Chair of Information Systems and Business Process Management (i17)
Department of Computer Science
TUM School of Computation, Information and Technology
Technical University of Munich

Bachelor’s Thesis in Information Systems

Automated Change Identification and Classification for Legal Documents and Their Amendments

Jacob Fehn

1. Introduction

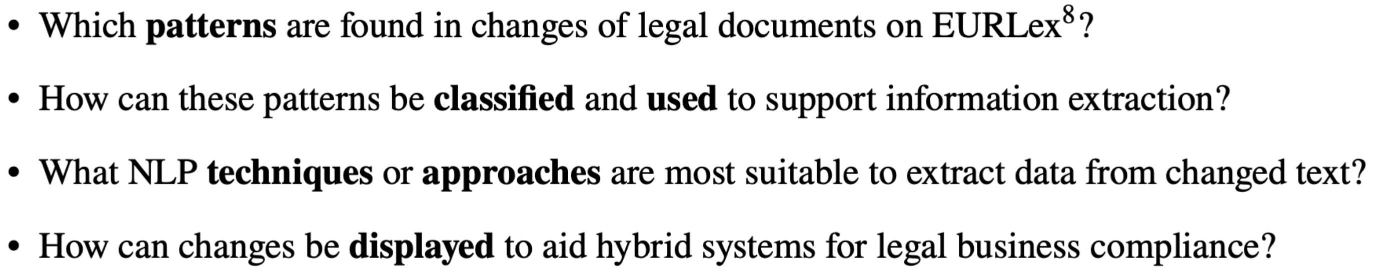
- 1. 1. Motivation
- 1. 2. Research Questions
- 1. 3. Research Methodology
- 1. 4. Structure

1. Introduction

- 1. 1. **Motivation** *(p. 8 - 10)*
  - Business Process Compliance (BPC)
    - Non-compliance is expensive!
  - Changes of Legal Documents
    - Only EU Law (EURLex) exceed 38 changes per year!
- 1. 2. Research Questions
- 1. 3. Research Methodology
- 1. 4. Structure

1. Introduction

- 1. 1. Motivation
- 1. 2. **Research Questions** *(p. 11)*



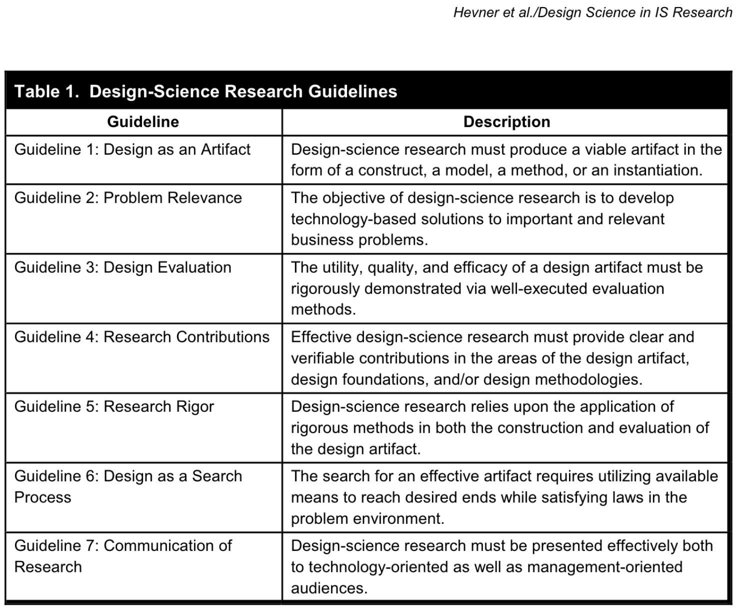
  - Which **patterns** are found in changes of legal documents on EURLex[8]?
  - How can these patterns be **classified** and **used** to support information extraction?
  - What NLP **techniques** or **approaches** are most suitable to extract data from changed text?
  - How can changes be **displayed** to aid hybrid systems for legal business compliance?
- 1. 3. Research Methodology
- 1. 4. Structure

1. Introduction

- 1. 1. Motivation
- 1. 2. Research Questions
- 1. 3. **Research Methodology** *(p. 11)*
  - Artifacts: set of data (*instantiation*), classification (*model*), web service (*method*)
  - Hevner’s guideline for research
  - Hevner’s research framework
- 1. 4. Structure

1. Introduction

- 1. 3. **Research Methodology** *(p. 11)*
  - Artifacts: set of data (*instantiation*), classification (*model*), web service (*method*)
  - Hevner's guideline for research:



Hevner et al./Design Science in IS Research

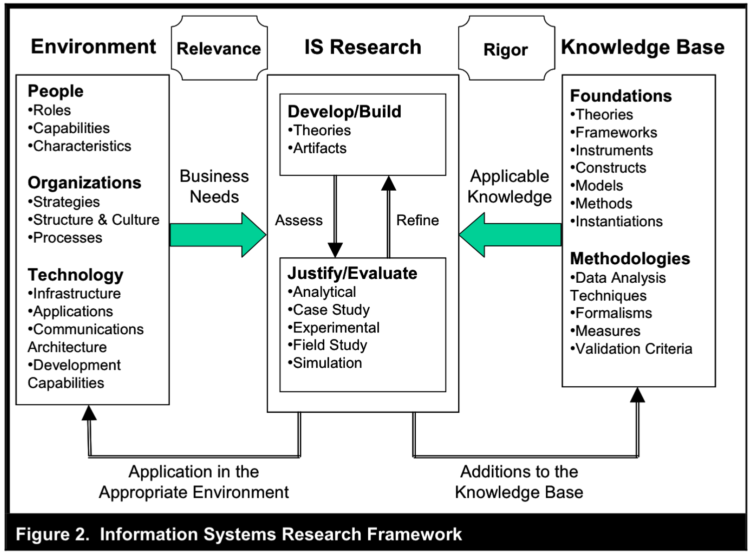
Table 1. Design-Science Research Guidelines

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

1. Introduction

- 1. 3. **Research Methodology** *(p. 11 - 12)*
  - Artifacts: set of data (*instantiation*), classification (*model*), web service (*method*)
  - Hevner's guideline for research
  - Hevner's research framework:

Hevner et al./Design Science in IS Research



Environment

Relevance

IS Research

Rigor

Knowledge Base

People
•Roles
•Capabilities
•Characteristics

Organizations
•Strategies
•Structure & Culture
•Processes

Technology
•Infrastructure
•Applications
•Communications Architecture
•Development Capabilities

Business Needs

Develop/Build
•Theories
•Artifacts

Assess

Refine

Justify/Evaluate
•Analytical
•Case Study
•Experimental
•Field Study
•Simulation

Applicable Knowledge

Foundations
•Theories
•Frameworks
•Instruments
•Constructs
•Models
•Methods
•Instantiations

Methodologies
•Data Analysis Techniques
•Formalisms
•Measures
•Validation Criteria

