LFUNK 2026 · 2026-09-30

autotool als Referenz für einen LLM-Grader

LLM-Bewertung und Aufgabenerzeugung für die Hochschullehre.

Jonas Gwozdz
Web & Software Engineering (WSE) Research Group
Hochschule für Technik, Wirtschaft und Kultur Leipzig (HTWK Leipzig)

jonas.gwozdz@htwk-leipzig.de

WSE Research · Webseite

Meine Forschung: freie Antworten bewerten

Studenten antworten in eigenen Worten

Zwei Antworten können dasselbe meinen und ganz unterschiedlich formuliert sein.

Bei der Korrektur zählt, ob die Erklärung fachlich stimmt. Das Feedback soll Fehler verständlich machen.

Wie gut bewertet ein Sprachmodell?

Ein großes Sprachmodell (LLM) bewertet die Antwort und formuliert Feedback zur Aufgabe.

Ich untersuche, wann diese Bewertung stimmt und wie wir Fehler im Urteil erkennen.

In Loop2Learn erprobe ich den Ablauf, einschließlich der Übungen nach dem Feedback.

HTWK Hochschule für Technik, Wirtschaft und Kultur Leipzig
Web & Software Engineering Research Group

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

ASAG: kurze freie Antworten bewerten

Automatic Short Answer Grading

Kurze Antworten in eigenen Worten

Von einer Wortgruppe bis zu einem Absatz: Die Antwort soll fachliches Wissen ausdrücken.

Bewertet wird ihr Inhalt. Unterschiedliche Formulierungen können dieselbe Bewertung erhalten.

Automatische Bewertungen untersuchen

Ein Modell ordnet der Antwort eine Bewertung zu. Als Vergleich dienen menschliche Bewertungen.

Wie gut erkennt es die Bedeutung der Antwort? Bei welchen Antworten liegt es falsch?

Mein Schwerpunkt: ASAG mit großen Sprachmodellen und fachlichen Referenzen.

Burrows, Gurevych & Stein · IJAIED 2015, S. 60 bis 62 · DOI: 10.1007/s40593-014-0026-8

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

Was der LLM-Grader bekommt und zurückgibt

Eingabe
Aufgabe + Antwort
Erwartungshorizont + Kursregeln

>

Bewertung
Antwort mit den Kriterien
der Aufgabe abgleichen

>

Ausgabe
richtig / teilweise / falsch
Leistungsanteil + Feedback

Fehlt ein geforderter Teil, kann der Grader die Antwort als teilweise richtig bewerten.

Der Leistungsanteil liegt zwischen 0 und 1; er gibt keine Sicherheit des Modells an.

Loop2Learn-Bewertungsprompt und Waldmann-Kurskonfiguration · gelesen 2026-09-29

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

ICWE: Vergleich mit der Dozentenbewertung

ÜBEREINSTIMMUNG MIT DEM DOZENTEN

90,67 %

408 von 450 Antworten

Claude Haiku 4.5
mit 40 bewerteten Beispielen im Prompt

Bewertete Antworten im Prompt

Das Modell bekommt bewertete Antworten zur jeweiligen Frage als Beispiele. Wir testen es an anderen, zurückgehaltenen Antworten.

Mit 10 Beispielen erreicht es 90 %.
Wer jede Antwort als richtig bewertet, kommt auf 56 % (252/450).

Untersucht: 9 Aufgaben aus einem Kurs, ein Dozent, richtig/falsch.

Gwozdz & Both · ICWE 2026 · Tab. 1/3, §4.1 bis 4.3 · DOI: 10.1007/978-3-032-29372-5_8

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

Womit vergleichen wir das LLM-Urteil?

Mit der Bewertung des Dozenten

Bei freien Erklärungen vergleichen wir das Modellurteil mit der Bewertung des Dozenten und prüfen die Abweichungen.

Dabei halten wir fest, welches Modell und welcher Prompt für welche Aufgabe verwendet wurden.

Mit dem autotool-Prüfer

Für unterstützte Aufgaben prüft autotool, ob die eingereichte Lösung die Bedingungen erfüllt.

Dieses Urteil entsteht unabhängig vom LLM.

Die formale Prüfung sagt noch nicht, ob eine freie Erklärung fachlich gut ist.

