

Heft 200

Oktober 2015



Institut für  
Wirtschaftsinformatik



**Automatisierte Identifikation und Analyse von Argumentationsstrukturen im Digital-Humanities-Projekt ARGUMENTUM –  
Ergänzende Materialien und Ergebnisse**

Constantin Houy, Tim Niesen, Philip Hake, Peter Fettke, Peter Loos



Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi)  
im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) GmbH

Herausgeber: Prof. Dr. Peter Loos

C. HOUY, T. NIESEN, P. HAKE, P. FETTKE, P. LOOS

# Automatisierte Identifikation und Analyse von Argumentationsstrukturen im Digital-Humanities-Projekt ARGUMENTUM – Ergänzende Materialien und Ergebnisse<sup>1</sup>

Veröffentlichungen des Instituts für Wirtschaftsinformatik

Herausgeber: Prof. Dr. Peter Loos

IWi-Heft Nr. 200

ISSN 1438-5678

Institut für Wirtschaftsinformatik (IWi)

im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI)  
Universität des Saarlandes, Campus, Geb. D32, D-66123 Saarbrücken

Telefon: +49 681 85775-3106, Fax: +49 681 85775-3696

E-Mail: [iwi@iwi.uni-sb.de](mailto:iwi@iwi.uni-sb.de), URL: <http://www.iwi.uni-sb.de/>

Oktober 2015

---

<sup>1</sup> Die vorliegenden Forschungsergebnisse entstammen dem Projekt „*Analyse und Synthese von Argumentationsstrukturen durch rechnergestützte Methoden am Beispiel der Rechtswissenschaft (ARGUMENTUM)*“, das durch das Bundesministerium für Bildung und Forschung (BMBF) unter dem Förderkennzeichen 01UG1237C gefördert wurde. Verschiedene Projektergebnisse wurden bereits in mehreren Publikationen dargelegt. Der vorliegende Arbeitsbericht liefert weitere und bisher unveröffentlichte Materialien, Ergebnisse und Überlegungen aus dem Projekt ARGUMENTUM, insbesondere zur Analyse des State-of-the-Art in den adressierten Themengebieten sowie zu den Tests und der Evaluation des finalen Prototyps.

## Inhalt

Abbildungen .....	ii
Abkürzungen .....	iii
1 Einleitung.....	1
2 Literaturanalyse „Computer-Supported Argumentation“ .....	2
2.1 Vorbemerkungen .....	2
2.2 Bezugsrahmen der Literaturuntersuchung.....	2
2.3 Forschungsmethode .....	5
2.4 Begriffliche Grundlagen.....	5
2.5 Literaturübersicht zu Computer-Supported Argumentation.....	8
2.6 Relevante Ansätze für Computer-Supported Argumentation.....	31
2.7 Fazit zur Literaturanalyse .....	42
3 Übersicht zu Konzepten und Prototyp.....	43
3.1 Fachkonzept.....	43
3.2 Architektur des Softwareprototyps.....	43
3.3 Implementierung und Nutzung des Prototyps .....	44
4 Test und Evaluation .....	46
4.1 Einleitung und Zielsetzung.....	46
4.2 Verwendete Methodik .....	47
4.3 Funktionale Testung .....	48
4.4 Fachliche Evaluation – Fokus: zweiter Prototyp.....	51
4.5 Fazit zu Test und Evaluation .....	59
5 Resümee.....	60
6 Literatur .....	61

## Abbildungen

Abb. 1: Vorgehensmodell zur Identifikation von Argumentationsstrukturen	3
Abb. 2: Bezugsrahmen der Untersuchung	4
Abb. 3: Argumentationsschema nach Toulmin	7
Abb. 4: Ergebnisübersicht der Ansätze in den untersuchten Arbeiten	30
Abb. 5: Übersicht Klassifikation von Textabschnitten	31
Abb. 6: Übersicht Erkennen von Grenzen in argumentativen Abschnitten	33
Abb. 7: Übersicht Klassifikation identifizierter Aussagen	34
Abb. 8: Übersicht Erkennen von Argumentationsstrukturen	35
Abb. 9: Übersicht Anreichern von Metainformationen	37
Abb. 10: Übersicht Information Retrieval	38
Abb. 11: Übersicht Inhaltliche Analyse	39
Abb. 12: Übersicht Vorschlagen von Argumentation	40
Abb. 13: Übersicht Strukturelle Abbildung von Argumentation	40
Abb. 14: Phasenkonzept	43
Abb. 15: Architekturentwurf des ARGUMENTUM-Software-Werkzeuges	44
Abb. 16: Bedienoberfläche des finalen Prototyps mit priorisierter Ergebnisliste	45
Abb. 17: Detaildarstellung mit annotierten und markierten Argumentationsmustern	45
Abb. 18: Sucheinstellungen I	53
Abb. 19: Ergebnisliste der Suchanfrage	54
Abb. 20: Detailansicht einer Entscheidung	55
Abb. 21: Filtern von Argumenten	56
Abb. 22: Sucheinstellungen II	57
Abb. 23: Änderung der Relevanzkriterien zur Ergebnissortierung	58

## Abkürzungen

AAS	Argument Assistance System
AC	Answer Classifier
AI	Artificial Intelligence
AIF	Argument Interchange Format
AML	Argument Markup Language
AO	Ontologien im Allgemeinen
AS	Argumentationsschema
BS	Beweisstandard
CBR	Case-based Reasoning
CDSS	Collaborative Decision Support Systems
CFG	Context-free Grammar
CQ	Critical Question
CSCL	Computer-Supported Collaborative Learning
CSCS	Computer-Supported Collaboration Scripts
DCS	Digital Conversational System
DM	Data Mining
ECHR	European Court of Human Rights
GG	Graph Grammar
GS	Graphstruktur
IE	Information Extraction
IR	Information Retrieval
ISS	Intelligent Support System
ITS	Intelligent Tutor System
LFG	Lexical Functional Grammar
LSA	Latent Semantic Analysis
MDS	Meeting Description Schema

MEM	Maximum Entropy Model
ML	Machine Learning
MNB	Multinomial Naive Bayes
NER	Named Entity Recognition
NLP	Natural Language Processing
OMT	Object Modeling Technique
PoS	Part-of-Speech
QC	Question Classifier
RADB	Relational Argument Database
RC	Rocchio Classifier
RST	Rhetorical Structure Theory
SMLA	Supervised Machine Learning Approach
SVM	Support Vector Machine
TAS	Toulmin Arguments Structures
TC	Text Categorization
TE	Textual Entailment
TF-IDF	Term Frequency, Inverse Document Frequency
TS	Toulmin Schema
VSM	Vector Space Model
WD	Widrow-Hoff-Algorithmus
WN	WordNet
WS	Walton Schema
WSD	Word Sense Disambiguation
XML	Extensible Markup Language

## 1 Einleitung

Das Argumentieren stellt eine grundlegende geistige Beschäftigung sowie eine zentrale Aufgabe der Forschung in sämtlichen Wissenschaftsdisziplinen dar.<sup>2</sup> Die Entwicklung von Argumentation und auch die Analyse bestehender Argumentationen sind insbesondere in der geisteswissenschaftlichen Forschung bedeutende Aufgaben des Forschers, so auch in der Rechtswissenschaft. Die Analyse von Argumentationsstrukturen ist ein anspruchsvoller intellektueller Vorgang, der stets an die natürliche Begrenzung der menschlichen Informationsverarbeitungskapazität geknüpft ist. Im Kontext der Rechtswissenschaft bieten Rechnersysteme vor dem Hintergrund der stetig wachsenden Verfügbarkeit elektronischer Rechtsprechungscorpora erhebliche Potentiale, um Argumentationsstrukturen auf Basis größerer Datenmengen zu identifizieren, zu begründen und für neue Argumentationen aufzubereiten.<sup>3</sup>

Vor diesem Hintergrund zielte das hier beschriebene Projekt ARGUMENTUM auf die Erforschung der Potentiale und Grenzen rechnergestützter Verfahren für die Analyse und das Retrieval von Argumentationsstrukturen ab und konkretisiert dies am Beispiel der Rechtswissenschaft. Der vorliegende Arbeitsbericht zielt insbesondere darauf ab, zusätzlich zu existierenden Publikationen ergänzende und bisher nicht publizierte Materialien und Überlegungen zu präsentieren. Im folgenden Kapitel zwei werden bisher nicht publizierte Ergebnisse der Literaturanalyse zur Erhebung des Forschungsstandes nach Abschluss des ersten Projektarbeitspaketes (Stand: Ende 2012) präsentiert. Kapitel drei gibt eine knappe Übersicht über das entwickelte Konzept, die Software-Architektur und die Implementierung des Prototyps, die bereits in mehreren Publikationen dargelegt wurden, bevor dann in Kapitel vier ausführlicher über Testergebnisse sowie erste Schritte der Evaluation des ARGUMENTUM-Prototyps berichtet wird. Kapitel fünf fasst die Ergebnisse zusammen.

---

<sup>2</sup> Vgl. hierzu und im Folgenden HOUY ET AL. (2015), S. 15f.

<sup>3</sup> Vgl. u. a. MOCHALES-PALAU ET AL. (2009), MOCHALES ET AL. (2011), ROONEY ET AL. (2012)

## 2 Literaturanalyse „Computer-Supported Argumentation“

### 2.1 Vorbemerkungen

Im Folgenden wird der Stand der Forschung zu relevanten Aspekten der Analyse, des Retrievals und des Vorschlagens von Argumentationsstrukturen aus der Sicht der Wirtschaftsinformatik erhoben (Stand: Ende 2012). In diesem Zusammenhang werden in einem ersten Schritt Literatur sowie erste Erfahrungsberichte zu den Themen *Analyse*, *Retrieval* und *Vorschlagen* von Argumentationsstrukturen aus Sicht der Wirtschaftsinformatik identifiziert, systematisiert und ausgewertet. Gegenstand des im Folgenden gegebenen Überblicks sind internationale Arbeiten, Projekte und auch Werkzeuge im adressierten Themengebiet. Bei der Auswertung der Literatur wurden insbesondere folgende Aspekte und Fragen berücksichtigt:

- Welche Anforderungen werden an Ansätze zur Argumentationsanalyse und zum Retrieval von Argumentationsstrukturen in der Literatur diskutiert?
- Welche Ansätze zur Umsetzung existieren in diesem Kontext?
- Welche Erfahrungen liegen bereits vor?

### 2.2 Bezugsrahmen der Literaturuntersuchung

Um diese Aspekte strukturiert und systematisch zu beleuchten, wird im ersten Schritt ein Bezugsrahmen eingeführt, der das zu untersuchende Themengebiet anhand der gesetzten Projektziele abgegrenzt. Im Projekt ARGUMENTUM sollten Ansätze und Methoden zur rechnergestützten Argumentation, insbesondere für das Retrieval sowie die Analyse und das Vorschlagen von Argumentationsstrukturen entwickelt und untersucht werden. Dabei ergeben sich verschiedene Probleme und Anforderungen, die mithilfe des folgenden Bezugsrahmens strukturiert werden sollen. Zunächst gilt es Texte, in denen Argumentationsstrukturen formuliert werden, derart aufzubereiten, so dass darauf folgend ein Information Retrieval, weiterführende Analysen bzw. das Vorschlagen nützlicher Argumentationen möglich werden können. In diesem Zusammenhang können Ansätze des Argumentation Mining, die seit wenigen Jahren entwickelt und untersucht

werden, zum Einsatz kommen, um zuerst Argumentationsketten zu strukturieren und dann mit für ein Retrieval notwendigen Metainformationen anzureichern. Auf dieser Basis lassen sich dann die weiteren Operationen ausführen. Auf der Grundlage zentraler Vorarbeiten zum Argumentation Mining<sup>4</sup> wurde folgendes Vorgehensmodell entwickelt (Abb. 1), das auch die Präsentation der Ergebnisse der Literaturanalyse strukturiert.

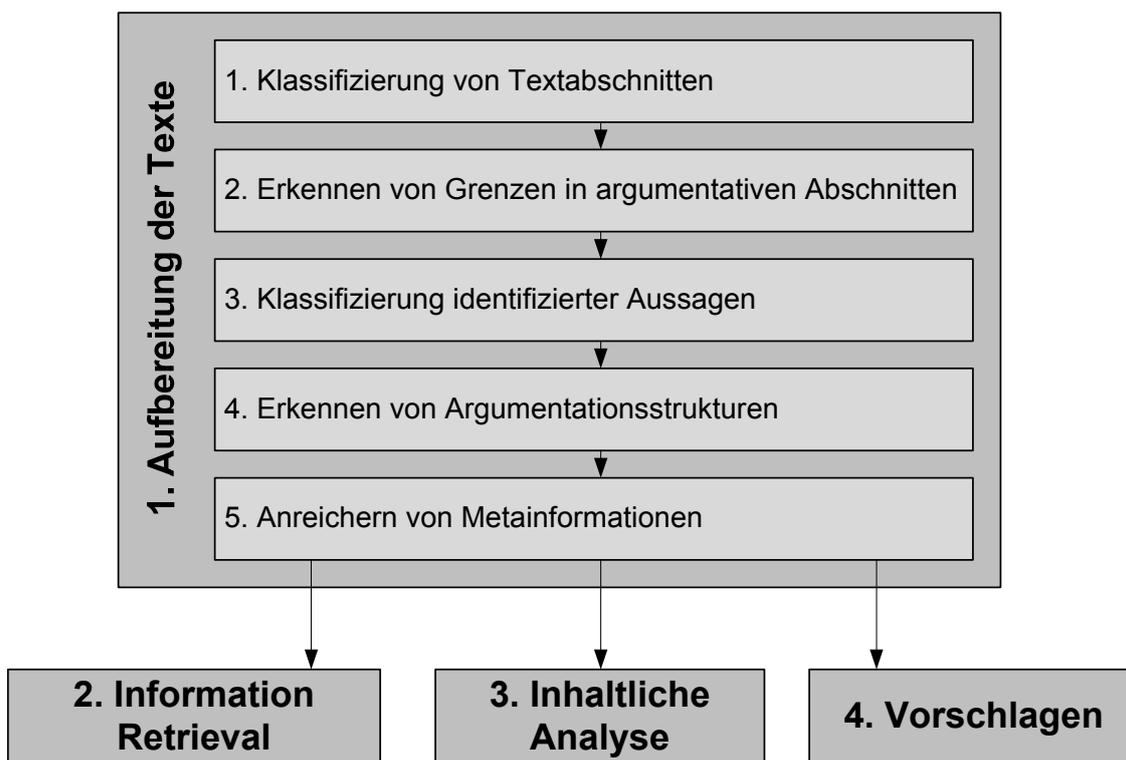


Abb. 1: Vorgehensmodell zur Identifikation von Argumentationsstrukturen<sup>5</sup>

Es ergeben sich innerhalb dieses Modells vier wichtige Funktionsbereiche zur Erreichung der Zielsetzung des Projektes ARGUMENTUM (*Aufbereitung der Texte, Information Retrieval, Inhaltliche Analyse von Argumenten und Vorschlagen von Argumentation*), die im Folgenden näher erläutert werden.

<sup>4</sup> Vgl. u. a. MOCHALES-PALAU ET AL. (2009), MOCHALES ET AL. (2011), ROONEY ET AL. (2012)

<sup>5</sup> Die ersten vier Prozessschritte orientieren sich an dem von Mochales und Moens vorgestellten Artikel: MOCHALES ET AL. (2011)

**Funktionsbereich 1: Aufbereitung der Texte**

Hauptziel dieses Funktionsbereiches ist die grundlegende Entdeckung und Gliederung von Argumentationsstrukturen in vorliegenden Texten. Es gilt dabei, vor allem Grenzen zwischen einzelnen Teilabschnitten einer Argumentation ihrer Funktion nach zu identifizieren und diese mit Metainformationen anzureichern, sodass alle weiteren Operationen, wie Information Retrieval, inhaltliche Analyse oder Vorschlagen von Argumentation möglich werden.

**Funktionsbereich 2: Information Retrieval (Hauptfunktionalität)**

Dieser Funktionsbereich beschäftigt sich mit dem Auffinden relevanter Informationen in einer Argumentationsstruktur. Für die Rechtswissenschaft, die das zentrale Untersuchungs- und Anwendungsgebiet des Projektes ARGUMENTUM darstellt, ist es von zentraler Bedeutung, diejenigen Begründungen oder auch Autoritäten zu identifizieren, die eine bestimmte These unterstützen bzw. auf die im Rahmen einer eigens zu erstellenden Argumentation zurückgegriffen werden kann. Elementare Begründungsstrukturen können in komplexe Strukturen eingebettet sein. Es gilt in diesem Zusammenhang, textuelle Argumentationsstrukturen derart aufzubereiten, dass diese Informationselemente automatisiert identifiziert werden können.

**Funktionsbereich 3: Inhaltliche Analyse**

In diesem Funktionsbereich sollen die identifizierten Argumentationsstrukturen weitergehend inhaltlich, d. h. thematisch, analysiert und abgelegt werden können. Es ist zu untersuchen, inwiefern und in welcher Form dies realisierbar ist.

**Funktionsbereich 4: Vorschlagen von Argumentationsstrukturen**

Gegenstand dieses Funktionsbereiches soll eine automatisierte Generierung von Vorschlägen für passende Argumente sein. Auch hier könnten auf Basis der Aufbereitung von Argumentation und unter Berücksichtigung weiterer Randbedingungen zu einer Situation thematisch passende Argumentationen identifiziert und vorgeschlagen werden. Es ist zu untersuchen, inwiefern und in welcher Form dies möglich und realisierbar ist.

*Abb. 2: Bezugsrahmen der Untersuchung*

Im folgenden Abschnitt wird die angewendete Forschungsmethode expliziert.

## 2.3 Forschungsmethode

Im Rahmen der folgenden Betrachtung wurde ein strukturiertes Review<sup>6</sup> zu verschiedenen Themen im Kontext des computergestützten Retrievals, der Analyse und des Vorschlagens von Argumentationsstrukturen in Texten durchgeführt (Stand: Ende 2012). Allgemeine Arbeiten zur computergestützten Argumentation sowie zentrale Begriffe der einzelnen Funktionsbereiche des Bezugsrahmens wurden systematisch recherchiert. Dazu wurden insbesondere folgende zentralen Suchterme im Titel, Abstract und den Schlüsselwörtern von wissenschaftlichen Artikeln gesucht (*title, abstract, keywords*): „computer-supported argumentation“, „argumentation mining“, „argument mining“, „argumentation detection“, „argument detection“, „argumentation analysis“, „argumentation synthesis“. Verwendet wurde die Wissenschaftsdatenbank *Scopus*. Ausgehend von der Suche mit den genannten Suchbegriffen (*Vorwärtssuche*) wurden in den so identifizierten Dokumenten weitere relevante Suchwörter entdeckt und untersucht sowie relevante dort zitierte Arbeiten berücksichtigt, um so einen umfassenden Überblick über den Stand der Forschung im adressierten Forschungsgebiet zu erhalten (*Rückwärtssuche*). Alle auf diese Weise identifizierten 33 Arbeiten wurden analysiert, ihre zentralen Ziele, die verwendeten Forschungsansätze sowie die Kernaussagen dokumentiert. Im folgenden Abschnitt werden zunächst wichtige Grundbegriffe zum Thema „Argumentationstheorie“ eingeführt, bevor die Kerninhalte der untersuchten Artikel präsentiert werden.

## 2.4 Begriffliche Grundlagen

### 2.4.1 Ansatz nach Toulmin

Die Grundlagen seiner Argumentationstheorie beschreibt TOULMIN in seinem 1975 erschienenen Hauptwerk „The Uses of Argument“.<sup>7</sup> Nach Toulmin dienen Argumente zur Begründung und Unterstützung einer Behauptung. Das eingeführte Argumentationschema setzt die Schlussfolgerung einer argumentativen Aussage in Bezug zu vorhan-

---

<sup>6</sup> Vgl. FETTKE (2006)

<sup>7</sup> Vgl. TOULMIN (1975)

denen Daten. Allgemein besteht ein Argumentationsschema nach Toulmin aus folgenden Elementen:

- Argument (engl. *datum*): eine vorliegende Aussage (engl. *proposition*) stellt eine Tatsachenbehauptung dar, die für den weiteren Aufbau der Argumentation als wahr angesehen wird. Sie stützt sich meist auf Beobachtungen oder vorhandenes Wissen einer Person.
- Schlussregel (engl. *warrant*): Schlussregeln dienen als Rechtfertigung für die Ableitung einer Schlussfolgerung aus einem Argument. Toulmin bezeichnet Schlussregeln als „brückenartige Aussagen“ zwischen den beiden Elementen. Sie bestehen aus Regeln, die mit gewisser Wahrscheinlichkeit bei Vorliegen eines Arguments zum Eintritt der Schlussfolgerung führen. Schlussregeln sind meist nur implizit in einer Argumentation enthalten.
- Stützung (engl. *backing*): Stützungen belegen die Gültigkeit von Schlussregeln. Sie basieren beispielsweise auf allgemein anerkannten Gesetzmäßigkeiten oder normativen Regeln und stellen ein Wertesysteme dar, auf dessen Grundlage eine Argumentation stattfinden kann.
- Operator (engl. *qualifier*): Operatoren modifizieren den Geltungsbereich einer Aussage bzw. schränken die Wahrscheinlichkeit ein, mit der das Eintreten einer Schlussfolgerung bezeichnet wird. Typischerweise sind Operatoren Modalausdrücke, beispielsweise „notwendigerweise“, das eine stark allgemeingültige Aussage betont oder das einschränkende „wahrscheinlich“, das eine gewisse Unsicherheit der Schlussfolgerung einschließt.
- Schlussfolgerung (engl. *conclusion*): die Schlussfolgerung ist die aus dem Argument abgeleitete Folge, die auf Basis der vorliegenden Daten unter Anwendung der gegebenen Schlussregeln als gültig angesehen wird. Sie kann durch weitere Argumente angegriffen und infrage gestellt werden.
- Ausnahmebedingung (engl. *rebuttal*): Ausnahmebedingungen stellen explizite Fälle dar, in denen die Anwendung einer Schlussregel (auch wenn diese nicht widerlegt wurde) nicht zur Schlussfolgerung führt. Sie schränken also ebenso wie Operatoren den Geltungsbereich einer Schlussfolgerung ein.

Die Beziehungen der einzelnen Elemente werden in Abbildung 3 veranschaulicht.

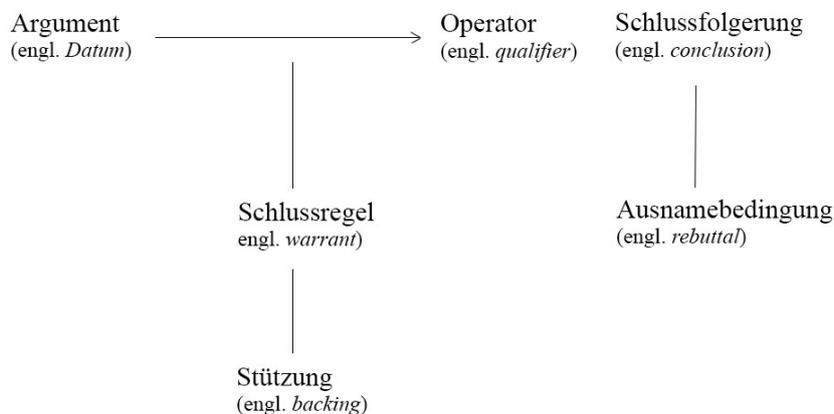


Abb. 3: Argumentationsschema nach Toulmin

### 2.4.2 Ansatz nach Walton

WALTON definiert Argumentationsschemata als die Form, in der Argumente auftreten und die folgernde Strukturen einer Argumentation abbildet. In seinem Standardwerk „Argumentation Schemes for Presumptive Reasoning“ aus dem Jahre 1996 listet WALTON unterschiedliche Schemata zusammen mit einer Menge von kritischen Fragen (*critical questions*, CQ) zu jedem Schema auf.<sup>8</sup> Die kritischen Fragen dienen dazu, ein Argument im vorliegenden Fall mit den Anforderungen an das verwendete Argumentationsschema abzugleichen.

Die Argumentationsschemata werden als *presumptive* (mutmaßend, vermutend) bezeichnet, was eine prinzipielle Anfechtbarkeit der Argumente verdeutlicht. Walton spricht in diesem Zusammenhang auch von *defeasible arguments*.

Ein Beispiel für ein solches Schema ist das *Argument from Position to Know*:

**Hauptvoraussetzung** (engl. *major premise*): Quelle a ist in der Position, über Dinge eines bestimmten Themenbereichs S, zu dem Aussage A gehört, Bescheid zu wissen.

**Nebenvoraussetzung** (engl. *minor premise*): a behauptet, dass A (im Bereich S) richtig (falsch) ist.

**Schlussfolgerung** (engl. *conclusion*): A ist richtig (falsch).

<sup>8</sup> Vgl. WALTON (1996)

Kritische Fragen zum Argumentationsschema *Argument from Position to Know*:

**CQ1:** Ist a in der Position zu wissen, ob A richtig (falsch) ist?

**CQ2:** Ist a eine vertrauenswürdige Quelle?

**CQ3:** Hat a behauptet, dass A richtig (falsch) ist?

### 2.4.3 Weitere Ansätze

Es werden in der Literatur noch weitere Ansätze für die Strukturierung von Argumentationen diskutiert, z. B. in den Arbeiten von PRAKKEN,<sup>9</sup> VAN EEMEREN und GROO-TENDORST,<sup>10</sup> auf die an dieser Stelle nur verwiesen werden soll.

## 2.5 Literaturübersicht zu Computer-Supported Argumentation

### 2.5.1 Vorbemerkung

Im folgenden Abschnitt werden die im Rahmen der Recherche identifizierten 33 Artikel unter besonderer Berücksichtigung ihrer Zielsetzung, Methode und der Kerninhalte zusammengefasst. Der adressierte Untersuchungszeitraum (bis einschließlich 2012) endet mit dem Abschluss des entsprechenden Arbeitspaketes im Projekt ARGUMENTUM (AP 1 zur Erhebung des *State-of-the-Art*). Die Arbeiten sind chronologisch nach ihrem Erscheinungsjahr sortiert und innerhalb eines Jahres alphabetisch nach Autorennamen angeordnet.

### 2.5.2 Relevante Forschungsarbeiten aus den Jahren 1982 – 2000

#### 1. ROESNER und LAUBSCH (1982): Formalization of Argumentation Structures in Newspaper Texts<sup>11</sup>

**Ziel:** Untersuchung der Anwendung von Argumentationsschemata und deren Beziehungen zu anderen Informationsquellen; besseres Verständnis von Zeitungsartikeln zum Thema Arbeitsplatzentwicklung; konzeptueller Ansatz zur Verarbeitung solcher Strukturen

**Methode:** Konzeptentwicklung, Anwendungsbeispiel

**Kerninhalte:**

- Argumentationsschemata definiert als Theorie zur Strukturierung von domänenspezifischen Abhängigkeiten (engl. *dependency*); Nutzung eines *Dependency Network* zum Abbilden der Vorkenntnisse des durchschnittlichen Zeitungslesers und der Fähigkeit zur qualitativen Schlussfolgerung.
- „Verstehen von Argumentation“ kann verstanden werden als Etablieren von Beschränkungen (engl. *constraints*) der Beziehungen zwischen Behauptungen und Schlussfolgerungsmechanismen.

---

<sup>9</sup> Vgl. PRAKKEN (2010)

<sup>10</sup> Vgl. VAN EEMEREN ET AL. (2004)

<sup>11</sup> Vgl. ROESNER ET AL. (1982)

- Erklärungsschemata, dienen der Begründung von Behauptungen; drei Arten werden unterschieden:
  - *WEGEN* <PROP-1> <PROP-2>: Die Behauptung <PROP-1> begründet <PROP-2>; im *Dependency Network* muss ein Pfad von <PROP-1> nach <PROP-2> existieren.
  - *DARUEBER-HINAUS* <PROP>: bisher konstatierte Begründungen belegen die Behauptung nicht ausreichend; *DARUEBER-HINAUS*-Schema eröffnet neuen Pfad im *Dependency Network*.
  - *TROTZ* <PROP-1> <PROP-2>: Abbilden von Erwartung und Nichterfüllung; implizite Einführung einer dritten Behauptung <PROP-2'>, die erstens eine Negation von <PROP-2> ist und zweitens als Standardkonsequenz aufgefasst werden kann.
- Regeln zur Verarbeitung der unterschiedlichen Schemata, Verfeinerung des *Dependency Network*, „*blow up*“-Prozedur zur detaillierteren Betrachtung einzelner Teile des Netzwerks durch Auswahl der unmittelbar benachbarten Knoten.

