnende Konjunktionen
Zeit	before, after, since, until, while, when
Ursache	because, as, since
Ort	where, wherever
Bedingung	if, whenever, as long as, unless

die semantische Rolle *AM-LOC*. Sie sind nicht mit Präpositionalobjekten wie *from the fridge* zu verwechseln. Im Anwendungsbereich von PARSE kommen sie vor, wenn es ähnliche Gegenstände an mehreren Orten gibt. Wenn es Gläser auf einem Tisch und in einem Schrank gibt, kann man mit diesen Parametern spezifizieren, welche Gläser man meint.

### 5.2.4.3. Umstandsparameter

Diese Parameter werden durch Adverbien gebildet und haben die Rolle *HOW*. Sie spezifizieren die Art und Weise, auf die eine Aktion ausgeführt werden soll. Umstandsparameter werden bei einer Referenz auf eine bereits spezifizierte Aktion meist weggelassen, da es selten vorkommt, dass sich zwei verschiedene Aktionen nur durch Umstandsparameter unterscheiden.

## 5.3. Erkennung von Aktionsidentität

Wenn eine Aktion, die nur einmal ausgeführt werden soll, mehrmals im Diskurs genannt wird, kann dies mehrere Gründe haben. Diese Gründe werden im Folgenden beschrieben und Möglichkeiten zur Erkennung dieser untersucht.

### 5.3.1. Temporale, lokale, und kausale Beziehungen

Oft ist es nötig, Aktionen wiederholt zu nennen, um eine neue Aktion in eine Zeitreihe einzuordnen. Wenn Aktion *a* bereits genannt worden ist, und man ausdrücken möchte, dass eine weitere Aktion *b* direkt vor *a* auszuführen ist, muss auf *a* Bezug genommen werden. Hierfür muss die Aktion in einem Temporalsatz wiederholt genannt werden. Es kann damit ausgedrückt werden, dass eine Aktion vor, nach oder auch parallel zu einer anderen Aktion stattfindet. Weiterhin kann die Kohärenz durch kausale Beziehungen hergestellt werden. Wenn eine Aktion ein Grund für eine andere Aktion ist, kann man dies durch einen Kausalsatz ausdrücken. In diesem Nebensatz wird diese Aktion als Grund aufgeführt. Falls die Aktion bereits genannt worden ist, soll sie hier nicht wiederholt ausgeführt werden, sondern dient lediglich als Einordnung. Ein weiterer Grund für das wiederholte Nennen einer Aktion kann die Bezugnahme auf einen Ort sein, an dem eine Aktion ausgeführt worden ist. Bedingungen, können ebenfalls bereits genannte Aktionen sein.

#### 5.3.1.1. Erkennung von Strukturen zur Einordnung

Wenn eine Aktion aus diesen Gründen wiederholt wird, steht sie in einem Adverbialsatz. Adverbialsätze werden durch eine unterordnende Konjunktion (*engl. subordinating conjunction*) eingeleitet [Eas05]. Eine Liste mit einigen wichtigen unterordnenden Konjunktionen ist in Tabelle 5.2 gegeben.

Die unterordnende Konjunktion steht im Englischen direkt vor dem Subjekt des Adverbialsatzes. In einem Adverbialsatz können mehrere Aktionen enthalten sein, die durch ein *and* verknüpft sind. In diesem Fall steht die unterordnende Konjunktion nur vor der ersten Aktion.

**Beispiel: Relativsätze**

- (1) Get me the oranges **after** you have opened the fridge
- (2) It's loud **because** you have turned on the radio
- (3) This is the room **where** Alice killed Bob
- (4) **If** you have bought oranges bring them to me

Das obige Beispiel zeigt (1) einen Temporalsatz, (2) einen Kausalsatz, (3) einen Lokalsatz und (4) einen Bedingungssatz. Die unterordnenden Konjunktionen sind fett gedruckt. Die Aktionen in den Adverbialsätzen dienen der Textkohärenz und beziehen sich auf bereits genannte Aktionen, wenn diese existieren. Es ist auch möglich, Aktionen in diesen Nebensätzen zum ersten mal zu nennen. In diesem Fall existiert keine vorangegangene Aktionsnennung mit der die Nennung im Nebensatz korreferent ist. Die Aktion kann auch zuerst in einem Nebensatz und anschließend in einem Hauptsatz referenziert werden, wie an folgendem Beispiel zu sehen ist:

**Beispiel: Nennung der Referenz zuerst**

Armar, take the orange juice after you have opened the fridge. Open the fridge now.

**5.3.1.2. Betrachtung der Zeitformen**

Weiterhin fällt auf, dass sich der Tempus ändern kann, wenn man sich auf eine in der Vergangenheit bereits ausgeführten Aktion bezieht. Wenn die anderen Anweisungen im Diskurs im Präsens ausgedrückt sind, wird in der Regel das Perfekt (*engl. present perfect*) oder wieder das Präsens verwendet, um auf bereits vollendete Aktionen Bezug zu nehmen. Ist der Diskurs im Präteritum (*engl. simple past*) verfasst, liegen Verweise auf bereits beschriebene Aktionen meist im Plusquamperfekt (*engl. past perfect*) oder im Präteritum vor. Falls also eine Aktion im Nebensatz in einer auf einer Zeitreihe später einzuordnenden Zeitform befindet als im Hauptsatz, kann angenommen werden, dass sie sich nicht auf diese Aktion bezieht.

**5.3.1.3. Korreferenz zwischen Aktionen in Haupt- und Nebensatz**

Es ist anzunehmen, dass keine Korreferenz zwischen einer Aktion in einem Adverbialsatz und der Aktion im Hauptsatz, zu dem der Adverbialsatz gehört, existiert. In Sätzen wie „Do *a1* after you have done *a2*.“ würde es keinen Sinn ergeben, wenn *a1* und *a2* dieselbe Aktion beschreiben würden. Da in dieser Arbeit gesprochene Sprache betrachtet wird und somit keine Satzzeichen zur Verfügung stehen kann nicht auf einfache Art und Weise erkannt werden, zu welchem Hauptsatz ein Adverbialsatz gehört. Der Adverbialsatz kann im Englischen vor oder hinter dem Hauptsatz stehen. Die Eingabe „open the fridge after you have opened the fridge get me a glass“ mit drei Aktionsnennungen kann auf zwei Arten interpretiert werden:

- „Open the fridge after you have opened the fridge. Get me a glass.“
- „Open the fridge. After you have opened the fridge get me a glass.“

In der ersten Interpretation können die beiden Aktionen in denen es um das Öffnen des Kühlschranks geht nicht korreferent sein, im Gegensatz zur zweiten Interpretation. Da in PARSE momentan keine Informationen über Satzgrenzen zur Verfügung stehen, kann diese Eigenschaft nur bedingt ausgenutzt werden, um festzustellen, dass Aktionen verschieden sind.

#### 5.3.1.4. Aktionskorreferenz in Kontrollstrukturen

In Strukturen wie „*IF a<sub>1</sub> (THEN) a<sub>2</sub> ELSE a<sub>3</sub>*“ kann keine Aktionskorreferenz zwischen *a<sub>1</sub>*, *a<sub>2</sub>* und *a<sub>3</sub>* vorliegen. Der Ausdruck *IF a<sub>1</sub> (THEN) a<sub>2</sub> ELSE a<sub>2</sub>* wäre zwar semantisch korrekt, würde aber immer unabhängig davon, ob *a<sub>1</sub>* wahr oder falsch ist zu *a<sub>2</sub>* auswerten. In natürlicher Sprache würde man einen solchen Ausdruck deshalb nicht verwenden. Tautologien wie beispielsweise „*If Armar is opening the door then he is opening the door*“, enthalten keine Informationen. Daher ist zu erwarten dass sie in natürlicher Sprache kaum vorkommen. Man kann sich dies zunutze machen und in diesem Fall Aktionskorreferenz ausschließen.

Strukturen wie „*IF a<sub>1</sub> (THEN) a<sub>2</sub> and a<sub>3</sub> ELSE a<sub>2</sub> and a<sub>4</sub>*“ können in natürlicher Sprache auftreten. Im Befehl „*If it's raining open the fridge and bring me orange juice else open the fridge and bring me apple juice*“ besteht Korreferenz zwischen den beiden Aktionen *open the fridge*. Der Satz könnte zwar durch herausziehen dieses Befehls zu „*Open the fridge if it's raining bring me orange juice else bring me apple juice*“ vereinfacht werden. In spontan gesprochener Sprache ist es jedoch vorstellbar, dass er dennoch so vorkommt. Aktionskorreferenz kann in Bedingungssätzen also nur ausgeschlossen werden, wenn im *THEN*-Teil und im *ELSE*-Teil genau eine Aktion steht.

#### 5.3.2. Relativsätze

Aktionen können in Relativsätze erneut genannt werden, um ein Subjekt oder Objekt im Hauptsatz näher zu bestimmen. Wenn es mehrere gleichartige Gegenstände gibt und mit einem bereits eine Aktion ausgeführt worden ist, kann man diese Aktion wiederholen, um sich auf jenen Gegenstand zu beziehen. Relativsätze werden im Englischen meist durch die Relativpronomen *who/whom* für Personen, *which* für Dinge oder *that* für Personen oder Dinge eingeleitet. Die Nominalphrase vor dem Relativpronomen nimmt in diesem Fall die Rolle des Akteurs oder eines Objekts in der Aktion im Relativsatz an. Sie ist der Akteur (bei Sätzen im Aktiv), wenn auf das Relativpronomen das Prädikat folgt und ein Objekt, wenn ein Subjekt im Relativsatz vorhanden ist.

##### Beispiel: Relativsätze

open the left door close the door which you have opened

Im Beispiel beziehen sich die beiden Prädikate *open* auf die selbe Aktionsidentität. Das Objekt des zweiten *open*-Prädikats ist *the door*.

Das Relativpronomen kann im Englischen allerdings auch weggelassen werden, wie in folgendem Beispiel:

##### Beispiel: Relativsatz ohne Relativpronomen

bake a cake bring me the cake you have baked

Wenn das Relativpronomen wegfällt tritt in unserem Fall Zweideutigkeit auf. Da uns bei der Analyse gesprochener Sprache keine Satzzeichen zur Verfügung stehen, kann der Satz aus dem Beispiel auf zwei Arten interpretiert werden:

- Bake a cake. Bring me the cake you have baked.
- Bake a cake. Bring me the cake. You have baked.

In der ersten Interpretation liegt ein Relativsatz vor, mit dem ausgedrückt wird, welcher Kuchen gebracht werden soll. In der zweiten Interpretation ist „*you have baked*“ ein unabhängiger Hauptsatz, der aussage, dass etwas gebacken wurde. Da es nicht auf einfache Weise möglich ist, Relativsätze ohne Relativpronomen zu erkennen, kann dieser Fall nicht weiter untersucht werden. Es wird deshalb angenommen, dass Relativsätze immer von einem Pronomen eingeleitet werden.

Demnach können Aktionsnennungen zu anderen Aktionsnennungen, zu denen sie konzeptuell gleich sind korreferent sein, wenn sie von einem Relativpronomen begleitet werden. Bei Aktionen in Relativsätzen muss die gleiche Bedingung an die Zeitform gelten, wie im vorangegangenen Abschnitt.

### 5.3.3. Spezifizierung und Generalisierung

Aktionen können spezifiziert und generalisiert werden, um diesen weitere Aspekte hinzuzufügen oder um sie umzuformulieren [DG03]. Die Spezifizierung bzw. Generalisierung einer Aktion folgt dabei direkt auf ihre erste Beschreibung. Die Definitionen werden grob aus [DG03] übernommen.

**Definition 5.3.1.** Zwei aufeinanderfolgende Nennungen  $M1$  und  $M2$  derselben Aktion  $a$  stehen in einer **Spezifizierungsbeziehung**, genau dann wenn  $M2$  neue Informationen über  $a$  vermittelt, die noch nicht in  $M1$  enthalten sind.

**Definition 5.3.2.** Zwei aufeinanderfolgende Nennungen  $M1$  und  $M2$  derselben Aktion  $a$  stehen in einer **Generalisierungsbeziehung**, genau dann wenn  $M2$  keine neuen Informationen über  $a$  vermittelt, die noch nicht in  $M1$  enthalten sind.

#### Beispiel: Spezifizierung

- (1) Armar **cook** me **some food**. **Bake a cake**.
- (2) Armar **bake a cake**. **Bake it** in the oven.
- (3) Armar **cook** me **some food**. **Next bake a cake**.

In diesem Beispiel liegt in (1) Troponymie zwischen *bake* und *cook* und Hyponymie zwischen *cake* und *food* vor. Der zweite Befehl enthält also die zusätzlichen Informationen, was genau zubereitet wird (ein Kuchen) und auf welche Art und Weise dies geschehen soll (backen). In (2) gibt der zweite Befehl Aufschluss darüber, mit welchem Hilfsmittel der Kuchen gebacken werden soll. In beiden Beispielen liegt eine Spezifizierung der Aktion vor. Es soll jeweils nur ein Kuchen gebacken werden. In (1) ist zudem auffällig, dass Korreferenz zwischen den indefiniten Ausdrücken *food* und *cake* vorliegt. Normalerweise bilden Entitäten mit indefiniten Artikeln keine Referenz auf bereits eingeführte Entitäten, sondern führen selbst Entitäten ein. Spezifizierung liegt dann vor, wenn jede Phrase im zweiten Befehl ein Troponym, ein Hyponym, eine Anapher oder eine Wiederholung der korrespondierenden Phrasen im ersten Satz ist, oder es dort keine entsprechende Phrase gibt. Spezifizierungen können durch Phrasen wie *more precisely* expliziert werden. Durch Wörter wie *next*, *then*, *afterwards*, *before* zwischen den beiden Aktionen in (3) kann deutlich gemacht werden, dass keine Aktionskorreferenz vorliegt, da es sich dann bei der zweiten Aktionsnennung um eine neue Aktion handelt. Ebenfalls liegt keine Aktionskorreferenz vor, wenn die Aktionen durch die Konjunktion *and* verbunden sind.

Tabelle 5.3.: Schlüsselwörter für Wiederholung

Typ	Schlüsselwörter
Adverbialphrasen	again, once again, once more, over again
Nominalphrasen	a second time, a third time, another time, ...

### Beispiel: Generalisierung

Armar is **baking a cake** therefore he is **preparing food**

In diesem Beispiel liegt Generalisierung vor. Der zweite Satz ist allgemeiner als der erste, da *prepare* und *food* Hyperonyme von *bake* und *cake* sind. Der zweite Satz liefert keine neuen Informationen. Generalisierungen werden zur Umschreibung von Sachverhalten verwendet. Der zweite Satz kann bestimmte Aspekte hervorheben, beispielsweise dass es sich bei dem Kuchen um etwas essbares handelt. Generalisierungen liegen dann vor, wenn jede Phrase im zweiten Befehl ein Hyperonym, eine Anapher oder eine Wiederholung der korrespondierenden Phrasen im ersten Satz ist. Sie werden oft von Wörtern wie *therefore* begleitet. Eine Generalisierung folgt immer logisch aus der vorangegangenen Beschreibung.

In Befehlen im Imperativ, welche hauptsächlich im Anwendungsgebiet von PARSE vorzufinden sind, kommen Generalisierung in der Regel nicht vor. Aus diesem Grund werden im weiteren Verlauf dieser Arbeit nur Spezifizierungen weiter betrachtet.

#### 5.3.4. Korrektur von falschen Aktionen

In gesprochener Sprache treten häufig Situationen auf, in denen man sich verspricht und falsche Wörter verwendet. Im Gegensatz zu niedergeschriebener Sprache auf Papier oder elektronischen Geräten kann man in gesprochener Sprache diese Fehler nicht einfach durchstreichen oder löschen. Um Fehler zu korrigieren muss man sagen, dass man einen Fehler gemacht hat und das was man eigentlich ausdrücken wollte wiederholen. Möchte man beispielsweise ausdrücken, dass der Kühlschrank geöffnet werden soll, kann man dies durch den Befehl „*open the fridge*“ tun. Verspricht man sich und sagt stattdessen „*open the bridge*“ kann man dies verbessern, indem man die eigentliche Intention mit einem Ausdruck, der die Falschheit der ersten Äußerung ausdrückt, sagt. Beispielsweise „*open the bridge I mean open the fridge*“. Der Ausdruck *I mean* deutet darauf hin, dass es sich um eine Korrektur der ersten Aktion handelt. Die beiden Aktionsnennungen *open the bridge* und *open the fridge* beziehen sich auf dieselbe Aktion in der realen Welt, nämlich auf das Öffnen des Kühlschranks. Es liegt Aktionskorreferenz zwischen den beiden Befehlen vor. Die Korrektur einer Aussage wird durch Signalphrasen wie *i mean*, *no* oder *that was wrong* eingeleitet. Die korrigierte Aussage bezieht sich auf das zuletzt Gesagte. Das Erkennen solcher Unstetigkeiten ist jedoch sehr aufwändig und würde den Rahmen dieser Bachelorarbeit überschreiten. Daher wird dieser Punkt nicht weiter betrachtet.

#### 5.3.5. Ausschlusskriterien für Aktionskorreferenz

Unabhängig von den oben genannten Gründen existieren Schlüsselwörter und -phrasen, die innerhalb einer Beschreibung auftreten können, um auszudrücken, dass es sich um eine neue Aktion handelt. Diese liegen in Form von Nominal- und Adverbialphrasen vor und werden in Tabelle 5.3 aufgelistet. Diese Phrasen geben an, dass sich eine Aktionsnennung auf eine erneutes Ausführung bezieht. Falls eine der Schlüsselphrasen in einer Aktionsnennung auftritt, kann daraus abgeleitet werden, dass es sich bei dieser Aktionsnennung nicht um einer bereits eingeführte Aktion handelt. Die Aktionsnennung kann aber mit später eingeführten Aktionen korreferent sein.

## 5.4. Spezialfälle

In einigen Fällen wird auf eine Aktion verwiesen ohne dabei die Wörter, mit denen die referenzierte Aktion beschrieben wird oder mit sprachlich verwandten Wörtern zu benutzen. Die Referenzaktion kann sich in der Satzstruktur von der Referentenaktion unterscheiden. Insbesondere ist Korreferenz zwischen einem Satz und einem einzelnen Wort möglich. Diese Randfälle werden in diesem Kapitel erläutert und deren Umsetzbarkeit analysiert.