Forschungsansatz; autotool-Integration · Belegstand 2026-09-29

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

Was wir für ein Modul festlegen

1. Welche Methode sollen die Studenten anwenden oder erklären? Danach wählen wir die Aufgabe.
2. Wir legen den autotool-Aufgabentyp, die konkrete Instanz und die erlaubte Antwortform fest.
3. LLM und autotool bewerten dieselbe formale Einsendung. Wir speichern beide Urteile.

Eine zusätzliche Erklärfrage prüft das Verständnis des Lösungswegs; sie wird separat bewertet.

HTWK Hochschule für Technik, Wirtschaft und Kultur Leipzig
Web & Software Engineering Research Group

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

Aus einer Aufgabe wird eine Verständnisfrage

Aufgabe zum Üben

Finde einen Aufruf mit `undefined`, der einen definierten Wert liefert.

```
oder x y = case x of
  False -> y
  True  -> True
```

Eine Lösung

```
oder True undefined
```

Ergebnis: `True`

Mögliche KI-Frage dazu

Warum liefert dieser Aufruf ein Ergebnis, obwohl ein Argument undefiniert ist?

Erwartete Erklärung

Das erste Argument ist `True`. Dieser Zweig liefert sofort `True`; das zweite Argument wird nicht ausgewertet.

Idee: autotool erzeugt die Aufgabe. Eine Lösung wird daran geprüft und dem LLM als Referenz gegeben. Der Dozent prüft die Erklärfrage und ihren Erwartungshorizont.

Lehrbeispiel, kein Generatorlauf · Generierung: Waldmann, ABP 2017, S. 2, 4, 7

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

Beispiele aus der Lehre

Lehrinhalt	Formale Aufgabe	Ergänzende Erklärung
Fortgeschrittene Programmierung Striktheit	Aufruf finden, der trotz `undefined` ein Ergebnis liefert	Welche Argumente werden ausgewertet, welche nicht?
Programmiersprachen / Compilerbau Typisierung	Typ eines Ausdrucks mit dem passenden Typprüfer untersuchen	Typableitung und Einschränkungen begründen
Weitere unterstützte Methoden	Für jede weitere Methode einen passenden Aufgabentyp und ein Eingabeformat wählen	Festlegen, was die Erklärung des Lösungswegs enthalten muss

Für jedes Lernziel brauchen wir einen passenden Aufgabentyp und Bewertungskriterien.

Waldmann-Kurskonfiguration: Strictness-Direct und TypeCheck-Direct · Stand 2026-09-29

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

Ein Ergebnis trotz undefined

```
f x y z = case (oder y x) of
  False -> y
  True  -> case (und z y) of
    False -> z
    True  -> False
oder x y = case x of { False -> y; True -> True }
und  x y = case x of { False -> False; True -> y }
```

Finde einen Aufruf mit mindestens einem `undefined`, der trotzdem ein definiertes Ergebnis liefert.

`f undefined True False` ergibt `False`: Das Argument `x` wird nicht benötigt.

Strictness-Aufgabe aus dem Waldmann-Kurs · LLM-Aufrufe 2026-09-29

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

LLM-Ergebnisse und autotool-Referenzfälle

Gespeicherte Modellantworten · 2026-09-29

offline fallback

Gültige Lösung

```
f undefined True False
```

autotool (historisch): **JA · Right**

LLM: **RICHTIG · 1 / 1**

Du hast mit „f undefined True False“ einen passenden Aufruf angegeben: `oder True undefined` ergibt `True`, danach ergibt `und False True` den Wert `False`, sodass das Ergebnis definiert ist, obwohl das erste Argument `undefined` ist. Damit ist die geforderte Nicht-Striktheit gezeigt.

Kein Ergebnis

```
f undefined undefined undefined
```

autotool (historisch): **NEIN · Semantics**

LLM: **FALSCH · 0 / 1**

Du hast die verlangte Form eines einzelnen Aufrufs eingehalten, aber bei `f undefined undefined undefined` wird schon `oder y x` ausgewertet und dafür zunächst `y` benötigt; das ist ebenfalls `undefined`, sodass das Ergebnis nicht definiert ist. Wähle stattdessen einen Aufruf, bei dem ein definiertes Argument die Auswertung zu einem Resultat führt, ohne das `undefined`-Argument zu benötigen. Im Kapitel zur Bedarfsauswertung hilft es, die Auswertungsreihenfolge der verschachtelten `case`-Ausdrücke nachzuverfolgen.