## 2. GORDON und KARACAPILIDIS (1997): The Zeno Argumentation Framework<sup>12</sup>

Ziel: Entwicklung eines mathematisch-formalen Modells zur Darstellung eines argumentativen Prozesses

Methode: Entwicklung eines Frameworks, Modellierung, gestaltungsorientierter Ansatz

- Kerninhalte:
- Das Konzept wird im Hinblick auf den Einsatz in einem sogenannten *Mediation System* entwickelt; Mediationssysteme stellen Beziehungen von Argumenten innerhalb einer Diskussion dar, lenken Diskussionen in Richtung vielversprechender Ansätze und unterstützen die Moderatoren der Diskussion.
  - Systementwurf für ein Mediation System mittels der *Object Modeling Technique* (OMT); drei Diagrammarten:
    - *Funktionales Modell*: stellen den Datenfluss im Diskussionsprozess dar; enthält ein generelles Argumentationsmodell.
    - *Objektmodell*: Objekte, Beziehungen und Vererbungshierarchie; Objekten können Modelle zugeordnet werden, um verschiedene Sichten / Interpretationen auf bspw. Nachrichten abzubilden: dialektische Graphen zur Darstellung des Standes einer Diskussion zu einem spezifischen Zeitpunkt, Fokus auf der Rolle und Funktion der Redebeiträge.
    - *Dynamisches Modell*: gibt an, welche Art von Redebeiträgen (engl. *Speech Acts*) in welchem Status der Diskussion möglich sind.
  - Mathematisch-formale Definition eines Argumentationsmodells (Diskussionsmodells); nicht geeignet zur direkten Umsetzung in Programmcode; definierte Elemente: Positionen, Terms, Argumente, Constraints, Streitfragen, dialektische Graphen.
  - Verschiedene Beweisstandards sind vorgesehen:
    - *Scintilla of Evidence*: erfüllt, wenn wenigstens eine Position das Argument stützt,
    - *Preponderance of the Evidence*: die Summe der unterstützenden Argumente wiegt die Summe der ablehnenden Argumente auf,
    - *No Better Alternative*: keine Alternative hat bessere Argumente als die gerade betrachtete,
    - *Best Choice*: gerade betrachtete Alternative ist besser als alle anderen („besser“ ist qualitativ zu bestimmen) und
    - *Beyond a Reasonable Doubt*: kein Argument spricht gegen die Alternative und kein Argumente spricht für eine andere Alternative.

## 3. RODRIGUEZ ET AL. (1997): Using WordNet to Complement Training Information in Text Categorization<sup>13</sup>

Ziel: Erweiterung bestehender Konzepte zur Kategorisierung von Texten (engl. *Text Categorization* (TC)) um eine Anbindung an lexikalische Wortdatenbanken wie WordNet

Methode: Konzeptentwicklung, Implementierung, Evaluation

- Kerninhalte:
- Die Autoren erweitern die bestehenden Ansätze zur Einteilung von Dokumenten in Kategorien um zusätzliche Informationsressourcen, um die lexikalische Datenbank WordNet.
  - Als Testcorpus verwenden die Autoren die *Reuters-21578 TC Test Collection*.<sup>14</sup>

<sup>12</sup> Vgl. GORDON ET AL. (1997)

<sup>13</sup> Vgl. RODRÍGUEZ ET AL. (1997)

<sup>14</sup> <http://www.daviddlewis.com/resources/testcollections/reuters21578/>

- Die zu kategorisierenden Terme werden mithilfe eines VSM kodiert und mit als Vektoren repräsentierten Kategorien abgeglichen.
- Die Klassifizierung von Termen erfolgt über zwei unterschiedliche Algorithmen, die im Anschluss jeweils beide mit der Erweiterung um WordNet kombiniert werden; beide Algorithmen und die jeweilige Berechnung von gewichteten Vektoren werden im Papier beschrieben:
  - *Rocchio-Algorithmus*: wird vornehmlich im IR eingesetzt, um relevantes Feedback zu einer Anfrage liefern zu können; in Bezug auf die Zuteilung von Kategorien ist zu beachten, dass der Algorithmus jeder Kategorie die gleiche Priorität zuweist.
  - *Widrow-Hoff-Algorithmus*: die Priorisierung einer Kategorie wird danach gewichtet, wie oft eine Kategorie im Trainingscorpus vorkommt (je häufiger, desto höher wird ihr Gewicht).
- Durch die Verwendung von WordNet wird der Begriff für eine Kategorie (z. B. Kategorie „Verdienst“) durch Synonyme sowie semantisch und konzeptuell verwandte Begriffe erweitert; nach festgelegten Kriterien wird zu jedem zu kategorisierenden Begriff seine semantische Nähe zur Kategorie berechnet und dementsprechend ein Anfangswert für die Gewichtung gesetzt → dieser dient als Ausgangspunkt für die Anwendung der beiden Algorithmen.
- Die Evaluation der Algorithmen in Verbindung mit WordNet führt zu besseren Ergebnissen als ohne; konkrete Leistung hängt von der Anzahl der Kategorien und vom jeweiligen Corpus ab.

#### 4. STRANIERI / ZELEZNIKOW (1999): A Survey of Argumentation Structures for Intelligent Decision Support<sup>15</sup>

Ziel: Erstellung einer Übersicht über Argumentationsstrukturen und Argumentationstheorien zur Verwendung in *Intelligent-Decision-Support-Systemen*, Erweiterung des Toulmin-Schemas zur Anwendung im juristischen Bereich des Familienrechts.

Methode: Literaturreview

Kerninhalte:

- Übersicht über Argumentationstheorien: analytische und dialektische Beweise nach Aristoteles, *Toulmin Arguments Structures* (TAS) zur Beschreibung von Argumentstrukturen unabhängig vom Inhalt und zur semantischen Erfassung von Schlussfolgerungen
- Begründung für die Verwendung von TAS für Anwendungen in der Künstlichen Intelligenz:
  - Argumentation entspricht praktischer Schlussfolgerung: im Gegensatz zu praktischer Schlussfolgerung besitzt analytische Schlussfolgerung Einschränkungen bzgl. praktischer Anwendung (z. B. Unsicherheit wird nicht erfasst, Behauptungen über Behauptungen sind nicht möglich, Logik ist monoton.<sup>16</sup>)
  - Argumentation kann mehrere unterschiedliche Typen von Schlussfolgerung (engl. *inferencing*) umfassen.
  - Argumentation ist verbunden mit Erklärungen: TAS hat Datenelemente, die Beweise für vorgebrachte Fakten enthalten.
  - Argumentation umfasst plausible Schlussfolgerungen: Art der Schlussfolgerung kann und soll abgebildet werden, da z. B. die Methode der Induktion im Vergleich zur Methode der Deduktion als „schwächer“ gilt.
  - Argumente können so im Rahmen von Schlussfolgerungsketten kombiniert werden.
- *Split-Up*-Projekt: Verwendung von TAS zur Strukturierung von juristischem Wissen zur Verwendung von IR-Mechanismen
- Es existieren verschiedene Variationen von TAS:
  - *Johnson's TAS*: führt 5 Typen für Begründungen ein (z. B. axiomatische Begründung, medizinische Diagnose) → Nutzung zur genaueren Bestimmung der Folgerungsmethode.
  - *Freeman variation*: Erweiterung des TAS-Konstrukts zur Abbildung des Artefakts „Beweislast“ im juristischen Bereich
  - *Bench-Capon's variation*: Einführung des Konstrukts Vorannahme (engl. *presupposition*) in der TAS
  - Erweiterung durch STRANIERI UND ZELEZNIKOW: Unterscheidung zweier Typen von Schlussregeln (engl. *warrant*) zur Differenzierung ihrer Funktion: 1.) Betonen der Relevanz eines Faktums und 2.) Beeinflussung der Folgerung. Dadurch entsteht die Möglichkeit, explizite Gründe für die Relevanz eines Datenitems anzugeben; Explikation der Folgerungsmethode (Aussagen über Qualität einer Folgerung möglich); explizite Angabe von Gründen für eine Folgerungsmethode möglich → insgesamt Erleichterung der Entwicklung halb-automatischer Folgerungssysteme.

<sup>15</sup> Vgl. STRANIERI ET AL. (1999)

<sup>16</sup> [http://en.wikipedia.org/wiki/Non-monotonic\\_logic](http://en.wikipedia.org/wiki/Non-monotonic_logic)

## 2.5.3 Relevante Forschungsarbeiten aus den Jahren 2001 – 2012

### 1. KARACAPILIDIS und PAPADIAS (2001): Computer Supported Argumentation and Collaborative Decision Making - The Hermes System<sup>17</sup>

- Ziel: Entwicklung eines Systems zur Unterstützung von Gruppendiskussionen, Strukturierung von Diskussionen und Berechnung von Konsistenz und Wichtigkeit argumentativer Strukturen
- Methode: Entwicklung, Implementierung und Evaluation eines Systems für Computer-supported Argumentation (eher „Diskursmodellierung“)
- Kerninhalte:
- Entwicklung eines *Collaborative-Decision-Support-Systems* (CDSS) namens HERMES zur Lösung von Entscheidungsproblemen in Teams. Das System soll menschliche Entscheider unterstützen.
  - Anforderungen an HERMES:
    - Unterstützung bei der gemeinsamen Formulierung eines Problems → entspricht der Vorarbeit, um formale Modellierungs- / Analysetools anwenden zu können.
    - Aktives Unterstützen bei der Entscheidungsfindung über Strukturierung der Fakten und das Anbieten möglicher Schlussfolgerungen (*reasoning mechanism*)
    - Identifikation und Vermeiden von Inkonsistenzen in modellierter Argumentation
  - Konstrukte für die Modellierung mit dem HERMES-System (eher diskursorientiert):
    - *Issue*: zu diskutierendes Thema / zu erreichendes Ziel,
    - *Alternative*: in einem Issue enthalten, Alternativen zur Wahl,
    - *Position*: Unterstützung / Ablehnung einer Alternative und
    - *Constraint*: Gewichtung eines Issues.
  - Das diskutierte Thema wird durch die Modellierungskonstrukte in einer Baumstruktur dargestellt; jeder Knoten hat einen Aktivierungsstatus (aktiv / inaktiv). Der Aktivierungsstatus berechnet sich aus spezifizierten Beweisstandards (engl. *proof standards*) und wirkt sich rekursiv auf Elemente weiter oben im Baum aus.
  - In HERMES integrierte Beweisstandards:
    - *Scintilla of Evidence*: eine Position ist aktiv, wenn sie von mindestens einer aktiven Position gestützt wird,
    - *Beyond Reasonable Doubt*: eine Position ist aktiv, wenn keine aktive Position gegen sie spricht,
    - *Preponderance of Evidence*: jedes Konstrukt hat ein Gewicht (engl. *score*), eine Position ist aktiv, wenn die Positionen, die sie unterstützen, ein größeres Gesamtgewicht haben als diejenigen, die gegen sie sprechen.
  - Jeder *Constraint* hat einen Konsistenzstatus (konsistent / inkonsistent); bei jedem Einfügevorgang eines Constraints wird überprüft, ob er in Konflikt zu einem bestehenden Constraint steht (falls ja, wird automatisch ein neues Issue zur Diskussion eröffnet).
  - Constraints definieren Relationen im Sinne von „gleich wichtig wie“ oder „wichtiger als“ bzgl. verschiedener Positionen; eine topologische Sortierung der Graphdarstellung solcher Relationen liefert Gewichte, die zur Bestimmung der Relevanz einer Position genutzt werden können. Diese Gewichte werden zur Festlegung des Aktivierungsstatus genutzt.
  - Funktionen zur Konstruktion und Bewahrung der Modellstruktur in Pseudocode (z. B. Update der Aktivierungsstatus), getrennt nach Nutzeraufrufen und systeminternen Aufrufen
  - Information-Retrieval-Tool ermöglicht die Abfrage externer Datenbanken, um Anfragen betreffend der laufenden Diskussion zu stellen.
  - Geplant wurden die Entwicklung eines Argument-Builder-Tools und eine regel-basierte Unterstützung des Nutzers zur Lösung von Problemen wie unzureichender Quellenbelege eines Arguments etc.
  - Es wurde eine Evaluation unter Laborbedingungen und in zwei realen Projekten durchgeführt.

### 2. VERHEIJ (2001): ARTIFICIAL ARGUMENT ASSISTANTS FOR DEFEASIBLE ARGUMENTATION<sup>18</sup>

- Ziel: Diskussion zweier *Argument-Assistance-Systeme* (AAS) auf Grundlage anfechtbarer Argumentation (engl. *defeasible argumentation*) im juristischen Bereich, d. h. auf Aussagen / Argumenten, die angegriffen und wiederlegt werden können
- Methode: Vorstellung der Systeme, Anwendbarkeit wird anhand eines Fallbeispiels demonstriert.

<sup>17</sup> Vgl. KARACAPILIDIS ET AL. (2001)

<sup>18</sup> Vgl. VERHEIJ (2003)

- Kerninhalte:**
- AAS können den Benutzer bei der Erzeugung von Argumentationen unterstützen, hier keine automatische Schlussfolgerung.
  - AAS können helfen, den Überblick über argumentative Prozesse zu behalten, die Einhaltung von Schlussfolgerungsregeln zu überprüfen, Streitfragen und Annahmen zu überblicken. Sie sind eher „passiver“ Natur, aber „einfach“ zu implementieren, da viele Aufgaben weiterhin dem Nutzer überlassen werden sollen.
  - *Defeasible argumentation* bezieht sich auf den dialektischen Charakter von juristischen Fragestellungen. Der Grundbegriff *defeater* gibt an, welche Menge von Argumenten welche andere Menge schlägt: *undercutting expectations / defeaters* bezeichnen Ausnahmen, welche die Verbindung zwischen einem Argument und seiner Schlussfolgerung blockieren.
  - **ARGUE!-System:** basierend auf dem CUMULA-System<sup>19</sup>
  - Grundannahmen: 1.) Argumentation ist ein Prozess in dem Argumente entwickelt und Gegenargumente angeführt werden, 2.) Argumente sind anfechtbar, d. h. ihre Schlussfolgerung hängt von den verfügbaren Gegenargumenten ab.
  - Argumente besitzen einen Status (engl. *defeat status*), der von drei Faktoren abhängt:
    - Argumentstruktur: Unterordnung / Nebenordnung hat Einfluss auf Status, z. B. sind Argumente mit vielen unterstützenden Beweisen „schwerer“ zu schlagen.
    - Attacken von Gegenargumenten in Abgrenzung zu inkonsistenten Schlussfolgerungen
    - Verlauf des Argumentationsprozesses
  - Beschränkungen des verwendeten CUMULA-Modells: keine formelle Struktur der zugrundeliegenden Modellierungssprache (keine Quantoren etc.); nicht alle relevanten argumentativen Informationen können abgebildet werden (z. B. keine Begründungen).
  - Modellierung des Fallbeispiels in ARGUE! und Beschreibung der Konstrukte und Logik (vier Programmmodi)
  - **ARGUMED-System:** basierend auf DEFLOG<sup>20</sup>
  - Grundannahme: Argumente sind grundsätzlich dialektisch, d. h. Unterstützung und Attacke sind nebengeordnet statt unter- / übergeordnet → Zwei Arten von Verbindungen zwischen Aussagen: *support* und *attack*; Verbindungen werden selbst wieder als eine Art von Aussage betrachtet.
  - Evaluation basierend auf *prima facie* (lat. von vornherein) als wahr betrachteten Aussagen; einfache logische Sprache als Grundlage (formale Definition im Artikel)
  - Modellierung des Fallbeispiels mit ARGUMED; Abweichung von Modellierung mit ARGUE!; Möglichkeit, mehr Informationen darzustellen (z. B. Präzedenzfälle).

### 3. CHKLOVSKI ET AL. (2005): User Interfaces with Semi-Formal Representations: a Study of Designing Argumentation Structures<sup>21</sup>

**Ziel:** Entwicklung einer semiformalen Repräsentation von Argumentationsstrukturen, um Freitext für automatisierte Verfahren leichter zugänglich zu machen.

**Methode:** Konzeptentwicklung und Evaluation

- Kerninhalte:**
- „Mittelweg“ bezüglich des User-Interfaces zwischen Maschinenlesbarkeit und ansprechender Präsentation der Inhalte für den Nutzer ist notwendig.
  - „Semiformale“ Darstellung ermöglicht Freitextaussagen mit maschinenlesbarer Struktur (Bereich „Computer-Assisted Argumentation“), semiformale Darstellung kann auf Basis von NLP- und ML-Ansätzen automatisiert vom System weiterstrukturiert und weiterverarbeitet werden.
  - Verweis auf frühere Studie<sup>22</sup> im Bereich *canonicalization* zum Abgleich (engl. *Mapping*) von Argumenten zwischen verschiedenen Texten
  - Entwicklung von drei Konzepten zur semiformalen Repräsentation:
    1. *Rich Trellis*: Verknüpfung von Freitexten durch argumentative Konnektoren (z. B. „wird gestützt durch“); Clustering-Algorithmen und NLP-Techniken (z. B. PoS-Tagging) können angewendet werden. Abgleich von Argumenten erscheint schwierig: Vielzahl an Konnektoren führt leicht zu Inkonsistenzen, gleiches Argument kann unterschiedlich strukturiert werden, Kontrollfluss kann durch AND- / OR-Konnektoren gestört werden.
    2. *Tree Trellis*: im Vergleich zu *Rich Trellis* vereinfachte Struktur, es existieren lediglich zwei Konnektoren (*pro* und *con*).

<sup>19</sup> Vgl. VERHEIJ (1996)

<sup>20</sup> Vgl. VERHEIJ (1999)

<sup>21</sup> Vgl. CHKLOVSKI ET AL. (2005)

<sup>22</sup> Vgl. CHKLOVSKI (2003)

3. *Table Trellis*: Tabellen- / Matrix-Darstellung; klar unterscheidbare Merkmale als Spaltennamen, zu vergleichende Alternativen als Zeilen, Zelleneinträge entsprechen Werten

- Evaluation der drei Ansätze: Mapping zwischen Argumenten erwies sich nur mit der *Table-Trellis*-Darstellung aufgrund der starken vorgegebenen Struktur einigermaßen robust möglich.

#### 4. FENG ET AL. (2006): Towards Modeling Threaded Discussions using Induced Ontology Knowledge<sup>23</sup>

Ziel: (1) Untersuchung des Zusammenhangs zwischen dem Thema eines Diskussionsthreads und den enthaltenen Nachrichten in Online-Foren, (2) Vorstellen eines Ansatzes zur automatischen Ableitung von Ontologien aus einem geordneten Text

Methode: Empirische Studie, Konzeptentwicklung

Kerninhalte:

- Ziel ist es, Themenabweichungen und Verschiebungen des Fokus einer Diskussion zu erkennen.
- Diskussionsthreads sind geordnete Mengen von Nachrichten → Analyse des Inhalts einer Nachricht, Beziehungen der Nachrichten untereinander und Beziehungen der Nachrichten zum übergeordneten Thread
- *Identifikation von Themen und Erkennen von Fokusverschiebungen*
  - Verwendung eines *Rocchio classifiers*: dieser berechnet einen Vektor pro Textkategorie (in diesem Fall pro Diskussionsthema) als gewichteter Durchschnitt von positiven und negativen Trainingsbeispielen. Nachrichten in Threads müssen hierzu als positiv / negativ bewertet werden. Falls eine solche Bewertung nicht verfügbar ist, werden die ersten  $n$  Klassifikationen als positive Beispiele angesehen.
  - Ähnlichkeit zwischen einem Diskussionsthema und einer Nachricht wird als Kosinus-Ähnlichkeit zwischen dem Vektor des Threads und dem Vektor der Nachricht bestimmt.
  - Jeder Vektor einer Nachricht enthält Elemente, die auf einer TF-IDF (*Term Frequency, Inverse Document Frequency*) Transformation basieren,<sup>24</sup> durch Berechnung der lexikalischen Ähnlichkeit zwischen Nachrichtenvektor und Themenvektor kann die Relevanz einer Nachricht für das gegebene Thema bestimmt werden.
- *Ableitung einer Ontologie*:
  - Automatisiertes Ableiten einer Ontologie aus einem geordneten Text (engl. *canonical text*), im vorliegenden Fall aus der hierarchischen Inhaltstabelle und dem Index eines Lehrbuches, dessen Inhalte Gegenstand der Diskussionen im Forum waren.
  - Jeder Begriff der Ontologie wird mit einer Häufigkeitsangabe versehen.
  - Identifizierte Themen werden in Themenvektoren übertragen (Elemente des Vektors sind Begriffe aus den Unterthemen) und können dann mit Nachrichtenvektoren verglichen werden.
- Klassifikation von Nachrichten, um Beziehungen zwischen ihnen und dem ganzen Thread bestimmen zu können, zwei Ansätze um Nachrichten zu klassifizieren:
  - *Classify-as-a-whole*-Strategie: alle Nachrichten eines Threads werden als Dokument betrachtet und als Vektor abgebildet.
  - *Classify-by-dominance*-Strategie: verschiedene Kriterien zu Bestimmung der Dominanz (z. B. maximale Anzahl von Nennungen eines Themas in allen Nachrichten)
- Empirische Auswertung der verschiedenen Klassifizierungsstrategien, gutes Abschneiden der entwickelten *classify-by-dominance* Strategie

#### 5. PINKWART ET AL. (2006): Toward Legal Argument Instruction with Graph Grammars and Collaborative Filtering Techniques<sup>25</sup>

Ziel: Entwicklung eines intelligenten Tutor-Systems (ITS), Unterstützung des Lernprozesses von Studenten, grafische Repräsentation von Argumenten

Methode: Entwicklung eines ITS-Prototyps

Kerninhalte:

- Der vorgestellte Ansatz verwendet keine *Natural-Language-Processing* (NLP)-Techniken.
- Bisher existieren kaum ITS, die Studenten bei der Konstruktion von Argumenten unterstützen (Ausnahmen: CATO, ArguMed).
- Der juristische Bereich ist nicht wohldefiniert im mathematischen Sinne, d. h. „korrekte“ Lösung nicht zwingend eindeutig → es können drei Arten von Schwachstellen identifiziert werden:
  1. *Strukturelle Schwachstellen*: isolierte Elemente, fehlende Tests etc.

<sup>23</sup> Vgl. FENG ET AL. (2006)

<sup>24</sup> <http://horicky.blogspot.de/2009/01/solving-tf-idf-using-map-reduce.html>

<sup>25</sup> Vgl. PINKWART ET AL. (2006b)

- Erkennung durch logische Formalismen, z. B. *Graph Grammar*,
  - Sehr gut adaptierbar auf Graphstrukturen von Argumentationen,
  - Deklarativer Charakter: Regeln können leicht als Parameter eines generischen Parsing-Algorithmus deklariert werden.
2. *Kontextuelle Schwachstelle*: fehlende Relationen zwischen Graph und Abschrift,
- Erkennung durch gleiche Graph Grammar wie in (1.),
  - Explizite Definition von nicht-relevanten Textstellen durch manuelles Markieren
3. *Inhaltliche Schwachstelle*: „collaborative filtering“ durch Feedback anderer Nutzer des Systems, Errechnung einer Wertung (Score) als Indikator für Qualität einer Lösung
- Einsatzbereich: Abschrift mündlicher Diskussionen des US Supreme Court, Anwälte formulieren Hypothesen (Vorschläge zur Lösung eines Problems), Richter prüfen diese Hypothesen auf Konsistenz durch Anwendung von Testscenarien → gute Lernressource für Jurastudenten.
  - Grafische Darstellung der Dokumentstruktur macht die Informationen für *intelligente Support-systeme* (ISS) zugänglich, Vorschlag einer speziellen Repräsentation zur Modellierung von Hypothesen und Tests.
  - Implementierung eines Prototyps aufbauend auf dem *Cool Modes Framework*<sup>26</sup>

#### 6. WEINBERGER AND FISCHER (2006): A Framework to Analyze Argumentative Knowledge Construction in Computer-Supported Collaborative Learning<sup>27</sup>

Ziel: Entwicklung eines Frameworks mit dem Ziel, den Einfluss von *Computer-Supported Collaboration Scripts* (CSCS) auf den Prozess der Konstruktion von Argumenten in Lerngruppen zu bestimmen

Methode: Entwicklung eines Frameworks, gestaltungsorientierter Ansatz

- Kerninhalte:
- Durch Analyse von Diskussionsforen sollen kognitive Lernprozesse erforscht werden.
  - Multidimensionaler Ansatz (Dimensionen *Teilnehmer*, *Erkenntniswert*, *Argument* und *soziale Aspekte*) soll zur Bestimmung des Beitrags von CSCS zu individueller Wissensaneignung dienen.
  - Das Framework hat vier Dimensionen:
    - *Teilnehmer*: Anzahl der Beiträge, Heterogenität der Beiträge
    - *Erkenntniswert*: on-topic- / off-topic-Anteil der Diskussion, teilnehmerspezifische Lernstrategie,
    - *Argument*: Konstruieren und Abwägen von Argumenten, Mikro- / Makro-Level-Darstellung von Argumenten (Mikro-Level nach Toulmin)
    - *Soziale Aspekte*: Interaktion innerhalb der Gruppe, Eingehen auf Beiträge etc., verschiedene Kategorien.
  - Zu analysierender Corpus wurde manuell segmentiert und nach einem festgelegten Schema kategorisiert.
  - Diskussion von Ansätzen zur Analyse von Segmenten innerhalb der vier Dimensionen (z. B. wortbasierte Typisierung von Sätzen als „qualifiziertes Argument mit Begründung“ in der Dimension *Argument*)
  - Durchführung einer Studie zur Evaluation des Frameworks, Ergebnisse: jede Script-Komponente hat positive Einflüsse auf das Lernen in dem Bereich, für den sie entwickelt wurde. Allerdings treten auch negative Nebeneffekte in Bereichen auf, für die eine Script-Komponente nicht entwickelt wurde.

#### 7. GORDON (2007): The Carneades Model of Argument and Burden of Proof<sup>28</sup>

Ziel: Präsentation eines formalen mathematischen Modells zur Strukturierung und Evaluation von Argumentation, Modellierung von *Beweislast* und *Beweisstandards*

Methode: Konzeptentwicklung

- Kerninhalte:
- *Carneades Modell*, basierend auf WALTONS Theorie von Argumentationsschemata → Formalisierung des Schemas inklusive kritischer Fragen zur automatischen Schlussfolgerung von Systemen
  - Unterscheidung von drei Arten von Prämissen:
    - normale Prämissen (engl. *ordinary premises*),
    - Annahmen und
    - Ausnahmen.
  - Differenzierung des Begriffs *burden of proof* (Beweislast) in *burden of production* (meist beim Kläger: Beweise für Anschuldigung bereitstellen) und *burden of persuasion* (meist beim Vertei-

<sup>26</sup> Vgl. PINKWART (2005)

<sup>27</sup> Vgl. WEINBERGER ET AL. (2006)

<sup>28</sup> Vgl. GORDON ET AL. (2007)

- diger: Ausnahmen von Anschuldigungen finden); kritische Fragen der Schemata beziehen sich auf *burden of production*; Grundannahmen des Systems über die Verteilung der Beweislast.
- Argumentgraphen zur Definition der logischen Struktur:
    - Aussagen (engl. *statements*): Syntax einer Aussage ist unwichtig, System muss nur entscheiden können, ob zwei Aussagen syntaktisch gleich sind und muss das logische Komplement bestimmen können.
    - Prämissen: Unterscheidung in drei Typen (siehe oben)
    - Argumente: Tupel aus Aussage, pro- / con-Attribut und Prämisse.
  - Argumentgraphen haben zwei Arten von Knoten: Aussageknoten und Argumentknoten. Die Kanten stellen entweder Prämissen oder Schlussfolgerungen dar. Argumentgraphen stellen eine Art von Beweisbäumen dar: Argumente sind dann akzeptabel, wenn ihr Argumentgraph die Aussage beweist.
  - Vergleich der Aussagekraft des vorgestellten Modells mit Toulmin und anderen Modellen
  - Evaluation von Argumenten hängt vom Status des Arguments (*verkündet, in Frage gestellt, akzeptiert, zurückgewiesen*) sowie vom zugewiesenen Beweisstandard ab → Definition des Kontexts eines Arguments, Vordefinition von drei Beweisstandards (Erweiterbar durch Nutzer).
  - Modellierung von kritischen Fragen in *Carneades*: schemaabhängige Modellierung als Ausnahme oder Annahme
  - Anwendung des Frameworks wird in einem Fallbeispiel gezeigt.

