BA: Structured Extraction of Terms and Conditions from German and English Online Shops

Tobias Schamel, 26th Apr 2021, KickOff Presentation (Advanced Seminar)

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

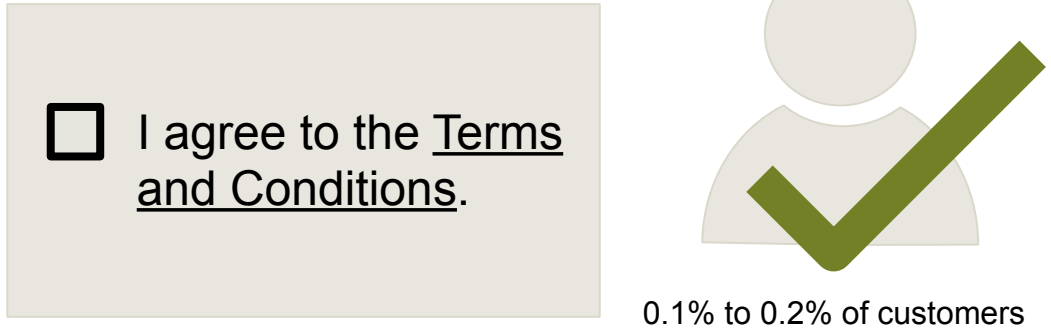
Outline

1. Motivation
2. Research Question
   - Context in AGB Check
3. Solution Approaches
4. Timeline

Motivation

Whom of you did ever read the Terms and Conditions when buying products online?

- no standard solutions for parts of the NLP pipe (structured extraction from websites)
- structured extraction is needed in legal documents to allow the drawing of conclusions later in the pipe
  - no existing solutions for structured & hierarchical extraction of T&C



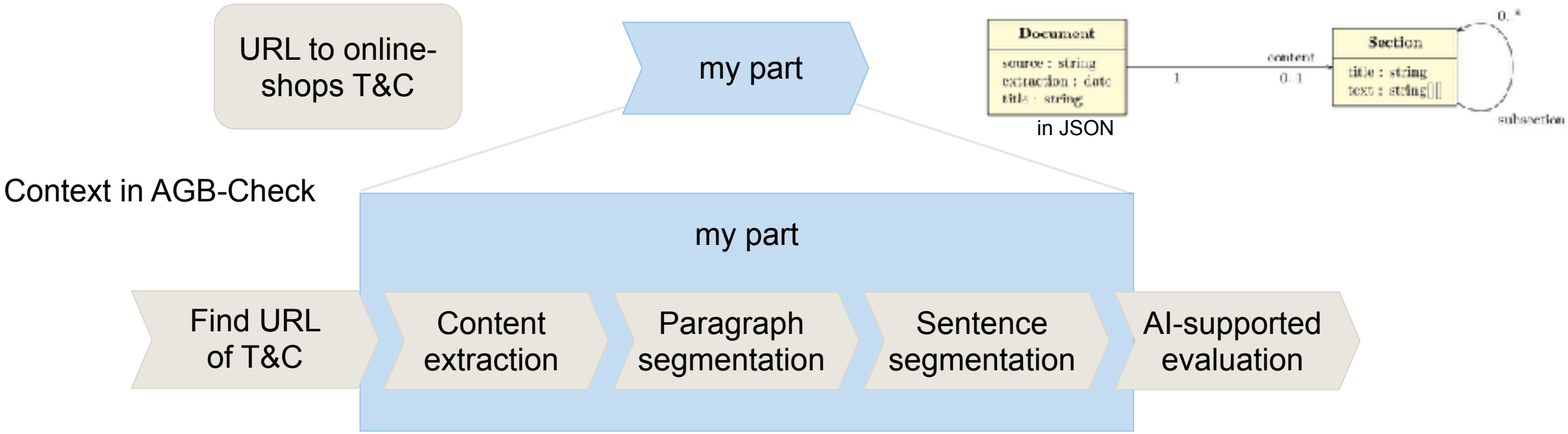
0.1% to 0.2% of customers read T&C in online shops

