Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Bastian Probst

Entwurf und Realisierung eines Information Extraction Systems für Formel-1-Berichte mithilfe des Sprachverarbeitungs-Frameworks UIMA

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Bastian Probst

Entwurf und Realisierung eines Information Extraction Systems für Formel-1-Berichte mithilfe des Sprachverarbeitungs-Frameworks UIMA

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Michael Neitzke
Zweitgutachter: Prof. Dr. Gunter Klemke

Eingereicht am: 01. Januar 1970

Bastian Probst

Thema der Arbeit

Entwurf und Realisierung eines Information Extraction Systems für Formel-1-Berichte mithilfe des Sprachverarbeitungs-Frameworks UIMA

Stichworte

Informationsextraktion, Künstliche Intelligenz, Regeln, Reguläre Ausdrücke, UIMA, ConceptMapper, Wörterbuch

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Erstellung eines Prototyps für das Extrahieren von Informationen aus Formel-1-Berichten. Für diese Zwecke wird das Framework UIMA verwendet, mit dem vereinfacht Informationsextraktionsprogramme entwickeln und diese Programme verwaltet werden können. Es werden die grundlegenden Konzepte der Informationsextraktion beschrieben. Auf diese Konzepte aufbauend wurde dann ein dynamischer Prototyp erstellt, mit dem es möglich ist, Datum, Ort, Gewinner, Unfälle und Ausfälle eines Rennens einem Formel-1-Berichten zu entnehmen.

Bastian Probst

Title of the paper

Design and Implementation of an information extraction system for formula 1 reports with the aid of the speech processing framework UIMA

Keywords

Information extraction, artificial inteligence, rules, regular expressions, UIMA, ConceptMapper, Dictionary

Abstract

This Paper is about the creation and development of a prototype for information extraction from formula 1 reports. UIMA, which is used here, is an architecture and software framework for creating, discovering, composing and deploying a broad range of multi-modal analysis capabilities and integrating them with search technologies. Basic concepts of information extraction are explained. A dictionary and an easy to configure rule concept is developed, which uses natural language. With the use of this concept a dynamic prototype is created, which is able to extract dates, locations, winner, crashs and malfunctions from a formula 1 race report. Finally the developed rules are checked and it will be shown, how the adding of rules will change the results.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

1.1 Zur Motivation dieser Arbeit

Durch die zunehmende Verbreitung des Internets ist eine unfassbar große Menge an Daten entstanden. Da es nicht mehr möglich ist, durch personellen Aufwand diese Daten zu katalogisieren, entstand der Bereich des Information Retrieval.
Mittlerweile ist selbst zu speziellen Themenbereichen eine Fülle an Dokumenten verfügbar. Deshalb ist es wünschenswert, Programme zu entwickeln, um maschinell und effizient die essenziellen Daten aus einem Text zu extrahieren. Dies ist der Bereich der Information Extraction (IE).
Diese Technologie verspricht eine Fülle von möglichen Anwendungen: Die automatische Beantwortung von Kundenmails, die Analyse von Börsenberichten oder - moralische Aspekte außer Acht lassend - das Überwachen von E-Mailtexten, Chatrooms, Foren oder anderer Kommunikationsmedien, zum Beispiel auf rechtsstaats-unverträgliche Tendenzen.
Beim IE ist es nicht zwingend erforderlich, textuelle Daten vorliegen zu haben. So kann man theoretisch Audiodaten in Textform umwandeln und diese dann weiter verarbeiten. In meiner Arbeit werde ich mich ausschließlich mit Textdokumenten auseinander setzen. Den Bereich der Audiodaten werde ich hier nicht behandeln, da dieser weitere Problematiken mit sich bringt.
Um die Qualität eines IE Programms zu bewerten, gibt es laut Hasegawa, Sekine und Grishman (2004) folgende Indikatoren: Recall, Precision und F-measure. Beim Recall geht es darum, wie viele der Informationen gefunden werden. Beim Precision wird die Frage geklärt, wie richtig diese Ergebnisse sind. Beim F-maesure werden diese beiden Werte zusammengefasst mit der Formel: 2*Recal*Precision/(Recal+Precision). Somit ist es eine wesentliche Motivation, ein System mit möglichst guten Werten für diese drei Indikatoren zu entwerfen.

1.2 Zur Aufgabenstellung dieser Arbeit

Das Ziel der Arbeit ist die Entwicklung eines Programms zur Extraktion von Informationen aus Formel-1-Berichten mithilfe des Frameworks UIMA. Dabei sollen einem Bericht folgende Informationen entnommen werden: Datum und Ort des Rennens, Gewinner des Rennens, Unfälle des Rennens, deren Beteiligte und - wenn möglich - die Runde des Unfalls.
Von besonderem Interesse sind die Fragen:

- Welche unterschiedlichen Arten an Informationen gibt es?
- Wie kann man diese kategorisieren?
- Was sind die besten Methoden, eine solche İnformationsartäus einem Text herauszuziehen?

Dabei ist es wichtig, möglichst effizient zu sein, um optimale Ergebnisse im Bereich von Recall und Precision zu erreichen.
Ein weiteres Ziel ist, das Programm möglichst dynamisch zu gestalten, so dass im besten Fall ein anpassen des Wörterbuchs und der Regeln ausreicht, um weitere Informationen aus dem Text zu ziehen. Bei der Entwicklung der Arbeit hat sich dieser Anspruch dahingehend erweitert, dass im Idealfall die Anpassung des Wörterbuchs und der Regeln ausreicht, um das in dieser Arbeit entwickelte Programm auch für andere Textarten anwendbar zu machen.
Bei der Entwicklung des Programms wird auf UIMA zurückgegriffen. UIMA ist eine Architektur, die es einem ermöglicht IE Anwendungen zu erstellen und diese zu verwalten.

1.3 Der Aufbau der Arbeit

Das erste Kapitel widmet sich der Motivation und der Aufgabenstellung, die dieser Arbeit zugrunde liegen.
Das zweite Kapitel beschäftigt sich mit den Grundlagen, die für diese Arbeit benötigt werden. Dazu zählt eine Einführung in den Bereich der IE. Hier wird darüber informiert, welche Methoden und Ansätze es gibt, um Informationen aus einem Text zu ziehen, und welche Techniken dabei eingesetzt werden. Da in dieser Arbeit das Framework UIMA verwendet wird, beschreibt ein wesentlicher Teil dieses Kapitels dieses Framework und dessen implementierungsspezifische Techniken.

Im dritten Kapitel wird analysiert, welche Techniken und Methoden verwendet werden können, um die gewünschten Informationen aus einem Text zu ziehen. Besonderes Augenmerk wird darauf gelegt, welche Informationsarten es gibt und welche Methoden sich besonders eignen, diese Informationen aus dem Text zu extrahieren. Desweiteren wird betrachtet, welche Skriptsprachen es gibt, die zum Definieren der Regeln benutzt werden können. Zum Abschluss wird eine alternative und selbst entwickelte Regelsprache vorgestellt und genauer betrachtet.
Im vierten Kapitel wird vorgestellt, wie das Design des Programms aufgebaut ist. Es wird aufgezeigt, in welcher Art und Weise die in der Analyse herausgearbeiteten theoretischen Extraktionsmethoden praktisch realisiert worden sind. Die von der UIMA mitgebrachten Tools werden hier näher beleuchtet. Es wird erläutert, wie mit deren Hilfe ein dynamisches Programm realisiert worden ist.
Im fünften Kapitel werden wichtige Aspekte der Realisierung erläutert. Es werden Implementierungen der Extraktionsmethoden und deren Stärken und Schwächen beschrieben. Ebenfalls werden einige konkrete Implementierungsdetails gezeigt, die das dynamische Verhalten des Programmes garantieren.
In den letzten beiden Kapiteln wird die Entwicklungsphase der Regeln genauer betrachtet. Die Regeln werden definiert, und es wird aufgezeigt, wie sich dadurch Recall und Precision verändern. Am Ende befindet sich eine genaue Analyse der Tests und Testergebnisse. Abschließend wird den Fragen nachgegangen, von welchem Nutzen ein solches Extraktionssystem sein kann, was man hätte anders machen können und wie dieses Programm weiter entwickelt werden könnte.

2 Grundlagen

In diesem Kapitel wird ein Überblick über aktuelle Techniken und Methoden der Information Extraction gegeben. Es werden grundsätzliche Konzepte erläutert und deren Vor- und Nachteile aufgezeigt. Diese bilden die Grundsteine der Arbeit, da sie im Wesentlichen verwendet werden, um die zugrundegelegten Anforderungen zu realisieren. Da in dieser Arbeit reguläre Ausdrücke eine entscheidende Rolle spielen, werden sie hier genauer erklärt, und es wird eine Übersicht über deren Darstellungsmöglichkeiten gegeben.
Im letzten Teil dieses Kapitels geht es um das verwendete Framework selbst und dessen Komponenten. Besonders die Komponenten, die in dieser Arbeit Verwendung finden, werden hier näher beleuchtet.

2.1 Information Extraction

Bei Grishman ist dazu folgendes zu lesen:
"*The identification of instances of a particular class of events or relationships in a natural language text, and the extraction of the relevant arguments of the event or relationship*"Grishman (1997)
Aus dieser Definition folgt, dass die aus dem Text extrahierten Informationen in strukturierter Form präsentiert werden. Informationen aus einem Text auf tabellarische Darstellung zu reduzieren, ist eine Idee, die nach Grishman nicht neu ist. Sie stammt ihm zufolge von Zelling Harris aus den 1950er Jahren.Grishman (1997)
Bei Appelt und Israel sind die Schritte, welche von einem IE (Information Extraction)-System durchgeführt werden müssen, in Abbildung 2.1 beschrieben:

2.1.1 Die Tokenisierung

Bei der Tokenisierung wird der Text erst in Sätze und dann in Tokens aufgeteilt. Tokens sind in der Regel einzelne Wörter oder Zahlen. Bei der Tokenisierung handelt es sich laut Appelt und Israel um ein triviales Problem der Informationsextraktion:

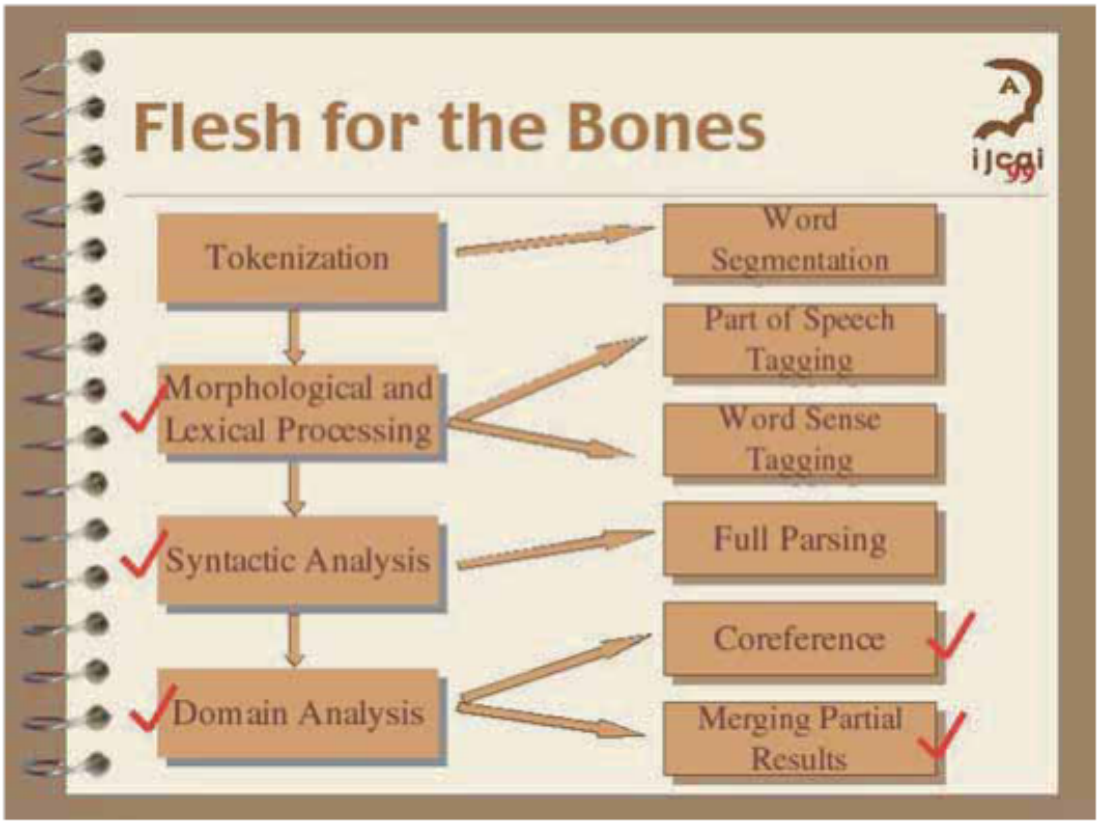


Abb. 2.1: Abbildung 2.1: Schritte eines IE-Programms Appelt und Israel (1999)

„Tokenization is a trivial problem for European languages, requiring only that one separate whitespace characters from nonwhitespace characters.“Appelt und Israel (1999) Demzufolge können einzelne Wörter anhand der Leerzeichen, die sich zwischen ihnen befinden, identifiziert werden. Dies gilt zumindest für europäische Sprachen. Da in dieser Arbeit ausschließlich deutsche Texte verwendet werden, werden hier Lösungen von Sprachen, in denen Wörter nicht einfach durch Leerzeichen zu differenzieren sind nicht weiter betrachtet. So wären zum Beispiel bei asiatischen Sprachen weitaus komplexere Lösungen erforderlich.

2.1.2 Die morphologische und lexikalische Bearbeitung

Bei der morphologischen Analyse werden die Wörter in ihre Bedeutungen zerlegt. Das führt dazu, dass zusammengesetzte Wörter zerlegt werden. Ein Beispiel hierfür wäre

„Regenrennen“. Es handelt sich hierbei um zwei zusammengesetzte Nomen. Das eine ist „Regen“, das andere „Rennen“. Bei der morphologischen Bearbeitung werden solche Zusammengesetzten Wörter aufgelöst. Nach Appelt und Israel Appelt und Israel (1999) hat in Sprachen wie Englisch eine morphologische Analyse keinen großen Nutzen. In Sprachen wie Deutsch ist es aber erforderlich.
Bei der lexikalischen Analyse werden die Wörter kategorisiert (Part of Speech Tagging) und deren Bedeutung festgelegt (word sense tagging). Es werden für jedes Wort des Textes sowohl die Art als auch die Eigenschaften festgelegt. Die Art eines Wortes kann zum Beispiel „Nomen“ oder „Verb“ sein. Die Attribute eines Nomens können „das Genus“, „der Numerus“ und „der Kasus“ sein. Hat das Wort eine bestimmte Bedeutung - wie zum Beispiel ein Datum - wird dieses ebenfalls identifiziert.

2.1.2.1 Das Part of Speech Tagging

Bei dem „Part of Speech Tagging“ werden die Tokens, die bei der Tokenisierung identifiziert wurden, mit Wörterbucheinträgen verglichen. Diesen Wörterbucheinträgen werden dann Wortart und Attribute entnommen. Wenn es für ein Wort mehrere Bedeutungen geben kann, werden entweder Regeln oder Modelle auf Basis von Wahrscheinlichkeiten angewendet, um die in diesem Kontext gemeinte Bedeutung zu identifizieren.
Ein Beispiel für eine Doppeldeutigkeit ist: „fest“, welches als Nomen (Für ihn war das Rennen ein „Fest“) aber auch als Adjektiv (Seine Lenkung klemmte „fest“) verwendet werden kann. Eine Regel hierfür könnte lauten: „folgt das Wort „fest“ auf einen unbestimmten Artikel, dann ist das Nomen gemeint. Folgt das Wort „fest“ auf ein Verb, dann ist das Adjektiv gemeint. Bei einem Modell, welches auf Basis von Wahrscheinlichkeiten arbeitet, wird anhand eines annotierten Referenztextes die wahrscheinlichste Wortart gesetzt.
Appelt und Israel Appelt und Israel (1999) stellten fest, dass „Part of Speech Tagging“-Verfahren sehr rechenaufwendig sind und eine maximale Korrektheit von 95% erreichen. Ferner arbeiteten sie eine Alternative heraus: man verweigert einfach Algorithmen, die eine seltene Bedeutung von einem Wort verwenden, wenn es einen Algorithmus gibt, der eine häufiger auftretende Wortbedeutung verwendet. Appelt und Israel (1999) bewerteten diese Methode als schnell und effizient und als einen guten Kompromiss zwischen Genauigkeit und Effizienz.

2.1.2.2 Das Word Sense Tagging

Spezielle Wortbedeutungen können zum Beispiel ein Datum oder eine Uhrzeit sein. Ein Datum zum Beispiel liegt immer in einer bestimmten Form vor. Es gibt aber trotzdem sehr viele unterschiedliche Daten. Appelt und Israel Appelt und Israel (1999) arbeiteten heraus, dass hier Wörterbucheinträge nicht sinnvoll sind. Vielmehr sind Daten über ihre Struktur zu identifizieren. Ein ähnliches Beispiel hierfür ist die Namenserkennung.
Bei der Namenserkennung werden Eigennamen identifiziert und klassifiziert. Sie werden über die Wortart, syntaktische Funktion und orthografischen Merkmale identifiziert. Personennamen können laut Grishman Grishman (1997) zum Beispiel durch die Anrede (Herr Probst), einen alltäglichen Vornamen (Bastian Probst), eine Endsilbe (Bastian Probst Jr.) oder durch mittlere Initiale (Bastian R. Probst) identifiziert werden. Im Bereich dieser Arbeit ist diese Aufgabe ein wenig leichter, da die Namen der beteiligten Personen im Großen und Ganzen bekannt sind. Ebenfalls ist es wichtig Aliase richtig zuzuordnen. So sollte die Namensanalyse leisten, dass „Herr Probst“ und „Bastian Probst“ für ein und dieselbe Person gehalten wird. Gleiches gilt für Orts- und Teamnamen. Sarawagi (2008)

2.1.3 Die syntaktische Analyse

Die syntaktische Analyse vereinfacht die Extraktion der Fakten und Events. Diese entsprechen im Großen und Ganzen den Beziehungen der Nominalphrasen eines Textes. Da diese Aufgabe eine recht große Herausforderung darstellt, gibt es hier eine Vielzahl unterschiedlicher Ansätze. Einige Systeme führen keine separate syntaktische Analyse durch, andere versuchen komplette Sätze zu parsen.Cowie und Wilks (1996)
Laut Appelt und Israel Appelt und Israel (1999) ist es sinnvoll, sich auf die Kernbestandteile der Sätze zu beschränken. Unter dieser Voraussetzung lassen sich endliche Automaten verwenden, um die Sätze zu parsen. Wenn man sich auf Namen und Verben beschränkt, ist es sehr leicht möglich, in einem Satz herauszufinden, wer was tut. Nimmt man zum Beispiel den Satz: „Fernando Alonso gewinnt das spektakuläre Regenrennen von Malaysia, mit einem bei der Zieldurchfahrt nicht mehr so sauber glänzenden Ferrari, der noch beim Start im schönsten Rot schimmerte.“. Wenn dieser Satz auf Namen und Verben reduziert wird, bleibt folgendes übrig: „Fernando Alonso gewinnt glänzenden schimmerte“. In diesem Satz ist es einfacher und mit deutlich weniger komplizierten Regeln herauszufinden, dass Fernando Alonso dieses Rennen gewonnen hat.

Ein weiterer Trick nach Appelt und Israel Appelt und Israel (1999) zum Extrahieren von Informationen ist das Beschränken auf domänenrelevante Satzbestandteile. Entfernen wir nun in dem vorherigen Beispiel noch die domänenirrelevanten Bestandteile: „glänzenden“ und „schimmerte“, dann bleibt folgendes übrig: „Fernando gewinnt“. Eine simple Regel könnte hier lauten: stehen in einem Satz ein Name und ein Verb, dann tätigt die Person mit dem Namen, was das Verb aussagt.
Es reicht nicht aus, einen Text auf Namen und Verben zu beschränken. Aber man sollte sich Gedanken machen, welche Satzbestandteile für eine Domäne und die gewünschten Informationen erforderlich sind.
Appelt und Israel Appelt und Israel (1999) stellten fest, dass eine derartige Reduzierung des Ursprungtextes notwendig ist, da eine vollständige syntaktische Analyse einen viel zu hohen Aufwand und viel zu viele Fehler mit sich bringt.

2.1.4 Die Domänenanalyse

Die Domänenanalyse besteht aus der Koreferenzanalyse und dem Zusammenführen von Teilergebnissen. Bei der Koreferenzanalyse werden zusammengehörende Entitäten als ein und dieselbe Entität identifiziert und bei dem Zusammenführen der Teilergebnisse werden die Teilergebnisse analysiert und zusammengeführt.
Entitäten sind eindeutig identifizierbare Objekte. Eine Entität steht für genau ein Objekt. Eine Entität kann zum Beispiel ein Fahrer sein. So sind zum Beispiel „Bastian Probst“ und „Herr Probst“ dieselbe Entität. Zumindest wenn mit „Herr Probst“ auch „Bastian Probst“ gemeint ist.

2.1.4.1 Die Koreferenzanalyse

Bei der „Koreferenzanalyse“ werden anaphorische Abhängigkeiten aufgelöst, wie zum Beispiel Pronomen. Das Bedeutet, dass sich zwei unterschiedliche Bezeichnungen sich auf dieselbe Entität beziehen. Auf diese Weise können nicht identifizierte Entitäten konkreten Personen zugewiesen werden. Ein einfaches Beispiel hierfür ist, dass „Herr Probst“ und „Bastian Probst“ dieselbe Person meinen. Eine weitere Problematik ist das Auflösen von Pronomen. Baldwin (1997)
Generell werden bei einer Koreferenzanalyse alle möglichen Kandidaten für eine nicht identifizierte Entität gesucht und dann analysiert. Die wahrscheinlichste oder eindeutig identifizierte Entität wird dann gewählt. So wird „Alonso“ in einem Formel 1- Bericht wohl immer „Fernando Alonso“ sein. „Nico“ hingegen kann entweder „Nico Rossberg“

oder „Nico Hülkenberg“ sein. Wenn in dem Satz davor aber von „Nico Rossberg“ die Rede ist, dann ist zu erwarten, dass „Nico Rossberg“ gemeint ist.Bagga und Biermann (2000)

2.1.4.2 Zusammenfassen der Teilergebnisse

Beim Zusammenfassen der Teilergebnisse werden die gefundenen Tätigkeiten und Beziehungen in die vom Anwender gewünschten Informationen umgewandelt. Hier wird unter Anderem berücksichtigt, dass für einige Dinge nur eine bestimmte Anzahl an Personen möglich ist. Ein Beispiel hierfür ist der Gewinner eines Rennens. Es gibt nur genau eine Person, die ein Rennen gewinnen kann. Wenn man dreimal eine Aussage gefunden hat, dass Lewis Hamilton das Rennen gewonnen hat, und einmal, dass Michael Schumacher das Rennen gewonnen hat, dann ist davon auszugehen, dass Lewis Hamilton der Gewinner dieses Rennens ist und Michael Schumacher fälschlicher Weise gefunden wurde.Appelt und Israel (1999)
Ebenfalls gilt es hier, die Informationen in die Darstellungsform zu bringen, die von Anwender gewünscht ist. Wird zum Beispiel die Anzahl der Siege von Michael Schumacher gesucht, können die einzelnen Rennen betrachtet werden und die gezählt werden, die Michael Schumacher gewonnen hat.

2.2 Reguläre Ausdrücke

Ein regulärer Ausdruck ist ein Pattern, welches eine Menge an Zeichenketten definiert. Er kann verwendet werden, um Teilstrings aus einem String herauszuholen. In dieser Arbeit werden sie verwendet, um Datumsangaben zu identifizieren und Rundenangaben im Text zu finden. Ein String ist eine Zeichenkette, welche aus einem oder mehreren Buchstaben und Sonderzeichen bestehen kann.
Es existiert eine ganze Menge an Regeln für reguläre Ausdrücke. Im Prinzip sind nahezu alle Zeichen erlaubt. Jedoch haben einige Sonderzeichen, wie zum Beispiel der Punkt, besondere Bedeutungen. Ein vorangestelltes „/“ maskiert die Sonderzeichen aus, was für Zeichen wie den Punkt von Bedeutung ist. Folgende Regeln werden bei Ullenboom (2012) beschrieben: Ullenboom (2010)
Um die Anzahl an Wiederholungen zu bestimmen, gibt es Quantifizierer:

Quantifizierer	Anzahl an Wiederholungen
X?	X kommt einmal oder keinmal vor
X*	X kommt kein mal oder beliebig oft vor
X+	X kommt einmal oder beliebig oft vor

Um zum Beispiel alle Buchstaben des Alphabets einzusetzen, also einen Bereich von Zeichen, gibt es die Möglichkeit, Zeichenklassen zu definieren.