#### 8. MOCHALES PALAU UND MOENS (2007): ACILA - Automatic Detection of Arguments in Legal Cases<sup>29</sup>

Ziel: Vergleich und Evaluation verschiedener Ansätze zur automatischen Erkennung von Argumentationsstrukturen und Argumenten in einem juristischen Text

Methode: Experimente, Evaluation

- Kerninhalte:
- Automatische Klassifikation eines Arguments nach bestimmten Typen (Gegenargument, Gegenbeweis) wird angestrebt.
  - Automatische Qualifizierung eines Arguments als relevant für die Entscheidung und Beschreibung durch Elemente einer zugeordneten Ontologie → ermöglicht das Schlussfolgern in einem fall-basierten Argumentationssystem.
  - Arbeit entstand im Rahmen des ACILA Projektes (2006-2010), Ziele: Studie juristischer Argumentationsstrukturen, Konstruktion einer Taxonomie für rhetorische Redestrukturen, linguistische Schlagwörter, NLP juristischer Texte, automatische Klassifizierung von Argumenten, Visualisierung
  - Erkennung von Argumenten wird als Klassifizierungsproblem betrachtet. Verschiedene Ansätze werden evaluiert und verglichen.
  - Konzentration auf Abbildung von juristischer Argumentation in geschriebener Sprache, rhetorische Struktur als Hauptindikator, insbesondere corpus-spezifische lexikalische Stichwörter (engl. *lexical cues*) sind wichtige Hinweise auf die Textstruktur.
  - Repräsentation eines Satzes als Vektor (Verwendung eines VSM), Dimensionen des Vektors entsprechen Merkmalen, die einfach aus dem Text zu extrahieren sind (z. B. Wortpaare, N-Gramme, Verben, Schlüsselwörter etc.).
  - Vektoren dienen als Eingabe für Klassifizierungsalgorithmen (*Multinomial Naive Bayes*, *Maximum Entropy Model*).
  - Manuelle Annotation eines Testkorpus nach Schemata des *Araucaria*-Systems und Vergleich der Textklassifizierung mit automatisierten Durchgängen unter Verwendung verschiedener Merkmale
  - Prinzipielle Erkennung auf Basis der rhetorischen Struktur gegeben; Inhalte aus „umgebenden Sätzen“ relativ zum gerade betrachteten Satz ist noch nicht implementiert; Hauptproblem ist die Mehrdeutigkeit linguistischer Schlüsselwörter; zukünftige Erweiterung um *relationale Markov-Netzwerke*.

<sup>29</sup> Vgl. MOCHALES PALAU ET AL. (2007)

**9. RAVI und KIM (2007): Profiling Student Interactions in Threaded Discussions with Speech Act Classifiers<sup>30</sup>**

Ziel: Klassifizieren von Forenbeiträgen als Frage oder Antwort und Identifizieren von offenen Fragen in Online-Diskussionen

Methode: Prototyp und Evaluation

Kerninhalte:

- Bezug auf *Speech-Act*-Theorie, wonach jede Nachricht in einer Diskussion *Frage*, *Antwort*, *Korrektur* oder *nähere Ausführung* ist.
- Entwicklung zweier Classifier: *Question Classifier* (QC, Identifikation einer Frage) und *Answer Classifier* (AC, Identifikation von Antworten).
- *Thread profiler* zur Klassifikation von Diskussionen → Erkennung von Diskussionen mit unbeantworteten Fragen
- Texte aus Foren sind häufig *unzusammenhängend* (in Bezug auf grammatikalische Struktur) und „rauschend“ (engl. *noisy*, in Bezug auf die Verwendung individueller Wörter pro Nutzer) → verschiedene Maßnahmen, um Rauschen zu reduzieren, z. B.:
  - *Stemming*: entfernen der Pluralendungen zur Vereinheitlichung von Begriffen auf Wortstämme,
  - Ersetzung von Personalpronomen durch *categ\_person* und technischer Begriffe durch *tech\_term* und
  - Austausch von Smileys durch einheitliche Worte
- Auswahl der Merkmale (engl. *features*) für Verwendung in SVM: *N-grams* (Folgen von *n* Wörtern); *Information Gain Theory*<sup>31</sup> zur Auswahl wichtiger Merkmale durch Berechnung einer Maßzahl dafür, wie stark das Vorhandensein / Nicht-Vorhandensein eines Merkmals die Klassifikation beeinflussen würde → für jedes Merkmal wird der *gain*-Wert für den QC und AC berechnet → Sortierung nach diesem Wert und Auswahl der ersten 200 Merkmale
- SVM-Implementierung der QC und AC als Merkmalsvektoren → Genauigkeit der Erkennung liegt bei 88% für QC und 73% für AC
- Beim *Thread Profiling* wird der Speech Act jedes Threads bestimmt und dieser in eine der folgenden vier Kategorien eingeordnet:
  - Thread enthält keine Frage,
  - Thread enthält Frage, die beantwortet wird,
  - Thread enthält mehrere Fragen, die alle beantwortet werden, und
  - Thread enthält offene Fragen.

**10. SOMBEKKE (2007): Argumentation Structures in Legal Dossiers<sup>32</sup>**

Ziel: Case Study zum Einsatz von *Argumentation Management Systemen*, die dem Nutzer die Strukturierung juristischer Dossiers zu einem Rechtsfall erlauben (zur Erleichterung der Übersicht über einen Fall, insbesondere im zeitlichen Verlauf einer Gerichtsverhandlung)

Methode: Fallstudienmethodik

Kerninhalte:

- Das vorgeschlagene System besitzt keine Automatismen, sondern bietet dem Nutzer lediglich die Möglichkeit, manuell Argumentationsstrukturen in einem Text zu definieren.
- Die Struktur umfasst Hauptinhalte, Positionen und Argumente.
- Zusätzlich zur statischen Struktur eines Dokuments wird die Veränderung von argumentativen Strukturen im Verlauf der Zeit betrachtet.
- Vorgehen bei Analyse im Rahmen der Fallstudie:
  - Unterstützungsbeziehungen zwischen Aussagen innerhalb eines Arguments und Konfliktbeziehungen wurden identifiziert.
  - Anschließend folgte die Identifizierung impliziter Bedingungen.
  - Schließlich wurden die gefundenen Argumente mit Argumentationsschemata aus der einschlägigen Literatur abgeglichen.
- Zur Analyse der Entwicklung der argumentativen Struktur eines Rechtsfalls wurde das Dossier in sieben Phasen unterteilt.
- Zukünftige Nutzung von XML-Schemata ist geplant.

<sup>30</sup> Vgl. RAVI ET AL. (2007)

<sup>31</sup> Vgl. YANG ET AL. (1997).

<sup>32</sup> Vgl. SOMBEKKE ET AL. (2007)

**11. ABBAS und SAWAMURA (2008): A First Step Towards Argument Mining and its Use in Arguing Agents and ITS<sup>33</sup>**

Ziel: Konzeptentwicklung zur Speicherung von Argumentationsstruktur in relationalen Datenbanken und Vorstellung eines Frameworks zum IR der abgelegten Informationen

Methode: Entwicklung eines Frameworks

- Kerninhalte:
- Verwendung von relationalen Datenbanken zur Speicherung von Argumentationsstruktur
  - Argumente können in Form von Knoten und gerichteten Kanten repräsentiert werden.<sup>34</sup> Wenn zyklische Graphen ausgeschlossen werden, ergibt sich hieraus eine gerichtete Baumstruktur, die in eine relationale Datenbank übertragen werden kann.
  - Jedes Argumentationsschema nach WALTON kann in einer baumartigen Struktur dargestellt werden → der Ansatz der Autoren sieht nun die Aufteilung der Informationen eines Schemas auf drei Tabellen vor:
    - *Scheme\_TBL*: enthält ID und Namen für jedes Schema,
    - *Scheme\_Struct\_TBL*: enthält detaillierte Informationen zu jedem Schema, bspw. den Typ (Prämisse, Schlussfolgerung etc.),
    - *Data\_TBL*: enthält die Nutzerdaten, die bei einer Analyse anfallen; eine Spalte nimmt z. B. den Text auf, der zum angegebenen Argumentationsschema passt. Die Typ-Spalte gibt eine Unterstützung oder Ablehnung an. Die Spalte *Child\_Of* ermöglicht den Aufbau der Baumstruktur.
  - Zur Durchführung von Argument Mining entwickeln die Autoren ein dreiteiliges Framework:
    - *Parser* Modul: erhält eine Anfrage vom Nutzer als String; dieser String wird in Tokens zerlegt und von unnötigen Wörtern (Artikel, Pronomen) befreit. Der Sinn dessen liegt vor allem in der Reduktion der Anfrage, um unnötige Iterationen des nächsten Moduls zu vermeiden.
    - *Classifier Agent*: führt die Anfrage an die zugrundeliegende Datenbank durch und klassifiziert die gefundenen Dokumente nach Wunsch des Nutzers (z. B. nach Priorität, Polarität, Schema); dabei werden iterativ alle Kombinationen der vom Parser übermittelten Wörter gesucht.
    - *ITS*: der Nutzer erhält die Möglichkeit, die gefundenen Daten zu analysieren und basierend auf dem gewählten Schema Argumente auszuwählen.

**12. ABBAS und SAWAMURA (2008): Argument Mining Using Highly Structured Argument Repertoire<sup>35</sup>**

Ziel: Vgl. ABBAS und SAWAMURA (2008) a first step towards argument mining and its use in arguing agents and ITS<sup>36</sup>

Methode: “

Kerninhalte: “

**13. KIM ET AL. (2008): Scaffolding On-line Discussions with past Discussions - An Analysis and Pilot Study of Pedabot<sup>37</sup>**

Ziel: Entwicklung eines Systems zur aktiven Unterstützung und Stimulation von Onlinediskussionen durch Präsentation von vorhanden Antworten und Beiträgen aus Diskussionen im Corpus, außerdem Ansätze des Informationen Retrieval zum Auffinden ähnlicher Beiträge

Methode: Implementierung eines Prototyps, Evaluation in einer Fallstudie

- Kerninhalte:
- Entwicklung der Software *PedaBot*, die aktiv Diskussionen in Onlineforen unterstützt; Vorschlag von Nachrichten aus vorherigen Threads; automatisches Retrieval; Integration in modifizierte Version von phpBB
  - Prozess der Analyse und des Retrieval als *Retrieval Pipeline*:
    - *TextTiling*<sup>38</sup> zur Unterteilung der Nachrichten in semantisch verwandte Kategorien,

<sup>33</sup> Vgl. ABBAS ET AL. (2008b)

<sup>34</sup> Vgl. CHESÑEVAR ET AL. (2006)

<sup>35</sup> Vgl. ABBAS ET AL. (2008a)

<sup>36</sup> Vgl. ABBAS ET AL. (2008b)

<sup>37</sup> Vgl. KIM ET AL. (2008)

<sup>38</sup> Vgl. HEARST (1994)

- Texte aus den Foren sind *unzusammenhängend* (in Bezug auf grammatikalische Struktur) und „rauschend“ (engl. *noisy*, in Bezug auf die Verwendung individueller Wörter pro Nutzer) → automatische Erstellung einer Datenbank von technischen, im Forum verwendeten Wörtern aus Inhaltsverzeichnissen des dem Forum zugrundeliegenden Lehrbuchs,
- Darstellung von Nachrichten als Vektoren mit Boolean-Werten für Vorhandensein der einzelnen technischen Wörter,
- *Topic Profiler* erzeugt einen Themenvektor aus thematischen Abschnitten des Lehrbuchs; Berechnung eines *term weight vectors* für jede Themenkategorie (*TF-IDF* Verfahren),
- *Term weights* werden zur Berechnung der Ähnlichkeit zwischen Nachrichten im Corpus und neuer Nachricht verwendet,
- *Latent Semantic Analysis* (LSA) zur Bestimmung der Beziehung zwischen Begriffen (engl. *terms*),<sup>39</sup>
- Abgleich von neuer Nachricht und Nachrichten im Corpus über Bestimmung der Kosinus-ähnlichkeit der Nachrichtenvektoren,
- Zwei Studien zur Evaluation der Ergebnisse; durchschnittliche Bewertung der präsentierten Vorschläge des Systems („somewhat useful“).

#### 14. MOCHALES AND MOENS (2008): Study on the Structure of Argumentation in Case Law<sup>40</sup>

Ziel: Formalisierung der sprachlichen und strukturellen Eigenschaften von Argumentation in juristischen Rechtsfällen und die Entwicklung einer kontextfreien Grammatik zur automatischen Erkennung des argumentativen Prozesses in einem Dokument

Methode: Entwicklung und Evaluation einer kontextfreien Grammatik

- Kerninhalte:
- Untersuchungsgegenstand ist natürlichsprachliche Argumentation in Texten des *European Court of Human Rights* (ECHR); die Texte waren (gegliedert nach verschiedenen Kategorien) online zugänglich.
  - Fokus liegt auf der Erkennung der Funktion einzelner Sätze im argumentativen Prozess.
  - Die vorliegenden Dokumente folgen zwar keiner streng formalen Struktur, wie sie etwa *MetaLex* bietet, besitzen jedoch fast immer die gleiche inhaltliche Gliederung in 5 Abschnitte (nicht alle Abschnitte tragen in gleichem Maße zur Argumentation bei):
    - *Introduction*,
    - *the Facts*,
    - *Proceedings before the Commission*,
    - *Complaints* und
    - *the Law (Reasons for the Decision)*.
  - Die vorgestellte argumentative Analyse basiert auf verschiedenen Argumentationsschemata, die im Gegensatz bspw. zu Formalismen nach TOULMIN eine Entscheidungsfindung auch bei Unsicherheit und Unvollständigkeit zulassen sollen (engl. *presumptive reasoning*).
  - Zwei Anwälte annotierten unabhängig von einander eine Auswahl an ECHR-Dokumenten hinsichtlich vorgegebener Argumentationsschemata und deren Beziehung untereinander (Übereinstimmung nach Überarbeitung 80%).
  - Ergebnis der Untersuchung: Argumente in bestimmten Abschnitten wurden immer von den gleichen Parteien vorgetragen (z. B. Kläger und Verteidiger im Abschnitt *The Law*); Unterteilung in sogenannte:
    - *reported arguments*: in Vergangenheitsform (*past tense*), werden als Fakten betrachtet und können im laufenden Prozess nicht attackiert werden.
    - *non-reported arguments*: in Präsensform (*present tense*), Argumente welche die aktuelle Entscheidung beeinflussen, wurden vom sogenannten *Factfinder* (in der ECHR ist dies eine Kommission) zusammengestellt. Diese Argumente sind für die Entscheidung des Falls am wichtigsten.
  - Wiederkehrende sprachliche Muster (rhetorische Marker) in den Dokumenten wurden identifiziert. Abgrenzung von generellen Ausdrücken sprachlicher Argumentation gegenüber Ausdrücken die typisch für juristische Argumentation sind (Worttabelle); Unterscheidung zweier Ebenen:
    - eine Schlussfolgerung, die zu einer Entscheidung führt, ist abgeleitet von den unterstützenden Prämissen,
    - eine Schlussfolgerung, die die Prämissen unterstützt, die wiederum zur ersten Schlussfolgerung beitragen.

<sup>39</sup> Vgl. LANDAUER ET AL. (1997)

<sup>40</sup> Vgl. MOCHALES ET AL. (2008)

- Entwicklung einer kontextfreien Grammatik zur Bottom-up-Erstellung einer Argumentstruktur; Grammatik arbeitet auf Satzebene; Implementierung in Java und JSCC.<sup>41</sup>
- Gute Erkennung von Entscheidungen; es ergeben sich zwei Problembereiche bei der Erkennung:
  - Schlussfolgerungen ohne rhetorische Marker,
  - Mehrdeutigkeit zwischen Argumentationsstrukturen (z.B. Erkennung von *premise* → *conclusion* → *conclusion* vs. (*premise & premise*) → *conclusion*)

#### 15. PINKWART ET AL. (2008): Graph Grammars - An ITS Technology for Diagram Representations<sup>42</sup>

Ziel: Entwicklung einer Grammatik zur Formalisierung von Argumentationsdiagrammen und zur Überprüfung modellierter Diagramme auf Konsistenz und potentielle Fehler

Methode: Konzeptentwicklung

- Kerninhalte:
- Juristischer Anwendungsbereich, mündliche Anhörungen des US Supreme Court als Testumfeld; Anwälte formulieren Vorschläge, wie ein Fall zu entscheiden ist, und präsentieren Tests, um ihre Vorschläge auf Sinnhaftigkeit zu testen; Richter überprüfen die Vorschläge durch Anwendung in hypothetischen Szenarien.
  - Die Autoren verwenden das *LARGO*-System<sup>43</sup> und erweitern es um eine Graph-Grammatik, welche die Grundlage für die Diagrammodellierung in *LARGO* darstellt.
  - Eine Graph-Grammatik besteht aus einer Menge von Symbolen, einem Startaxiom und einer Menge von Produktionsregeln; Worte innerhalb einer Grammatik repräsentieren vom Nutzer erstellte Diagramme.
  - Worte einer Graph-Grammatik sind 4-Tupel der Form  $(N, E, M, C)$ ;  $N$  bezeichnet eine Menge an Knoten,  $E$  eine Menge an Kanten,  $M$  bezeichnet Metadaten als Zusammenfassung von Informationen über ein Element (z. B. Anzahl der Argumente) und  $C$  stellt Charakteristiken in Form von Metadaten dar.
  - Die Grammatik enthält zwei Arten von Produktionsregeln:
    - *Generierungsregeln* (engl. *generation rules*): kommen bei der Erstellung von Diagrammen zur Anwendung; es kann vom Nutzer nur das modelliert werden, was explizit als Produktionsregel vorhanden ist.
    - *Feedbackregeln* (engl. *feedback rules*): werden manuell auf Wunsch des Nutzers aufgerufen und auf alle modellierten Diagrammelemente angewendet; Erkennung von strukturellen Schwächen im Graph (z. B. unwahrscheinliche Verbindungen zweier Elemente).
  - Priorisierung und Auswahl der identifizierten Schwachstellen und Präsentation des Feedbacks stellt eine große Herausforderung dar.

#### 16. REED ET AL. (2008): Language Resources for Studying Argument<sup>44</sup>

Ziel: Entwicklung eines annotierten Corpus zur Forschung im Bereich „Argumentation Theory“, „Argumentation Mining“ etc.

Methode: Manuelle Erstellung des Corpus

- Kerninhalte:
- Entwicklung des eigenen Corpus *AraucariaDB*, der ausschließlich Texte mit argumentativem Inhalt enthält (im Gegensatz zu bestehenden allgemeinen (*general text*) Corpora).
  - Manuelle Annotation von Texten mittels *Araucaria*, Verwendung der *Argument Markup Language* (AML) zur Darstellung von Beziehungen zwischen Textelementen und Argumentationsstrukturen, 2004 Erweiterung um Ontologie auf Basis von Argumentationsschemata nach WALTON.<sup>45</sup>
  - Herausforderungen während der Annotation:
    - Vergleichbarkeit bei Annotationen durch unterschiedliche Personen: je nach subjektiver Vorgehensweise kommt es zu unterschiedlichen identifizierten Strukturen im Corpus.
    - Unzulänglichkeiten des AML-Formats: führt zur Entwicklung des *Argument Interchange Format* (AIF).
    - Zeitintensiv: Analyse einiger hundert Wörter dauert mehrere Stunden.
  - Analyse der 2003er-Version des Corpus in REED (2005)<sup>46</sup>
  - Anwendbarkeit des Corpus bspw. zur automatischen Erkennung von Argumenten

<sup>41</sup> <http://jscc.jmksf.com/>

<sup>42</sup> Vgl. PINKWART ET AL. (2008)

<sup>43</sup> Vgl. PINKWART ET AL. (2006a)

<sup>44</sup> Vgl. REED ET AL. (2008)

<sup>45</sup> Vgl. WALTON (1996)

<sup>46</sup> Vgl. REED (2005)

**17. MOCHALES AND IEVEN (2009): Creating an Argumentation Corpus - Do Theories Apply to Real Arguments?**<sup>47</sup>

Ziel: Entwicklung eines annotierten Corpus mit juristischen Texten des *European Court of Human Rights* (ECHR), der als Trainingscorpus insbesondere für ML-Techniken im Bereich Argumentation Mining dienen kann; Beobachtungen sollen auf allgemeine Argumentationen verallgemeinert werden.

Methode: Vorstudie zur Entwicklung eines annotierten Corpus

- Kerninhalte:
- Juristische Texte des ECHR eignen sich besonders, da sie im Gegensatz zu natürlichsprachlichen Texten eine relativ strenge formale Struktur haben.
  - Die ECHR-Rechtsprechung basiert auf Präzedenzfällen; die meisten Fälle folgen dem gleichen Schema: 1.) der Kläger bringt Fakten und Gesetze vor, die zu einer von ihm gewünschten Entscheidung des Falls führen sollen, 2.) der Angeklagte kann die Anschuldigungen entweder ablehnen oder zusätzliche Fakten anführen, die einen anderen Ausgang zur Folge haben können, 3.) die Richter begutachten alle Fakten und leiten aus den Gesetzen eine rechtliche Entscheidung ab.
  - Die Annotation eines Corpus hängt von drei Faktoren ab:
    - **Daten**
      - Die Wahl des ECHR wird dadurch begründet, dass das Gericht ein eigenes Muster zur Schlussfolgerung entwickelt hat.
      - Die Dokumente sind *per se* bereits strukturiert (Abschnitte für einzelne Teile des Prozesses wie Entscheidung etc.).
      - Aufgrund der strukturierten Darstellung der Argumentation in ECHR-Fällen lässt sich eine Trennung vornehmen in Argumente, die von Kläger und Angeklagtem vorgebracht wurden und keinen Einfluss auf den Verlauf des Verfahrens haben (sog. *reported arguments*), und in Argumente, die von den Richtern erarbeitet werden und die Entscheidung beeinflussen (sog. *current argumentation*).
    - **Theoretisches Framework:** Bestimmung der Informationen, die im annotierten Corpus erfasst werden sollen; Verwendung von zwei etablierten Argumentationstheorien:
      - *Pragma-dialectics:* Argumentation wird immer als Versuch angesehen, eine Partei von der eigenen Meinung zu überzeugen; Unterscheidung verschiedener Argumentationswege (einfache Argumentation, Argumentation durch mehrere Argumente, Argumentationsketten).
      - *Argumentationsschema:* Zuordnung von Argumentationsschemata nach WALTON durch Anwendung der im Schema enthaltenen kritischen Fragen
      - *Framework der Autoren:* Treffen einiger vereinfachender Annahmen zu Schemata (alle Prämissen müssen im Argument vorhanden sein); geringfügige Modifikation der Schemaliste nach WALTON; Ausschluss einiger Schemata
    - **Annotatoren:** zwei Gruppen unterschiedlicher „Professionalität“
  - Diskussion unterschiedlicher Annotation bei unterschiedlichen Annotatoren; Hinweis auf verschiedene Interpretationsmöglichkeiten und Möglichkeiten zur Vermeidung von Fehlern

**18. MOCHALES-PALAU und MOENS (2009): Argumentation Mining – The Detection, Classification and Structure of Arguments in Text**<sup>48</sup>

Ziel: Integration der Konzepte aus vorherigen Ansätzen in einer umfassenden Arbeit, Klassifizierung von argumentativen Grundeinheiten, argumentativen Strukturen, Beziehungen zwischen Argumenten; Entwicklung von Mining-Ansätzen zur Argumententdeckung, Klassifizierung und Erkennung von Argumentationsstrukturen, Evaluation mit Araucaria und ECHR-Corpus

Methode: Konzeptentwicklung, Implementierung, Evaluation

- Kerninhalte:
- Weiterentwicklung der 2009 in MOCHALES AND IEVEN (2009): **Creating an Argumentation Corpus - Do Theories Apply to Real Arguments?**<sup>49</sup> vorgestellten Inhalte
  - Vorläufer zu MOCHALES AND MOENS (2011): **Argumentation Mining**<sup>50</sup> → Kerninhalte werden dort zusammengefasst.

<sup>47</sup> Vgl. MOCHALES ET AL. (2009)

<sup>48</sup> Vgl. MOCHALES-PALAU ET AL. (2009)

<sup>49</sup> Vgl. MOCHALES ET AL. (2009)

<sup>50</sup> Vgl. MOCHALES ET AL. (2011)

### 19. ABBAS und SAWAMURA (2010): Argument Mining from RADB and its Usage in Arguing Agents and Intelligent Tutoring Systems<sup>51</sup>

Ziel: Vorstellung von Techniken zum Argument Mining und deren Anwendung in der relationalen Argumentdatenbank (*Relational Argument Database (RADB)*) sowie Integration der Konzepte in ein *Intelligent Tutoring System (ITS)*

Methode: Konzeption und Implementierung eines ITS und Argument-Mining-Systems

- Kerninhalte:
- Weiterentwicklung der in **ABBAS und SAWAMURA (2008)** vorgestellten Inhalte
  - Verwendung von relationalen Datenbanken zur Speicherung von Argumentationsstrukturen
  - Laut AIF-Ontologie<sup>52</sup> können Argumente in der Form von Knoten und gerichteten Kanten repräsentiert werden; wenn zyklische Graphen ausgeschlossen werden, ergibt sich hieraus eine gerichtete Baumstruktur, die in eine relationale Datenbank übertragen werden kann; Es sind zwei Arten von Knoten in Graphen zu unterscheiden:
    - Informationsknoten (*I-nodes*): tragen Information oder Daten, z. B. Behauptungen, Thesen, Beweise etc.,
    - Schemaknoten (*S-nodes*): repräsentieren die Anwendung von Schemata, können weiter unterschieden werden in 3 Arten.
  - Jedes Argumentationsschema nach WALTON kann in einer baumartigen Struktur dargestellt werden → der Ansatz der Autoren sieht nun die Aufteilung der Informationen eines Schemas auf drei Tabellen vor (*Relational Argument Database (RADB)*):
    - *Scheme\_TBL*: enthält ID und Namen für jedes Schema,
    - *Scheme\_Struct\_TBL*: enthält detaillierte Informationen zu jedem Schema, bspw. dem Typ (Prämisse, Konklusion etc.),
    - *Data\_TBL*: enthält die Nutzerdaten, die bei einer Analyse anfallen; eine Spalte nimmt z. B. den Text auf, der zum angegebenen Argumentationsschema passt, die Typ-Spalte gibt eine Unterstützung oder Ablehnung an; die Spalte *Child\_Of* ermöglicht den Aufbau der Baumstruktur.
  - Die Speicherung von Informationen in Form von RADB erleichtert die Nutzung in unterschiedlichen Tools (oftmals Probleme bei Verwendung spezieller Dateiformate); RADB ist generisch genug, um Strukturen aus verschiedenen Domänen vorzuhalten.
  - SQL-Anfragen nach bestimmten Schemata oder Stichwörtern bspw. in einer Schlussfolgerung sind möglich.
  - Entwicklung eines Tools für die Datenbank, welches das Einpflegen und Updaten neuer Schemata und das Stellen von Suchanfragen nach verschiedenen Methoden ermöglicht,
  - Integration weitergehender Data-Mining (DM)-Techniken zur Klassifikation von Argumenten und Beantwortung von Suchanfragen nur durch die Präsentation relevanter Ergebnisse sowie zur Erkennung von versteckten Korrelationen zwischen verschiedenen Argumentteilen; Entwicklung eines *classifier agents*, zur Kontrolle folgender DM-Techniken:
    - *Apriori/Tid Mining Classification*: Identifikation des relevantesten Arguments bzgl. der Suchanfrage (Relevanz basierend auf Übereinstimmung mit dem Suchbegriff); Berechnung von möglichen Kombinationen der Suchbegriffe zur Erstellung einer Ordnung nach Unterstützungen,
    - *Tree Substructure Mining*: Breitensuche auf der Baumstruktur des RADB-Repository, Extraktion des Teilbaumes mit den meisten Knoten mit Bezug zum Suchbegriff,
    - *Rule Extraction Mining*: Erkennen von Substrukturen (Teilbäumen) im RADB-Repository durch Reduktion der Suchanfrage auf Schlüsselbegriffe; Ausgabe einer „Regel“
  - Integration der RADB-Struktur und des Mining Classifier in ein ITS zur Unterstützung des Lernvorgangs bei der Analyse und Konstruktion von Argumenten; vier Module (*Domain Model, Student Model, Interface Model, Pedagogical Model*)
  - Übertragung der entwickelten DM-Ansätze auf den Einsatzbereich von *Arguing Agents*; *Argument Transformation Rules* und *Knowledge Transformation Rules* zur Verbesserung der Argumente (z. B. Verfeinerung und Erweiterung)

<sup>51</sup> Vgl. ABBAS ET AL. (2010)

<sup>52</sup> Vgl. CHESÑEVAR ET AL. (2006)

## 20. GKOTSIS und KARACAPILIDIS (2010): On the Exploration and Exploitation of Structural Similarities in Argumentative Discourses<sup>53</sup>

- Ziel: Entwicklung eines generischen, berechenbaren Modells (engl. *computational model*) als Graphstruktur zur Identifikation und Bewertung von strukturellen Ähnlichkeiten in argumentativen Diskussionen
- Methode: Konzeptentwicklung und Anwendung in verschiedenen Use Cases
- Kerninhalte:
- Unterscheidung zwischen Mikrostruktur (Attribute eines Arguments) und Makrostruktur (zugrundeliegendes Argumentationsmodell)
  - Graphbasierter Ansatz zur Beschreibung von argumentativen Dialogen; ermöglicht die Quantifizierung von strukturellen Ähnlichkeiten; Betrachtungen auf Mikro- / Makrolevel möglich.
  - Grundlage: verbundener, gewichteter ungerichteter Graph; *Streitfrage* (engl. *issue*) ist Wurzel des Graphen; *Alternativen* werden durch neutrale Kantentypen mit Gewicht 0 an die Wurzel angefügt; *Argumente* werden mit anderen Argumenten oder Alternativen verbunden, unterstützende Argumente haben das Gewicht 1, ablehnende Argumente Gewicht -1:
    - *Ähnlichkeit* zwischen zwei Teilgraphen  $Sim(x,y)$  ist bestimmt durch die Gegenzahl der normalisierten Euklidischen Differenz.
    - *Vertex Refinement Query*  $H_i(x)$ : rekursive Methode, um strukturelle Ähnlichkeiten in benachbarten Knoten (der Tiefe  $i$ ) zu identifizieren,
    - Berechnung der *Sequence Similarity* zwischen zwei Knotenmengen: maximale Summe von Ähnlichkeiten zwischen Paaren von Knoten aus jeweils einer Menge,
    - *Vertex Equivalence*: Ähnlichkeit zweier verfeinerter Knoten  $Sim(H_i(x), H_i(y))$  liegt über einem Grenzwert *threshold*; *Vertex Identity*: Ähnlichkeit liegt für alle Werte  $t$  aus  $[0,i]$  über dem Grenzwert *threshold*.
  - Zwei Use Cases zur Demonstration des vorgestellten Verfahrens; Bestimmung der Ähnlichkeit / Identität zweier Argumente; mögliche Anwendungen:
    - Filtern nach widerlegten Argumenten,
    - Extraktion noch nicht beachteter Argumentsequenzen,
    - bei Verwendung zusätzlicher Attribute kann nach diesen gefiltert werden (z. B. „Zeige Argumentsequenzen, in denen User X User Y schlägt, aber von User Z geschlagen wird.“).

