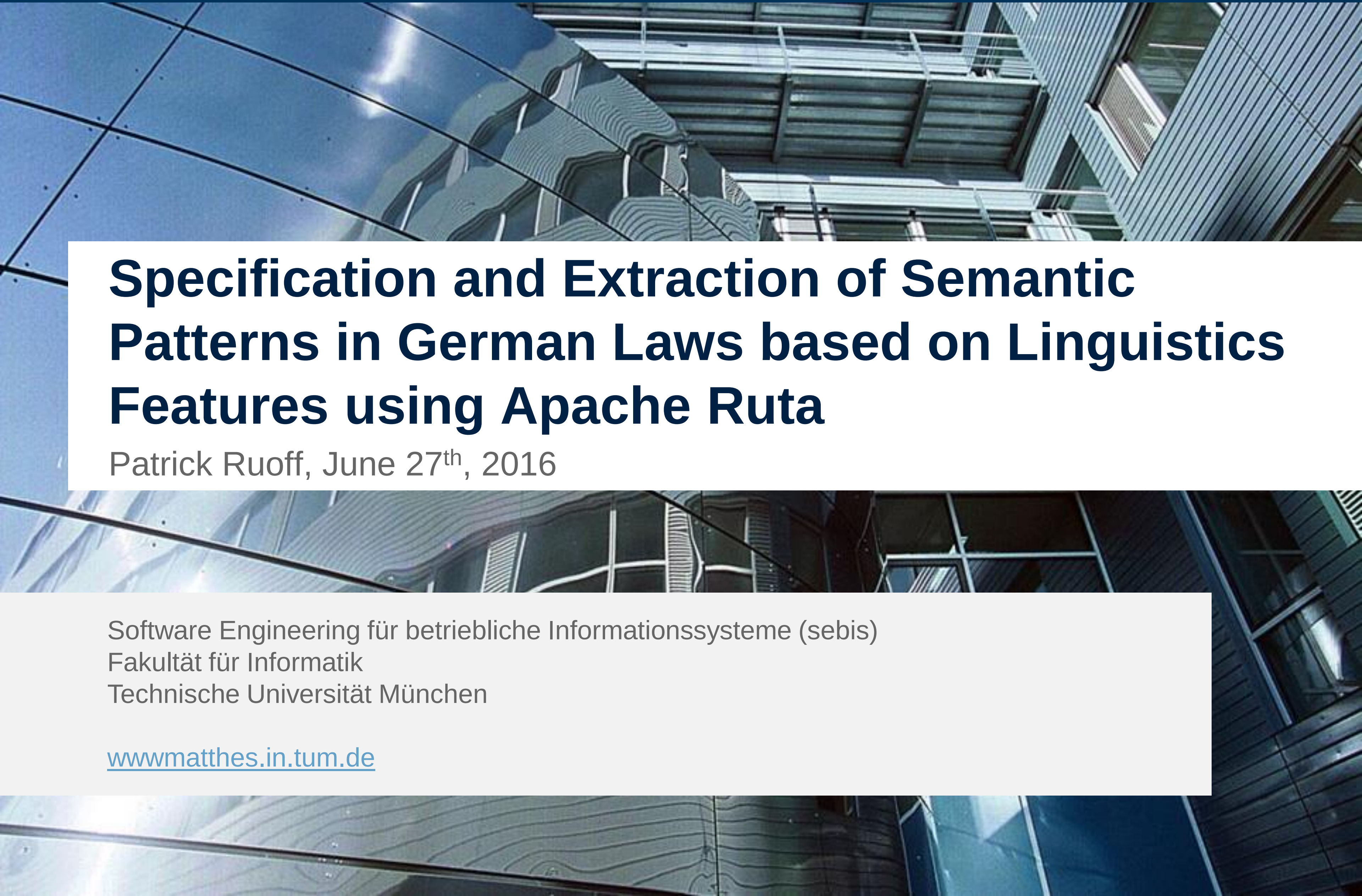
Technische Universität München



Specification and Extraction of Semantic Patterns in German Laws based on Linguistics Features using Apache Ruta