Zeichenklasse	Enthält
[aeiou]	Zeichen a, e, i, o oder u
[^aeiou]	Nicht die Zeichen a, e, i, o, u
[0-9a-fA-F]	Zeichen 0,1,2,...,9 oder Groß-/ Kleinbuchstaben a, b, c, d, e, f

Zum Ersparen der Schreibarbeit gibt es vorgefertigte Zeichenklassen:

Zeichenklasse	Enthält
.	Jedes Zeichen
\d	Ziffer: [0-9]
\D	Keine Ziffer[^0-9]
\s	Weißraum [\t\n\x0B\f\r]
\S	Kein Weißraum
\w	Wortzeichen [a-zA-Z_0-9]
\W	Kein Wortzeichen [^ \w]

Darüber hinaus gibt es noch Regeln, die bestimmen, an welcher Stelle im Text sich die Zeichenkette befinden muss. Beispiele hierfür sind der Anfang einer Zeile oder eine Wortgrenze.

Matcher	Bedeutung
^	Beginn einer Zeile
$	Ende einer Zeile
\b	Wortgrenze
\B	Keine Wortgrenze
\A	Beginn der Eingabe

Ullenboom (2010) Durch diese Regeln wird klar, was mit dem Regulären Ausdruck für das Datum gemeint ist:

“\\b[0-3][0-9]\\. [01][0-9] 2010“

Eine Unstructured Information Management Architecture (UIMA) ist eine Architektur und Software zum Kreieren, Auffinden, Entwerfen und Einsetzen einer Vielfalt unterschiedlichster Analysetools. Diese können in Suchtechnologien integriert werden. Das „Apache UIMA Framework“ ist ein von Apache lizenziertes Open-Source-Framework der UIMA Architektur und stellt eine Laufzeitumgebung zur Verfügung. Diese ermöglicht Entwicklern, ihre eigenen UIMA Komponenten hinzuzufügen, womit sie UIM Applikationen entwerfen und einsetzen können.
Die wichtigsten Komponenten bei UIMA sind Annotatoren und Type Systems. Im Prinzip werden ein Annotator, ein aus mehreren Annotatoren zusammengesetzter Annotator oder mehrere Annotatoren gestartet, welche dann Annotationen auf Basis der zugehörigen Type Systems erzeugen. Sowohl die Annotatoren als auch die Type Systems werden in XML definiert. UIMA liefert hier aber mit dem Component Descriptor Editor ein Tool, welches die Erstellung von sowohl Annotatoren als auch Type Systems deutlich vereinfacht.
Wesentliche Bestandteile der Programmlogik, die nicht weiter verändert werden, sind die Analysis Engine und die Common Analysis Structure. Der Analysis Engine wird beim Erstellen der Annotatoren mitgegeben. Dieser hat ebenfalls Referenzen auf weitere in ihm enthaltene Annotatoren und Type Systems. Mithilfe der Analysis Engine wird die Common Analysis Structure erstellt. In ihr werden alle Annotationen und Typen festgehalten. Folgende Abbildung 2.2 veranschaulicht und macht deutlich, welche der Komponenten vom Entwickler erstellt und welche von UIMA zur Verfügung gestellt werden.
Der Entwickler muss bei einer Analyse-Komponente eine XML-Datei zur Beschreibung und eine Java Klasse erstellen, in der die konkreten programmtechnischen Ausführungen realisiert werden. Die Analysis Engine erstellt dann anhand des Deskriptors eine Common Analysis Structure (=CAS). Die CAS kann daraufhin von dem Entwickler im erstellten Programm über einen Controller verwendet werden, um Dokumente zu analysieren und die Analyseergebnisse auszuwerten. Zur Analyse eines Dokumentes wird die Process-Methode verwendet, welche der Entwickler in seiner Annotator-Klasse realisieren muss.Foundation (2011)

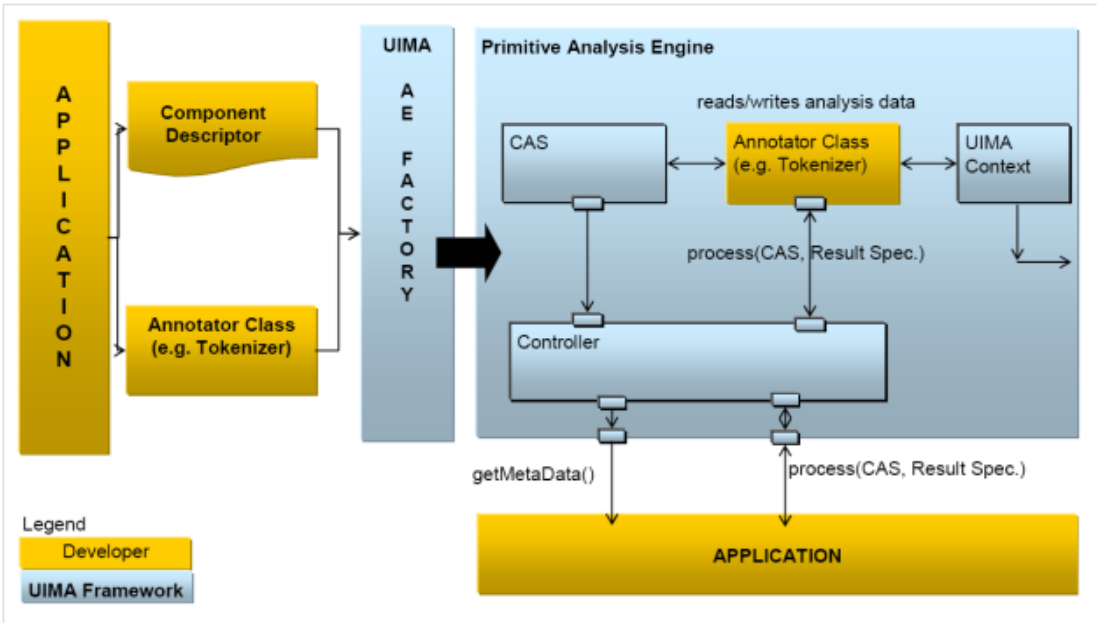


Abb. 2.2: Abbildung 2.2: Übersicht der UIMA Architektur Foundation (2011)

2.2.1 Type System

Im Type System werden die Eigenschaften einer Annotation festgelegt. In der Regel erbt eine selbsterstellte Annotation von der UIMA-Klasse Annotation. Die Abbildung 2.3 zeigt, wie man beim Component Descriptor Editor die Eigenschaften des Datumstypen festlegen kann.
Der Abbildung 2.3 ist zu entnehmen, dass es Variablen für den Tag, den Monat und das Jahr gibt, die jeweils durch Integer-Werte repräsentiert werden. Das von UIMA mitgelieferte Tool JCasGen bietet die Möglichkeit, anhand der XML Definitionen die entsprechenden Java-Klassen automatisch zu erstellen. Diese stellen dann Getter- und Setter-Methoden zur Verfügung, um die Variablen zugreifbar zu machen.

2.2.2 Annotator

Ein Annotator ist die Komponente, in welcher die Art der Annotation realisiert wird. Der Annotator besteht aus zwei Teilen:

- einer XML Datei, dem Component Descriptor;

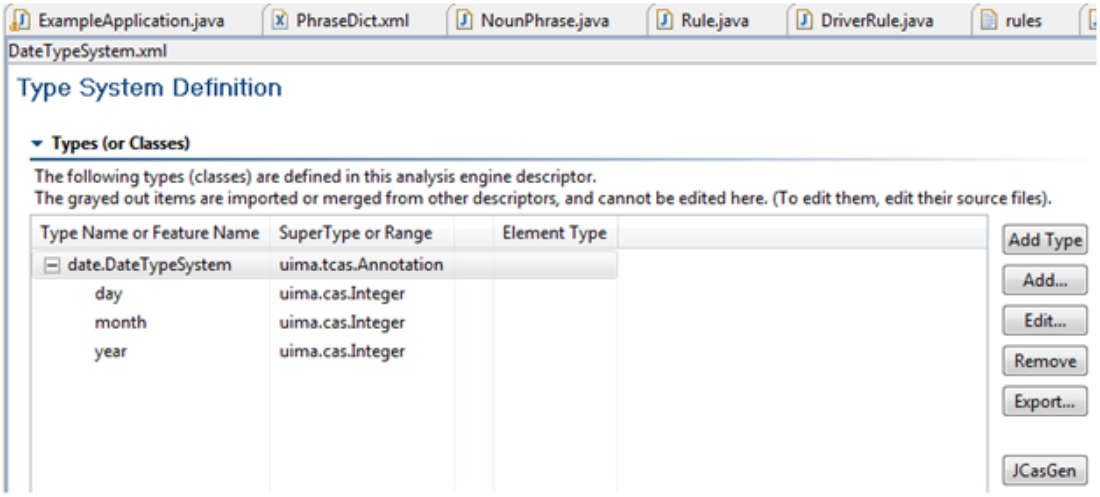


Abb. 2.3: Abbildung 2.3: Typdefinitionen bei UIMA Foundation (2011)

- einer Java-Klasse, der Annotator Class.

In der Component Descriptor Datei wird genau festgelegt, welche Annotationen der Annotator als Input verwendet und welche er als Output wieder zurückliefert. So hat zum Beispiel der Datumsannotator keinen Input und liefert Datumsannotationen als Output, wohingegen der Annotator des Wörterbuches Tokens als Input verlangt und Wörterbuch Annotationen als Output ausgibt.
In der Annotator Klasse wird das Verhalten des Annotators spezifiziert. Hier ist besonders die Process-Methode zu betrachten, welche von der UIMA zum Analysieren des Dokumentes verwendet wird. Die Process-Methode wird mit einer JCas, einer für Javaprogramme optimierten Common Analysis Structure aufgerufen. Diese JCas wird durch die Analysis Engine mithilfe des Component Descriptors erstellt, wie im vorherigen Absatz beschrieben. Es besteht nun die Möglichkeit, sich den kompletten Dokumententext oder die als Input deklarierten Annotationen aus der jCas zu holen. Beim Datumsannotator wird zum Beispiel der komplette Text geholt. Dies sieht programmtechnisch wie folgt aus:

```
1 // get document text
2 String docText = aJCas.getDocumentText();
```

Beim Wörterbuchannotator werden die Annotationen aus der jCas geholt. Dies ist dann im Programm auf folgende Weise realisiert:

```
FSIterator<AnnotationFS> it = tcas.getAnnotationIndex(type).iterator();
```

Die Methode „getAnnotationIndex(type)“ liefert eine Liste an Annotationen zurück, welche dem Typ „type“ entspricht. Wird diese Methode parameterlos aufgerufen, dann liefert sie alle Input-Annotationen zurück. Mit der Methode „iterator()“ wird dann ein Iterator erstellt, welcher Schritt für Schritt die gefundenen Annotationen durchgeht. Dieser Iterator wird zum Auffinden bestimmter Tokens benutzt, welche im Wörterbuch enthalten sind.
Wenn eigene Annotationen hinzugefügt werden sollen, muss ein neues Annotationsobjekt erstellt werden. Diesem Annotationsobjekt müssen dann die entsprechenden Attribute gesetzt werden. Abschließend muss diese Annotation dann zum Annotationenindex hinzugefügt werden. Bei der Datumsannotation sieht die Realisierung im Programm dann zum Beispiel so aus:

```
//Datumsannotation erstellen und ihr die jCas bei der Erstellung übergeben
DateTypeSystem annotation = new DateTypeSystem(aJCas);
//Die Attribute der Annotation setzen
annotation.setBegin(matcher.start());
annotation.setEnd(matcher.end());
//hier wird der Substring für den Tag im Originaltext repräsentiert geholt
String day = docText.substring(matcher.start(), matcher.start() + 2);
annotation.setDay(Integer.parseInt(day));
//hier wird der Substring für den Monat im Originaltextrepräsentiert geholt
String month = docText.substring(matcher.start() + 3, matcher.start() + 5);
annotation.setMonth(Integer.parseInt(month));
//hier wird der Substring für das Jahr im Originaltextrepräsentiert geholt
String year = docText.substring(matcher.start() + 6, matcher.start() + 10);
annotation.setYear(Integer.parseInt(year));
//hier wird der Annotation gesagt, dass sie sich zum Index hinzufügen soll
annotation.addToIndexes();
//alternativ könnte man auch über den jCas die Annotation hinzufügen:
//-> aJCas.addFsToIndexes(annotation);
```

2.2.3 Der ConceptMapper

Der ConceptMapper ist eine UIMA Komponente. Diese Komponente ermöglicht es, Wörterbucheinträge im Text zu finden und anhand der Wörterbucheinträge Annotationen zu erstellen. Es wurde besonders auf eine hohe Skalierbarkeit und eine gute Performance geachtet.
Der ConceptMapper bietet unterschiedliche Suchstrategien. In dieser Arbeit wird sich auf die einfachste beschränkt, bei der aufeinander folgende Tokens gesucht werden. Die Wörterbucheinträge können aus mehreren Tokens bestehen. Die Wörterbucheinträge

werden im Hauptspeicher abgelegt. Dies hat zur Folge, dass die Performance kaum beeinträchtig wird. Das gilt selbst für Wörterbücher mit mehreren Millionen Einträgen. Beachtet werden muss dabei, dass genügend Speicher vorhanden ist, da – wie bereits erwähnt – die Einträge alle im Hauptspeicher gehalten werden.
Die Wörterbucheinträge haben variable Attribute. Das bedeutet aber nicht, dass willkürlich Attribute im Wörterbuch definiert werden können. Zunächst muss ein Annotator erstellt werden, welcher diese Attribute ebenfalls implementiert hat. Diese Attribute müssen sowohl in der FeatureList als auch in der AttributeList deklariert werden. Die AttributeList spiegelt die Attribute wieder, welche im Wörterbuch den Einträgen zugeordnet werden können. Weitere Bedingungen sind zum einen, dass die Einträge der AttributeList mit den Einträgen der FeatureList übereinstimmen, und zum anderen, dass alle diese Einträge nur vom Typ „String“ sind. Es ist also nur möglich, Wörterbuchattribute als „Strings“ wieder zu geben.Foundation (2011)
In der Methode:

```
protected void makeAnnotation(CAS tcas, int start, int end,
        EntryProperties properties, Annotation spanAnnotation,
        String matchedText, Collection<AnnotationFS> matched, Logger log)
```

wird die Annotation erstellt. Wenn man also eigene Annotationen einbauen oder dem ConceptMapper ermöglichen möchte, mehrere unterschiedliche Annotationen zu erstellen, ist dies die Stelle, wo derartiges möglich ist. So kann man, ein Attribut der zu erstellenden Annotation abfragen und - in Abhängigkeit von diesem - eine neue Annotation erstellen. Dies sieht am Beispiel der EntityAnnotation wie folgt aus:

```
PhraseAnnotation phraseAnnotation = ((PhraseAnnotation) annotation);
                //Add Driver Annotation
if(ProgrammConstants.ENTITY.equals(phraseAnnotation.getKind())){
        EntityAnnotation entity = new EntityAnnotation(jcas);
        entity.setId(id++);
        entity.setBegin(phraseAnnotation.getBegin());
        entity.setEnd(phraseAnnotation.getEnd());
        entity.setCase(phraseAnnotation.getCase());
        entity.setGender(phraseAnnotation.getGender());
        entity.setName(phraseAnnotation.getName());
        entity.setTeam(phraseAnnotation.getTeam());
        entity.setNumber(phraseAnnotation.getNumber());
        SentenceAnnotation sAnnot = g    etSentenceAnnotation(entity.getBegin());
        FSArray roundList = sAnnot.getPhrases();
        if (roundList == null) {
                roundList = new FSArray(jcas, 1);
                roundList.set(0, entity);
        } else {
                FSArray helper = new FSArray(jcas, roundList.size() + 1);
```

```
                int i;
                for (i = 0; i < roundList.size(); i++)
                        helper.set(i, roundList.get(i));
                helper.set(i, entity);
                roundList = helper;
        }
        sAnnot.setPhrases(roundList);
        tcas.getIndexRepository().addFS(entity);
}
```

In diesem Beispiel wird die EntityAnnotation erstellt. Ihre Werte werden von der durch den ConceptMapper erstellten PhraseAnnotation gesetzt. Auf diese Weise werden alle Nominal-, Verbal- und Signalphrasenannotationen erstellt. Dies wird in Kapitel 4 und 5 genauer beschrieben.

```
<token canonical="Michael Schumacher" kind="Entität" name="Michael Schumacher"
        team="Mercedes GP Petronas F1 Team"
        case="nom,dat,akk" gender="male" number="sg">
        <variant base="Michael Schumacher" />
        <variant base="Michael" />
        <variant base="Schumi" />
        <variant base="Schumacher" />
        <variant base="der Spa-Rekordsieger" />
</token>
```

3 Analyse

In diesem Kapitel wird zunächst die Frage geklärt, welche Informationen aus einem Formel 1-Bericht gezogen werden sollen. Diesbezüglich wird betrachtet, wie man Informationen kategorisieren kann. Ist dies geklärt, wird überlegt, welche Methoden anzuwenden sind, um diese Informationen aus einem Text heraus zu ziehen. Es wird ein kurzer Einblick in vorhandene Methoden der Verwendung von Regeln gewährt und dann eine selbstentwickelte Regelsprache vorgestellt. Die dieser Regelsprache zugrunde liegenden Überlegungen werden dargestellt und die von ihr verwendeten Typen näher erläutert.
Als Grundlage für diese Analyse wurden alle Formel 1 Rennberichte des Jahres 2010 von den Seiten „formel1.de“, „spox.de“und „auto-motor-und-sport.de“ betrachtet. Deshalb beschränkt sich die Anzahl der unterschiedlichen Grand Prix auf 19. Insgesamt fuhren 28 unterschiedliche Fahrer aus 12 unterschiedlichen Teams in der Saison bei den Rennen mit.
Die vorliegende Arbeit beschreibt, wie folgende Informationen aus einem Formel 1-Bericht extrahiert werden können:

- Datum des Rennens
- Ort des Rennens
- Gewinner des Rennens
- Unfälle und die beteiligten Fahrer sowie
- Ausfälle.

3.1 Welche Informationenarten gibt es?

Zunächst ist die Frage zu klären, welche Informationen überhaupt aus einem Text entnommen werden können. Hierzu ist bei Feldman und Sanger zu lesen: *„There are four basic types of elements that can, at present, be extracted from text“*.Feldman und Sanger (2006) Laut Feldman und Sanger sind dies folgende:

- Entitäten. Sie sind die Basisbausteine eines Textes, zum Beispiel: Fahrer, Rennställe, Grandprix-Name und Rennorte.
- Attribute. Dies sind Eigenschaften der Entitäten. Einige Beispiele sind Weltmeistertitel, Rennsiege und teaminterne Positionen.
- Fakten. Das sind Beziehungen, die zwischen den Entitäten bestehen. Hierzu zählt zum Beispiel das Arbeitsverhältnis zwischen einem Fahrer und seinem Team.
- Ereignisse. Ein Ereignis ist die aktive oder passive Beteiligung einer oder mehrerer Entitäten an einem Geschehen.

Es stellt sich die Frage, um welche Informationsarten es sich bei den hier gestellten Anforderungen handelt.
Beim Datum handelt es sich um eine Entität. Es ist ein Basisbaustein des Textes und kann in einer Beziehung zu anderen Entitäten stehen. So kann ein Rennen zum Beispiel in einer „fand statt“-Beziehung zu einem Grand Prix stehen; oder in einer „Tag der Geburt“-Beziehung zu einem Fahrer. Bei dem Ort handelt es sich ebenfalls um eine Entität. Der Ort des Rennens ist ein Basisbaustein des Textes und kann zum Beispiel zum Rennen in einer „fand statt“-Beziehung stehen.
Eine andere Sichtweise wäre es, Ort und Datum eines Rennens als Attribute des Rennens zu sehen. Es werden zunächst alle genannten Orte und Daten identifiziert .Dann wird durch einen – wenn auch sehr einfachen – Algorithmus die zwischen ihnen und dem Grandprix bestehende Beziehung gesucht. Daraus folgt, dass das Datum und der Ort selbst zwar in dieser Arbeit als eine Entität betrachtet werden. Geht es jedoch darum, Datum und Ort eines Rennens herauszufinden, so wird dies als ein Fakt angesehen.
Beim Gewinner eines Rennens handelt es sich um einen Fakt, da hier eine „gewinn“-Beziehung zwischen einer Entität, genauer einem Fahrer, und einem Grandprix besteht. Die Fahrer und der Grandprix selbst sind somit als Entitäten zu betrachten. Die Handlung des Gewinnens wäre als Ereignis zu sehen, aber die Tatsache, dass ein Fahrer einen Grandprix gewonnen hat, ist als Fakt zu betrachten. Ein simples Beispiel verdeutlicht diese Aussage. Ein Fahrer gewinnt einen Grandprix, wird im Nachhinein aber disqualifiziert. In diesem Fall gab es das Ereignis des Gewinnens. Dieses Ereignis hat stattgefunden. Trotzdem stehen das Rennen und der Fahrer nicht in einer „hat gewonnen“-Beziehung.

Unfälle und Ausfälle werden als Ereignisse betrachtet. Hierbei handelt es sich um Geschehnisse, bei denen eine oder mehrere Entitäten aktiv oder passiv beteiligt waren und die zu einem bestimmten Zeitpunkt stattgefunden haben.

3.2 können die Informationen aus dem Text extrahiert werden?

Als erstes muss die Frage geklärt werden, welche Textbausteine benötigt werden, um die gewünschten Informationen aus dem Text zu ziehen. Benötigte Entitäten sind Daten, Orte, Fahrer, Pronomen, Synonyme für Fahrer und Rundenangaben. Um die Beziehungen zwischen den Entitäten herauszufinden, werden Verben benötigt. Ebenfalls können einzelne Wörter oder Satzteile ein Ereignis signalisieren. Kapitel 3.3 ist den Textbausteinen und ihrer Identifizierung gewidmet. In Kapitel 3.4 wird ein Konzept eines Wörterbuches erarbeitet. Regelsprachen werden in Kapitel 3.5 betrachtet. Und in 3.6 wird eine für das Wörterbuchkonzept am besten passende Regelsprache entwickelt.

3.3 Die Textbausteine

Ein Textbaustein ist ein für die Regeln wesentlicher Teil des Textes, der nicht weiter aufgeteilt werden kann. Es sind also die atomaren Bausteine, die von den Regeln verwendet werden. Ein Textbaustein kann aus einem oder mehreren Wörtern bestehen. Wichtig ist nur, dass der Textbaustein nicht weiter zerlegt werden kann. In diesem Kapitel wird zunächst geklärt, welche Textbausteine zum Erfüllen derInformationsextraktionaufgaben benötigt werden. Danach wird überlegt, wie die Textbausteine im Text identifiziert werden können.

3.3.1 Welche Textbausteine werden benötigt?

Folgende Textbausteine werden als notwendig zum Lösen der gestellten Anforderungen identifiziert:

- Das Datum. Ein Datum ist eine spezielle Entität, die nicht mit den anderen im Text gefundenen Entitäten in Verbindung gebracht werden soll. Es ist also ein spezieller Datumstyp erforderlich. Ein Beispiel für ein Datum ist: „14. März 2010“ (Der Tag des ersten Rennens der Saison 2010).

- Die Zeitangabe. Eine Zeitangabe ist eine spezielle Entität, die mit Ereignissen in Verbindung gesetzt werden soll. Da Zeitangaben nichts mit anderen Entitäten (wie Fahrern) gemeinsam haben, wird für sie ein eigener Typ erstellt. Ein Beispiel für eine Zeitangabe ist: „Runde 12“.

- Der Ort. Orte sind spezielle Entitäten, die - wie das Datum - nicht mit den Ereignissen in direkte Verbindung gebracht werden sollen. Daher wird hier ein spezieller Entitätstyp erstellt. Sollte das Programm benutzt werden, um Ereignisse aus Texten zu ziehen, bei denen der Ort eine Rolle spielt, muss die Regelsprache um Orte erweitert werden. Der Ortstyp kann aber in dieser Form bestehen bleiben. Ein Beispiel für einen Ort ist: „Hockenheim“.