## 21. SCHEUER ET AL. (2010): Computer-Supported Argumentation: A Review of the State of the Art<sup>54</sup>

- Ziel: Überblick über den Stand der Forschung im Bereich *Computer-supported Argumentation* im Hinblick auf das Erlernen von Argumentation
- Methode: Literaturreview
- Kerninhalte:
- **Argumentation System:** Argumentationssysteme lassen sich nach ihrer Nutzung in Einzel- und Mehrbenutzersysteme differenzieren; die Arbeit nennt einige bekannte Systeme zur Darstellung von Argumenten, Nutzung von Ontologien, Visualisierung, Analyse und Feedback.
  - **Argument Representations:** Argumentationssysteme erlauben es dem Benutzer, Argumente grafisch zu erstellen und zu manipulieren; die Darstellung kann dabei nach folgenden Formen unterschieden werden: *linear*, *threaded*, *graph-based*, *container* und *matrix*; jede Darstellungsform verfügt über Besonderheiten und eignet sich meist besonders für einen bestimmten Zweck.
  - **Interaction Design:** Je nach System unterscheidet sich die Interaktion zwischen Software und Nutzer, z. B. im Freiheitsgrad bei der Erstellung von Argumenten oder in der Art der gemeinschaftlichen Zusammenarbeit mit anderen Nutzern durch dasselbe System.
  - **Ontologies:** Ontologien beschreiben die Komponenten von Argumentation und der Beziehungen untereinander; Ziel ist es auch, durch Limitierung der Gestaltungsmöglichkeit, Abschweifungen vom Thema zu verhindern; Ontologien lassen sich nach *educated ontologies* und *informal ontologies* unterscheiden.
  - **Automated Analysis:** Argumentationssysteme können die syntaktische Korrektheit modellierter Argumentationskonstrukte überprüfen und Schwächen im Argumentationsverlauf frühzeitig erkennen; die Arbeit listet eine Übersicht über verschiedene Ansätze zu automatisierter Analyse von Argumentation.
  - **Tutorial Feedback:** Zur Rückgabe von Feedback an den Nutzer durch das System existieren verschiedene Ansätze, zum einen hinsichtlich der Art des Feedbacks (textuell, metrisch bewertet etc.) und zum anderen hinsichtlich des Zeitpunkts (unmittelbar, summativ am Ende des Argumentationsprozesses etc.).

<sup>53</sup> Vgl. GKOTSIS ET AL. (2011)

<sup>54</sup> Vgl. SCHEUER ET AL. (2010)

- **Architecture and Technology:** Es existieren viele verschiedene Softwarearchitekturen und Systemansätze, die sich hinsichtlich Stabilität, Erweiterbarkeit und Geschwindigkeit unterscheiden.
- **Empirical Studies:** Der Einsatz von Argumentationssystemen kann bezüglich verschiedener Kriterien bewertet werden, z. B. der Repräsentierung von Wissen und der damit verbundenen Leitfunktion eines Systems für Studenten während des Argumentationsprozesses.

## 22. WYNER ET AL. (2010): Approaches to Text Mining Arguments from Legal Cases<sup>55</sup>

- Ziel: (1) *Information Extraction* zur Identifikation juristischer Entscheidungen und deren Begründungen, (2) Extraktion linguistischer Merkmale, die zum fallbasierten Schlussfolgern genutzt werden können.
- Methode: Literaturreview, Prototypentwicklung
- Kerninhalte:
- „Common Law“-Rechtssystem als Grundlage, d. h. fallbasierte Rechtsprechung
  - Verwendung von *Information Extraction* (IE), Ziel von IE ist die automatische Extraktion von strukturierten Informationen aus unstrukturierten, maschinenlesbaren Texten, wobei sich „Struktur“ hierbei auf semantische Eigenschaften und Beziehungen im Text bezieht (in Abgrenzung dazu dient IR vor allem der Identifizierung relevanter Dokumente, weniger der darin enthaltenen Informationen).
  - Besprechung von existierenden Ansätzen im Bereich *Argumentation Theory and Analysis*:
    - XML,
    - *Argumentation Frameworks*,
    - *Argumentationsschemata*,
    - *Defeasible Logic und Text Mining*.
  - Besprechung versch. Corpora, die zur Extraktion von Informationen genutzt werden können:
    - Araucaria,
    - MOCHALES und MOENS,<sup>56</sup>
    - WYNER und MILWARD.<sup>57</sup>
  - Vorstellung dreier Ansätze zur Analyse einer Menge von juristischen Rechtsfällen:
    - 1) Manuelle Markierung von Argumenten mittels *Argument Markup Language* (AML) in AraucariaDB durch Experten, Abbilden von Schemata und unterschiedlichen Thesen, Beziehungen, unterstützende (Meta-)Informationen (Auto, Datum der Analyse).
    - 2) *Kontextfreie Grammatik*: Entwicklung in Anlehnung an MOCHALES und MOENS<sup>58</sup> zur Identifikation von Argumentationsstrukturen.
    - 3) *Case-based Reasoning* (CBR): spezieller Ansatz im Common Law zur Identifikation von Präzedenzfällen; Präzedenzfälle werden anhand von „Faktoren“ identifiziert, wobei ein Faktor ein textuelles Muster ist, welches die Entscheidung eines Rechtsfalles beeinflusst; Faktoren werden mittels IE durch semantische Ähnlichkeiten gefunden, unabhängig von der textuellen Form.  
Zum Durchsuchen der Datenbasis an Rechtsfällen wurde *Linguamatics* verwendet, das u. a. folgende Funktionen bietet:
      - Suche mit regulären Ausdrücken,
      - Auflisten alternativer Wörter,
      - Suchen innerhalb eines syntaktischen Rahmens (Satz, Paragraph etc.) und
      - Integration von spezifischen Ontologien.
  - Diskussion von Nachteilen der vorgestellten Lösungen:
    - *Kontextfreie Grammatik*: (1) Satzteilstruktur für Sätze fordert, dass Satzteile in einer bestimmten Reihenfolge auftreten müssen, dass sie vollständig sein müssen, dass Argumente innerhalb der gegebenen Struktur auftauchen, (2) semantische / ontologische Informationen sind nicht integriert.
    - *Case-based Reasoning*: hoher manueller Aufwand bei der Erstellung von Ontologien.

<sup>55</sup> Vgl. WYNER ET AL. (2010)

<sup>56</sup> Vgl. MOCHALES ET AL. (2008)

<sup>57</sup> Vgl. WYNER ET AL. (2008)

<sup>58</sup> Vgl. MOCHALES ET AL. (2008)

**23. HOGENBOOM (2011): Mining Economic Sentiment Using Argumentation Structures<sup>59</sup>**

Ziel: Entwicklung eines Sentiment-Mining-Frameworks zur Ableitung von Informationen über die wirtschaftliche Stimmung aus Argumentationsstrukturen

Methode: Entwicklung eines Frameworks, gestaltungsorientierter Ansatz

- Kerninhalte:
- Ein wichtiges Element für wirtschaftliche Entscheidungen ist die Stimmung der Interessenvertreter. → Es besteht der Wunsch nach kontinuierlicher Überwachung dieser Variablen.
  - Wichtigstes zu überwachendes Konzept im Artikel ist die Kundenzufriedenheit (Zusammenhang mit der Bereitschaft, Geld auszugeben).
  - Ergänzung traditioneller makroökonomischer Indikatoren um Konzepte, welche die „generelle Stimmung“ erfassen → Analyse frei verfügbarer Texte durch Text Mining liefert Informationen zur Entscheidungsfindung.
  - Analyse der semantischen Struktur von Argumentation, da Argumentation immer auch Ausdruck von Meinungen ist,
  - Text Mining Ansätze nach FELDMAN und SANGER<sup>60</sup>
  - Verwendung des freien *Text-Processing*-Frameworks *GATE*<sup>61</sup> (Funktionen: Tokenization, PoS-Tagging, semantische Analyse etc.)
  - Die Auswertung einzelner Worte kann die intendierte Stimmung nicht immer erfassen bzw. kann für unterschiedliche Parteien verschieden interpretiert werden. → Bedarf nach semantischer Struktur; Argumentationsstruktur liefert Anhaltspunkte zum Zusammenhang einzelner Textabschnitte mit der Gesamtstimmung des Textes
  - Entwicklung einer *Information Extraction Pipeline* zur schrittweisen Abstraktion von Rohertexten zu formalisierten Textstücken, Wiederverwendung von Komponenten der GATE-Standard-Pipeline:
    - *Document Reset*: Entfernen von Artefakten (z. B. Tags),
    - *English Tokenizer*: Aufteilen von Text in Tokens (z. B. Worte),
    - *Sentence Splitter*: Aufteilung von Text in Sätze,
    - *PoS-Tagger*: Bestimmung der Wortfunktionen,
    - *Argumentation Gazetteer*: Erkennung von Markern für Argumentation durch eine spezifische Ontologie (enthält Marker und ihre Beziehungen zu argumentativen Textelemente), Ansatz zur Strukturierung von Argumentation in wirtschaftlichen Texten: Ausgangspunkt sind die Argumentationstheorien *Rhetorical Structure Theory* (RST) und *Textual Entailment* (TE),
    - *Argumentation Parser*: Identifikation von Textsegmenten und Bestimmung ihrer Rolle in der Argumentationsstruktur gemäß der Argumentationsontologie,
    - *Sentiment Analyzer*: identifiziert die Stimmung eines Textsegments und setzt es mit der verknüpften Argumentationsstruktur in Verbindung; abhängig von der Rolle eines Textsegments in der Argumentationsstruktur erhält es ein Gewicht für den Beitrag zur Gesamtstimmung des Texts.
 Ergebnis dieses Prozesses ist eine „on the fly“-Ontologie, die Wissen zur aktuellen wirtschaftlichen Stimmung im Text repräsentiert.
  - Die konkrete Umsetzung soll durch folgende Techniken realisiert werden: *Neural Networks*, *Self-Organizing Maps*, *Evolutionary Computation*, *Cluster Analysis*, *Bayesian Networks*.

**24. MOCHALES und MOENS (2011): Argumentation Mining<sup>62</sup>**

Ziel: Vorstellen von Methoden zur Erkennung und Analyse von Argumenten und Argumentationsstrukturen, Anwendung von ML-Techniken auf den Bereich *Argumentation Mining*, Entwicklung einer kontextfreien Grammatik, Aufzeigen weiterer Forschungsgebiete und -ansätze

Methode: Entwicklung von Konzepten, Implementierung und Evaluation

- Kerninhalte:
- Kombination von Ansätzen aus dem Bereich NLP, IR und Argumentation Theory → *Argumentation Mining*, Ziel: Argumentation in einem Text automatisch erkennen, d. h. alle Argumente und ihre individuelle und lokale Struktur sowie Beziehungen zwischen ihnen.
  - **Elementare Argumentationseinheiten:** zur Unterscheidung zwischen Aussagen und Behauptungen definieren die Autoren ein Argument als eine Menge von mindestens zwei Aussagen.

<sup>59</sup> Vgl. HOGENBOOM ET AL. (2010)

<sup>60</sup> Vgl. FELDMAN ET AL. (2006)

<sup>61</sup> Vgl. CUNNINGHAM (2002)

<sup>62</sup> Vgl. MOCHALES ET AL. (2011)

- **Interne Struktur zwischen Elementareinheiten:** Aussagen können entweder als Prämisse oder als Schlussfolgerung klassifiziert werden; die vorgestellte Arbeit basiert auf Argumentations-schemata nach WALTON.
- **Beziehungen zwischen Elementareinheiten:** Unterscheidung von Nebenordnung, Unterordnung und multipler Argumentation
- **Argumentationsstruktur:** Repräsentation als Baumstruktur, Blätter stellen die Argumente dar, Beziehungen werden nur zwischen Argumenten betrachtet.
- Experimente mit Hilfe zweier Corpora: allgemeiner englischer Corpus und ECHR-Corpus (siehe früher Arbeiten von MOCHALES und MOENS).
- **Erkennung von Argumenten:** jeder Textabschnitt wird als argumentativ oder nicht-argumentativ klassifiziert → die Summe aller argumentativen Abschnitte stellt die Argumentation des Textes dar; die Unterteilung argumentativer Abschnitte in einzelne Argumente wird als Segmentierungsproblem bezeichnet.
  - *Klassifizierungsproblem:* Identifizieren von argumentativen Abschnitten mit statistischen Classifier: *Naive Bayes*, *Maximum Entropy Model*, *Support Vector Machine (SVM)* → Klassifizierungssystem wird mit generischen Merkmalen trainiert → es werden nur Informationen genutzt, die direkt aus dem Text extrahiert werden können (Wörter im Text, ihre Position etc.).
  - *Grenzen von argumentativen Abschnitten:* 1.) Orientierung an der Struktur der zu analysierenden Texte (Abschnitte, Gliederung), 2.) semantische Merkmale zur Erkennung einzelner Argumente (Ontologie- / Corpus-basiert, semantischer Abstand zwischen Sätzen, Nutzung von Ontologien wie *WordNet*)
  - *Klassifizierung der identifizierten Aussagen als Prämisse oder Konklusion:* Nutzung statistischer Classifier (SVM) nach verschiedenen textuellen Merkmalen
  - *Entdeckung der argumentativen Gesamtstruktur des Textes:* Erkennung der Grenzen eines einzelnen Arguments und der Beziehungen zu umliegenden Argumenten → Entwicklung einer kontextfreien Grammatik basierend auf Erkenntnissen der *Rhetorical Structure Theory*; Regeln der Grammatik sind im Artikel enthalten.
- Evaluation der Grammatik am ECHR-Corpus, erreichte Genauigkeit von ungefähr 60%
- Anwendung der Methoden vor allem im juristischen Bereich: dieser bietet eine ausgewogene Mischung zwischen natürlichsprachlichen Freitexten und einer gewissen Strukturierung.

## 25. PALLOTTA und DELMONTE (2011): Automatic Argumentative Analysis for Interaction Mining<sup>63</sup>

Ziel: „Interaction Mining“-Ansatz zur Abbildung der argumentativen Struktur eines Texts über Strukturregeln, Erweiterung bestehender IR-Techniken auf Basis dieser Struktur, Entwicklung eines Algorithmus zur automatischen Analyse, Visualisierung von Argumentstrukturen

Methode: Erstellung und Implementierung eines Frameworks

- Kerninhalte:
- *Interaction Mining* ist ein Ansatz, um Informationen über menschliche Interaktion durch digitale Medien zu gewinnen.
  - Vorgestellte Ansätze sind universell, im Artikel allerdings Fokussierung auf Analyse von Konversationen (= zweckgebunden Interaktion) → Kontext einer Aussage ist sehr wichtig in Konversationen und häufig sind keine Schlüsselbegriffe vorhanden; Ansatz basiert auf der argumentativen Struktur eines Textes.
  - Unterscheidung zwischen synchronen und asynchronen *Digital Conversational Systems (DCS)* (Instant Messaging, VoIP vs. Foren, Blogs, Microblogs) → Mischung aus erzählerischer und argumentativer Struktur erschwert die Analyse von Texten asynchroner Systeme.
  - Grenzen traditioneller IR-Ansätze bei der Analyse der Beziehungen zwischen Teilnehmern einer Konversation (leichte Verbesserung durch automatische Umformulierung der Suchbegriffe)
  - Zwei Ansätze zur Erkennung argumentativer Strukturen:
    - *relationaler Ansatz* (engl. *relational*): basiert auf rhetorischen Beziehungen zwischen zwei oder mehr Dialogeinheiten (z. B. Sprechakt), kann zu rekursiver (baumartiger) Struktur führen, schwierig zu berechnen,
    - *flacher Ansatz* (engl. *flat*): eine argumentative Struktur kann in einer Dialogeinheit selbst auftreten; einfacher in sprachlicher Analyse zu verarbeiten.
  - Verwendung eines eigenen, „flachen“ argumentativen Modells:
    - Jede Konversation / Diskussion wird in *argumentative Episoden* zerlegt (Streitfragen (engl. *issue*), Vorschläge, Ausführungen, Positionen), jede Episode kann mit Sprechakten in Beziehung stehen.

<sup>63</sup> Vgl. VINCENZO ET AL. (2011)

- Anlehnung an *Meeting Description Schema* (MDS), wonach die Argumentationsstruktur aus einer Menge von Diskussionsepisoden zu einem Thema besteht; jede Themendiskussion besteht aus einer Menge Streitfragen; jeder Streitfrage ist eine Menge von Vorschlags-episoden zugeordnet (Lösungen, Alternativen, Ideen).
- Definition einer Relation *replies\_to*, die eine (Re-)Aktion einer oder mehreren vorherigen Aktionen zuordnet und eine argumentative Kettenstruktur definiert (Menge an Regeln wird im Paper vorgestellt).
- Umkehr der Regeln ergibt eine „Struktur von Erwartungen“ und gibt an, welche Aktionen eintreten müssen, damit andere Aktionen stattfinden können; Umkehr wird als Heuristik beim automatischen Aufbau einer Threadstruktur genutzt.
- *Automatic Argumentative Analysis* (A3): Domänenneutral und leicht auf andere Domänen portierbar, regelbasiertes System; Funktionen:
  - argumentative Annotationen (z. B. Aussage, Ergebnis, Grund), zum Teil abgeleitet aus RST und anderen Theorien → Mapping auf MDS,
  - Berechnung der *replies\_to* Relation des MDS und
  - Erstellung einer argumentativen Struktur des Eingabetexts.
- *GETARUNS*-System: als Pipeline organisiertes, modulares System basierend auf dem *Lexical-Functional-Grammar* (LFG)-Framework:<sup>64</sup>
  - unterstes Modul: Parsing, Strategien zur Verarbeitung von Sätzen,
  - mittleres Modul: semantische Interpretation, Aufbau des Modells der Diskussionsstruktur,
  - höchstes Modul: Schlussfolgern (engl. *reasoning*).
- Besonderheiten bei der Analyse von Konversationen mit GETARUNS:
  - Überlappungen (engl. *overlaps*): werden normalerweise als neuer Sprechakt interpretiert → problematisch zur Erkennung der Diskussionsstruktur, da sie den lokalen Kontext beeinflussen können; daher zeitliches Ordnen von Sprecherinteraktionen mittels Zeitstempel,
  - Fragmente und kurze Redebeiträge.
- Verschiedene Darstellungsarten zur Visualisierung von argumentativem Inhalt (*Conversation Graph, Aggregated View*)
- Erstellung von abstrakten Zusammenfassungen (engl. *abstractive summarisations*) aus einer generierten narrativen Beschreibung des Diskussionsprozesses

## 26. WALKER ET AL. (2011): A Framework for the Extraction and Modeling of Fact-Finding Reasoning from Legal Decisions: Lessons from the Vaccine / Injury Project Corpus<sup>65</sup>

Ziel: Beschreibung des „*Vaccine/Injury Project Corpus* (V/IP)“ und der gemachten Erfahrungen bei der Extraktion von logischen Strukturen und Modellen, Aufzeigen von (Unter-)Aufgaben des Extraktionsprozesses, Möglichkeiten zur Automatisierung

Methode: Konzeptentwicklung, Erfahrungen mit V/IP Corpus

Kerninhalte:

- Betrachtung von juristischen Entscheidungen ist mit den gleichen Probleme konfrontiert wie NLP: große Unterschiede im sprachlichen Ausdruck zwischen Parteien → Erschweren der Automatisierung.
- Der erstellte Corpus enthält Texte der Rechtsprechung in ungekürzter Form sowie Modelle zur jeweiligen Entscheidung (zusätzlich Anmerkungen zum Mapping der Elemente des Modells auf Sätze im Text).
- Beschreibung des Prozesses der Modellerstellung und Analyse der verwendeten Methodik → Ziel: Entwicklung von verlässlichen, exakten und kosteneffizienten Methoden zum Extrahieren von logischen Strukturen aus juristischen Entscheidungen.
- Dem Corpus liegen Entscheidungen im Zusammenhang mit dem *Vaccine Injury Compensation Program* zugrunde; sogenannte *special masters* fungieren als „Factfinder“ (zusammenstellen und dokumentieren der Fakten) und entscheiden nach medizinischen und wissenschaftlichen Gesichtspunkten sowie Expertenmeinung über jeweiligen Fall.
- Die vorgestellte Methodik basiert auf dem sog. *Default-Logic Paradigm* (*Default Logic* ist eines nicht-monotone Art der Logik, die es erlaubt, üblicherweise geltende Standardannahmen zu modellieren), demnach werden Regelsysteme als umgekehrte Regelbäume modelliert (Wurzel ist zu beweisende Streitfrage, Schlussfolgerung von Blättern zur Wurzel):

<sup>64</sup> Vgl. BRESNAN (2000)

<sup>65</sup> Vgl. WALKER ET AL. (2011)

- *Logische Konnektoren*: AND (alle Bedingungen wahr), OR (mindestens eine Bedingung wahr), UNLESS (Contra-Bedingung wahr → Schlussfolgerung falsch, egal wie Pro-Bedingungen auswerten) und RULEFACTORS (aufgelistete Faktoren müssen bei der Zuweisung von Wahrheitswerten berücksichtigt werden),
- Drei Wahrheitswerte: true, false, undecided
- Manueller Modellierungsprozess für eine einzelne Entscheidung:
  - Auswahl des Regelbaums, der die Regeln enthält, die den zu modellierenden Sachverhalt bestmöglich abdecken,
  - Beurteilung von Beweisen durch Extraktion derjenigen Erklärungen (engl. *assertions*), die vom *special master* als relevant beurteilt wurden; Fakten werden ebenfalls als Erklärungen modelliert; Erklärungen werden Plausibilitätswerte zugewiesen (ordinalskaliert).
  - Zuordnung der Aussagen im Regelbaum zu den gefundenen Fakten, Struktur durch „Plausibilitätskonnektoren“: MIN als Gegenstück zu AND (Schlussfolgerung erhält den geringsten Wert ihrer Bedingungen), MAX zu OR (Schlussfolgerung erhält den größten Wert ihrer Bedingungen) und REBUT zu UNLESS (Schlussfolgerung wird unplausibel, wenn eine Erklärung plausibel ist); zusätzlicher Konnektor EVIDENCE FACTORS für Sonderfälle mit Zusatzinformationen,
  - Plausibilitätswerte werden allen Konstrukten im Modell zugewiesen, sodass es der Fallevaulation des *Factfinders* entspricht.
  - Modellierung wurde in der Software *Legal Apprentice* durchgeführt; XML Ausgabe der Modelle.
- Abstraktion von den beschreibenden Modellen führt zu Schlussfolgerungsmustern (engl. *reasoning patterns*), welche die Bewertung von Beweisen generisch abbilden; Plausibilitätsschemata (engl. *plausibility schemas*) stellen Verfeinerungen dar und können Bedingungen angeben, unter denen eine Schlussfolgerung bis zu einem gewissen Fehlergrad akzeptabel ist.
- Ziel war auch die Entwicklung eines Testprotokolls zur Erstellung von logischen Modellen aus juristischen Entscheidungen, d. h. konkrete Vorgehensmodelle, die Ansprüchen nach Verlässlichkeit und Validität genügen.
- Entwicklung der Modelle erfolgt händisch durch zwei Annotatoren; der Artikel diskutiert ausführlich Problematiken, die zwischen beiden Annotatoren auftraten → Unteraufgaben im Modellierungsprozess können so identifiziert und Probleme in Zukunft vermieden werden; Probleme ergaben sich in folgenden Bereichen:
  - Identifizieren anzuwendender juristischer Regeln in der textuellen Entscheidung,
  - Identifizieren von „dynamischen Subjekten“ („der Kläger“ variiert von Fall zu Fall, daher dynamisch) und deren Beziehung zueinander,
  - Identifizieren und Repräsentieren von auf Beweisen beruhenden Erklärungen,
  - Modellieren derjenigen Erklärungen, die auf den Entdeckungen des *Factfinders* beruhen und Formulierung der Kernstruktur von Schlussfolgerungen,
  - Sortieren verbleibender Erklärungen nach „horizontaler Relevanz“ (Relevanz zu einem oder mehreren Zweigen des Regelbaums),
  - Erstellen von Ebenen von Schlussfolgerungen („vertikale Relevanz“) und Bestimmen von Plausibilitätskonnektoren zwischen den Ebenen,
  - Bestimmen der Modellierungstiefe (Detailgenauigkeit) und
  - Zuweisen von Plausibilitätswerten für Erklärungen, die auf den Erkenntnissen des *Factfinders* basieren.

## 27. ABBAS und SAWAMURA (2012): Argument Mining Based on a Structured Database and its Usage in an Intelligent Tutoring Environment<sup>66</sup>

Ziel: Vgl. ABBAS und SAWAMURA (2010)<sup>67</sup>

Methode: VGL. ABBAS und SAWAMURA (2010)

Kerninhalte: Weiterentwicklung der in ABBAS und SAWAMURA (2010) vorgestellten Inhalte, z. B. Ergänzung formaler Definitionen

<sup>66</sup> Vgl. ABBAS ET AL. (2012)

<sup>67</sup> Vgl. ABBAS ET AL. (2010)

**28. MU ET AL. (2012): The ACODEA Framework: Developing Segmentation and Classification Schemes for Fully Automatic Analysis of Online Discussions<sup>68</sup>**

Ziel: Präsentation eines Frameworks zur Entwicklung von Kodierungsschemata, welche die automatische Segmentierung und das inhalts- / kontextunabhängige Kodieren von Onlinediskussionen ermöglichen; Übertragung von Modellen zwischen verschiedenen Themenbereichen, Kernidee: semantische und syntaktische Analyse jedes einzelnen Wortes im vorliegenden Text

Methode: Entwicklung eines Frameworks, Evaluation in Studie

- Kern-  
inhalte:
- Automatische Klassifizierung von Onlinediskussionen ermöglicht Echtzeit-Analysen und Feedbackmechanismen und beschleunigt den zeitintensiven Prozess der manuellen Aufbereitung der Texte.
  - Bestehende Ansätze sind halbautomatisch, da sie eine manuelle Segmentierung benötigen.
  - Nutzung der frei verfügbaren Tools *TagHelper*<sup>69</sup> und *SIDE*<sup>70</sup> (beide entwickelt zur automatischen Inhaltsanalyse in Onlinediskussionen) → beispielhaft annotierte Daten als Trainingsdaten, anschließend automatische Extraktion von Merkmalen aus dem Text (Unterstützung von regulären Ausdrücken und *Named Entity Recognition, NER*)
  - Zur Anwendung von Kodierungsschemata (engl. *coding schemes*) muss festgelegt werden, welche Segmentgröße (einzelne Wörter, Sätze etc.) einer zu kodierenden Aktivität entsprechen → „sliding window“ (Bereich mit festgelegter Wortanzahl, der über den Text „geschoben“ wird) zur Bestimmung von möglichen Segmentgrenzen.
  - ACODEA-Ansatz verwendet Techniken aus dem Bereich Information Extraction:
    - PoS-Tagging und *Named-Entity Recognition (NER)*: Klassifizierung aller Elemente in bestimmte Kategorien (z. B. Ort)
    - NER stellt kontextunabhängige Möglichkeiten zur Verfügung, um Classifier zu trainieren.
  - Ablauf des ACODEA-Prozesses:
    - Parallele Anwendung des PoS-Tagger und des NER-Moduls,
    - Extraktion von Mustern, die generisch genug sind, um allgemeingültige Modelle zu erhalten,
    - Auf semantischer und syntaktischer Ebene werden spezielle Aufgaben und Aktivitäten klassifiziert,
    - Manuelle Segmentierung und Annotation mit Kodierungsschema erforderlich; nötig zum Trainieren eines Classifier; ML-Modell lernt von den so erhaltenen generischen Mustern.
  - Fallstudie, Training-Corpus (Teilmenge des gesamten Corpus) mit manueller Annotation, Cross-Validierung; Eingabe für SIDE als Trainingsdaten zur Klassifizierung der übrigen Daten; Beurteilung der Verlässlichkeit der Kodierung (*Cohen's Kappa*)
  - Anwendung des Frameworks:
    - Betrachtung syntaktischer Attribute: Kategorien wie Verb, Eigenschaft etc.,
    - Betrachtung semantischer Attribute: Kategorien wie Fall (Schlüsselwort aus dem Problemraum), Theorie (Schlüsselwort aus der Konzeptmenge) etc.,
    - Analyseeinheit (engl. *unit of analysis*) : Satzteil, der „syntaktisch wichtig für die Struktur ist“, Trennung zwischen Haupt- und Nebensätzen
  - Auswertung: Verbesserungen im automatischen Kodieren bei gleichzeitiger Berücksichtigung der Kontextunabhängigkeit
  - Beschränkung auf mehrdimensionale Kodierungsschemata, Einschränkung der Kontextunabhängigkeit auf spezielle Diskussionsaktivitäten; Preprocessing durch PoS-Tagging und NER hängt von den Textcharakteristiken ab. → Kodierungsschema kann nur in diesen Fällen angewendet werden.