Patrick Ruoff, June 27th, 2016

Software Engineering für betriebliche Informationssysteme (sebis)
Fakultät für Informatik
Technische Universität München

wwwmatthes.in.tum.de

Agenda

1. Introduction

2. Problem Relevance

3. Text Mining
   - In General
   - Specific

4. Working Process

5. Pattern Recognition - Example

6. Roadmap

Introduction - Lexia

- Processes of Legal Experts (Scientists and Lawyers) are…
  - ... time-intensive
  - ... knowledge-intensive
  - ... data-intensive

- Legal Data Science is becoming more and more attractive, because
  - ... process time and memory space are cheap
  - ... algorithms can process data fast and accurate

- In order to achieve highest accuracy, algorithms and data models need an adaption to the domain
  - German legal texts (laws, contracts, etc.)
  - Data model tailored to legislative data

→ There are still more Patterns that are not yet implemented

Problem Relevance

1. Maat and Winkels (2010)
   - Classification of norms regarding linguistic structures
   - Regular Expressions
   - Limitation: no consideration of linguistic properties, such as nouns, etc.

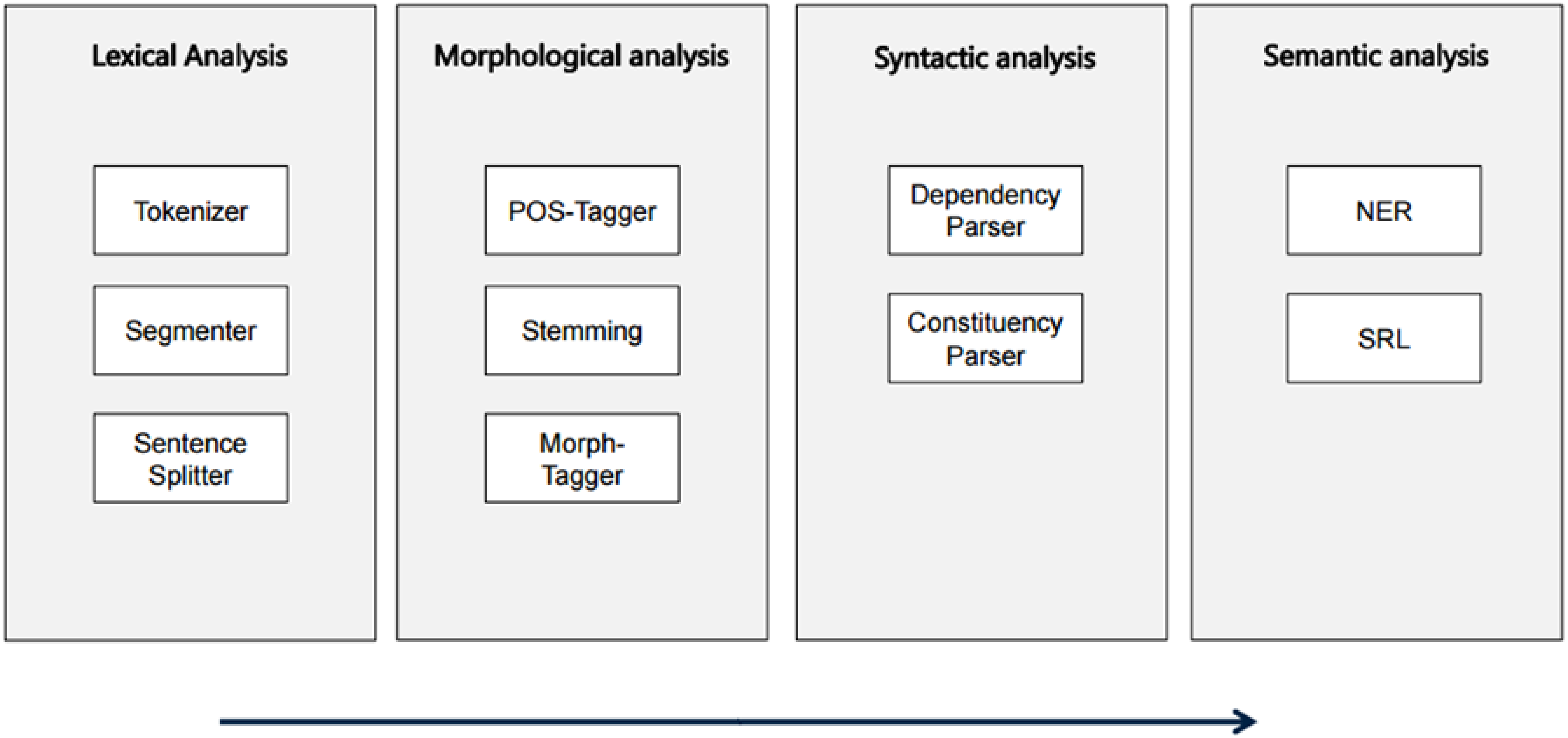
2. Bommarito and Katz (2014)
   - Analysis of semantic and structural properties