- Die Signalphrase. Eine Signalphrase ist ein Satzbaustein, der ein bestimmtes Ereignis signalisiert. Es kann sich um ein einzelnes Wort wie „Unfall“ handeln; möglich ist aber auch ein Satzteil wie „als erster über die Ziellinie“.

- Die Nominalphrase. Eine Nominalphrase ist ein Satzbaustein, der sich auf eine Entität bezieht. Es kann damit entweder die Entität selbst, ein Synonym oder ein Pronomen gemeint sein.
  - Die Entität. Entitäten sind in dieser Arbeit ausschließlich Fahrer. Es ist aber möglich Entitäten, wie Teams, Hunde, Fabelwesen oder Autos, zu verwenden. Wesentlich ist nur, dass mit einer Entität genau ein Objekt gemeint ist. Ein Beispiel für eine Entität ist: „Michael Schumacher“.
  - Das Synonym. Ein Synonym ist eine Umschreibung einer Entität. Diese kann eindeutig sein, aber auch sehr viele mögliche Entitäten vertreten. Ferner kann ein Synonym auch kontextsensitiv sein, wie zum Beispiel „Teamkollege“. Ein anderes Beispiel für ein Synonym ist: „Der Ferrari-Pilot“.
  - Das Pronomen. Pronomen sind Satzbausteine die für jeden Fahrer stehen können. Ein Pronomen hat also keine bestimmten Entitäten, für die es stehen kann, ist aber kontextsensitiv. Das bedeutet, dass ein Pronomen für eine vor dem Pronomen genannte Person stehen kann, wenn beide mindestens im Kasus übereinstimmen. Ein Beispiel für ein Pronomen ist „er“.

- Die Verbalphrase. Eine Verbalphrase ist ein Satzbaustein, der eine bestimmte Handlung signalisiert. Verbalphrasen können Signalverben, Phrasenverben, Hilfsverben oder Spezialverben sein.

- Das Signalverb. Ein Signalverb ist ein Verb, das ein bestimmtes Ereignis signalisiert. Ein Beispiel hierfür wäre: „gewinnt“.
- Das Phrasenverb. Ein Phrasenverb ist ein Verb, welches in Verbindung mit einer Signalphrase auftaucht. Phrasenverben verdeutlichen, dass es sich bei einer Signalphrase wirklich um eine Signalphrase handelt. Ein Beispiel hierfür wäre „heißt“.
- Das Hilfsverb. Ein Hilfsverb ist ein Verb, welches zur Bildung einer bestimmten Tempus- oder Modusform benötigt wird. Ein Beispiel hierfür wäre „hat“, welches zum Beispiel für die Bildung des Perfekts benötigt wird.
- Das Spezialverb. Das Spezialverb ist ein Verb, welches für einen bestimmten Zweck benötigt wird. Ist es zum Beispiel Intention des Anwenders zwischen Passiv und Aktiv zu unterscheiden, reicht die Kategorie Hilfsverb nicht mehr aus, da sowohl das Passiv im Präsens als auch das Aktiv im Perfekt mithilfe eines Hilfsverbs gebildet werden. Bespiele hierfür wären also „wird“ oder „hat“. Das Spezialverb schließt diese Lücke.

3.3.2 Wie können die Textbausteine identifiziert werden?

In diesem Kapitel wird herausgearbeitet, dass die Daten und ein Großteil der Rundenangaben über reguläre Ausdrücke zu identifizieren sind. Dann wird aufgezeigt, dass alle anderen Textbausteine über Wörterbucheinträge realisiert werden. Das Wörterbuch-Konzept wird in Kapitel 3.4 betrachtet.

3.3.2.1 Identifikation durch reguläre Ausdrücke

Liegen Daten in einer strukturierten Form vor, bietet es sich an diese mit regulären Ausdrücken zu identifizieren. In den Formel 1-Berichten sind dies Daten und ein Teil der Rundenangaben.

3.3.2.1.1 Das Datum identifizieren Da ein Datum immer in einer speziellen Form vorliegt, bietet sich an, es über einen regulären Ausdruck zu identifizieren. Eine andere Möglichkeit wäre die Position des Datums. Bei den unterschiedlichen Nachrichtenagenturen divergiert diese aber. Deshalb wäre dies eine Alternative, wenn nur Texte einer ganz bestimmten Agentur verwendet werden. Die drei unterschiedlichen Datumsformate der unterschiedlichen Seiten sehen wie folgt aus:

Spox.de	Sonntag, 01.08.2010
formel1.de	01.08.2010 | 15:46 Uhr
auto-motor-und-sport.de	1. August 2010

Das Datum befindet sich bei jedem Bericht einer speziellen Seite immer an der gleichen Stelle. Wenn man die Uhrzeit und den Wochentag ignoriert, dann liegt das Datum entweder in DIN 1355-1 (01.08.2010) oder in DIN 5008 (1. August 2010) konformer Form vor.wikipedia (2011) Somit wird in dieser Arbeit ein regulärer Ausdruck verwendet. Das einzige Problem, das beachtet werden muss, ist, dass gegebenenfalls der Bericht einen Tag nach dem Rennen geschrieben wurde. Man müsste also zur Sicherheit das Datum noch einmal mit einem Kalender abgleichen, damit man auch tatsächlich den Rennsonntag als Datum erhält. Da es bei den 57 analysierten Formel-1-Berichten aber nie der Fall ist, kann diese Überprüfung vernachlässigt werden.
Trennt man nun den Teil des Textes anhand von Punkten und Leerzeichen, hat man Tag, Monat und Jahr voneinander getrennt. Nun muss man diese nur noch in das Format umwandeln, welches man hierfür verwenden möchte. In dieser Arbeit werden diese Variablen mit Integer-Variablen realisiert.

3.3.2.1.2 Die Rundenangabe identifizieren Die Rundenangaben können sehr umschrieben sein, wie zum Beispiel „Fünf Runden vor Schluss“. Sie können aber auch in einer strukturierten Form vorliegen, wie zum Beispiel „Runde 7“ oder „44. Runde“. Für letzteres bietet sich ein regulärer Ausdruck an. Nun müssen die gefundenen Rundenangaben durch Leerzeichen getrennt werden, und man hat die Zahl, die für die Runde steht, separiert. In dieser Arbeit wird die Rundenzahl in einer Integer-Variablen gespeichert. Sind die Rundenangaben aber nicht in einer strukturierten Form angegeben, werden sie über Wörterbucheinträge identifiziert.

3.3.2.2 Identifikation durch Wörterbucheinträge

Die Satzbausteine die durch Wörterbücher realisiert werden sollen sind:

- Rundenangaben (zumindest der nicht strukturierte Teil),
- Orte,
- Nominalphrasen,
- Verbalphrasen und

- Signalphrasen.

3.3.2.2.1 Die Rundenangabe identifizieren Bei Rundenangaben ist ein weiterer Faktor, warum in dieser Arbeit sowohl reguläre Ausdrücke als auch Wörterbucheinträge verwendet werden, dass diese weitere Informationen enthalten, die nur sehr schwer oder gar nicht aus der Aussage selbst zu holen sind. Ein Beispiel hierfür wäre: „zehn Runden vor Schluss“. In dieser Rundenangabe ist eine zur maximalen Rundenzahl der Strecke angegebene relative Rundenzahl enthalten. Die Lösung dieses Problems ist in dem Wörterbuchkonzept in Kapitel 3.4.1 beschrieben.

3.3.2.2.2 Den Ort identifizieren Grandprixnamen sind in einer strukturierten Form vorhanden. So kommt immer ein „GP“ oder ein „Grandprix“ darin vor. Wie schon bei den Rundenangaben angesprochen, werden weitere Informationen benötigt, welche nicht aus der Textstelle selbst zu holen sind, wie zum Beispiel die maximale Rundenzahl. Ähnliches gilt für den Streckennamen. Dieser hat fast immer ein „Circuit“ oder ein „Ring“ im Namen. Orts- und Ländernamen haben keine strukturierte Form.
Da nur eine bestimmte Anzahl an Orten, Grandprixnamen, Städten und Streckennamen in Frage kommt, kann man speziell für diese Aufgabe ein eigenes Wörterbuch erstellen, welches die Ort-, Länder-, Strecken- und Grand Prix Namen aller Rennen enthält. Dies stellt, vor allem wenn man die Berichte von einem bestimmten Zeitraum wählt, keine große Herausforderung dar. Da wir auf diese Weise ein relativ kleines Wörterbuch erhalten, besteht die Möglichkeit, nicht erst nach Namen zu suchen, sondern alle Wörter des Textes mit dem relativ kleinen Wörterbuch zu vergleichen. Hierfür ist zumindest eine Tokenisierung erforderlich. Auf diese Weise vermeidet man Fehler, die die Namenserkennung verursachen könnte. Da dieser Weg die besten Ergebnisse verspricht und einen überschaubaren Aufwand verursacht, wird er für diese Arbeit verwendet.
Bei den meisten Berichten wird der Name des aktuellen Grand Prix am häufigsten genannt. Es kommt aber, vor allem im Bereich der letzten Rennen, zu Nennungen der anderen Grand Prix („mit dem Doppelsieg... von Sao Paulo“). Der Grand Prix Name des aktuellen Rennes wird aber in allen Berichten als erstes genannt. Somit bietet es sich an, den Grand Prix Namen, der als erstes genannt wird, als Rennort zu setzen. Dies kann gegebenenfalls noch über die Häufigkeit verifiziert werden.

3.3.2.2.3 Eine Phrase identifizieren Als Phrasen werden hier Nominalphrasen, Verbalphrasen und Signalphrasen gesehen. Diese liegen in keiner strukturierten Form vor.

Sie haben Attribute, die einem Wörterbuch entnommen werden müssen. Aufgrund dieser beiden Aspekte wird deutlich, dass hier ein Wörterbuch erforderlich ist.

3.4 Das Wörterbuchkonzept

Als Wörterbuch wird in dieser Arbeit der in Kapitel 2.3.3 beschriebene ConceptMapper verwendet. Dieser bietet die Möglichkeit, einzelne Tokens zu definieren und ein oder mehrere Varianten dafür anzugeben. Ebenfalls können für ein Token Attribute definiert werden.

3.4.1 Die Attribute der Rundenangaben

Die Rundenangaben werden über ein separates Wörterbuch realisiert. Eine Rundenangabe kann eine konkrete Angabe sein, wie zum Beispiel „Runde 4“, oder eine relative Angabe wie „fünf Runden später“. Wenn das Programm auch andere Zeitangaben als Runden verarbeiten können soll, ist hier ein Typ für die Zeitangabe wichtig. Somit sind die Attribute, die eine Zeitangabe braucht: „Zeittyp“, „Zeitpunkt“ und „relativer Zeitpunkt“. Im Wörterbuch selbst werden entweder der „relative Zeitpunkt“ oder der „Zeitpunkt“ selbst bestimmt. Wie der Zeitannotator die relativen Zeitpunkte auflöst, wird genauer in Kapitel 4.2.1 beschrieben.

3.4.2 Die Attribute der Orte

Zu einem Ort ist die Stadt, die zu einem Grandprix gehört, zu benennen. Ebenfalls gehören zu einem Grandprix: der Name, das Land und der Streckenname. Für die Berechnung einer relativen Rundenangabe in Bezug auf die Gesamtrundenzahl, muss also die Rundenzahl eines Grandprix angegeben werden. Somit werden folgende Attribute benötigt: „die Stadt“, „das Land“, „der Grandprix“- und „der Streckenname“.

3.4.3 Die Attribute der Nominalphrasen

Eine Nominalphrase kann, wie in Kapitel xxx beschrieben, entweder eine Entität, ein Synonym oder ein Pronomen sein. Allen Nominalphrasen ist gemeinsam, dass sie einen Kasus, ein Genus und einen Numerus haben. Die Synonyme haben darüber hinaus eine Liste von Fahrernamen, für die sie stehen können. Die Entitäten haben einen Namen und ein Team. Das Attribut „Name“ wird bei Entitäten für den Fahrernamen verwendet, bei Synonymen für die möglichen Fahrernamen. Daraus resultieren folgende Attribute –

die nicht bei allen Nominalphrasen gesetzt sein müssen – für die Nominalphrasen: „der Typ“, „der Kasus“, „das Genus“, „der Numerus“, „der Name“ und „das Team“.

3.4.4 Die Attribute der Verbalphrasen

Eine Verbalphrase kann, wie in Kapitel xxx analysiert, entweder ein Signalverb, ein Phrasenverb, ein Hilfsverb oder ein Spezialverb sein. Alle Verbarten besitzen die Attribute: „die Person“, der „Numerus“ und „das Tempus“. Da ein Signalverb eine bestimmte Art eines Events signalisiert, muss dieser Signaltyp angegeben werden. Daraus folgt, dass die Attribute: „der Verbtyp“, „die Person“, der „Numerus“, „das Tempus“ und „der Signaltyp“ benötigt werden.

3.4.5 Die Attribute der Signalphrasen

Eine Signalphrase hat keine besonderen Attribute. Es kann sich hier um unterschiedliche Wortarten handeln. Entscheidend ist nur, dass eine Signalphrase ein Ereignis signalisiert. Daher braucht sie nur ein Attribut: „der Signaltyp“.
Folgende Tabelle zeigt eine Übersicht über alle Satzbausteine und Attribute:

Satzbaustein	Attribute
Rundenangabe	Zeittyp, Zeitpunkt, relativer Zeitpunkt
Ort	Stadt, Land, Grandprixname, Streckenname
Nominalphrase	Typ, Kasus, Genus, Numerus, Name, Team
Verbalphrase	Typ, Person, Numerus, Tempus, Signaltyp

Folgende Tabelle zeigt Beispiele für die Satzbausteine:

Satzbaustein	Attribute
„Runde 2“	Zeittyp = „Runde“, Zeitpunkt = „2“, relativer Zeitpunkt = keine Angabe
„GP Bahrain“	Stadt = „as-Sachir“, Land = „Bahrain“, Grandprixname = „GP Bahrain“, Streckenname = „Bahrain International Circuit“
„Schumi“	Typ = „Entität“, Kasus = „Nominativ, Dativ, Akkusativ“, Genus = „Maskulin“, Numerus = „Singular“ , Name = „Michael Schumacher“, Team = „Mercedes GP Petronas F1 Team”
„gewinnt“	Typ = „Signalverb”, Person = „3“ , Numerus = „Singular“, Tempus = „Präsens” , Signaltyp = „gewinnen“

3.5 Regelsprachen und Satzstrukturen

Bei den aktuellen Informationsextraktionsprogrammen werden als erstes die Wortarten bestimmt. Die Wortarten werden dabei einem Wörterbuch entnommen. Die Methoden, um nicht eindeutige Wortarten zu kalkulieren, sind recht kompliziert. Sie kommen im glücklichsten Fall gerade an einen Algorithmus heran, der einfach die am häufigsten auftretende Wortart für ein bestimmtes Wort wählt. Daher werden diese Algorithmen nicht weiter betrachtet. Außerdem sind sie für die Lösung in dieser Arbeit nicht relevant.(LIU10)
Die Regeln werden über Pattern realisiert. Hier ist entscheidend, wie genau die vorangegangene Bestimmung der Wortarten durchgeführt wurde. Ein Pattern sieht dann wie folgt aus: Wenn in einem Satz „eine Person“, „ein Verb“ und dann „eine Person“ stehen, dann folgt daraus:

$$\rightarrow Verb(Person1, Person2)$$

Mit diesen Pattern können dann Regeln erstellt werden, wie:

$$\rightarrow Unfall(Person1, Person2) : -rammte(Person1, Person2)$$

Ist eine der Personen ein Synonym oder ein Pronomen, muss eine Koreferenz-Analyse durchgeführt werden, um festzustellen, wer damit gemeint sein könnte.
Ein Nachteil besteht darin, dass die Qualität der Wortartbestimmung das Endergebnis wesentlich beeinflusst. Auch die Koreferenz-Analyse kann sehr komplex und fehlerträchtig sein. Ein performantes System dieser Art zu erstellen, welches ebenfalls gute Werte bezüglich Precision und Recall liefert, ist mit einem sehr hohen Aufwand verbunden, so dass es in dieser Arbeit nicht geleistet werden kann.(PASC06)
Vielmehr ist hier eine alternative Lösungsmöglichkeit gewählt worden, bei der berücksichtigt wird, dass Synonyme eines Spezialgebietes in der Regel sehr einfach aufzulösen sind. Außerdem sind Synonyme immer durch eine vorangegangene Entität beschrieben. Dies wird im folgenden Kapitel genauer erläutert.

3.6 Die für diese Arbeit erstellte Regelsprache

Die wesentlichen Fragen, die in diesem Kapitel geklärt werden, sind:

- Welche Eigenschaften hat die Regelsprache?
- Welche Typen unterstützt die Regelsprache?
- Welche Anforderungen gibt es an die Regelsprache?
- Was kann durch die Regelsprache ausgedrückt werden?
- Wo liegen die Schwächen der Regelsprache?

In Kapitel 3.5 werden einige Beispiele für Regeln vorgestellt, welche definiert werden müssen. Diese Regeln sind nur ein Teil der im Programm realisierten Regeln. Sie repräsentieren aber alle erforderlichen Anforderungen, die an die Regelsprache gestellt werden.

3.6.1 Die Eigenschaften der Regelsprache

Ein entscheidender Vorteil, der in dieser Arbeit entwickelten Lösung, liegt darin, dass nur die Wörter verwendet werden, welche für die Aufgaben von Bedeutung sind. Verben, die nichts mit Gewinnen, Crash, Ausfällen oder den Phrasen zu tun haben, werden bei der Annotation - und somit für die Regeln - ignoriert. Dies vermindert das Risiko fehlerhafter Ergebnisse.
Zunächst einmal können die Regeln nach zwei unterschiedlichen Kategorien unterschieden werden. Es gibt „harte“ und „weiche“ Regeln.
Die „harten“ Regeln sind Regeln, bei denen alle Nominal- und Verbalphrasen sehr genau definiert werden.
Die „weichen“ Regeln haben wenige oder keine Konkretisierungen der Nominal- oder Verbalphrasen.
Ein Regelsatz, der nur aus „harten“ Regeln besteht, benötigt deutlich mehr Regeln als ein Regelsatz mit „weichen“ Regeln. Im Grunde gilt, dass eine Regel so „weich“ wie möglich, aber so „hart“ wie nötig sein sollte.
Ein weiterer Punkt, in dem sich die Regeln unterscheiden, ist die Anzahl der Sätze, die von der Regel mit einbezogen werden. Dies kann ein einzelner Satz aber auch das gesamte Dokument sein. Letzteres ist aber nur eine theoretische Möglichkeit. Eine solche Regel hätte keinen praktischen Nutzen, weil sie nur für ein einziges Dokument gültig und der Aufwand zur Erstellung sehr hoch wäre.
Außerdem unterscheiden sich die Regeln im Hinblick auf Koreferenzen. Eine Regel kann keine, eine oder mehrere Koreferenzen haben.

Schließlich ist noch die Anzahl der aus einer Regel folgenden Ereignisse zu beachten. Aus einer Regel können ein oder mehrere Ereignisse folgen. Da die Übersichtlichkeit deutlich beeinträchtigt wird, wenn mehrere Ereignisse aus einer Regel folgen, ist davon abzuraten, wenn das nicht unbedingt notwendig ist.

3.6.2 Typen in der Regelsprache

Es werden:

- Signalphrasen,
- Zeitangaben,
- Nichtphrasen,
- Verbalphrasen und
- Nominalphrasen

verwendet. Die Nominalphrasen teilen sich in die Unterkategorien: Entität, Synonym und Pronomen.

3.6.2.1 Signalphrase

Eine Signalphrase hat keine weiteren Eigenschaften. Sie besteht lediglich aus einem oder mehreren Wörtern, die eine gewisse Aussage signalisieren. So signalisiert zum Beispiel das Wort „Gewinner“, dass es an dieser Stelle ums Gewinnen geht. Meistens erfordern diese Signalphrasen ein Verb, welches die Gewinnaussage untermauert. Hätte man zum Beispiel den Satz: „Der Sieger des letzten Rennes ist diesmal nicht mitgefahren.“, dann befindet sich hier zwar eine Signalphrase des Gewinnens, aber das untermauernde Verb fehlt. Hat man aber den Satz: „Fernando Alonso heißt der Sieger des Rennens.“, dann untermauert das Verb: „heißen“ hier die Gewinnaussage.

3.6.2.2 Zeitphrase

Zeitangaben können in unterschiedlichen Formen vorliegen. Einerseits können sie konkret oder relativ sein, andererseits können sie auch unterschiedliche Einheiten haben, wie zum Beispiel eine Runde oder ein Datum.

Relative Zeitangaben aufzulösen, wie zum Beispiel: „drei Runden später“, wird nicht über die Regeln realisiert. Relative Rundenangaben werden in Bezug zu einer davor liegenden Angabe gesetzt und somit konkretisiert.
Die Einheit der Zeitangabe wird über einen String realisiert, damit nur Angaben des gleichen Typs in Verbindung gesetzt werden können. Somit ist die Einheit ebenfalls für die Regeln selbst nicht weiter relevant.
Letztendlich ist also für die Regeln nur zu beachten, an welcher Stelle in einem Satz die Zeitangabe steht. Dies ist interessant bei der Frage des Satzaufbaus für die Regel. Ebenfalls muss in der Regel angegeben werden, an welcher Stelle im Satz die zum Ereignis gehörende Zeitangabe steht.

3.6.2.3 Nichtphrase

Eine Nichtphrase hat keine besonderen Eigenschaften. Nichtphrasen haben einzig und allein den Zweck auszudrücken, dass hier eine Aussage negiert wird. Hat man zum Bespiel den Satz: „Antonio Liuzzi hat nicht gewonnen“, dann muss die Gewinnaussage hier negiert werden. Um eine solche Aussage abbilden zu können, wird die Nichtphrase verwendet.

3.6.2.4 Nominalphrase

Wie in Kapitel 3.5.1 erwähnt, teilen sich die Typen der Nominalphrase in Entität, Synonym und Pronomen. Also muss in den Regeln angeben werden, von welchem Typ eine Nominalphrase ist. Die anderen Attribute sind: der Kasus, der Numerus und das Genus.
Die bisherigen Typen sind: eine Entität, ein Synonym oder ein Pronomen. Alle drei Typen haben die gleichen Attribute. Diese Typen werden dann verglichen, wenn die Satzeigenschaften auf die konkreten Satzannotationen überprüft werden.
Der Kasus eignet sich dazu, Regeln zu verfeinern. Hat man zum Beispiel den Satz: „Schumi krachte in die Mauer, der Spanier fuhr an ihm vorbei.“, könnte man nun definieren, dass das zu „Schumi“ und „krachte“ gehörende Objekt im Akkusativ stehen muss. Somit kann sichergestellt werden, dass „ihm“ und „der Spanier“ als Möglichkeiten wegfallen. Man muss also bei den Verbalphrasen angeben können, in welchem Kasus das Objekt steht, welches darauf folgt. Dazu mehr in Kapitel 3.5.1.4.
Der Numerus ist ebenfalls mit der Verbalphrase in Verbindung zu setzen. Eine Nominalphrase im Plural kann nicht mit einem Verb zusammenhängen, welches im Singular steht.

So ist zum Beispiel bei dem Satz: „Die Fahrer fuhren, Schumi siegte.“, das Wort „siegte“ eindeutig nicht mit „Die Fahrer“ in Verbindung zu setzen. Dieses gilt auch umgekehrt. So kann ein Verb im Plural nicht zu einer Nominalphrase im Singular gehören, wohl aber zu zwei Nominalphrasen im Singular. Im vorangegangenen Beispiel kann „Schumi“ niemals allein zu dem Verb „fuhren“ gehören.
Das Genus ist vor allem für die Koreferenz-Analyse notwendig. Liegen folgende zwei Sätze vor: „Schumi überholte Alonso. Zwei Runden später krachten sie ineinander“, dann kann sich das „sie“ nicht auf „Schumi“ oder „Alonso“ allein beziehen. Dies hat damit zu tun, dass sowohl „Schumi“ als auch „Alonso“ maskuline Entitäten sind, sich das „sie“ im Singular auf eine feminine Entität beziehen muss. Wenn also zwei Nominalphrasen verglichen werden sollen, benötigt man ebenfalls das Genus, um eine höhere Präzision zu erreichen.