**29. ROONEY und BROWN (2012): Applying Kernel Methods to Argumentation Mining<sup>71</sup>**

Ziel: Übertragung von Konzepten aus dem Bereich *Kernel Methods* auf Argumentation Mining

Methode: Konzeptentwicklung, Implementierung, Evaluation

- Kerninhalte:
- In Anlehnung an MOCHALES-PALAU und MOENS<sup>72</sup> definieren die Autoren ein Argument als eine Menge von Elementareinheiten oder Aussagen.

<sup>68</sup> Vgl. MU ET AL. (2012)

<sup>69</sup> Vgl. DÖNMEZ ET AL. (2005)

<sup>70</sup> Vgl. MAYFIELD ET AL. (2010)

<sup>71</sup> Vgl. ROONEY ET AL. (2012)

<sup>72</sup> Vgl. MOCHALES-PALAU ET AL. (2009)

- Fokus liegt auf der automatischen Erkennung von Prämissen und Schlussfolgerung innerhalb aller Sätze, die das Argument enthält.
- Ein *Kernel* stellt hier eine Technik dar, um die Ähnlichkeit zwischen zwei Objekten zu berechnen; dazu werden die Objekte in einen Vektorraum übertragen, wobei die Werte des Vektors Merkmale der Objekte darstellen.
- Ähnlichkeit zwischen zwei Sätzen ist definiert als die Zahl an Übereinstimmungen von Teilsequenzen von Tags (längste gemeinsame Teilfolgen); Tags sind beispielsweise die PoS einer Phrase; die maximale Lücke zwischen Elementen einer Teilfolge gibt der sog. *gap penalization factor* an.
- Die Autoren implementieren ihren Ansatz als Plug-in für *GATE* und benutzen den *AraucariaDB*-Corpus, um Prämissen und Schlussfolgerungen automatisch zu annotieren; dabei verwenden sie einen nicht näher spezifizierten SVM Classifier.
- Durch die Variation verschiedener Parameter erreichen sie eine Gesamtgenauigkeit von 65% (gegenüber 73% von MOCHALES-PALAU und MOENS).

## 2.5.4 Ergebnisübersicht

Abbildung 4 präsentiert eine Übersicht über verwendete Ansätze und diejenigen Arbeiten, die solche Ansätze genauer beschreiben.

### Legende Abbildung 4:

*Argumentationstheorien:*

TS = Toulmin Schema, WS = Walton Schema

*Argumentationsschemata:*

AS = Argumentationsschemata

*Austauschformate:*

AML = Argument Markup Language, AIF = Argument Interchange Format

*Beweisstandards:*

BS = Beweisstandards

*inhaltliche Strukturierung:*

CFG = Context-free Grammar, RST = Rhetorical Structure Theory

*ML- und NLP-Ansätze:*

MEM = Maximum Entropy Model, MNB = Multinomial Naive Bayes, NER = Named Entity Recognition, PoS = Part-of-Speech-Tagging, RC = Rocchio Classifier, SVM = Support Vector Machine, VSM = Vector Space Model, WD = Widrow-Hoff Algorithmus

*Ontologien:*

AO = Ontologien in Allgemeinen, WN = WordNet

*Strukturierungsansätze:*

GS = Graphstruktur, GG = Graph Grammar



## 2.6 Relevante Ansätze für Computer-Supported Argumentation

### 2.6.1 Vorbemerkung

Im folgenden Abschnitt werden die einzelnen, in den analysierten Beiträgen verwendeten Ansätze den verschiedenen Phasen des eingeführten Vorgehensmodells und relevanten Funktionsbereichen zugeordnet.

### 2.6.2 Aufbereiten von Texten / Argumentation Mining

#### 2.6.2.1 Schritt 1: Klassifizierung von Textabschnitten

Arbeit	verwendete Ansätze
1. Rodriguez et al. (1997)	Rocchio Classifier, VSM, Widrow-Hoff Algorithmus, WordNet
2. Feng et al. (2006)	Rocchio Classifier, VSM, Ontologien
3. Mochales Palau und Moens (2007)	eigene Argumentationsschemata, MEM, MNB, VSM
4. Sombekke (2007)	eigene Argumentationsschemata,
5. Mochales und Moens (2011)	Walton Argumentationsschema, eigene Argumentationsschemata, Beweisstandards, CFG, RST, MEM, MNB, PoS, SVM, VSM, Ontologien (OA), WordNet
6. Palotta und Delmonte (2011)	eigene Argumentationsschemata, CFG, RST
7. Mu et al. (2012)	NER, PoS

Abb. 5: Übersicht Klassifikation von Textabschnitten

#### 1. Ansätze nach RODRIGUEZ ET AL. (1997): Using WordNet to Complement Training Information in Text Categorization

- Erweiterung bestehender Konzepte zur Kategorisierung von Texten (engl. *Text Categorization (TC)*) um eine Anbindung an lexikalische Wortdatenbanken wie WordNet.
- Die zu kategorisierenden Terme werden mithilfe eines VSM kodiert und mit als Vektoren repräsentierten Kategorien abgeglichen.
- Die Klassifizierung von Termen erfolgt über zwei unterschiedliche Algorithmen, die im Anschluss jeweils beide mit der Erweiterung um WordNet kombiniert werden:
  - *Rocchio-Algorithmus*: in Bezug auf die Zuteilung von Kategorien ist zu beachten, dass der Algorithmus jeder Kategorie die gleiche Priorität zuweist.
  - *Widrow-Hoff-Algorithmus*: die Priorisierung einer Kategorie wird danach gewichtet, wie oft eine Kategorie im Trainingscorpus vorkommt (je häufiger, desto höher wird ihr Gewicht).
- Durch die Verwendung von WordNet wird der Begriff für eine Kategorie (z. B. Kategorie „Verdienst“) durch Synonyme sowie semantisch und konzeptuell verwandte Begriffe erweitert; nach festgelegten Kriterien wird zu jedem zu kategorisierenden Begriff seine semantische Nähe zur Kategorie berechnet und dementsprechend ein Anfangswert für die Gewichtung gesetzt → dieser dient als Ausgangspunkt für die Anwendung der beiden Algorithmen.

## 2. Ansätze nach Feng et al. (2006): Towards Modeling Threaded Discussions using Induced Ontology Knowledge

- Ziel ist es, Themenabweichungen und Verschiebungen des Fokus einer Diskussion zu erkennen.
- *Identifikation von Themen und Erkennen von Fokusverschiebungen*
  - Verwendung eines *Rocchio classifiers*: dieser berechnet einen Vektor pro Textkategorie (in diesem Fall pro Diskussionsthema) als gewichteter Durchschnitt von positiven und negativen Trainingsbeispielen. Nachrichten in Threads müssen hierzu als positiv / negativ bewertet werden. Falls eine solche Bewertung nicht verfügbar ist, werden die ersten  $n$  Klassifikationen als positive Beispiele angesehen.
  - Ähnlichkeit zwischen einem Diskussionsthema und einer Nachricht wird als Kosinus-Ähnlichkeit zwischen dem Vektor des Threads und dem Vektor der Nachricht bestimmt.
  - Jeder Vektor einer Nachricht enthält Elemente, die auf einer TF-IDF (*Term Frequency, Inverse Document Frequency*) Transformation basieren, durch Berechnung der lexikalischen Ähnlichkeit zwischen Nachrichtenvektor und Themenvektor kann die Relevanz einer Nachricht für das gegebene Thema bestimmt werden.
- Klassifikation von Nachrichten, um Beziehungen zwischen ihnen und dem ganzen Thread bestimmen zu können, zwei Ansätze um Nachrichten zu klassifizieren:
  - *Classify-as-a-whole*-Strategie: alle Nachrichten eines Threads werden als Dokument betrachtet und als Vektor abgebildet.
  - *Classify-by-dominance*-Strategie: verschiedene Kriterien zu Bestimmung der Dominanz (z. B. maximale Anzahl von Nennungen eines Themas in allen Nachrichten)

## 3. Ansätze nach Mochales Palau und Moens (2007): ACILA - Automatic Detection of Arguments in Legal Cases

- Erkennung von Argumenten wird als Klassifizierungsproblem betrachtet. Verschiedene Ansätze werden evaluiert und verglichen.
- Konzentration auf Abbildung von juristischer Argumentation in geschriebener Sprache, rhetorische Struktur als Hauptindikator, insbesondere corpus-spezifische lexikalische Stichwörter (engl. *lexical cues*) sind wichtige Hinweise auf die Textstruktur.
- Repräsentation eines Satzes als Vektor (Verwendung eines VSM), Dimensionen des Vektors entsprechen Merkmalen, die einfach aus dem Text zu extrahieren sind (z. B. Wortpaare, N-Gramme, Verben, Schlüsselwörter etc.)
- Vektoren dienen als Eingabe für Klassifizierungsalgorithmen (*Multinomial Naive Bayes, Maximum Entropy Model*).

## 4. Ansätze nach Sombekke (2007): Argumentation Structures in Legal Dossiers

- Vorgehen bei Analyse im Rahmen der Fallstudie:
  - Unterstützungsbeziehungen zwischen Aussagen innerhalb eines Arguments und Konfliktbeziehungen wurden identifiziert.
  - Anschließend folgte die Identifizierung impliziter Bedingungen.
  - Schließlich wurden die gefundenen Argumente mit Argumentationsschemata aus der einschlägigen Literatur abgeglichen.
- Zur Analyse der Entwicklung der argumentativen Struktur eines Rechtsfalls wurde das Dossier in sieben Phasen unterteilt.

## 5. Ansätze nach Mochales und Moens (2011) Argumentation Mining

- **Erkennung von Argumenten:** jeder Textabschnitt wird als argumentativ oder nicht-argumentativ klassifiziert → die Summe aller argumentativen Abschnitte stellt die Argumentation des Textes dar; die Unterteilung argumentativer Abschnitte in einzelne Argumente wird als Segmentierungsproblem bezeichnet.
  - *Klassifizierungsproblem:* Identifizieren von argumentativen Abschnitten mit statistischen Classifier: *Naive Bayes, Maximum Entropy Model, Support Vector Machine (SVM)* → Klassifizierungssystem wird mit generischen Merkmalen trainiert → es werden nur Informationen genutzt, die direkt aus dem Text extrahiert werden können (Wörter im Text, ihre Position etc.).
  - *Klassifizierung der identifizierten Aussagen als Prämisse oder Konklusion:* Nutzung statistischer Classifier (SVM) nach verschiedenen textuellen Merkmalen
  - *Entdeckung der argumentativen Gesamtstruktur des Textes:* Erkennung der Grenzen eines einzelnen Arguments und der Beziehungen zu umgebenden Argumenten → Entwicklung einer kontextfreien Grammatik basierend auf Erkenntnissen der *Rhetorical Structure Theory*; Regeln der Grammatik sind im Artikel enthalten

## 6. Ansätze nach Pallotta und Delmonte (2011) Automatic Argumentative Analysis for Interaction Mining

- Verwendung eines eigenen, „flachen“ argumentativen Modells (Teil eines Frameworks):
  - Jede Konversation / Diskussion wird in *argumentative Episoden* zerlegt (Streitfragen (engl. *issue*), Vorschläge, Ausführungen, Positionen).
  - Anlehnung an *Meeting Description Schema* (MDS), wonach die Argumentationsstruktur aus einer Menge von Diskussionsepisoden zu einem Thema besteht.
  - Definition einer Relation *replies\_to*, die eine (Re-)Aktion einer oder mehreren vorherigen Aktionen zuordnet und eine argumentative Kettenstruktur definiert (Menge an Regeln wird im Paper vorgestellt).
  - Umkehr der Regeln ergibt eine „Struktur von Erwartungen“ und gibt an, welche Aktionen eintreten müssen, damit andere Aktionen stattfinden können; Umkehr wird als Heuristik beim automatischen Aufbau einer Threadstruktur genutzt.
- *Automatic Argumentative Analysis* (A3): Domänenneutral und leicht auf andere Domänen portierbar, regelbasiertes System; Funktionen:
  - argumentative Annotationen (z. B. Aussage, Ergebnis, Grund), zum Teil abgeleitet aus RST und anderen Theorien → Mapping auf MDS,
  - Berechnung der *replies\_to* Relation des MDS und
  - Erstellung einer argumentativen Struktur des Eingabetexts.

## 7. Ansätze nach MU ET AL. (2012) The ACODEA Framework: Developing Segmentation and Classification Schemes for Fully Automatic Analysis of Online Discussions

- ACODEA-Ansatz verwendet Techniken aus dem Bereich Information Extraction:
  - PoS-Tagging und *Named-Entity Recognition* (NER): Klassifizierung aller Elemente in bestimmte Kategorien (z. B. Ort)
  - NER stellt kontextunabhängige Möglichkeiten zur Verfügung, um Classifier zu trainieren.
- Ablauf des ACODEA-Prozesses:
  - Parallele Anwendung des PoS-Tagger und des NER-Moduls,
  - Extraktion von Mustern, die generisch genug sind, um allgemeingültige Modelle zu erhalten,
  - Auf semantischer und syntaktischer Ebene werden spezielle Aufgaben und Aktivitäten klassifiziert,
  - Manuelle Segmentierung und Annotation mit Kodierungsschema erforderlich; nötig zum Trainieren eines Classifier; ML-Modell lernt von den so erhaltenen generischen Mustern.
- Anwendung des Frameworks:
  - Betrachtung syntaktischer Attribute: Kategorien wie Verb, Eigenschaft etc.,
  - Betrachtung semantischer Attribute: Kategorien wie Fall (Schlüsselwort aus dem Problemraum), Theorie (Schlüsselwort aus der Konzeptmenge) etc.,
  - Analyseeinheit (engl. *unit of analysis*): Satzteil, der „syntaktisch wichtig für die Struktur ist“, Trennung zwischen Haupt- und Nebensätzen

### 2.6.2.2 Schritt 2: Erkennen von Grenzen in argumentativen Abschnitten

Arbeit	verwendete Ansätze
1. Mochales und Moens (2011)	Walton Argumentationsschema, eigene Argumentationsschemata, Beweisstandards, CFG, RST, MEM, MNB, PoS, SVM, VSM, Ontologien (OA), WordNet

Abb. 6: Übersicht Erkennen von Grenzen in argumentativen Abschnitten

### 1. Ansätze nach MOCHALES und MOENS (2011): Argumentation Mining

*Grenzen von argumentativen Abschnitten*: 1.) Orientierung an der Struktur der zu analysierenden Texte (Abschnitte, Gliederung), 2.) semantische Merkmale zur Erkennung einzelner Argumente (Ontologie- / Corpus-basiert, semantischer Abstand zwischen Sätzen, Nutzung von Ontologien wie *WordNet*). Problematisch: Argumente können sich durchaus über mehrere Abschnitte erstrecken, in einem Abschnitt können mehrere Argumente enthalten sein.

### 2.6.2.3 Schritt 3: Klassifizierung identifizierter Aussagen

Arbeit	verwendete Ansätze
1. Rodriguez et al. (1997)	Rocchio Classifier, VSM, Widrow-Hoff Algorithmus, WordNet
2. Mochales Palau und Moens (2007)	Ontologien (OA)
3. Ravi und Kim (2007)	SVM
4. Mochales und Moens (2008)	eigene Argumentationsschemata, CFG
5. Mochales und Moens (2011)	Walton Argumentationsschema, eigene Argumentationsschemata, Beweisstandards, CFG, RST, MEM, MNB, POS, SVM, VSM, Ontologien (OA), WordNet
6. Rooney und Brown (2012)	eigene Argumentationsschemata, VSM

Abb. 7: Übersicht Klassifikation identifizierter Aussagen

#### 1. Ansätze nach RODRIGUEZ ET AL. (1997): Using WordNet to Complement Training Information in Text Categorization

- Die Klassifizierung von Termen erfolgt über zwei unterschiedliche Algorithmen, die im Anschluss jeweils beide mit der Erweiterung um WordNet kombiniert werden; beide Algorithmen und die jeweilige Berechnung von gewichteten Vektoren werden im Papier beschrieben:
  - Rocchio-Algorithmus*: in Bezug auf die Zuteilung von Kategorien ist zu beachten, dass der Algorithmus jeder Kategorie die gleiche Priorität zuweist.
  - Widrow-Hoff-Algorithmus*: die Priorisierung einer Kategorie wird danach gewichtet, wie oft eine Kategorie im Trainingscorpus vorkommt (je häufiger, desto höher wird ihr Gewicht).
- Durch die Verwendung von WordNet wird der Begriff für eine Kategorie (z. B. Kategorie „Verdienst“) durch Synonyme sowie semantisch und konzeptuell verwandte Begriff erweitert; nach festgelegten Kriterien wird zu jedem zu kategorisierenden Begriff seine semantische Nähe zur Kategorie berechnet und dementsprechend ein Anfangswert für die Gewichtung gesetzt → dieser dient als Ausgangspunkt für die Anwendung der beiden Algorithmen.

#### 2. Ansätze nach Mochales-Palau et al. (2007): ACILA - Automatic Detection of Arguments in Legal Cases

- Automatische Klassifikation eines Arguments nach bestimmten Typen (Gegenargument, Gegenbeweis) wird angestrebt.
- Automatische Qualifizierung eines Arguments als relevant für die Entscheidung und Beschreibung durch Elemente einer zugeordneten Ontologie → ermöglicht das Schlussfolgern in einem fallbasierten Argumentationssystem.

#### 3. Ansätze nach Ravi und Kim (2007): Profiling Student Interactions in Threaded Discussions with Speech Act Classifiers

- Entwicklung zweier Classifier: *Question Classifier* (QC, Identifikation einer Frage) und *Answer Classifier* (AC, Identifikation von Antworten).
- Thread profiler* zur Klassifikation von Diskussionen → Erkennung von Diskussionen mit unbeantworteten Fragen
- Texte aus Foren sind häufig *unzusammenhängend* (in Bezug auf grammatikalische Struktur) und „rauschend“ (engl. *noisy*, in Bezug auf die Verwendung individueller Wörter pro Nutzer) → verschiedene Maßnahmen, um Rauschen zu reduzieren, z. B.:
  - Stemming*: entfernen der Pluralendungen zur Vereinheitlichung von Begriffen auf Wortstämme,
  - Ersetzung von Personalpronomen durch *categ\_person* und technischer Begriffe durch *tech\_term* und
  - Austausch von Smileys durch einheitliche Worte

- Auswahl der Merkmale (engl. *features*) für Verwendung in SVM: *N-grams* (Folgen von *n* Wörtern); *Information Gain Theory* zur Auswahl wichtiger Merkmale durch Berechnung einer Maßzahl dafür, wie stark das Vorhandensein / Nicht-Vorhandensein eines Merkmals die Klassifikation beeinflussen würde → für jedes Merkmal wird der *gain*-Wert für den QC und AC berechnet → Sortierung nach diesem Wert und Auswahl der ersten 200 Merkmale

#### 4. Ansätze nach Mochales / Moens (2008): Study on the Structure of Argumentation in Case Law

- Fokus der Autoren liegt darauf, die Funktion einzelner Sätze im argumentativen Prozess zu erkennen.
- Die vorliegenden Dokumente folgen zwar keiner streng formalen Struktur, wie sie etwa *MetaLex* bietet, besitzen jedoch fast immer die gleiche inhaltliche Gliederung in 5 Abschnitte.
- Die vorgestellte argumentative Analyse basiert auf verschiedenen Argumentationsschemata, die im Gegensatz bspw. zu Formalismen nach TOULMIN eine Entscheidungsfindung auch bei Unsicherheit und Unvollständigkeit zulassen sollen (engl. *presumptive reasoning*).
- Wiederkehrende sprachliche Muster (rhetorische Marker) in den Dokumenten wurden identifiziert. Abgrenzung von generellen Ausdrücken sprachlicher Argumentation gegenüber Ausdrücken die typisch für juristische Argumentation sind (Worttabelle); Unterscheidung zweier Ebenen:
  - eine Schlussfolgerung, die zu einer Entscheidung führt, ist abgeleitet von den unterstützenden Prämissen,
  - eine Schlussfolgerung, die die Prämissen unterstützt, die wiederum zur ersten Schlussfolgerung beitragen.
- Entwicklung einer kontextfreien Grammatik zur Bottom-up-Erstellung einer Argumentstruktur; Grammatik arbeitet auf Satzebene

#### 5. Ansätze nach Mochales und Moens (2011): Argumentation Mining

*Klassifizierung der identifizierten Aussagen als Prämisse oder Folgerung:* Nutzung statistischer Classifier (*Support Vector Machine*, SVM) nach verschiedenen textuellen Merkmalen (siehe Tabelle im Artikel, z.B. absolute Position einer Aussage im Gesamt-Textcorpus, erkannte Referenz auf einen juristischen Artikel etc.).

#### 6. Ansätze nach Rooney und Brown (2012): Applying Kernel Methods to Argumentation Mining

- Fokus liegt auf der automatischen Erkennung von Prämissen und Schlussfolgerung innerhalb aller Sätze, die das Argument enthält.
- Ähnlichkeit zwischen zwei Sätzen ist definiert als die Zahl an Übereinstimmungen von Teilsequenzen von Tags (längste gemeinsame Teilfolgen); Tags sind beispielsweise die PoS einer Phrase; die maximale Lücke zwischen Elementen einer Teilfolge gibt der sog. *gap penalization factor* an.

### 2.6.2.4 Schritt 4: Erkennen von Argumentationsstrukturen

Arbeit	verwendete Ansätze
1. Pinkwart et al. (2006)	PoS, Graph Grammar
2. Pinkwart et al. (2008)	Graphstrukturen, Graph Grammar
3. Gkotsis und Karacapilidis (2010)	Graphstrukturen
4. Hogenboom (2011)	RST, PoS, Ontologien (OA)
5. Mochales und Moens (2011)	Walton Argumentationsschema, eigene Argumentationsschemata, Beweisstandards, CFG, RST, MEM, MNB, PoS, SVM, VSM, Ontologien (OA), WordNet

Abb. 8: Übersicht Erkennen von Argumentationsstrukturen

### 1. Ansätze nach Pinkwart et al. (2006): Toward Legal Argument Instruction with Graph Grammars and Collaborative Filtering Techniques

- Identifikation *struktureller Schwachstellen*: isolierte Elemente, fehlende Tests etc.
  - Erkennung durch logische Formalismen, z. B. *Graph Grammar*,
  - Sehr gut adaptierbar auf Graphstrukturen von Argumentationen,
  - Deklarativer Charakter: Regeln können leicht als Parameter eines generischen Parsing-Algorithmus deklariert werden.

### 2. Ansätze nach Pinkwart et al. (2008): Graph Grammars - An ITS Technology for Diagram Representations

- Die Autoren verwenden das *LARGO*-System und erweitern es um eine Graph-Grammatik, welche die Grundlage für die Diagrammodellierung in LARGO darstellt; die Grammatik dient zur Formalisierung von Argumentationsdiagrammen und zur Überprüfung modellierter Diagramme auf Konsistenz und potentielle Fehler
- Die Grammatik enthält zwei Arten von Produktionsregeln:
  - *Generierungsregeln* (engl. *generation rules*): kommen bei der Erstellung von Diagrammen zur Anwendung; es kann vom Nutzer nur das modelliert werden, was explizit als Produktionsregel vorhanden ist.
  - *Feedbackregeln* (engl. *feedback rules*): werden manuell auf Wunsch des Nutzers aufgerufen und auf alle modellierten Diagrammelemente angewendet; Erkennung von strukturellen Schwächen im Graph (z. B. unwahrscheinliche Verbindungen zweier Elemente).
- Priorisierung und Auswahl der identifizierten Schwachstellen und Präsentation des Feedbacks stellt eine große Herausforderung dar.

### 3. Ansätze nach Gkotsis und Karacapilidis (2010): On the Exploration and Exploitation of Structural Similarities in Argumentative Discourses

- Unterscheidung zwischen Mikrostruktur (Attribute eines Arguments) und Makrostruktur (zugrundeliegendes Argumentationsmodell)
- Graphbasierter Ansatz zur Beschreibung von argumentativen Dialogen; ermöglicht die Quantifizierung von strukturellen Ähnlichkeiten; Betrachtungen auf Mikro- / Makrolevel möglich.
- Grundlage: verbundener, gewichteter ungerichteter Graph; *Streitfrage* (engl. *issue*) ist Wurzel des Graphen; *Alternativen* werden durch neutrale Kantentypen mit Gewicht 0 an die Wurzel angefügt; *Argumente* werden mit anderen Argumenten oder Alternativen verbunden, unterstützende Argumente haben das Gewicht 1, ablehnende Argumente Gewicht -1:
  - *Ähnlichkeit* zwischen zwei Teilgraphen  $Sim(x,y)$  ist bestimmt durch die Gegenzahl der normalisierten Euklidischen Differenz.
  - *Vertex Refinement Query*  $H_i(x)$ : rekursive Methode, um strukturelle Ähnlichkeiten in benachbarten Knoten (der Tiefe  $i$ ) zu identifizieren,
  - Berechnung der *Sequence Similarity* zwischen zwei Knotenmengen: maximale Summe von Ähnlichkeiten zwischen Paaren von Knoten aus jeweils einer Menge,
  - *Vertex Equivalence*: Ähnlichkeit zweier verfeinerter Knoten  $Sim(H_i(x), H_i(y))$  liegt über einem Grenzwert *threshold*; *Vertex Identity*: Ähnlichkeit liegt für alle Werte  $t$  aus  $[0,i]$  über dem Grenzwert *threshold*,

### 4. Ansätze nach Hogenboom (2011): Mining Economic Sentiment Using Argumentation Structures

- Entwicklung einer *Information Extraction Pipeline* zur schrittweisen Abstraktion von Rohertexten zu formalisierten Textstücken, Wiederverwendung von Komponenten der GATE-Standard-Pipeline:
  - *Document Reset*: Entfernen von Artefakten (z. B. Tags),
  - *English Tokenizer*: Aufteilen von Text in Tokens (z. B. Worte),
  - *Sentence Splitter*: Aufteilung von Text in Sätze,
  - *PoS-Tagger*: Bestimmung der Wortfunktionen,
  - *Argumentation Gazetteer*: Erkennung von Markern für Argumentation durch eine spezifische Ontologie (enthält Marker und ihre Beziehungen zu argumentativen Textelemente), Ansatz zur Strukturierung von Argumentation in wirtschaftlichen Texten: Ausgangspunkt sind die Argumentationstheorien *Rhetorical Structure Theory* (RST) und *Textual Entailment* (TE),
  - *Argumentation Parser*: Identifikation von Textsegmenten und Bestimmung ihrer Rolle in der Argumentationsstruktur gemäß der Argumentationsontologie,
  - *Sentiment Analyzer*: identifiziert die Stimmung eines Textsegments und setzt es mit der verknüpften Argumentationsstruktur in Verbindung; abhängig von der Rolle eines Textsegments in der Argumentationsstruktur erhält es ein Gewicht für den Beitrag zur Gesamtstimmung des Texts.
- Ergebnis dieses Prozesses ist eine „on the fly“-Ontologie, die Wissen zur aktuellen wirtschaftlichen Stimmung im Text repräsentiert.

### 5. Ansätze nach Mochales und Moens (2011): Argumentation Mining

*Entdeckung der argumentativen Gesamtstruktur des Textes*: Erkennung der Grenzen eines einzelnen Arguments und der Beziehungen zu umgebenden Argumenten → Entwicklung einer kontextfreien Grammatik basierend auf Erkenntnissen der *Rhetorical Structure Theory*; Regeln der Grammatik sind im Artikel enthalten; generischer Algorithmus zum Parsing des Textes unter Verwendung der erstellten Grammatik, einmaliger Aufwand bei der Erstellung, hohe Wiederverwendbarkeit und leichte Erweiterbarkeit

#### 2.6.2.5 Schritt 5: Anreichern von Metainformationen

Arbeit	verwendete Ansätze
1. Mochales Palau und Moens (2007)	---
2. Reed et al. (2008)	Walton Argumentationsschemata, eigene Argumentationsschemata, AIF, AML
3. Wyner et al. (2010)	eigene Argumentationsschemata, AML, CFG, Ontologien (OA)
4. Rooney und Brown (2012)	eigene Argumentationsschemata, VSM

Abb. 9: Übersicht Anreichern von Metainformationen

#### 1. Ansätze nach Mochales Palau und Moens (2007): ACILA - Automatic Detection of Arguments in Legal Cases

- Manuelle Annotation eines Testkorpus nach Schemata des *Araucaria*-Systems und Vergleich der Textklassifizierung mit automatisierten Durchgängen unter Verwendung verschiedener Merkmale

#### 2. Ansätze nach Reed et al. (2008): Language Resources for Studying Argument

- Manuelle Annotation von Texten mittels *Araucaria*, Verwendung der *Argument Markup Language* (AML) zur Darstellung von Beziehungen zwischen Textelementen und Argumentationsstrukturen, 2004 Erweiterung um Ontologie-Elemente auf Basis von Argumentationsschemata nach WALTON.