3. Grabmair et al. (2015)
   - Using Apache UIMA for legal text analysis (LUIMA)

Short summary

- Data analysis is well established in legal informatics
- Adaption to domain is crucial to achieve highest accuracy …
  - ... data model
  - ... algorithms
- Reusable code and implementations to avoid re-inventions

Text Mining – In General



Lexical Analysis

Tokenizer

Segmenter

Sentence Splitter

Morphological analysis

POS-Tagger

Stemming

Morph-Tagger

Syntactic analysis

Dependency Parser

Constituency Parser

Semantic analysis

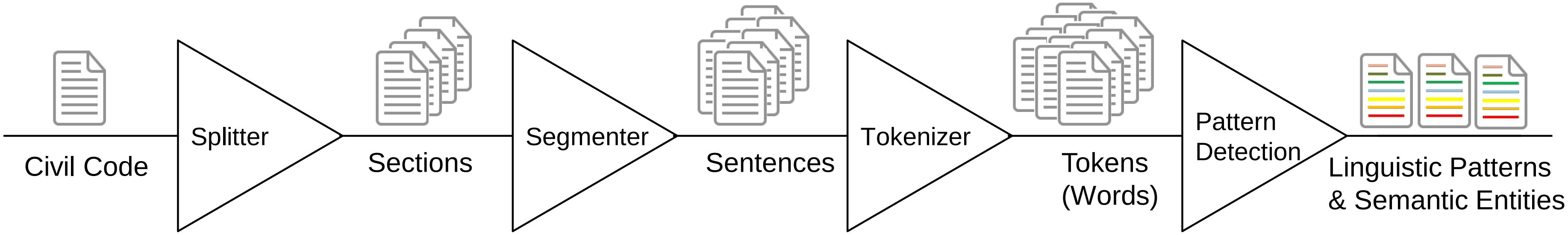
NER

SRL

Text Mining - Specific

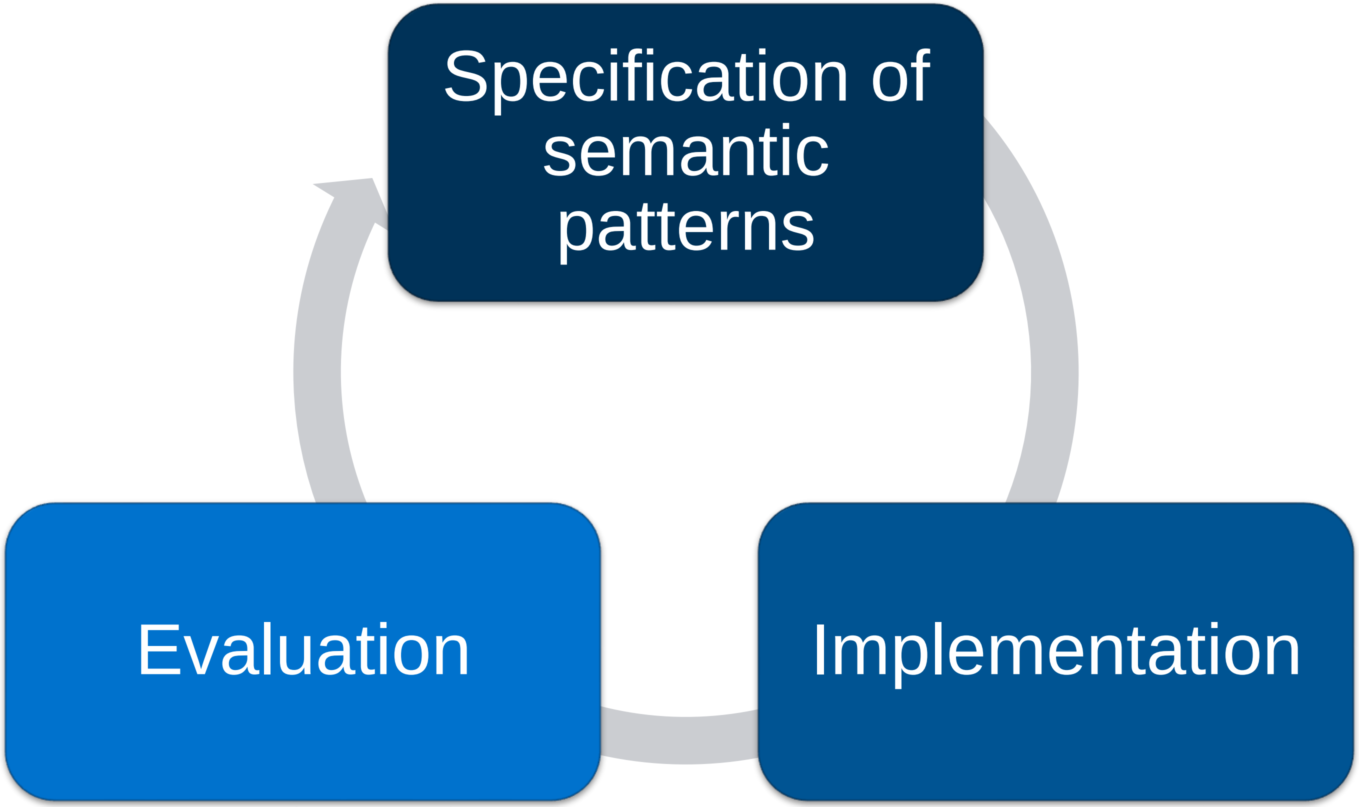
UIMA (Unstructured Information Management Architecture)