Outline

1. Motivation
2. Research Question
   - Context in AGB Check
3. Solution Approaches
4. Timeline

Research Question (Context in AGB-Check)

- RQ1: What are special requirements for the extraction of contracts in comparison with *regular* content?
- RQ2: How to extract the relevant parts from contracts in raw html?
- RQ3: How to handle certain characteristic elements (avoid classification as main content)?
- RQ4: How to extract structure and hierarchy of paragraphs, (sub-)titles and related clauses?
  - Which existing HTML and CSS properties provide useful information and can be used to structure the content?



For more information visit AGB-Check (http://wwwmatthes.in.tum.de)

Research Question (Context in AGB-Check)

- Example

Terms and Conditions of Aldi UK, 210325

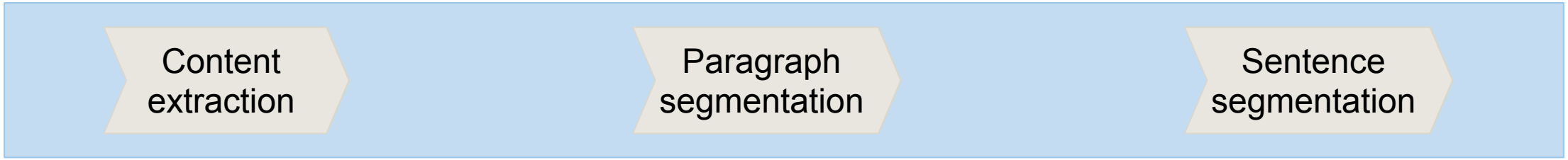

```
{
    "subsections": [
        {
            "subsections": [],
            "text": [
                [
                    "our",
                    "Privacy",
                    "Notice",
                    ",",
                    "which",
                    "sets",
                    "out",
                    "the",
                    "terms",
                    "on",
                    "which",
                    "we",
                    "process",
                    "any",
                    "personal",
                    "data",
                    "we",
                    "collect",
                    "from",
                    "you",
                    ",",
                    "or",
                    "that",
                    "you",
                    "provide",
                    "to",
                    "us",
                    "."
                ],
                [
                    "You",
                    "consent",
                    "to",
                    "such",
                    "processing",
                    "by",
                    "using",
                    "our",
                    "Site",
                    "/",
                    "App",
                    ";"
                ]
            ],
            "title": ""
        },
        {
            "subsections": [],
            "text": [
                [
                    "our",
                    "Content",
                    "Management",
                    "Policy",
                    ",",
                    "which",
                    "governs",
                    "how",
                    "we",
                    "deal",
                    "with",
                    "content",
                    "which",
                    "you",
                    "submit",
                    "to",
                    "us",
                    ";",
                    "and"
                ]
            ],
            "title": ""
        },
        {
            "subsections": [],
```


excerpt of parsed T&C

Outline

1. Motivation
2. Research Question
   - Context in AGB Check
3. Solution Approaches
4. Timeline

Solution Approaches



Selenium with driver to read entire page source code (+style)

Removing irrelevant content (based on tags, amount of text)

trafilatura, jusText, boilerpipe

Segmenting paragraphs (based on font-size [style], containers, enumerations)

