Das LLM als Spiegel

Was Extraktionsfehler über das eigene Datenmodell verraten

Nicole Dresselhaus[1,2], Till Grallert[1,2]

Datum: 2026-11-05

[1] Humboldt-Universität zu Berlin
Berlin, Deutschland [2] NFDI4Memory
Berlin, Deutschland

Workshop zum Einsatz von KI in den historisch arbeitenden Geistes- und Sozialwissenschaften, Universität Regensburg, 5.–6. November 2026

Lizenz: CC BY-SA 4.0

Zusammenfassung:

HSozKult hat seit 1996 zehntausende Konferenzberichte publiziert. Wir extrahieren daraus mit einem lokal betriebenen LLM strukturierte Angaben zu einzelnen Beiträgen und publizieren Teile des Datensatzes auf Wikidata. Der Beitrag berichtet, was in diesem Workflow schiefging: Was wir für Extraktionsfehler des Sprachmodells hielten, waren fast immer Hinweise auf Lücken unseres Datenmodells. An drei Fällen zeigen wir, wie Anomalien in den Zwischendaten zu Modelländerungen führten und was daraus für Einsteiger:innen folgt.

Schlüsselwörter:

Informationsextraktion · Large Language Models · Datenmodellierung · Quellenkritik · Wikidata · Digital History

1 Worum es geht

Im Rahmen eines Field Surveys zur Geschichte computationeller Methoden in den Geschichtswissenschaften bildet die graue Literatur von Konferenzberichten und -programmen einen Einblick in die Entwicklung von Themen und Methoden, der sich mit etablierten Publikationsformaten in dieser zeitlichen Granularität und thematischen Breite nicht abbilden lässt (Hohls 2004; Meyer 2018). HSozKult hat sich seit 1996 als zentrales Informationsportal der deutschsprachigen Geschichtswissenschaften etabliert und in drei Jahrzehnten zehntausende Konferenzberichte und -programme publiziert. Wir extrahieren daraus strukturierte Angaben zu einzelnen Konferenzbeiträgen — Vortragstitel, Beitragende, Rollen, Orte, Item-Typen und Statusinformationen —, klassifizieren die Beiträge später nach Themen und Methoden und publizieren Teile des Datensatzes als Linked Open Data auf Wikidata.

Unser Workflow besteht aus sechs Schichten vom HTML-Crawl über die LLM-gestützte Extraktion bis zum Wikidata-Export (Anhang B). Wir konzentrieren uns im Folgenden auf die vierte Schicht und die Reibung mit den Schichten davor und danach.

Der wichtigste Befund aus der Projektarbeit ist eine Verschiebung des Blicks: Was wir für Extraktionsfehler des Sprachmodells hielten, waren fast immer Hinweise auf Lücken unseres Datenmodells. Drei dieser Lücken stellen wir unten vor.

2 Warum ein LLM — und was das LLM nicht entscheidet

Zu Beginn haben wir regelbasiert extrahiert. HSozKult-Berichte folgen einem redaktionellen Rahmen, aber Positionierung, Formatierung und sprachliche Gestaltung der Programm- und Beitragsangaben variieren über drei Jahrzehnte erheblich; diese Heterogenität war in früheren Analysen von Subsets bereits sichtbar (Burckhardt 2017). Regelbasierte Ansätze scheiterten an dieser Variabilität; der Aufwand wuchs exponentiell gegenüber dem Ertrag. Ein LLM erkennt demgegenüber semantische Muster, ohne auf stabile Formatierungsregeln angewiesen zu sein, und eignet sich dadurch für die Extraktion aus lose strukturierten Quellen (vgl. Hiltmann u. a. 2025; Baránek 2026).

Technisch läuft die Extraktion über mehrteilige Prompts, die Berichtstext, erwartetes Ausgabeformat (ein JSON-Schema) und explizite Extraktionsregeln übergeben;[1] Prompt-Version, Modell und Rohantwort werden archiviert.[2] Entscheidend ist die Rolle des LLM im Workflow: Es bleibt eine Black Box — was in ihm vorgeht, versuchen wir nicht nachzuvollziehen —, aber wir übernehmen seine Ausgabe nicht ungeprüft. Sie ist eine von mehreren Reconciliation-Quellen und durchläuft manuelle QA und Normalisierung, bevor daraus publizierbare Records und Exportstatements werden.