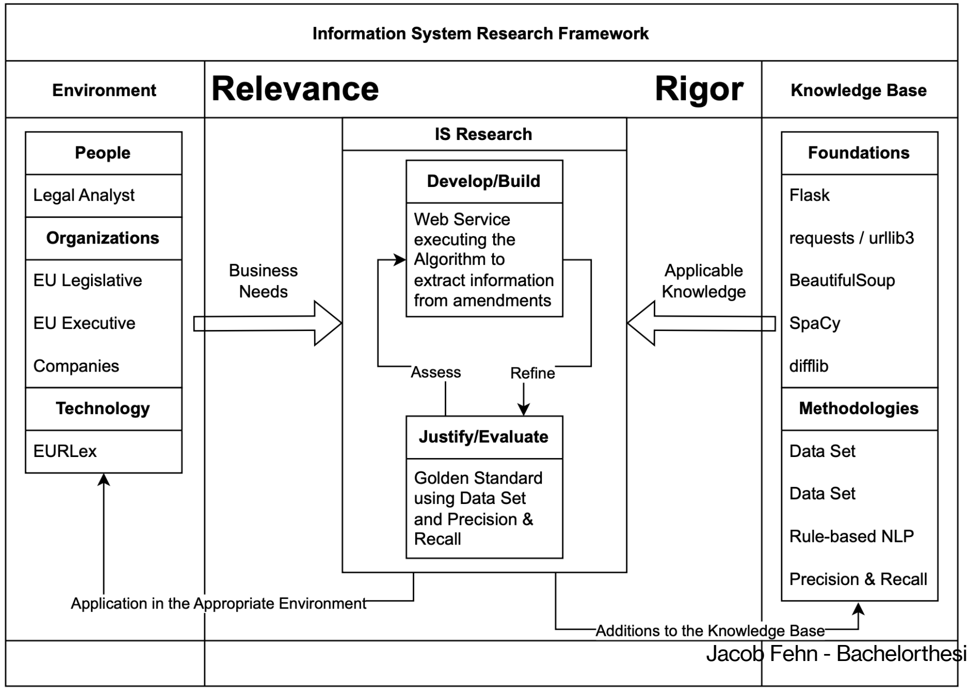
Application in the Appropriate Environment

Additions to the Knowledge Base

Figure 2. Information Systems Research Framework

Figure 3

The Information System Research Framework by Hevner([5] p.80) filled with the thesis' artifacts.



Information System Research Framework

Environment

Relevance

Rigor

Knowledge Base

People
Legal Analyst

Organizations
EU Legislative
EU Executive
Companies

Technology
EURLex

Business Needs

IS Research

Develop/Build
Web Service executing the Algorithm to extract information from amendments

Assess

Refine

Justify/Evaluate
Golden Standard using Data Set and Precision & Recall

Applicable Knowledge

Foundations
Flask
requests / urllib3
BeautifulSoup
SpaCy
difflib

Methodologies
Data Set
Data Set
Rule-based NLP
Precision & Recall

Application in the Appropriate Environment

Additions to the Knowledge Base

1. Introduction

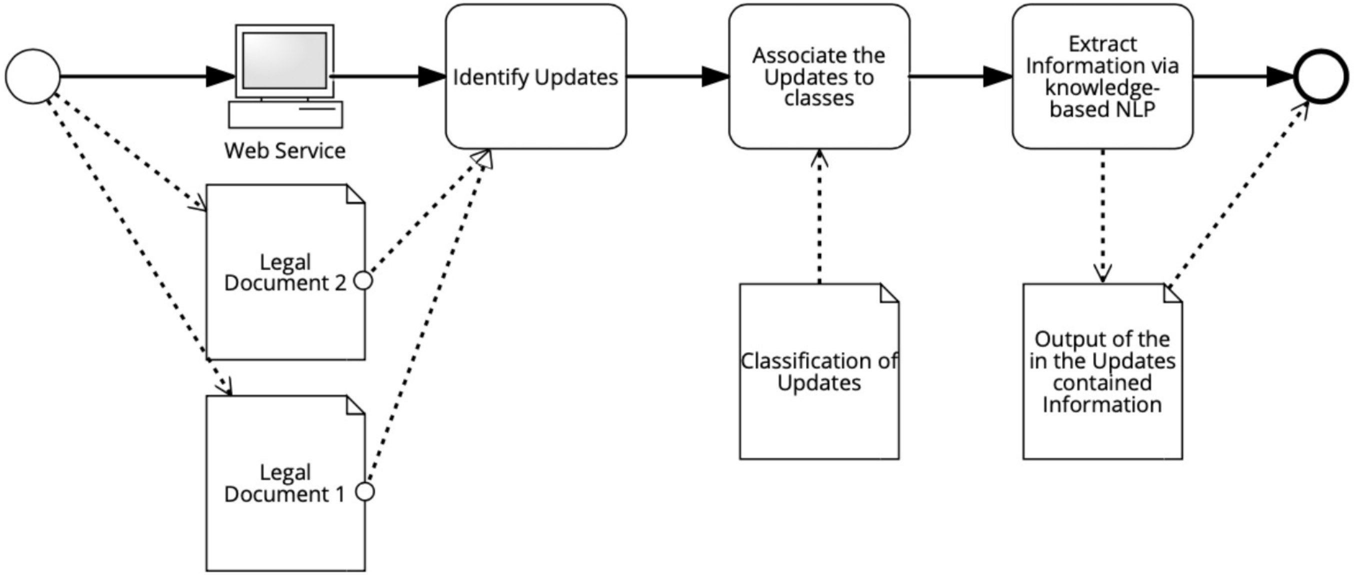
- 1. 1. Motivation
- 1. 2. Research Questions
- 1. 3. Research Methodology
- 1. 4. **Structure** *(p. 11)*
  - Motivation to solve a problem
  - Research Questions to be answered
  - Research Methodology for a Solution Design to approach the solution
  - Implementation of the Solution Design
  - Evaluation of Design and Implementation as answer to Research Questions
  - Discussion of Contribution and Challenges
  - Conclusion with the prospect of Future Work

1. Introduction *(p. 7 – 12)*

- Established a possible work process for the web service *(p. 10)*

Figure 2

This business process diagram shows the process behind the web service.

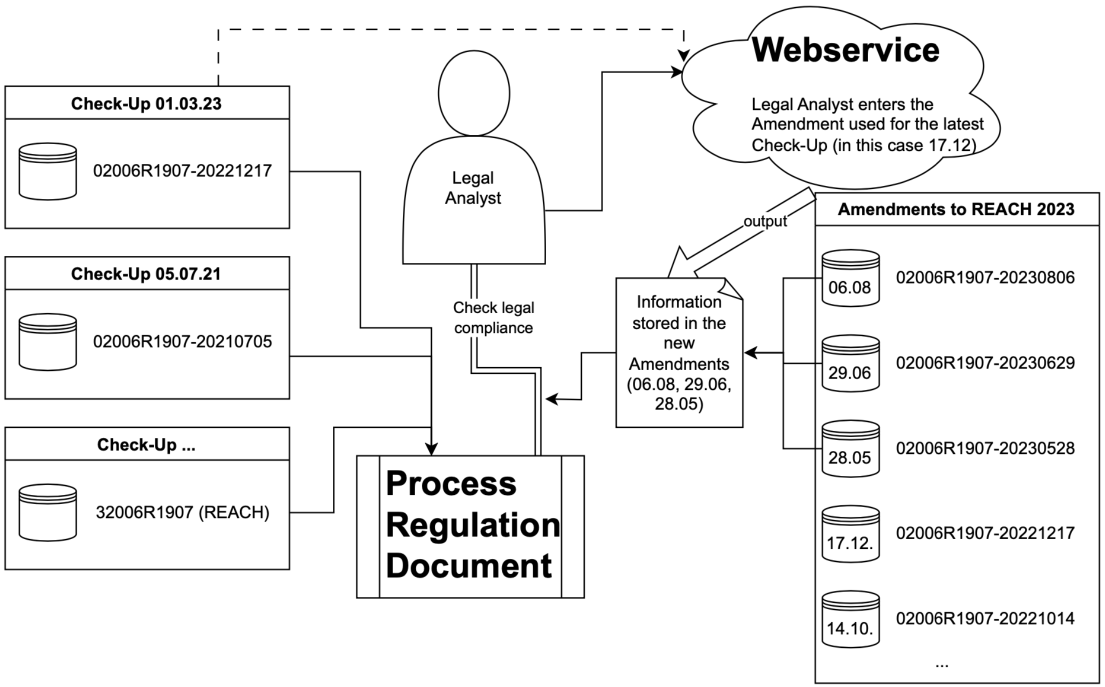

Web Service
Identify Updates
Associate the Updates to classes
Extract Information via knowledge-based NLP
Legal Document 2
Legal Document 1
Classification of Updates
Output of the in the Updates contained Information

1. Introduction *(p. 7 – 12)*

- Established a use case in need of answering *(p. 10)*

Figure 1

Use-case diagram of an analyst performing a compliance check via this thesis' web service.