- A common software architecture for text mining/processing
  - Alternatives: GATE, NLTK, etc.
  - Base line for IBM Watson
- Pipes & Filters architecture



- Thread-safe (usage in a web application with multiple users/requests)
- Apache Ruta for complex pattern specification engine
  - Analogy: Jape grammar (GATE)

Working Process


Specification of semantic patterns

Implementation

Evaluation

Working Process

Specification of semantic patterns

- In Cooperation with Konrad Heßler from the Juristic Faculty of the LMU
- Introduction into juristic work needed

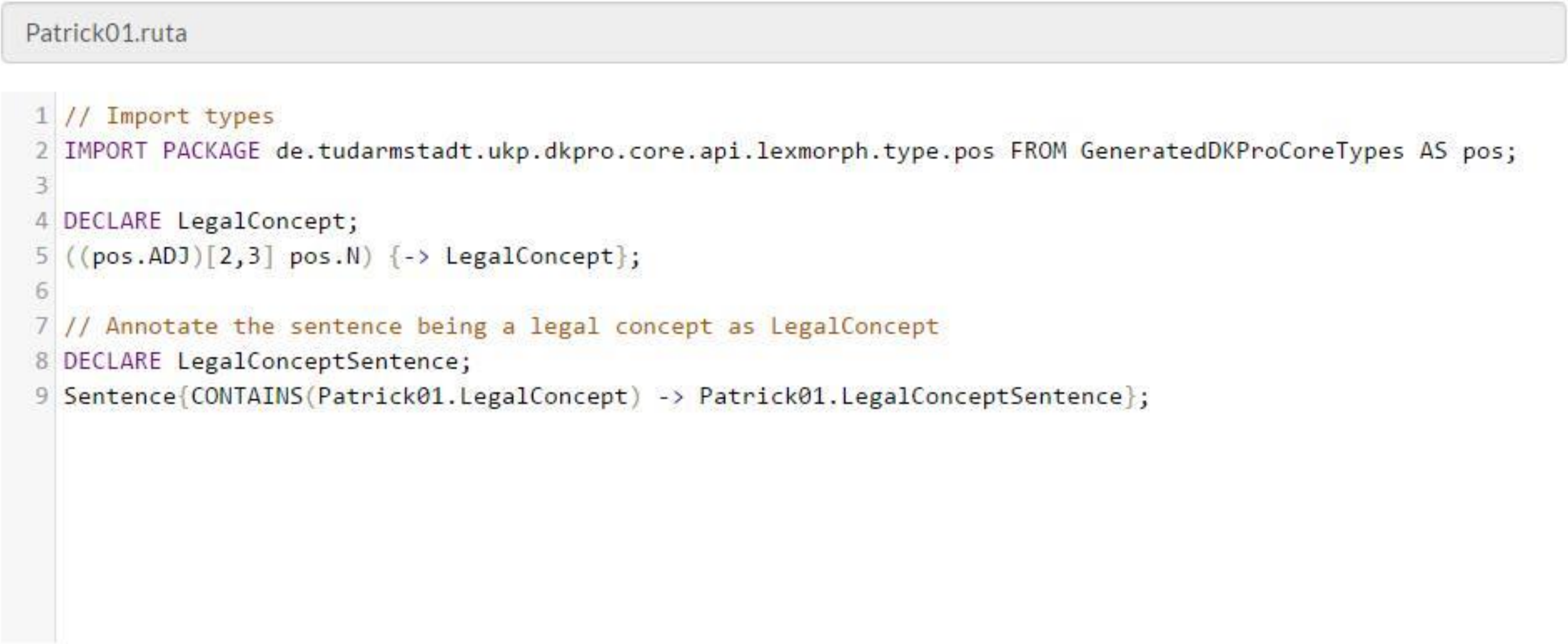