3.6.2.5 Verbalphrase

In der bereits erwähnten Verbalphrase gibt es vier Eigenschaften. Diese Eigenschaften sind die Person, der Numerus, das Tempus und das Objekt. Die Verbalphrase selbst kann unterschiedliche Typen haben, wie Hilfsverb, Signalverb, Verb, das zu einer Signalphrase gehört, oder weitere vom Anwender selbst definierbare Verbarten. Der Numerus und das Tempus sind für die Regelerstellung relevant. So gibt zum Beispiel der Numerus an, ob eine oder mehrere Entitäten an diesem Ereignis beteiligt sein müssen. Wenn zum Beispiel „krachten ineinander“ verwendet wird, dann müssen mindestens zwei Entitäten beteiligt sein.
Das Tempus ist zum Erkennen von passiven Verbformen und der daraus resultierenden umgedrehten Entitätsbestimmungen notwendig. Hat man zum Beispiel den Satz: „Robert Kubica torpedierte Michael Schumacher.“, dann ist der als erste genannte Fahrer offensichtlich der Unfallverursacher. Würde der Satz aber lauten: „Robert Kubica wurde von Michael Schumacher torpediert.“, ist der als zweites genannte Fahrer der Unfallverursacher. Dies ist an der Verwendung des Passivs zu erkennen.
Ebenfalls deutet die Verwendung des Futurs oder des Konjunktivs darauf hin, dass es sich hier nicht um das Ereignis des Rennens handelt. Lautet der Satz: „Wäre das Rennen nach 10 Runden zu Ende gewesen, wäre Sebastian Vettel der Gewinner des Rennens gewesen.“, deutet der Konjunktiv darauf hin, dass das Ereignis so nicht stattgefunden hat. Und wenn der Satz das Futur verwendet, wie: „Ich werde das nächste Rennen gewinnen, sagte Lewis Hamilton.“, ist das Futur ein Zeichen dafür, dass das Ereignis so –zumindest noch nicht – stattgefunden hat.

Die Person ist hilfreich bei der Frage, auf welche Person sich ein Verb bezieht. So könnte man zum Beispiel Verben ignorieren, die in der ersten Person vorliegen, weil es sich zumeist um wörtliche Rede handelt. Oder es werden Regeln erstellt, die die Tatsache der wörtlichen Rede beachten. Zum Beispiel könnte man dann aus dem Satz: „Ich hatte in der ersten Kurve eine Kollision mit Mark Webber, sagte Jenson Button.“ herausziehen, dass Jenson Button und Mark Webber eine Kollision hatten.
Das letzte Attribut ist das Objekt. Genauer könnte man hier sagen: die möglichen Kasus, die das darauffolgende Objekt haben kann. Dies ermöglicht, die Regeln für Satzkonstruktionen noch stärker zu verfeinern. Auf diese Weise kann zum Beispiel, wie im vorherigen Kapitel beschrieben, definiert werden, dass auf „krachte in“ ein Akkusativ folgen muss. Ein weiteres Beispiel wäre, dass auf ein „kollidierte mit“ ein Dativ folgen muss.
Nun muss noch beachtet werden, dass eine Form eines Verbs für mehrere Fälle gelten kann. So ist zum Beispiel „krachte“ zum einen erste Person Singular im Präsens, aber auch Konjunktiv eins für sowohl erste als auch dritte Person Singular. Somit muss es ermöglicht werden, die Attribute in Abhängigkeit zueinander anzugeben. Dies bedeutet, dass in diesem Beispiel angeben werden muss, dass nur, wenn der Konjunktiv gemeint ist, auch die dritte Person Singular möglich ist.
Die folgende Tabelle gibt eine Übersicht über die für die Regeln erforderlichen Typen und deren Attribute:

Typ	Attribute
Signalphrase	Keine
Zeitangabe	Keine
Nominalphrasen	Typ, Kasus, Numerus, Genus
Verbalphrase	Person, Numerus, Tempus, Objekt

Alle diese Typen können aus einem Wort oder einem Wortgefüge bestehen.

3.6.3 Anforderungen an die Regelsprache

Die Regelsprache muss alle in dem vorherigen Kapitel herausgearbeiteten Typen abbilden können. Ebenfalls muss die Sprache alle Eigenschaften von diesen darstellen können. Sie muss aus folgenden Teilen bestehen:

- Satzeigenschaften,

- Ereignisse und
- Koreferenzen.

3.6.3.1 Die Satzeigenschaft

Bei den Satzeigenschaften müssen die an einer Regel beteiligten Sätze angegeben werden. Dies bedeutet, dass hier alle Typen und deren Attribute eines Satzes beschrieben werden müssen. Außerdem muss ihre Reihenfolge beschrieben werden.

3.6.3.2 Das Ereignis

Die Ereignisse, die aus den Sätzen resultieren, müssen angeben werden. Zu den Ereignissen muss genannt werden:

- wie das Ereignis heißt, bzw. in welchem Wort des Satzes die Ereignisaussage zu finden ist;
- welche Entitäten auf welche Weise an dem Ereignis beteiligt sind.

3.6.3.3 Die Koreferenz

Bei den Koreferenzen muss angeben werden, welche Nominalphrasen überprüft werden sollen. Es geht im Wesentlichen darum, ob ein Synonym für eine Entität stehen kann. „Der Sauber-Pilot“ kann zum Beispiel nicht durch „Michael Schumacher“ aufgelöst werden.

3.6.4 Struktur der Regelsprache

Zunächst werden die Sätze, dann die daraus folgenden Ereignisse und schließlich die zu überprüfenden Nominalphrasen angegeben. Die Regelsprache hat also folgende Struktur:

Regelstruktur	s
Satzeigenschaften	• Eine Satzeigenschaft kann aus mehreren Phrasen bestehen. Eine Satzeigenschaft gilt für einen Satz. Geht eine Regel über mehrere Sätze, dann werden auch mehrere Satzeigenschaften definiert.
Ereignisse	• Ein Ereignis besteht aus einem Signalwort, also einer Angabe, in welchem Wort die Ereignisaussage zu finden ist. • Es kann eine Zeitangabe beinhalten, also eine Angabe, in welcher Phrase die Zeitaussage zu finden ist. • Die beteiligten Entitäten werden ebenfalls angegeben. Hier ist die Betitelung der Beteiligung dem Verfasser der Regeln selbst überlassen. So kann man zum Beispiel Täter und Opfer definieren. Dies ist aber bei den Unfällen in den Formel-1-Berichten fast nie eindeutig definiert, so dass hier generell alle als Beteiligte bezeichnet werden.
Koreferenzen	• Hier wird angegeben, welche zwei Wörter zueinander passen sollen. Numerus, Genus, Kasus werden hier nicht berücksichtig, da dies über die Satzeigenschaften schon geregelt werden kann. Hier werden lediglich Synonyme auf mögliche Entitäten geprüft.

3.6.5 Die Fähigkeiten der Regelsprache

Mit der Regelsprache ist es möglich, alle Variationen – unter Berücksichtigung der Attribute - der in Kapitel 3.5.1 herausgearbeiteten Phrasen in einem Satz zu definieren. Die Regelsprache erlaubt es, sehr „weiche“ Regeln zu erstellen, wie zum Beispiel: „wenn in einem Satz Nominalphrase des Typs Entität und ein Verb als Signalverb vorkommen,

dann ist die Entität der Beteiligte und in dem Verb ist der Ereignistyp zu finden.“. Es können aber auch sehr „harte“ Regeln definiert werden, wie „Wenn in einem Satz eine Nominalphrase des Typs Entität im Nominativ, Plural und Maskulin und eine Verbalphrase, welche ein Signalverb ist, in der dritten Person Plural, Präteritum, vorkommen, dann ist die Entität der Beteiligte und in dem Verb ist der Ereignistyp zu finden.“.
Es ist möglich, Regeln über sehr viele Sätze mit sehr vielen Ereignissen zu definieren. Dies ist aber aufgrund der unnötig hohen Komplexität und Unübersichtlichkeit nicht zu empfehlen.
Es ist mit der Regelsprache möglich, jede Art eines Ereignisses zu extrahieren. Bei entsprechender Gestaltung des Wörterbuches ist es möglich, aus unterschiedlichsten Bereichen unterschiedliche Ereignisse herauszuziehen.

3.6.6 Die Schwächen der Regelsprache

Der Nachteil der Regelsprache liegt darin, dass die erstellten Regeln sehr fein definiert werden können. Dies hat zur Folge, dass die erstellten Regeln meistens nur für eine bestimmte Art von Texten sinnvoll sind. Die in dieser Arbeit erstellten Regeln sind für Formel-1-Berichte vorgesehen und haben somit immer “Beteiligte“ an einem Event. Soll herausgezogen werden, wer von welcher Firma eingestellt wurde, müssen „Arbeiter“ und „Firma“ definiert werden, anstatt von „Beteiligter“. Dies ist zwar ebenfalls durch die Sprache realisierbar, aber man kann nicht beides auf einmal abdecken.
Die Wörterbucheinträge sind für eine gute Leistung des Programms entscheidend. Daher müssen sie sehr sorgfältig erstellt werden. Gerade bei der Erstellung der Signalphrasen ist eine gründliche Arbeitsweise sinnvoll, da sich überschneidende Phrasen ungewollte Probleme hervorrufen können.
Ein Beispiel hierfür wäre:

$\rightarrow$ `$Erfielaus_1$und`$ErfielausdenPunkten_1$

In diesem Beispiel wird deutlich, dass, wenn man „aus“ als eine Signalphrase für einen Ausfall deklariert, beide Sätze einen Ausfall signalisieren würden. Im Programm ist dies so gelöst, dass „aus den Punkten“ ein eigener Wörterbucheintrag ist, der keine Annotation zur Folge hat. Auf diese Weise wird nur der erste Satz erkannt. Eine weitere Möglichkeit wäre gewesen, die Wörterbucheinträge so zu belassen. Allerdings hätte dies zur Folge gehabt, dass einige Fehler auftreten.

Dies zeigt deutlich die größte Schwäche der Sprache auf. Sicher gibt es hierfür bessere Lösungsmöglichkeiten. Aber aufgrund der zeitlichen Grenzen bei der Anfertigung dieser Arbeit wird die Tatsache hingenommen.

3.7 Beispiele

Die einfachste Form, wie ein Gewinnfakt beschrieben ist, ist folgende:

Button gewinnt Melbourne-Krimi.

Dieser Satz besteht aus einem Subjekt und einem Verb. Das Objekt ist hier zu vernachlässigen, da es sich um ein nicht relevantes Wort handelt, welches nicht ins Wörterbuch mit aufgenommen wird. Das männliche Subjekt ist im Nominativ Singular vorhanden. Das Verb ist in der 3. Person Singular und im Präsens. Darüber hinaus ist das Verb ein Signalverb, welches auf einen Sieg hindeutet. Somit ist als Regel festzulegen: Besteht ein Satz aus:

- einer Entität, welche im Nominativ Singular vorliegt (das Geschlecht ist hier nicht relevant) und
- einer Verbalphrase in der 3. Person Singular, welches das Signalwort für „gewinnen“ ist,

dann ist dieser Satz ein eindeutiges Zeichen, dass die Entität hier der Gewinner des Rennens sein muss.
Ein Unfall kann wie folgt beschrieben sein:
Adrian Sutil kollidierte schon in der zweiten Kurve beim Kampf um Rang neun mit Robert Kubica.
Dieser Satz besteht aus einem Subjekt, einem Verb, einer Zeitangabe und einem Objekt. Subjekt und Objekt sind maskulin und im Singular. Das Subjekt ist im Nominativ und das Objekt im Dativ vorhanden. Das Verb ist in der 3. Person Singular. Die Zeitangabe deutet auf die erste Runde hin. Alle weiteren Wörter haben keine Relevanz für unsere Regeln. Nun ist als Regel festzulegen: Besteht ein Satz aus:

- einer Entität, welche im Nominativ Singular vorliegt (das Geschlecht ist hier nicht relevant)
- einer Verbalphrase in der 3. Person Singular, welches ein Signalwort für Unfall ist,

- einer Zeitphrase und
- einer Entität, welche im Dativ Singular vorliegt (das Geschlecht ist hier nicht relevant),

dann ist dieser Satz ein eindeutiges Zeichen, dass die beiden Entitäten in einen Unfall verwickelt wurden. Bei den Sätzen:
Dazu zählten besonders McLaren-Mercedes-Pilot Lewis Hamilton und Mark Webber (Red Bull). Drei Runden vor Schluss kollidierten die beiden allerdings beim Versuch, Fernando Alonso vom vierten Platz zu verdrängen.
muss die Regel lauten:
Besteht ein Satz aus:

- einer Zeitangabe,
- einer Verbalphrase in der 3. Person Singular, welches ein Signalwort für Unfall ist,
- einem Pronomen, welches im Nominativ Plural vorliegt und
- einer Entität, welche im Akkusativ Singular vorliegt (das Geschlecht ist hier nicht relevant),

und enthält der Satz davor:

- ein Synonym, welches im Nominativ Singular vorliegt (das Geschlecht ist hier nicht relevant),
- eine Entität, welche im Nominativ Singular vorliegt (das Geschlecht ist hier nicht relevant) und
- eine zweite Entität, welche im Nominativ Singular vorliegt (das Geschlecht ist hier nicht relevant),

dann folgt daraus, dass die beiden Entitäten aus dem Satz davor in einen Crash verwickelt waren, wenn das Synonym für die darauffolgende Entität stehen kann.

4 Design

In diesem Kapitel wird beschrieben, auf welche Weise die in der Analyse des vorherigen Kapitels herausgearbeiteten theoretischen Extraktionsmethoden im Programm zu realisieren sind und welche designtechnischen Entscheidungen dieses gewährleisten. Zunächst wird ein Überblick über die verwendeten Typen von Annotationen gegeben. Im darauffolgenden Kapitel werden die Kommentatoren beschrieben und auf welche Weise sie die Typdefinitionen benutzen. Das nächste Kapitel erklärt, auf welche Weise die Kommentatoren kombiniert werden. Das letzte Kapitel widmet sich der Regelsprache, wie diese ins Programm umgesetzt werden kann und welche Datenstrukturen ihr zugrunde liegen.

4.1 Typdefinitionen

Zur Realisierung der Anforderungen haben sich die in der Abbildung 4.1 gezeigten Annotationen als erforderlich herauskristallisiert:

4.1.1 Die Datumsannotation

Die Datumsannotation besteht einfach aus drei Variablen, die jeweils vom Typ „Integer“ sind und Tag, Monat und Jahr repräsentieren.

4.1.2 Die Zeitannotation

Die Zeitannotation dient der Darstellung der Runde, in welcher ein Unfall passiert ist. Diese Annotation ist sehr dynamisch gehalten, um andere Zeitangaben zu ermöglichen, wie zum Beispiel eine konkrete Uhrzeit oder eine Angabe, wie „die Mitte des Rennens“. Diese Annotation besteht aus einem String, der die Zeitart repräsentiert und zwei Integervariablen, die einen Zeitpunkt und eine Zeitdifferenz widerspiegeln.

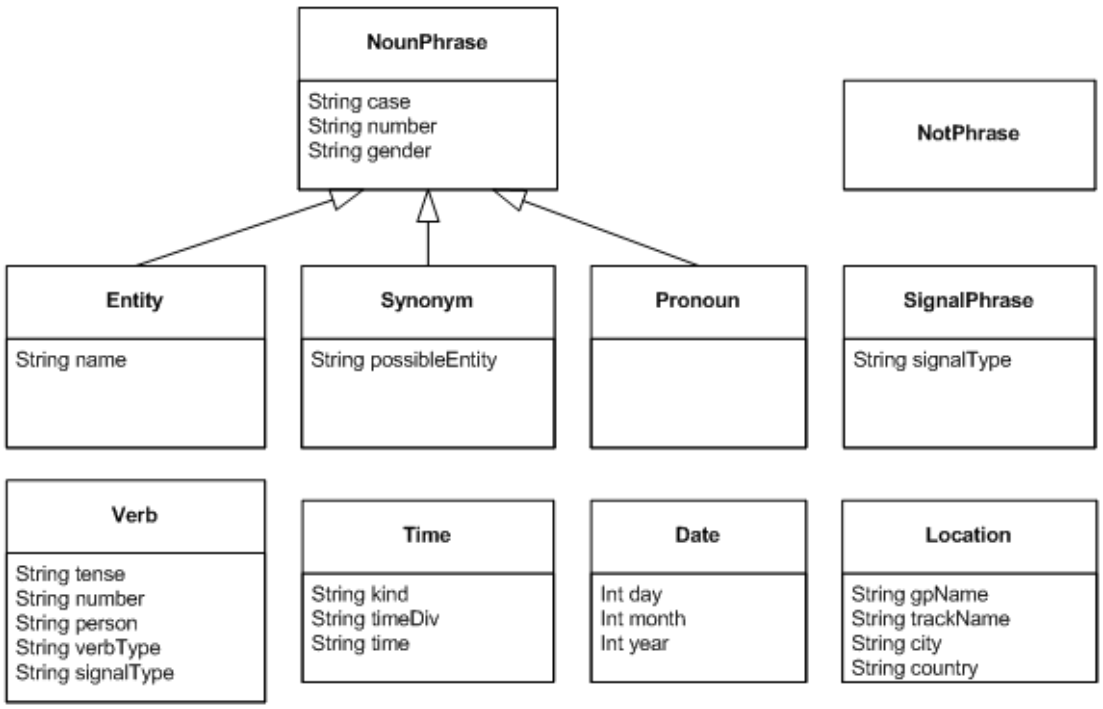


Abb. 4.1: Abbildung 4.1: Typdefinitionen

4.1.3 Die Ortsannotation

Die Ortsannotation ist sehr spezifisch auf diese Arbeit abgestimmt, da Rennorte auf sehr spezielle Weise repräsentiert werden. Sie haben zum Beispiel einen Grand Prix Namen und einen Streckennamen. Die Attribute dieser Annotation sind alle Stringvariablen. Sie beinhalten den Grand Prix-Namen, den Streckennamen, den Namen der Stadt und den Namen des Landes.

4.1.4 Die Verbannotation

Die Annotation für Verben besteht aus fünf Stringvariablen, welche Kasus, Numerus, Genus, die Verbart und den Signaltyp beinhalten. Bei der Verbart kann es sich zum Beispiel um ein Hilfsverb oder ein Signalverb handeln. Handelt es sich um ein Signalverb, dann wird ebenfalls die Variable des Signaltyps gesetzt. Dies kann dann zum Beispiel ein Signal für „gewinnen“ sein.

4.1.5 Die Signalphrasenannotation

Bei der Signalphrase kann es sich um einzelne Wörter wie „Sieger“ handeln oder um Satzteile wie „als erster über die Ziellinie“. Diese Phrasen haben nur ihr Signal in Stringdarstellung als Attribut. In der Regel bedürfen Signalphrasen eines Phrasenverbs. Dies wird in den Regeln festgelegt.

4.1.6 Die Nominalphrasenannotation

Es gibt drei unterschiedliche Nominalphrasen:

- eine Entität, bei der es sich um eine ganz bestimmte Person oder andere eindeutige Entität handelt;
- ein Synonym, welches für eine Reihe an möglichen Entitäten steht;
- ein Pronomen, welches für jede Entität stehen kann, die in Kasus, Genus und Numerus übereinstimmt.

Alle Nominalphrasen bestehen aus mindestens den drei Stringattributen Kasus, Numerus und Genus. Bei der Entität kommt noch der Name der Entität hinzu. Beim Synonym kommt ein String hinzu, welcher alle möglichen Entitäten anhand ihrer Namen speichert. Beim Pronomen sind keine weiteren Attribute erforderlich.

4.1.7 Die Nichtphrasenannotation

Bei der Nichtphrasenannotation handelt es sich um eine Annotation, die eine Ereignisaussage negiert. Wenn in einem Satz steht: „Marc Webber hat das Rennen nicht gewonnen“, dann wird das Ereignis „gewinnen“ negiert, also zu „nicht gewinnen“.

4.2 Kommentatoren

In dieser Arbeit werden sechs unterschiedliche einfache Annotatoren verwendet:

- LocationAnnotator,
- MyDateAnnotator,
- OffsetTokenizer,
- PhraseTokenizer,

- PointOfTimeAnnotator und
- WhitespaceTokenizer.

Die einfachsten Annotatoren sind in der Abbildung 4.2 dargestellt. Sie benötigen keinen Input und arbeiten direkt mit dem Text.

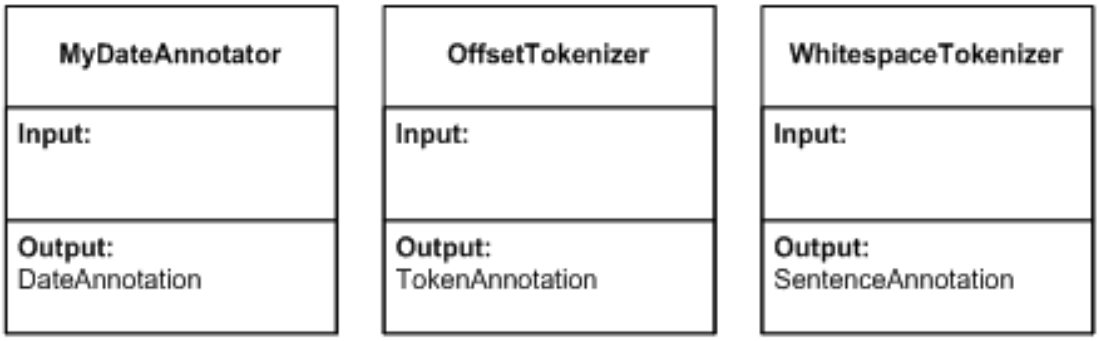


Abb. 4.2: Abbildung 4.2: einfache Annotatoren

4.2.1 Der Datumsannotator

Der MyDateAnnotator verwendet reguläre Ausdrücke, um Daten aufzufinden und zu annotieren. Hierbei wird ein Matcher erstellt und verwendet, der im Text nach den in der Analyse herausgearbeiteten Datumsformaten sucht: DIN 1355-1 oder DIN 5008.

4.2.2 Der Tokenannotator

Der OffsetTokenizer sucht nach Leer- und Trennzeichen, um Wörter zu identifizieren. Dies ist eine sehr einfache, aber für die Aufgabe völlig ausreichende Methode. Die Ausgaben dieses Tokenisierers werden von dem LocationAnnotator, PointOfTimeAnnotator und PhraseAnnotator als Input verwendet. Diese drei Annotatoren basieren auf dem ConceptMapper und benötigen zum Überprüfen der Wörterbucheinträge WortTokens.

4.2.3 Der Satzannotator

Der WhitespaceTokenizer sucht den Text nach Satztrennzeichen ab. Anhand der Satztrennzeichen können dann unter Verwendung minimalistischer Regeln Satzannotationen erstellt werden. Diese Regeln sind:

- wenn vor einem Punkt eine Zahl steht, ist es kein Satzzeichen;
- kommt nach einem Satztrennzeichen ein Absatz, dann ist es auf jeden Fall ein Satzende.

Auf die Ausgabe des WhitespaceTokenizer bauen die drei Annotatoren: LocationAnnotator, PointOfTimeAnnotator und PhraseAnnotator auf. Dies zeigt Abbildung 4.3

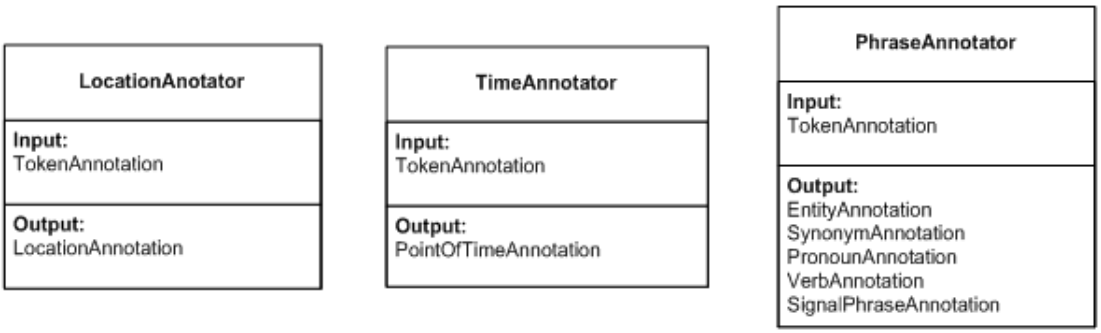


Abb. 4.3: Abbildung 4.3: komplexe Annotatoren

4.2.4 Der Ortsannotator

Der LocationAnnotator verwendet den ConceptMapper, um Wörterbucheinträge aufzufinden. Der Ort kann entweder in Form des Landesnamens, des Ortsnamens, des Grand Prix-Namens oder des Streckennamens vorliegen. Somit müssen die Wörterbucheinträge diese vier Möglichkeiten und gegebenenfalls auch unterschiedliche Varianten davon enthalten. Ein Beispiel für einen Wörterbucheintrag lautet:

```
<token gpName="GP Bahrain" trackName="Bahrain International Circuit"
       city="as-Sachir" country="Bahrain">
       <variant base="GP Bahrain"/>
       <variant base="Bahrain"/>
       <variant base="as-Sachir"/>
       <variant base="Bahrain International Circuit"/>
</token>
```

Es werden der Grand Prix-Name, der Streckenname, der Name der Stadt und der Name des Landes eingetragen. Dann wird jeder von ihnen als eine Variante deklariert.

