

Exam DSL

Niklas de Bruyn

Malte Schulze Balhorn

Robin Thielking

Inhalt

- Ziel des Projekts
- Umsetzung
- Probleme
- Ergebnis
- Lessons Learned
- Zukunft

Was ist das Ziel des Projekts?

- Lehrkräften ermöglichen einfach Klassenarbeiten für E-Learning (z.B. ILIAS oder moodle) Plattformen zu erstellen
- Eigene DSL entwickeln, die leicht zu verstehen ist
- IR soll einfach zu verarbeiten sein
- Eingangssprache Deutsch

Umsetzung

- Grammatik erstellen mittels ANTLR
- IR ist eine JSON

Umsetzung

```
prog:    tasks NEWLINE? EOF;
tasks:   (task_definition NEWLINE)* task_definition ;
task_definition:   endless_words task;

task:    '(' RIGHT_OR_FALSE ')' ':' NEWLINE? true_false_task (NEWLINE true_false_task)* ';' 
        | '(' SORTING ')' ':' NEWLINE? sorting_task (NEWLINE sorting_task)* ';' 
        | '(' MATCHING ')' ':' NEWLINE? matching_task (NEWLINE matching_task)* ';' 
        | '(' MARKING ')' ':' NEWLINE? marking_task ';' 
        | '(' CLOZE_TEXT ')' ':' NEWLINE? cloze_task ';' 
        | '(' CORRECTION_TEXT ')' ':' NEWLINE? correction_task ';' 
        | '(' CHOICE_TEXT ')' ':' NEWLINE? choice_task (NEWLINE choice_task)* ';' ;
```

Umsetzung - Lückentext - Grammatik

```
cloze_task:           question_or_statement NEWLINE? cloze_text;  
  
cloze_text:          ((sentence | cloze_sentence)+ NEWLINE?)*;  
cloze_sentence:      (endless_words? (cloze_word endless_words?)+ PUNCTUATION);  
cloze_word:          ('(' word ',' positive_task_point ')');
```

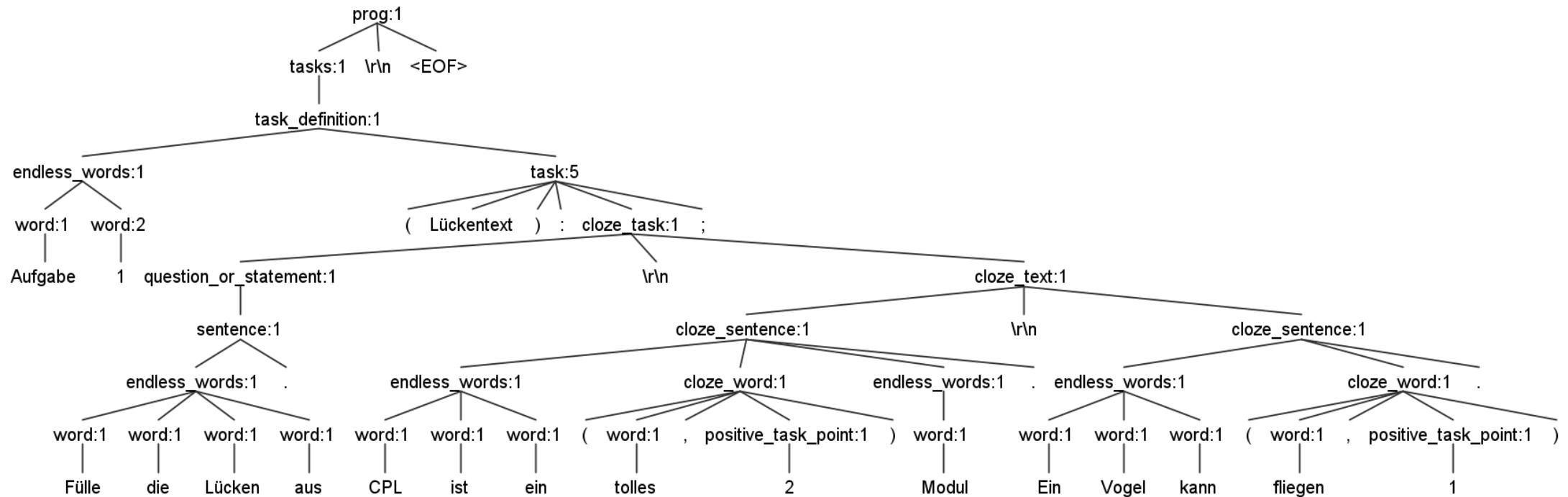
Umsetzung - Lückentext - Input

Aufgabe 1(Lückentext): Fülle die Lücken aus.