Implementation

- Ruta scripts
- Executed on Lexia

Evaluation

- Testing on German laws
- Comparison with handmade Classifications

Code Example



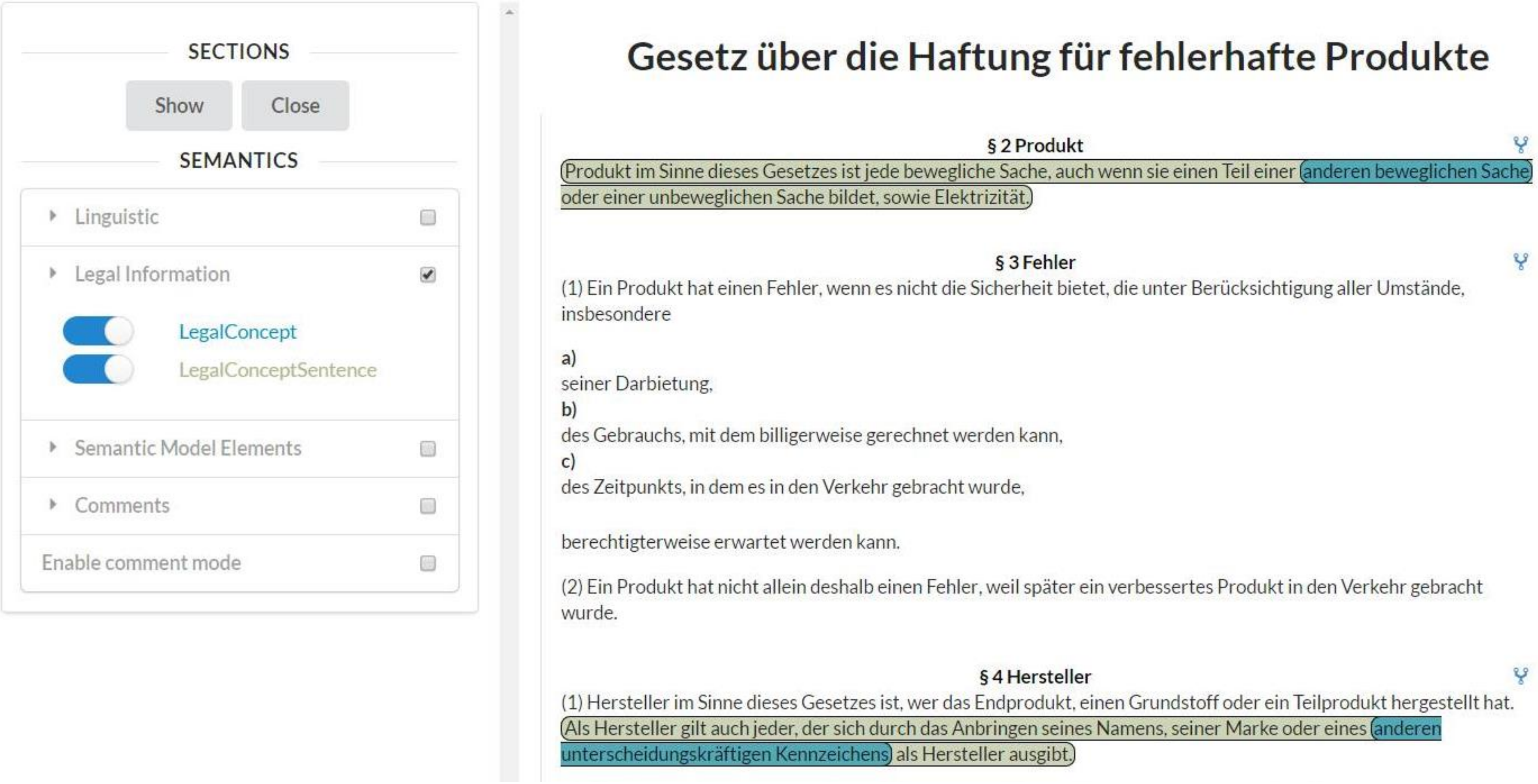
Patrick01.ruta

```
// Import types
IMPORT PACKAGE de.tudarmstadt.ukp.dkpro.core.api.lexmorph.type.pos FROM GeneratedDKProCoreTypes AS pos;

DECLARE LegalConcept;
((pos.ADJ)[2,3] pos.N) {-> LegalConcept};

// Annotate the sentence being a legal concept as LegalConcept
DECLARE LegalConceptSentence;
Sentence{CONTAINS(Patrick01.LegalConcept) -> Patrick01.LegalConceptSentence};
```