### 5.4.1. Korreferenz zwischen Aktion und Demonstrativpronomen

Aktionen können im Englischen mit dem Demonstrativpronomen *that* beschrieben werden. Dieses bezieht sich immer auf die Aktion im letzten Satz. In der Regel bezieht sich *that* auf eine Aktion, wenn es von den Verben *do* oder *happen* oder Synonymen von diesen Wörtern begleitet wird und kein Substantiv darauf folgt. Die Verwendung eines Demonstrativpronomens als Aktion kann aus den gleichen Gründen erfolgen, wie das wiederholte Nennen der Aktion. Die Wörter *do that* können in diesem Fall durch die Ausformulierung der referenzierten Aktion ersetzt werden. Das Demonstrativpronomen nimmt im Satz die Rolle des Objektes ein, wenn das Verb *do* ist und die Rolle des Subjekts, wenn das Verb *happen* ist.

#### Beispiel: *that* bezieht sich auf eine Aktion

- (1) Open the fridge. After you have **done that** close it again.
- (2) Armar baked a cake. **That happened** yesterday.
- (3) Alice bought a new car. Bob liked **that**.

In (1) bezieht sich *done that* auf *open the fridge*. Der Befehl ist gleichbedeutend mit „*Open the fridge after you have opened the fridge close it again*“. (2) ist gleichbedeutend mit „*Armar baked a cake he baked it yesterday*“. Es handelt sich um eine Spezifizierung. In (3) kann sich *that* auf den Kauf des Autos von Alice beziehen. Das Demonstrativpronomen wird hier nicht von einem *do* begleitet. Da sich *that* auch auf Entitäten beziehen kann, ist es in diesem Beispiel jedoch schwer zu entscheiden, ob sich das Wort auf die Aktion oder das Auto bezieht. Es werden daher nur Vorkommnisse des Wortes *that* betrachtet, die in einer Aktion vorkommen, in der das Prädikat *do* oder *happen* oder ein Synonym dieser Wörter ist.

### 5.4.2. Korreferenz zwischen Aktion und Substantivierung

Aktionen können nicht nur durch Prädikate mit Akteuren und Parametern, sondern auch durch Substantive beschrieben werden. Die einfachste Form der Substantivierung ist im Englischen die Verwendung des Gerundiums, welches durch Anhängen des Suffixes *-ing* an den Infinitiv gebildet wird. Es gibt aber auch Substantivierungen, die mit anderen Endungen, wie *-ion*, *-ation* oder *-ment* gebildet werden. Bei Substantivierungen muss der Akteur nicht unbedingt genannt werden. Parameter fallen bei Substantivierungen ebenfalls oft weg. Wenn Objekte genannt werden, geschieht dies meist durch das Anhängen des Objekts mit dem Wort *of*.

Da Substantivierungen nicht als Aktionen vom Aktionserkennung erkannt werden und es schwer ist, Substantivierungen zu erkennen, werden Referenzen von Substantiven auf Aktionen in dieser Arbeit jedoch nicht betrachtet.

### 5.4.3. Partielle Aktionskorreferenz

Mehrere Teilaktionen können zu einer komplexeren Aktion zusammengefasst werden und eine Aktion kann in mehrere Teilschritte zerlegt werden, ähnlich wie Eltern- und Unterereignisse in [HVM<sup>+</sup>13]. Das mehrfache Ausführen einer Aktionen, wie das Bringen mehrerer Tassen, kann zusammengefasst werden, indem man die Aktion einmal ausdrückt und sagt, wie oft die Aktion ausgeführt werden soll. Ähnliche Aktionen mit dem gleiche Prädikat können durch Allquantifizierung in einem Ausdruck zusammengefasst werden. Aktionen können Abläufe aus mehreren Teilschritten beschreiben. Wir sprechen von partieller Aktionskorreferenz, wenn eine Aktionsnennung im Diskurs eine Teilaktion einer anderen Aktionsnennung referenziert. Dabei kann die Teilaktion vor oder nach der zusammengesetzten Aktion beschrieben werden.

#### Beispiel: Partielle Aktionskorreferenz

- (1) Get me three cups. Get me a red cup, a yellow cup and a green cup.
- (2) Get me a red cup. Then get me a blue cup. After you have got me both cups open the fridge.
- (3) Clap twice. After you have clapped for the first time jump.
- (4) Armar, bake me a cake. Put flour into a bowl. Then put two eggs into the bowl. Then...

Abschnitt 5.4.3 zeigt einige Sätze, in denen partielle Aktionskorreferenz auftritt. In (1) gibt es die drei Aktionen, in denen eine rote Tasse, eine gelbe Tasse und eine grüne Tasse gebracht werden soll. „*Get me three cups*“ fasst die drei Aktionen in einer Aktion zusammen. In (2) werden *red cup* und *green cup* zu *both cups* zusammengefasst. Im Gegensatz zu (1) werden zuerst die Unterereignisse beschrieben. In (3) wird die Aktion des Klatschens zwei mal ausgeführt. Die Referenz bezieht sich nur auf das erste Ausführen der Aktionen. (4) beschreibt Teilschritte des Kuchenbackens. Die Aktionen, die die Teilschritte beschreiben beziehen sich auf den ersten Satz. Es soll nur ein Kuchen gebacken werden.

Fälle, in denen eine zusammengesetzte Aktion weiter spezifiziert wird, wie in (4), und die Teilaktionen andere Wörter benutzen als die Elternaktion, können ohne externes Wissen nicht aufgelöst werden. In Fällen wie (1) und (2) beinhaltet eine der Aktionsnennungen eine Quantifizierung eines Objekts durch eine Kardinalzahl oder Adjektive wie *all*, *some* oder *both*. Wenn in der zusammengesetzten Aktion ein Objekt mit  $n$  quantifiziert ist, gibt es  $n$  Teilaktionen. Wenn es mit *both* quantifiziert ist, gibt es genau zwei. In Fällen wie (3) wird die Aktion mehrmals ausgeführt, was durch Angabe der Wiederholungen (im Beispiel: zwei) ausgedrückt wird. In der Referenz wird in diesem Fall eine Ordinalzahl verwendet, um auf eine einzelne Ausführung zu referenzieren.

Aus Zeitgründen wird auch dieser Spezialfall nicht weiter untersucht.

## 6. Entwurf und Implementierung

Im Rahmen dieser Bachelorarbeit wurde mithilfe der in Kapitel 5 erlangten Erkenntnisse ein Vorgehen zur Auflösung von Aktionskorreferenz entworfen und als Agent von PARSE (siehe Kapitel 3) implementiert. Die Implementierung erfolgte in der Sprache Java. In diesem Kapitel wird der Entwurf des Agenten und dessen Implementierung beschrieben. In Abschnitt 6.1 wird der Ablauf der Aktionskorreferenzauflösung beschrieben. In Abschnitt 6.3 werden wichtige Implementierungsdetails erläutert.

### 6.1. Ansatz zur Auflösung von Aktionskorreferenz

Im folgenden wird der Lösungsansatz zur Auflösung von Aktionskorreferenz grob skizziert. Die einzelnen Teilschritte werden in den darauffolgenden Abschnitten genauer erläutert. Die Voraussetzung für diesen Ansatz ist die Verfügbarkeit folgender Informationen:

- Die zu untersuchende Eingabe in Textform.
- Die Wortarten der Wörter.
- Die Phrasenzugehörigkeit der einzelnen Wörter.
- Die Lemmata der Wörter
- Die Befehlsnummern (*instruction numbers*) der Wörter. Ein Befehl beginnt bei einer Verbalphrase und endet vor der nächsten. Wörter wie *and* spalten einen Befehl in mehrere Befehle auf.
- Die Informationen über Aktionen, wie sie in [Ou16] definiert sind. Dabei bestehen Aktionen aus Prädikat, Akteur und Parametern die die W-Fragen beantworten. Diese werden im Folgenden **WHAT-**, **WHO-**, **HOW-**, **WHERE-**, **WHEN-**Parameter genannt.
- Informationen über semantische Relationen.

Zusätzlich können Informationen über Korreferenzen zwischen Entitäten und Informationen über bedingte Anweisung das Ergebnis verbessern.

Das Vorgehen des Werkzeugs ist in Abbildung 6.1 dargestellt. Um alle Korreferenzen zwischen Aktionen vollständig auflösen zu können, müssen alle Aktionen, die im Diskurs aufgeführt werden miteinander verglichen werden. Dafür müssen die Aktionen zuerst aus dem Eingabegraphen gelesen werden. Nachdem die Aktionen aus der Eingabe extrahiert

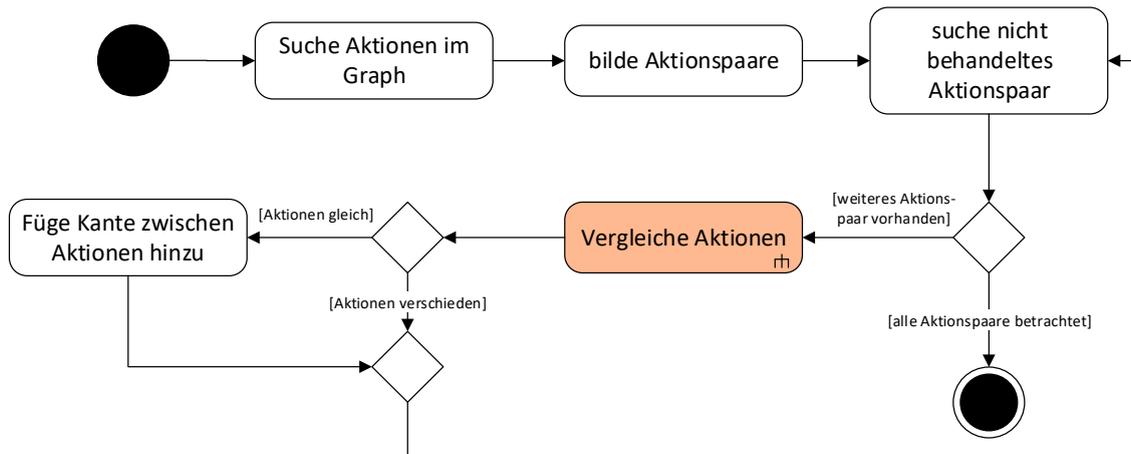


Abbildung 6.1.: Vorgehen des Agenten

worden sind, werden diese paarweise verglichen. Dazu wird jede Aktion  $a_i$  mit jeder anderen Aktion  $a_j$  verglichen, die vor dieser Aktion im Diskurs aufgeführt wird, das heißt für die gilt  $instructionNumber(a_j) \leq instructionNumber(a_i)$ . (Zwei Aktionen können sich Verben teilen. In diesem Fall ist die Befehlsnummer gleich.) Durch dieses Vorgehen kann einfach zwischen der erst- und der zweitgenannten Aktion unterschieden werden. Beim paarweisen Vergleich zweier Aktionen wird anhand der in den Abschnitten 5.2 und 5.3 gewonnenen Erkenntnisse überprüft, ob diese Aktionen vollständig korreferent sind oder ob sie konzeptionell gleich sind. Das genaue Vorgehen des paarweisen Vergleichers wird im folgenden Abschnitt erläutert. Falls eine gewisse Gleichheit zwischen zwei Aktionen festgestellt wird, wird dies durch eine neue Kante im Graph gespeichert.

## 6.2. Paarweiser Vergleich der Aktionen

Im paarweisen Vergleich wird für jeweils zwei Aktionen  $a_1$ ,  $a_2$  mit  $instructionNumber(a_1) \leq instructionNumber(a_2)$  untersucht, inwiefern beide Aktionsnennungen die gleiche Identität beschreiben und konzeptionelle Gleichheit zwischen diesen vorliegt. Da der Agent auf natürlicher gesprochener Sprache arbeitet, die im Gegensatz zu formalen Sprachen mehrdeutig sein kann und eher Fehler enthält, kann im Allgemeinen nicht eindeutig gesagt werden, ob zwei Ausdrücke gleich sind. Es werden daher Wahrscheinlichkeitswerte zwischen 0,0 und 1,0 für die Ähnlichkeit zweier Aktionen angegeben. Zu Beginn wird allen Aktionspaaren eine Wahrscheinlichkeit von 1,0 für Aktionskorreferenz und konzeptionelle Gleichheit zugeordnet. Diese nimmt bei Indizien, die dafür sprechen, dass es sich um verschiedene Aktionen handelt, ab. Die genauen Werte, auf die die Wahl bei der Implementierung des Ansatzes fiel, sind in Anhang Abschnitt B zu finden.

Der Ablauf des paarweisen Vergleichs wird in Abbildung 6.2 illustriert.

### 6.2.1. Erkennung von konzeptioneller Gleichheit

Da nach Definition aus Abschnitt 5.1 konzeptionelle Gleichheit die Voraussetzung für Aktionskorreferenz ist, wird diese zuerst festgestellt. Wie in Abschnitt 5.2 beschrieben, müssen sich dafür Prädikate, Akteure, WHAT- und WHO- und HOW-Parameter der beiden Aktionen auf die jeweils selben Konzepte bzw. dieselben Entitäten beziehen. Wenn Informationen über Korreferenz zwischen Entitäten vorliegt, können diese Informationen direkt übernommen werden, um Korreferenz zwischen den genannten Objekten festzustellen.

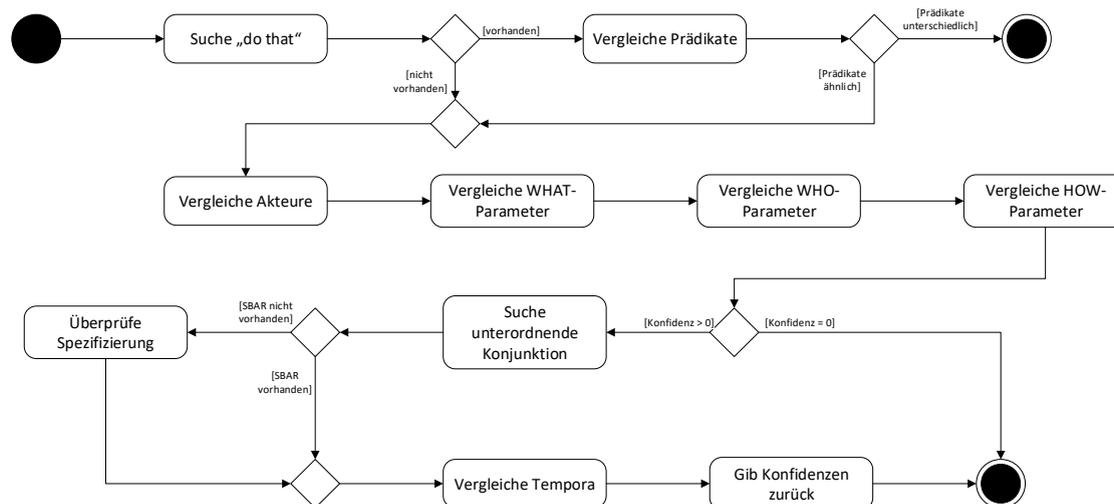


Abbildung 6.2.: Vergleich zweier Aktionen

### 6.2.1.1. Vergleich der Prädikate

Die Prädikatengleichheit ist die Grundvoraussetzung für konzeptionelle Gleichheit. Es muss zuerst überprüft werden, ob sich die Prädikate zweier Aktionen auf das gleiche Konzept beziehen. Dies kann, wie in Abschnitt 5.2.1 beschrieben, dann der Fall sein, wenn die Lemmata der Prädikate identisch sind, diese Synonyme sind, oder ein Prädikat ein Troponym des anderen Prädikates ist. Wenn die Lemmata gleich sind wird davon ausgegangen, dass diese das gleiche Konzept beschreiben. Da angenommen wurde, dass Sprecher für gleichartige Aktionen das gleiche Prädikat verwenden, wird etwas von den Wahrscheinlichkeitswerten abgezogen, wenn die Prädikate verschieden sind. Es wird ein kleiner Betrag von den Wahrscheinlichkeitswerten eines Aktionspaars abgezogen, wenn die Prädikate synonym sind. Bei einer Troponymbeziehung wird unterschieden, ob das Troponym in der erst- oder zweitgenannten Aktion verwendet wird. Da angenommen wurde, dass in der Regel zuerst der speziellere Begriff genannt wird, wird mehr abgezogen, wenn  $a_2$ , und weniger wenn  $a_1$  das Troponym ist. Falls keine semantische Ähnlichkeit zwischen den Prädikaten besteht, wird 1,0 abgezogen. Die Wahrscheinlichkeit, dass zwischen den Aktionen Gleichheit herrscht ist dann 0,0 und das Paar wird nicht weiter betrachtet.

Eine Ausnahme bilden die Wörter *do* und *happen*, sowie deren Synonyme. Weil sich diese Wörter in Verbindung mit einem Demonstrativpronomen auf beliebige Aktionen beziehen können, wird bei Auftreten dieser Wörter vorerst nichts abgezogen.

### 6.2.1.2. Vergleich der Akteure

Bei Aktionspaaren, die einen Wahrscheinlichkeitswert von über 0,0 haben werden nun die Akteure und Parameter verglichen, da zwei Aktionen vom selben Akteur ausgeführt werden müssen, damit konzeptionelle Gleichheit herrschen kann. Es wird zuerst überprüft, ob Akteure in beiden Aktionen vorhanden sind. Der Aktionserkennungsteil teilt Aktionen den Akteur aus der vorangegangenen Aktion zu, falls eine Aktion keinen Akteur besitzt, da das Prädikat entweder im Imperativ oder im Passiv vorliegt. Eine Aktion hat nur dann keinen Akteur, wenn alle vorangegangenen Aktionen ebenfalls keinen Akteur haben. Falls eine oder beide Aktionen keinen Akteur haben, wird ein kleiner Betrag von den Wahrscheinlichkeitswerten abgezogen, da es möglich ist, dass die Aktionen, bei denen kein Akteur genannt wird, von verschiedenen Akteuren ausgeführt werden. Anschließend wird die Korreferenzanalyse [Hey16] durchgeführt. Die Korreferenzanalyse berechnet selbst Wahr-

scheinlichkeitswerte für die Korreferenz zwischen Nominalphrasen. Dieser Wert kann übernommen werden und mit einem Gewicht von 0,5 in die hiesigen Werte einfließen. Falls die Korreferenzanalyse einen Wert von 0,0 ausgibt handelt es sich um verschiedene Akteure. Da auf gesprochener Sprache gearbeitet wird und diese Fehler enthält wird nicht sofort 1,0 abgezogen, wenn die Akteure verschieden sind. Falls die Informationen über Korreferenz zwischen den Akteuren nicht vorliegen und es sich bei den Akteuren um Substantive (gegebenenfalls mit Adjektiven) handelt, werden zuerst die Lemmata der Substantive verglichen. Es wird untersucht, ob die Lemmata identisch oder synonym sind und ob eine Hypo-/Hyperonymbeziehung zwischen ihnen besteht. Bei identischen Lemmata wird davon ausgegangen, dass die Akteure gleich sind. In diesem Fall wird nichts von den Wahrscheinlichkeitswerten abgezogen. Bei Synonymen wird wie bereits bei den Prädikaten ein kleiner Teil von den Wahrscheinlichkeitswerten abgezogen. Bei einer Hypo-/Hyperonymbeziehung zwischen den Substantiven wird auch wieder ein kleiner Betrag, falls zuerst das Hyponym und ein größerer Betrag falls zuerst das Hypernym genannt wird, abgezogen. Typischerweise werden Akteure durch Eigennamen beschrieben, welche keine Synonyme oder Hypo- bzw. Hypernyme haben. Da Korreferenz zwischen Eigennamen und Sammelbezeichnungen bestehen kann, wie z.B. zwischen *Armar* und *the robot* und diese ohne Korreferenzinformationen nicht aufgelöst werden kann, wird ein kleiner Betrag abgezogen, falls genau ein Akteur ein Eigenname und der andere eine Sammelbezeichnung mit bestimmtem Artikel ist, da es in diesem Fall möglich ist, dass sich die beiden Bezeichnungen auf denselben Akteur beziehen. Akteure werden oft durch Personalpronomen beschrieben. Falls die Akteure gleichartige Personalpronomen sind oder ein Akteur ein Substantiv und das andere ein Personalpronomen ist, wird ebenfalls nur ein kleiner Betrag abgezogen, da die Möglichkeit besteht, dass es sich um denselben Akteur handelt. Wenn beide Akteure durch unterschiedliche Personalpronomen realisiert sind (z.B. *he* und *she*), handelt es sich um verschiedene Akteure. Es wird dann 1,0 abgezogen.