2. Related Work

- 2. 1. Changes in Legal Documents
- 2. 2. Functional Types of Legal Statements
- 2. 3. Information Extraction by Natural Language Processing
- 2. 4. Compliance

2. Related Work

- 2. 1. **Changes in Legal Documents** *(p. 13 – 15)*
  - Papers on types of updates and working with EURLex
  - Recommendation for the Usage of pattern-based NLP
- 2. 2. Functional Types of Legal Statements
- 2. 3. Information Extraction by Natural Language Processing
- 2. 4. Compliance

2. Related Work

- 2. 1. Changes in Legal Documents
- 2. 2. **Functional Types of Legal Statements** *(p. 15 – 16)*
  - Papers on extracting information from Legal Texts based on patterns and working with SpaCy as NLP library.
  - Output solution
- 2. 3. Information Extraction by Natural Language Processing
- 2. 4. Compliance

2. Related Work

- 2. 1. Changes in Legal Documents
- 2. 2. Functional Types of Legal Statements
- 2. 3. **Information Extraction by Natural Language Processing** *(p. 16 – 18)*
  - Papers on working with (pattern-based) NLP (preprocessing and feature generation)
  - Output solution
- 2. 4. Compliance

2. Related Work

- 2. 1. Changes in Legal Documents
- 2. 2. Functional Types of Legal Statements
- 2. 3. Information Extraction by Natural Language Processing
- 2. 4. **Compliance** *(p. 18 - 19)*
  - Paper on semantic rules and automated compliance
  - Interesting for Future Work

2. Related Work *(p. 12 – 19)*

- Searched on Google Scholar and DBLP, especially JURIX and ICALI
  - Criteria for **inclusion** and **exclusion**
  - Whole bibliography in the git repository!

- **Conclusion**:

Collection of State-of-The-Art research

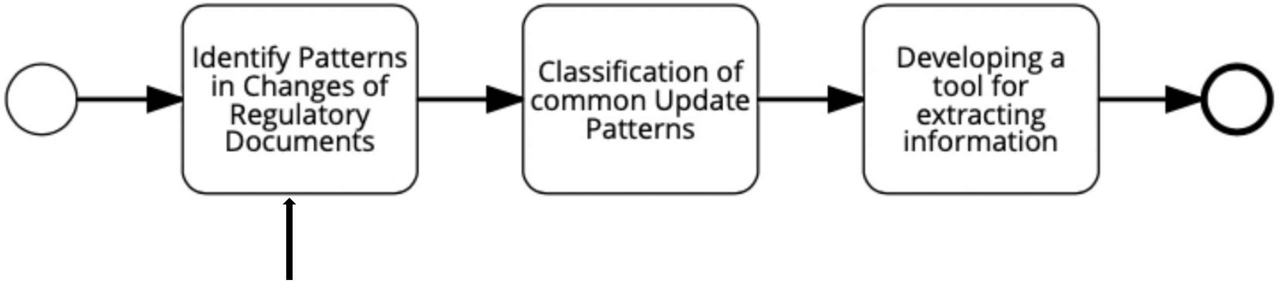
Learning and Education material

Clear recommendation to use pattern/knowledge-based NLP

3. Solution Design

- Business plan for the thesis and the web service as output *(p. 19)*

Figure 5
This (very simple) diagram shows the process plan of the thesis.


Identify Patterns in Changes of Regulatory Documents
Classification of common Update Patterns
Developing a tool for extracting information

- Additionally, the collection of test data for Evaluation!

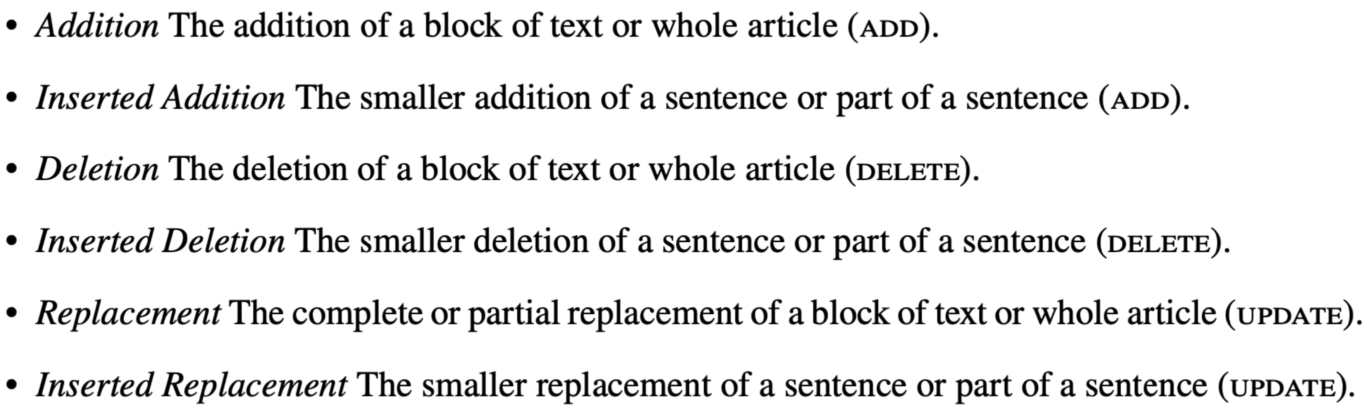
3. Solution Design

- 3. 1. Classification
- 3. 2. Patterns for NLP

3. Solution Design

- 3. 1. **Classification** *(p. 19 – 22)*
  - Changes and Their Modifications
  - Triangular Arrows and Change Names: ▶**M1** ▼**M1**
  - Classifications from EURLex and Related Work
  - Conclusively:

    (p. 22)



- *Addition* The addition of a block of text or whole article (ADD).
- *Inserted Addition* The smaller addition of a sentence or part of a sentence (ADD).
- *Deletion* The deletion of a block of text or whole article (DELETE).
- *Inserted Deletion* The smaller deletion of a sentence or part of a sentence (DELETE).
- *Replacement* The complete or partial replacement of a block of text or whole article (UPDATE).
- *Inserted Replacement* The smaller replacement of a sentence or part of a sentence (UPDATE).