Code Example



Gesetz über die Haftung für fehlerhafte Produkte

§ 2 Produkt

Produkt im Sinne dieses Gesetzes ist jede bewegliche Sache, auch wenn sie einen Teil einer anderen beweglichen Sache oder einer unbeweglichen Sache bildet, sowie Elektrizität.

§ 3 Fehler

(1) Ein Produkt hat einen Fehler, wenn es nicht die Sicherheit bietet, die unter Berücksichtigung aller Umstände, insbesondere

a)
seiner Darbietung,
b)
des Gebrauchs, mit dem billigerweise gerechnet werden kann,
c)
des Zeitpunkts, in dem es in den Verkehr gebracht wurde,

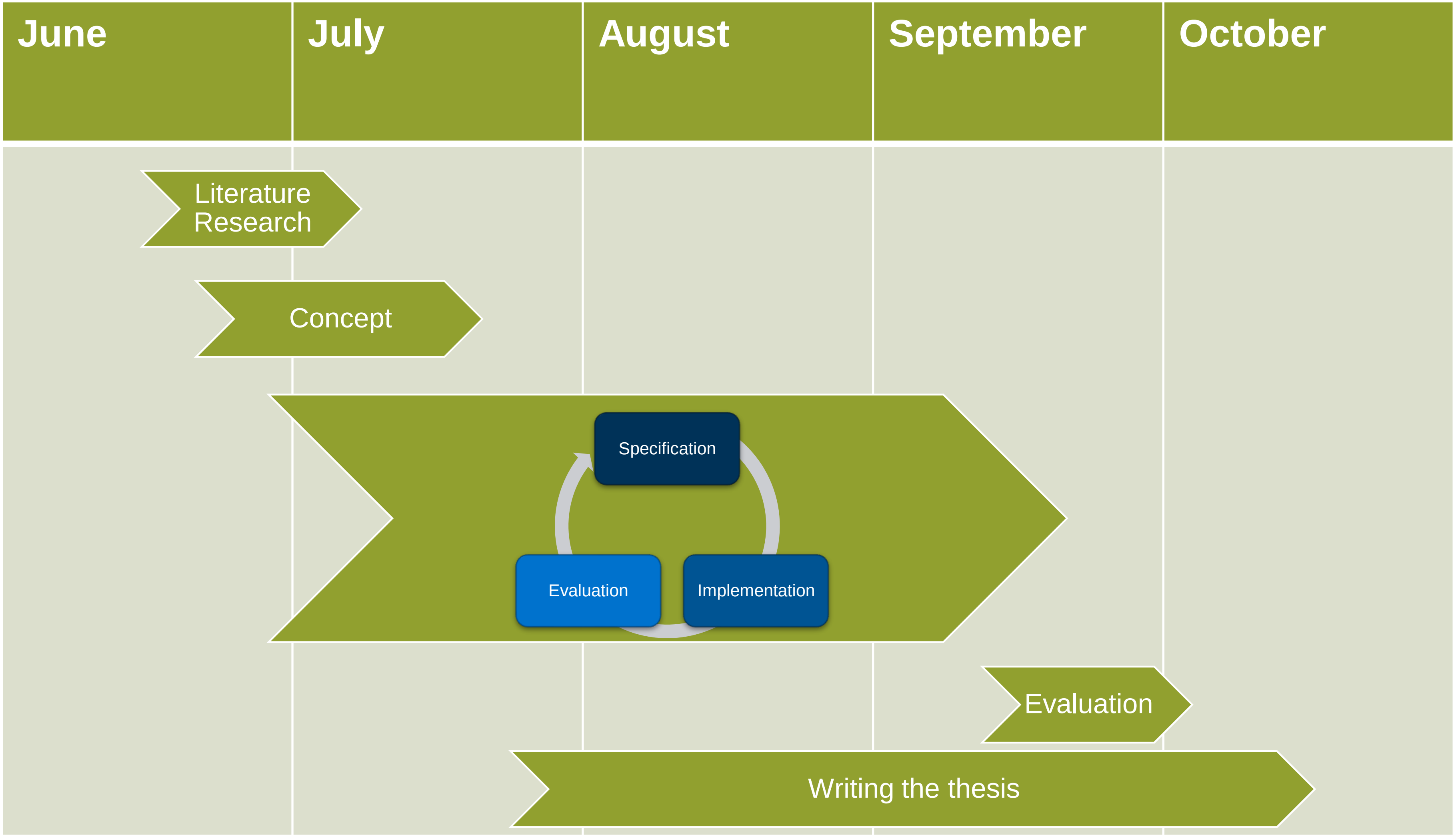
berechtigterweise erwartet werden kann.