### 6.2.1.3. Vergleich der Objekte

Als nächstes werden die direkten und indirekten Objekte der beiden Aktionen verglichen. Wie in Abschnitt 5.2.3 beschrieben, müssen auch diese sich jeweils auf dieselben Entitäten beziehen. Der Aktionserkennung unterscheidet zwischen WHAT- und WHO-Parametern. Parameter mit der Rolle WHO sind Eigennamen von Personen und Personalpronomen, außer *it* [Ou16]. Alle anderen Objekte erhalten vom Aktionserkennung die Rolle WHAT. Bei Objekten mit der Rolle WHAT werden wie auch beim Vergleich der Akteure die Lemmata der Substantive betrachtet. Im Gegensatz zu Akteuren, können Objekte auch bei Verben im aktiven Indikativ fehlen. Da typischerweise redundante Objekte eher in der zweiten Nennung einer Aktion ausgelassen werden, wird ein kleiner Betrag von den Wahrscheinlichkeitswerten abgezogen, wenn das jeweilige Objekt in *a2* und ein etwas größerer Betrag, wenn es in *a1* fehlt. Falls die jeweiligen Objekte in beiden Aktionen vorhanden sind werden sie miteinander verglichen. Es können hier ebenfalls die Korreferenzinformationen verwendet werden. Vor der Betrachtung der Korreferenzinformationen über diese Parameter werden jedoch bei den WHAT-Parametern die Artikel der Phrasen extrahiert. Dieselbe Aktion kann nämlich mit zwei Nennungen von unbestimmten Objekten beschrieben werden, wie in Abschnitt 5.2.3 festgestellt worden ist. Zwischen diesen Objekten liegt jedoch keine Korreferenz vor, die der Korreferenzauflöser erkennt. Aus diesem Grund werden in dem Fall, in dem beide WHAT-Parameter unbestimmte Artikel haben, die Korreferenzaktionen nicht betrachtet.

WHAT-Parameter können aus Artikeln, Adjektiven und Substantiven bestehen. Alternativ können sie aus dem Personalpronomen *it* bestehen. Es werden die Zeichenketten betrachtet, die entsteht, wenn die Wörter ohne Artikel in den Phrasen lemmatisiert werden. Falls beide Objekte durch die gleichen Wörter beschrieben werden, wird nichts vom Wahrscheinlichkeitswert abgezogen. Die Zeichenkette eines lemmatisierten Ausdrucks ist im anderen

enthalten, wenn ein Ausdruck keine oder weniger Adjektive beinhaltet. Da angenommen wurde, dass eher die erste Nennung mehr spezifiziert wird, wird ein geringer Betrag abgezogen, wenn die zweite Zeichenkette in der ersten Zeichenkette echt enthalten ist und ein größerer Betrag, wenn die erste in der zweiten Objekt enthalten ist. Wenn beide Objekte nur durch ein Substantiv beschrieben werden und diese synonym sind, wird ebenfalls ein kleiner Betrag abgezogen, da angenommen wurde, dass ein Mensch denselben Gegenstand mit den gleichen Wörtern beschreibt. Wenn eine Hyperonymbeziehung vorliegt, wird ein etwas größerer Betrag abgezogen, falls das erste Objekt und ein noch größerer Betrag falls das zweite Objekt das Hyponym ist, da angenommen, dass eher der spezifischere Begriff zuerst verwendet wird und Hyponyme mehr Informationen enthalten. Objekte mit verschiedenen Adjektiven können sich auf dieselbe Entität beziehen, sofern die Adjektive verschiedenen Attributen zuzuordnen sind. Beispielsweise kann es sich bei einer roten Tasse und einer großen Tasse um dieselbe Tasse handeln. Bei einer großen und einer kleinen Tasse jedoch nicht. Daher werden bei unterschiedlichen Adjektiven nur die Teile der Phrasen, die Substantive sind betrachtet. Weil angenommen wird, dass dieser Fall nicht oft eintritt, wird ein großer Betrag abgezogen, wenn die Substantive gleich sind, und noch größere Beträge bei Synonym-, Hyponym- und Hyponymbeziehungen. Bei verschiedenen Substantiven wird 0,5 abgezogen. Es kann sich zwar nicht um konzeptionell gleiche Aktionen handeln, wenn die Parameter sich auf verschiedene Entitäten beziehen, jedoch ist es möglich, dass Fehler in der Eingabe vorliegen oder semantische Beziehungen nicht korrekt gefunden werden.

Wenn keine zwei unbestimmten Artikel vorliegen und die Ergebnisse des Korreferenzauflösers verfügbar sind, wird der vom Korreferenzauflöser berechnete Wahrscheinlichkeitswert mit Gewicht 0,5 übernommen. Andernfalls müssen die Objekte vom hier entwickelten Werkzeug auf Korreferenz überprüft werden. Wenn keines der WHAT-Objekte ein Pronomen ist, werden die Objekte, wie in dem Fall, in dem beide unbestimmte Artikel haben, verglichen. Wenn das erste WHAT-Objekt einen unbestimmten Artikel und das zweite WHAT-Objekt einen bestimmten Artikel besitzt, wird zusätzlich etwas von den Wahrscheinlichkeitswerten abgezogen, weil sich eine neu eingeführte Entität nicht auf eine bereits eingeführte Entität beziehen kann, wie in Abschnitt 5.2.3 erläutert wurde.

WHO-Objekte, die Personennamen sind haben keinen Artikel und keine näher beschreibenden Adjektive. Hier wird lediglich überprüft, ob beide Eigennamen gleich sind. Falls Objekte durch Personalpronomen beschrieben werden, wird wie bei dem Vergleich der Akteure verfahren.

#### 6.2.1.4. Vergleich der sonstigen Parameter

Als letztes werden die Umstandsparameter (siehe Abschnitt 5.2.4.3) der beiden Aktionen, welche die Rolle HOW haben und durch Adverbien gebildet werden, verglichen, sofern sie vorhanden sind. Da diese Parameter eher in der ersten Nennung einer Aktion verwendet werden wird nur ein geringer Betrag abgezogen, wenn sie dort vorhanden sind, aber in der zweiten Nennung fehlen. Wenn sie nur in der zweiten Nennung vorhanden sind wird ein größerer Betrag vom Wahrscheinlichkeitswert für konzeptionelle Gleichheit zwischen den Aktionen abgezogen, da Adverbien die Aktionen ebenfalls genauer beschreiben und angenommen wird, dass die erste Aktionsnennung in der Regel spezifischer aufgeführt wird. Falls beide Adverbien die gleichen Wörter sind, wird nichts abgezogen. Falls die Adverbien Synonyme sind, wird abgezogen. Es müssen nur zwei Wörter verglichen werden, da eine Aktion höchstens ein Adverb enthalten kann, da Befehle wie „Open the fridge slowly and carefully“ vom Aktionserkennung als zwei Aktionen erkannt wird. In diesem Fall sind beide Aktionen gleich, obwohl sich die Adverbien unterscheiden. Wenn die beiden Aktionen verschiedene Adverbien haben und sich nicht nur durch diese unterscheiden, wird ein größerer Betrag abgezogen.

Tabelle 6.1.: Verwendbare Begriffe anstelle von *do*, *happen* und *that*

Wort	Alternativen
do	do, make, perform, execute, complete, finish
happen	happen, hap, go on, pass off, occur, pass, fall out, come about, take place
that	that, this, it, the previous action, the last action, so

### 6.2.1.5. Spezialfälle

Falls sich zwei Aktionen ein Prädikat teilen und sich in mindestens einem Parameter oder dem Akteur unterscheiden wird der Wahrscheinlichkeitswert auf 0,0 gesetzt. In diesem Fall beschreiben die Aktionen verschiedene Konzepte. Beispielsweise unterscheiden sich die Aktionen „Armar bring me an apple“ und „Armar bring me an orange“ in „Armar bring me an apple and an orange“ nur im WHAT-Parameter und sind verschieden.

Wenn das Prädikat *do* oder *happen* ist, und der Akteur oder ein Objekt *that* oder ein Synonym ist, handelt es sich um eine Aktion, die mit der vorhergehenden Aktion konzeptionell gleich ist. In diesem Fall wird der Wahrscheinlichkeitswert für konzeptionelle Gleichheit auf 1,0 gesetzt, wenn sich die Befehlsnummern der Aktionen um eins unterscheiden und „*do that*“ in der zweiten Aktion auftritt. Lediglich die Akteure werden verglichen, da diese verschieden sein können. In Tabelle 6.1 sind die verwendbaren Alternativen für *do*, *happen* und *that* aufgelistet.

Falls sich die Befehlsnummer zweier Aktionen um mehr als eins unterscheiden, und beide Aktionen durch die Verwendung *do* oder *happen* und *that* gebildet werden, handelt es sich, obwohl die gleichen Wörter verwendet werden, im Allgemeinen nicht um dieselbe Aktion. Falls beide Aktionen eine solche Formulierung beinhalten, werden die Wahrscheinlichkeitswerte auf 0,0 gesetzt.

### 6.2.2. Erkennung von Aktionsidentität

Falls der nun errechnete Wahrscheinlichkeitswert für die konzeptionelle Gleichheit der Aktionen größer als 0 ist, wird mit der Erkennung der Aktionsidentität fortgefahren. Um nun festzustellen, ob sich beide Aktionsnennungen auf dieselbe Aktionsidentität beziehen, muss der Ort und die Zeit der Nennungen verglichen werden, falls diese Angaben in den Nennungen auftreten. Weiterhin muss überprüft werden, ob Anzeichen für einen der Fälle, in denen eine Aktion wiederholt genannt wird, die in Abschnitt 5.3 beschrieben werden, vorliegt. Bei der Erkennung von echter Aktionskorreferenz wird hier ebenfalls einen Wahrscheinlichkeitswert zwischen 0,0 und 1,0 berechnet. Da nach Definition konzeptionelle Gleichheit eine Voraussetzung für Aktionskorreferenz ist, ist die Wahrscheinlichkeit, dass zwei Aktionen korreferent sind höchstens so hoch wie die Wahrscheinlichkeit, dass sie konzeptionell gleich sind. Bei der Berechnung dieses Wahrscheinlichkeitswertes wird deshalb vom Wert für konzeptionelle Gleichheit ausgegangen.

Es wird zunächst nach Kriterien gesucht, die dafür sprechen, dass sich beide Nennungen auf verschiedene Aktionsidentitäten beziehen. Dies sind zunächst unterschiedliche Nennungen von Orts- und Zeitangaben in den Aktionsnennungen, also Parameter vom Typ **WHERE** und **WHEN**. Da verschiedene sprachliche Konstrukte dieselbe Zeit und denselben Ort beschreiben können, wird höchstens ein kleiner Betrag abgezogen, wenn in beiden Aktionen Ort- oder Zeitangaben vorkommen und sich diese unterscheiden. Wenn die Parameter nur in der ersten Aktion genannt werden, wird nichts abgezogen, da angenommen wurde, dass sie dort typischerweise eher vorkommen. Wenn die Parameter nur in der zweiten Aktion genannt werden, wird hingegen ein kleiner Betrag abgezogen.

Anschließend wird nach Schlüsselphrasen in der zweiten Aktionsnennung gesucht, die darauf hinweisen, dass es sich bei dieser Nennung um eine wiederholte Ausführung der ersten Aktion handelt. Diese Phrasen werden in Abschnitt 5.3.5 beschrieben. Falls sich eine dieser Phrasen in der Umgebung der zweiten Aktionsnennung befindet, handelt es sich um eine neue Aktion. Die Umgebung bilden dabei alle Wörter zwischen dem ersten und letzten Wort der Aktion, sowie jeweils die zwei Wörter direkt vor und nach der Aktionsnennung. Mit dieser erweiterten Suche wirken sich einzelne Verzögerungslaute direkt vor oder hinter den Wörtern, die die Aktion bilden, nicht auf das Auffinden dieser Schlüsselphrasen aus. Wenn eine Schlüsselphrase gefunden wird, wird der Wahrscheinlichkeitswert für Aktionskorreferenz auf 0,0 gesetzt da es sich dann um eine wiederholte Ausführung handelt. Sofern Aktionskorreferenz bis hierher noch nicht ausgeschlossen worden ist, werden nun die Fälle, in denen diese auftreten kann getrennt voneinander untersucht und der Fall mit dem höchsten Wahrscheinlichkeitswert verwendet.

#### 6.2.2.1. Temporale, lokale und kausale Beziehungen

Wenn eine Aktion mehrmals genannt wird, um temporale, lokale oder kausale Beziehung herzustellen, wie in Abschnitt 5.3.1 beschrieben, steht eine Nennung in einem Nebensatz. Es kann sowohl die erste als auch die zweite Aktion im Nebensatz stehen. Es werden beide Fälle untersucht und für beide Fälle wird ein Wahrscheinlichkeitswert berechnet. Mit dem größeren Wert wird weitergearbeitet. Der Nebensatz wird wie in Abschnitt 5.3.1.1 beschrieben von einer unterordnenden Konjunktion eingeleitet. Diese befindet sich in geschriebenen Sätzen direkt vor dem Subjekt. Da in gesprochener Sprache Verzögerungslaute auftreten können, werden die zwei Wörter vor dem Subjekt nach Wörtern abgesucht, die zu einer Phrase vom Typ „unterordnenden Konjunktion“ gehören. Falls keine unterordnende Konjunktion vor der Aktionsnennung gefunden wird, handelt es sich nicht um einen Nebensatz und ein großer Betrag wird vom Wahrscheinlichkeitswert ab, da es sich dann vermutlich nicht um eine Verwendung einer Bereits genannten Aktion zur Erzeugung von Kohärenz handelt. Anschließend werden die Zeitformen der Prädikate verglichen. Die Zeitform wird an der Wortart des Prädikats erkannt. Aktionen, die zum Zeitpunkt der Nennung bereits ausgeführt worden sind, werden in der Regel durch Verben in einer Vergangenheitsform ausgedrückt. Es wird deshalb mehr abgezogen, wenn dies nicht der Fall ist. Die Werte, die vom Wahrscheinlichkeitswert abgezogen werden, werden ebenfalls im Anhang Abschnitt B aufgelistet.

Danach müsste überprüft werden, ob der Nebensatz zum Hauptsatz gehört. Ist dies der Fall, kann keine Aktionskorreferenz zwischen den beiden Nennungen herrschen, wie in Abschnitt 5.3.1.3 erläutert wurde. Da bei dem hier erläuterten Ansatz aber keine Satzzeichen zur Verfügung stehen und diese nicht festgestellt werden können, wird lediglich ein sehr geringer Betrag abgezogen, falls beide Aktionsnennungen direkt aufeinander folgen, da es dann möglich ist, dass die beiden Aktionen in einem Satz vorliegen. Es kann nur dann sicher gesagt werden, dass ein Nebensatz zu einem Hauptsatz gehört, wenn er der erste oder letzte Satz im Diskurs ist. In diesem Fall wird ein größerer Betrag abgezogen.

Des Weiteren wurde in Abschnitt 5.3.1.4 erläutert, dass zwei Aktionsnennungen, von denen eine in der Folgeanweisung und die andere in der Alternativanweisung eines Bedingungssatzes auftreten nicht korreferent sein können, wenn sie jeweils die einzigen Aktionen in ihrem Teil sind. Falls dieser Fall beim Vergleich der Aktionen auftritt wird deshalb ein großer Betrag vom Wahrscheinlichkeitswert für Aktionskorreferenz abgezogen.

#### 6.2.2.2. Spezifizierungen

Wenn eine Aktion wiederholt genannt wird, um einer anderen Aktion mehr Informationen hinzuzufügen, geschieht dies wie in Abschnitt 5.3.3 beschrieben direkt nach dieser Aktion.

Die Befehlsnummer der zweiten Aktion muss also um eins größer sein als die Befehlsnummer der ersten Aktion und die zweite Aktion muss mindestens eine Information beinhalten, die die erste Aktion nicht enthält. Wenn die Aktionen durch eine Konjunktion oder durch eine Präposition wie „after“ voneinander getrennt sind liegt keine Spezifizierung vor. Es gibt auch bestimmte Wörter, wie „next“, die aussagen, dass es sich um eine neue Aktion handelt. Des Weiteren muss der Tempus der beiden Prädikate übereinstimmen. Die zweite Aktion enthält zusätzliche Informationen, wenn sie mindestens einen Parameter enthält, den die erste Aktion nicht enthält oder wenn ein Parameter in der zweiten Aktion spezifischer ist als in der ersten Aktion. Es wird für **WHAT**-, **WHO**-, **HOW**-, **WHEN**- und **WHERE**-Parameter überprüft, ob sie nur in der ersten Aktion leer sind. Bei **WHAT**-Parametern zusätzlich untersucht, ob das Substantiv im zweiten Parameter ein Hyponym des Substantivs im ersten Parameter ist, wenn diese Parameter aus Substantiven bestehen, und ob der zweite Parameter Adjektive enthält, die im ersten Parameter nicht vorkommen. Bei den Prädikaten wird überprüft, ob das zweite Prädikat ein Troponym des ersten Prädikats ist. Es wird also untersucht, ob die zweite Aktionsnennung mindestens eine neue Information einführt, ob die Tempora übereinstimmen und ob sich zwischen den Aktionen kein Wort befindet, das explizit sagt, dass es sich bei der zweiten Aktion um eine neue Aktion handelt. Ist dies der Fall, wird davon ausgegangen, dass zwischen den beiden Aktionsnennungen Korreferenz besteht. Die Wahrscheinlichkeit wird dann auf 1,0 gesetzt, andernfalls auf 0,0.