CPL ist ein (tolles,2) Modul.

Ein Vogel kann (fliegen,1).;

Umsetzung - Lückentext - Parsing Tree



Umsetzung - Lückentext - IR

```
{  
    "type": "Lückentext",  
    "header": "Aufgabe 1",  
    "task": {  
        "question": "Fülle die Lücken aus.",  
        "sentences": [  
            {  
                "parts": [  
                    {  
                        "text": "CPL ist ein"  
                    },  
                    {  
                        "blank": {  
                            "solution": "tolles",  
                            "points": 2  
                        }  
                    },  
                    {  
                        "text": "Modul."  
                    }  
                ]  
            },  
            { ... }  
        ]  
    }  
}
```

```
{  
    "type": "Lückentext",  
    "header": "Aufgabe 1",  
    "task": {  
        "question": "Fülle die Lücken aus.",  
        "sentences": [  
            { ... },  
            {  
                "parts": [  
                    {  
                        "text": "Ein Vogel kann"  
                    },  
                    {  
                        "blank": {  
                            "solution": "fliegen",  
                            "points": 1  
                        }  
                    },  
                    {  
                        "text": ".  
                    }  
                ]  
            },  
            { ... }  
        ]  
    }  
}
```

Umsetzung - Auswahl - Grammatik

```
choice_task: question_or_statement correct_choice+ false_choices;

correct_choice: '-' endless_words('(' positive_task_point ')');

false_choices: ('-' endless_words('(' negative_task_point ')'))+
               | '-' endless_words)+;
```

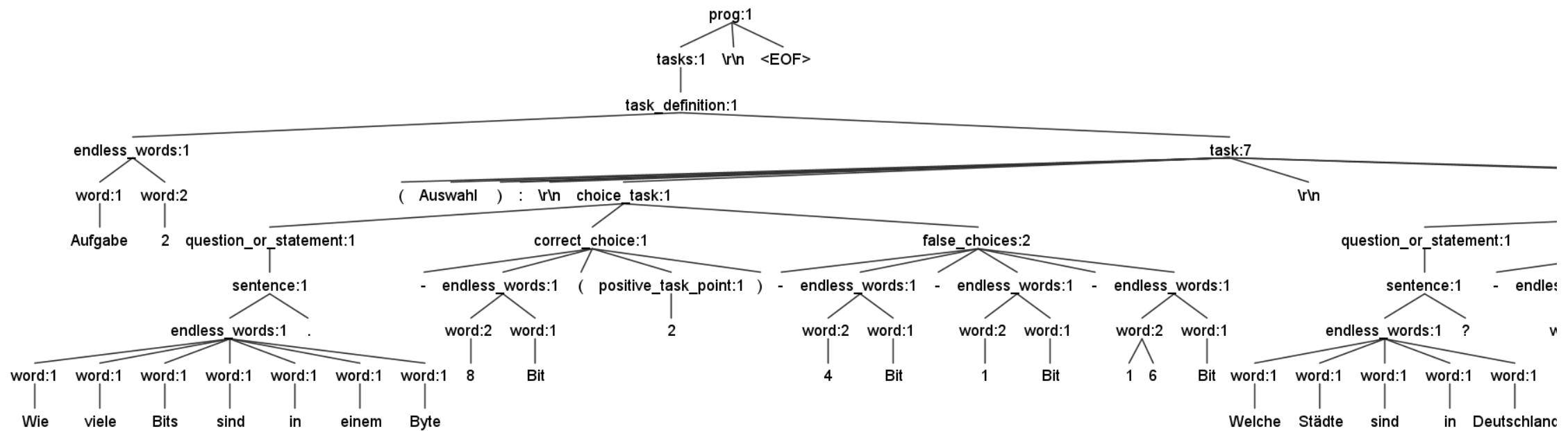
Umsetzung - Auswahl - Input

Aufgabe 2 (Auswahl):

Wie viele Bits sind in einem Byte. -8 Bit(2) -4 Bit -1 Bit -16 Bit

Welche Städte sind in Deutschland? -Berlin(1) -Minden(1) -Paris(-1) -London(-1);