Das eigentliche Arbeitsmittel ist dabei nicht der Prompt, sondern das Schema, das er befüllt: Pro Tagungsbericht entsteht eine Liste von Konferenz-Items, die Item-Typ (Vortrag, Poster, Panel, Keynote, Roundtable), Beitragende und deren Rollen, Statusinformation, ein Freitextfeld für Unsicherheit und einen Konfidenzwert getrennt ausweist. Der Prompt liefert dafür nur Zulieferungen. Unserer Erfahrung nach sind Halluzinationen in der Regel der Versuch des LLM, Anforderungen aus einem Prompt in ein dafür nicht passendes Schema zu zwingen — und genau hier bietet das bewusste Freitextfeld die nötige Flexibilität, diesen Widerspruch auszudrücken.

Diese Schichtung ist kein Luxus, sondern eine Konsequenz des Gegenstands. Modellierung ist keine rein technische Angelegenheit, sondern eine epistemische Praxis: Jede Entscheidung,

[1] Sinngemäße, gekürzte Rekonstruktion des Prompts in Anhang D; der zeilengenaue Wortlaut steht im Repository (`data_clio_conferences/app/dumper/Main.hs`).

[2] Verwendet wird ein lokal betriebenes Modell in 4-Bit-Quantisierung über `llama.cpp`; die Quelltexte verlassen die Projektinfrastruktur nicht, und die Rohantworten bleiben als Projektartefakte prüfbar.

einen Namen als `author` oder einen Programmabschnitt als `presentation` zu erfassen, ist eine interpretative Setzung, die festlegt, welche Aspekte einer Quelle sichtbar werden (Drucker 2011; Ciula u. a. 2023). Daten sind immer schon modelliert; die quellenkritische Reflexion verschiebt sich damit von der Textkritik zur Kritik des Datenmodells (Dommann 2012; Fickers 2020). Zugleich gilt die Warnung aus der Literaturwissenschaft: Was robust ist, ist oft trivial, und was interessant ist, oft nicht robust (Da 2019); ein Datenmodell, das vorab festlegt, was zählbar ist, produziert genau dieses Muster. Wir legen das Modell deshalb nicht a priori fest, sondern behandeln die LLM-Ausgabe als Fuzz-Test der eigenen Kategorien und entwickeln diese damit induktiv:[3] Ihre Gleichgültigkeit gegenüber unseren[4] Modellierungskonventionen erzeugt systematisch Fälle, die wir von Hand nie gezielt gesucht hätten. Entscheidend ist die veränderte Arbeitsteilung: Die computationelle Bearbeitung des Korpus läuft im Hintergrund und kostet „nur“ Rechenzeit (ohne die Umweltkosten auszublenden). Auch die Anpassung des Prompts kostet nur Minuten aktiver Arbeit; der eigentliche Aufwand bleibt die Revision des Schemas. Früher erlaubte der manuelle Aufwand meist nur einen Durchgang durch den Datensatz: Was in der vorab gezogenen Stichprobe (Anhang A) durchfiel, ließ sich später kaum korrigieren. Heute kommen zur schnellen Iteration auf der Stichprobe komplette Durchgänge hinzu, wenn wider Erwarten doch noch Probleme mit dem Schema auftreten. Die Architektur muss die Fälle also dort sichtbar machen, wo Menschen sie noch sehen können.

[3] Das Prinzip ist aus dem Software-Testen bekannt: Werkzeuge wie `fuzz` erzeugen systematisch Eingaben, die gezielte manuelle Tests nicht abdecken, und legen damit Randfälle offen (Miller, Fredriksen, und So 1990); die testgetriebene Entwicklung formalisiert dasselbe Prinzip für das Design (Beck 2002). Der Unterschied liegt in den Fehlerkosten: keine Rechenzeit, sondern Quellenarbeit.

[4] Dies bedeutet *nicht*, dass LLMs keine inhärenten Biases haben — im Gegenteil. Es wurde lediglich nicht auf *unsere* Konventionen und Biases trainiert.