### 6.3. Implementierungsdetails

Die Implementierung des Agenten erfolgte als Maven-Projekt in der objektorientierten Programmiersprache Java. In Abbildung 6.3 ist das Zusammenspiel der Klassen des Agenten dargestellt. Ein vollständiges Klassendiagramm ist in Anhang Abschnitt C zu finden. In diesem Abschnitt wird zunächst die Eingabe des Agenten beschrieben. Anschließend werden die einzelnen Klassen im Klassendiagramm kurz vorgestellt.

#### 6.3.1. Eingabe

Die Eingabe des Agenten ist ein Graph, der als gemeinsame Datenstruktur für alle Agenten in PARSE dient. Ein Knoten hat einen Typ `type` und beliebig viele Attribute. Eine Kante hat einen Start- und Endknoten, einen Typ `type` und beliebig viele Attribute. Die Knoten mit Typ `token` des Graphen repräsentieren die einzelnen Wörter (Tokens) der Eingabe. Der Graph ist eine Implementierung der Schnittstelle `IGraph`, Knoten Implementierungen von `Inode` und Kanten Implementierungen von `IArc`. Um den hier zu entwickelnden Agenten benutzen zu können, muss mindestens die seichte Sprachverarbeitung (SNLP) und der Aktionserkennung zuvor ausgeführt werden. Die Informationen des Korreferenzauflösers und Bedingungsanalytikers können die Qualität der Ausgabe des Agenten verbessern.

#### SNLP

In einem Knoten vom Typ `token` sind nach Ausführung von SNLP folgende, für unsere Implementierung relevante, Informationen gespeichert:

- `value`: Das Wort als Zeichenkette.
- `pos`: Die Markierung Wortart des Wortes aus dem Penn-Treebank-Tagset (Abschnitt 2.2.1).
- `chunkIOB`: Die Phrase zu der das Wort gehört im IOB-Format (Abschnitt 2.2.2).
- `instructionNumber`: Die Nummer des Befehls, zu dem das Wort gehört, beginnend bei 0. Ein Befehl beginnt bei einer Verbalphrase und endet vor der nächsten. Wörter wie *and* spalten einen Befehl in mehrere Befehle auf.
- `position`: Die Stelle an der das Wort in der Eingabe steht, beginnend bei 0.

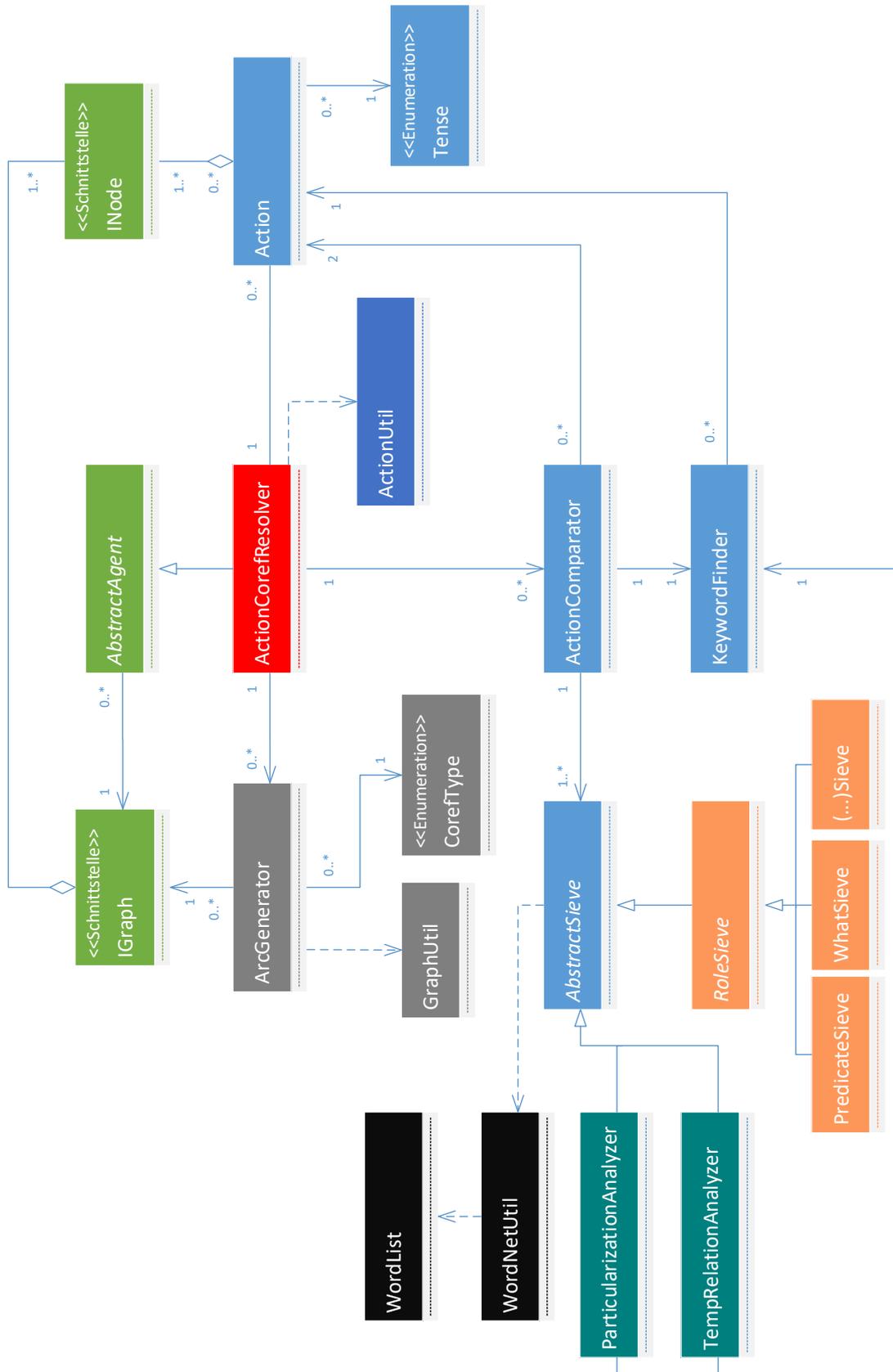


Abbildung 6.3.: Vereinfachtes Klassendiagramm des Agenten

- **lemma**: Das Lemma des Wortes (Abschnitt 2.1.5).
- **stem**: Der Wortstamm des Wortes.

Die Tokens werden mit Kanten vom Typ `relation` so verbunden, dass jedes Token mit dem in der Eingabe darauffolgenden Token verbunden ist. Diese Kanten haben das Attribut `value` mit dem Wert `NEXT`.

### Aktionserkenner

Der Aktionserkenner fügt den Knoten vom Typ `token` das Attribut `role` hinzu, welches angibt, welche Rolle eine Phrase innerhalb einer Aktion einnimmt. Mögliche Werte von `role` sind `ACTOR`, `PREDICATE`, `WHAT`, `WHEN`, `WHO`, `WHERE`, `WHY` und `HOW`. Das erste Token einer Phrase ist mit einer Kante vom Typ `relationInAction` mit Attribut `type: PREDICATE_TO_PARA` mit dem Prädikat verbunden. Die einzelnen Tokens der Phrasen, die eine Rolle in einer Aktion einnehmen sind mit Kanten vom Typ `relationInAction` mit Attribut `type: INSIDE_CHUNK` miteinander verbunden. Aufeinanderfolgende Aktionen sind verbunden, indem ihre Prädikate Kanten vom Typ `relationInAction` mit Attribut `type: NEXT_ACTION` miteinander verbunden sind. Falls ein Prädikat zu mehreren Aktionen gehört, wird ein neuer Knoten vom Typ `newPredicate` erstellt. Dieser hat das Attribut `value` mit einer Kopie des Wortes der Aktion.

### Korreferenzauflöser

Der Korreferenzauflöser führt für jede Entität (Substantive und Personal- und Reflexivpronomen) im Diskurs einen neuen Knoten vom Typ `contextEntity` ein, der mit einer Kante vom Typ `reference` auf das beschreibende Token verweist. Entitäten zwischen denen Korreferenz (Abschnitt 2.1.6) besteht werden mit einer Kante vom Typ `contextRelation`, deren Attribut `name` den Wert `anaphoraReferent` bzw. `objectIdentityReferent` besitzt, je nachdem, ob es sich bei der Referenz um ein Pronomen oder ein Substantiv handelt. In der Kante ist auch eine Konfidenz zwischen 0,0 und 1,0 angegeben.

### Bedingungsanalyisierers

Der Bedingungsanalyisierers fügt jedem Token das Attribut `commandType` hinzu, das angibt, ob es zu einem Konditionalsatz gehört und ob es in der Bedingung, der Folgeanweisung oder der Alternativanweisung steht.

### Beispiel

In Abbildung 6.4 sind die, für diese Arbeit relevanten, Teile der Ausgabe der anderen Agenten für den Satz „Armar take the cup fill it“ dargestellt. Die Teile, die SNLP liefert sind schwarz dargestellt. Die Teile des Aktionserkenners sind orange, die des Korreferenzauflöser blau und die des Bedingungsanalyisierers grün.

#### 6.3.2. Hauptklasse: `ActionCorefResolver`

Die Klasse `ActionCorefResolver` bildet den Startpunkt des Agenten. Sie erbt, wie alle `PARSE-Agenten`, von der abstrakten Klasse `AbstractAgent` und beinhaltet die Realisierung der abstrakten Methode `exec()`, welche zur Ausführung des Agenten aufzurufen ist. Um die Methode aufrufen zu können, muss zuvor der Graph mithilfe von `setGraph(IGraph graph)` an den Agenten übergeben werden. Bei Aufruf von `exec()` werden zunächst mit statischen Methoden der Klasse `ArcGenerator` alle bereits vorhandenen `actionCoref`-Kanten im Graph gelöscht, um idempotentes Verhalten zu gewährleisten. Der Kantentyp `actionCoref` wird neu angelegt, falls dieser noch nicht existiert. Anschließend werden die

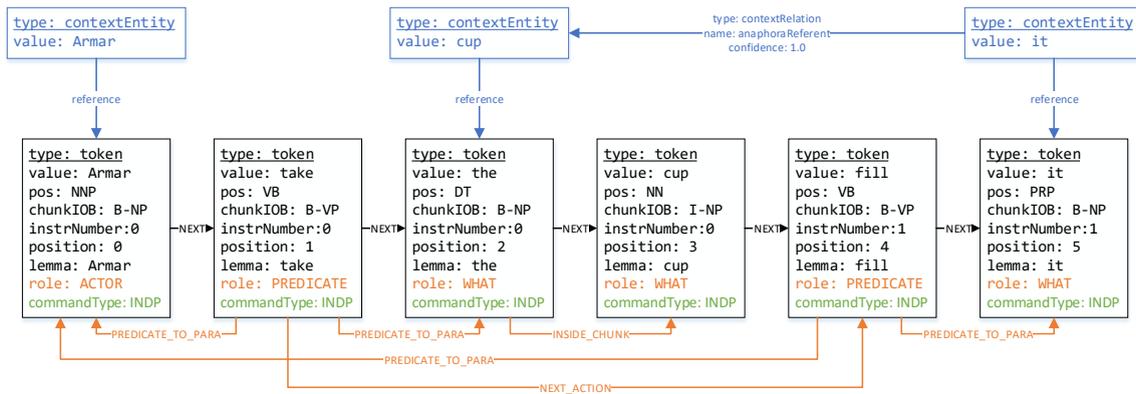


Abbildung 6.4.: Beispielhafte Eingabe des Agenten (nur relevante Teile)

Aktionsnennungen mit der statischen Methode `readActionsFromGraph(IGraph graph)` in `ActionUtil` aus dem Graph extrahiert. Für jedes Paar von extrahierten Aktionsnennungen wird eine neue Instanz der Klasse `ActionComparator` angelegt und die Nennungen werden mit der `compareActions()`-Methode auf konzeptionelle Gleichheit und Aktionskorreferenz untersucht. Wenn konzeptionelle Gleichheit oder Aktionskorreferenz zwischen zwei Aktionsnennungen gefunden wird, wird für sie mithilfe eines `ArcGenerators` eine neue `actionCoref`-Kante dem Graph hinzugefügt. Nachdem alle Aktionspaare untersucht worden sind, werden transitive Kanten hinzugefügt.

### 6.3.3. Repräsentation von Aktionsnennungen

Aktionsnennungen werden durch die Klasse `Action` repräsentiert. Die Klasse enthält die Knotenlisten `predicate`, `actor`, `what`, `who`, `when`, `where` und `how`. Diese Listen enthalten alle Knoten vom Typ `token`, die die Wörter, die im jeweiligen Aktionsparameter enthalten sind, beinhalten. Zusätzlich enthält die Klasse eine Liste `tokens` mit allen Knoten der Aktionsnennung, sortiert nach Auftreten in der Eingabe. Ein weiteres Attribut ist der Inode `predToken`, welcher den Knoten repräsentiert, zu bzw. von dem `actionCoref`-Kanten führen oder ausgehen, wenn konzeptionelle Gleichheit oder Aktionskorreferenz festgestellt wird. Dieser Knoten ist in der Regel der Kopf des Prädikats. Bei Prädikaten, die zu einer zweiten Aktion gehören, ist `predToken` der Knoten vom Typ `newPredicate`, der die Kopie des Prädikats enthält. Die Klasse `Action` enthält auch die Methode `getTense`, die das Tempus des Prädikats anhand seiner Wortart und Modalwörter bestimmt.

### 6.3.4. Vergleich zweier Aktionen

Ein `ActionComparator` nimmt zwei Aktionsnennungen entgegen und berechnet bei Aufruf der Methode `compareActions()` Konfidenzen für konzeptionelle Gleichheit und Aktionskorreferenz. Zur Berechnung der Konfidenzen werden verschiedene Filter verwendet, die das Vorgehen aus Abschnitt 6.2 umsetzen und eine Instanz der Klasse `KeywordFinder`. Die Strafwerte, die verwendet werden, sind im Anhang Abschnitt B aufgelistet. Der `KeywordFinder` hat die zweite Aktionsnennung als Attribut und stellt Methoden bereit, um in ihr und ihrer Umgebung verschiedene Schlüsselphrasen zu finden. Er sucht beispielsweise nach unterordnenden Konjunktionen und Wiederholungswörtern wie *again*.

Alle Filter erben von der abstrakten Klasse `AbstractSieve`, welche die abstrakte Methode `compare()` beinhaltet, die zwei Aspekte der Aktionen vergleicht und einen Strafwert zurückgibt. Die Filter sind unterteilt in `RoleSieves`, welche jeweils zwei Teile der Aktionsnennungen vergleichen, die eine bestimmte Rolle haben, und die Filter `ParticularizationAnalyzer` und `TempRelationAnalyzer` welche nach sonstigen Indizien für

Aktionskorreferenz suchen. Es existieren die Rollenfilter `PredicateSieve`, `ActorSieve`, `WhatSieve`, `WhoSieve`, `HowSieve`, `WhenSieve` und `WhereSieve`. Diese überprüfen, ob die jeweiligen Parameter in beiden Aktionen vorhanden sind und ob sie gleich sind oder in einer semantischen Beziehung zueinander stehen. Die Filter `ActorSieve`, `WhatSieve` und `WhoSieve` verwenden die Ergebnisse des Korreferenzauflösers, um zu überprüfen, ob es sich bei den Parametern um identische Entitäten handelt. Der `TempRelationAnalyzer` überprüft, ob die Aktion wiederholt worden sein könnte, um beispielsweise eine temporale Beziehung herzustellen (siehe Abschnitt 6.2.2.1). Dafür sucht er mithilfe des `KeywordFinders` nach unterordnenden Konjunktionen und vergleicht die Zeitformen. Der `ParticularizationAnalyzer` überprüft, ob die zweite Aktionsnennung eine Spezifizierung der Ersten ist (Abschnitt 6.2.2.2, indem er feststellt, ob mindestens ein Parameter in der zweiten Nennung spezifischer ist, als in der ersten Nennung).

### 6.3.5. Erkennung semantischer Relationen

Die Klasse `WordNetUtil` stellt statische Methoden zur Suchen nach semantischen Relationen in WordNet (Abschnitt 2.3) bereit und wird von den Filtern verwendet. Um auf WordNet zuzugreifen, wird die API `extJWNL` (Extended Java WordNet Library) [ext] verwendet. `WordNetUtil` beinhaltet Methoden um Synonyme, Hyperonyme, Hyponyme und Troponyme für ein Wort mit Wortart zu finden und den Abstand zwischen einem Hyponym und seinem Hyperonym zu bestimmen. Gefundene Relationen werden in der Instanz des Einzelstücks `WordList` gespeichert, um WordNet-Anfragen bei mehrfacher Suche nach gleichen Relationen zu minimieren.

### 6.3.6. Darstellung der Ergebnisse

Konzeptionelle Gleichheit und Aktionsidentitäten werden durch Kanten zwischen den Köpfen der Prädikate im Graph gespeichert. Der Startknoten der Kanten ist immer das Prädikat der zweiten Nennung der Aktion und der Zielknoten das Prädikat der ersten Nennung. In den Kanten werden die Art der Gleichheit (`sameConcept`, `sameIdentity`) sowie der errechnete Wahrscheinlichkeitswert gespeichert. Falls sowohl konzeptionelle Gleichheit, als auch Aktionskorreferenz zwischen zwei Aktionen besteht, werden beide Beziehungen in eigenen Kanten gespeichert. Diese Darstellung wurde gewählt, da man sich dann nur eine Teilmenge der Kanten anschauen muss, wenn man sich nur für echte Aktionskorreferenz interessiert.

In Abbildung 6.5 ist der relevante Teil der Ausgabe der Eingabe „Armar bring me an apple juice open the door after you have brought me the juice“ dargestellt. Die Knoten beinhalten die Wörter, Wortarten, Phrasen im IOB-Format, Positionen in der Eingabe, Lemmata, und die vom Aktionserkennner zugeordneten Rollen. Die Aktionen „Armar bring me an apple juice“ und „you have brought me the juice“ beziehen sich auf dieselbe Aktionsidentität und somit auch auf dasselbe Konzept. Die Konfidenzen für konzeptionelle Gleichheit ist 1,0, da Prädikate übereinstimmen und sich die jeweiligen Akteure und Objekte auf dieselbe Entität beziehen. Die Konfidenz für Aktionskorreferenz ist ebenfalls 1,0, weil durch die unterordnende Konjunktion „after“ erkannt wird, dass sich die Aktion in einem Adverbialsatz befindet.