4.2.5 Der Zeitannotator

Der TimeAnnotator benutzt sowohl reguläre Ausdrücke als auch den ConceptMapper. Dies ist der Tatsache geschuldet, dass Rundenangaben sich sehr ähneln. Von der Idee, die Rundenangaben komplett in das Wörterbuch zu integrieren – was ein dynamischeres Verhalten garantieren würde –, wurde Abstand genommen. Um aber trotzdem ein dynamisches Verhalten zu gewährleisten, was auf jede Art einer Zeitangabe anwendbar ist, wurde auf das Wörterbuch nicht verzichtet. Es können konkrete Zeitpunkte angeben werden. Es sind aber auch relative Zeitangaben möglich. So kann man zum Beispiel sagen: „in der 32. Runde“. Dies ist eine exakte Zeitangabe, die darüber hinaus noch sehr einfach über einen regulären Ausdruck zu finden ist. Ein anderes Beispiel wäre: „zehn Umläufe später“. Das ist eine relative Zeitangabe, welche sich auf eine davor liegende Zeitangabe beziehen muss. Auch hier wäre die Verwendung eines regulären Ausdrucks denkbar, es wird aber über Wörterbucheinträge realisiert. Der Wörterbucheintrag hierfür sieht folgendermaßen aus:

```
<token kind="Runde" timeDiv="10">
                <variant base="zehn␣Umläufe␣weiter" />
</token>
```

Der Zeittyp wird hier durch „kind“ angegeben. Die Tatsache, dass „time“ hier nicht gesetzt wurde, gibt hier an, dass es sich um eine relative Zeitangabe handelt. Angegeben werden der Zeitunterschied von 10 und die eine Variante, die hier möglich ist. Der TimeAnnotator geht alle Zeitannotationen durch und versucht alle relativen Angaben aufzulösen. Dafür wird die in dem gleichen Satz davor liegende Zeitangabe als Referenz benutzt. Der Zeitwert ist dann die Summe aus „timeDiv“ der Zeitannotation und „time“ der davor liegenden Zeitannotation. Ist keine Zeitangabe davor zu finden, wird einfach „timeDiv“ als „time“ gesetzt.

4.2.6 Der Phrasenannotator

Der PhrasenAnnotator ist das Herzstück der gesamten Analyse. Hier werden alle zum Erkennen eines Events oder Fakts notwendigen Annotationen realisiert. Dies sind die Entitäten, die Synonyme, die Pronomen, die Verben und die Signalphrasen. Dieser Annotator verwendet nur den ConceptMapper und erstellt anhand der Wörterbucheinträge die entsprechenden Annotationen. Sieht der Wörterbucheintrag zum Beispiel wie folgt aus:

```
<token kind="signalPhrase" signalType="win">
        <variant base="Sieger" />
```

```
        <variant base="als erster über die Ziellinie" />
</token>
```

dann wird eine SignalPhraseAnnotation erstellt, die den Signaltyp „win“ hat. Sieht der Wörterbucheintrag jedoch so aus:

```
<token kind="verb" verbType="phraseVerb" person="3" tense="present" number="sg">
        <variant base="heißt"/>
</token>
```

dann wird eine VerbAnnotation erstellt, deren Verbart „phraseVerb“ ist. Die Eigenschaften werden auf 3. Person Singular und Präsens gesetzt. Die von diesem Annotator und vom PointOfTimeAnnotator erstellten Annotationen werden als Grundlage für regelbasierte Informationsextraktion verwendet.

4.3 Kombinieren von Kommentatoren

UIMA bietet die Möglichkeit, mehrere Annotatoren nacheinander von einem Annotator zu starten. In dieser Arbeit wird dafür der Starter Annotator verwendet, der den Analysefluss auf folgende Weise definiert:

```
<flowConstraints>
        <fixedFlow>
                <node>DateAnnotator</node>
                <node>SentenceTokenizer</node>
                <node>Tokenizer</node>
                <node>PointOfTimeAnnotator</node>
                <node>LocationAnnotator</node>
                <node>PhraseAnnotator</node>
        </fixedFlow>
</flowConstraints>
```

Die aufgeführten Annotators müssen wie am Beispiel des DateAnnotators vorher importiert werden.

```
<delegateAnalysisEngine key="DateAnnotator">
        <import location="MyDateAnnotator.xml" />
</delegateAnalysisEngine>
```

4.4 Regelsprache

Eine Regel besteht - wie in Kapitel 3.5 herausgearbeitet - aus einer Arrayliste von Satzeigenschaften und einer Arrayliste von Regeln für die daraus resultierenden Ereignisse. Das beschreibt Abbildung 4.4:

Abb. 4.4: Abbildung 4.4: Regeln

4.4.1 Satzeigenschaften

Eine Satzeigenschaft beinhaltet eine Liste an Phrasen. Eine Phrase kann entweder eine Signalphrase, eine Zeitphrase, eine Nominalphrase oder eine Verbalphrase sein. Dies wird in Abbildung 4.5 gezeigt.

Abb. 4.5: Abbildung 4.5: Satzeigenschaften

Die Phrasen werden in einer Liste gespeichert. Eine Übersicht über die Phrasen findet sich in Abbildung 4.6.
Die in der Analyse herausgearbeiteten Attribute, die für die jeweiligen Phrasen erforderlich sind, wurden hier als Strings umgesetzt.

4.4.2 Regeln für die Ereignisse

Um die Regeln für die Ereignisse zu erklären, müssen zunächst zwei dafür erstellte Typen definiert werden. Diese sind MeaningRule und Koreferenz. Eine MeaningRule gibt an, an welcher Stelle in welchem Satz etwas mit einer bestimmten Bedeutung steht. Das können die Beteiligten oder das Eventsignalwort sein. Eine MeaningRule sieht wie Abbildung 4.7 zeigt.
Eine MeaningRule besteht aus zwei Integerwerten, welche die Wortposition und die Satzposition angeben, und aus einem String, der den Typ dieser Regel angibt, wie zum Beispiel: „Beteiligter“.
Eine Koreferenz beinhaltet zwei MeaningRules. Das zeigt Abbildung 4.8.

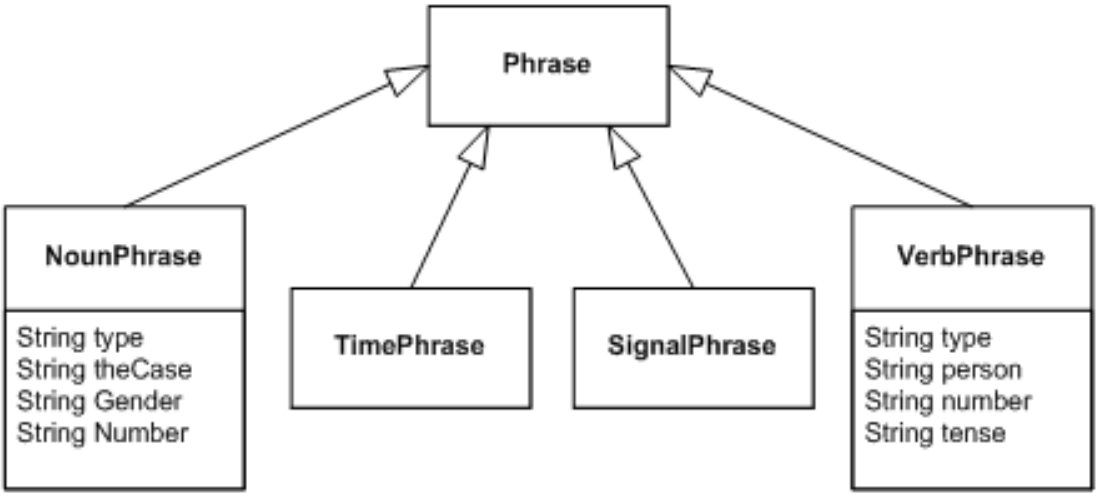


Abb. 4.6: Abbildung 4.6: Phrasen

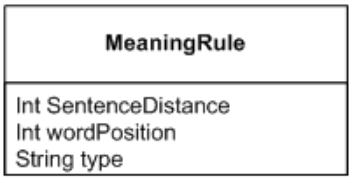


Abb. 4.7: Abbildung 4.7: MeaningRule

Somit besteht die Koreferenz aus zwei MeaningRules. Diese beiden sollen dann bei der späteren Analyse auf Kompatibilität überprüft werden.
Die Entitäten brauchen – der Analyse aus Kapitel 3.5 zufolge – Satzeigenschaften, Entitätsregeln und Koreferenz-Regeln. Zu einer Regel können mehrere Ereignisse definiert werden. Ein Ereignis hat bestimmte Entitäten und Koreferenzregeln.
Daher macht es Sinn, eine Klasse „Ereignisregel“ zu erstellen. Diese Klasse beinhaltet alle für ein Ereignis wichtigen Eigenschaften wie Entitäts- und Koreferenzregeln.
Die Klassen sehen dann wie Abbildung 4.9 zeigt aus.
Jede Regel besteht aus einer Liste von Satzeigenschaften und einer Liste von EventRules. Die Satzeigenschaften wurden im vorherigen Kapitel beschrieben. Die EventRules spiegeln eine Regel für einen Event wieder. Ein Event besteht aus einem Typ und einer Zeitangabe, die durch eine MeaningRule repräsentiert werden.

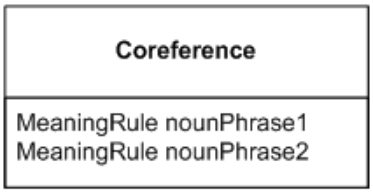


Abb. 4.8: Abbildung 4.8: Koreferenz

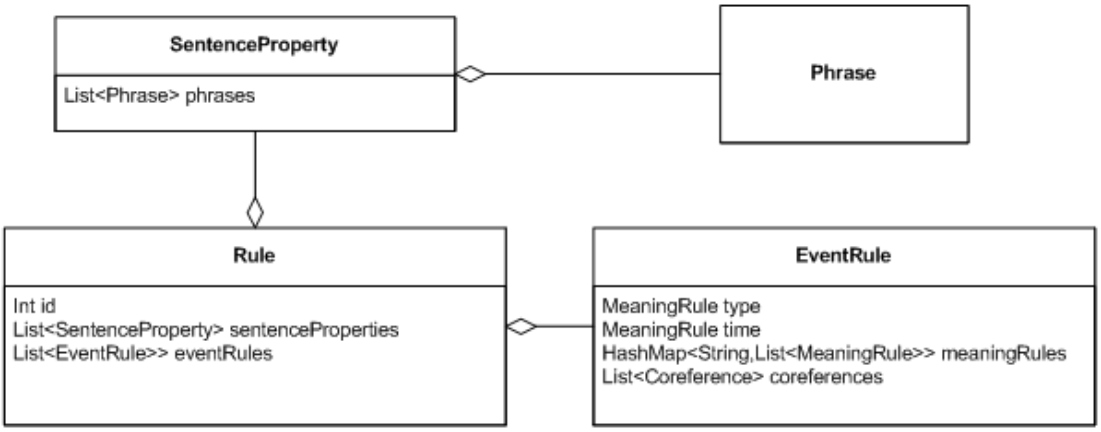


Abb. 4.9: Abbildung 4.9: Regeln

Ferner gibt es eine HashMap, welche alle weiteren Beteiligungen unter dem Namen der Beteiligung speichert. In dieser Arbeit werden dort nur Beteiligte gespeichert. Es wäre aber generell möglich, hier einen anderen Namen als Beteiligung zu verwenden.
Die Koreferenzen sind eine Liste an Koreferenzen. Alle enthaltenen Koreferenzen müssen im Programm dann auf Kompatibilität überprüft werden.

5 Realisierung

5.1 Datum des Rennens

Um das Datum des Rennes aus den Texten zu ziehen, wird ein regulärer Ausdruck verwendet. Zunächst einmal wird ein Typ definiert. Da es sich hier um ein Datum handelt, wird logischerweise ein Datumstyp erstellt. Dieser Datumstyp besteht jeweils aus einem Integer für den Tag, den Monat und das Jahr. Bei UIMA ist dies relativ einfach in der in Abbildung 5.1 gezeigten Eingabemaske zu realisieren:

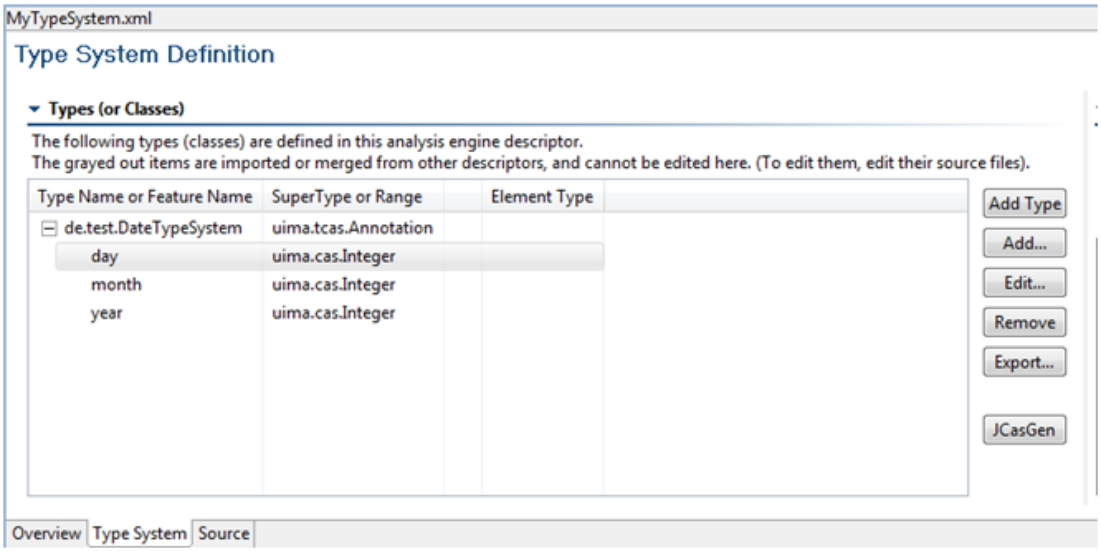


Abb. 5.1: Abbildung 5.1: Typdefinitionen von Datum

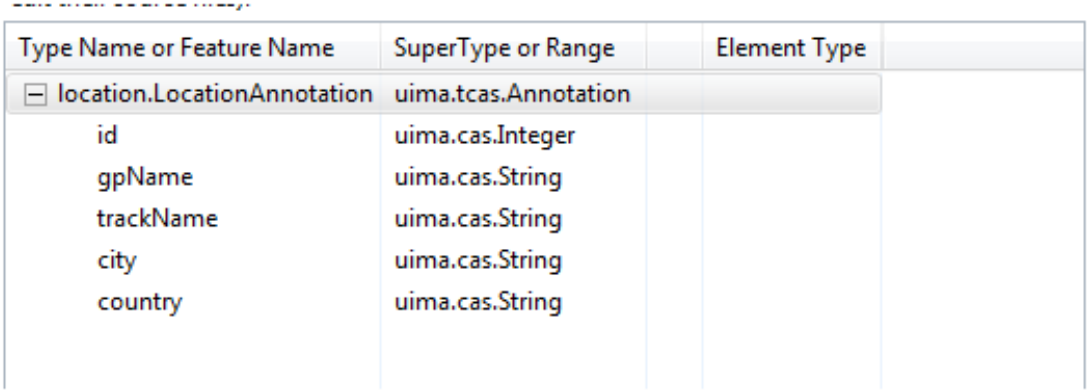
Als nächstes muss der Annotator definiert werden. Hier werden die regulären Ausdrücke definiert, welche die Datumsangaben beschreiben. Die regulären Ausdrücke sehen wie folgt aus:
Die regulären Ausdrücke werden als Pattern in der Annotatorklasse gespeichert. In der Process-Methode wird anhand des Patterns ein Matcher erstellt, welcher dann auf

Matches überprüft wird. Für jeden Match wird ein Annotator erstellt und dieser dann zum Index hinzugefügt. Der Teil der Process-Methode für das letzte Pattern sieht wie folgt aus:
Die regulären Ausdrücke werden als Pattern in der Annotatorklasse gespeichert. In der Process-Methode wird anhand des Patterns ein Matcher erstellt, welcher dann auf Matches überprüft wird. Für jeden Match wird ein Annotator erstellt und dieser dann zum Index hinzugefügt. Der Teil der Process-Methode für das letzte Pattern sieht wie folgt aus:

```
matcher = myIntPattern.matcher(docText);
while (matcher.find()) {
//Datumsannotation erstellen und ihr die jCas bei der Erstellung übergeben
DateTypeSystem annotation = new DateTypeSystem(aJCas);
//Die Attribute der Annotation setzen
annotation.setBegin(matcher.start());
annotation.setEnd(matcher.end());
//hier wird der Substring für den Tag im Originaltext repräsentiert geholt
String day = docText.substring(matcher.start(), matcher.start() + 2);
annotation.setDay(Integer.parseInt(day));
//hier wird der Substring für den Monat im Originaltextrepräsentiert geholt
String month = docText.substring(matcher.start() + 3, matcher.start() + 5);
annotation.setMonth(Integer.parseInt(month));
//hier wird der Substring für das Jahr im Originaltextrepräsentiert geholt
String year = docText.substring(matcher.start() + 6, matcher.start() + 10);
annotation.setYear(Integer.parseInt(year));
//hier wird der Annotation gesagt, dass sie sich zum Index hinzufügen soll
annotation.addToIndexes();
//alternativ könnte man auch über den jCas die Annotation hinzufügen:
//-> aJCas.addFsToIndexes(annotation);
}
```

5.2 Ort des Rennens

Zum Extrahieren des Ortes des Rennens wird der ConceptMapper verwendet. Die Inputtokens werden auf die Wörterbucheinträge hin überprüft, und es werden LocationAnnotations erstellt. Unter Verwendung des Component Descriptor Editors sieht das Erstellen der LocationAnnotation wie Abbildung 5.2 zeigt aus:
Der leicht umgewandelte ConceptMapper wurde, was die Ausgabeannotation angeht, auf LocationAnnotation umgewandelt. Ebenfalls wurden die Attribut-und Variablenliste angepasst, die übereinstimmen müssen. Ansonsten wurden keine weiteren Eigenschaften des ConceptMappers umgewandelt.



Type Name or Feature Name	SuperType or Range	Element Type
location.LocationAnnotation	uima.tcas.Annotation	
id	uima.cas.Integer	
gpName	uima.cas.String	
trackName	uima.cas.String	
city	uima.cas.String	
country	uima.cas.String	

Abb. 5.2: Abbildung 5.2: Typdefinitionen von Ort

Der erste Ort eines Rennens, der in einem Text vorkommt, wird dann von der Applikation als der Ort des Rennens angesehen. Es hat sich herausgestellt, dass der Grand Prix selbst immer in dem zugehörigen Text als erstes genannt wird. Die Überlegung, den am häufigsten genannten Rennort in einem Bericht als den dazugehörigen Grand Prix zu betrachten, hat sich als problematisch herausgestellt. In den Berichten gegen Ende der Saison werden häufig andere Grand Prix genannt. Dies kann sogar so weit gehen, dass ein Grand Prix, um den es in einem Bericht geht, nicht am häufigsten genannt wird. Die verwendete Methode, einfach den als ersten genannten Grand Prix zu nehmen, ist somit die zuverlässigste.

5.3 Gewinner des Rennens

Durch die im nächsten Kapitel beschriebenen Methoden werden Gewinn-Fakten aus dem Text gezogen. Bei einem perfekten Programm würden alle diese Fakten den wahren Gewinner als den Gewinner des Rennes beschreiben. Diese Frage wird in Kapitel **??** genauer betrachtet und soll hier nicht weiter vertieft werden. Um nun den Sieger des Rennens zu bestimmen, wird einfach der Fahrer des ersten Gewinnfakts als Gewinner des Rennens gesetzt. Dies ist im Programm wie folgt realisiert:

```
1 if (!(result.getResult().get("win") == null))
2         result.setWinner(result.getResult().get("win").get(0).getParticipants().get(0));
```

Events und Fakten werden im Result-Datentyp in einer HashMap mit dem Signaltyp als Key und einer Liste von Resulttypen als Value gespeichert. Dies wird in dem nächsten

Kapitel genauer beschrieben. Auf diese Liste wird zugegriffen und es wird ihr der erste Eintrag entnommen. Von diesem Eintrag wird eine Liste der beteiligten Entitäten geholt. Von dieser Liste der beteiligten Entitäten wird der erste Beteiligte genommen und als Gewinner des Rennens gesetzt. Weil es sich beim Gewinner des Rennens nicht um mehrere Entitäten handeln kann, ist natürlich die Frage zu klären, warum eine Liste von Beteiligten und nicht einfach ein einziger Beteiligter beim Fakt gespeichert wird. Dies ist der Tatsache geschuldet, dass so eine hohe Dynamik ermöglicht werden kann. Dies wird im nächsten Kapitel genauer beschrieben.

5.4 Fakten und Events des Rennens

Zunächst wird eine Übersicht gegeben über die Wörterbucheinträge, die benötigt werden und die Signalphrasen und Verbalphrasen zu identifizieren. Dann wird ein kurzer Einblick gegeben, wie die Regeln mit den Annotationen verglichen werden und wie entschieden wird, welche Regel genommen wird.