#### 3. Ansätze nach Wyner et al. (2010): Approaches to Text Mining Arguments from Legal Cases

- Manuelle Markierung von Argumenten mittels *Argument Markup Language* (AML) in *AraucariaDB* durch zwei Experten, Abbilden von Schemata und unterschiedlichen Thesen, Beziehungen zwischen ihnen, unterstützende (Meta-)Informationen (Auto, Datum der Analyse).

#### 4. Ansätze nach Rooney und Brown (2012): Applying Kernel Methods to Argumentation Mining

- Die Autoren implementieren ihren Ansatz als Plug-in für *GATE* und benutzen den *AraucariaDB*-Corpus, um Prämissen und Schlussfolgerungen automatisch zu annotieren; dabei verwenden sie einen nicht näher spezifizierten SVM classifier.

### 2.6.3 Information Retrieval

Arbeit	verwendete Ansätze
1. Karacapilidis und Papadias (2001)	Beweisstandards, Graphstrukturen
2. Abbas und Sawamura (2008)	Walton Argumentationsschema, eigene Argumentationsschemata, AIF, Graphstrukturen
3. Kim et al. (2008)	VSM
4. Abbas und Sawamura (2010)	Walton Argumentationsschema, eigene Argumentationsschemata, AIF, Graphstrukturen
5. Wyner et al. (2010)	eigene Argumentationsschemata, AML, CFG, Ontologien (OA)

Abb. 10: Übersicht Information Retrieval

#### 1. Ansätze nach Karacapilidis und Papadias (2001): Computer Supported Argumentation and Collaborative Decision Making - The Hermes System

*IR Tool des Hermes Systems*: Information-Retrieval-Tool ermöglicht die Abfrage externer Datenbanken, um Anfragen betreffend der laufenden Diskussion zu stellen.

#### 2. Ansätze nach Abbas und Sawamura (2008): A First Step Towards Argument Mining and its Use in Arguing Agents and ITS

- Zur Durchführung von Argument Mining entwickeln die Autoren ein dreiteiliges Framework:
  - *Parser* Modul: erhält eine Anfrage vom Nutzer als String; dieser String wird in Tokens zerlegt und von unnötigen Wörtern (Artikel, Pronomen) befreit. Der Sinn dessen liegt vor allem in der Reduktion der Anfrage, um unnötige Iterationen des nächsten Moduls zu vermeiden.
  - *Classifier Agent*: führt die Anfrage an die zugrundeliegende Datenbank durch und klassifiziert die gefundenen Dokumente nach Wunsch des Nutzers (z. B. nach Priorität, Polarität, Schema); dabei werden iterativ alle Kombinationen der vom Parser übermittelten Wörter gesucht.
  - *ITS*: der Nutzer erhält die Möglichkeit, die gefundenen Daten zu analysieren und basierend auf dem gewählten Schema Argumente auszuwählen.

#### 3. Ansätze nach Kim et al. (2008): Scaffolding On-line Discussions with past Discussions - An Analysis and Pilot Study of Pedabot

- Prozess der Analyse und des Retrieval als *Retrieval Pipeline*:
  - *TextTiling* zur Unterteilung der Nachrichten in semantisch verwandte Kategorien,
  - Texte aus den Foren sind *unzusammenhängend* (in Bezug auf grammatikalische Struktur) und „rauschend“ (engl. *noisy*, in Bezug auf die Verwendung individueller Wörter pro Nutzer) → automatische Erstellung einer Datenbank von technischen, im Forum verwendeten Wörtern aus Inhaltsverzeichnissen des dem Forum zugrundeliegenden Lehrbuchs,
  - Darstellung von Nachrichten als Vektoren mit Boolean-Werten für Vorhandensein der einzelnen technischen Wörter,
  - *Topic Profiler* erzeugt einen Themenvektor aus thematischen Abschnitten des Lehrbuchs; Berechnung eines *term weight vectors* für jede Themenkategorie (*TF-IDF* Verfahren),
  - *Term weights* werden zur Berechnung der Ähnlichkeit zwischen Nachrichten im Corpus und neuer Nachricht verwendet,
  - *Latent Semantic Analysis* (LSA) zur Bestimmung der Beziehung zwischen Begriffen (engl. *terms*),
  - Abgleich von neuer Nachricht und Nachrichten im Corpus über Bestimmung der Kosinusähnlichkeit der Nachrichtenvektoren

#### 4. Ansätze nach Abbas und Sawamura (2010): Argument Mining from RADB and its Usage in Arguing Agents and Intelligent Tutoring Systems

- *AprioriTid Mining Classification*: Identifikation des relevantesten Arguments bzgl. der Suchanfrage (Relevanz basierend auf Übereinstimmung mit dem Suchbegriff); Berechnung von möglichen Kombinationen der Suchbegriffe zur Erstellung einer Ordnung nach Unterstützungen,
- *Tree Substructure Mining*: Breitensuche auf der Baumstruktur des RADB-Repository, Extraktion des Teilbaumes mit den meisten Knoten mit Bezug zum Suchbegriff,
- *Rule Extraction Mining*: Erkennen von Substrukturen (Teilbäumen) im RADB-Repository durch Reduktion der Suchanfrage auf Schlüsselbegriffe; Ausgabe einer „Regel“

#### 5. Ansätze nach Wyner et al. (2010): Approaches to Text Mining Arguments from Legal Cases

- Vorstellung verschiedener Ansätze zur Analyse einer Menge von juristischen Rechtsfällen
  - 1) *Kontextfreie Grammatik*: Entwicklung in Anlehnung an MOCHALES und MOENS (2008) zur Identifizierung von Argumentationsstrukturen.
  - 2) *Case-based Reasoning* (CBR): spezieller Ansatz im Common Law zur Identifikation von Präzedenzfällen; Präzedenzfälle werden anhand von „Faktoren“ identifiziert, wobei ein Faktor ein textuelles Muster ist, welches die Entscheidung eines Rechtsfalles beeinflusst; Faktoren werden mittels IE durch semantische Ähnlichkeiten gefunden, unabhängig von der textuellen Form.  
Zum Durchsuchen der Datenbasis an Rechtsfällen wurde *Linguamatics* verwendet, das u. a. folgende Funktionen bietet:
    - Suche mit regulären Ausdrücken,
    - Auflisten alternativer Wörter,
    - Suchen innerhalb eines syntaktischen Rahmens (Satz, Paragraph etc.) und
    - Integration von spezifischen Ontologien.

### 2.6.4 Inhaltliche Analyse

Arbeit	verwendete Ansätze
1. Feng et al. (2006)	Rocchio Classifier, VSM, Ontologien (OA)
2. Pinkwart et al. (2006)	PoS, Graph Grammar

Abb. 11: Übersicht Inhaltliche Analyse

#### 1. Ansätze nach Feng et al. (2006): Towards Modeling Threaded Discussions using Induced Ontology Knowledge

- Ableitung einer Ontologie:
  - Automatisiertes Ableiten einer Ontologie aus einem geordneten Text (engl. *canonical text*), im vorliegenden Fall aus der hierarchischen Inhaltstabelle und dem Index eines Lehrbuches, dessen Inhalte Gegenstand der Diskussionen im Forum waren.
  - Jeder Begriff der Ontologie wird mit einer Häufigkeitsangabe versehen.
  - Identifizierte Themen werden in Themenvektoren übertragen (Elemente des Vektors sind Begriffe aus den Unterthemen) und können dann mit Nachrichtenvektoren verglichen werden.

#### 2. Ansätze nach Pinkwart et al. (2006): Toward Legal Argument Instruction with Graph Grammars and Collaborative Filtering Techniques

- Identifikation *kontextueller Schwachstellen*: fehlende Relationen zwischen Graph und Abschrift,
  - Erkennung durch Graph Grammar,
  - Definition von nicht-relevanten Textstellen durch manuelles Markieren.
- Identifikation *inhaltlicher Schwachstellen*: „collaborative filtering“ durch Feedback anderer Nutzer des Systems, Errechnung einer Wertung (Score) als Indikator für Qualität einer Lösung

## 2.6.5 Vorschlägen von Argumentation

Arbeit	verwendete Ansätze
1. Karacapilidis und Papadias (2001)	Beweisstandards, Graphstrukturen

Abb. 12: Übersicht Vorschlägen von Argumentation

### 1. Ansätze nach Karacapilidis und Papadias (2001): Computer Supported Argumentation and Collaborative Decision Making - The Hermes System

*Unterstützende Funktion bei der Erstellung neuer Argumente:* geplant wurde die Entwicklung eines Argument Builder Tools, das auf Basis der definierten Modellierungsstruktur die Konstruktion von Argumenten durch den Benutzer unterstützt; regelbasierte Unterstützung des Nutzers zur Lösung von Problemen wie unzureichender Quellenbelege eines Arguments, d. h. Sicherstellung syntaktischer Korrektheit der Argumente nach zugrundeliegender Modelllogik.

## 2.6.6 Abbildung der Struktur von Argumentation

Arbeit	verwendete Ansätze
1. Gordon und Karacapilidis (1997)	Beweisstandards, Graphstrukturen
2. Stranieri und Zeleznikow (1999)	Toulmin Argumentationsschemata,
3. Chklovski et al. (2005)	Graphstrukturen
4. Gordon (2007)	Walton-Schemata, eigene Argumentationsschemata, Beweisstandards, Graphstrukturen
5. Abbas und Sawamura (2008)	Walton Argumentationsschemata, eigene Argumentationsschemata, AIF, Graphstrukturen
6. Walker et al. (2011)	Default Logic

Abb. 13: Übersicht Strukturelle Abbildung von Argumentation

### 1. Ansätze nach Gordon und Karacapilidis (1997): The Zeno Argumentation Framework

- Systementwurf für ein Mediation System mittels der *Object Modeling Technique* (OMT); drei Diagrammarten:
  - *Funktionales Modell:* stellen den Datenfluss im Diskussionsprozess dar; enthält ein generelles Argumentationsmodell.
  - *Objektmodell:* Objekte, Beziehungen und Vererbungshierarchie; Objekten können Modelle zugeordnet werden, um verschiedene Sichten / Interpretationen auf bspw. Nachrichten abzubilden: dialektische Graphen zur Darstellung des Standes einer Diskussion zu einem spezifischen Zeitpunkt, Fokus auf der Rolle und Funktion der Redebeiträge.
  - *Dynamisches Modell:* gibt an, welche Art von Redebeiträgen (engl. *Speech Acts*) in welchem Status der Diskussion möglich sind.

## 2. Ansätze nach Stranieri und Zeleznikow (1999): A Survey of Argumentation Structures for Intelligent Decision Support

- *Toulmin Arguments Structures* (TAS) zur Beschreibung von Argumentstrukturen unabhängig vom Inhalt und zur semantischen Erfassung von Schlussfolgerungen
- Erweiterung der TAS durch STRANIERI UND ZELEZNIKOW: Unterscheidung zweier Typen von Schlussregeln (engl. *warrant*) zur Differenzierung ihrer Funktion: 1.) Betonen der Relevanz eines Faktums und 2.) Beeinflussung der Folgerung. Dadurch entsteht die Möglichkeit, explizite Gründe für die Relevanz eines Datenitems anzugeben; Explikation der Folgerungsmethode (Aussagen über Qualität einer Folgerung möglich); explizite Angabe von Gründen für eine Folgerungsmethode möglich → insgesamt Erleichterung der Entwicklung halb-automatischer Folgerungssysteme.

## 3. Ansätze nach Chklovski et al. (2005): User Interfaces with Semi-Formal Representations: a Study of Designing Argumentation Structures

- Eine „semiformale“ Darstellung ermöglicht Freitextaussagen mit maschinenlesbarer Struktur (Bereich „Computer-Assisted Argumentation“), semiformale Darstellung kann auf Basis von NLP- und ML-Ansätzen automatisiert vom System weiterstrukturiert und weiterverarbeitet werden.
- Entwicklung von drei Konzepten zur semi-formalen Repräsentation:
  1. *Rich Trellis*: Verknüpfung von Freitexten durch argumentative Konnektoren (z. B. „wird gestützt durch“); Clustering-Algorithmen und NLP-Techniken (z. B. PoS-Tagging) können angewendet werden. Abgleich von Argumenten erscheint schwierig: Vielzahl an Konnektoren führt leicht zu Inkonsistenzen, gleiches Argument kann unterschiedlich strukturiert werden, Kontrollfluss kann durch AND- / OR-Konnektoren gestört werden.
  2. *Tree Trellis*: im Vergleich zu *Rich Trellis* vereinfachte Struktur, es existieren lediglich zwei Konnektoren (*pro* und *con*).
  3. *Table Trellis*: Tabellen / Matrix-Darstellung; klar unterscheidbare Merkmale als Spaltennamen, zu vergleichende Alternativen als Zeilen, Zelleneinträge entsprechen Werten

## 4. Ansätze nach Gordon (2007): The Carneades Model of Argument and Burden of Proof

- Argumentgraphen zur Definition der logischen Struktur:
  - Aussagen (engl. *statements*): Syntax einer Aussage ist unwichtig, System muss nur entscheiden können, ob zwei Aussagen syntaktisch gleich sind und muss das logische Komplement bestimmen können.
  - Prämissen: Unterscheidung in drei Typen
  - Argumente: Tupel aus Aussage, pro- / con-Attribut und Prämisse.
- Argumentgraphen haben zwei Arten von Knoten: Aussageknoten und Argumentknoten. Die Kanten stellen entweder Prämissen oder Schlussfolgerungen dar. Argumentgraphen stellen eine Art von Beweisbäumen dar: Argumente sind dann akzeptabel, wenn ihr Argumentgraph die Aussage beweist.

## 5. Ansätze nach Abbas und Sawamura (2008): A First Step Towards Argument Mining and its Use in Arguing Agents and ITS

- Verwendung von relationalen Datenbanken zur Speicherung von Argumentationsstruktur
- Argumente können in Form von Knoten und gerichteten Kanten repräsentiert werden. Wenn zyklische Graphen ausgeschlossen werden, ergibt sich hieraus eine gerichtete Baumstruktur, die in eine relationale Datenbank übertragen werden kann.
- Jedes Argumentationsschema nach WALTON kann in einer baumartigen Struktur dargestellt werden → der Ansatz der Autoren sieht nun die Aufteilung der Informationen eines Schemas auf drei Tabellen vor:
  - *Scheme\_TBL*: enthält ID und Namen für jedes Schema,
  - *Scheme\_Struct\_TBL*: enthält detaillierte Informationen zu jedem Schema, bspw. den Typ (Prämisse, Schlussfolgerung etc.),
  - *Data\_TBL*: enthält die Nutzerdaten, die bei einer Analyse anfallen; eine Spalte nimmt z. B. den Text auf, der zum angegebenen Argumentationsschema passt. Die Typ-Spalte gibt eine Unterstützung oder Ablehnung an. Die Spalte *Child\_Of* ermöglicht den Aufbau der Baumstruktur.

## 6. Ansätze nach Walker et al. (2011): A Framework for the Extraction and Modeling of Fact-Finding Reasoning from Legal Decisions: Lessons from the Vaccine / Injury Project Corpus

- Die vorgestellte Methodik basiert auf dem sog. *Default-Logic Paradigm* (*Default Logic* ist eines nicht-monotone Art der Logik, die es erlaubt, üblicherweise geltende Standardannahmen zu modellieren), demnach werden Regelsysteme als umgekehrte Regelbäume modelliert (Wurzel ist zu beweisende Streitfrage, Schlussfolgerung von Blättern zur Wurzel):

- *Logische Konnektoren*: AND (alle Bedingungen wahr), OR (mindestens eine Bedingung wahr), UNLESS (Contra-Bedingung wahr → Schlussfolgerung falsch, egal wie Pro-Bedingungen auswerten) und RULE-FACTORS (Faktoren müssen bei der Zuweisung von Wahrheitswerten berücksichtigt werden),
- Drei Wahrheitswerte: true, false, undecided
- Manueller Modellierungsprozess für eine einzelne Entscheidung:
  - Auswahl des Regelbaums, der die Regeln enthält, die den zu modellierenden Sachverhalt bestmöglich abdecken,
  - Beurteilung von Beweisen durch Extraktion derjenigen Erklärungen (engl. *assertions*), die vom *special master* als relevant beurteilt wurden; Fakten werden ebenfalls als Erklärungen modelliert; Erklärungen werden Plausibilitätswerte zugewiesen (ordinalskaliert).
  - Zuordnung der Aussagen im Regelbaum zu den gefundenen Fakten, Struktur durch „Plausibilitätskonnektoren“: MIN als Gegenstück zu AND (Schlussfolgerung erhält den geringsten Wert ihrer Bedingungen), MAX zu OR (Schlussfolgerung erhält den größten Wert ihrer Bedingungen) und REBUT zu UNLESS (Schlussfolgerung wird unplausibel, wenn eine Erklärung plausibel ist); zusätzlicher Konnektor EVIDENCE FACTORS für Sonderfälle mit Zusatzinformationen,
  - Plausibilitätswerte werden allen Konstrukten im Modell zugewiesen, sodass es der Fallevaluation des *Factfinders* entspricht.
  - Modellierung wurde in der Software *Legal Apprentice* durchgeführt; XML Ausgabe der Modelle
- Abstraktion von den beschreibenden Modellen führt zu Schlussfolgerungsmustern (engl. *reasoning patterns*), welche die Bewertung von Beweisen generisch abbilden; Plausibilitätsschemata (engl. *plausibility schemas*) stellen Verfeinerungen dar und können Bedingungen angeben, unter denen eine Schlussfolgerung bis zu einem gewissen Fehlergrad akzeptabel ist.

## 2.7 Fazit zur Literaturanalyse

Im Rahmen der vorliegenden Literaturanalyse wurden zunächst unter besonderer Berücksichtigung zentraler Begriffe ein Bezugsrahmen für das durchgeführte strukturierte Review entwickelt und Details zur Review-Methode erläutert. Terminologische Grundlagen der Argumentationstheorie wurden ebenso erläutert. Im Anschluss wurden zentrale Forschungsarbeiten in diesen Bereichen hinsichtlich relevanter Inhalte exzerpiert und resümiert. Eine Synthese der dadurch erarbeiteten Erkenntnisse bietet eine zentrale Übersicht über verwendete Techniken für die rechnergestützte Verarbeitung von Argumenten unter besonderer Berücksichtigung des Argumentation Mining. Anschließend wurden relevante Textpassagen und Erläuterungen über verwendete Ansätze und Techniken den einzelnen Verfahrensschritten des zugrunde gelegten Vorgehensmodells zugeordnet und dargestellt. Eine Reihe von Ansätzen des *Natural Language Processing* erweisen sich im Kontext des Projektes ARGUMENTUM als sehr fruchtbringend einsetzbar. Da diese Ansätze allerdings bisher für sehr unterschiedliche Zwecke eingesetzt werden, ist ein Vergleich der Leistungsfähigkeit der Ansätze allein auf Basis der Literatur und ohne eine geeignete Vergleichsgrundlage nicht möglich. Es zeigte sich weiterhin, dass existierende Ansätze auf Basis unterschiedlicher Argumentationsschemata arbeiten, wobei eine Vielzahl von Arbeiten auf die Argumentationsschemata von TOULMIN und WALTON zurückgreifen.

### 3 Übersicht zu Konzepten und Prototyp

#### 3.1 Fachkonzept

Entwickelt wurde das zugrunde gelegte Fachkonzept unter Berücksichtigung des aktuellen Forschungsstands im Bereich *Computer-supported Argumentation* und *Argumentation Mining*, der typischen Struktur des BVerfG-Entscheidungskorpus sowie auf Basis einer gemeinsamen Anforderungsanalyse der Konsortialpartner unter Berücksichtigung verschiedener Nutzungsszenarien. Eine detailliertere Beschreibung des Fachkonzeptes kann in HOUY ET AL. (2015) nachvollzogen werden.<sup>73</sup> Folgende Abbildung zeigt eine Übersicht über das ARGUMENTUM-Phasenkonzept, welches als Rahmenwerk für die Implementierung des Software-Werkzeuges dient.<sup>74</sup>

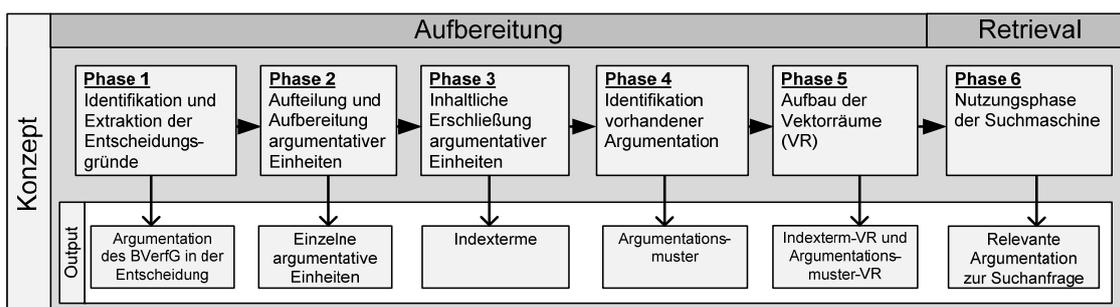


Abb. 14: Phasenkonzept

#### 3.2 Architektur des Softwareprototyps

Abbildung 15 gibt einen Überblick über die Software-Architektur, die aus dem oben gezeigten Phasenkonzept entwickelt wurde. Eine Detailbeschreibung der Systemarchitektur und ihrer Komponenten kann in HOUY ET AL. (2015) nachvollzogen werden.<sup>75</sup>

<sup>73</sup> Vgl. HOUY ET AL. (2015), S. 17f.

<sup>74</sup> Diese Darstellung ist auch zu finden in HOUY ET AL. (2015), S. 18, und präzisiert das Phasenkonzept, das in HOUY ET AL. (2013) ausführlicher und in HOUY ET AL. (2014) in Kurzform präsentiert wurde.

<sup>75</sup> Vgl. HOUY ET AL. (2015), S. 19f.

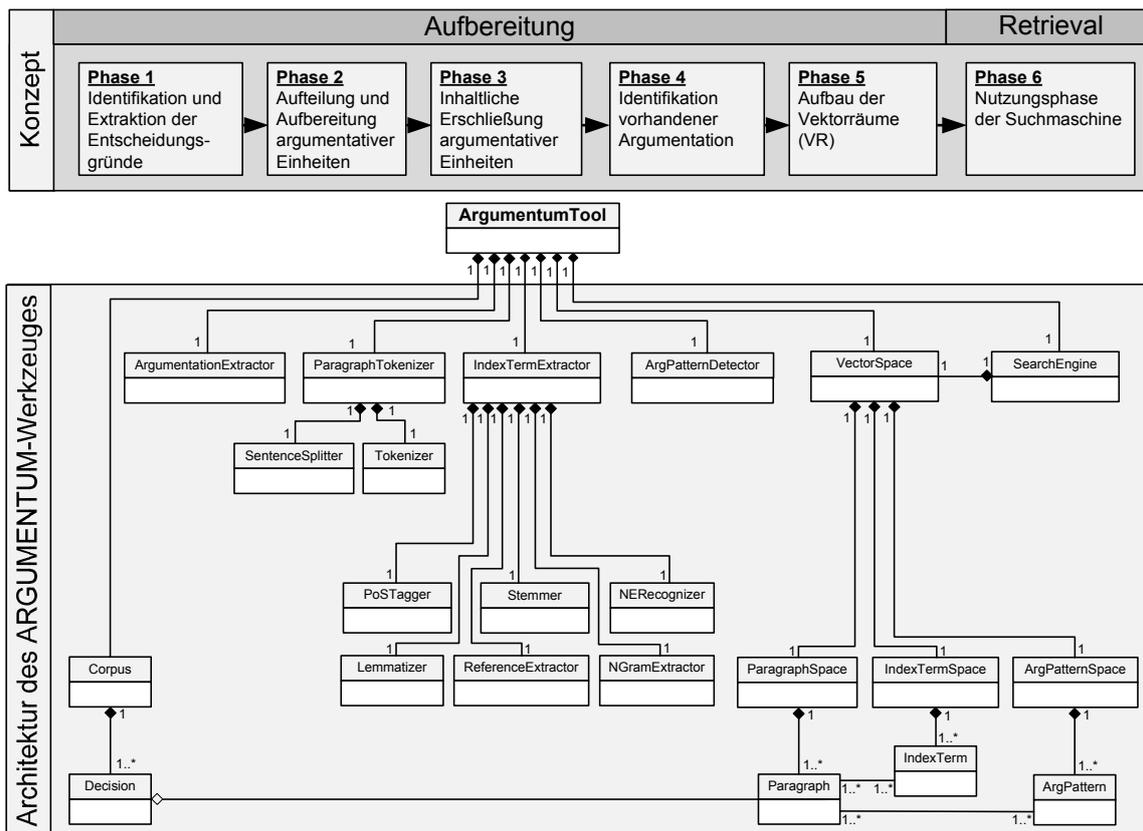


Abb. 15: Architekturentwurf des ARGUMENTUM-Software-Werkzeuges<sup>76</sup>

### 3.3 Implementierung und Nutzung des Prototyps

Folgende Abbildungen geben einen Überblick über die Implementierung des ARGUMENTUM-Prototyps und deuten dessen Nutzung beim Retrieval von Argumentationsstrukturen in Gerichtsurteilen an. Eine detailliertere Beschreibung des finalen Prototyps kann in HOUY ET AL. (2015) nachvollzogen werden.<sup>77</sup>

<sup>76</sup> Diese Darstellung ist auch zu finden in HOUY ET AL. (2015), S. 19.

<sup>77</sup> Vgl. HOUY ET AL. (2015), S. 20ff.

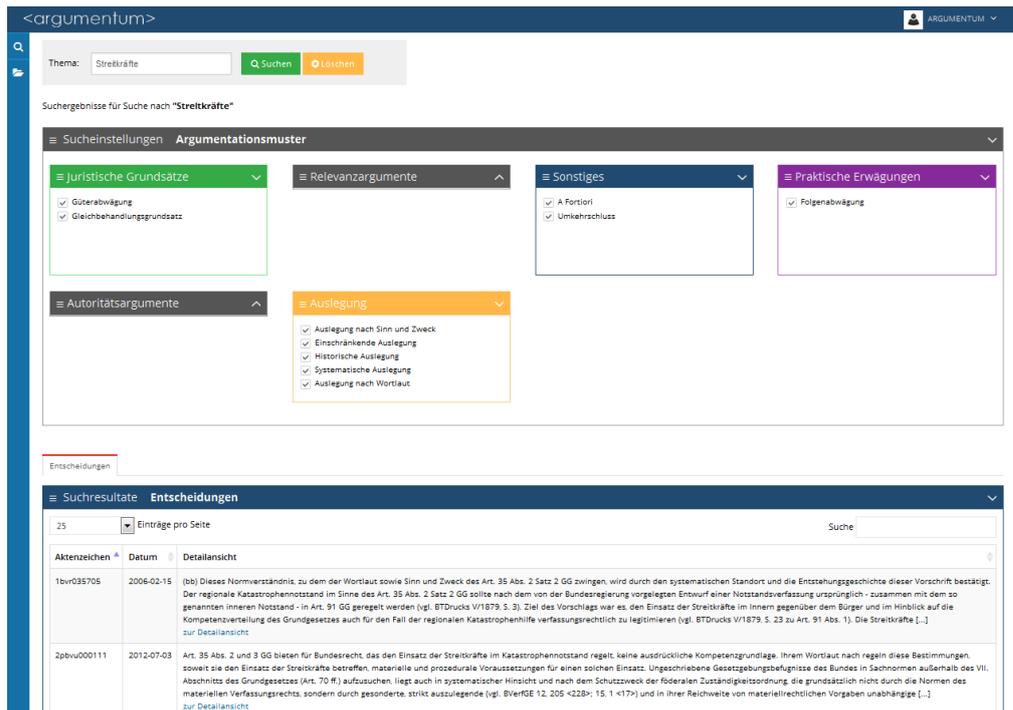


Abb. 16: Bedienoberfläche des finalen Prototyps mit priorisierter Ergebnisliste<sup>78</sup>



Abb. 17: Detaildarstellung mit annotierten und markierten Argumentationsmustern<sup>79</sup>

<sup>78</sup> Diese Darstellung ist auch zu finden in HOUY ET AL. (2015), S. 21.

<sup>79</sup> Diese Darstellung ist auch zu finden in HOUY ET AL. (2015), S. 22.