Nachdem die Kanten mit den im paarweisen Vergleich errechneten Konfidenzen eingefügt worden sind, werden nun auch transitive Kanten hinzugefügt. Dies ist sinnvoll, da bei einer Aktion  $a_1$ , die durch Strukturen wie „do that“ repräsentiert wird, nur Aktionskorreferenz mit der direkt vorangehenden Aktion  $a_0$  gefunden wird. Falls nun eine dritte Aktionsnennung  $a_2$  mit der Ersten korreferent ist, wird zwar die Korreferenz zwischen  $a_2$  und  $a_0$ , nicht jedoch zwischen  $a_2$  und  $a_1$  erkannt. Transitive Kanten werden gebildet, indem Pfade, bestehend aus ungerichteten `sameConcept`- bzw. `sameIdentity`-Kanten, zwischen

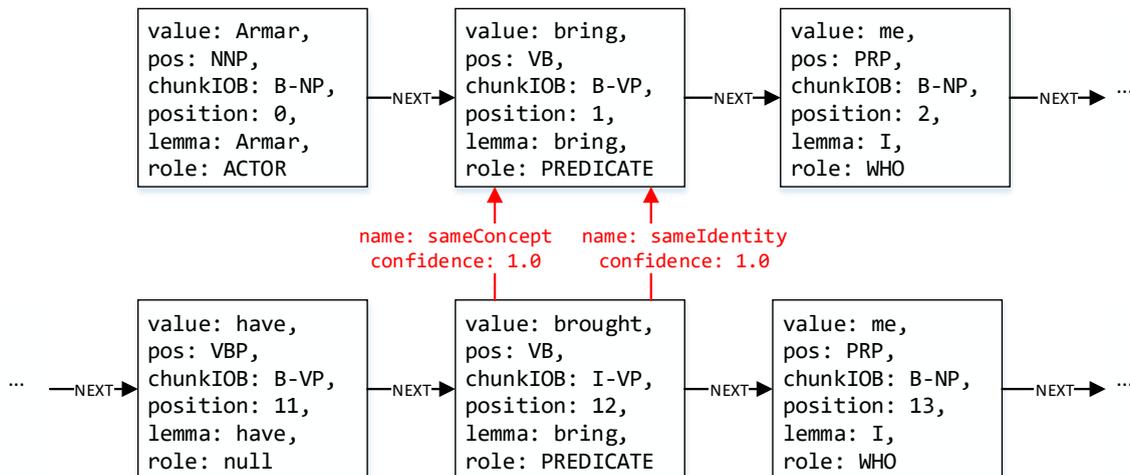


Abbildung 6.5.: Darstellung der Ergebnisse der Eingabe „Armar bring me an apple juice open the door after you have brought me the juice“.

zwei Prädikaten gesucht werden. Die Konfidenz eines Pfades wird durch Multiplizieren der Konfidenzen der Kanten auf dem Pfad berechnet. Falls ein Pfad mit positiver Konfidenz gefunden wird, wird für den Pfad mit der größten Konfidenz eine neue Kante eingeführt. In Abbildung 6.6 wird die konzeptionelle Gleichheit zwischen den Aktionsnennungen „do

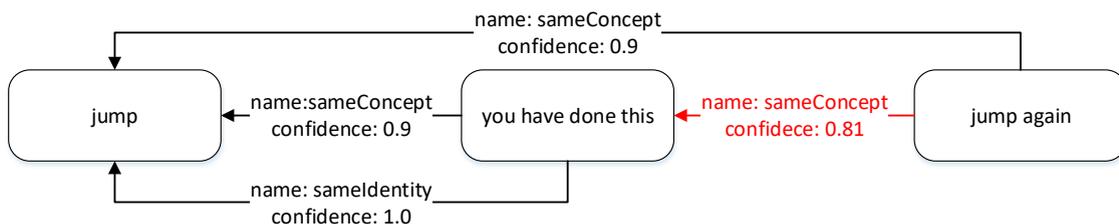


Abbildung 6.6.: Transitive Kanten (rot)

that“ und „jump again“ im Satz „jump after you have done this please jump again“ zunächst nicht erkannt. Da Kanten mit dem Typ `sameConcept` zwischen „jump again“ und „jump“, sowie zwischen „jump“ und „you have done that“ mit einer Konfidenz von jeweils 0,9 existieren, ergibt sich eine Konfidenz von  $0,9 * 0,9 = 0,81$  für konzeptionelle Gleichheit zwischen „do that“ und „jump again“. Es wird eine neue Kante zwischen diesen Aktionen hinzugefügt.

In der Implementierung wird diese Ergebnisspeicherung durch die Klasse `ArcGenerator` realisiert. Um dem Graphen eine neue Kante hinzuzufügen, muss eine neue Instanz der Klasse `ArcGenerator` angelegt werden. Der Konstruktor von `ArcGenerator` erhält die Referenten- und die Referenzaktion, sowie den Graph. Mithilfe der Methode `addArc(CorefType type, double confidence)` wird eine neue `actionCoref`-Kante mit den Attributen `name` und `confidence` von dem `predToken` der Referenzaktion zum `predToken` der Referentenaktion angelegt. Das Attribut `name` enthält dabei den Wert `sameConcept`, wenn konzeptionelle Gleichheit und `sameIdentity`, wenn Aktionskorreferenz herrscht.

Die Klasse `GraphUtil` enthält die statische Methode `calcTransitiveConf(...)`, die für zwei Knoten die transitive Konfidenz für konzeptionelle Gleichheit und Aktionskorreferenz berechnet. Die Methode verwendet eine modifizierte Version des Dijkstra-Algorithmus, der die negierten logarithmierten Konfidenzen als Kantengewichte auffasst und darauf kürzeste

Wege berechnet. Diese Methode wird ausgeführt, nachdem alle nicht-transitiven Kanten gesucht worden sind.

## 7. Evaluation

Das Ziel dieser Bachelorarbeit ist der Entwurf und die Implementierung eines Agenten zur Erkennung von konzeptioneller Gleichheit und Aktionskorreferenz wie sie in Kapitel 5 definiert werden. Der Agent wurde in Kapitel 6 implementiert. In diesem Kapitel wird nun untersucht, wie gut der entwickelte Agent konzeptionelle Gleichheit und Aktionskorreferenz erkennt. Dafür wird zuerst ein Evaluationskorpus erstellt und händisch mit Musterlösungen versehen. Die Ausgabe des Agenten wird mit den Musterlösungen verglichen. Dabei werden die Metriken Präzision, Ausbeute und  $F_1$ -Maß berechnet und diskutiert, wie der Agent verbessert werden kann. Um zu untersuchen, wie sich andere Agenten auf das Ergebnis des hiesigen Agenten auswirken, werden hier verschiedene Evaluationsvarianten betrachtet.

### 7.1. Evaluationskorpus

Zur Evaluation der Ergebnisse wird ein Korpus mit Sprachaufnahmen benötigt, in denen gleiche und ähnliche Aktionen mehrmals in einer Sprachaufnahme vorkommen. Das bereits existierende Korpus, der im Rahmen der Bachelorarbeit von Günes [G15] entstand, und durch diverse Projekte im Rahmen von PARSE erweitert wurde, enthält keine Eingaben, in denen Aktionen mehrmals genannt werden. Das bestehende Korpus muss deshalb um geeignete Aufnahmen erweitert werden. Es sind dafür Szenarien zu entwickeln, die dazu führen, dass die daraus resultierenden Sprachaufnahmen folgende Kriterien erfüllen:

- Die Aufnahmen enthalten mindestens eine Aktion, die wiederholt genannt wird.
- Die Aufnahmen enthalten Aktionsnennungen, die ähnlich sind, aber unterschiedliche Aktionen referenzieren.
- Die Sprache der Aufnahmen ist Englisch.
- Die Aktionen sollen nach Möglichkeit mit unterschiedlichen Wörtern beschrieben werden.

Die Wahl der Einsatzumgebung fiel auf eine Küche, in der der humanoide Roboter ARMAR-III verschiedene Aufgaben erledigen soll. Somit steht die Erweiterung des Korpus in Einklang mit dem bereits bestehenden Korpus.

Um Sprachaufnahmen zu sammeln, wurden zwei Szenarien erstellt. Anschließend wurden 10 freiwillige Probanden eingeladen und gebeten, die Szenarien in eigenen Worten zu formulieren. Da anzunehmen ist, dass Menschen Aktionen eher selten mehrmals nennen, wenn



## 7.2. Evaluationsvorgang

Um die Ergebnisse des hier entwickelten Agenten zu bewerten, werden die Metriken Präzision, Ausbeute und  $F_1$ -Maß verwendet. Deren Definition wird im folgenden vorgestellt. Anschließend wird vorgestellt unter welchen Umständen der Agent evaluiert wird.

### 7.2.1. Bewertungsmetriken

Es muss überprüft werden, ob die vom Agenten erkannten Vorkommen von konzeptioneller Gleichheit und Aktionskorreferenz denen in der Musterlösung entsprechen. Diese Vorkommen werden vom Agenten als Kanten gespeichert und sind in der Musterlösung ebenfalls als Kanten dargestellt. Für jede vom Agenten erzeugte Kante muss überprüft werden, ob die entsprechende Kante in der Musterlösung zu finden ist. Für alle Kanten in der Musterlösung muss überprüft werden, ob der Agent dem Graphen eine übereinstimmende Kante hinzugefügt hat. Kanten werden als übereinstimmend betrachtet, wenn sie denselben Start- und Zielknoten und den gleichen Typ (konzeptionelle Gleichheit, Aktionskorreferenz) haben.

Die Kanten werden in vier Gruppen eingeteilt:

- richtig-positiv: Die Kante wird vom Agenten hinzugefügt und soll auch laut Musterlösung vorliegen.
- richtig-negativ: Die Kante wird nicht vom Agenten hinzugefügt soll auch laut Musterlösung nicht vorliegen.
- falsch-positiv: Die Kante wird vom Agenten hinzugefügt, aber soll laut Musterlösung nicht vorliegen.
- falsch-negativ: Die Kante wird vom Agenten nicht hinzugefügt und soll laut Musterlösung vorliegen.

Die Anzahl der richtig-positiv erkannten Kanten sei als  $tp$  (*engl. true positive*), die Anzahl der falsch-positiv erkannten Kanten als  $fp$  (*engl. false positive*) und die Anzahl der falsch-negativ erkannten Kanten als  $fn$  (*engl. false negative*) definiert. Mit diesen Werten können die Maßzahlen Präzision, Ausbeute und  $F_1$ -Maß berechnet werden.

Die Präzision (*engl. precision*) gibt den Anteil der richtig erzeugten unter allen erzeugten Kanten an. Sie wird durch folgende Formel berechnet:

$$\text{Präzision} = \frac{tp}{tp + fp} \quad (7.1)$$

Die Ausbeute (*engl. recall*) gibt den Anteil der richtig erzeugten Kanten unter allen Kanten, die laut Musterlösung erzeugt werden sollen an. Sie wird durch folgende Formel berechnet:

$$\text{Ausbeute} = \frac{tp}{tp + fn} \quad (7.2)$$

Die Präzision kann erhöht werden, indem nur Kanten zwischen zwei Aktionsnennungen erzeugt werden, die sicher richtig sind. In diesem Fall könnte die Ausbeute sinken, da richtige Kanten, bei denen sich der Agent nicht sicher ist, nicht hinzugefügt werden würden. Die Ausbeute kann hingegen erhöht werden, indem alle Kanten erzeugt werden, die ein Indiz für Richtigkeit aufweisen. In diesem Fall sinkt die Präzision, da viele falsche Kanten erzeugt werden würden. Um eine einzelne aussagekräftige Metrik zu haben, wird das  $F_1$ -Maß (*engl.  $F_1$ -Score*) betrachtet. Das  $F_1$ -Maß ist das harmonische Mittel aus Präzision und Ausbeute:

$$F_1 = 2 * \frac{\text{Präzision} * \text{Ausbeute}}{\text{Präzision} + \text{Ausbeute}} \quad (7.3)$$

Die Werte für Präzision, Ausbeute und  $F_1$ -Maß liegen zwischen 0 und 1. Ein  $F_1$ -Maß von 1 bedeutet, dass die Ausgabe des Agenten genau mit der Musterlösung übereinstimmt.

Tabelle 7.2.: Evaluationsvarianten

Variante	Eingabe	Aktionen	Korreferenz-Informationen
Variante 1	Transkription	Aktionserkennung	verfügbar
Variante 2	Transkription	Musterlösung	verfügbar
Variante 3	Transkription	Musterlösung	nicht verfügbar

### 7.2.2. Evaluationsvarianten

Um zu untersuchen, wie sich Fehler des Aktionserkenners auf das Ergebnis des hier entwickelten Agenten auswirken, wird die Evaluation sowohl auf der Ausgabe des Aktionserkenners als auch auf den selbst erstellten Musterlösungen des Aktionserkenners durchgeführt. Zudem wird untersucht, wie der Agent ohne Entitätskorreferenz-Informationen abschneidet. Auf Experimente mit der Ausgabe von automatischen Spracherkennern als Eingabe wurde verzichtet, da der Aktionserkennung auf diesen Daten schlechter funktioniert und der zeitliche Rahmen dieser Arbeit es nicht erlaubt hätte ebenso Musterlösungen für diese Daten zu erstellen. Die Erkennung von Korreferenz zwischen Entitäten enthält ebenfalls einige wenige Fehler. Da die Fehleranzahl des Agenten zur Auflösung dieser gering ist ( $F_1$ -Maß von 0,881 [Hey16]) wurde auf das Erstellen von Musterlösungen von diesen Agenten ebenfalls verzichtet. Zusammengefasst fiel die Wahl auf folgende Evaluationsvarianten:

- Variante 1: Die Aktionen sind durch den Aktionserkennung annotiert. Informationen über Korreferenz zwischen Objekten sind verfügbar.
- Variante 2: Die Musterlösung der Aktionserkennung liegen vor. Informationen über Korreferenz zwischen Objekten sind verfügbar.
- Variante 3: Die Musterlösung der Aktionserkennung liegen vor. Informationen über Korreferenz zwischen Objekten sind nicht verfügbar.

Tabelle 7.2 stellt die Varianten tabellarisch dar.

Da der hier entwickelte Agent mit Konfidenzen arbeiten werden Kanten für diese Evaluation nur dann als erkannt betrachtet, wenn ihre Konfidenz einen bestimmten Schwellwert überschreiten. Die Evaluation wird mit verschiedenen Schwellwerten in Schritten von 0,1 zwischen 0,1 und 1,0 durchgeführt. Es ist anzunehmen, dass die Präzision bei Erhöhung des Schwellwerts zunimmt, da nur Kanten derer sich der Agent sicher ist eine hohe Konfidenz haben. Dabei werden richtige Kanten zwischen zwei Aktionsnennungen mit leichter sprachlicher Variation nicht erkannt, weshalb zu erwarten ist, dass die Ausbeute bei hohem Schwellwert steigt. Umgekehrt ist bei niedrigem Schwellwert mit einer hohen Ausbeute und einer niedrigen Präzision zu rechnen, da auch verschiedene Aktionsnennungen als korreferent eingestuft werden, wenn sie genug Gemeinsamkeiten haben.

## 7.3. Evaluationsergebnisse

Im Folgenden werden die Ergebnisse der drei Evaluationsvarianten vorgestellt und miteinander verglichen. Anschließend werden Schwächen des hier entwickelten Agenten diskutiert.

### 7.3.1. Variante 1

Tabelle 7.3 zeigt Präzision, Ausbeute und  $F_1$ -Maß für Variante 1, in der der Agent die tatsächliche Ausgabe des Aktionserkenners und die Informationen über Korreferenzen zwischen Entitäten zur Verfügung hat. Bei der Berechnung der Metriken wurden Kanten

betrachtet, deren Konfidenz mindestens so groß ist, wie der Schwellwert. Die Abhängigkeit der Ergebnisse vom gewählten Schwellwert ist in Abbildung 7.1 grafisch dargestellt.

Tabelle 7.3.: Ergebnisse mit Annotationen des Aktionserkenners und Korreferenzinformationen (Variante 1)

Konfidenz $\geq$	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
Präzision	0,24	0,27	0,32	0,37	0,51	0,69	0,75	<b>0,87</b>	0,83	0,73
Ausbeute	<b>0,66</b>	<b>0,66</b>	0,62	0,62	0,62	0,59	0,57	0,47	0,33	0,14
$F_1$	0,36	0,38	0,42	0,46	0,56	0,64	<b>0,65</b>	0,61	0,47	0,23

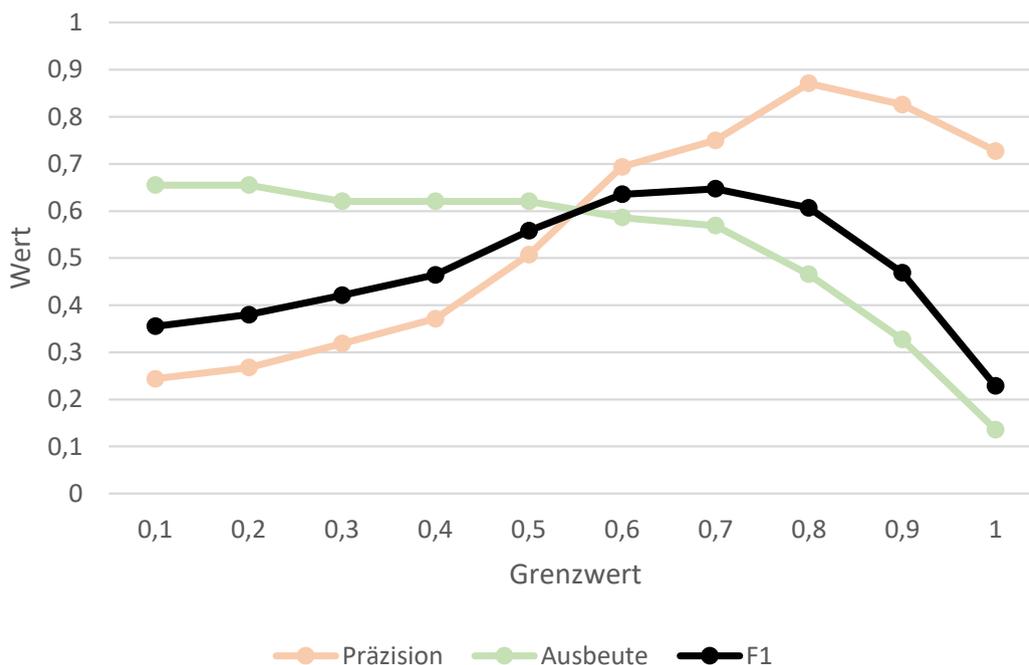


Abbildung 7.1.: Ergebnisse mit Annotationen des Aktionsanalyserers und Korreferenzinformationen (Variante 1)

Wie zu erwarten steigt die Präzision und sinkt die Ausbeute bei Erhöhung des Schwellwerts. Bei einem Schwellwert von 0,9 sinkt die Präzision jedoch überraschenderweise, da nur die Zahl der richtig-positiven Kanten stark sinkt, die Zahl der falsch-positiven Kanten jedoch fast gleich bleibt. Bei einem Schwellwert von 1,0 werden nur 11 Kanten eingefügt, von denen 8 richtig sind, wohingegen bei einem Schwellwert von 0,8 31 eingefügt werden, von denen 27 korrekt sind. Die Ausbeute beträgt 0,66, wenn alle Kanten betrachtet werden, deren Konfidenz mindestens 0,1 ist. Die Ausbeute ist gering, da viele Aktionen vom Aktionserkennner nicht oder falsch erkannt werden. Beispielsweise wird in dem Satz „Armar please open the window“ das Wort *please* als Prädikat erkannt. *Open* wird hingegen nicht annotiert. Nicht annotierte Aktionen werden von dem hier entwickelten Agenten nicht aus dem Graphen extrahiert und mit anderen Aktionen verglichen, weshalb es nicht möglich ist, konzeptionelle Gleichheit oder Aktionskorreferenz bei diesen festzustellen. Zudem teilt der Aktionserkennner Wörtern häufig falsche Rollen zu. Besonders oft werden ACTOR- als WHO- und WHERE- als WHAT-Parameter erkannt. Das  $F_1$ -Maß hat seinen höchsten Wert bei einem Schwellwert von 0,7 und beträgt dort 0,65. Dies bedeutet, dass in dieser Variante

alle Kanten mit einer Konfidenz von unter 0,7 nicht als richtig betrachtet werden sollten, um den besten Kompromiss aus Präzision und Ausbeute zu erhalten.