5.4.1 Die Wörterbucheinträge für Signalphrasen

```
1       <token kind="eine Signalphrase" signalType="win">
2               <variant base="als Erster die Ziellinie" />
3               <variant base="als Erster über die Ziellinie" />
4               <variant base="das grösse Los in der Regenlotterie" />
5               <variant base="der König" />
6               <variant base="der neue König" />
7               <variant base="Gewinner" />
8               <variant base="Heimerfolg" />
9               <variant base="Heimsieg" />
10              <variant base="König" />
11              <variant base="Regenkönig" />
12              <variant base="Saisonsieg" />
13              <variant base="Sieg" />
14              <variant base="Sieger" />
15              <variant base="Siegerpokal" />
16              <variant base="Vorjahressieg" />
17      </token>
18      <token kind="eine Signalphrase" signalType="crash">
19              <variant base="ab" />
20              <variant base="Abflug" />
21              <variant base="aneinander" />
22              <variant base="Berührung" />
23              <variant base="Crash" />
24              <variant base="Crashs" />
25              <variant base="Dreierkollision" />
```

```
26              <variant base="Horrorcrash" />
27              <variant base="in␣den␣gegnerischen␣McLaren" />
28              <variant base="in␣die␣Bande" />
29              <variant base="in␣die␣Reifenstapel" />
30              <variant base="in␣einen␣Reifenstapel" />
31              <variant base="Kollision" />
32              <variant base="Unfall" />
33              <variant base="von␣der␣Piste␣geräumt" />
34      </token>
35      <token kind="eine␣Signalphrase" signalType="malfunction">
36              <variant base="aus" />
37              <variant base="Rad␣ab" />
38              <variant base="raus" />
39              <variant base="Schluss" />
40              <variant base="sein␣Renault-Motor␣den␣Geist␣aufgab" />
41              <variant base="zum␣Stehen" />
42      </token>
43
44      <token kind="eine␣Verbalphrase" signalType="crash" verbType="Signalverb" person="3" ten
45              <variant base="abflog" />
46              <variant base="abschoss" />
47              <variant base="anschob" />
48              <variant base="crasht" />
49              <variant base="in␣die␣Prärie␣feuerte" />
50              <variant base="kollidierte" />
51              <variant base="krachte" />
52              <variant base="rammte" />
53              <variant base="riss" />
54              <variant base="schlug␣in␣die␣Mauer␣ein" />
55              <variant base="spießte" />
56              <variant base="torpedierte" />
57              <variant base="kollidierten" number="pl" />
58              <variant base="nach␣einer␣kurzen␣Berührung␣in␣die␣Wiese␣mussten" number="pl" /
59              <variant base="knallt" tense="present" />
60              <variant base="kracht" tense="present" />
61              <variant base="kollidiert" tense="participleII,present" />
62              <variant base="touchiert" person="all" tense="participleII" />
63              <variant base="torpediert" person="all" tense="participleII,present" number="a
64              <variant base="zusammengekracht" person="all" tense="participleII" number="all
65      </token>
66      <token kind="eine␣Verbalphrase" signalType="win" verbType="Signalverb">
67              <variant base="gewinnt" person="3" tense="present" number="sg" />
68              <variant base="siegt" person="3" tense="present" number="sg" />
69              <variant base="triumphiert" person="3" tense="present" number="sg" />
70              <variant base="gewann" person="3" tense="past" number="sg" />
71              <variant base="gewonnen" person="all" tense="participleII" number="sg" />
72              <variant base="gewinnen" person="all" tense="infinitive" number="all" />
73              <variant base="triumphieren" person="all" tense="infinitive" number="all" />
74      </token>
75      <token kind="eine␣Verbalphrase" signalType="malfunction" verbType="Signalverb" person="
```

```
76                <variant base="ausfiel" />
77                <variant base="explodierte" />
78                <variant base="Motor explodierte" />
79                <variant base="sah die Zielflagge nicht" />
80                <variant base="stoppte" />
81                <variant base="verabschiedete sich vorzeitig" />
82                <variant base="beendet" tense="participleII,past" />
83                <variant base="abstellen" person="all" tense="infinitive" number="all" />
84                <variant base="aufgeben" person="all" tense="infinitive" number="all" />
85                <variant base="stehen" person="all" tense="infinitive" number="all" />
86        </token>
87        <token kind="eine Verbalphrase" verbType="Phrasenverb" person="3" tense="past" number=
88                <variant base="feierte" />
89                <variant base="feuerte" />
90                <variant base="fiel" />
91                <variant base="flog" />
92                <variant base="fuhr" />
93                <variant base="hieß" />
94                <variant base="kam" />
95                <variant base="räumte" />
96                <variant base="rollte" />
97                <variant base="schied" />
98                <variant base="überquerte" />
99                <variant base="zog" />
100               <variant base="gerieten" number="pl" />
101               <variant base="feiert" tense="past" />
102               <variant base="fliegt" tense="present" />
103               <variant base="heißt" tense="present" />
104               <variant base="ist" tense="present" />
105               <variant base="scheidet" tense="present" />
106               <variant base="schießt" tense="present" />
107               <variant base="sichert" tense="present" />
108               <variant base="gelandet" tense="participleII" />
109               <variant base="gesichert" tense="participleII" />
110               <variant base="verwickelt" tense="participleII" />
111               <variant base="feiern" tense="infinitive" />
112               <variant base="wiederholen" tense="infinitive" />
113               <variant base="war" verbType="Phrasenverb,sein" />
114               <variant base="wurde" verbType="Phrasenverb, werden" />
115               <variant base="freuen" person="all" tense="infinitive" number="all" />
116               <variant base="gefeiert" person="all" tense="participleII" number="all" />
117               <variant base="hatte" person="1,3" verbType="Phrasenverb, Hilfsverb" />
118               </token>
119       <token kind="eine Verbalphrase" verbType="Hilfsverb" person="3" tense="past" number="s
120               <variant base="blieb" />
121               <variant base="konnte" />
122               <variant base="musste" />
123               <variant base="schaffte" />
124               <variant base="darf" tense="present" />
125               <variant base="hat" tense="present" />
```

```
126                 <variant base="muss" tense="present" />
127         </token>
```

5.4.2 Die Wörterbucheinträge für Verbalphrasen

5.4.3 Verwendung der Regeln

Die Einzelnen Annotationen der Sätze werden in der Satzannotation in ihrer Reihenfolge gespeichert. Wie in Kapitel 3.6.2 beschrieben, werden bei den Regeln die Phrasen in einer SentenceProperty gespeichert. In der Klasse Regel ist eine Methode definiert worden in der überprüft werden kann, ob eine Regel zu einem bestimmten Satz passt. Dafür geht die Regel ihre SentencePropertys durch und überprüft, ob ihre Phrasen zu den Annotationen die in der Satzannotation gespeichert wurden passen. Hierfür müssen Phrasen mit Annotationen vergleichbar gemacht werden. Dies ist in folgender Methode für die klasse Phrase definiert:

```
1  public abstract boolean equalToAnnotation(Annotation ann);
2
3  Ihr konkreter Code für die Klasse Verbalphrase sieht dann wie folgt aus:
4  @Override
5  public boolean equalToAnnotation(Annotation ann) {
6          if (ann instanceof VerbAnnotation) {
7                  VerbAnnotation vAnn = ((VerbAnnotation) ann);
8                  if (!(vAnn.getTense() == null
9                  || "all".equals(vAnn.getTense())
10                 || "all".equals(getTense())
11                 || contains(vAnn.getTense().split(","), getTense())))
12                         return false;
13                 if (!(vAnn.getPerson() == null
14                 || "all".equals(vAnn.getPerson())
15                 || "all".equals(getPerson())
16                 || contains(vAnn.getPerson().split(","), getPerson())))
17                         return false;
18                 if (!(vAnn.getNumber() == null
19                 || "all".equals(vAnn.getNumber())
20                 || "all".equals(getNumber())
21                 || contains(vAnn.getNumber().split(","), getNumber())))
22                         return false;
23                 if (!(vAnn.getVerbType() == null || "all".equals(vAnn.getVerbType())
24                 || "all".equals(getVerbType())
25                 || contains(vAnn.getVerbType().split(","), getVerbType())))
26                         return false;
27                         return true;
28         }
29         return false;
30 }
```

Im Programm werden nun die Sätze der Reihe nach durchgegangen. Dann werden für jeden Satz, die Regeln in der Reihenfolge in der sie in der Datei „rules.dat“ definiert sind durchgegangen. Passen Satz und Regel zusammen wird diese Regel für diesen Satz genommen. Es wird also immer die erste und nur die erste Regel die passt verwendet. Es ist also möglich durch die Reihenfolge der Regeln die Precision zu erhöhen. Passt keine Regel zu einem Satz geht das Programm davon aus, dass in diesem Satz keine gesuchte Aussage gefunden wurde.

6 Test und Testauswertung

In diesem Kapitel wird das im Rahmen dieser Bachelorarbeit erstellte Programm getestet, Hierfür wird zunächst die Frage geklärt, wie kann ein IE -Programm überhaupt bewertet werden. Dann wird getestet, was das erstellen einzelner Regeln bewirkt. Am Ende werden die drei Rennen der Saison 2012 genommen, die den ersten drei Rennen der Saison 2010 entsprechen und es wird überprüft inwieweit hier die Vorhandenen Methoden und Wörterbucheinträge ohne weitere Bearbeitung greifen.
Als Basisinformationen werden in diesem Kapitel, die Inforationen Betrachtet, die nur einmal zu einem Grandprix gehören. Dies sind:

- Datum
- Grandprix
- Gewinner

Als Zusatzinformationen werden die Informationen über die Ereignisse bezeichnet, diese sind:

- win (Gewinnereignis)
- crash (Unfallereignis)
- malfunction (Ausfallereignis)

Eine Übersicht über alle für dieses Programm definierten Regeln ist im Anhang zu finden.

6.1 Bewertung eines Informationsextraktionsprogramms

Nach hansegawa, sekine, grishman gibt es drei Indikatoren für die Qualität eines IE-Programms. Diese sind:

- Recall
- Precision
- F-measure

6.1.1 Recall

Recall gibt an, wie viele richtige Informationen gefunden wurden. Es werden also die richtigen durch die möglichen Ergebnisse geteilt:

$$R = \frac{N_{correct}}{N_{Key}} \tag{6.1}$$

6.1.2 Precision

Precision gibt an wie viele der gefundenen Ergebnisse richtig sind. Es werden also die richtigen durch die gefundenen Ergebnisse geteilt:

$$P = \frac{N_{correct}}{N_{correct} + N_{incorrect}} \tag{6.2}$$

6.1.3 F-measure

F-measure ist eine Kombination der beiden Werte Precision und Recall und wird durch folgende Formel berechnet:

$$F = \frac{2RP}{R + P} \tag{6.3}$$

6.2 Testen des Programms ohne Regeln

Als erstes wird betrachtet, welche Ergebnisse das Programm liefert, wenn noch keine Wörterbuch Einträge oder Regeln vorhanden sind, außer den Einträgen für die Streckennamen.
Richtige Basissachen: 112 von: 171 Gefundene Events: 0 Mögliche Events: 387 Richtige Events: 0
Recall: 0.0 Precision: NaN F-measure: NaN
Overall Recall: 0.20071685 Overall Precision: 0.65497077 Overall F-measure: 0.30727026
Da noch keine Regeln vorhanden sin, können auch keine Ereignisse gefunden werden. Wenn keine Ereignisse gefunden werden, werden auch keine Gewinner gefunden. Bleiben

also nur das Datum und der Grandprix. Hierbei handelt es sich um jeweils 57. Somit gibt es 114 Basisinformationen
Die Testergebnisse zeigen deutlich, dass bei diesem Programm eine solide Methode gefunden wurde Orte und Daten aus dem Text zu ziehen. Da die Gewinner eines Rennes anhand der gefundenen Ereignisse identifiziert werden, können hier noch keine Daten gefunden werden.

6.3 Testen des Programms mit einer Regel

Regeln lassen sich nach dem Komplexitätsgrad ordnen. Betrachten wir also zunächst, wie sich das Programm verhält, wenn man eine sehr einfache Regel hinzufügt.
Folgende Regel wird hinzugefügt: 1. Regel–>Wenn in einem Satz: eine Verbalphrase(Signalverb;3;sg;past) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus: Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
Das Programm leistet nun folgendes: Richtige Basissachen: 126 von: 171 Gefundene Events: 19 Mögliche Events: 387 Richtige Events: 17
Recall: 0.043927647 Precision: 0.8947368 F-measure: 0.08374383
Overall Recall: 0.2562724 Overall Precision: 0.7526316 Overall F-measure: 0.38235295
Diese Regel hat dazu geführt, dass 17 richtige Ereignisse gefunden wurden. Dies hat dazu geführt, dass sich die richtigen Basisinformationen auf 126 erhöht haben. Dies bedeutet, dass 14 richtige Basisinformationen gefunden wurden. Ebenfalls wurden aber auch 2 fehlerhafte Ereignisse identifiziert. Eine sehr einfache, relativ schwammig definierte Regel (kein Kasus oder Genus) führt also dazu, dass mehrere Ereignisse gefunden werden können. Birgt aber auch das Risiko, fehlerhafte Ergebnisse zu bekommen.

6.4 Testen des Programms mit 10 Regeln

Richtige Basissachen: 146 von: 171 Gefundene Events: 76 Mögliche Events: 387 Richtige Events: 61
Recall: 0.15762274 Precision: 0.80263156 F-measure: 0.26349893
Overall Recall: 0.37096775 Overall Precision: 0.8380567 Overall F-measure: 0.5142857
Diese Zehn relativ einfachen und nicht sehr ungenau definierten Regeln, haben dazu geführt, dass 61 richtige und 15 falsche Ergebnisse geliefert wurden. Hier wird die Tendenz deutlich, dass einfache und nicht so genau definierte Regeln zu vielen, aber

nicht immer richtigen Ergebnissen führen. Pro Regel sind hier nur noch 6 Ergebnisse gefunden worden.

6.5 Testen des Programms mit allen Regeln

Richtige Basissachen: 165 von: 171 Gefundene Events: 330 Mögliche Events: 387 Richtige Events: 264
Recall: 0.68217057 Precision: 0.8 F-measure: 0.73640174
Overall Recall: 0.7688172 Overall Precision: 0.8562874 Overall F-measure: 0.81019825
Mit allen 162 im Programm definierten Regeln kommt man lediglich auf 264 richtige Ereignisse. Die Anzahl der fehlerhaften Ergebnisse beträgt 66.

6.6 Die Verbesserung durch weitere Regeln

Abbildung 6.1 zeigt wie sich durch das Hinzufügen von Regeln die Menge der richtigen Basissachen, der gefundenen Events und der richtigen Events verändert hat.

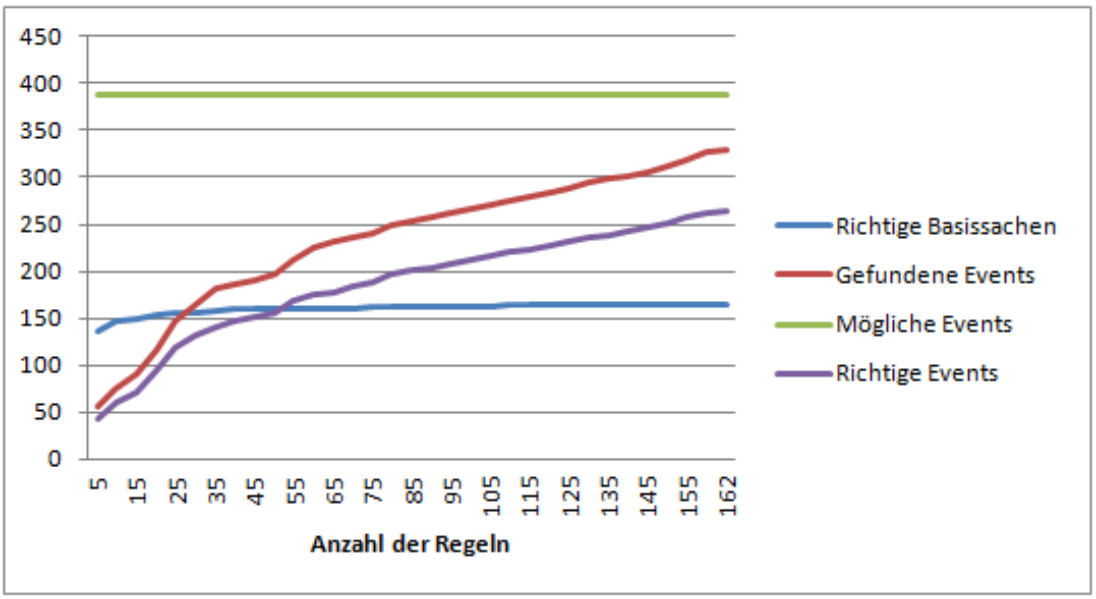


Abb. 6.1: Abbildung 6.1: Verbesserung des Programms durch Hinzufügen von Regeln

Anhand der Grafik wird deutlich, dass bis ungefähr zur 35 Regel, die Menge der Ereignisse sich stark vermehrt. Ab dann ist der Anstieg nicht mehr so stark, aber relativ konstant. Der Verlauf der richtigen Events ist ähnlich, der starke Zuwachs bricht aber etwas früher ein.
Dies ist eine interessante und auch unerwartete Tatsache, die daraufhin deutet, dass bis zu einem Punkt, das hinzufügen von Regeln eine starke Verbesserung zur Folge hat und ab diesem Punkt die Verbesserung nur noch in abgeschwächter, aber trotzdem noch konstanten Form auftritt.
Abbildung 6.2 zeigt den Verlauf von Recall, Precision und F-Measure, wenn nur die Events berücksichtigt werden.

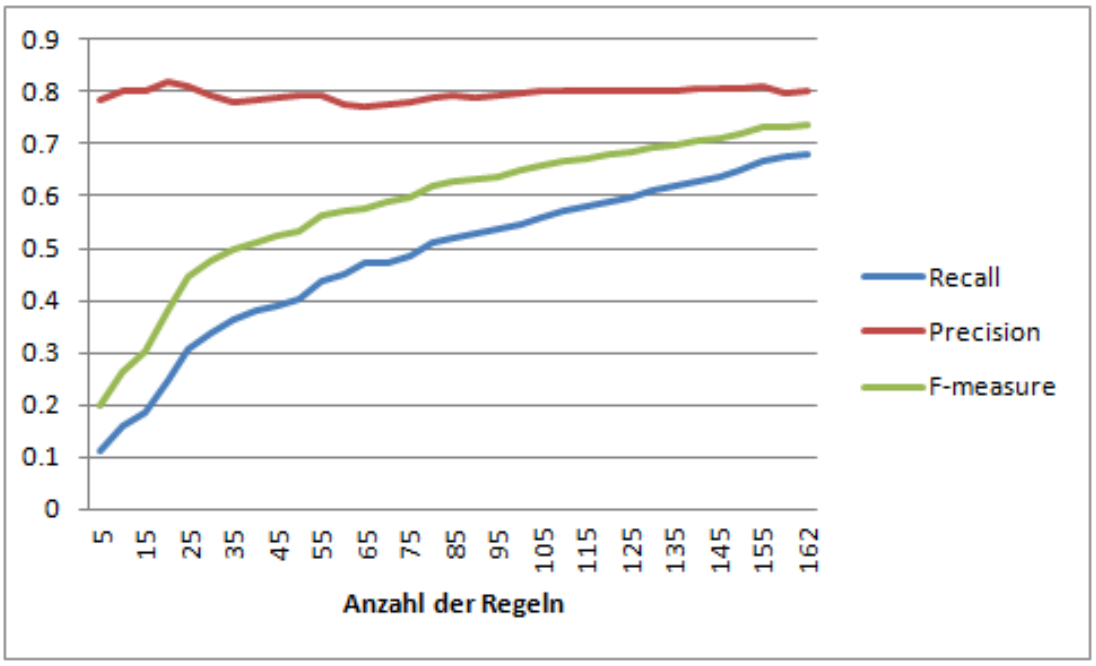


Abb. 6.2: Abbildung 6.2: Veränderung von Recall und Precision der Ereignisse durch Hinzufügen von Regeln

Der Verlauf der Precision ist immer konstant. Die Precision liegt immer um 80% herum. Der Verlauf von Recall fängt erst stark steigend an, bis zu einem bestimmten Punkt, der ungefähr bei 25 liegt. Ab dann steigt der Wert relativ konstant, aber deutlich langsamer weiter. F-measure hat einen ähnlichen Verlauf, da es von Recall und Precision abhängt. Da Precision relativ konstant ist, hängt also F-measure nurnoch von dem Recall ab und hat somit einen nahezu identischen Verlauf.

Dieses Ergebnis zeigt, dass die in die hier entstandene Lösung, einen konstant guten Wert im Bereich der Precision hat. Es gibt zwei unterschiedliche Ansätze sich der IE zu nähern. Der eine ist viel Wert auf Recall zu legen und dann die Precision zu verbessern. Der andere und auch hier verwendete Ansatz ist, einen hohen Wert auf Precision zu legen und nach und nach Recall zu verbessern.
Dies bedeutet also, dass in dieser Arbeit, die Ergebnisse die gefunden werden, sehr zuverlässig sind. Durch das hinzufügen von Regeln wird die Menge der Ergebnisse erhöht. Offensichtlich verbessern oder verschlechtern zusätzliche Regeln die Precision kaum.
Abbildung 6.3 zeigt die Veränderung von Recall und Precision beim Hinzufügen von Regeln unter Berücksichtigung sowohl der Basisinformationen als auch der Ereignisinformationen.

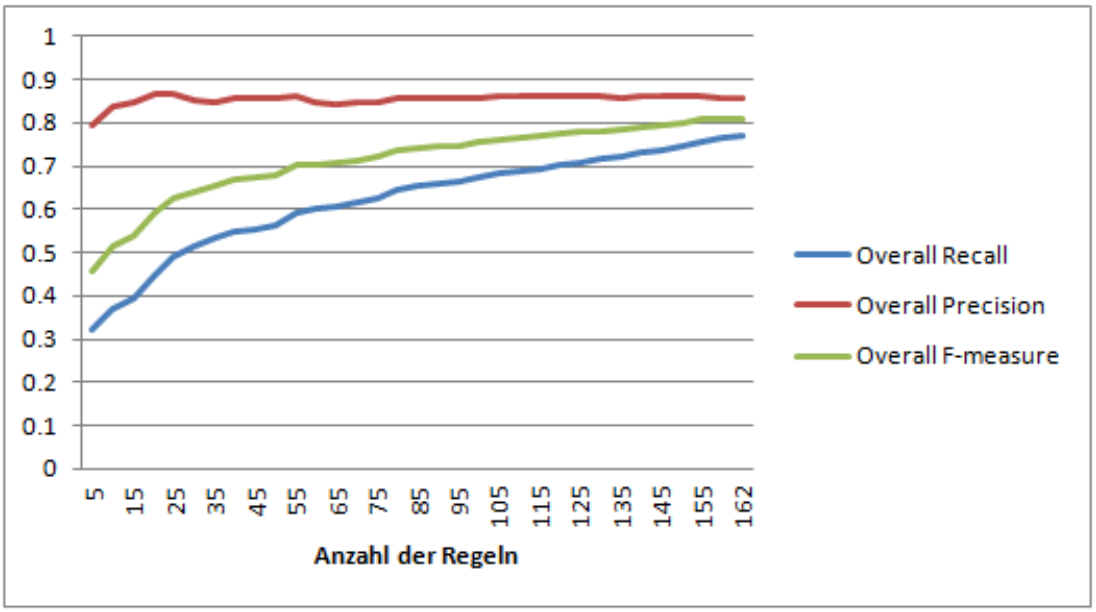


Abb. 6.3: Abbildung 6.3: Veränderung von Recall und Precision aller Informationen durch Hinzufügen von Regeln

Diese Grafik bietet aufgrund des hohen Ähnlichkeitsgrades zur Grafik xx keine weiteren Erkenntnisse, die ihr entnommen werden können.

6.7 Analyse nicht gefundener Ereignisse

In diesem Kapitel werden ein nicht gefundenes Siegereignis und ein nicht gefundenes Unfallereignis betrachtet. Es werden die Gründe gezeigt, warum diese Ereignisse nicht aufgefunden wurden. Danach werden Lösungen vorgestellt und ihre Vor- und Nachteile erläutert.

6.7.1 Beispiel: nicht gefundenes Siegereignis

„Von wegen war es ein Fehler, dass Red Bull keine Stallorder zugunsten von Mark Webber ausgesprochen hat: In einem wenig actionreichen, aber dafür strategisch umso interessanteren Grand Prix von Abu Dhabi sicherte sich Sebastian Vettel heute nicht nur den Sieg, sondern auch den ersten deutschen Fahrer-WM-Titel seit Michael Schumacher im Jahr 2004!“
Wenn man nun betrachtet, welche Satzbausteine identifiziert werden, wird die Problematik etwas deutlicher. Siehe Abbildung 6.4:

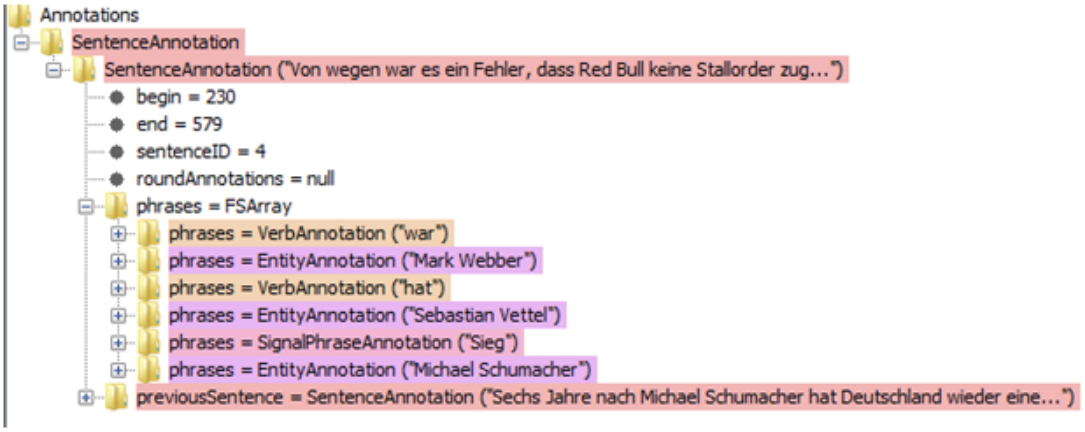


Abb. 6.4: Abbildung 6.4: Satzannotationen vom Beispiel: nicht gefundenes Siegereignis

In dem Satz befinden sich also eine Verbalphrase, eine Entität, eine weitere Verbalphrase, eine weitere Entität, eine Signalphrase und noch eine Entität.
Hier wäre eine Regel möglich, die es einem erlaubt, Sebastian Vettel als Sieger zu identifizieren. Das Problem hierbei ist aber, dass diese Regel sehr speziell wäre und in anderen Kontexten zu Fehlern führen würde.