- 3. 2. Patterns for NLP

3. Solution Design

- 3. 1. Classification
- 3. 2. **Patterns for NLP** *(p. 22 – 23)*
  - NLP Entity Recognition refining by Patterns
  - Regular Expressions for recognizing Token-Sequence
  - Limiting Entities (LAW, ORG, GPE, … ; not: ordinal, cardinal, …)

```
(1){"label": "LAW",
"pattern": [{"LOWER": "point"}, {"SHAPE": "d", "OP": "+"}]}
```

3. Solution Design *(p. 19 – 23)*

- **Classification**
- 6 Classes of Modifications for 3 different Types of textual Modification

- **Patterns for NLP**
- 4 Patterns for Organizations
- 12 Patterns for Legal References

4. Implementation

- 4. 1. Test Data
- 4. 2. HTML Processing
- 4. 3. Natural Language Processing

4. Implementation

- 4. 1. **Test Data** *(p. 24 - 25)*
  - Collecting 1% of Documents for each of the 20 Directories
  - 270 (one more, but it is oversized and was not used in testing)
  - At least 1 consolidated Version: 1 test
  - More than 1 consolidated Version: 3 tests
  - <u>594 tests</u> with by-hand counted expected results
- 4. 2. HTML Processing
- 4. 3. Natural Language Processing

4. Implementation

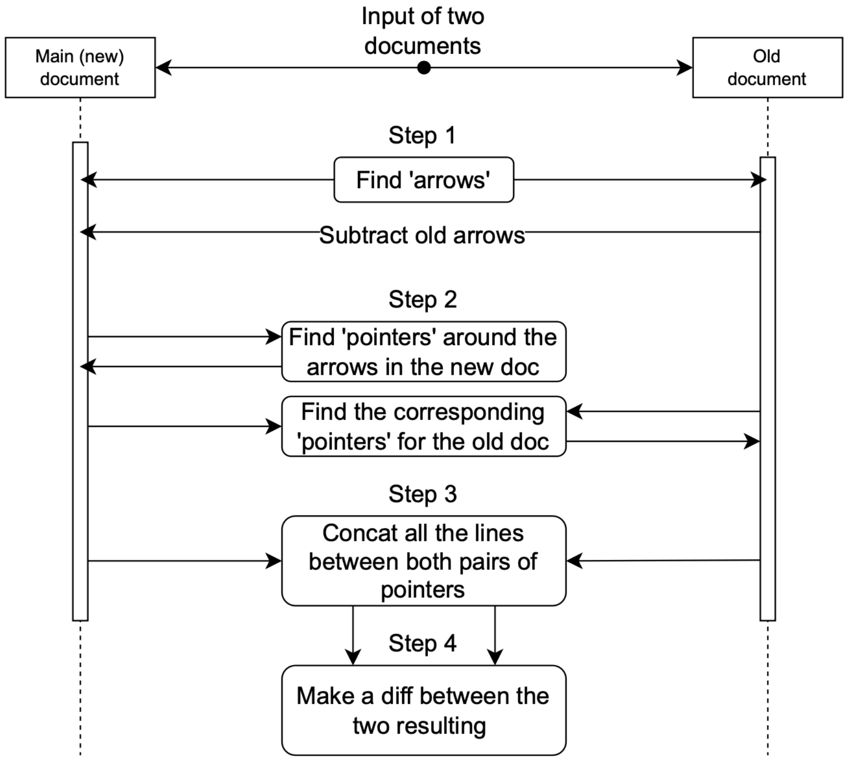
- 4. 1. Test Data
- 4. 2. **HTML Processing** *(p. 25 - 27)*
  - Input of two EURLex documents as HTML
  - Output as List of Modifications (lists of their attributes)
  - 4 Steps to find Arrows, outline Documents, compare passages
- 4. 3. Natural Language Processing

4. Implementation

- 4. 2. **HTML Processing**
  - 4 Steps: *(p.26)*

Figure 6

The four steps of the HTML processing.


Input of two documents
Main (new) document
Old document
Step 1
Find 'arrows'
Subtract old arrows
Step 2
Find 'pointers' around the arrows in the new doc
Find the corresponding 'pointers' for the old doc
Step 3
Concat all the lines between both pairs of pointers
Step 4
Make a diff between the two resulting

return: names (arrows), contents (between arrows), positions (upper pointer) and the diffs

4. Implementation

- 4. 1. Test Data
- 4. 2. HTML Processing
- 4. 3. **Natural Language Processing** *(p. 27 - 30)*
  - Input filename (for web view), HTML processing result and Boolean for NLP model
  - Output changes and their modifications in 3-dimensional list

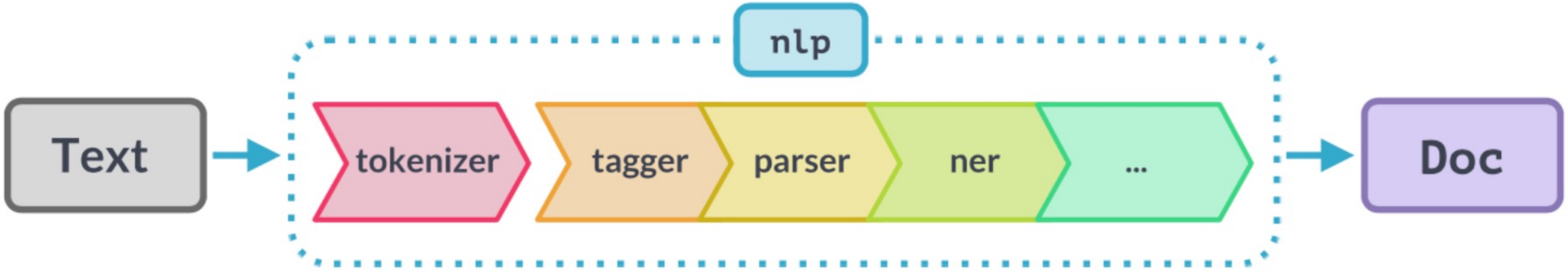
  - Fast or Accurate SpaCy standard web model
  - Sort Modifications to their Changes, process Modification content, refine diff and Classifying!
  - Write HTML file to be rendered in the web service
  - Return lists

4. Implementation

- 4. 3. **Natural Language Processing**
  - Return includes full SpaCy Pipeline with augmented Entity Recognition *(p. 29)*

Figure 8

The (shortend) natural language pipeline from `https://spacy.io/usage/processing-pipelines` *(Last access: 07.11.23). Tokenizer, (POS) Tagger, (Dependencies) Parser and (Entity Recognition) NER are shown inside the pipeline. Attribute Ruler and Lemmatizer are missing!*

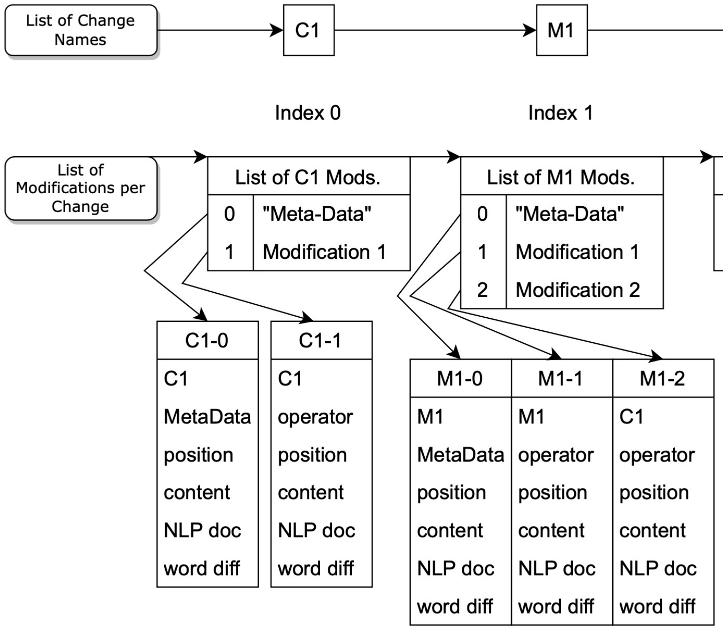


4. Implementation

- 4. 3. **Natural Language Processing** *(p. 29)*

Figure 7

This shows the algorithms output. The first list is a list of all the changes detected. The second is a three-dimensional list, with the first dimension indicating the changes (indexes align with the first list). The second dimension shows given a first-dimensional index (change) a list of modifications that are part of this change. The last dimension is the "frame" of the modifications containing all relevant information about the modification.