3 Was wir messen können — und was nicht

Eine Vollprüfung aller bisher extrahierten Beiträge übersteigt unsere personellen Möglichkeiten, eine ungeprüfte Erfolgsmeldung wäre methodisch nicht vertretbar. Wir haben deshalb eine stratifizierte Stichprobe manuell gegen die Quellen geprüft: 20 Berichte, gleichmäßig über fünf Zeiträume von Dezember 1996 bis September 2025 verteilt, mit 304 extrahierten Konferenz-Items. Für die publizierbare Teilmenge (Vortrag, Keynote, Workshop, Panel, Roundtable; 273 Items, von denen 272 bewertbar waren) ergab sich eine nutzbare Präzision von 94,5 % (257 von 272; 95-%-Konfidenzintervall nach Wilson: 91,1–96,6 %). Die Rate schädlicher Fehler — falsch akzeptierte oder falsch zugeordnete Items — lag bei 1,10 % (3 von 272). Kein einziges Item war vollständig erfunden, obwohl Fabrikationen bei LLM-gestützter Datenextraktion als gängiges Problem dokumentiert sind (Huang u. a. 2025). Eine ergänzende Recall-Prüfung fand 1 von 274 publizierbaren Items nicht (99,6 %), publizierte aber 16 Items, die inhaltlich zwar korrekt waren, deren Ereignistyp sich außerhalb unseres Interesses (Scope) befand (Begrüßungsworte, Exkursionen).

Wir lesen diese Zahlen als Plausibilitätsprüfung, nicht als Genauigkeitsmessung: Die Stichprobe ist klein, die Bewertung erfolgte durch eine Person, das Design erlaubt keine Inferenz auf die Grundgesamtheit. Aufschlussreicher als die Quote ist die Fehlerart. Alle drei schädlichen Fehler waren entweder falsch zugeordnete Beitragende — die Items existierten, wurden aber der falschen Person zugeschrieben — oder falsch klassifizierte Programmeinheiten. Es sind Fehlertypen aus der Struktur der Quelle — unklare Programmzeilen, Namen im Fließtext —, nicht aus einer Eigenheit des Modells.

Aufschlussreicher als die Zahl ist das Raster dahinter: Wir unterscheiden strikte von nutzbarer Korrektheit — wörtliche Übereinstimmung gegenüber oberflächlicher Abweichung bei gleicher Bedeutung — und trennen davon schädliche Fehler sowie fehlende Items.

4 Drei Lücken, die unser Datenmodell verändert haben

4.1 Die Rollen-Lücke: Moderation ist kein Beitragstyp

In einer frühen Phase extrahierte der Prompt alle im Bericht genannten Personen und ordnete Programmzeilen zu; Funktionsbezeichnungen wie „Moderation“ wurden in den Extraktionsregeln explizit ausgefiltert. Das war zweckmäßig, solange Personenextraktion das Ziel war. Der Wechsel der Analyseeinheit von Personen zu Konferenz-Items machte sichtbar, dass das Schema kein konzeptuelles Feld für Contributor-Rollen jenseits von `author` besaß: Eine Sitzung mit einer Moderatorin und drei Vortragenden wurde zu einem Item vom Typ „Moderation“ mit vier `author`-Einträgen. Der Fehler saß nicht im Prompt, sondern im Modell — die Kategorien erzwangen eine Gleichsetzung von Rolle und Beitragstyp. Die Lösung war ein struktureller Umbau: separate, vom Item-Typ unabhängige Contributor-Rollen, getrennt in administrative Rollen (Moderation, Chair, Diskussion) und substantielle Beitragstypen (Vortrag, Poster, Keynote).

4.2 Die Status-Lücke: Wenn die KI selbst die Lücke meldet

Ein bewusst eingebautes `note`-Feld diente als Eskalationsmechanismus: Ließ die Quellenlage keine robuste Kategorisierung zu, sollte das LLM seine Unsicherheit im Freitext dokumentieren, statt sie in eine scheinbar präzise Kategorie zu pressen. In einem Fall meldete es dort einen ausgefallenen Vortrag: `note: "ausgefallen"`. Damit wurde ein blinder Fleck sichtbar — unser bis dahin implizites Modell behandelte alle extrahierten Items als gehalten und kannte keinen kontrollierten Status für geplante, aber nicht realisierte Beiträge. Die Statuskategorien `held`, `cancelled`, `planned` und `unclear` entstanden nicht aus einer theoretischen Vorüberlegung, sondern als Reaktion auf eine von der KI selbst gemeldete Anomalie. Die methodische Lektion reicht über diesen Einzelfall hinaus: Gezielt eingeräumte Freitext-Ausweichstellen für KI-Unsicherheit sind kein Eingeständnis von Schwäche, sondern ein legitimes Instrument. Sie zeigen, welche Kategorien das Modell noch nicht abbildet, und verhindern, dass epistemische Unsicherheit in vorzeitiger struktureller Eindeutigkeit verschwindet.