Umsetzung - Auswahl - Parsing Tree



Umsetzung - Auswahl - IR

```
{  
    "type": "Auswahl",  
    "header": "Aufgabe 2",  
    "task": {  
        "lines": [  
            {  
                "question": "Wie viele Bits sind in einem Byte.",  
                "options": [  
                    {  
                        "text": "8 Bit",  
                        "points": 2,  
                        "isCorrect": true  
                    },  
                    {  
                        "text": "4 Bit",  
                        "points": 0,  
                        "isCorrect": false  
                    },  
                    {  
                        "text": "1 Bit",  
                        "points": 0,  
                        "isCorrect": false  
                    },  
                    {  
                        "text": "16 Bit",  
                        "points": 0,  
                        "isCorrect": false  
                    }  
                ],  
                "..."  
            }  
        ]  
    }  
}
```

```
{  
    "type": "Auswahl",  
    "header": "Aufgabe 2",  
    "task": {  
        "lines": [  
            {  
                "...",  
                {  
                    "question": "Welche Städte sind in Deutschland?",  
                    "options": [  
                        {  
                            "text": "Berlin",  
                            "points": 1,  
                            "isCorrect": true  
                        },  
                        {  
                            "text": "Minden",  
                            "points": 1,  
                            "isCorrect": true  
                        },  
                        {  
                            "text": "Paris",  
                            "points": -1,  
                            "isCorrect": false  
                        },  
                        {  
                            "text": "London",  
                            "points": -1,  
                            "isCorrect": false  
                        }  
                    ]  
                }  
            }  
        ]  
    }  
}
```

Umsetzung - RoF - Grammatik

```
true_false_task:           question_or_statement true_false_answer;  
  
true_false_answer:   ANSWER_TRUE  
                    | ANSWER_FALSE ARROW reason;  
reason: sentence;
```

Umsetzung - RoF - Input

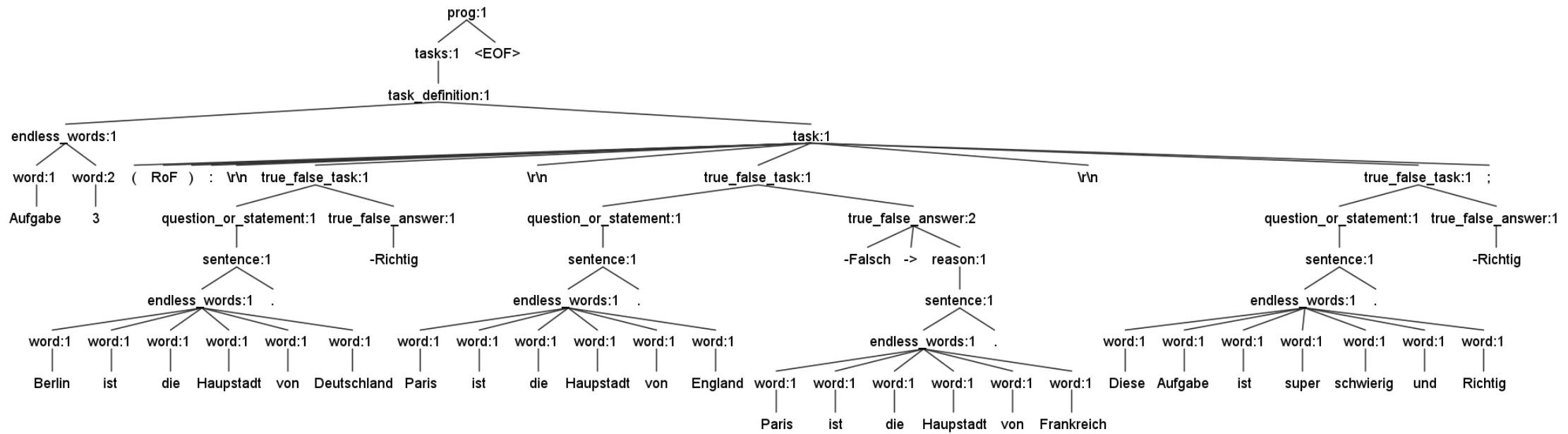
Aufgabe 3(RoF):

Berlin ist die Haupstadt von Deutschland. -Richtig

Paris ist die Haupstadt von England. -Falsch → Paris ist die Haupstadt von Frankreich.