Mod.-Lists

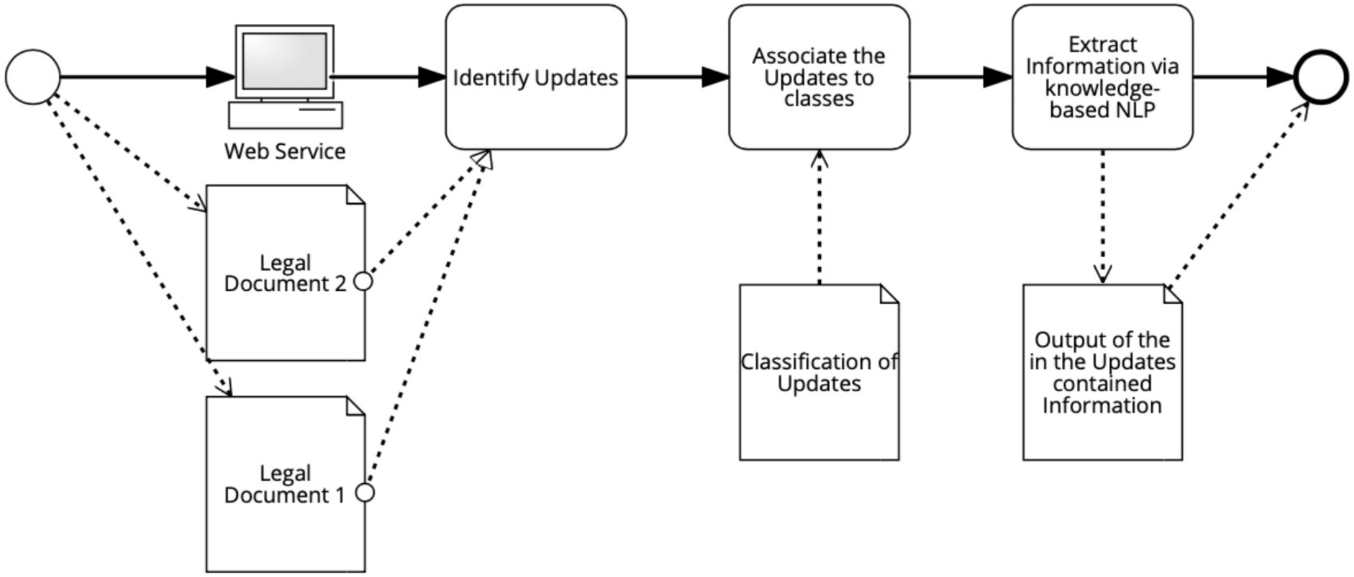
Index 0: Change Name
Index 1: Operator
Index 2: Position
Index 3: Content
Index 4: Content as doc
Index 5: Word-level Diff

Doc being a SpaCy Class, in which the preprocessing is stored.

RECAP: Introduction

- Established a possible work process for the web service *(p. 10)*

Figure 2
This business process diagram shows the process behind the web service.


Web Service
Identify Updates
Associate the Updates to classes
Extract Information via knowledge-based NLP
Legal Document 2
Legal Document 1
Classification of Updates
Output of the in the Updates contained Information

4. Implementation

- Comparison:

(p. 10)

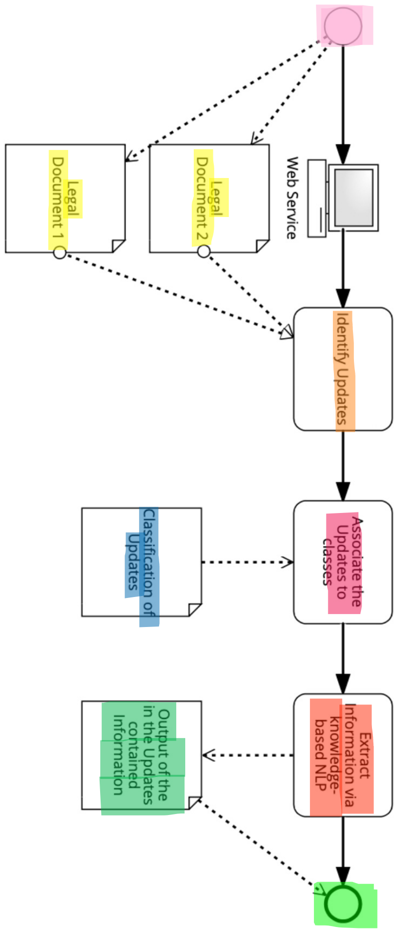
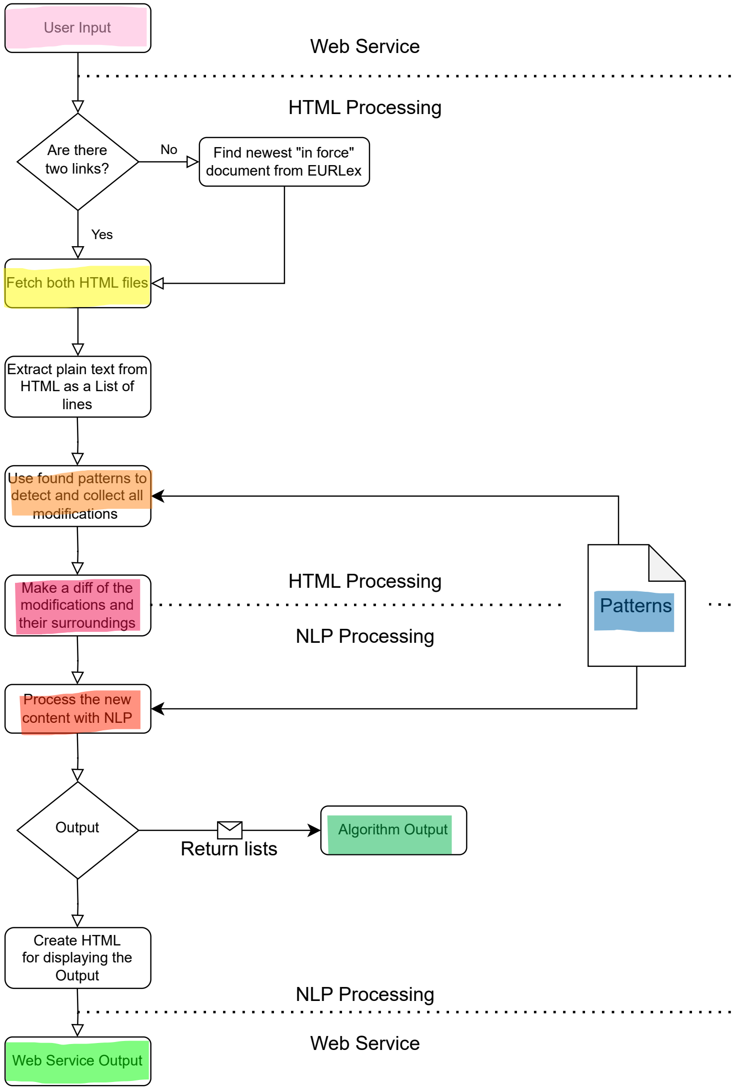


Figure 2

This business process diagram shows the process behind the web service.