Der Kanal blieb danach in Gebrauch, und die Auswertung der Antwortdaten zeigt, wohin das führt: Von 153.761 extrahierten Items tragen 7,2 % eine Notiz. Rund 30 % dieser Notizen folgen wiederkehrenden Mustern — Programm ohne Bestätigung im Bericht (19 % der Notizen), aus

dem Bericht abgeleitete Titel (6 %), Vortragsformate wie Abendvorträge (5 %), Absagen (2 %); der Rest sind individuelle Einzelbemerkungen. Ein Teil davon ist inzwischen strukturell abgebildet: Die Programm/Bericht-Differenz steckt heute im Statusvokabular (4,8 % aller Items sind als `planned` ausgewiesen), und jeder extrahierte Wert trägt einen Konfidenzwert. Nicht abgebildet ist dagegen die Herkunft eines Titels: ob er wörtlich aus dem Programm übernommen oder aus dem Bericht abgeleitet wurde, steht bis heute nur in der Notiz. Ein Freitextkanal für KI-Unsicherheit bleibt also nicht Freitext: Wo Muster wiederkehren, gehören sie ins Schema, sonst wandert dieselbe Unterscheidung bei jedem Lauf erneut in die Notiz.

4.3 Die Quelle/Objekt-Lücke: Wikidata als Prüfinstanz

Der geplante Wikidata-Export erzwang eine Unterscheidung, die in den lokalen JSON-Zwischendaten unsichtbar geblieben war: Trägt ein Statement die Quellen-URL des Tagungsberichts (`P973`), muss der Sprach-Qualifier (`P407`) die Sprache des Quellartefakts beschreiben, nicht die Sprache des beschriebenen Objekts. Ein deutscher Bericht über eine englischsprachige Konferenz erfordert daher `P407 → Deutsch`, nicht Englisch. Diese Unterscheidung wurde erst durch den Linked-Data-Export sichtbar und durchsetzbar: Wikidata fungiert hier nicht als passives Publikationsziel, sondern als aktive Modellierungsfläche, die vorgelagerte konzeptuelle Ungenauigkeiten aufdeckt (Sohmen, Rossenova, und Blümel 2026; Grallert u. a. 2026).

5 Warum das skaliert

Unser Korpus umfasst 10.578 Tagungsberichte mit 153.761 extrahierten Items; manuell sichten lässt sich das nicht. Die Notizen wirken deshalb als Aufmerksamkeitsfilter: 7,2 % der Items tragen eine Aussage, die das Modell selbst für bemerkenswert hält, und nach dem Clustering bleiben rund 3.300 Items mit wiederkehrenden Mustern — gut zwei Prozent des Datensatzes. Aus 153.000 Fällen wird so eine Menge, die man tatsächlich ansehen kann: „hier könnte etwas nicht stimmen“ wird zum Prüfauftrag statt zur Haltung.

Der Filter arbeitet dabei mit der Quelle selbst: auslesen, sortieren, zählen. Was obenauf schwimmt, sind Kandidaten für Schema- und Prompt-Änderungen — bei uns sind daraus die Statuskategorien und später `substituted` geworden. Unter *Prompt* verstehen wir die Regelformulierung für das Schema; das Schema selbst steht im Code (Anhang D). Dass die Regeln der Regler sind, lässt sich an unserem Material messen: Für denselben Stichprobensatz von rund 120 bewusst ausgewählten, quellenkritisch schwierigen Berichten (Programmlage unklar, Titel nur im Prosabericht) liegen zwei aufeinanderfolgende Prompt-Fassungen vor, die sich nur in den Regeln unterscheiden, nicht im Schema. Die neuere verlangt verbindlich, dass jeder Programmpunkt im Ergebnis erscheint — im Zweifel als `planned`. Das Ergebnis: 8 % mehr Items (1.880 auf 2.036), doppelt so viele `planned`-Ausweise (101 auf 200) und weniger Notizen (294 auf 272; 15,6 % → 13,4 % der Items). Die Unterscheidung wandert also aus der Notiz in das Feld — und die Quote sinkt messbar mit. Damit ist dieser Scan der Prüfstand für jede weitere Prompt-Änderung. Der volle Korpus zeigt zusätzlich, wie viel davon schon erledigt ist: 22 % der Notizen stammen aus der Prompt-Regel für Programm/Bericht-Abweichungen, 15 % wiederholen nur den gesetzten Status und sind als „schon adressiert“ ignorierbar (Anhang E).