Diese Aufgabe ist super schwierig und Richtig. -Richtig;

Umsetzung - RoF - Parsing Tree



Umsetzung - RoF - IR

```
{  
    "type": "RoF",  
    "header": "Aufgabe 3",  
    "lines": [  
        {  
            "question": "Berlin ist die Haupstadt von Deutschland.",  
            "answer": {  
                "isTrue": true  
            }  
        },  
        {  
            "question": "Paris ist die Haupstadt von England.",  
            "answer": {  
                "isTrue": false,  
                "reason": "Paris ist die Haupstadt von Frankreich."  
            }  
        },  
        {  
            "question": "Diese Aufgabe ist super schwierig und Richtig.",  
            "answer": {  
                "isTrue": true  
            }  
        }  
    ]  
}
```

Probleme

- Viele Fragen zu klären bzgl. der Aufgaben
 - Wie sollen die Punkte verteilt werden?
 - Haben Aufgaben Einfluss auf andere Aufgaben?
- Große Schwierigkeit besteht in der Übersetzung zu z.B. ILIAS XML
- ILIAS ändert regelmäßig Import/Export-Funktionen, daher großer Wartungsaufwand

Ergebnis

```
▼<respcondition continue="Yes">
  ▼<conditionvar>
    <varequal respident="gap_0">tolles</varequal> ←
  </conditionvar>
  <setvar action="Add">2</setvar> ←
  <displayfeedback feedbacktype="Response" linkrefid="0_Response_0"/>
</respcondition>
▼<respcondition continue="Yes">
  ▼<conditionvar>
    <varequal respident="gap_1">fliegen</varequal> ←
  </conditionvar>
  <setvar action="Add">1</setvar> ←
  <displayfeedback feedbacktype="Response" linkrefid="1_Response_0"/>
</respcondition>
```

ILIAS XML

```
<quiz>
  <question type="cloze">
    <name>
      <text>Aufgabe 1</text>
    </name>
    <questiontext format="html">
      <text><![CDATA[<p>CPL ist ein {2:SHORTANSWER:=tolles#Correct}<br/>Modul.<br/>Ein Vogel kann {1:SHORTANSWER:=fliegen#Correct}.</p>]]></text>
    </questiontext>
    <generalfeedback format="html">
      <text></text>
    </generalfeedback>
  </question>
</quiz>
```