User Input
Web Service
HTML Processing
Are there two links?
No
Find newest "in force" document from EURLex
Yes
Fetch both HTML files
Extract plain text from HTML as a List of lines
Use found patterns to detect and collect all modifications
Patterns
HTML Processing
Make a diff of the modifications and their surroundings
NLP Processing
Process the new content with NLP
Output
Return lists
Algorithm Output
Create HTML for displaying the Output
NLP Processing
Web Service
Web Service Output

(p. 46)

4. Implementation *(p. 23 – 30)*

- A working prototype of the web service (without deployment)
- Limitations:
  - Diff => mistakes in classification
  - Modifications in Title or Introduction => mislabeling of those

5. Evaluation

Just for understanding!

- Little Mistake:
  - *On page 31*: Table 2 has in the description "*In Directory 3 (marked with *) on document was not tested because of its oversize.*" but no marking inside the table. That's because this is for Table 3!
  - Also in the text: "(…) how many different documents [were tested] for each directory (…)"

  and some other *wording* and *phrasing* ambiguous!

5. Evaluation

- Quantitative Testing by numbers of Changes and Modifications (594 or 270 test cases)
- 8 of 594 tests fail (6 of 270)
  - 4 Reasons: *mods. in images, untitled mods., recognition error, nested overwritten modification*
- In 594 the algorithm finds 1632 Changes with overall 9401 Modifications (856 / 4692)
- With the expected results, **Precision** and **Recall** can be calculated: *(p. 35)*

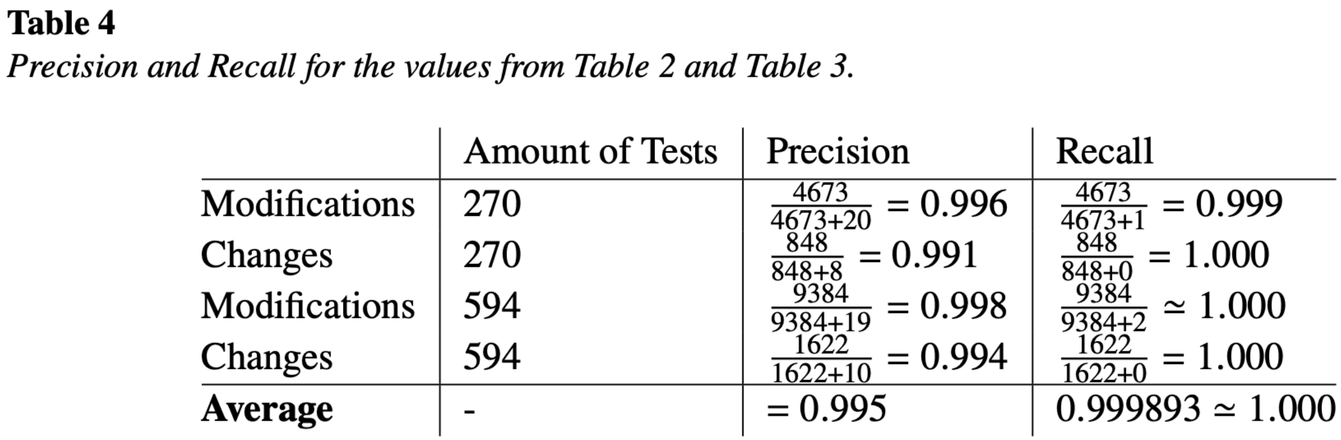


Table 4

Precision and Recall for the values from Table 2 and Table 3.

	Amount of Tests	Precision	Recall
Modifications	270	$\frac{4673}{4673+20} = 0.996$	$\frac{4673}{4673+1} = 0.999$
Changes	270	$\frac{848}{848+8} = 0.991$	$\frac{848}{848+0} = 1.000$
Modifications	594	$\frac{9384}{9384+19} = 0.998$	$\frac{9384}{9384+2} \simeq 1.000$
Changes	594	$\frac{1622}{1622+10} = 0.994$	$\frac{1622}{1622+0} = 1.000$
Average	-	$= 0.995$	$0.999893 \simeq 1.000$

5. Evaluation

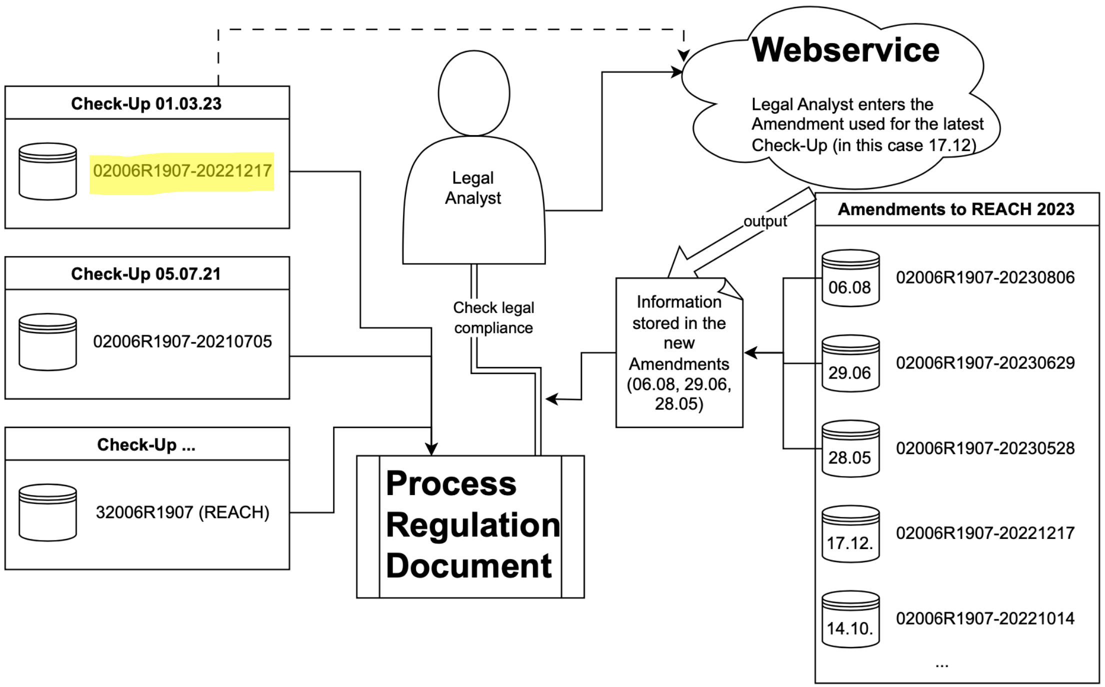
- Qualitative: *(p. 35 - 36)*
  - **Usability** with additional functionality and not many options
  - Acceptable **Performance** with 7,5 seconds per documents (8 documents per minute: 594 / 73 min)
  - **Clear** arrangement of Changes and Modifications in the output
  - **Links** to original documents for further analysis
  - Some mistakes due to **limited diff**!