### 7.3.2. Variante 2

Um nun zu untersuchen, wie gut der hier entwickelte Agent wäre, wenn der Aktionserkennner alle Aktionen richtig erkennt, wurden die echten Annotationen des Aktionserkennners durch Musterlösungen ersetzt. Die Ergebnisse bei Verwendung von Musterlösungen und bei vorheriger Ausführung des Entitätenkorreferenzerkennners werden in Tabelle 7.4 und Abbildung 7.2 dargestellt.

Tabelle 7.4.: Ergebnisse mit korrekt annotierten Aktionen und Korreferenzinformationen (Variante 2)

Konfidenz $\geq$	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
Präzision	0,34	0,38	0,45	0,53	0,63	0,88	0,95	0,95	0,97	<b>1,00</b>
Ausbeute	<b>0,93</b>	<b>0,93</b>	<b>0,93</b>	<b>0,93</b>	0,88	0,74	0,67	0,60	0,55	0,27
$F_1$	0,50	0,54	0,61	0,68	0,73	<b>0,80</b>	0,79	0,74	0,70	0,43

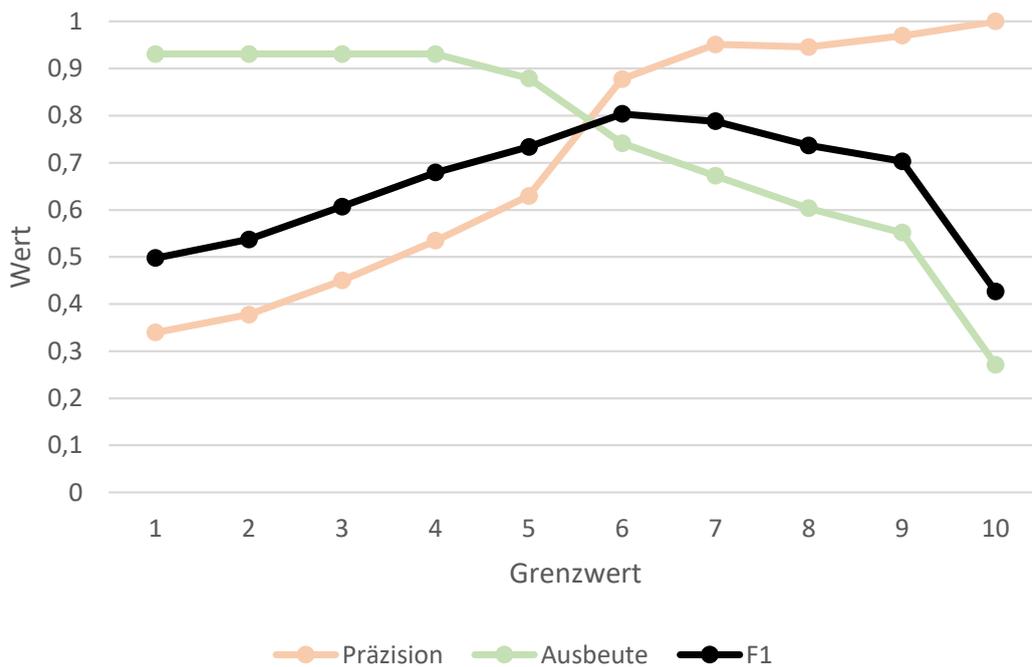


Abbildung 7.2.: Ergebnisse mit korrekt annotierten Aktionen und Korreferenzinformationen (Variante 2)

Beim Vergleich mit Variante 1 fällt auf, dass Ausbeute und  $F_1$ -Maß für alle Schwellwert zunehmen. Die Ausbeute beträgt 0,93 bei einem Schwellwert von 0,1, da nun auch Aktionen, die vorher nicht erkannt worden sind betrachtet werden. Bei steigendem Schwellwert sinkt die Verbesserung der Ausbeute gegenüber Variante 1. Bei einem Schwellwert von 0,7 ist sie nur um 0,1 höher. Die Präzision ist ebenfalls bei jedem Schwellwert höher als in Variante 1. Bei dem Schwellwert 0,7 ist sie um 0,2 höher. Wenn nur Kanten mit einer Konfidenz von 1,0 betrachtet werden, ist die Präzision 1,0. Das bedeutet, dass alle Kanten, die der Agent gefunden hat, richtig sind. Bei niedrigeren Schwellwerten gibt es vereinzelt jedoch Fälle,

bei denen in Variante 2 fälschlicherweise eine Kante gefunden wird, in Variante 1 hingegen diese Kante richtigerweise nicht gefunden wird. Beispielsweise wird in einem der Eingabetexte des Fensters explizit zwei mal geöffnet und später nur auf eine der Öffnen-Aktionen Bezug genommen wird. Der Agent erkennt in Variante 2 fälschlicherweise auch eine Kante zwischen der Referenz-Aktion und der anderen Öffnen-Aktion. Eine dieser Aktionen wurde in Variante 1 nicht erkannt, weshalb dort die Kante korrekt richtig-negativ ist. Wegen der höheren Präzision und Ausbeute ist auch das  $F_1$ -Maß in Variante 2 höher. Das  $F_1$ -Maß hat in Variante 2 sein Maximum bei 0,6. Bei diesem Grenzwert konnte das  $F_1$ -Maß durch Verwendung der Musterlösungen des Aktionserkenners um 0,16 auf 0,8 erhöht werden.

### 7.3.3. Variante 3

In Variante 3 wird nun untersucht, wie gut das Werkzeug funktioniert, wenn keine Informationen über Korreferenzen zwischen Entitäten zur Verfügung stehen. Tabelle 7.5 und Abbildung 7.3 zeigen die Ergebnisse.

Tabelle 7.5.: Ergebnisse mit korrekt annotierten Aktionen und ohne Korreferenzinformationen (Variante 3)

Konfidenz $\geq$	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
Präzision	0,34	0,40	0,50	0,62	0,68	0,85	0,88	<b>0,89</b>	0,88	0,83
Ausbeute	<b>0,93</b>	<b>0,93</b>	<b>0,93</b>	<b>0,93</b>	<b>0,93</b>	0,86	0,76	0,67	0,38	0,17
$F_1$	0,50	0,56	0,65	0,74	0,78	<b>0,85</b>	0,81	0,76	0,53	0,28

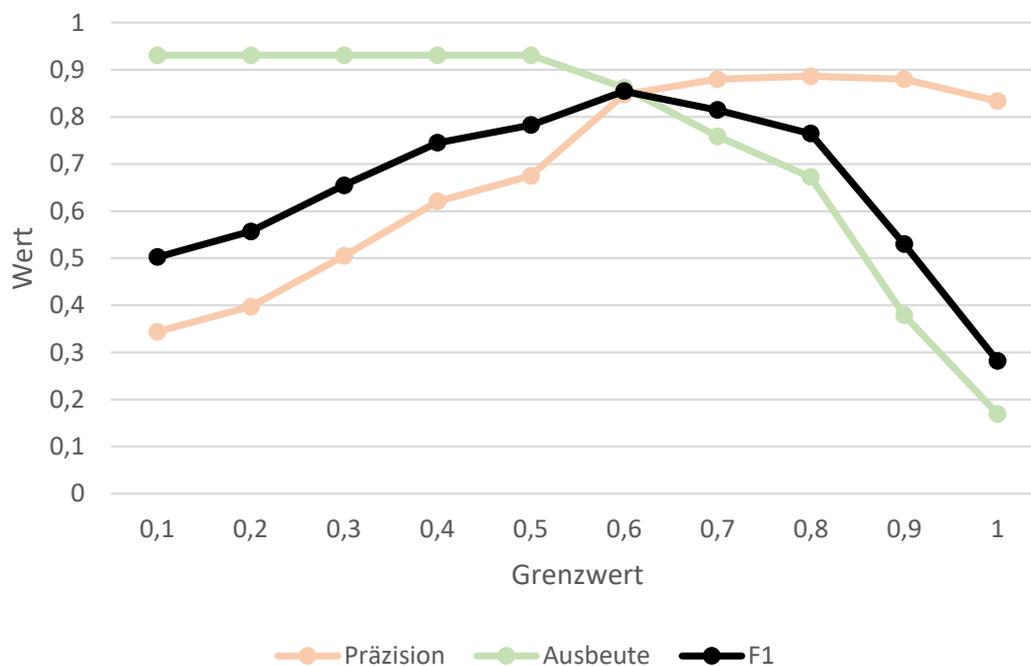


Abbildung 7.3.: Ergebnisse mit korrekt annotierten Aktionen und ohne Korreferenzinformationen (Variante 3)

Wie in Variante 2 hat das  $F_1$ -Maß sein Maximum hier bei 0,6. Interessanterweise ist das  $F_1$ -Maß bei Grenzwerten unter 0,8 in Variante 3 etwas höher als in Variante 2. Dies liegt daran, dass sich Fehler von SNLP sowohl auf den Aktionserkennner, als auch auf den Korreferenzauflöser auswirken. Beispielsweise werden in der Eingabe „close the window look

at the cup on the table“ die Wörter „the window look“ fälschlicherweise zu einer Nominalphrase zusammengefasst. Der Korreferenzauflöser findet in diesem Fall keine Korreferenz zu früher genannten Nennungen von der Entität *window*. Da in der Musterlösung des Aktionserkenners nur „the window“ den WHAT-Teil der Aktion bildet, wird in Variante 3 auch nur dieser Teil mit den WHAT-Teilen anderer Aktionen verglichen. Der hier entwickelte Agent vergleicht also *the window* mit *the window* und stellt fest, dass die Entitäten gleich sind. Um genauer untersuchen zu können, wie Korreferenzinformationen das Ergebnis verbessern, müsste eine Musterlösung für den Korreferenzauflöser erstellt werden. Aus Zeitgründen wurde darauf verzichtet. Ein weiterer Grund, weshalb die Ergebnisse ohne Korreferenzauflöser relativ gut sind, ist, dass wenige gleiche Prädikate mit unterschiedlichen Objekten in den Aufnahmen im Korpus vorliegen. So wird zum Beispiel in keiner Aufnahme sowohl der Kühlschrank als auch das Fenster geöffnet. Falls diese beiden Aktionen vorkommen würden und beide mit „open it“ beschrieben werden würden, würde die Präzision sinken, da eine fehlerhafte Beziehung von dem hier entwickelten Agenten zwischen diesen gefunden werden würde. Der Korreferenzerkennung würde in diesem Fall feststellen, dass sich die beiden Verwendungen von *it* auf verschiedene Entitäten bezieht. Bei den Schwellwerten von 0,9 und 1,0 ist das  $F_1$ -Maß in Variante 3 geringer als in Variante 2, da die Ausbeute bei einem Schwellwert von über 0,8 sprunghaft abnimmt. Die Ausbeute sinkt, da immer etwas von der Konfidenz abgezogen wird, wenn ein Akteur in einer Aktionsnennung ein Substantiv und in der anderen Nennung ein Personalpronomen ist. Wenn dies der Fall ist, wird 0,15. Die Präzision sinkt bei hohen Schwellwerten auch, weil wie in Variante 1 die Anzahl der richtig-positiven Kanten stark abnimmt, wohingegen die Anzahl der falsch-positiven nur wenig sinkt.

### 7.3.4. Fehler durch den Agenten

Wie die Evaluationsergebnisse gezeigt haben, erzielt der hier entwickelte Agent selbst unter optimalen Bedingungen verbesserungsfähige Ergebnisse. Einige Fehler resultieren aus fehlerhafter Grammatik und falscher Wortwahl in den Sprachaufnahmen. Der Agent kann Da keiner der eingeladenen Probanden Englisch als Muttersprache spricht, war das Auftreten von sprachlichen Fehlern zu erwarten. Aber auch bei perfektem Englisch findet der Agent einige Vorkommnisse von Aktionskorreferenz und konzeptioneller Gleichheit nicht und hält dagegen Aktionen für gleich, die unterschiedlich sind. Im Folgenden werden Fehlerklassen erläutert und diskutiert, wie diese behoben werden können.

#### 7.3.4.1. Zufällige semantische Beziehungen zwischen unterschiedlichen Bedeutungsausprägungen

Es wird nicht untersucht, welche Bedeutung ein Wort hat. Es kann vorkommen, dass ein Wort mehrere Bedeutungen hat (Polysemie) und eine falsche Bedeutung in einer Relation zu einer Bedeutung eines anderen Wortes steht. Beispielsweise wird fälschlicherweise Troponymie zwischen *open* und *move* in Anweisungen wie „open the fridge“ und „move to the fridge“ festgestellt. In WordNet existiert diese Beziehung nämlich zwischen den folgenden Synsets:

- S: (v) open (make the opening move) "*Kasparov opened with a standard opening*"
- S: (v) move, go (have a turn; make one's move in a game) "*Can I go now?*"

*Open* kann also auf das Eröffnen im Schach und *move* das allgemeinere Ziehen im Schach bedeuten. Um dieses Problem zu lösen, müssten sprachliche Mehrdeutigkeiten aufgelöst werden. Es existiert ein Agent für Disambiguierung (*engl. Word-sense disambiguation*) in PARSE, der im Rahmen der Masterarbeit von Keim [Kei18] entwickelt wurde. Mit den Ergebnissen dieses Agenten wäre es möglich nur die jeweils auftretenden Bedeutungsausprägungen auf semantische Relationen zu untersuchen.

#### 7.3.4.2. Falsche Annahmen über Verwendung von Zeitformen

In Abschnitt 5.3.1.2 wurde angenommen, dass im Imperativ beschriebene Aktionen mit dem Present Perfect referenziert werden, um eine andere Aktion zeitlich einzuordnen. Die Idee hinter dieser Annahme war, dass die Ausführung dieser Aktion für die Aktion die eingeordnet wird, relevant ist. Diese Annahme war zu streng. In ungefähr der Hälfte der Eingaben wird das Simple Present verwendet, um diese Aktionen zu referenzieren. Die falsche Annahme hat zur Folge, dass in diesen Fällen 0,05 von der Konfidenz für Aktionskorreferenz abgezogen wird. Dieser Fehler kann einfach behoben werden, indem in diesem Fall nichts mehr abgezogen wird.

#### 7.3.4.3. Mehrdeutigkeit des Wortes *do*

Eine Aufnahme enthielt die Sätze „take one cup [...] do it on the table“. Hier nimmt der Agent an, dass die zweite Aktion eine Spezifizierung der ersten Aktion ist, weil das Prädikat *do* und der WHAT-Parameter *it* ist. Jedoch ist mit „do“ „stellen“ gemeint und „it“ bezieht sich auf den Becher und nicht auf die Aktion. Das Verb wird zwar in dieser Aufnahme nicht korrekt verwendet, es sind jedoch auch korrekte Verwendungen denkbar, in denen *do* das Verb und *it* ein Objekt ist und die eine neue Aktion beschreiben. So würde der Agent bei der Eingabe „The teacher gave John homework[.] John did it after school“ Aktionskorreferenz zwischen den beiden Sätzen feststellen. Um das Problem zu lösen, müsste man überprüfen, ob Korreferenz zwischen dem Wort *it* und einer Entität besteht. Falls dies so ist, kann sich das Personalpronomen nicht auf eine Aktion beziehen.

#### 7.3.4.4. Spezifizierung und *do that* bei nicht direkt aufeinanderfolgenden Aktionen

Es wurde davon ausgegangen, dass eine Wiederholung einer Aktionsnennung zur Anreicherung mit weiteren Informationen direkt auf die Aktion, die unvollständig ist folgt. Ebenso wurde angenommen, dass sich Formulierung wie „do that“ immer auf die direkt davor genannten Aktionen beziehen. Ein Proband wählte die Formulierung „pick up the cup on the table make sure that you pick up the green cup“, um eine Aktion genauer zu beschreiben. Hierbei wird „make sure“ als Aktion erkannt, was bedeutet, dass die generelle Aktion und die Aktion, die diese spezifiziert sich in der Befehlsnummer nicht um genau eins unterscheiden. Fehler in der Eingabe können ebenfalls dazu führen, dass Aktionen nicht direkt aufeinander folgen. Einzelne Verzögerungslaute wirken sich nicht auf das Ergebnis aus, aber Versprecher wie „open the door ehm after you took ehm I mean after you did that...“ führen dazu, dass zusätzliche Aktionen zwischen der referenzierten und referenzierenden Aktionsnennung vorliegen. Man könnte das Problem beheben, indem man die Überprüfung auf Spezifizierung auch dann durchführt, wenn sich die Befehlsnummern um einen größeren Betrag  $d$  als eins unterscheiden. Dabei würde die Ausbeute bei größerem  $d$  steigen. Die Präzision würde jedoch bei zu großem  $d$  sinken, da das mehrfache Ausführen einer Aktion falsch erkannt werden würde, wenn die Nennungen nah bei einander sind und die zweite Nennung mehr Informationen enthält. Es kann vermutet werden, dass  $d = 3$  ein Wert ist, bei dem sich die Präzision kaum verringert. Es wäre denkbar, bestimmte Formulierungen wie „make sure“ gesondert zu betrachten und in diesem Fall eine weitere Aktionsnennung zwischen Generalisierung und Spezifizierung zuzulassen.