Falsche Form

```
f 1 2 3
```

autotool (historisch): **NEIN · Syntax**

LLM: **FALSCH · 0 / 1**

Du hast zwar die verlangte Form eines einzelnen dreistelligen Aufrufs getroffen, aber `1`, `2` und `3` gehören nicht zu den erlaubten Argumenten `undefined`, `True` und `False`. Wähle einen Aufruf mit mindestens einem `undefined`, dessen Auswertung dennoch einen Bool-Konstruktor liefert; prüfe dafür, welche Argumente durch die `case`-Ausdrücke tatsächlich ausgewertet werden.

LLM: Aufzeichnung 2026-09-29 · autotool: historische Fälle, damalige Instanz nicht archiviert

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

Freitext: Eine unvollständige Erklärung

Frage zum selben Programm: In welchen Argumenten ist `f` strikt? Begründe am Programm.

Eingereichte Antwort, gekürzt

„Eine Funktion ist strikt in einem Argument, wenn undefined an dieser Stelle auch das Ergebnis undefiniert macht.“

Die Antwort erklärt den Begriff allgemein. Welche Argumente von `f` strikt sind, beantwortet sie nicht.

Feedback des Graders

Teilweise richtig · Leistungsanteil 0,35 / 1

„Du gibst die Definition von Striktheit im Kern richtig wieder und erklärst Bedarfsauswertung sinnvoll. Es fehlt jedoch die Anwendung auf dieses konkrete `f`: Nenne, dass `f` im zweiten Argument `y` strikt ist und begründe das mit der zuerst ausgewerteten `case (oder y x)`-Auswertung. Außerdem musst du die Nicht-Striktheit für `x` und `z` jeweils mit einem konkreten Aufruf belegen; schau dir dazu an, welche Zweige des verschachtelten `case` ausgewertet werden.“

Feedback im Originalwortlaut.

GPT-6 Luna · Aufruf 2026-09-29 · synthetische Beispielantwort

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

Was wir als Nächstes untersuchen

Bisherige Beispiele

Die gespeicherten LLM-Antworten zeigen die Bewertung mit Aufgabe und Kursregeln. Modell und Zeitpunkt sind dokumentiert.

Zum Vergleich stehen historische autotool-Fälle zur Verfügung.

Offene Forschungsfragen

Welche Bewertungsfehler erkennen wir durch den Vergleich mit autotool?

Wie hängen Abweichungen von Modell, Prompt und Aufgabe ab?

Passen die erzeugten Fragen zum Lernziel, und hilft das Feedback beim Üben?

Für Aussagen über Verlässlichkeit und Lernwirkung brauchen wir eine systematische Untersuchung.

HTWK Hochschule für Technik, Wirtschaft und Kultur Leipzig
Web & Software Engineering Research Group

Jonas Gwozdz · jonas.gwozdz@htwk-leipzig.de

Vielen Dank

Freie Antworten mit LLMs bewerten.

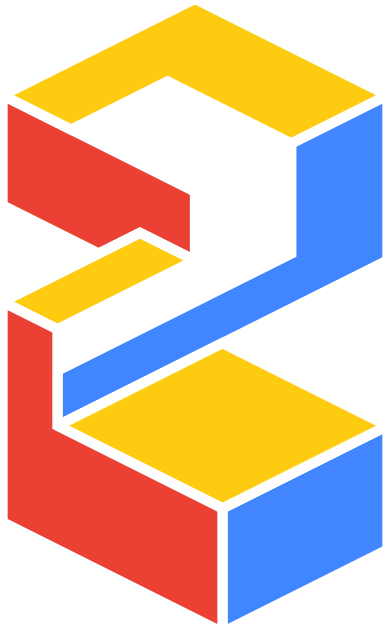
Formale Lösungen mit autotool prüfen.

Als Nächstes wollen wir Aufgaben mit geprüften Referenzlösungen bereitstellen und die Bewertungen für dieselben Instanzen vergleichen.

Jonas Gwozdz
Web & Software Engineering (WSE) Research Group
Hochschule für Technik, Wirtschaft und Kultur Leipzig (HTWK Leipzig)

jonas.gwozdz@htwk-leipzig.de

WSE Research · Webseite



Loop2Learn

Für Ihre Lehre

Loop2Learn ist einsetzbereit.

Sie möchten Loop2Learn einsetzen oder ausprobieren?

Ich freue mich sehr, wenn Sie sich bei mir melden!

jonas.gwozdz@htwk-leipzig.de

Jonas Gwozdz
Web & Software Engineering (WSE) Research Group
Hochschule für Technik, Wirtschaft und Kultur Leipzig (HTWK Leipzig)