Handelt es sich bei so einer Satzkonstellation um einen Unfall, könnte es sich um einen Unfall an dem alle drei Fahrer beteiligt sind handel, um einen bei dem zwei der drei Fahrer beteiligt sind oder um einen, wo nur einer der Fahrer beteiligt ist. Es ist also sehr schwierig hier eine eindeutige Regel zu definieren, die für alle Arten von Events gültig ist und nicht zu viele Fehler verursacht.
Ein Beispiel ist:
„In der dritten Runde war ein spannendes Ereignis: Mark Webber hat mit Sebastian Vettel einen schweren Unfall und Michael Schumacher konnte überholen.“
Eine weitere Möglichkeit ist es Regelsprache und Annotationen zu erweitern. Dies hätte zur Folge, dass solche langen und verschachtelten Sätze besser aufzulösen sind. Die großen Nachteile sind aber, dass die Regelsprache komplexer wird, der Aufwand der Berechnung größer wird und die erforderliche Zeit Regeln zu definieren astronomisch hoch wird.

6.7.2 Beispiel: nicht gefundenes Unfallereignis

„Michael Schumacher: Das Rennen war gar nichts. Am Start kam er nicht so gut nach vorne wie gewohnt, dann hat er sukzessive Boden auf Rosberg verloren. Beim Boxenstopp klemmte es dann auch noch, sodass Schumacher bis auf Rang 16 durchgereicht wurde. Zur Krönung dann noch der Crash mit Heidfeld, dessen Schuldfrage ähnlich zu bewerten ist wie bei Webber/Hamilton.“
Hier wird ein besonderes Stilmittel verwendet. Am Anfang des Absatzes wird verdeutlicht, dass sich der Absatz mit Michael Schumacher auseinandersetzt. Betrachtet man die gefundenen Satzbausteine sieht zeigt sich in Abbildung 6.5:
Der Satz besteht aus einer Signalphrase, einer Entität, einer Verbalphrase, einer weiteren Entität und noch einer Entität.
Anhand der Annotationen wird deutlich, dass mit einfachen Regeln hier keine Lösung zu finden ist. Schließlich sind in dem Satz drei weitere Entitäten zu finden, aber die gesuchte Entität ist erst im Satz davor zu finden.
Eine Aussage wie: „Zur Krönung dann noch der Crash mit Heidfeld. . . “ ist mit dem in dieser Arbeit entwickelten Ansatz nicht auszulösen. Sollen solche Aussagen dennoch aufgelöst werden, muss ein System entwickelt werden, welches bemerkt, wenn ein Absatz sich auf eine bestimmte Person bezieht und dem es möglich ist Phrasen wie: „Zur Krönung dann noch der Crash mit Heidfeld“ aufzulösen, als einen Unfall von Heidfeld mit der Person um die es in dem Absatz geht.

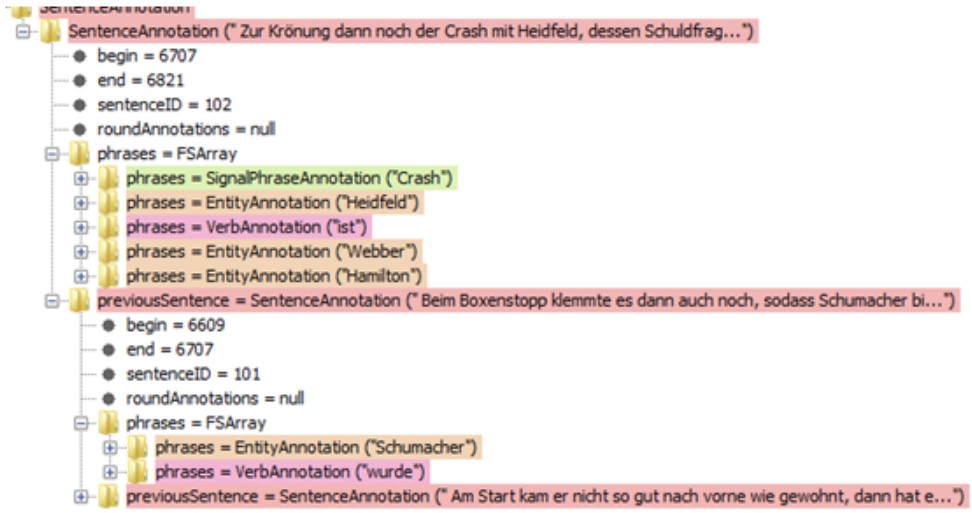


Abb. 6.5: Abbildung 6.5: Satzannotationen vom Beispiel: nicht gefundenes Unfallereignis

6.8 Analyse fehlerhaft identifizierter Ereignisse

In diesem Kapitel werden zwei Textstellen betrachten, bei denen fälschlicher Weise einmal ein Gewinnereignis und einmal ein Unfallereignis gefunden wurden. Es wird erklärt, wieso an dieser Stelle das Programm fehlerhaft arbeitet. Es werden Lösungen vorgestellt diese Fehler zu vermeiden und diese werden dann auf ihre Vor- und Nachteile genauer betrachtet.

6.8.1 Beispiel: fehlerhaft identifiziertes Gewinnereignis

„Der große Verlierer der Safety-Car-Phase hieß Ferrari. Fernando Alonso bremste regelgerecht hinter dem Führungsfahrzeug ab, verlor dabei aber viel Zeit. Viele Piloten, die zuvor hinter dem Lokalmatador lagen, waren noch nicht an der Box vorbei und konnten durch einen frühen Stopp Plätze gutmachen. Der erste Gewinner hieß Rubens Barrichello.“
Der eigentliche Satz um den es hier geht ist:
„Der erste Gewinner hieß Rubens Barrichello.“.
Die vom Programm erstellten Annotationen sehen wie in Abbildung 6.6 gezeigt aus:
Die identifizierten Satzbausteine sind also: eine Signalphrase, eine Verbalphrase und eine Entität. Die Regel die hierfür passt ist folgende: Wenn in einem Satz: eine Signalphrase,

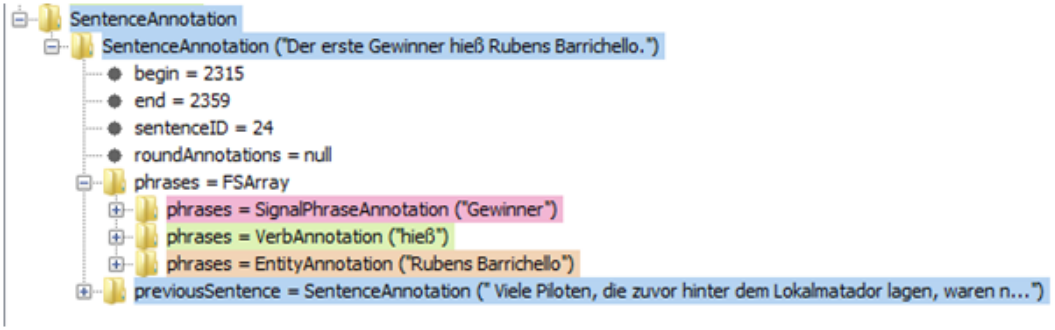


Abb. 6.6: Abbildung 6.6: Satzannotationen vom Beispiel: fehlerhaft identifiziertes Gewinnereignis

eine Verbalphrase(Phrasenverb;3;sg;past) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus: Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
Diese Regel führt fast immer zu richtigen Ergebnissen. In diesem Fall aber, ist das Gewinnen nicht auf das Rennen bezogen, sondern auf die vorher angesprochene Safety-Car-Phase.
Eine Lösung für dieses Problem wäre es, wenn man alle Signalphrasen noch mal auf ihre Gültigkeit überprüft. Das bedeutet eine Überprüfung, auf was sich eine Signalphrase oder ein Signalverb eigentlich beziehen. Dies erfordert aber ein viel genaueres Textverständnis. Es müssten also viel komplexere Analysen durchgeführt werden.
Die Häufigkeit solcher Fehler ist sehr gering. Der Aufwand diese zu verhindern aber sehr hoch. Aufgrund dieser beiden Tatsachen empfiehlt es sich diese Fehler zu tolerieren.

6.8.2 Beispiel: fehlerhaft identifiziertes Unfallereignis

„In diese Kollision wäre übrigens beinahe auch Button hineingezogen worden.“
Die in diesem Satz definierten Satzbausteine sind die in Abbildung 6.7 gezeigten.
Hier sind eine Signalphrase und eine Entität als Satzbausteine gefunden worden. Die dazugehörige Regel, die dazu führte, dass das Ereignis fälschlicherweise gefunden wurde, ist folgende:
Wenn in einem Satz: eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus: Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
Diese Regel ist sehr einfach und führt fast immer zu richtigen Ergebnissen. Das Problem in diesem Satz ist, dass hier der Konjunktiv verwendet wird.

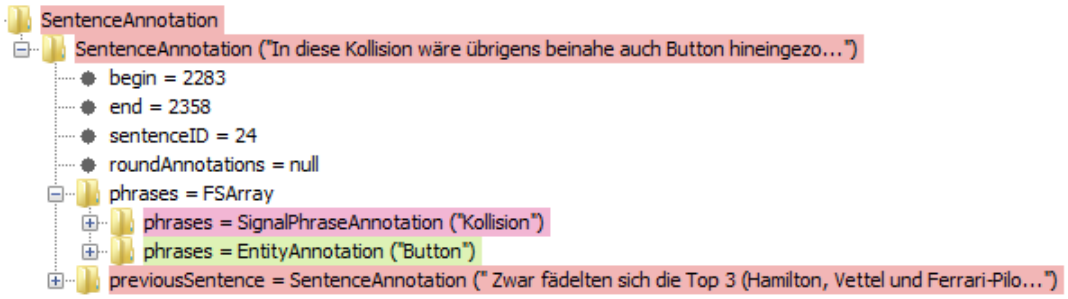


Abb. 6.7: Abbildung 6.7: Satzannotationen vom Beispiel: fehlerhaft identifiziertes Unfallereignis

Eine Lösung dieses Problems ist es, wenn man im Wörterbuch „wäre“ als Konjunktivform von „sein“ mit einträgt. Dies hätte zur Folge, dass das Verb „sein“ ebenfalls als Satzbaustein dazukommt und die Regel nicht mehr greifen kann. Wird dann noch bei allen Regeln die Konjunktivform ausgeschlossen, dann können solche Fehler nicht mehr auftreten.
Die entscheidende Frage ist also, wie viele Wörter und welche in das Wörterbuch aufgenommen werden sollten. Da in dieser Arbeit das Erstellen eines Programms und nicht die perfekte Definition von Regeln und Wörterbucheinträgen im Vordergrund stand, kann aus zeitlichen Gründen hier keine fundierte und durch empirische Tests bestätigte Aussage getätigt werden.

6.9 Testen des Programms mit drei Texten von 2012

Die drei hier gewählten Texte über Rennen aus dem Jahr 2012 sind die Rennen die an den gleichen Orten stattfanden, wie die ersten drei Rennen der Saison 2010. Es wurden, außer bei den Fahrernamen, keine weiteren Wörterbucheinträge gemacht. Die Regeln blieben komplett unverändert.
Richtige Basissachen: 9 von: 9 Gefundene Events: 10 Mögliche Events: 15 Richtige Events: 8
Recall: 0.53333336 Precision: 0.8 F-measure: 0.64000005
Overall Recall: 0.7083333 Overall Precision: 0.8947368 Overall F-measure: 0.79069763
Die Ergebnisse zeigen deutlich, dass die gefundenen Regeln und Wörterbucheinträge eine solide Basis darstellen, um Informationen aus einem Formel-1-Bericht zu ziehen. Das Auffinden der Basisinformationen, wie der Ort, das Datum und der Gewinner haben

sich als besonders effizient herausgestellt. Die Lösung zum identifizieren der Ereignisse ist akzeptabel, da selbst bei Texten, für die die Regeln und Wörterbucheinträge nicht erstellt wurden, mehr als die Hälfte der Ereignisse gefunden wurde.

7 Bewertung und Ausblick

In diesem Kapitel wird zunächst betrachtet, wie die Entwicklung des Prototyps verlaufen ist und welche Schwierigkeiten dabei auftraten. Danach wird der Stand der Arbeit zusammengefasst. Am Ende wird ein Ausblick darüber gegeben, wie der Prototyp verbessert werden könnte, wofür der Prototyp verwendet werden kann.

7.1 Rückblick

Bei der Entwicklung dieser Arbeit wurden viele Erkenntnisse gewonnen. Zunächst war nur geplant, Sätze auf Signalwörter und Entitäten zu reduzieren. Das dies ein sehr schwaches Konzept sei, wurde schnell klar. Es hat sich herausgestellt, dass mindestens Verbalphrasen und Nominalphrasen definiert sein müssen.
Eine weitere wesentliche Erkenntnis ist, dass die Art wie ein Text geschrieben ist, sich signifikant auf das Ergebnis der IE auswirkt. So wird es nahezu unmöglich aus Texten, bei denen besonders viele lange und verschachtelte Sätze auftreten, Informationen zu extrahieren. Ebenfalls sind Texte, wo sich ganze Absätze auf eine Person beziehen, es dem Programm sehr erschweren richtige Ergebnisse zu liefern.
Außerdem wurde deutlich, dass eine geringe Menge an Aufwand - was Wörterbucheinträge und Regeln betrifft – die meisten Ergebnisse liefert. Je dichter man dem Maximum an Precision und Recall kommt, desto komplexer und aufwendiger werden die Regeln.
Schließlich hat sich herausgestellt, dass einfache Informationen, wie ein Datum oder ein Rennort mit sehr einfachen Mittel aus einem Text herauszuholen sind. Hat man aber komplexere Ereignisse, an denen im Zweifel auch mehrere Entitäten beteiligt sein können, benötigt man deutlich kompliziertere Verfahren. Fragen zu klären, wer an einem Unfall schuld hat, erfordern ein solches Textverständnis, das die Entwicklung eines solchen Programms, dass allein aus Zeitgründen, die Entwicklung eines solchen Systems nicht im Rahmen einer Bachelorarbeit möglich ist.

7.2 Bewertung

Die in Kapitel 1.2 beschriebenen Ziele waren vor allem ein dynamisches und ebenfalls effizientes Programm zu erschaffen.
Als erstes wird betrachtet, wie dynamisch das Programm geworden ist. Was die Frage, des Ortes angeht, ist das Programm sehr statisch. Es ist lediglich möglich einen Ort für einen Text zu bestimmen. Hier könnte man eine deutlich flexiblere Lösung finden, die ähnlich, wie die Zeitlösung in das Regelsystem integriert wird.
Bei der Zeitinformation ist eine sehr flexible Lösung gefunden worden. Es ist möglich die Zeit eines Ereignisses in jeder erdenklichen Art zu extrahieren. Lediglich die speziell für diese Arbeit entwickelte Lösung, was die relativen Rundenangaben – vor allem auf die Gesamtrundenzahl bezogen – ist hier nur für Rundenangaben anwendbar.
Die Lösung der Phrasen ist im Großen und Ganzen sehr dynamisch. Es ist möglich jede Verbformen einzupflegen. Selbst eigens kreierte Formen , wie Singalverben oder Phrasenverben, kann man erstellen. Gleiches gilt für Nominalphrasen. Auch hier ist es möglich unterschiedliche Formen, wie Entität oder Synonym zu verwenden.
Da Signalphrasen und Nichtphrasen keine Attribute – außer der Ereignisart bei Signalphrasen - haben, kann ist es ebenfalls sehr dynamisch.
Was die Frage betrifft, ob das Programm auf andere Ereignisse erweiterbar ist, so ist diese klar mit einem Ja zu beantworten. Dies zeigt allein die Tatsache, dass im Laufe der Entwicklung die gewünschten Informationen um die Ausfälle erweitert wurden, dies hatte keine Änderung am Programm zur Folge. Es musste nur das Wörterbuch angepasst werden.
Das Konzept, dass Beteiligten unterschiedlichste Rollen zugewiesen werden können, macht es ebenfalls sehr dynamisch. So könnte auch Ereignisse aus einem Text gezogen werden, bei denen die Beteiligten in unterschiedlichen Rollen agieren. Wie zum Beispiel Opfer und Täter.
Bei der Frage der Effizienz sind die Testergebnisse genauer zu betrachten. Hier liefert das Programm ganz gute Werte, unter Berücksichtigung, dass es nur ein Prototyp ist und keine monatelange Entwicklungsphase der Regeln und des Wörterbuches möglich war.
Schließlich kann man sagen, dass das Programm ein effizientes und dynamisches Verhalten hat. Es zeigt aber auch, dass es Schwierig ist bestimmte Informationen aus einem Text zu ziehen. Ebenfalls erschwert das Erweitern um weitere Events die Informationsextraktion

erheblich. Diese Tatsache verdeutlicht, dass ein vollständiges Textverständnis eine große Herausforderung darstellt.

7.3 Ausblick

Für den hier entwickelten Prototypen gibt es eine Vielzahl an Erweiterungs- und Ergänzungsmöglichkeiten.
Zum einen besteht die Möglichkeit, die Regeln und Wörterbucheinträge so zu erweitern, dass bessere Werte für Recall und Precision erreicht werden. Hier könnten Tests durchgeführt werden, um herauszufinden, wie sich bestimmte Änderungen im Wörterbuch oder in den Regeln sich auf die Werte für Recall und Precision auswirken.
Außerdem kann der Prototyp im Bereich der Ortserkennung verbessert werden. Man könnte das Programm an dieser Stelle dynamischer machen, so dass nicht nur Grandprix-Orte unterstützt werden, sondern auch andere Orte erkannt werden. Dieser Teil kann auch in die Regeln integriert werden. Man kann aber auch einen Ansatz untersuchen, bei dem einem Satz eine Zeit und oder ein Ort zugewiesen wird.
Des Weiteren wäre es auch möglich, beide Ansätze zu implementieren und diese dann umfangreich auf ihre Tauglichkeit zu überprüfen. Hierfür könnte man einige Testreihen durchführen, um herauszufinden, welcher Ansatz der bessere ist.
Man könnte das Konzept der Verbalphrasen überarbeiten, damit zusammengesetzte Verben besser unterstützt werden. Hier liegt noch ein Schwachpunkt des aktuellen Systems, welcher durch eine geschickte Lösung der Verbalphrasenfrage gelöst werden kann.
Eine weitere interessante Möglichkeit wäre, ein Programm zu schreiben, welches anhand von einem annotierten Text Regeln erstellt. Hier könnte auch überprüft werden, welche Regeln eher dazu neigen, Fehler zu verursachen, und welche Regeln eher richtige Werte liefern.
Schließlich wäre es möglich, das Programm um weitere Wortarten zu erweitern, wie zum Beispiel Adjektive. Hier ist zu erwarten, dass die Regeln zu komplex werden. Diese Behauptung ist durch empirische Tests nachzuweisen oder zu wiederlegen.

8 Anhang

1. Regel–>Wenn in einem Satz: eine Verbalphrase(Signalverb;3;sg;past) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
2. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;3;sg;present) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1);
3. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1);
4. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;3;sg;present) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1);
5. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1);
6. Regel–>Wenn in einem Satz: eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
7. Regel–>Wenn in einem Satz: eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
8. Regel–>Wenn in einem Satz: eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
9. Regel–>Wenn in einem Satz: eine Verbalphrase(Hilfsverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;all;all;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2);

10. Regel–>Wenn in einem Satz: eine Verbalphrase(Hilfsverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2);
11. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2);
12. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2);
13. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
14. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;past) und eine Verbalphrase(Signalverb;all;all;infinitive) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
15. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(sein;3;sg;past) und eine Verbalphrase(Signalverb;all;all;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
16. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present) und eine Verbalphrase(Signalverb;all;all;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
17. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(Signalverb;3;sg;past) und eine Nominalphrase(Entität;dat;sg;male) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
18. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(Signalverb;3;sg;present) und eine Nominalphrase(Entität;nom;sg;male) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1);
19. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Signalverb;3;sg;present) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann

folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
20. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;present) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
21. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;male), eine Verbalphrase(Phrasenverb;3;sg;past) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
22. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
23. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;past) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
24. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;dat;sg;male), eine Nominalphrase(Entität;dat;sg;male) und eine Verbalphrase(Signalverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2);
25. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
26. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
27. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
28. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
29. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);

30. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
31. Regel–>Wenn in einem Satz: eine Signalphrase, eine Verbalphrase(Phrasenverb;3;sg;past) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
32. Regel–>Wenn in einem Satz: eine Signalphrase, eine Nominalphrase(Entität;all;sg;all) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
33. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Nichtphrase, eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);
34. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present) und eine Verbalphrase(Signalverb;all;all;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:2);
35. Regel–>Wenn in einem Satz: eine Verbalphrase(Hilfsverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;all;all;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
36. Regel–>Wenn in einem Satz: eine Verbalphrase(Signalverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
37. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Zeitphrase und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2);
38. Regel–>Wenn in einem Satz: eine Verbalphrase(Hilfsverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Signalphrase und eine Verbalphrase(Phrasenverb;all;all;infinitive) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2);
39. Regel–>Wenn in einem Satz: eine Verbalphrase(Hilfsverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all), eine Signalphrase und eine Verbalphrase(Phrasenverb;3;sg;participleII)

vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2);
40. Regel–>Wenn in einem Satz: eine Verbalphrase(sein;all;all;all), eine Nominalphrase(Entität;all;sg;all), eine Signalphrase und eine Verbalphrase(Hilfsverb;3;sg;present) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2);
41. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Signalphrase und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2);
42. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;present), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:3);
43. Regel–>Wenn in einem Satz: eine Verbalphrase(Hilfsverb;3;sg;present), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
44. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;past), eine Verbalphrase(Signalverb;3;sg;past) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);
45. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present), eine Verbalphrase(Signalverb;3;sg;present) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4);
46. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(Hilfsverb;3;sg;present), eine Nominalphrase(Entität;nom;sg;male) und eine Verbalphrase(Signalverb;all;all;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:1);
47. Regel–>Wenn in einem Satz: eine Nominalphrase(Pronomen;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;all;all;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:3);

48. Regel–>Wenn in einem Satz: eine Nominalphrase(Pronomen;all;all;all), eine Verbalphrase(Hilfsverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:3);
49. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;past), eine Zeitphrase und eine Verbalphrase(Signalverb;all;all;infinitive) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
50. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(Signalverb;3;sg;past), eine Zeitphrase und eine Nominalphrase(Entität;dat;sg;male) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4);
51. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Zeitphrase und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
52. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;male), eine Verbalphrase(Hilfsverb;3;sg;present), eine Signalphrase und eine Verbalphrase(Phrasenverb;3;sg;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
53. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase und eine Nominalphrase(Entität;nom;sg;male) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4);
54. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Hilfsverb;3;sg;past), eine Signalphrase und eine Nominalphrase(Entität;dat;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4);
55. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present), eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4);
56. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase und eine Signalphrase vorkommen, dann

folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
57. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Nichtphrase und eine Verbalphrase(Signalverb;3;sg;present) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
58. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present) und eine Verbalphrase(Signalverb;all;all;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:2);
59. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Signalphrase und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
60. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1);
61. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Verbalphrase(Signalverb;3;sg;past), eine Nominalphrase(Entität;nom;sg;all) und eine Nominalphrase(Entität;dat;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);
62. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Signalverb;3;sg;present) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:4);
63. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;present) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
64. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;present) und eine Signalphrase vorkommen, dann

folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
65. Regel–>Wenn in einem Satz: eine Signalphrase, eine Verbalphrase(Hilfsverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;all;all;infinitive) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:3);
66. Regel–>Wenn in einem Satz: eine Signalphrase, eine Verbalphrase(Hilfsverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;all;all;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:3);
67. Regel–>Wenn in einem Satz: eine Signalphrase, eine Verbalphrase(all;all;all;all), eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
68. Regel–>Wenn in einem Satz: eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Phrasenverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
69. Regel–>Wenn in einem Satz: eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Phrasenverb;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
70. Regel–>Wenn in einem Satz: eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);
71. Regel–>Wenn in einem Satz: eine Signalphrase, eine Nominalphrase(Synonym;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);
72. Regel–>Wenn in einem Satz: eine Verbalphrase(sein;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Pronomen;all;pl;all) und eine Verbalphrase(Signalverb;3;pl;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);