RECAP: Introduction *(p. 7 – 12)*

- Established a use case in need of answering *(p. 10)*

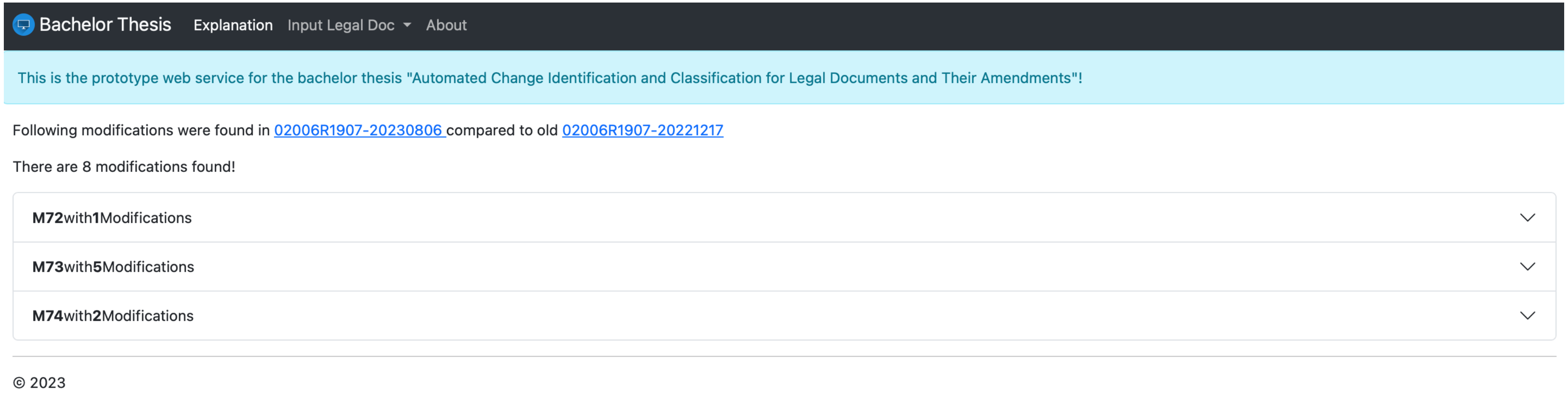
Figure 1

Use-case diagram of an analyst performing a compliance check via this thesis' web service.



5. Evaluation *(not in the paper)*

- Use Case
  - ➢ Start the Web Service
  - ➢ Navigate to CELEX number (in this case)
  - ➢ Enter the last used consolidated version: 02006R1907-20221217
  - ➢ Check the output!



This is the prototype web service for the bachelor thesis "Automated Change Identification and Classification for Legal Documents and Their Amendments"!

Following modifications were found in 02006R1907-20230806 compared to old 02006R1907-20221217

There are 8 modifications found!

M72with**1**Modifications

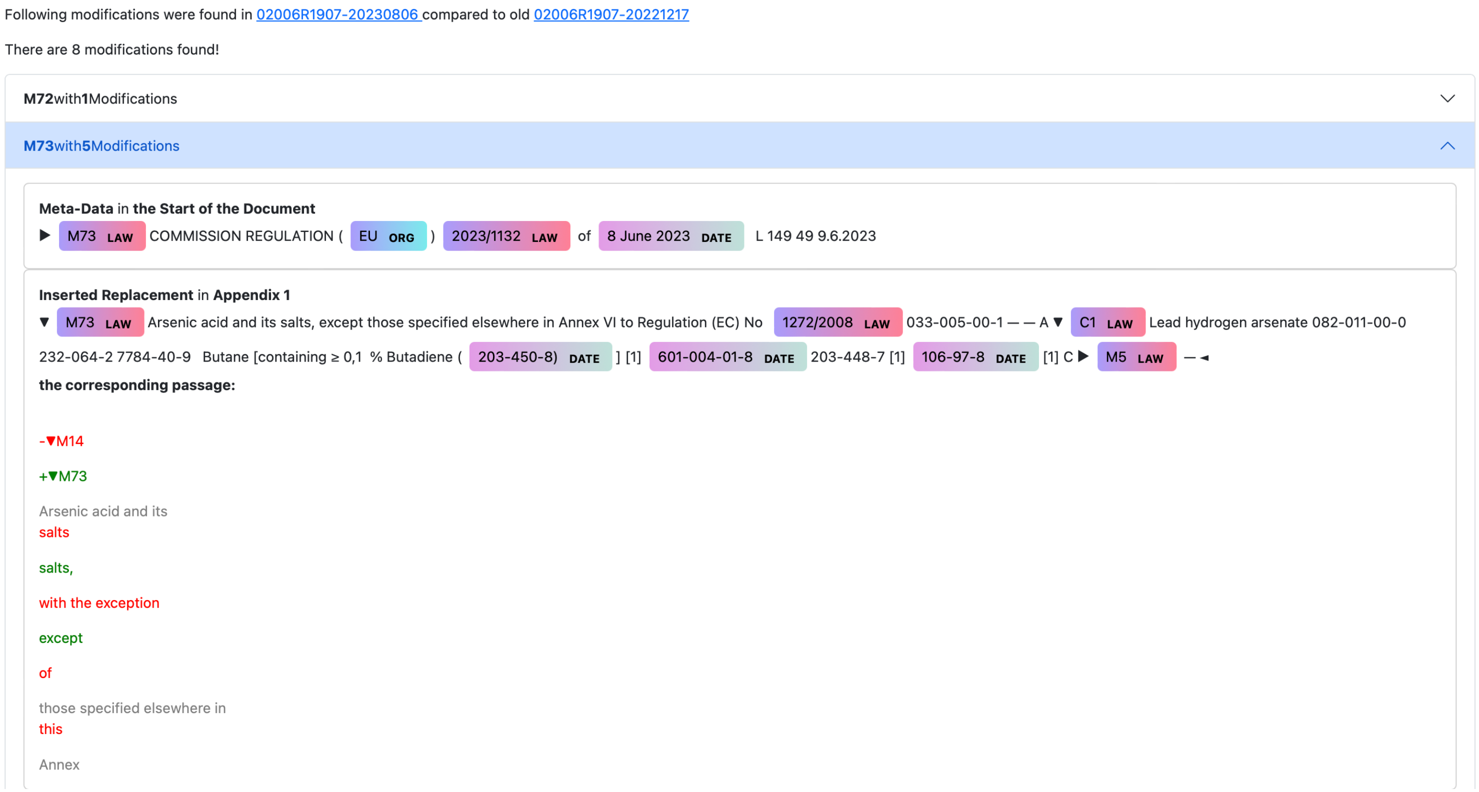
M73with**5**Modifications

M74with**2**Modifications

© 2023

5. Evaluation *(not in the paper)*

- Use Case



Following modifications were found in 02006R1907-20230806 compared to old 02006R1907-20221217

There are 8 modifications found!