6 Was wir Einsteiger:innen raten

1. **Analyseeinheit vor dem ersten Prompt festlegen** — Person, Item oder Aussage? Ein Wechsel der Einheit ist eine Modelländerung und keine Prompt-Änderung.
2. **Schema explizit machen und versionieren.** Ein JSON-Schema, das Rollen, Item-Typen, Status und Konfidenz trennt, kostet einen Nachmittag und spart Wochen.
3. **Eine Ausweichstelle für Unsicherheit einbauen.** Freitext ist kein Rauschen, sondern ein Datum — und oft der einzige Ort, an dem eine fehlende Kategorie auffällt.
4. **Ausgabe als Hypothese behandeln.** Kein Pfad führt direkt von der Rohantwort in die Publikation; Review und Normalisierung gehören in die Architektur, nicht in die Nachbereitung.
5. **Stichprobe statt Vollkontrolle:** stratifiziert ziehen, Fehlertypen zählen statt nur eine Quote berichten und die Stichprobe dokumentieren, damit Dritte sie nachvollziehen können.

Der Aufwand liegt nicht im Prompting. Rechnen Sie mit Tagen für Schemaarbeit und nur mit Minuten für Prompts.

7 Zur Diskussion

Drei Fragen, die über unser Projekt hinaus für Einsteiger:innen relevant sein dürften:

1. Wie viel Schema-Design muss vor der ersten Abfrage stehen und wie viel darf sich aus der Inspektion der Zwischenergebnisse entwickeln? Wir neigen inzwischen zu „so wenig a priori wie möglich, so viel wie nötig“ — das kostet aber Zeit, die in Projekten mit knappen Ressourcen zuerst fehlt.
2. Sind Freitext-Ausweichstellen für KI-Unsicherheit ein Workaround oder ein methodisches Instrument — und wie dokumentiert man sie so, dass Dritte die Entscheidung nachvollziehen können?
3. Downstream-Exporte wie Wikidata erzwingen konzeptuelle Präzision, die die Quelle selbst nicht hergibt. Wann ist diese Präzision quellenkritisch vertretbar und wann eine Überinterpretation?

8 Offenlegung zum Einsatz generativer KI

Bei der Erstellung dieses Kurzbeitrags wurden generative Sprachmodelle unterstützend eingesetzt: zur Strukturierung von Textabschnitten auf Basis der Projekt- und Git-Dokumentation, zur Literaturverifikation per DOI-Abgleich und zur Konsistenzprüfung zwischen zitierten Quellen und den im Text aufgestellten Aussagen. Die Extraktion selbst nutzt ein lokal betriebenes Modell; die Prüfstichprobe wurde manuell durchgeführt. Alle Textvorschläge wurden von den Autor:innen geprüft, überarbeitet und verantwortet.

9 Anhang

A: Stichprobe und Prüfzahlen

Kennzahl	Wert
Stichprobe	20 Berichte, fünf Zeit-Buckets, Dezember 1996 – September 2025
Extrahierte Konferenz-Items	304
Publizierbare Teilmenge	273 Items (davon 272 bewertbar)
Nutzbare Präzision	94,5 % (257/272; 95-%-KI nach Wilson: 91,1–96,6 %)
Schädliche Fehler	1,10 % (3/272)
Vollständig halluzinierte Items	0
Recall (publizierbare Items)	99,6 % (1 von 274 nicht gefunden)
Überpubliziert	16 Items außerhalb der Publikationskategorien

Tabelle 1: *Ergebnisse der stratifizierten manuellen Prüfung.*

Die Beispiele des Beitrags sind strukturtreu anonymisiert: Namen, Titel und Institutionen wurden durch synthetische Platzhalter ersetzt, die strukturellen Beziehungen — welches Feld welche Art von Fehler enthielt, welche Schema-Komponente versagte — sind unverändert.