#### 7.3.4.5. Fehlende Parameter in Aktionsnennungen

Wie zu erwarten, werden oft bestimmte Teile von Aktionen in der zweiten Nennung weglassen, da sie bereits in der Nennung genannt werden. In diesem Fall wird wie erwartet ein kleiner Betrag von der Konfidenz abgezogen. Es wurde davon ausgegangen, dass eher in der ersten Nennung mehr Parameter genannt werden, weshalb mehr abgezogen wird,

wenn ein Parameter nur in der zweiten Aktionsnennung genannt wird. Jedoch treten in den Sprachaufnahmen Formulierungen wie „Open the fridge and take the orange juice [...] after you have taken the orange juice from the fridge[...]“ auf. In der ersten Nennung, die beschreibt, dass der Orangensaft genommen werden soll, wird der Ort nicht explizit genannt. Für Menschen ist es eindeutig, dass der Saft im Kühlschrank ist, da dieser zuvor zu diesem Zweck geöffnet worden ist. In der zweiten Nennung der Aktion wird der Ort jedoch genannt. Da die Annahme, dass die erste Aktionsnennung mehr Informationen enthält, sich als nicht immer zutreffend herausstellte, wäre es sinnvoll, bei Ortsangaben weniger abzuziehen, wenn die Angabe nur in der zweiten Aktion gegeben ist.

## 7.4. Bewertung der Ergebnisse

Die Evaluationsergebnisse zeigen, dass der hier entwickelte Agent durchaus in der Lage ist, selbst bei fehlerbelasteter Eingaben, die der Aktionserkennung liefert, in den hier gewählten Szenarien mehr als die Hälfte aller Vorkommnisse von konzeptioneller Gleichheit und Aktionskorreferenz richtig zu erkennen. Es hat sich gezeigt, dass die besten Ergebnisse in Variante 2 und 3 vorliegen, wenn nur Kanten mit einer Konfidenz von mindestens 0,6 betrachtet werden, da in diesem Fall das  $F_1$ -Maß maximal ist. Bei diesem Schwellwert ist das  $F_1$ -Maß in Variante 2 und 3 um 0,16 bzw. 0,21 höher als in Variante 1. Es hat sich auch gezeigt, dass die Verfügbarkeit von Informationen über Korreferenz zwischen Entitäten das Ergebnis nicht verbessern konnten. Die Ergebnisse des Agenten könnten gesteigert werden, indem die in Abschnitt 7.3.4 diskutierten Änderungen umgesetzt werden.

Die Aussagekräftigkeit dieser Evaluation ist allerdings fraglich. Es wurden lediglich zwei kurze Szenarien untersucht. Dabei traten nur wenige sprachliche Variationen bei der Beschreibung der Aktionen auf, wie zum Beispiel die Verwendung von Synonymen. Korreferenzinformationen waren in den Eingaben eher selten nötig, da wenige ähnliche Entitäten existierten. Um den Agenten genauer bewerten zu können, müssten Aufnahmen zu längere Szenarien erstellt werden. Es wäre auch interessant zu untersuchen, wie der Agent bei Eingaben mit anderen Einsatzumgebungen als einer Küche abschneidet.

Eine Einordnung der Ergebnisse ist nur schwer möglich, da Korreferenz zwischen Aktionen in der Forschung nicht untersucht wird. Der Agent erzielt zwar in Variante 3 bessere Ergebnisse als die meisten in Kapitel 4 vorgestellten verwandten Arbeiten, wenn man nur Kanten mit einer Konfidenz von mindestens 0,6 betrachtet, jedoch untersuchen diese komplexere Ereignisse und arbeiten auf geschriebener Sprache. Die verwandten Arbeiten wurden auf deutlich größeren Korpora getestet, was deren Ergebnisse viel aussagekräftiger macht.

## 8. Zusammenfassung und Ausblick

Das Ziel dieser Bachelorarbeit war die Entwicklung eines Agenten zur Erkennung von Aktionskorreferenz in gesprochener Sprache. Es wurde zuerst definiert, wann zwei Aktionen gleich sind und welche Arten von Gleichheit es zwischen Aktionen geben kann. Anschließend wurde untersucht, welche Voraussetzungen für Gleichheit erfüllt sein müssen und unter welchen Bedingungen Aktionen mehrmals genannt werden. Mit diesen Erkenntnissen wurde ein Vorgehen abgeleitet und als Agent für die Rahmenarchitektur PARSE implementiert. Der implementierte Agent wurde evaluiert. Die Evaluation hat gezeigt, dass die meisten Vorkommnisse von konzeptioneller Gleichheit und Aktionskorreferenz richtig aufgelöst werden konnten. Bei der Wahl eines Schwellwerts von 0,6 für die Konfidenzwerte erzielte der Agent ein  $F_1$ -Maß von über 0,8, wenn alle Aktionen richtig erkannt werden. Die Fehler des Aktionserkenners wirken sich deutlich auf die Ergebnisse des hier entwickelten Agenten aus. Das  $F_1$ -Maß war bei Verwendung der Ausgabe des Aktionserkenners nur 0,61 bei Betrachtung des gleichen Schwellwerts. Bei der Wahl der Strafwerte der einzelnen Filter wurden Schätzwerte verwendet. Die Strafwerte könnten in Zukunft mithilfe maschineller Lernverfahren genauer eingestellt werden, um bessere Ergebnisse zu erzielen.

Die Verwendung der in PARSE bereits vorhandenen Aktionserkennung stellte eine Einschränkung im Entwurf des Agenten dar. Der Aktionserkennung erlaubt höchstens einen Parameter jedes Typs in einer Aktion. Wenn eine Aktion jedoch mehrere WHAT-Parameter hat, werden diese konkateniert und als ein einzelner Parameter dargestellt. Dies machte den Vergleich dieser Parameter in verschiedenen Aktionen nur eingeschränkt möglich. Der hier entwickelte Agent vergleicht nur Parameter vom jeweils gleichen Typ in zwei Aktionen. Beim Vergleich von Aktionen wie „Alice killed Bob“ und „Alice killed a man“ werden „Bob“ und „a man“ nicht verglichen, da „Bob“ ein WHO- und „a man“ ein WHAT-Parameter ist. Dies liegt daran, dass im Aktionserkennung Eigennamen WHO-Parameter sind. Eine vermutlich sinnvollere Alternative als die Verwendung des Aktionserkenners wäre die Betrachtung der semantischen Relationen. In beiden Sätzen nehmen die Parameter die gleiche semantische Rolle ein.

Einige Fälle wurden in dieser Arbeit nicht betrachtet. So wurden immer nur Aktionsnennungen betrachtet, die sich auf höchstens eine Aktionsidentität beziehen. Fälle, in denen sich eine Aktionsnennung auf eine Mehrzahl von bereits genannten Aktionen beziehen, wurden nicht implementiert. Ebenfalls nicht betrachtet wurde der Fall, bei dem sich ein Substantiv (z.B. *development*) auf eine Aktion (*develop*) bezieht. Das Demonstrativpronomen *that* wird nur als Referenz auf eine Aktion erkannt, wenn es von einem Verb wie *do* begleitet wird. Es gibt jedoch auch Fälle, wo das Demonstrativpronomen alleine steht. Da

angenommen wurde, dass diese Fälle eher selten auftreten, wurde der Fokus dieser Arbeit nicht auf diese Fälle gelegt. Des Weiteren wurden keine Fälle betrachtet, in denen zwei Aktionen durch Prädikate beschrieben werden, die für sich genommen nicht durch semantische Beziehungen verbunden sind. So erkennt das Vorgehen keine Korreferenz zwischen „kill yourself“ und „commit suicide“ oder „compliment someone“ und „tell someone that they are pretty“. Diese Fälle sind sehr kompliziert und lassen sich durch einfachen Vergleich der einzelnen Wörter nicht lösen. Bei einer Weiterentwicklung des Agenten könnten die oben genannten Fälle weiter untersucht werden.

# Literaturverzeichnis

- [ALHM14] ARAKI, Jun ; LIU, Zhengzhong ; HOVY, Eduard H. ; MITAMURA, Teruko: Detecting Subevent Structure for Event Coreference Resolution. In: *LREC*, 2014, S. 4553–4558 (zitiert auf den Seiten 17 und 18).
- [ARA<sup>+</sup>06] ASFOUR, Tamim ; REGENSTEIN, Kristian ; AZAD, Pedram ; SCHRODER, J. ; BIERBAUM, Alexander ; VAHRENKAMP, Nikolaus ; DILLMANN, RÄ $\frac{1}{4}$ diger: ARMAR-III: An integrated humanoid platform for sensory-motor control. In: *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, IEEE, 2006, S. 169–175 (zitiert auf den Seiten 1 und 9).
- [BB98] BAGGA, Amit ; BALDWIN, Breck: Algorithms for scoring coreference chains. In: *The first international conference on language resources and evaluation workshop on linguistics coreference* Bd. 1, Granada, Spain, 1998, S. 563–566 (zitiert auf Seite 15).
- [BB99] BAGGA, Amit ; BALDWIN, Breck: Cross-document event coreference: Annotations, experiments, and observations. In: *Proceedings of the Workshop on Coreference and its Applications*, Association for Computational Linguistics, 1999, S. 1–8 (zitiert auf den Seiten 15, 16 und 22).
- [BH10] BEJAN, Cosmin A. ; HARABAGIU, Sanda: Unsupervised event coreference resolution with rich linguistic features. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2010, S. 1412–1422 (zitiert auf den Seiten xi, xiii, 14, 17 und 22).
- [Car12] CARNIE, A.: *Syntax: A Generative Introduction 3rd Edition and The Syntax Workbook Set*. Wiley, 2012 (Introducing Linguistics). <https://books.google.de/books?id=jhGKMAEACAAJ>. – ISBN 978–1–118–51792–5 (zitiert auf Seite 4).
- [CH17] CHOUBEY, Prafulla K. ; HUANG, Ruihong: Event Coreference Resolution by Iteratively Unfolding Inter-dependencies among Events. In: *arXiv:1707.07344 [cs]* (2017), Juli. <http://arxiv.org/abs/1707.07344>. – arXiv: 1707.07344 (zitiert auf Seite 19).
- [CJ09] CHEN, Zheng ; JI, Heng: Graph-based event coreference resolution. In: *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, Association for Computational Linguistics, 2009, S. 54–57 (zitiert auf Seite 16).
- [CM05] CARRERAS, Xavier ; MÃ RQUEZ, LluÃs: Introduction to the CoNLL-2005 shared task: Semantic role labeling. In: *Proceedings of the Ninth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2005, S. 152–164 (zitiert auf Seite 7).
- [CP04] CHKLOVSKI, Timothy ; PANTEL, Patrick: Verbocean: Mining the web for fine-grained semantic verb relations. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004 (zitiert auf Seite 5).

- [CV14] CYBULSKA, Agata ; VOSSEN, Piek: Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. In: *LREC*, 2014, S. 4545–4552 (zitiert auf Seite 20).
- [CWB<sup>+</sup>11] COLLOBERT, Ronan ; WESTON, Jason ; BOTTOU, Léon ; KARLEN, Michael ; KAVUKCUOGLU, Koray ; KUKSA, Pavel P.: Natural Language Processing (almost) from Scratch. In: *CoRR* abs/1103.0398 (2011). <http://arxiv.org/abs/1103.0398> (zitiert auf Seite 9).
- [Dal16] DALE, Robert: The return of the chatbots. In: *Natural Language Engineering* 22 (2016), September, Nr. 5, 811–817. <http://dx.doi.org/10.1017/S1351324916000243>. – DOI 10.1017/S1351324916000243. – ISSN 1351–3249, 1469–8110 (zitiert auf Seite 1).
- [Dav69] DAVIDSON, Donald: The individuation of events. In: *Essays in honor of Carl G. Hempel*. Springer, 1969, S. 216–234 (zitiert auf Seite 22).
- [DG03] DANLOS, Laurence ; GAIFFE, Bertrand: Event coreference and discourse relations. In: *PHILOSOPHICAL STUDIES SERIES* 102 (2003), S. 191–210 (zitiert auf Seite 31).
- [DMP<sup>+</sup>04] DODDINGTON, George R. ; MITCHELL, Alexis ; PRZYBOCKI, Mark A. ; RAMSHAW, Lance A. ; STRASSEL, Stephanie ; WEISCHEDEL, Ralph M.: The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. In: *LREC* Bd. 2, 2004, S. 837–840 (zitiert auf Seite 13).
- [Eas05] EASTWOOD, John: *Oxford guide to English grammar*. [Impr.] 9. Oxford : Oxford Univ. Press, 2005. – ISBN 978–0–19–431351–3. – OCLC: 179987249 (zitiert auf Seite 28).
- [ext] *extJWNL (Extended Java WordNet Library)*. <http://extjwnl.sourceforge.net/> (zitiert auf Seite 46).
- [Fel05] FELLBAUM, C: WordNet and wordnets. (2005), 01, S. 665–670 (zitiert auf den Seiten 7 und 9).
- [FGM05] FINKEL, Jenny R. ; GRENAGER, Trond ; MANNING, Christopher: Incorporating non-local information into information extraction systems by gibbs sampling. In: *Proceedings of the 43rd annual meeting on association for computational linguistics*, Association for Computational Linguistics, 2005, S. 363–370 (zitiert auf Seite 18).
- [G15] GÜNES, Zeynep: *Aufbau eines Sprachkorpus zur Programmierung autonomer Roboter mittels natürlicher Sprache*, Karlsruher Institut für Technologie (KIT) – IPD Tichy, Bachelor’s Thesis, Mai 2015. [https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/guenes\\_ba](https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/guenes_ba) (zitiert auf Seite 49).
- [geo] *GeoNames*. <http://www.geonames.org/> (zitiert auf Seite 18).
- [GS96] GRISHMAN, Ralph ; SUNDHEIM, Beth: Message understanding conference-6: A brief history. In: *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics* Bd. 1, 1996 (zitiert auf Seite 13).
- [Hey16] HEY, Tobias: *Kontext- und Korreferenzanalyse für gesprochene Sprache*, Karlsruher Institut für Technologie (KIT) – IPD Tichy, Master’s Thesis, September 2016. [https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/hey\\_ma](https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/hey_ma) (zitiert auf den Seiten 10, 11, 25, 37 und 52).

- [HVM<sup>+</sup>13] HOVY, Eduard ; MITAMURA, Teruko ; VERDEJO, Felisa ; ARAKI, Jun ; PHILPOT, Andrew: Events are not simple: Identity, non-identity, and quasi-identity. In: *NAACL HLT* Bd. 2013, 2013, S. 21 (zitiert auf den Seiten 14, 18 und 34).
- [JM09] JURAFSKY, Daniel ; MARTIN, James H.: *Speech and Language Processing (2Nd Edition)*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 2009. – ISBN 0–13–187321–0 (zitiert auf den Seiten 4, 5 und 26).
- [Kei18] KEIM, Jan: *Themenextraktion zur Domänenauswahl für Programmierung in natürlicher Sprache*, Karlsruher Institut für Technologie (KIT) – IPD Tichy, Master’s Thesis, Februar 2018. [https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/keim\\_ma](https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/keim_ma) (zitiert auf Seite 56).
- [Koc15] KOCYBIK, Markus: *Projektion von gesprochener Sprache auf eine Handlungsrepräsentation*, Karlsruher Institut für Technologie (KIT) – IPD Tichy, Bachelor’s Thesis, 2015. [https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/kocybik\\_ba](https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/kocybik_ba) (zitiert auf Seite 10).
- [LAHM14] LIU, Zhengzhong ; ARAKI, Jun ; HOVY, Eduard ; MITAMURA, Teruko: Supervised Within-Document Event Coreference using Information Propagation. In: *Proceedings of the International Conference on Language Resources and Evaluation*, 2014 (zitiert auf Seite 18).
- [LIJ<sup>+</sup>15] LEHMANN, Jens ; ISELE, Robert ; JAKOB, Max ; JENTZSCH, Anja ; KONTOKOSTAS, Dimitris ; MENDES, Pablo N. ; HELLMANN, Sebastian ; MORSEY, Mohamed ; KLEEF, Patrick v. ; AUER, Sören ; BIZER, Christian: DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. In: *Semantic Web* 6 (2015), Nr. 2, 167–195. <http://ub-madoc.bib.uni-mannheim.de/37476/> (zitiert auf Seite 18).
- [LPC<sup>+</sup>11] LEE, Heeyoung ; PEIRSMAN, Yves ; CHANG, Angel ; CHAMBERS, Nathanael ; SURDEANU, Mihai ; JURAFSKY, Dan: Stanford’s multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In: *Proceedings of the fifteenth conference on computational natural language learning: Shared task*, Association for Computational Linguistics, 2011, S. 28–34 (zitiert auf Seite 17).
- [LRC<sup>+</sup>12] LEE, Heeyoung ; RECASENS, Marta ; CHANG, Angel ; SURDEANU, Mihai ; JURAFSKY, Dan: Joint Entity and Event Coreference Resolution Across Documents. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Stroudsburg, PA, USA : Association for Computational Linguistics, 2012 (EMNLP-CoNLL ’12), 489–500 (zitiert auf den Seiten 16 und 17).
- [Luo05] LUO, Xiaoqiang: On coreference resolution performance metrics. In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2005, S. 25–32 (zitiert auf Seite 16).
- [Mor38] MORRIS, Charles: *Foundations of the Theory of Signs*. Chicago : University of Chicago Press, 1938. – ISBN 978–0–226–57577–3 (zitiert auf den Seiten 3 und 4).
- [Mur03] MURPHY, M. L.: *Semantic Relations and the Lexicon: Antonymy, Synonymy and other Paradigms*. Cambridge : Cambridge University Press, 2003. <http://dx.doi.org/10.1017/CB09780511486494>. <http://dx.doi.org/10.1017/CB09780511486494>. – ISBN 978–0–511–48649–4 (zitiert auf Seite 5).