M72with**1**Modifications

M73with**5**Modifications

Meta-Data in **the Start of the Document**

▶ M73 LAW COMMISSION REGULATION (EU ORG) 2023/1132 LAW of 8 June 2023 DATE L 149 49 9.6.2023

Inserted Replacement in **Appendix 1**

▼ M73 LAW Arsenic acid and its salts, except those specified elsewhere in Annex VI to Regulation (EC) No 1272/2008 LAW 033-005-00-1 — — A ▼ C1 LAW Lead hydrogen arsenate 082-011-00-0 232-064-2 7784-40-9 Butane [containing ≥ 0,1 % Butadiene (203-450-8) DATE] [1] 601-004-01-8 DATE 203-448-7 [1] 106-97-8 DATE [1] C ▶ M5 LAW — ◂

the corresponding passage:

-▼M14

+▼M73

Arsenic acid and its
salts

salts,

with the exception

except

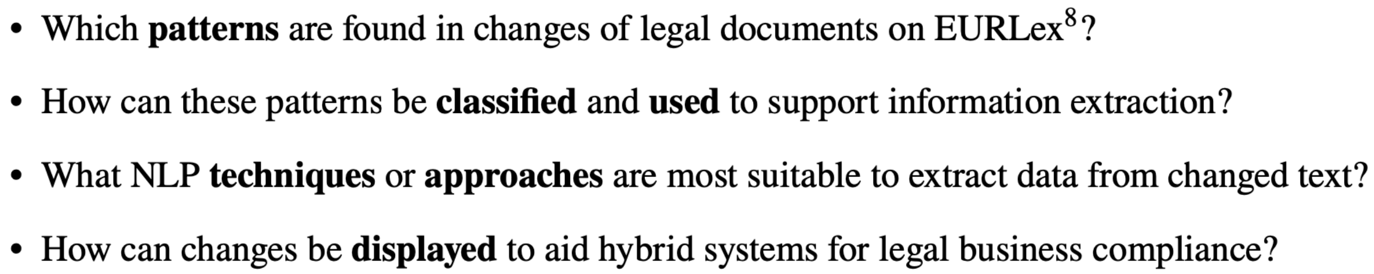
of

those specified elsewhere in
this

Annex

RECAP: Introduction

- 1. 1. Motivation
- 1. 2. **Research Questions** *(p.11)*



- Which **patterns** are found in changes of legal documents on EURLex[8]?
- How can these patterns be **classified** and **used** to support information extraction?
- What NLP **techniques** or **approaches** are most suitable to extract data from changed text?
- How can changes be **displayed** to aid hybrid systems for legal business compliance?

- 1. 3. Research Methodology
- 1. 4. Structure

5. Evaluation

- **Research Questions:** *(p. 36 – 37)*
  - 1. Change Patterns
  - 2. Classification of Patterns
  - 3. NLP approach
  - 4. Output

5. Evaluation

- **Research Questions:** *(p. 36 – 37)*
  - 1. **Change Patterns**: *(Chapter Solution Design)*
    - Arrows: ▼M1 ▶M1
    - Changes and Modifications
    - Pattern Phrases for NLP

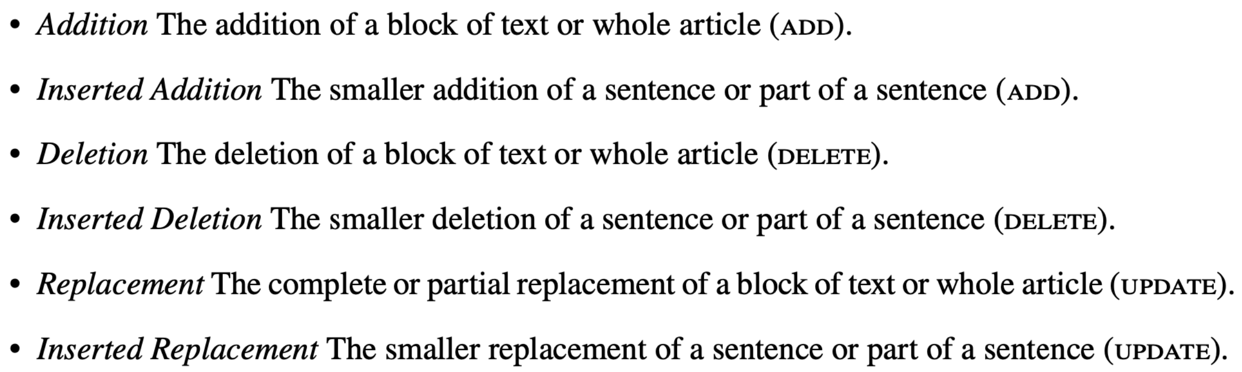
```
(1){"label": "LAW",

"pattern": [{"LOWER": "point"}, {"SHAPE": "d", "OP": "+"}]}
```

  - 2. Classification of Patterns
  - 3. NLP approach
  - 4. Output

5. Evaluation

- **Research Questions:** *(p. 36 – 37)*
  - 1. Change Patterns
  - 2. **Classification of Patterns**: *(Chapter Solution Design)*



- *Addition* The addition of a block of text or whole article (ADD).
- *Inserted Addition* The smaller addition of a sentence or part of a sentence (ADD).
- *Deletion* The deletion of a block of text or whole article (DELETE).
- *Inserted Deletion* The smaller deletion of a sentence or part of a sentence (DELETE).
- *Replacement* The complete or partial replacement of a block of text or whole article (UPDATE).
- *Inserted Replacement* The smaller replacement of a sentence or part of a sentence (UPDATE).

  - 3. NLP approach
  - 4. Output

5. Evaluation

- **Research Questions:** *(p. 36 – 37)*
  - 1. Change Patterns
  - 2. Classification of Patterns
  - 3. **NLP approach**:
    - Pattern-based NLP *(Chapter Related Work)*
    - Pattern for this from the Phrases in answer to RQ 1 *(Chapter Solution Design)*
  - 4. Output

5. Evaluation

- **Research Questions**: *(p. 36 – 37)*
  - 1. Change Patterns:
  - 2. Classification of Patterns:
  - 3. NLP approach:
  - 4. **Output**: *(Chapter Implementation)*
    - Human (*web service*) output: sorted by Changes and Occurrence!
    - Machine (*algorithm*) output: in lists also sorted by Changes and Occurrence, containing the SpaCy doc!

5. Evaluation *(p. 30 – 37)*

- **Research Questions**:
  - 1. Change Patterns
  - 2. Classification of Patterns
  - 3. NLP approach
  - 4. Output
- **Artifacts**:
  - Instantiation: **Data Set** of 271 Documents (containing 594 usable Tests)
  - Model: identified **form** of modification and **classification**
  - Method: **Web Service** with **Algorithm** in the backend

6. Discussion

- 6. 1. **Contribution** *(p. 37)*
  - *Identified* way of changes in EURLex
  - *Classified* the type of Modifications
  - *Refined* NLP Entity Recognition for EU legislative
  - *Created* a Web Service and Algorithm to find changes and extract information
  - *Reduced* workflow of collecting changed contents for further legal work
- 6. 2. Challenges

6. Discussion

- 6. 1. Contribution
- 6. 2. **Challenges** *(p. 38 – 40)*
  - Processing HTML instead of just the Text
  - The diff
    - Avoiding duplicate processing
    - Reduce unnecessary context and improve visibility and accuracy
    - Thus, improving the classification
  - Exceptions
    - Title and Introduction mislabeled
    - Single Line documents
    - Missing: the test failures

6. Conclusion *(p. 40 – 41)*

- Resulted in usable prototype with good detecting capabilities
- Useful to get an overview and integrate changes into a business plan

- **Future Work**:
  - Refining and Optimizing the Web Service
  - ML models for specific research or business solution design
  - Integration into BPC software or other in-depth application

99. Appendix

- Bibliography:
  - 22 sources in the paper as well as 50 footnotes
  - All research paper in the git repository
- Terminology:
  - a few words from **Related Work**, this **Thesis Work** and **Future Work** explained

The git repository: https://github.com/affentypi/Webservice_Thesis

Thank you for your attention!

Jacob Fehn