## 4 Test und Evaluation

### 4.1 Einleitung und Zielsetzung

Im Rahmen des Projektes ARGUMENTUM wurden die erarbeiteten Detailkonzepte zur Analyse der elektronischen Rechtsprechungscorpora des Bundesverfassungsgerichts sowie die beiden darauf aufbauenden Softwareimplementierungen (1. und 2. Entwicklungsiteration) verschiedenen Tests unterzogen und evaluiert. Dieses Kapitel beschreibt die Ergebnisse dieser Untersuchungen und die daraus abgeleiteten Entscheidungen im Entwicklungsprozess.

Die durchgeführten Test- und Evaluationsaktivitäten konzentrieren sich in erster Linie auf technische Aspekte der Implementierungen. Hierzu zählen Erkenntnisse aus funktionalen Tests, die auf eine Überprüfung der korrekten Funktionsweise sowohl einzelner Softwaremodule als auch integrierter Systemkomponenten abzielen. Des Weiteren wurden Aspekte der Gebrauchstauglichkeit bzw. Bedienbarkeit der beiden Prototypen bei *Usability*-Tests untersucht, um die Bedienoberflächen der Systeme an die Bedürfnisse fachspezifischer Anwender anzupassen. Diese Untersuchungen wurden gemeinsam mit Fachanwendern unterschiedlicher Disziplinen vorgenommen, wobei eine möglichst heterogene Gruppenzusammensetzung aus den Fachbereichen der beteiligten Konsortialpartner (*Wirtschaftsinformatik, Rechtsinformatik, Theoretische Philosophie*) angestrebt wurde. Hierbei konnten vielfältige Erkenntnisse gewonnen werden, die mit Blick auf die unterschiedlichen Zielsetzungen bei der Nutzung der Prototypen interessante Rückschlüsse auf Potenziale zur Unterstützung fachspezifischer Arbeitsabläufe zulassen.

Im Anschluss an die funktionale Testung der Systeme wurde anhand verschiedener exemplarischer Anwendungsszenarien überprüft, inwiefern ihr Einsatz zur einer Effektivitäts- und Effizienzsteigerung bei der Recherche nach argumentativen Zusammenhängen führen kann. Als Vergleichsbasis wurde hierzu die „manuelle Vorgehensweise“ herangezogen. Diese Evaluationsmaßnahmen wurden ausschließlich durch die Konsortialpartner mit rechtswissenschaftlichem Background durchgeführt. An dieser Stelle werden diesbezüglich keine Detailergebnisse dazu präsentiert.

Zusammenfassend werden im vorliegenden Bericht insbesondere folgende Inhalte präsentiert, mit den sich das IWi im DFKI beschäftigt hat:

- Funktionale Testung der einzelnen Systemmodule und integrierten Systemkomponenten sowie deren *Zusammenwirkung* in der Gesamtimplementierung,
- Untersuchung der *Usability* der entwickelten Prototypen in Bezug auf Bedienbarkeit und Unterstützung fachspezifischer Arbeitsabläufe,
- erste Ansätze einer fachlichen Evaluation anhand praktischer Anwendungsszenarien zur Vorbereitung weiterer Evaluationsschritte der Projektpartner.

## 4.2 Verwendete Methodik

Das Testen der beiden Prototypimplementierungen erfolgte jeweils in mehreren Stufen. Zunächst wurde eine rein funktionale Testung von einer fachbezogenen Testung mit Blick auf die erzielten Analyseergebnisse unterschieden. Die funktionale Testung erfolgte darüber hinaus in mehreren Stufen und unter Einbeziehung verschiedener Mitglieder des Entwicklungsteams. Der grundsätzliche Ablauf ist im Folgenden dargestellt:

1. Es wurden technische Funktionstests auf Seiten des Implementierungspartners IWi durchgeführt. Diese umfassten separate **Modultests** für jede Softwarekomponente mit dem Ziel, die korrekte Implementierung und Einhaltung der zuvor erstellten Spezifikation sicherzustellen. Die konkrete Funktionsweise implementierter Komponenten wurde zunächst vom jeweiligen Entwickler getestet und dokumentiert. Anschließend erfolgte ein übergreifender Test im Rahmen des gesamten Entwicklerteams, um die Objektivität der Tests sicherzustellen und eine gute Fehlererkennungsrate zu gewährleisten.

Weiterhin wurden im Anschluss an die individuellen Modultests übergreifende **Integrationstests** durchgeführt, um das Zusammenwirken der Module untereinander zu überprüfen.

2. Neben der rein technischen Testung der Software wurden gemeinsam mit dem Projektpartner IfRI erste **fachbezogene Tests** durchgeführt. Diese dienten einer Plausibilitätskontrolle der implementierten Funktionen und ermöglichten eine

Feinabstimmung von internen Parametern zur Verbesserung der erzielten Ergebnisse. Auf diese Weise konnte frühzeitig im Entwicklungsprozess Feedback zum bisherigen Stand der Implementierung gesammelt und für die weiteren Arbeiten entsprechend berücksichtigt werden.

Die Ergebnisse sowohl der technischen als auch der fachbezogenen Tests wurden in einem projektübergreifenden Bug-Tracking-System dokumentiert, auf das alle Projektpartner Zugriff hatten. Auf diese Weise wurde der gesamte Prozess der Testung transparent für alle Beteiligten dargestellt und konnte jederzeit um Kommentare, Fragen und Anregungen zu einzelnen Punkten ergänzt werden.

Einen zweiten Schwerpunkt bildete neben der funktionalen Testung die **fachliche**, d. h. stärker inhaltlich ausgerichtete Evaluation der entstandenen Softwareprototypen mit juristischen Fachanwendern. Diese wurde durch verschiedene Testmethoden realisiert, welche bereits frühzeitig im Entwicklungsprozess eingesetzt wurden. Auf diese Weise konnte das hierbei gewonnene Feedback in der weiteren Entwicklung umgesetzt und effizient zu einer engen Ausrichtung des Prototyps an den Bedürfnissen von Fachanwendern verwendet werden.

## 4.3 Funktionale Testung

### 4.3.1 Vorbemerkungen

Die Verfeinerung des Basiskonzeptes zur Analyse und Synthese von argumentativen Strukturen und die anschließende Implementierung erfolgten im Projekt ARGUMENTUM in zwei größeren Iterationen. Zu jeder dieser Iterationen existiert ein Softwareprototyp, der in verschiedenen Modulen die jeweiligen Schritte des Basiskonzeptes umsetzt. Diese wurden gemäß den in den folgenden Abschnitten dargestellten Testmethoden zunächst auf technischer Ebene (*Modul- und Integrationstests*) und fachlicher Ebene (*Fachbezogene Tests*) geprüft und abschließend in Bezug auf die Kompatibilität zu verschiedenen Computersystemen und Softwareversionen hin untersucht (*Systemtests*).

### 4.3.2 Modul- und Integrationstests

Die Softwareprototypen wurden inkrementell entwickelt und in verschiedene Softwaremodule gekapselt, um eine funktionale Trennung einzelner Systembereiche zu ermöglichen. Jedes dieser Module wurde sowohl separat als auch im Zusammenspiel mit interagierenden Modulen gemäß seinen spezifischen Ausgabewerten getestet. Die Implementierung der einzelnen Module kann grundsätzlich unterschieden werden in:

- A. eine Implementierung unter Einbeziehung existierender externer Funktionalitäten und
- B. die vollständige Eigenentwicklung der benötigten projektspezifischen Funktionalitäten.

In den folgenden Abschnitten werden diese beiden Herangehensweisen mit Blick auf die Besonderheiten für die Funktionstestung beschrieben.

#### A. Modulrealisierung durch externe Funktionalität

Zur Realisierung einiger Module wurden, unter Berücksichtigung von modularen Abhängigkeiten, bestimmte Funktionalitäten durch die Einbindung externer Softwarebibliotheken umgesetzt. So wurden die verwendeten *Stemmer*, *Part-of-Speech-Tagger*, *Lemmatizer* und *Named Entity Recognizer* aufgrund der Komplexität der jeweils durch sie realisierten Aufgabe unter Verwendung von *State-of-the-Art*-Implementierungen für die deutsche und englische Sprache umgesetzt. Hierbei wurden etablierte Implementierungen verwendet, die in Bezug auf Performance und Erkennungsgenauigkeit optimiert sind und breite Anwendung in verschiedenen Bereichen erfahren. Um die korrekte Funktionalität dieser Bibliotheken – speziell im Hinblick auf die sprachlichen Eigenheiten der im Corpus verwendeten Rechtssprache – sicherzustellen, wurden im Vorfeld der Implementierung Untersuchungen an verschiedenen Teilmengen des Corpus durchgeführt. Die Ergebnisse dieser Untersuchungen wurden hinsichtlich ihrer Sinnhaftigkeit geprüft, um eine hohe Analysequalität sicherzustellen und somit eine verlässliche Grundlage für weitere Schritte im Verarbeitungsprozess zur Erkennung argumentativer Strukturen zu schaffen. Dieser Schritt wurde insbesondere deshalb notwendig, weil die meisten der verwendeten Implementierungen mit standardisierten deutschen Corpora trainiert wurden, die nicht an die Besonderheiten der Rechtssprache angepasst sind.

## **B. Modulrealisierung durch Eigenimplementierung**

Die übrigen Module wurden unabhängig von extern eingebundenen Bibliotheken implementiert und konnten während der Implementierungszeit detailliert verfeinert werden. In Bezug auf die funktionale Testung wurde zunächst für jedes Modul in theoretischer Form das gewünschte Verhalten spezifiziert und im Anschluss entsprechend durch individuelle Testfälle verifiziert. Die Verhaltensspezifikation der Module *ArgPattern-Detector* und *ReferenceExtractor* gestaltet sich aufgrund der großen inhaltlichen und strukturellen Variabilität von argumentativen Strukturen besonders komplex. Sie wurde daher im Verlauf der Implementierung inkrementell entwickelt und im Anschluss an jede Iteration getestet.

### **4.3.3 Fachbezogene Tests**

Zur Verifikation der korrekten Funktionsweise von eigens entwickelten Algorithmen – beispielsweise zur Bestimmung des Übergangs zwischen nicht-argumentativen und argumentativen Passagen eines Urteils – wurde eine enge Abstimmung zwischen den beteiligten Entwicklungspartnern angestrebt. So wurden während der Entwicklungszeit die Ausgabeergebnisse der Implementierungen in zahlreichen Sitzungen gemeinsam untersucht, diskutiert und verfeinert. Aufgrund der intensiven Kooperation konnte die fachliche Qualität der implementierten Funktionalitäten frühzeitig sichergestellt und durch die gemeinsame Variation programmiertechnischer Parameterwerte iterativ optimiert und an die Besonderheiten des vorliegenden Corpus angepasst werden.

### **4.3.4 Systemtests**

Die Implementierung des ersten Prototyps erfolgt als native Java-Anwendung und wurde auf verschiedenen Entwicklungssystemen unter den Betriebssystemen *Linux*, *Microsoft Windows* und *Mac OS X* getestet. Angesichts der grundsätzlichen Plattformunabhängigkeit der Java-Laufzeitumgebung konnte dadurch eine breite Abdeckung aller gängigen Betriebssystemversionen sowie ein uneingeschränkter Zugang zum System gewährleistet werden. Demgegenüber wurde der zweite Softwareprototyp in Form einer Webanwendung umgesetzt, sodass auf Seiten eines Anwendersystems lediglich ein Internetbrowser zur Darstellung der grafischen Oberfläche vorausgesetzt wird. Hierbei

wurde durchgehend auf die Einhaltung wichtiger Standards zur Gestaltung von Webseiten geachtet, um auf allen Systemen eine fehlerfreie Darstellung zu ermöglichen. Zu diesem Zweck wurde die Oberfläche weiterhin detailliert mit den aktuellen Versionen der Internetbrowser *Google Chrome*, *Internet Explorer*, *Mozilla Firefox* und *Safari* getestet. Mit jedem Browser war im Testverlauf eine uneingeschränkte Bedienbarkeit der Webanwendung möglich.

## **4.4 Fachliche Evaluation – Fokus: zweiter Prototyp**

### **4.4.1 Vorbemerkungen**

Im Anschluss an die funktionale Testung der Softwareprototypen erfolgte jeweils – unter der validen Annahme einer technisch einwandfreien Funktion – die fachliche Bewertung und Evaluation der erzielten Analyseergebnisse. Diese fand unter Berücksichtigung der intendierten Nutzungsszenarien jeweils pro Prototyp statt; der *erste* Softwareprototyp diente in erster Linie dem Ziel, das verwendete Rechtsprechungs corpus initial für eine Untersuchung zugänglich zu machen und die grundsätzlichen, innerhalb des Basiskonzepts formulierten Hypothesen zu überprüfen. Weiterhin konnten auf diese Weise mit Blick auf das detaillierte Feinkonzept konzeptuelle und methodische Anpassungen frühzeitig vorgenommen und um Feedback von Fachanwendern ergänzt werden. Aus Sicht der Bedienbarkeit zeigte sich zudem, dass eine webbasierte Softwarelösung von den befragten Anwendern als allgemein einfacher nutzbar empfunden wurde. Der *zweite* Softwareprototyp zielte dagegen weitaus stärker auf eine optimale Unterstützung von Anwendungsfällen der zuvor identifizierten anwendungs- und erkenntnisorientierten Nutzergruppen ab und wurde daher auch speziell vor diesem Hintergrund evaluiert. In verschiedenen Sitzungen wurde durch den Projektpartner IfRI untersucht, wie juristische Fachanwender das System konkret zur Bearbeitung einer definierten Aufgabenstellung nutzen und wie das System hinsichtlich realisierbarer Effektivitäts- und Effizienzsteigerungen im Vergleich zu einer klassischen Recherche nach argumentativen Aussagen zu bewerten ist.

#### 4.4.2 Exemplarische Anwendungsszenarien

Für einen ersten fachlichen Evaluationsversuch, der zur Vorbereitung der tiefergehenden Evaluation durch die juristischen Partner im Projekt dienen sollte, wurden zwei exemplarische Anwendungsszenarien für die Nutzung des webbasierten Softwareprototyps beschrieben. An dieser Stelle werden diese Szenarien erläutert und beispielhaft durch Screenshots des Systems für die einzelnen Prozessschritte detailliert.

Für weitere Untersuchungen – besonders im Hinblick auf tiefergehende Aussagen bezüglich Effektivitäts- und Effizienzgewinne durch den Einsatz der entstandenen Softwareprototypen – sind zukünftig verschiedene Testszenerien denkbar. So stellt die Definition einer detaillierten Benchmarking-Metrik (z. B. in Form einer Zeitmessung) eine Möglichkeit zur Bewertung von zeitlichen Einsparpotenzialen gegenüber einer „manuellen Suchstrategie“ mit herkömmlichen Informationssystemen zur Recherche im rechtswissenschaftlichen Kontext dar.

##### Anwendungsszenario 1

Eine Anwenderin möchte Argumentationen bezüglich der Entscheidungen des Bundesverfassungsgerichts im Hinblick auf die sich gegenüberstehenden Gebiete der Privatsphäre und der Pressefreiheit untersuchen. Sie möchte dabei sämtliche Argumente ermitteln, die das Recht der *Pressefreiheit* erfolgreich über das Recht der *Privatsphäre* stellen und umgekehrt. Zur Erreichung des Ziels wird zunächst der Suchanfragenabschnitt angepasst. Abbildung 18 beschreibt die notwendigen Anpassungen. Im ersten Schritt (1) werden die Begriffe *Privatsphäre* und *Pressefreiheit* in das Suchfeld eingetragen.

Abb. 18: Sucheinstellungen I

Da die Argumente bezüglich der Kontroverse zwischen *Privatsphäre* und *Pressefreiheit* ermittelt werden sollen, wird in Schritt (2) die Option *Alle Begriffe müssen gefunden werden* selektiert. Im Beispiel wird davon ausgegangen, dass die Nutzerin zunächst nicht nach spezifischen Argumentationsmustern suchen möchte, daher wird in Schritt (3) die Option *Mindestens eines der ausgewählten Argumentationsmuster muss gefunden werden* selektiert und anschließend in Schritt (4) die Schaltfläche *Alle aktivieren* gewählt. Eine zeitliche Einschränkung wird nicht angegeben, wodurch alle erfassten Entscheidungstexte durchsucht werden. Nach der Wahl aller Suchparameter wird in Schritt (5) die Suche über einen Klick auf die Schaltfläche *Argumente finden!* gestartet.

### Ergebnisdarstellung einer Suchanfrage

Abbildung 19 zeigt die Ergebnisse der Suchanfrage, sortiert nach ihrer Relevanz bezüglich der gewählten Suchparameter. Die Sortierung der Ergebnisdarstellung kann anschließend durch eine Verschiebung der Gewichtung zwischen Inhalt und Argumentati-

on in Schritt (2) gemäß der verfolgten Suchstrategie angepasst werden. Sie dient bei der gezielten Suche nach bestimmten Argumentationsmustern der Festlegung, ob Ergebnisse eine höhere Relevanz beigemessen werden soll, wenn sie auf der einen Seite eine hohe Überschneidung mit den Suchbegriffen oder auf der anderen Seite eine hohe Übereinstimmung mit den gewählten Argumentationsmustern besitzen. Durch Auswahl des Links zur Detailansicht aus der Spalte *Detailansicht* gelangt die Anwenderin zu den aufbereiteten Entscheidungstexten (3).

**Treffer-Sortierung**

Gewichtung "Inhalt vs. Argumentationsmuster": Stufe 1 von 100

Inhalt: \_\_\_\_\_ Argumentationsmuster: \_\_\_\_\_

Sortierung anpassen

Entscheidungen

**Suchbegriffe** Privatsphäre UND Pressefreiheit  
**Argumentationsmuster** Auslegung nach Sinn und Zweck ODER Einschränkung der Auslegung ODER Historische Auslegung ODER Systematische Auslegung ODER Auslegung nach Wortlaut ODER Literatur ODER Güterabwägung ODER Gleichbehandlungsgrundsatz ODER Folgenabwägung ODER A Fortiori ODER Umkehrschluss  
**Zeitraum** 01.01.1999 bis 31.12.2014

Die Suche lieferte 5 Treffer.

**Suchresultate Entscheidungen**

25 Einträge pro Seite

Relevanz	Aktenzeichen	Datum	Detailansicht	Suchbegriff-Relevanzfaktor	Muster-Relevanzfaktor	Score
1	I BvR 1842/08	2010-09-14	<a href="#">zur Detailansicht</a>	0,080339634013	0,45	0,02171568506813707
2	I BvR 653/96	1996-12-15	<a href="#">zur Detailansicht</a>	0,0919743079512697	0,6	0,020609735931803614
3	I BvR 1505/99	2000-04-04	<a href="#">zur Detailansicht</a>	0,01772306467011957	0,65	0,01837310544653454
4	I BvR 1352/09	2000-03-31	<a href="#">zur Detailansicht</a>	0,01687005209031452	0,45	0,0174321061486224
5	I BvR 1213/97	2000-04-05	<a href="#">zur Detailansicht</a>	0,015396463180719128	0,65	0,015683647444502282

Zeige 1 bis 5 von 5 Einträgen

Vorherige 1 Nächste

Abb. 19: Ergebnisliste der Suchanfrage

### Detaildarstellung einer Entscheidung

Im konkreten Beispiel liefert die durchgeführte Suche, bei Auswahl des Ergebniseintrages mit der höchsten Relevanz, die in Abbildung 20 beschriebene Detailansicht. Da die Detailansicht ausschließlich den für die Argumentmustersuche relevanten Bereich der Entscheidungen beinhaltet, kann im gekennzeichneten Bereich (1) die vollständige Entscheidung des Bundesverfassungsgerichts eingesehen werden. Im Bereich (2) werden alle auftretenden Argumentationsmuster im Entscheidungstext aufgelistet. Die Argumentationsmuster lassen sich gleichzeitig auch im Entscheidungstext durch die entsprechend farblich hinterlegten Textstellen erkennen.

Die mit (3) gekennzeichneten Begriffe stellen die Suchbegriffe dar. Sie werden in der Detailansicht im Entscheidungstext fett hervorgehoben und sind daher leicht auffindbar.

The screenshot displays a legal decision interface with several key components:

- Table of Contents (Gliederungsübersicht):** Located on the left, it lists the decision's structure (1 BvR 1842/08, I., 1., a), b), aa), (1), (2), bb), (1), (2), c), 2., a), b), c), d), e), 3., 1., a), b), aa), bb), 2., a), b), aa), bb), 3., a), b), 4., 5.).
- Decision Text (Entscheidungstext):** The main text on the right, containing paragraphs 32 and 33. Key terms are highlighted in red boxes and numbered:
  - 1:** "Entscheidungstext (öffne Volltext auf bundesverfassungsgericht.de)"
  - 2:** "enthaltene Argumentationsmuster"
  - 3:** "Privatsphäre" (bolded in the text)
  - 3:** "Pressefreiheit" (bolded in the text)
  - 4:** "umso schwerer" (underlined in the text)
  - 5:** "Argumentationsmuster-Info"
  - 6:** A red circle highlighting the table of contents area.
- Argumentation Patterns (enthaltene Argumentationsmuster):** A panel on the right showing checked boxes for "A Fortiori" and "Güterabwägung".
- Argumentation Pattern Info (Argumentationsmuster-Info):** A panel at the bottom right showing "Zugeordnete Argumentationsmuster für schwerer (Rn. 32)" and "A Fortiori: ++".

Abb. 20: Detailansicht einer Entscheidung

Die visuelle Aufbereitung ermöglicht es Anwendern, auftretende Suchbegriffe und Argumentationsmuster zu verknüpfen und somit relevante Argumente zu erkennen. Die unterstrichenen Wörter (4) entsprechen den Indikatoren für spezifische Argumentationsmuster. Durch Anklicken per Maus erscheinen im Bereich (5) Informationen zu demjenigen Argumentationsmuster, das dem markierten Argument zugrunde liegt. Im konkreten Fall handelt es sich um ein *A-Fortiori*-Argument, welches u. a. über die Indi-

katoren *umso schwerer* identifiziert wurde. Anwender können an dieser Stelle prüfen, ob das identifizierte Argument sich auf konkrete Problemstellung übertragen lässt, d. h. ob das Argument plausibel wiederverwendet werden kann. Bereich (6) visualisiert die extrahierte Hierarchie des Entscheidungstextes. Sie kann herangezogen werden, um das Gewicht des Arguments im gesamten Kontext der Entscheidung abzuschätzen.

Für Nutzende besteht weiterhin die Möglichkeit, in der Detailansicht die auftretenden Argumente nach Argumentationsmustern zu filtern. Im aufgeführten Beispiel entscheidet sich die Nutzerin zur ausschließlichen Untersuchung von *A-Fortiori*-Argumenten. Sie wählt dazu im Bereich (2) das selektierte Feld *Güterabwägung* ab. Dies hat zur Folge, dass farbliche Markierungen und ein Unterstreichen der Indikatoren nur noch für *A-Fortiori*-Argumente vorhanden sind. Das Resultat ist in Abbildung 21 dargestellt.

Schutzbedürfnis bestehe auch für solche Personen, die wegen ihrer Stellung oder ihrer Taten besondere öffentliche Beachtung fänden. Vorliegend sei die **Privatsphäre** der Klägerin berührt, weil es allein um Spekulationen über ihr Aussehen und ihren Lebenswandel gehe.

- 32 Zwar stehe die **Privatsphäre** bei einer Presseveröffentlichung in einem Spannungsverhältnis mit der ebenfalls grundrechtlich geschützten Äußerungs- und **Pressefreiheit**. Dabei wiege der Schutz der Persönlichkeit **umso schwerer, je geringer der Informationswert für die Öffentlichkeit sei**. Hiernach falle die Abwägung zugunsten der Klägerin aus. Diese nehme allenfalls untergeordnet und nicht selbständig repräsentative Aufgaben wahr und sei anders als ihre Mutter keine sogenannte absolute Person der Zeitgeschichte. Sie müsse es daher nicht hinnehmen, dass sie über einen

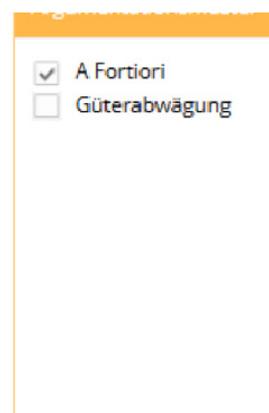


Abb. 21: Filtern von Argumenten

## Anwendungsszenario 2

Im zweiten Anwendungsszenario wird der Fall betrachtet, dass die Anwenderin ihre Suche auf Argumente einschränken möchte, die in Entscheidungen der letzten fünf Jahre vorgetragen worden sind. Zudem möchte sie den Argumenthorizont einseitig erweitern, d. h., sie möchte zunächst nach allen Argumentationsinhalten suchen, die im Themenbereich *Meinungsfreiheit* oder *Pressefreiheit* eingesetzt wurden. Weiterhin sollen dann ausschließlich Autoritätsargumente in Form von Literaturbelegen und Argumente basierend auf dem juristischen Grundsatz der Güterabwägung berücksichtigt werden. Dazu

ist im ersten Schritt eine Anpassung der Suchmaske notwendig. Abbildung 22 zeigt die notwendigen Manipulationen der Suchparameter. Im ersten Schritt (1) fügt die Anwenderin in das Suchfeld die Begriffe *Meinungsfreiheit* und *Pressefreiheit* ein. Da die Suchergebnisse nicht zwingend beide Themen aufgreifen müssen, wird in Schritt zwei die Option *Mindestens ein Begriff muss gefunden werden* gewählt (2). Anschließend erfolgt die Auswahl der zu identifizierenden Argumentationsmuster. Da standardmäßig alle Muster aktiviert sind, werden über das Betätigen der Schaltfläche *alle deaktivieren* in Schritt (3) zunächst alle Muster deaktiviert. Darauf folgt in Schritt (4) die Auswahl der gewünschten Argumentationsmuster *Literatur* aus der Gruppe der *Autoritätsargumente* und *Güterabwägung* aus der Gruppe der *Juristischen Grundsätze*. Da die Nutzerin nach beliebigem Auftreten der gewählten Muster suchen möchte, wird in Schritt fünf die Option *Mindestens eines der ausgewählten Argumentationsmuster muss gefunden werden* gewählt (5). In Schritt (6) werden durch Anklicken der in der Abbildung rot eingrahmten Pfeilspitze die Einstellungsmöglichkeiten zur *Zeitlichen Einschränkung* angezeigt. Da die

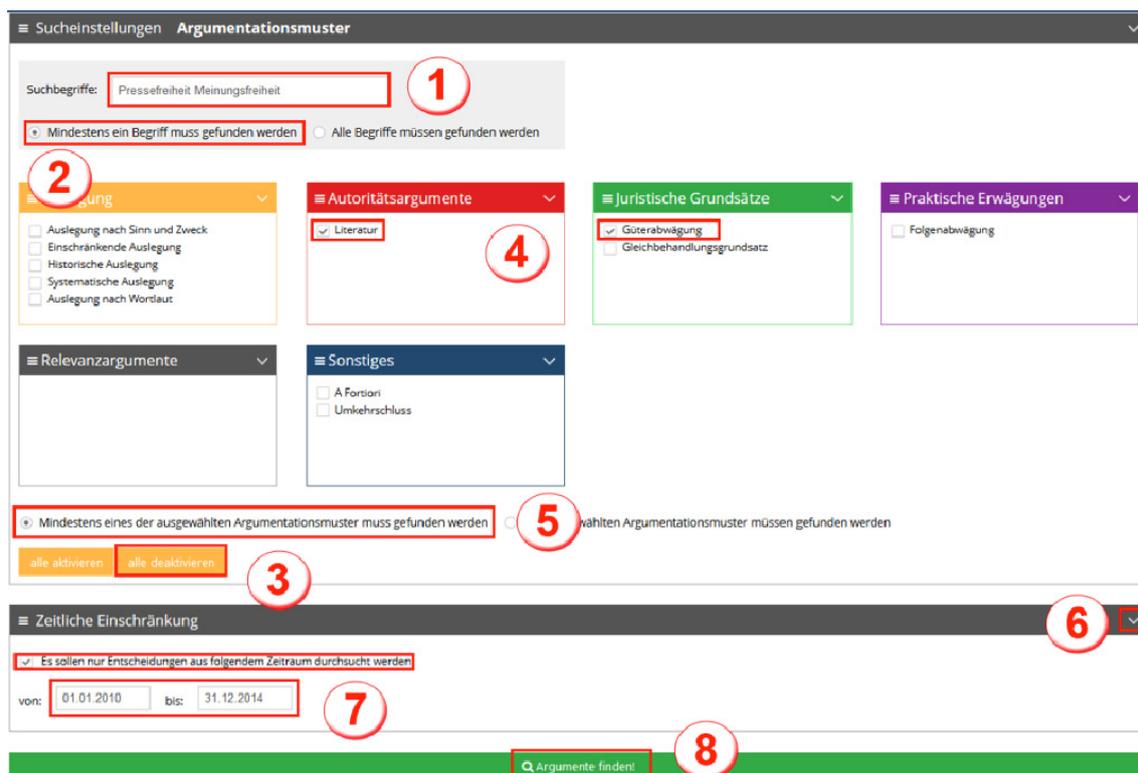


Abb. 22: Sucheinstellungen II

In Schritt (6) werden durch Anklicken der in der Abbildung rot eingrahmten Pfeilspitze die Einstellungsmöglichkeiten zur *Zeitlichen Einschränkung* angezeigt. Da die

Anwenderin ausschließlich Argumente der letzten fünf Jahre untersuchen möchte, markiert sie in Schritt (7) das Feld *Es sollen nur Entscheidungen aus folgendem Zeitraum durchsucht werden* und gibt die Datumsgrenzen *von* und *bis* in die jeweiligen Felder ein. Nutzern steht es frei, das Datum entsprechend dem Format TT.MM.JJJJ einzugeben oder durch einen Klick in das Eingabefeld eine Kalenderansicht zu öffnen und ein Datum per Mausclick auszuwählen. Die Nutzerin startet abschließend in Schritt (8) die Suchanfrage mit einem Klick auf die Schaltfläche *Argumente finden!*

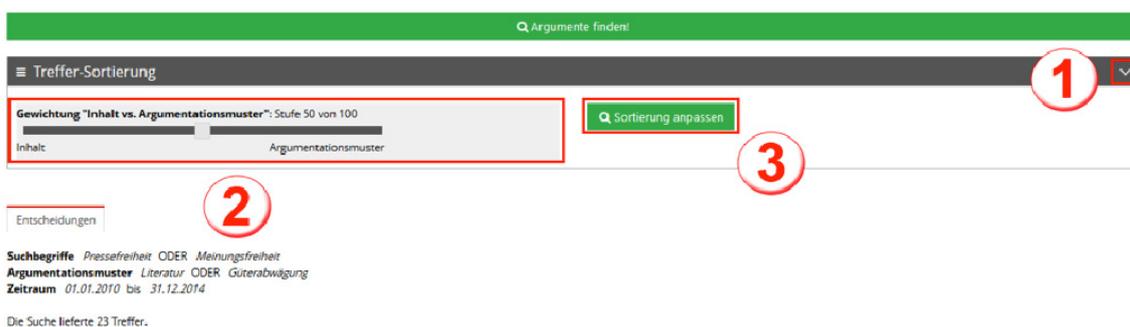


Abb. 23: Änderung der Relevanzkriterien zur Ergebnissortierung

Da im Vergleich zu Anwendungsszenario 1 Wert darauf gelegt wird, dass sowohl die ausgewählten Muster, als auch die angegebenen Suchbegriffe in die Suche gleichermaßen einfließen sollen, erfolgt eine Anpassung der Relevanzberechnung der Ergebnisanzeige. Die Anpassung wirkt sich nur auf die Sortierung der Ergebnisse aus und hat keinen Einfluss auf den Umfang der Ergebnismenge. Abbildung 23 zeigt die vorzunehmenden Anpassungen. Dazu werden in Schritt (1) die Einstellungsmöglichkeiten über den rot markierten Pfeil in der Leiste *Treffer-Sortierung* geöffnet. Im Anschluss wird in Schritt (2) eine Gewichtung zwischen Suchbegriffen und Argumentationsmustern gewählt. Für eine gleichmäßige Gewichtung beider Kriterien wird der Schieberegler mittig auf den Wert *50 von 100* verschoben. Abschließend wird die gewählte Gewichtung durch Betätigen der Schaltfläche *Sortierung anpassen* (3) auf das Suchergebnis angewandt.