SoMaJo shows good results for sentence segmentation & tokenization

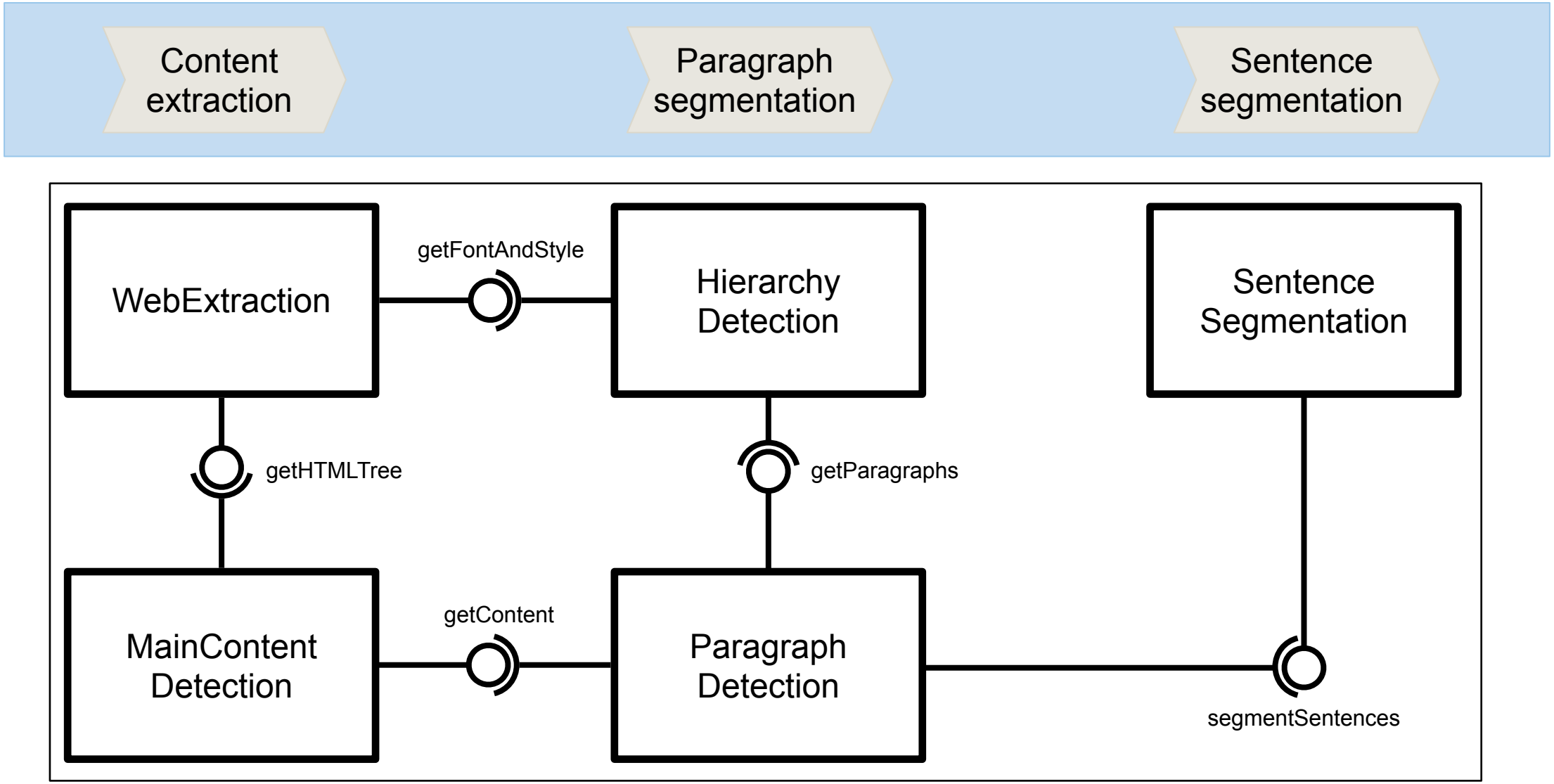
NLTK, spaCy

- exist. solutions
- approaches
- exist. libraries

[SoMaJo]Proisl, T., Uhrig, P.: SoMaJo: State-of-the-art tokenization for German web and social media texts

[github]boilerpipe (Canola & Article); [github]trafilatura; [github]jusText; [github]selenium; [github]NLTK; [github]spaCy