(2) Ein Produkt hat nicht allein deshalb einen Fehler, weil später ein verbessertes Produkt in den Verkehr gebracht wurde.

§ 4 Hersteller

(1) Hersteller im Sinne dieses Gesetzes ist, wer das Endprodukt, einen Grundstoff oder ein Teilprodukt hergestellt hat. Als Hersteller gilt auch jeder, der sich durch das Anbringen seines Namens, seiner Marke oder eines anderen unterscheidungskräftigen Kennzeichens als Hersteller ausgibt.

Roadmap



June

July

August

September

October

Literature Research

Concept

Specification

Implementation

Evaluation

Evaluation

Writing the thesis

Thank you for your attention!

Patrick Ruoff

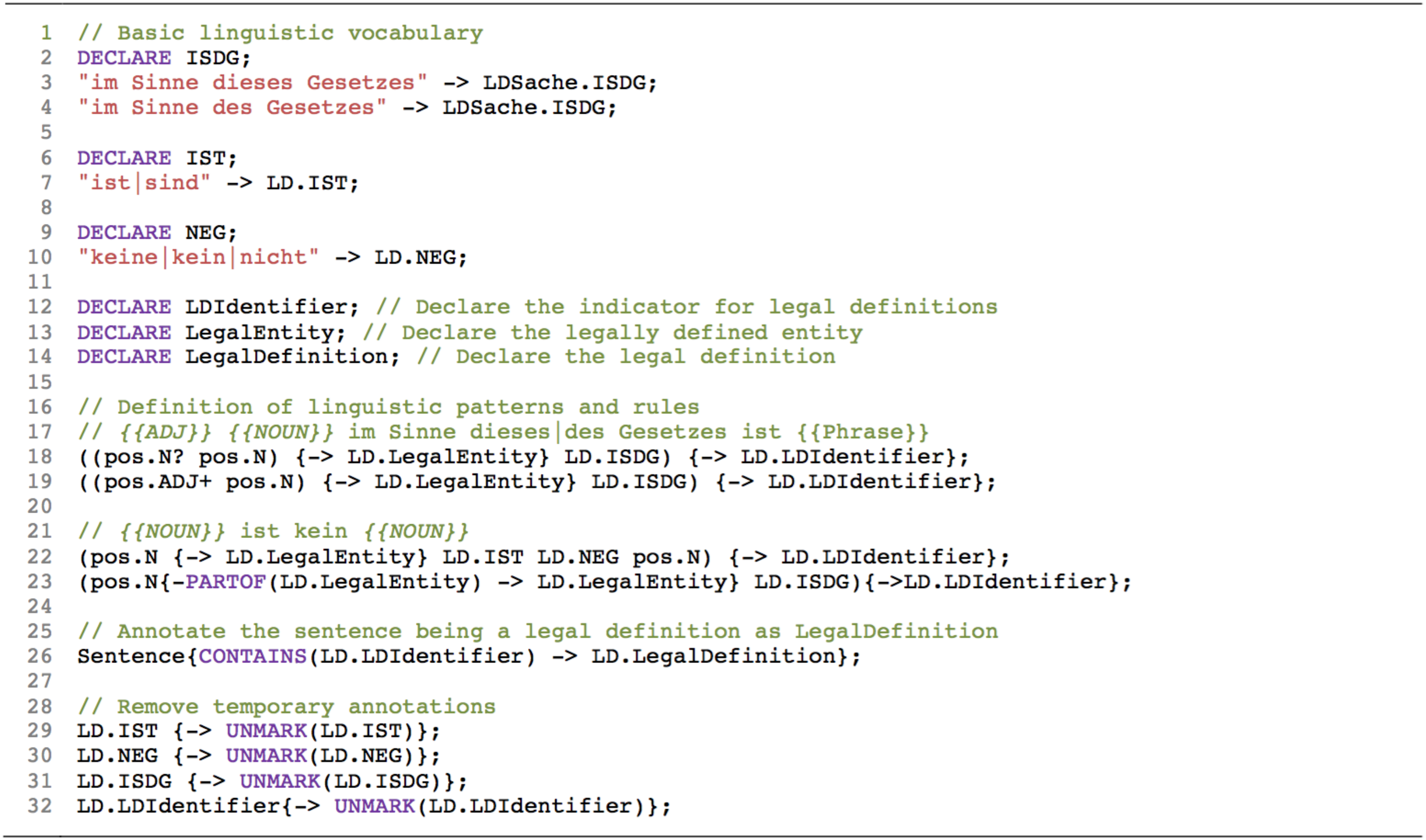
Technische Universität München
Department of Informatics
Chair of Software Engineering for
Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289.17124
Fax +49.89.289.17136

ga54kuc@mytum.de
wwwmatthes.in.tum.de

Ruta Script
Legal Definition (excerpt)



```
// Basic linguistic vocabulary
DECLARE ISDG;
"im Sinne dieses Gesetzes" -> LDSache.ISDG;
"im Sinne des Gesetzes" -> LDSache.ISDG;

DECLARE IST;