- [Ou16] OU, Yue: *Erkennung von Aktionen in gesprochener Sprache*, Karlsruher Institut für Technologie (KIT) – IPD Tichy, Bachelor’s Thesis, September 2016. [https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/ou\\_ba](https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/ou_ba) (zitiert auf den Seiten 10, 11, 21, 35 und 38).
- [PK17] PFISTER, Beat ; KAUFMANN, Tobias: *Sprachverarbeitung*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2017. <http://dx.doi.org/10.1007/978-3-662-52838-9>. <http://dx.doi.org/10.1007/978-3-662-52838-9>. – ISBN 978-3-662-52837-2 978-3-662-52838-9 (zitiert auf Seite 3).
- [PLR<sup>+</sup>14] PRADHAN, Sameer ; LUO, Xiaoqiang ; RECASENS, Marta ; HOVY, Eduard ; NG, Vincent ; STRUBE, Michael: Scoring coreference partitions of predicted mentions: A reference implementation. In: *Proceedings of the conference. Association for Computational Linguistics. Meeting Bd.* 2014, NIH Public Access, 2014, S. 30 (zitiert auf Seite 20).
- [RM99] RAMSHAW, Lance A. ; MARCUS, Mitchell P.: Text chunking using transformation-based learning. In: *Natural language processing using very large corpora*. Springer, 1999, S. 157–176 (zitiert auf Seite 6).
- [Sch17] SCHLERETH, Mario: *Entwicklung eines Dialogagenten für dialogbasierte Programmierung*, Karlsruher Institut für Technologie (KIT) – IPD Tichy, Master’s Thesis, April 2017. [https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/schlereth\\_ma](https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/schlereth_ma) (zitiert auf Seite 12).
- [SM00] SHI, Jianbo ; MALIK, Jitendra: Normalized cuts and image segmentation. In: *IEEE Transactions on pattern analysis and machine intelligence* 22 (2000), Nr. 8, S. 888–905 (zitiert auf Seite 16).
- [Ste16] STEURER, Vanessa: *Strukturerkennung von Bedingungen in gesprochener Sprache*, Karlsruher Institut für Technologie (KIT) – IPD Tichy, Bachelor’s Thesis, 2016. [https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/steuerer\\_ba](https://code.ipd.kit.edu/weigelt/parse/wikis/Theses/steuerer_ba) (zitiert auf Seite 11).
- [WT15] WEIGELT, Sebastian ; TICHY, Walter: Poster: ProNat: An Agent-Based System Design for Programming in Spoken Natural Language, 2015, S. 819–820 (zitiert auf Seite 9).

# Anhang

## A. Markierungssätze der Penn-Treebank

### A.1. Wortartmarkierungen

Tabelle A.1.: Wortartmarkierungen

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP\$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense

VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP\$	Possessive wh-pronoun
WRB	Wh-adverb

## A.2. Phrasenmarkierungen

Tabelle A.2.: Phrasenmarkierungen

Tag	Description
NP	noun phrase
PP	prepositional phrase
VP	verb phrase
ADVP	adverb phrase
ADJP	adjective phrase
SBAR	subordinating conjunction
PRT	particle
INTJ	interjection

## B. Strafwerte

In Tabelle B.3 werden die Werte aufgelistet, die die einzelnen Filter von den Konfidenzen abziehen, wenn bestimmte Voraussetzungen erfüllt werden. Es handelt sich dabei um die Werte, die bei der Implementierung aufgrund von den Annahmen in Kapitel 5 gewählt wurden. Diese Werte sind nicht optimiert.

Tabelle B.3.: Strafwerte der Filter.  $p_n$  bezeichnet Prädikate,  $a_n$  Nominalphrasen ohne Artikel und  $h_n$  für Adverbien.  $n$  ist 1 bei der erstgenannten und 2 bei der zweitgenannten Aktion.

Filter	Fall	Strafwert	
PredicateSieve	$p_1 = p_2$	0,0	
	$p_1$ enthält $p_2$	0,15	
	$p_1$ ist Synonym von $p_2$	0,25	
	$p_1$ ist Troponym von $p_2$	0,4	
	$p_2$ ist Troponym von $p_1$	0,75	
	keine semantische Beziehung	1,0	
	$p_1$ oder $p_2$ fehlt	nicht möglich	
ActorSieve	$a_1$ und $a_2$ fehlen	0,0	
	$a_1$ fehlt, $a_2$ vorhanden	0,1	
	$a_2$ fehlt, $a_1$ vorhanden	nicht möglich	
	Ergebnisse des Korreferenzanalysierers vorhanden und keine zwei unbest. Artikel	(1-Konfidenz der Korref-Kante)*0,5	
	$a_1 = a_2$	0,0	
	$a_1$ enthält $a_2$	0,1	
	$a_1$ ist Synonym von $a_2$	0,1	
	$a_1$ ist Hyponym von $a_2$	0,15	
	$a_1$ ist Hyperonym von $a_2$	0,2	
	Substantive in $a_1$ sind Substantive in $a_2$	0,2	
	keine semantische Beziehung	0,5	
	Artikel von $a_1$ bestimmt und Artikel von $a_2$ unbestimmt	0,25 (zusätzlich)	
	$a_1$ und $a_2$ sind gleiche Pronomen	0,0	
	$a_1$ und $a_2$ sind verschiedene Pronomen genau $a_1$ oder $a_2$ ist Pronomen	0,5	
	$a_1$ ist Substantiv und $a_2$ Eigenname oder umgekehrt	0,2	
WhatSieve	$a_1$ und $a_2$ fehlen	0,0	
	$a_1$ fehlt, $a_2$ vorhanden	0,1	
	$a_2$ fehlt, $a_1$ vorhanden	0,2	
	Ergebnisse des Korreferenzanalysierers vorhanden und keine zwei unbest. Artikel	(1-Konfidenz der Korref-Kante)*0,5	
	$a_1 = a_2$	0,0	
	$a_1$ enthält $a_2$	0,1	
	$a_1$ ist Synonym von $a_2$	0,1	
	$a_1$ ist Hyponym von $a_2$	0,15	
	$a_1$ ist Hyperonym von $a_2$	0,2	
	Substantive in $a_1$ sind Substantive in $a_2$	0,2	
	keine semantische Beziehung	0,5	
	Artikel von $a_1$ bestimmt und Artikel von $a_2$ unbestimmt	0,25 (zusätzlich)	
	WhoSieve	$a_1$ und $a_2$ fehlen	0,0

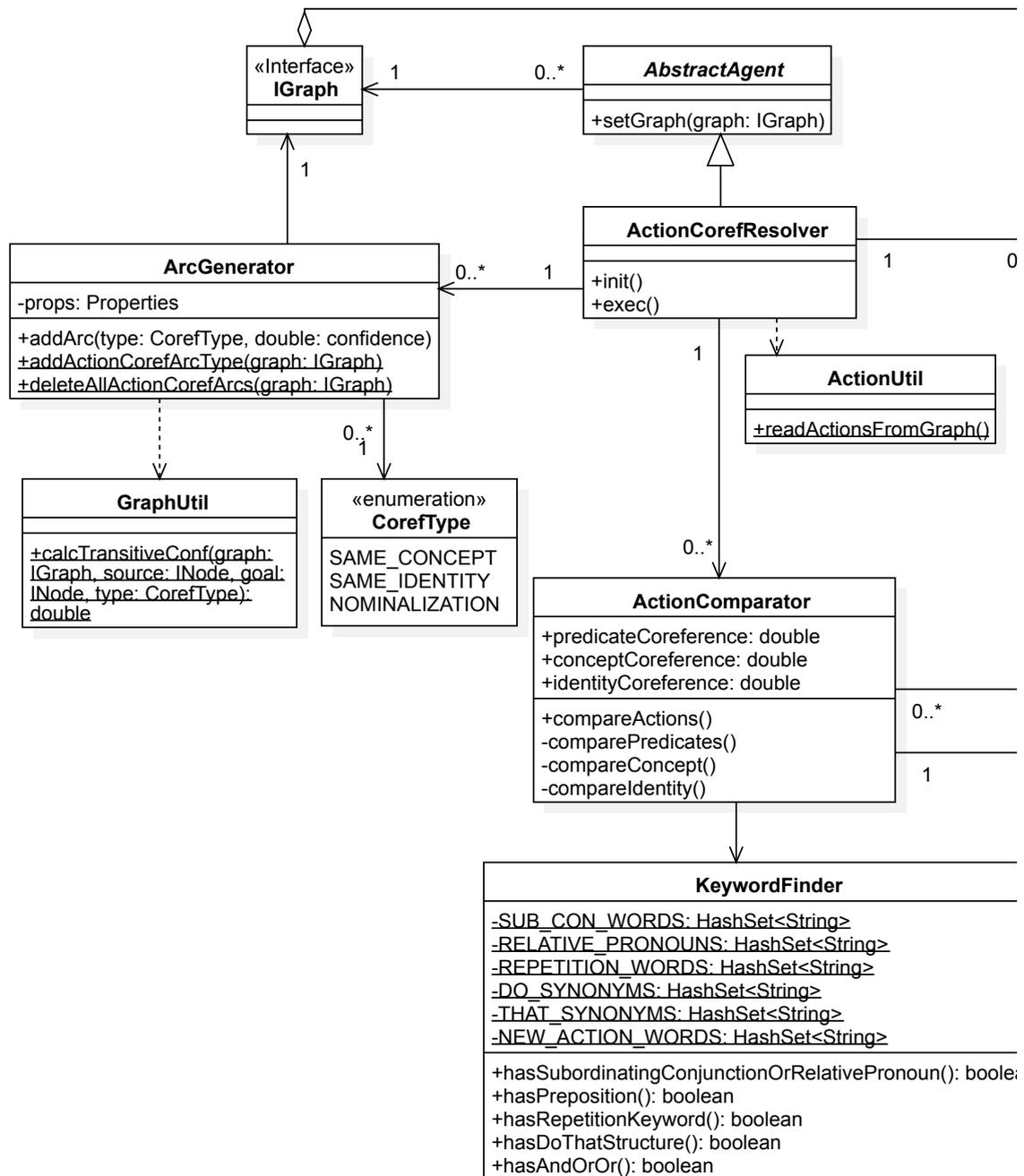
HowSieve	$a_1$ fehlt, $a_2$ vorhanden	0,1
	$a_2$ fehlt, $a_1$ vorhanden	0,2
	Ergebnisse des Korreferenzanalysierers vorhanden	(1-Konfidenz der Korref-Kante)*0,5
	$a_1 = a_2$	0,0
	$a_1$ enthält $a_2$	0,2
	$a_2$ enthält $a_1$	0,2
	$a_1$ und $a_2$ sind verschiedene Pronomen	0,5
	nur $a_2$ ist Pronomen	0,15
	nur $a_1$ ist Pronomen	0,2
	keine Beziehung zwischen $a_1$ und $a_2$	0,5
	$h_1$ und $h_2$ fehlen	0,0
	$h_2$ fehlt, $h_1$ vorhanden	0,0
	$h_1$ fehlt, $h_2$ vorhanden	0,2
	$h_1 = h_2$	0,0
	$h_1$ ist Synonym von $h_2$	0,1
TempRelAna.	$h_1$ enthält $h_2$	0,05
	$h_2$ enthält $h_1$	0,15
	keine Beziehung zwischen $h_1$ und $h_2$	0,3
	keine unterordnende Konjunktion	0,5
	eine Aktion in Folgeanweisung und andere Aktion in Alternativanweisung	0,5
	Befehlsnummern unterscheiden sich um 1	0,05
	Befehlsnummern unterscheiden sich um 1 und Aktionen sind die zwei ersten oder letzten Aktionen	0,3
Tempora verschieden	siehe Tabelle B.4	

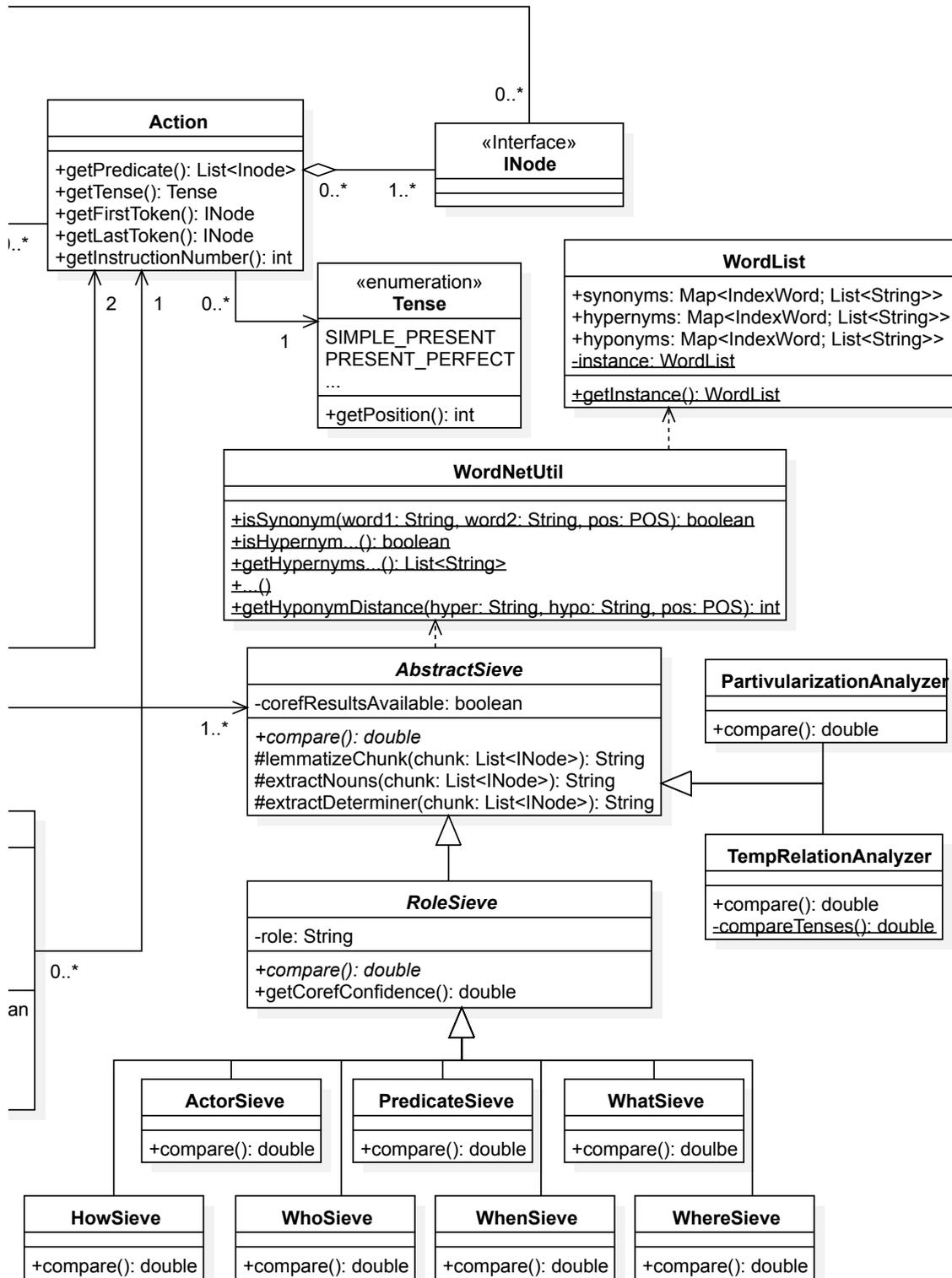
Tabelle B.4.: Strafwerte für Tempora. In den Zeilen befinden sich die Tempora im Hauptsatz und in den Spalten die Tempora im Adverbialsatz. Die Zeitformen beinhalten auch die Progressiv-Formen. *Future* enthält alle Zukunftsformen.

	Present	Present Perfect	Past	Past Perfect	Future
Present	0,0	0,0	0,05	0,2	0,25
Pr. Perfect	0,25	0,0	0,0	0,05	0,25
Past	0,25	0,25	0,00	0,0	0,25
Pa. Perfect	0,25	0,25	0,25	0,00	0,25
Future	0,25	0,25	0,25	0,25	0,0

## C. Klassendiagramm

Hier ist das Klassendiagramm des Agenten zu sehen.





## D. Erweiterung des Sprachkorpus

Es wurden zwei Szenarien erstellt, um Sprachaufnahmen zu sammeln, in denen gleiche und ähnliche Aktionen mehrfach genannt werden. Die Szenarien wurden so gestellt, dass die Probanden versuchen sollten, auf bereits genannte Aktionen Bezug zu nehmen, um andere Aktionen zeitlich einzuordnen. Die Probanden wurden über die Projektsituation, das Zielsystem ARMAR-III und die Einsatzumgebung in der Küche aufgeklärt. Das erste Szenario besteht aus zwei Teilen, wobei die Bearbeitung des zweiten Teils erst nach Erstellen der Aufnahme für den ersten Teil erfolgte.

### D.1. Scene XI: Orange juice from the fridge

*This scenario consists of two parts. Please don't read part 2 before you have finished part 1.*

*Part 1: In this scene you want the robot to open the fridge and take the orange juice. Then the robot should go to the kitchen table and put the juice onto it. The robot should take the cup. Figure D.1 shows your view on the scene, with the robot to your left, the table on the right, and the sink in the back. Figure D.2 shows the opened fridge and the kitchen table.*

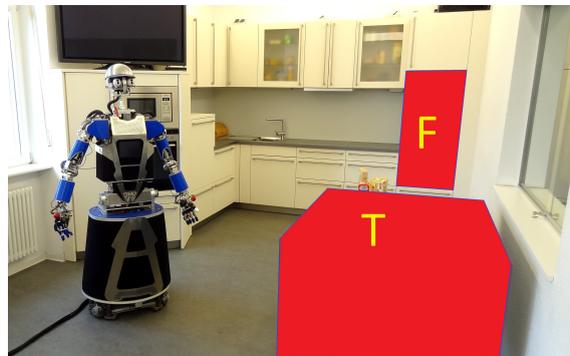


Abbildung D.1.: Kitchen (subject view)

*Part 2: After you have given your instructions to the robot you notice that you have forgotten to tell the robot to close the fridge again. The robot should close the fridge at some point after taking the juice. Tell the robot to close the fridge and tell him at which point to do it.*

### D.2. Scene XII: Window and cup

*In this scene the robot should perform different small tasks. The robot should open and then close the window. Next the robot should go to the dishwasher, take one of the cups and put*



(a) Figure A



(b) Figure B

Abbildung D.2.: Scenery: Fridge and table



(a) Figure A

(b) Figure B

Abbildung D.3.: Scenery: Dishwasher

*it on the kitchen table. He should fill the cup with water. Then he should open the window again. Last he should look at the cup that he has filled earlier.*

*The robot should perform these tasks in that order. Feel free to give the instructions to the robot in a non-chronological order and use words like "before" and "after" to tell the robot when to execute an instruction.*

*Figure 8.3(a) shows your view on the scene, with the robot to your left. The window is marked with W, the dishwasher with D and the kitchen table with T. The opened dishwasher is displayed in figure Figure 8.3(b).*

*Note: The windows are closed in the beginning. The dishwasher is open.*