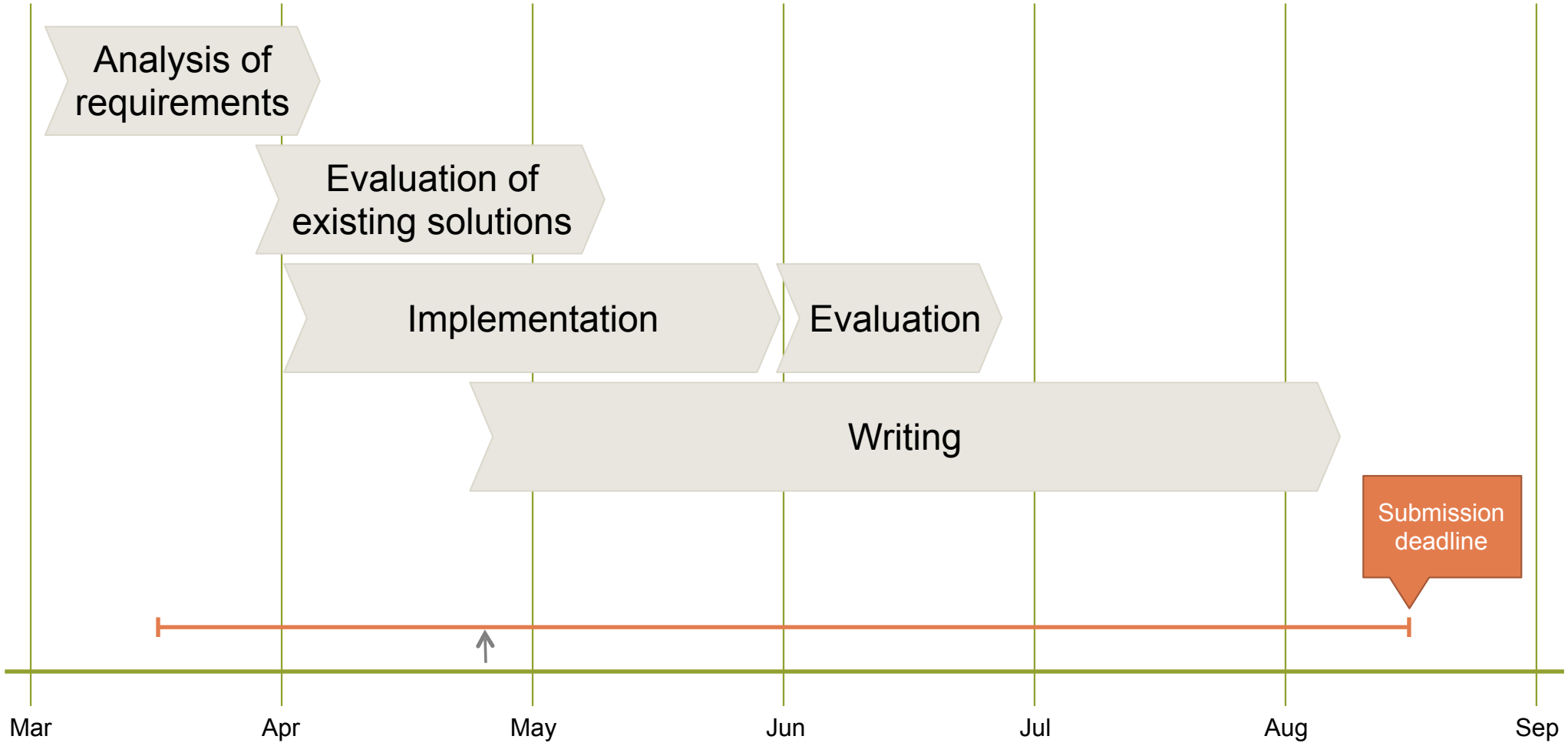
Solution Approaches



Outline

1. Motivation
2. Research Question
   - Context in AGB Check
3. Solution Approaches
4. Timeline

Timeline



Tobias Schamel

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

Boltzmannstraße 3
85748 Garching bei München

wwwmatthes.in.tum.de

tobias.schamel@tum.de