"ist|sind" -> LD.IST;

DECLARE NEG;
"keine|kein|nicht" -> LD.NEG;

DECLARE LDIdentifier; // Declare the indicator for legal definitions
DECLARE LegalEntity; // Declare the legally defined entity
DECLARE LegalDefinition; // Declare the legal definition

// Definition of linguistic patterns and rules
// {{ADJ}} {{NOUN}} im Sinne dieses|des Gesetzes ist {{Phrase}}
((pos.N? pos.N) {-> LD.LegalEntity} LD.ISDG) {-> LD.LDIdentifier};
((pos.ADJ+ pos.N) {-> LD.LegalEntity} LD.ISDG) {-> LD.LDIdentifier};

// {{NOUN}} ist kein {{NOUN}}
(pos.N {-> LD.LegalEntity} LD.IST LD.NEG pos.N) {-> LD.LDIdentifier};
(pos.N{-PARTOF(LD.LegalEntity) -> LD.LegalEntity} LD.ISDG){->LD.LDIdentifier};

// Annotate the sentence being a legal definition as LegalDefinition
Sentence{CONTAINS(LD.LDIdentifier) -> LD.LegalDefinition};

// Remove temporary annotations
LD.IST {-> UNMARK(LD.IST)};
LD.NEG {-> UNMARK(LD.NEG)};
LD.ISDG {-> UNMARK(LD.ISDG)};
LD.LDIdentifier{-> UNMARK(LD.LDIdentifier)};
```

Listing 1: Linguistic pattern descriptions (LD.ruta) for the semantic entity Legal Definition using Apache Ruta