73. Regel–>Wenn in einem Satz: eine Verbalphrase(sein;3;sg;past), eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Zeit-Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);
74. Regel–>Wenn in einem Satz: eine Verbalphrase(werden;3;sg;past), eine Signalphrase, eine Nominalphrase(Entität;gen;sg;all), eine Nominalphrase(Synonym;dat;sg;all) und eine Nominalphrase(Entität;nom;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:5);
75. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present), eine Verbalphrase(Signalverb;all;all;infinitive), eine Nominalphrase(Entität;all;all;all) und eine Nominalphrase(Entität;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
76. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present), eine Nominalphrase(Synonym;dat;sg;all), eine Nominalphrase(Entität;dat;sg;all) und eine Verbalphrase(Signalverb;all;all;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Beteiligter(Satz:1,Wort:1);
77. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Signalverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1);
78. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Hilfsverb;3;sg;past), eine Signalphrase, eine Nominalphrase(Entität;dat;sg;all) und eine Verbalphrase(Phrasenverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4);
79. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4);
80. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase, eine Zeitphrase und eine Signalphrase

vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:1);
81. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Signalverb;3;sg;past), eine Nichtphrase und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
82. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all),eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Signalverb;3;sg;present), eine Nominalphrase(Synonym;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2);
83. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Signalverb;3;sg;present) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:5);
84. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Signalphrase, eine Verbalphrase(all;all;all;all), eine Nominalphrase(Pronomen;all;sg;all) und eine Nichtphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1);
85. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:4);
86. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Signalverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:4);
87. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Pronomen;all;sg;all), eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:5);

88. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);
89. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Zeitphrase und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:6); Zeit-Signalwort(Satz:1,Wort:5); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
90. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Pronomen;all;all;all), eine Nominalphrase(Pronomen;all;all;all) und eine Verbalphrase(Signalverb;3;pl;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:6); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
91. Regel–>Wenn in einem Satz: eine Verbalphrase(Hilfsverb;3;sg;present), eine Signalphrase, eine Nominalphrase(all;all;sg;all), eine Verbalphrase(Signalverb;3;sg;past), eine Nominalphrase(Synonym;all;all;all) und eine Nominalphrase(Entität;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
92. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present), eine Verbalphrase(Phrasenverb;3;sg;present), eine Nominalphrase(Synonym;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:6); Beteiligter(Satz:1,Wort:1);
93. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(sein;3;sg;past),
eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Phrasenverb;all;all;participleII), eine Nominalphrase(Pronomen;nom;pl;all) und eine Verbalphrase(Signalverb;3;pl;past) vorkommen,
dann folgt daraus:
Signalwort(Satz:1,Wort:6); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
94. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Signalverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all), eine Nominal-

phrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1);
95. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Synonym;dat;all;all), eine Nominalphrase(Entität;dat;all;all), eine Signalphrase und eine Verbalphrase(werden;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Beteiligter(Satz:1,Wort:1);
96. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(sein;3;sg;past), eine Zeitphrase, eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Signalverb;all;all;participleII) und eine Verbalphrase(Hilfsverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4);
97. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Signalverb;3;sg;past), eine Zeitphrase, eine Nominalphrase(Entität;dat;sg;all), eine Signalphrase und eine Verbalphrase(Signalverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4);
98. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;present), eine Zeitphrase, eine Signalphrase, eine Verbalphrase(Signalverb;3;sg;present) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
99. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase, eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:6);
100. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase, eine Nominalphrase(Entität;dat;sg;all), eine Signalphrase und eine Verbalphrase(Phrasenverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4);

101. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(werden;3;sg;past), eine Signalphrase, eine Zeitphrase, eine Nominalphrase(Entität;nom;sg;male) und eine Verbalphrase(Phrasenverb;all;all;participleII) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:5);
102. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(all;all;all;all), eine Verbalphrase(Phrasenverb;3;sg;past) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:6); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
103. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present), eine Verbalphrase(Phrasenverb;3;sg;present) und eine Nominalphrase(Pronomen;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
104. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Signalphrase und eine Nominalphrase(Pronomen;all;pl;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);
105. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Signalphrase, eine Nominalphrase(Synonym;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;past) und eine Nichtphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4);
106. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(all;all;all;all) und eine Nominalphrase(Entität;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);
107. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(sein;3;sg;past), eine Zeitphrase, eine Verbalphrase(Signalverb;3;sg;past), eine Verbalphrase(all;all;all;all), eine Nominalphrase(Synonym;all;all;all) und eine Nominal-

phrase(Entität;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
108. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Zeitphrase, eine Nominalphrase(Synonym;all;sg;all), eine Nominalphrase(Pronomen;all;sg;all), eine Verbalphrase(Signalverb;3;sg;past) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:6); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
109. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;present), eine Signalphrase, eine Verbalphrase(Phrasenverb;3;sg;present), eine Nominalphrase(Synonym;all;sg;all), eine Verbalphrase(all;all;all;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
110. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase, eine Nominalphrase(Synonym;all;sg;all), eine Signalphrase, eine Nominalphrase(Synonym;all;sg;all) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1);
111. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase, eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(all;all;all;all) und eine Nichtphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:5);
112. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Synonym;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present), eine Signalphrase, eine Nominalphrase(Pronomen;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;present) und eine Nominalphrase(Pronomen;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:1);
113. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Verbalphrase(Signalverb;3;sg;past), eine Nominalphrase(Entität;nom;sg;all), eine Nominalphrase(Entität;akk;sg;all), eine Nominalphrase(Pronomen;nom;sg;all), eine Zeitphrase und eine Verbalphrase(sein;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);
114. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all),

eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(werden;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:5);
115. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Nominalphrase(Synonym;nom;sg;male), eine Nominalphrase(Entität;nom;sg;male), eine Nominalphrase(Synonym;nom;sg;male), eine Verbalphrase(Phrasenverb;3;sg;past), eine Zeitphrase und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:7); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
116. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Signalphrase, eine Verbalphrase(all;all;all;all), eine Nichtphrase und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
117. Regel–>Wenn in einem Satz: eine Signalphrase, eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Hilfsverb;3;sg;past), eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:6); Beteiligter(Satz:1,Wort:7);
118. Regel–>Wenn in einem Satz: eine Signalphrase, eine Verbalphrase(all;all;all;all), eine Nominalphrase(Entität;all;sg;all), eine Signalphrase, eine Verbalphrase(all;all;all;all), eine Verbalphrase(all;all;all;all), eine Signalphrase und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
119. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;nom;sg;all) und eine Verbalphrase(Signalverb;3;sg;past) vorkommen, in dem Satz davor: eine Verbalphrase(Hilfsverb;3;sg;past), eine Nominalphrase(Pronomen;nom;sg;all), eine Nominalphrase(Entität;nom;sg;all) und eine Nominalphrase(Entität;nom;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:2,Wort:4);
120. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;nom;sg;all) und eine Verbalphrase(Signalverb;3;sg;past) vorkommen, in dem Satz davor: eine Verbalphrase(Hilfsverb;3;sg;past), eine Nominalphrase(Pronomen;nom;sg;all), eine Nominalphrase(Entität;nom;sg;all), eine Nominalphrase(Synonym;dat;sg;all) und eine Nominalphra-

se(Entität;nom;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:2,Wort:5);
121. Regel–>Wenn in einem Satz: eine Nominalphrase(Pronomen;nom;pl;all) und eine Verbalphrase(Signalverb;3;pl;past) vorkommen, in dem Satz davor: eine Verbalphrase(all;all;all;all), eine Nominalphrase(Entität;nom;sg;all), eine Nominalphrase(Entität;nom;sg;all), eine Signalphrase und eine Nominalphrase(Synonym;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:2,Wort:2); Beteiligter(Satz:2,Wort:3);
122. Regel–>Wenn in einem Satz: eine Verbalphrase(Hilfsverb;3;sg;present), eine Nominalphrase(Pronomen;all;sg;all) und eine Verbalphrase(Signalverb;all;all;infinitive) vorkommen, in dem Satz davor: eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(all;all;all;all) und eine Nichtphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:2,Wort:1); Beteiligter(Satz:2,Wort:2);
123. Regel–>Wenn in einem Satz: eine Verbalphrase(Hilfsverb;3;sg;present), eine Signalphrase und eine Nominalphrase(Synonym;all;sg;all) vorkommen, in dem Satz davor: eine Zeitphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:2,Wort:1); Beteiligter(Satz:2,Wort:2);
124. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Signalverb;3;sg;past) und eine Nominalphrase(Synonym;all;sg;all) vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;all) und eine Nominalphrase(Pronomen;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:2,Wort:1); Beteiligter(Satz:1,Wort:1);
125. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;all;sg;all), eine Zeitphrase und eine Verbalphrase(Signalverb;3;sg;past) vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(sein;3;sg;past) und eine Zeitphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:1,Wort:2); Beteiligter(Satz:2,Wort:1);
126. Regel–>Wenn in einem Satz: eine Signalphrase, eine Verbalphrase(Hilfsverb;3;sg;past) und eine Nominalphrase(Synonym;all;sg;all) vorkommen, in dem Satz davor: eine Verbalphrase(all;all;all;all) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:2,Wort:2);
127. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Synonym;gen;sg;all), eine Zeitphrase und eine Signalphrase vorkommen, in

dem Satz davor: eine Nominalphrase(Entität;nom;sg;all) vorkommen, dann folgt daraus: Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:2,Wort:1);
128. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;dat;sg;all), eine Verbalphrase(sein;3;sg;past), eine Zeitphrase und eine Signalphrase vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;all;all), eine Verbalphrase(all;all;all;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:2,Wort:1);
129. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;nom;sg;all), eine Verbalphrase(Hilfsverb;3;sg;past), eine Signalphrase und eine Zeitphrase vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:1,Wort:4); Beteiligter(Satz:2,Wort:1);
130. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;nom;sg;male), eine Nominalphrase(Synonym;dat;sg;male), eine Verbalphrase(Phrasenverb;3;sg;past) und eine Signalphrase vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;male) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:2,Wort:1);
131. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;all;all), eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Signalverb;3;sg;past) und eine Signalphrase vorkommen, in dem Satz davor: eine Zeitphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:2,Wort:1); Beteiligter(Satz:1,Wort:2);
132. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Signalverb;3;sg;present) vorkommen, in dem Satz davor: eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(all;all;all;all) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:2,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:2,Wort:4);
133. Regel–>Wenn in einem Satz: eine Nominalphrase(Pronomen;all;sg;all), eine Nominalphrase(Pronomen;all;sg;all), eine Signalphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4); Beteiligter(Satz:2,Wort:1);
134. Regel–>Wenn in einem Satz: eine Nominalphrase(Pronomen;all;sg;all), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all) und eine Nominalphrase(Pronomen;all;sg;all) vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;all) vorkommen, dann

folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3);
135. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Verbalphrase(Hilfsverb;3;sg;past), eine Nominalphrase(Pronomen;nom;sg;male) und eine Verbalphrase(Signalverb;all;all;infinitive) vorkommen, in dem Satz davor: eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(all;all;all;all) und eine Nichtphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:2,Wort:1);
136. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Verbalphrase(Signalverb;3;pl;past), eine Nominalphrase(Pronomen;nom;pl;male) und eine Nominalphrase(Entität;nom;sg;male) vorkommen, in dem Satz davor: eine Nominalphrase(Synonym;nom;sg;male), eine Nominalphrase(Entität;nom;sg;male) und eine Nominalphrase(Entität;nom;sg;male) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:2,Wort:2); Beteiligter(Satz:2,Wort:3);
137. Regel–>Wenn in einem Satz: eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past) und eine Nominalphrase(Synonym;all;sg;all) vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(all;all;all;all), eine Nominalphrase(Pronomen;all;sg;all) und eine Zeitphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:2,Wort:1);
138. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;nom;sg;male), eine Verbalphrase(Hilfsverb;3;sg;past), eine Verbalphrase(Signalverb;all;all;infinitive), eine Nominalphrase(Pronomen;nom;sg;male) und eine Zeitphrase vorkommen, in dem Satz davor: eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(all;all;all;all) und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:1,Wort:5); Beteiligter(Satz:2,Wort:1);
139. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Signalphrase und eine Verbalphrase(Hilfsverb;3;sg;past) vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:2,Wort:1);
140. Regel–>Wenn in einem Satz: eine Nominalphrase(Pronomen;nom;sg;male), eine Verbalphrase(Phrasenverb;1;sg;past), eine Zeitphrase, eine Signalphrase und eine Nominalphrase(Entität;nom;sg;male) vorkommen, in dem Satz davor: eine Verbalphra-

se(all;all;all;all) und eine Nominalphrase(Entität;nom;sg;male) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:4); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:2,Wort:2); Beteiligter(Satz:1,Wort:5);
141. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Synonym;nom;sg;male), eine Nominalphrase(Entität;dat;sg;male) und eine Verbalphrase(Signalverb;3;sg;participleII) vorkommen, in dem Satz davor: eine Zeitphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4); Beteiligter(Satz:2,Wort:2);
142. Regel–>Wenn in einem Satz: eine Signalphrase, eine Verbalphrase(Hilfsverb;3;sg;present), eine Nominalphrase(Synonym;all;sg;all), eine Nominalphrase(Synonym;all;sg;all) und eine Nominalphrase(Entität;all;sg;all) vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Beteiligter(Satz:2,Wort:1);
143. Regel–>Wenn in einem Satz: eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Zeitphrase, eine Nominalphrase(Pronomen;all;sg;all) und eine Verbalphrase(all;all;all;all) vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:2,Wort:1);
144. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;nom;sg;all), eine Verbalphrase(sein;3;sg;past), eine Zeitphrase, eine Verbalphrase(Hilfsverb;3;sg;past), eine Signalphrase und eine Verbalphrase(Phrasenverb;all;all;infinitive) vorkommen, in dem Satz davor: eine Verbalphrase(sein;all;all;all) und eine Nominalphrase(Entität;nom;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:2,Wort:2);
145. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase, eine Signalphrase, eine Nominalphrase(Synonym;nom;sg;male) und eine Verbalphrase(Hilfsverb;3;sg;past) vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;male), eine Verbalphrase(all;all;all;all) und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:2,Wort:1);

146. Regel–>Wenn in einem Satz: eine Nominalphrase(Pronomen;nom;sg;male), eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Signalverb;all;all;participleII), eine Verbalphrase(werden;3;sg;past), eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Pronomen;nom;sg;male), eine Zeitphrase und eine Signalphrase vorkommen, in dem Satz davor: eine Nominalphrase(Entität;nom;sg;male) und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:1,Wort:7); Beteiligter(Satz:1,Wort:2); Beteiligter(Satz:2,Wort:1);
147. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;present) und eine Signalphrase vorkommen, in dem Satz davor: no phrase vorkommen, in dem Satz davor: eine Zeitphrase und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:3,Wort:1); Beteiligter(Satz:3,Wort:2);
148. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Synonym;all;sg;all), eine Signalphrase und eine Signalphrase vorkommen, in dem Satz davor: eine Nominalphrase(Synonym;all;sg;all), eine Verbalphrase(all;all;all;all) und eine Zeitphrase vorkommen, in dem Satz davor: eine Verbalphrase(all;all;all;all), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:2,Wort:3); Beteiligter(Satz:3,Wort:3);
149. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;all;sg;all), eine Zeitphrase, eine Signalphrase, eine Verbalphrase(all;all;all;all) und eine Verbalphrase(all;all;all;all) vorkommen, in dem Satz davor: eine Zeitphrase vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Zeit-Signalwort(Satz:1,Wort:2); Beteiligter(Satz:3,Wort:1);
150. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Signalverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(Phrasenverb;3;sg;present), eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1); und es folgt daraus:
Signalwort(Satz:1,Wort:6); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:5);
151. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;all;male), eine Verbalphrase(Phrasenverb;3;sg;past), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Signalphrase und eine Nominalphrase(Synonym;all;sg;all) vorkommen, dann folgt daraus:

Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:4); und es folgt daraus:
Signalwort(Satz:1,Wort:5); Beteiligter(Satz:1,Wort:4);
152. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(Signalverb;3;sg;past), eine Zeitphrase, eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Entität;dat;sg;male), eine Zeitphrase und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:1); und es folgt daraus:
Signalwort(Satz:1,Wort:8); Zeit-Signalwort(Satz:1,Wort:7); Beteiligter(Satz:1,Wort:4);
153. Regel–>Wenn in einem Satz: eine Nominalphrase(Pronomen;all;sg;all), eine Zeitphrase, eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Pronomen;all;pl;all), eine Verbalphrase(Signalverb;3;pl;past), eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(all;all;all;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Zeit-Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:6); und es folgt daraus:
Signalwort(Satz:1,Wort:8); Zeit-Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:6);
154. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;nom;sg;male), eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(Phrasenverb;3;sg;past), eine Zeitphrase, eine Signalphrase, eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(Phrasenverb;3;sg;past), eine Zeitphrase und eine Verbalphrase(Signalverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Zeit-Signalwort(Satz:1,Wort:4); Beteiligter(Satz:1,Wort:2); und es folgt daraus:
Signalwort(Satz:1,Wort:9); Zeit-Signalwort(Satz:1,Wort:8); Beteiligter(Satz:1,Wort:6);
155. Regel–>Wenn in einem Satz: eine Nominalphrase(Synonym;nom;pl;all), eine Nominalphrase(Pronomen;nom;pl;all), eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(werden;3;sg;past), eine Zeitphrase, eine Signalphrase, eine Verbalphrase(Phrasenverb;all;all;particip
eine Nominalphrase(Entität;nom;sg;all), eine Verbalphrase(Phrasenverb;3;sg;past), eine Zeitphrase und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:6); Zeit-Signalwort(Satz:1,Wort:5); Beteiligter(Satz:1,Wort:3); und es folgt daraus:
Signalwort(Satz:1,Wort:11); Zeit-Signalwort(Satz:1,Wort:10); Beteiligter(Satz:1,Wort:8);
156. Regel–>Wenn in einem Satz: eine Signalphrase, eine Zeitphrase, eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Synonym;nom;sg;male), eine No-

minalphrase(Entität;nom;sg;male), eine Nominalphrase(Synonym;nom;sg;male), eine Nominalphrase(Entität;nom;sg;male), eine Nominalphrase(Synonym;nom;sg;male), eine Nominalphrase(Entität;nom;sg;male), eine Zeitphrase, eine Signalphrase, eine Nominalphrase(Entität;nom;sg;male), eine Verbalphrase(Signalverb;3;sg;participleII) und eine Verbalphrase(sein;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Zeit-Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:5); Beteiligter(Satz:1,Wort:7); Beteiligter(Satz:1,Wort:9); und es folgt daraus:
Signalwort(Satz:1,Wort:13); Zeit-Signalwort(Satz:1,Wort:10); Beteiligter(Satz:1,Wort:12);
157. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Entität;all;sg;all), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Hilfsverb;3;sg;present) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:2);
158. Regel–>Wenn in einem Satz: eine Nominalphrase(Entität;all;sg;all), eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all) und eine Nominalphrase(Entität;all;sg;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Beteiligter(Satz:1,Wort:1);
159. Regel–>Wenn in einem Satz: eine Verbalphrase(Signalverb;3;sg;past), eine Nominalphrase(Synonym;all;sg;all) und eine Nominalphrase(Entität;all;sg;all) vorkommen, in dem Satz davor: eine Nominalphrase(Synonym;all;sg;all), eine Verbalphrase(sein;3;sg;past) und eine Zeitphrase vorkommen, in dem Satz davor: eine Nominalphrase(Entität;all;sg;all) und eine Signalphrase vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:1); Zeit-Signalwort(Satz:2,Wort:3); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:3,Wort:1); wenn folgende Nominalphrasen zusammenpassen: (Satz:3,Wort:1) zu (Satz:1,Wort:2); (Satz:3,Wort:1) zu (Satz:2,Wort:1)
160. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Signalphrase, eine Nominalphrase(Entität;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Verbalphrase(all;all;all;all) und eine Verbalphrase(all;all;all;all) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);
161. Regel–>Wenn in einem Satz: eine Zeitphrase, eine Signalphrase, eine Nominalphrase(Entität;all;sg;all) und eine Verbalphrase(Phrasenverb;3;sg;past) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:2); Zeit-Signalwort(Satz:1,Wort:1); Beteiligter(Satz:1,Wort:3);
162. Regel–>Wenn in einem Satz: eine Verbalphrase(Phrasenverb;3;sg;past), eine Nominalphrase(Pronomen;all;sg;all), eine Nominalphrase(Entität;all;sg;all), eine Nominalphra-

se(Entität;all;sg;all), eine Signalphrase und eine Verbalphrase(Phrasenverb;3;sg;present) vorkommen, dann folgt daraus:
Signalwort(Satz:1,Wort:5); Beteiligter(Satz:1,Wort:3); Beteiligter(Satz:1,Wort:4);

Literaturverzeichnis

[Appelt und Israel 1999] Appelt, Douglas E. ; Israel, David J.: *Introduction to information extraction*. 1999. – URL http://iospress.metapress.com/content/HQDWYFKJPM6B1G3A

[Bagga und Biermann 2000] Bagga, Amit ; Biermann, Alan W.: A Methodology for Cross-Document Coreference. In: *In Proceedings of the Fifth Joint Conference on Information Sciences (JCIS 2000*, 2000, S. 207–210

[Baldwin 1997] Baldwin, Breck: CogNIAC: high precision coreference with limited knowledge and linguistic resources. In: *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*. Stroudsburg, PA, USA : Association for Computational Linguistics, 1997 (ANARESOLUTION '97), S. 38–45. – URL http://dl.acm.org/citation.cfm?id=1598819.1598825

[Cowie und Wilks 1996] Cowie, Jim ; Wilks, Yorick: *Information Extraction*. 1996

[Feldman und Sanger 2006] Feldman, Ronen ; Sanger, James: *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge, MA, USA : Cambridge University Press, December 2006. – ISBN 0521836573

[Foundation 2011] Foundation, The Apache S.: *UIMA Tutorial and Developers' Guides*. 2011. – URL http://uima.apache.org/d/uimaj-2.4.0/tutorials_and_users_guides.pdf. – [Online; Stand 28. August 2012]

[Grishman 1997] Grishman, Ralph: Information extraction: Techniques and challenges. In: Pazienza, Maria (Hrsg.): *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology* Bd. 1299, Springer Berlin / Heidelberg, 1997, S. 10–27. – URL http://dx.doi.org/10.1007/3-540-63438-X_2. – ISBN 978-3-540-63438-6

[Liu u. a. 2010] Liu, Bin ; Chiticariu, Laura ; Chu, Vivian ; Jagadish, H. V. ; Reiss, Frederick R.: Automatic rule refinement for information extraction. In:

Proc. VLDB Endow. 3 (2010), September, Nr. 1-2, S. 588–597. – URL http://dl.acm.org/citation.cfm?id=1920841.1920916. – ISSN 2150-8097

[Paşca u. a. 2006] Paşca, Marius ; Lin, Dekang ; Bigham, Jeffrey ; Lifchits, Andrei ; Jain, Alpa: Names and similarities on the web: fact extraction in the fast lane. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA : Association for Computational Linguistics, 2006 (ACL-44), S. 809–816. – URL http://dx.doi.org/10.3115/1220175.1220277

[Sarawagi 2008] Sarawagi, Sunita: Information Extraction. In: *Found. Trends databases* 1 (2008), März, Nr. 3, S. 261–377. – URL http://dx.doi.org/10.1561/1900000003. – ISSN 1931-7883

[Ullenboom 2010] Ullenboom, Christian: *Java ist auch eine Insel: Das umfassende Handbuch (Galileo Computing)*. 9. Galileo Computing, 2010. – ISBN 3836215063

[wikipedia 2011] wikipedia: *Datumsformat*. 2011. – URL http://de.wikipedia.org/wiki/Datumsformat. – [Online; Stand 28. August 2012]

Glossar

IE Informationsextraktion, 1, 2

IE Informationsextraktion, 55

IE Informationsextraktion, 60

IE Informationsextraktion, 68

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 01. Januar 1970 Bastian Probst