#### 4.4.3 Präsentation vor Fachpublikum und Experteninterviews

In Ergänzung zu den Testungen anhand der vorgestellten Anwendungsszenarien mit Fachanwendern wurden die Projektergebnisse und insbesondere die entstandenen Prototypen zwecks Beurteilung einem breiten Fach- und Expertenpublikum vorgeführt. Hervorzuheben ist an dieser Stelle die Präsentation des *ersten Softwareprototyps* auf der *CeBit 2014* in Hannover am Stand des *Deutschen Forschungszentrums für Künstliche Intelligenz* (DFKI GmbH). Durch die öffentliche Präsentation konnten in Gesprächen und bei Diskussionen mit Fachbesuchern aufschlussreiche Rückmeldungen über mögliche Einsatzszenarien und Ansatzpunkte zur Weiterentwicklung gesammelt werden.

Eine direkte Rückmeldung zur Qualität der Analyseergebnisse und detaillierte Einschätzung zu den realisierten Funktionalitäten konnte auch durch Präsentationen und Gespräche des Projektpartners *Institut für Rechtsinformatik* (IfRI) u. a. beim *Bundesverfassungsgericht* in Karlsruhe oder beim jährlichen *EDV-Gerichtstag* in Saarbrücken erlangt werden. Die Fachexpertise der befragten Juristen und Wissenschaftler begründet die hohe Relevanz der Rückmeldungen, die Impulse für den weiteren Projektverlauf gaben.

#### 4.5 Fazit zu Test und Evaluation

Neben technischen Untersuchungen im Rahmen von Modul-, Integrations- und Systemtests wurden zur Vorbereitung weiterer inhaltlicher Evaluationsmaßnahmen durch die juristischen Projektpartner auch erste fachbezogene Untersuchungen durchgeführt, um die Korrektheit der implementierten Funktionalitäten sicherzustellen. Es wurde hier insbesondere der zweite Prototyp aus der Perspektive juristischer Fachanwender anhand zweier potentieller Anwendungsszenarien getestet und diese Szenarien dargestellt. Dem Nutzerfeedback folgend, das im Rahmen sämtlicher Tests und Evaluationsmaßnahmen im Projekt ARGUMENTUM gesammelt wurde, bietet der Prototyp erhebliche Potentiale für die Arbeitsprozesse von Juristen und kann helfen, typische Suchen nach geeigneten Argumentationen zu beschleunigen und zu verbessern.

## 5 Resümee

Im vorliegenden Arbeitsbericht wurden in Ergänzung zu den existierenden Publikationen zum ARGUMENTUM-Projekt weitere, bislang unveröffentlichte Materialien und Überlegungen präsentiert. Dadurch soll die Darstellung der im Projektkontext erarbeiteten Ergebnisse komplettiert und in ihrer Gesamtheit für Interessierte zugänglich gemacht werden. In Kapitel zwei wurde zunächst die strukturierte Literaturanalyse zu relevanten Themen im Projektkontext dargestellt. In knapper Form wurde in Kapitel drei ein Überblick zum Konzept und der Implementierung des entwickelten Softwareprototyps gegeben. Detailliertere Ausführungen hierzu können in den jeweils angegebenen Publikationen nachvollzogen werden. Abschließend wurden in Kapitel fünf die Ergebnisse funktionaler Tests und einer ersten fachlichen Evaluation aus Sicht des Instituts für Wirtschaftsinformatik erläutert. Exemplarisch wurden zwei Anwendungsszenarien zur Nutzung des finalen Softwareprototyps anhand konkreter Fragestellungen detaillierter beschrieben.

## 6 Literatur

- Abbas, S. und Sawamura, H. (2008a): Argument mining using highly structured argument repertoire; in: *1st International Conference on Educational Data Mining (EDM 2008)*, Montreal, Canada, 2008a, S. 202-209.
- Abbas, S. und Sawamura, H. (2008b): A first step towards argument mining and its use in arguing agents and ITS; in: I. Lovrek, R. J. Howlett und L. C. Jain (Hrsg.): *Knowledge-Based Intelligent Information and Engineering Systems, LNAI 5177*; Springer, Berlin; 2008b, S. 149-157.
- Abbas, S. und Sawamura, H. (2010): Argument mining from RADB and its usage in arguing agents and intelligent tutoring system; in: D. Srinivasan und L. C. Jain (Hrsg.): *Innovations in Multi-Agent Systems and Applications*; Springer, Berlin; 2010, S. 113-147.
- Abbas, S. und Sawamura, H. (2012): Argument mining based on a structured database and its usage in an intelligent tutoring environment; *Knowledge and Information Systems*, 30. Jg.; 2012, H. 1; S. 213-246.
- Bresnan, J. (2000): *Lexical-functional syntax*; Wiley Blackwell, Hoboken, NJ; 2000.
- Chesñevar, C., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G., South, M., Vreeswijk, G. und Willmott, S. (2006): Towards an argument interchange format; *The Knowledge Engineering Review*, 21. Jg.; 2006, H. 4; S. 293–316.
- Chklovski, T. (2003): *Using Analogy to Acquire Commonsense Knowledge from Human Contributors*, MIT Artificial Intelligence Laboratory technical report AITR-2003-002.
- Chklovski, T., Ratnakar, V. und Gil, Y. (2005): User interfaces with semi-formal representations: a study of designing argumentation structures; in: *Proceedings of the 10th international conference on Intelligent user interfaces (IUI '05)*, AMC, New York, 2005, S. 130-136.
- Cunningham, H. (2002): *GATE, a General Architecture for Text Engineering*; *Computers and the Humanities* 36. Jg.; 2002, H. 2; S. 223–254
- Dönmez, P., Rosé, C., Stegmann, K., Weinberger, A. und Fischer, F. (2005): Supporting CSCL with automatic corpus analysis technology; in: *Proceedings of the 2005 Conference on Computer Support for Collaborative Learning: Learning 2005: The Next 10 Years!*, Taipei, Taiwan, 2005, S. 125-134.
- Feldman, R. und Sanger, J. (2006): *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*; Cambridge University Press, Cambridge 2006.
- Feng, D., Kim, J., Shaw, E. und Hovy, E. (2006): Towards Modeling Threaded Discussions using Induced Ontology Knowledge; in: Y. Gil und R. J. Mooney (Hrsg.): *Proceedings of the National Conference on Artificial Intelligence*, Boston, Massachusetts, 2006, S. 1289-1294
- Fettke, P. (2006): State-of-the-Art des State-of-the-Art. Eine Untersuchung der Forschungsmethode „Review“ innerhalb der Wirtschaftsinformatik; *Wirtschaftsinformatik*, 48. Jg.; 2006, H. 4; S. 257-266.
- Gkotsis, G. und Karacapilidis, N. (2011): On the exploration and exploitation of structural similarities in argumentative discourses; in: *6th International*

- Conference on Web Information Systems and Technologies (WEBIST 2010)*, J. Filipe und J. Cordeiro (Hrsg.), INSTICC Press Valencia, 2011, S. 137-143.
- Gordon, T. F. und Karacapilidis, N. (1997): The Zeno Argumentation Framework; in: J. Zeleznikow, D. Hunter und L. K. Branting (Hrsg.): Proceedings of the 6th international conference on Artificial intelligence and law (ICAIL '97); ACM New York, NY, USA; 1997, S. 10-18.
- Gordon, T. F., Prakken, H. und Walton, D. (2007): The Carneades Model of Argument and Burden of Proof; *Artificial Intelligence*, 171. Jg.; 2007, H. 10-15; S. 875-896.
- Hearst, M. A. (1994): Multi-paragraph segmentation of expository text; in: Proceedings of the 32th Annual Meeting of the Association for Computational Linguistics (ACL '94), Las Cruces, NM, 1994, S. 9-16
- Hogenboom, A., Hogenboom, F., Kaymak, U., Wouters, P. und de Jong, F. (2010): Mining Economic Sentiment Using Argumentation Structures; in: J. Trujillo, G. Dobbie, H. Kangassalo, S. Hartmann, M. Kirchberg, M. Rossi, I. Reinhartz-Berger, E. Zimányi und F. Frasincar (Hrsg.): *Advances in Conceptual Modeling – Applications and Challenges*, LNCS 6413; Springer Berlin, Heidelberg; 2010, S. 200-209.
- Houy, C., Niesen, T., Calvillo, J., Fettke, P. und Loos, P. (2014): Konzept und Architektur eines Software-Werkzeuges zur automatisierten Identifikation und Analyse von Argumentationsstrukturen; in: *Gemeinsam Electronic Government (ziel)gruppengerecht gestalten und organisieren. Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) (FTVI & FTRI-14). Lecture Notes in Informatics (LNI), Vol. 229*, D. Lück-Schneider, T. Gordon, S. Kaiser, J. von Lucke, E. Schweighofer, M. A. Wimmer und M. G. Löhe (Hrsg.), Gesellschaft für Informatik (GI), Bonn, Berlin, Germany, 2014, S. 113-124.
- Houy, C., Niesen, T., Calvillo, J., Fettke, P., Loos, P., Krämer, A., Schmidt, K., Herberger, M., Speiser, I., Gass, A., Schneider, L. und Philippi, T. (2015): Konzeption und Implementierung eines Werkzeuges zur automatisierten Identifikation und Analyse von Argumentationsstrukturen anhand der Entscheidungen des Bundesverfassungsgerichts im Digital-Humanities-Projekt ARGUMENTUM; *Datenbank-Spektrum*, 15. Jg.; 2015, H. 1; S. 15-23.
- Houy, C., Niesen, T., Fettke, P. und Loos, P. (2013): Towards Automated Identification and Analysis of Argumentation Structures in the Decision Corpus of the German Federal Constitutional Court; in: *7th IEEE International Conference on Digital Ecosystems and Technologies (IEEE-DEST)*, IEEE Computer Society, Menlo Park, California, USA, 2013, S. 506-516.
- Karacapilidis, N. und Papadias, D. (2001): Computer supported argumentation and collaborative decision making: the Hermes system; *Information Systems*, 26. Jg.; 2001, H. 4; S. 259-277.
- Kim, J., Shaw, E., Ravi, S., Tavano, E., Arromratana, A. und Sarda, P. (2008): Scaffolding On-Line Discussions with Past Discussions - An Analysis and Pilot Study of PedaBot; Proceedings of the 9th international conference on Intelligent Tutoring Systems (ITS 2008), 2008, S. 343-352.

- Landauer, T. und Dumais, S. (1997): A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge; *Psychological Review*, 104. Jg.; 1997, H. 2; S. 211-240.
- Mayfield, E. und Rosé, C. (2010): An interactive tool for supporting error analysis for text mining; in: *Proceedings of the NAACL HLT 2010 Demonstration Session* Los Angeles, California, 2010, S. 25–28.
- Mochales-Palau, R. und Moens, M.-F. (2009): Argumentation mining: The detection, classification and structure of arguments in text; in: *The 12th International Conference on Artificial Intelligence and Law (ICAIL '09)*, ACM, Barcelona, 2009, S. 98-107.
- Mochales Palau, R. und Moens, M.-F. (2007): ACILA - Automatic detection of arguments in legal cases; in: *Proceedings of the Workshop on Semantic Web Technology for Law*, Stanford, CA, 2007, S. 5-9.
- Mochales, R. und Ieven, A. (2009): Creating an argumentation corpus: do theories apply to real arguments?; in: *Creating an argumentation corpus: do theories apply to real arguments? - A case study on the legal argumentation of the ECHR (ICAIL'09)*, ACM, New York, 2009, S. 21-30.
- Mochales, R. und Moens, M.-F. (2008): Study on the Structure of Argumentation in Case Law; in: *Proceedings of JURIX 2008: The 21st International Conference on Legal Knowledge and Information Systems, Frontiers in Artificial Intelligence and Applications Vol. 189*, E. Francesconi, G. Sartor und D. Tiscornia (Hrsg.), IOS Press, Florence, Italy, 2008, S. 11-20.
- Mochales, R. und Moens, M.-F. (2011): Argumentation mining; *Artificial Intelligence and Law*, 19. Jg.; 2011, H. 1; S. 1-22.
- Mu, J., Stegmann, K., Mayfield, E., Rosé, C. und Fischer, F. (2012): The ACODEA framework: Developing segmentation and classification schemes for fully automatic analysis of online discussions; *International Journal of Computer-Supported Collaborative Learning*, 7. Jg.; 2012, H. 2; S. 285-305.
- Pinkwart, N. (2005): *Collaborative Modeling in Graph Based Environments*; dissertation.de 2005.
- Pinkwart, N., Alevén, V., Ashley, K. und Lynch, C. (2006a): Schwachstellenermittlung und Rückmeldungsprinzipien in einem intelligenten Tutorensystem für juristische Argumentation; in: *DeLFI 2006, Tagungsband der 4. e-Learning Fachtagung Informatik*, Gesellschaft für Informatik, Darmstadt, 2006a, S. 75-86.
- Pinkwart, N., Alevén, V., Ashley, K. und Lynch, C. (2006b): Toward legal argument instruction with graph grammars and collaborative filtering techniques; in: M. Ikeda, K. D. Ashley und T.-W. Chan (Hrsg.): *Intelligent Tutoring Systems, LNCS 4053*; Springer, Berlin, Heidelberg; 2006b, S. 227-236.
- Pinkwart, N., Alevén, V., Lynch, C. und Ashley, K. (2008): Graph Grammars: An ITS Technology for Diagram Representations; in: *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference*, W. David und C. H. Lane (Hrsg.), Coconut Grove, Florida, USA, 2008, Paper 15.
- Prakken, H. (2010): On the nature of argument schemes; in: C. Reed und C. Tindale (Hrsg.): *Dialectics, Dialogue and Argumentation. An Examination of Douglas Walton's Theories of Reasoning and Argument*; College Publications, London; 2010, S. 167-185.

- Ravi, S. und Kim, J. (2007): Profiling Student Interactions in Threaded Discussions with Speech Act Classifiers; in: R. Luckin, K. R. Koedinger und J. Greer (Hrsg.): Proceedings of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work; IOS Press, Amsterdam, The Netherlands; 2007, S. 357-364.
- Reed, C. (2005): Preliminary results from an argument corpus; in: E. M. Bermudez und L. R. Miyares (Hrsg.): Proceedings of the IX. Symposium on Social Communication, Santiago de Cuba, Cuba, 2005, S. 576-580.
- Reed, C., Mochales Palau, R., Rowe, G. und Moens, M.-F. (2008): Language resources for studying argument; in: Proceedings of the 6th conference on language resources and evaluation (LREC 2008), Marrakech, Morocco, 2008, S. 91-100.
- Rodríguez, M. d. B., Gómez-Hidalgo, J. M. und Díaz-Agudo, B. (1997): Using Wordnet to Complement Training Information in Text Categorization; in: N. Nicolov und R. Mitkov (Hrsg.): Recent Advances in Natural Language Processing II; John Benjamins, Amsterdam; 1997, S. 353-364.
- Roesner, D. F. und Laubsch, J. (1982): Formalization Of Argumentation Structures In Newspaper Texts; in: 9th International Conference on Computational Linguistics (COLING 1982), Prague, 1982, S. 325-330.
- Rooney, N., Wang, H. und Browne, F. (2012): Applying Kernel Methods to Argumentation Mining; in: *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, G. M. Youngblood und P. M. McCarthy (Hrsg.), Association for the Advancement of Artificial Intelligence, Marco Island, Florida, USA, 2012, S. 272-275.
- Scheuer, O., Loll, F., Pinkwart, N. und McLaren, B. M. (2010): Computer-Supported Argumentation: A Review of the State of the Art; *International Journal of Computer-Supported Collaborative Learning*, 5. Jg.; 2010, H. 1; S. 43-102.
- Sombekke, J., van Engers, T. und Prakken, H. (2007): Argumentation Structures in Legal Dossiers; in: *Proceedings of the 11th International Conference on Artificial Intelligence and Law (ICAIL'07)*, R. Winkels (Hrsg.), ACM (Association for Computing Machinery), Stanford, California, USA, 2007, S. 277-281.
- Stranieri, A. und Zeleznikow, J. (1999): A survey of argumentation structures for intelligent decision support; in: *Proceedings of the Fifth International Conference of the International Society for Decision Support Systems (ISDSS'99)*, Melbourne, 1999.
- Toulmin, S. (1975): *Der Gebrauch von Argumenten*; Scriptor Verlag, Kronberg; 1975.
- van Eemeren, F. H. und Grootendorst, R. (2004): *A systematic Theory of Argumentation - The pragma-dialectical approach*; Cambridge University Press, Cambridge; 2004.
- Verheij, B. (1996): *Rules, reasons, arguments. Formal studies of argumentation and defeat*; Universiteit Maastricht 1996.
- Verheij, B. (1999): Automated argument assistance for lawyers; in: Proceedings of the Seventh International Conference on Artificial Intelligence and Law (ICAIL-99), ACM, Oslo, Norway, 1999, S. 43-52.
- Verheij, B. (2003): Artificial argument assistants for defeasible argumentation; *Artificial Intelligence*, 150. Jg.; 2003, H. 1-2; S. 291-324.

- Vincenzo, P. und Delmonte, R. (2011): Automatic argumentative analysis for interaction mining; *Argument & Computation*, 2. Jg.; 2011, H. 2-3; S. 77-106.
- Walker, V. R., Carie, N., DeWitt, C. C. und Lesh, E. (2011): A framework for the extraction and modeling of fact-finding reasoning from legal decisions: lessons from the Vaccine/Injury Project Corpus; *Artificial Intelligence and Law*, 19. Jg.; 2011, H. 4; S. 291-331.
- Walton, D. N. (1996): *Argumentation Schemes for Presumptive Reasoning*; Lawrence Erlbaum Associates, Mahwah, NJ; 1996.
- Weinberger, A. und Fischer, F. (2006): A framework to analyze argumentative knowledge construction in computer-supported collaborative learning; *Computers & Education*, 46. Jg.; 2006, H. 1; S. 71 - 95.
- Wyner, A. und Milward, D. (2008): Legal text-mining using linguamatics' I2E; in: Presentation at Workshop on Natural Language Engineering of Legal Argumentation as part of JURIX 2008, Florence, Italy, 2008.
- Wyner, A., Mochales-Palau, R., Moens, M.-F. und Milward, D. (2010): Approaches to Text Mining Arguments from Legal Cases; in: E. Francesconi, S. Montemagni, W. Peters und D. Tiscornia (Hrsg.): *Semantic Processing of Legal Texts*, LNCS 6036; Springer, Berlin; 2010, S. 60-79.
- Yang, Y. und Pedersen, J. O. (1997): A Comparative Study on Feature Selection in Text Categorization; in: *Proceedings of the 14th International Conference on Machine Learning*, Nashville, Tennessee, USA, 1997, S. 412-420.

Die Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) erscheinen in unregelmäßigen Zeitabständen.

- Heft 199:** Tom Thaler, Sharam Dadashnia, Andreas Sonntag, Peter Fettke, Peter Loos: The IWi Process Model Corpus, October 2015
- Heft 198:** Constantin Houy, Tim Niesen, Johannes Frank, Peter Fettke, Peter Loos: Zur Verwendung von Theorien in der Wirtschaftsinformatik – Eine quantitative Literaturanalyse, Dezember 2014. Heft 198 wurde auch in englischer Sprache herausgegeben:  
Constantin Houy, Tim Niesen, Johannes Frank, Peter Fettke, Peter Loos: On the Usage of Theories in the Field of *Wirtschaftsinformatik* – A Quantitative Literature Analysis, December 2014
- Heft 197:** Peter Fettke, Constantin Houy, Philipp Leupoldt, Peter Loos: Discourse-Oriented in Conceptual Model Quality Research - Foundations, Procedure Model and Applications, January 2014
- Heft 196:** Constantin Houy, Peter Fettke, Peter Loos: Understanding understandability of conceptual models. What are we actually talking about? – Supplement, Juni 2013
- Heft 195:** Constantin Houy, Markus Reiter, Peter Fettke, Peter Loos: Prozessorientierter Web-2.0-basierter integrierter Telekommunikationsservice (PROWIT) - Anforderungserhebung, Konzepte, Implementierung und Evaluation, Oktober 2012
- Heft 194:** Isabelle, Aubertin, Constantin Houy, Peter Fettke, Peter Loos: Stand der Lehrbuchliteratur zum Geschäftsprozessmanagement - Eine quantitative Analyse, Mai 2012
- Heft 193:** Silke Balzert, Thomas Kleinert, Peter Fettke, Peter Loos: Vorgehensmodelle im Geschäftsprozessmanagement - Operationalisierbarkeit von Methoden zur Prozesserhebung, November 2011
- Heft 192:** Constantin Houy, Peter Fettke, Peter Loos: Einsatzpotentiale von Enterprise-2.0-Anwendungen - Darstellung des State-of-the-Art auf Basis eines Literaturreviews, November 2010
- Heft 191:** Peter Fettke, Constantin Houy, Peter Loos: Zur Bedeutung von Gestaltungswissen für die gestaltungsorientierte Wirtschaftsinformatik – Ergänzende Überlegungen und weitere Anwendungsbeispiele, November 2010. Heft 191 wurde auch in englischer Sprache herausgegeben:  
Peter Fettke, Constantin Houy, Peter Loos: On the Relevance of Design Knowledge for Design-Oriented Business and Information Systems Engineering – Supplemental Considerations and further Application Examples, November 2010
- Heft 190:** Oliver Thomas, Thorsten Dollmann: Entscheidungsunterstützung auf Basis einer Fuzzy-Regelbasierten Prozessmodellierung: Eine fallbasierte Betrachtung anhand der Kapazitätsplanung, Juni 2008
- Heft 189:** Oliver Thomas, Katrina Leyking, Florian Dreifus, Michael Fellmann, Peter Loos: Serviceorientierte Architekturen: Gestaltung, Konfiguration und Ausführung von Geschäftsprozessen, Januar 2007
- Heft 188:** Christine Daun, Thomas Theling, Peter Loos: ERPeL - Blended Learning in der ERP-Lehre, Dezember 2006
- Heft 187:** Oliver Thomas: Das Referenzmodellverständnis in der Wirtschaftsinformatik: Historie, Literaturanalyse und Begriffsexplikation, Januar 2006
- Heft 186:** Oliver Thomas, Bettina Kaffai, Peter Loos: Referenzgeschäftsprozesse des Event-Managements, November 2005
- Heft 185:** Thomas Matheis, Dirk Werth: Konzeption und Potenzial eines kollaborativen Data-Warehouse-Systems, Juni 2005
- Heft 184:** Oliver Thomas: Das Modellverständnis in der Wirtschaftsinformatik: Historie, Literaturanalyse und Begriffsexplikation, Mai 2005

Frühere Hefte sind verzeichnet unter: [www.iwi.uni-sb.de/publikationen/iwi-hefte.html](http://www.iwi.uni-sb.de/publikationen/iwi-hefte.html)



Unter der wissenschaftlichen Leitung von Professor Dr. Peter Loos sind am Institut für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) mehr als 60 Mitarbeiter im Bereich der anwendungsnahen Forschung beschäftigt. Seit das Institut vor über 30 Jahren durch Prof. Dr. Dr. h.c. mult. August-Wilhelm Scheer gegründet wurde, wird hier in Forschung und Lehre das Informations- und Prozessmanagement in Industrie, Dienstleistung und Verwaltung vorangetrieben. Ein besonderer Anspruch liegt dabei auf dem Technologietransfer von der Wissenschaft in die Praxis.

Die interdisziplinäre Struktur der Mitarbeiter und Forschungsprojekte fördert zusätzlich den Austausch von Spezialwissen aus unterschiedlichen Fachbereichen. Die Zusammenarbeit mit kleinen und mittelständischen Unternehmen (KMU) hat einen bedeutenden Einfluss auf die angewandte Forschungsarbeit – wie auch Projekte im Bildungs- und Wissensmanagement eine wichtige Rolle spielen. So werden in virtuellen Lernwelten traditionelle Lehrformen revolutioniert. Das Institut für Wirtschaftsinformatik berücksichtigt den steigenden Anteil an Dienstleistungen in der Wirtschaft durch die Unterstützung servicespezifischer Geschäftsprozesse mit innovativen Informationstechnologien und fortschrittlichen Organisationskonzepten. Zentrale Themen sind Service Engineering, Referenzmodelle für die öffentliche Verwaltung sowie die Vernetzung von Industrie, Dienstleistung und Verwaltung.

Am Standort im DFKI auf dem Campus der Universität des Saarlandes werden neben den Lehrtätigkeiten im Fach Wirtschaftsinformatik die Erforschung zukünftiger Bildungsformen durch neue Technologien wie Internet und Virtual Reality vorangetrieben. Hier führt das Institut Kooperationsprojekte mit nationalen und internationalen Partnern durch: Lernen und Lehren werden neu gestaltet; Medienkompetenz und lebenslanges Lernen werden Realität. Zudem beschäftigen sich die Mitarbeiterinnen und Mitarbeiter mit dem Einsatz moderner Informationstechniken in der Industrie. In Kooperation mit industrieorientierten Lehrstühlen der technischen Fakultäten saarländischer Hochschulen werden Forschungsprojekte durchgeführt. Hauptaufgabengebiete sind die Modellierung und Simulation industrieller Geschäftsprozesse, Workflow- und Groupware-Systeme sowie Konzepte für die virtuelle Fabrik.

Universitätscampus D 3<sub>2</sub>  
D-66123 Saarbrücken  
Tel.: +49 (0) 681 / 85775 - 3106  
Fax: +49 (0) 681 / 85775 - 3696  
[iwi@iwi.uni-sb.de](mailto:iwi@iwi.uni-sb.de)  
[www.iwi.uni-sb.de](http://www.iwi.uni-sb.de)  
[www.dfki.de](http://www.dfki.de)