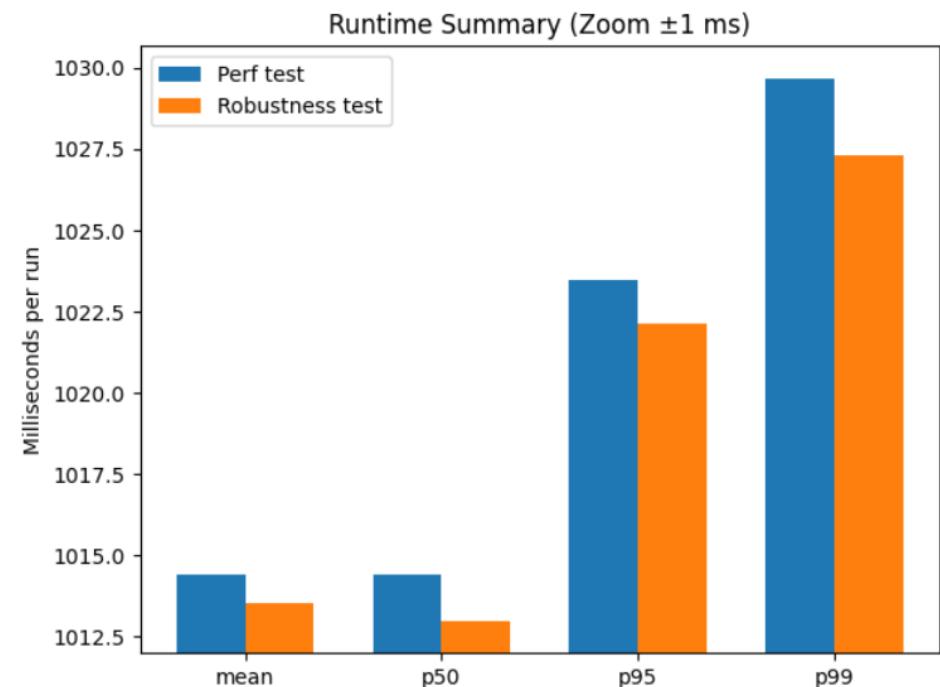
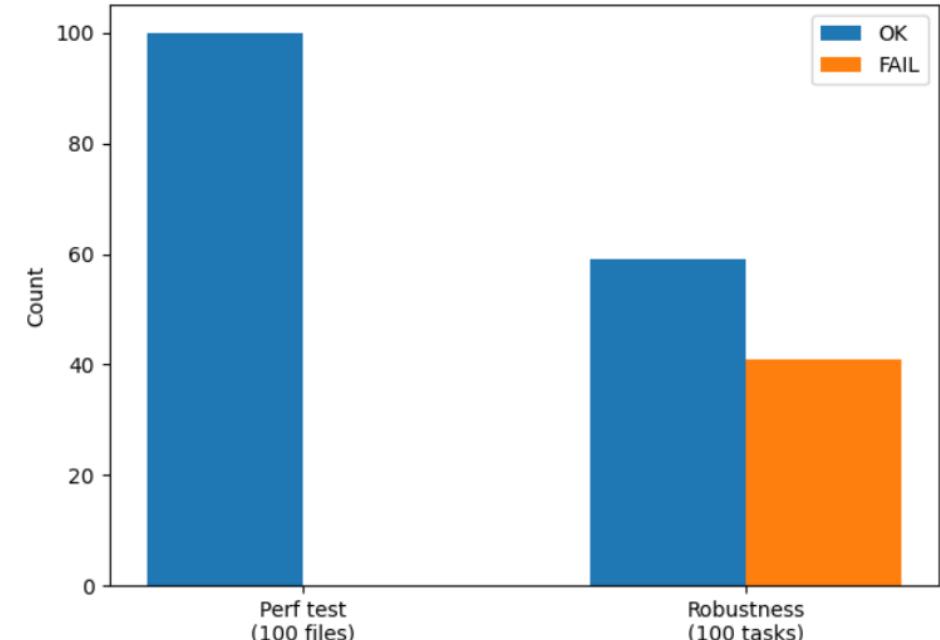
moodle XML

```
{
  "type": "Lückentext",
  "header": "Aufgabe 1",
  "task": {
    "question": "Fülle die Lücken aus.",
    "sentences": [
      {
        "parts": [
          {
            "text": "CPL ist ein"
          },
          {
            "blank": {
              "solution": "tolles",
              "points": 2
            }
          }
        ],
        "text": "Modul."
      },
      {
        "parts": [
          {
            "text": "Ein Vogel kann"
          },
          {
            "blank": {
              "solution": "fliegen",
              "points": 1
            }
          }
        ],
        "text": "."
      }
    ]
  }
},
```

|R

Evaluation - Performance

- Perf test:
 - Geprüftes Aufgaben**set**
 - 100 mal kompilieren lassen
 - Performance sehr stabil
- Robustness:
 - 100 synthetische Aufgaben
 - Jede Aufgabe **isoliert** geprüft
 - Hohe Fehlerrate
 - Lückentexten/Lueckentext
 - "C++" "Sortiere die Zahlen--1 –2.3 –4"



Lessons Learned

- Bei deutscher Eingabesprache sicherstellen, dass die Umgebungsvariablen richtig eingestellt sind:

```
$env:JAVA_TOOL_OPTIONS="-Dfile.encoding=UTF-8"
```

- Frühzeitig ein DSL-Manual anlegen

Nicht enthaltende Funktionen

- Verschiedene Metadaten rund um die Lernplattformen selber
- Plattform-spezifische Antwortmöglichkeiten
(z.B. Lückentextkombinationen in ILIAS)
- Mehrere Antwortmöglichkeiten für ein Lücke
- Multiple Zuordnungen von Termen und Definitionen in Zuordnung
- Ready2Use Pipeline von DSL zu z.B. ILIAS

→ Ziel: Bereitstellung funktionaler Kernkomponenten

Ausblick

- Gleichgewicht zwischen verteilten Aufgabenpunkten
- Übersetzung von IR zu kompatiblen Lernplattform-Format (z.B. ILIAS, moodle -> XML)
- Ähnlicher Prototyp zu unserem Projekt:
[exams2ilias: Generation of Exams in ILIAS Format in exams: Automatic Generation of Exams in R](#)