B: Die sechs Verarbeitungsschichten

#	Schicht	Artefakt	Prüfpunkt
1	Source-Capture	HTML-Cache	Quellen-URL und Abrufdatum
2	Strukturierte Quellenextraktion	Berichtstext und Metadaten als JSON	Vergleich mit dem HTML
3	Metadaten-Reconciliation	abgeglichene Orte und Normdaten-QIDs	manuelle Annotation und erneuter Lauf
4	LLM-gestützte Extraktion	Rohantwort, Prompt-Version, Modell	Archivierung, keine Direktpublikation
5	Review und Normalisierung	Konferenz-Item-Records	manuelle QA, stratifizierte Stichprobe
6	Export	Wikidata-Statements, QuickStatements	Mapping- und Qualifier-Regeln

Tabelle 2: *Die sechs Schichten der Pipeline.*

C: Die drei Fälle im Überblick

Fall	Symptom im Zwischenergebnis	Modellierungsentscheidung
Rollen-Lücke	Sitzung als Item „Moderation“ mit vier `author`	Rollen vom Item-Typ getrennt
Status-Lücke	`note: "ausgefallen"` im Freitextfeld	Statusvokabular `held`, `cancelled`, `planned`, `unclear`
Quelle/Objekt-Lücke	Sprach-Qualifier am Quellen-Statement	`P407` = Sprache der Quelle, nicht des Objekts

Tabelle 3: *Die drei Modellierungslücken und ihre Auflösung.*

D: Der Extraktionsprompt (gekürzt)

Der Kasten folgt dem Aufbau des Aufrufs: Aufgabe, zwei der Regeln, an denen wir im Verlauf der Arbeit gearbeitet haben, Ausgabeformat — und am Ende die Eingabe, die dem Modell nach den Regeln angehängt wird. Sinngemäß gekürzt und nicht zeilengenau.

```
# AUFGABE
Extrahiere Konferenz-Items aus zwei deutschsprachigen Eingaben; antworte
ausschließlich mit einem JSON-ARRAY, ein Element pro Konferenz-Item.

# REGEL: VOLLSTÄNDIGKEIT UND STATUS          <- Stellschraube
Jedes Item des PROGRAMMS muss im Ergebnis erscheinen, jeweils mit Status:

    im BERICHT bestätigt                        -> held
    im Bericht ausdrücklich abgesagt            -> cancelled
    Inhalt anders geliefert (Ersatz, Zusammenfassung, Diskussion) -> substituted
    nur ein Abstract eingereicht                -> abstract_only
    im Programm, im Bericht nicht erwähnt       -> planned
    Quellen widersprüchlich                     -> unclear

Im Zweifel `planned`; Items nicht stillschweigend weglassen.

# REGEL: TITEL UND NOTIZFELD                <- Stellschraube
Hat das PROGRAMM einen klaren Titel, wird er 1:1 in `title` übernommen;
`title` ist kein Ort für Umformulierung.
`note` ist nur für das, was kein anderes Feld ausdrücken kann:
    - eine Ambiguität der Quelle
    - eine relevante Rollenunsicherheit
    - Details eines Status, die der Statuswert allein nicht trägt
Niemals einen Status, einen Typ oder eine Titelentscheidung in `note` wiederholen.

# AUSGABEFORMAT   (jedes Feld: { "content": ..., "score": 0-10 })
[ { "title":  { "content": "...", "score": 8 },
    "type":   { "content": "presentation", "score": 9 },
    "status": { "content": "held", "score": 7 },
    "contributors": [ { "name": { "content": "..." }, "role": { "content": "presenter" },
                        "location": { "content": "..." } } ],
    "note": "..." } ]

# EINGABE   (wird nach den Regeln an das Modell angehängt)
PROGRAMM:
    <Programmzeilen aus dem HTML, oft ein Programmpunkt pro Zeile>

BERICHT:
    <Berichts-Text>   <- hier wird der Tagungsbericht als Fließtext eingefügt
```

E: Herkunft der Notizen im vollen Korpus

Notiz-Herkunft	Notizen	Anteil aller Notizen	Status-gedeckt
Abweichung Programm/Bericht oder Absage (Prompt-Regel, dedupliziert)	2.401	21,8 %	1.614
Herkunft eines Titels (kein Feld)	655	5,9 %	8
Vortragsform (Abend-/Eröffnungsvortrag)	547	5,0 %	36
Ersatz für abgesagten Beitrag	109	1,0 %	12
ohne erkennbares Muster	7.505	68,0 %	479

F: Literatur

Baránek, Daniel. 2026. „Machine Learning Assisted Conversion of Biographical Records Into Wikidata Triples“. *Journal of Open Humanities Data* 12. https://doi.org/10.5334/johd.466.

Beck, Kent. 2002. *Test-Driven Development: By Example*. Boston: Addison-Wesley.

Burckhardt, Daniel. 2017. „Comparing Disciplinary Patterns: Exploring the Humanities through the Lens of Scholarly Communication“. *Digital Humanities Quarterly* 11 (2). https://doi.org/10.63744/3brxyxzxj3jj.

Ciula, Arianna, Øyvind Eide, Cristina Marras, und Patrick Sahle. 2023. „Model and Modelling in Digital Humanities“. In *Modelling Between Digital and Humanities*, 19–42. Open Book Publishers. https://doi.org/10.11647/obp.0369.01.

Da, Nan Z. 2019. „The Computational Case against Computational Literary Studies“. *Critical Inquiry* 45 (3): 601–39. https://doi.org/10.1086/702594.

Dommann, Monika. 2012. „Schlüssel für Clio: Digitale Daten und die Suche nach einer neuen Quellenkritik“. *SAGW Bulletin*, Nr. 1: 56–57. https://doi.org/10.5167/uzh-77381.

Drucker, Johanna. 2011. „Humanities Approaches to Graphical Display“. *Digital Humanities Quarterly* 5 (1). https://doi.org/10.63744/r4ysrh7ae534.

Fickers, Andreas. 2020. „Update für die Hermeneutik: Geschichtswissenschaft auf dem Weg zur digitalen Forensik?“ 2020. https://doi.org/10.14765/zzf.dok-1765.

Grallert, Till, Sophie Eckenstaler, Claus-Michael Schlesinger, Nicole Dresselhaus, Isabell Trilling, und Sophie Stark. 2026. „Open Tool Registries! Resolving the Directory Paradox with Wikidata“. *Digital Humanities Quarterly* 20 (1). https://doi.org/10.63744/c6fmeb495g4w.

Hiltmann, Torsten, Martin Dröge, Nicole Dresselhaus, Till Grallert, Melanie Althage, Paul Bayer, Sophie Eckenstaler, u. a. 2025. „NER4all or Context is All You Need: Using LLMs for Low-Effort, High-Performance NER on Historical Texts. A Humanities Informed Approach“. https://arxiv.org/abs/2502.04351.

Hohls, Rüdiger. 2004. „H-Soz-u-Kult: Kommunikation und Fachinformation für die Geschichtswissenschaften“. *Historical Social Research* 29 (1): 212–32. https://doi.org/10.12759/hsr.29.2004.1.212-232.

Huang, Lei, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, u. a. 2025. „A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions“. *ACM Transactions on Information Systems* 43: 1–55. https://doi.org/10.1145/3703155.

Meyer, Thomas. 2018. „H-Soz-Kult und Clio-online: Von der Mailingliste zur Online-Community“. *Bibliotheksdienst* 52 (3–4): 185–95. https://doi.org/10.1515/bd-2018-0026.

Miller, Barton P., Lars Fredriksen, und Bryan So. 1990. „An Empirical Study of the Reliability of UNIX Utilities“. *Communications of the ACM* 33 (12): 32–44. https://doi.org/10.1145/96267.96279.

Sohmen, Lucia, Lozana Rossenova, und Ina Blümel. 2026. „Wikidata 4 Open Culture: Lessons Learned from Hands-On Work with Cultural Heritage Data in the Expanded Wikibase Ecosystem“. *Journal of Open Humanities Data* 12: 52. https://doi.org/10.5334/johd.440.

G: Förderung

Diese Arbeit entstand im Rahmen des NFDI-Konsortiums 4Memory (https://www.4memory.de). Wir danken der Deutschen Forschungsgemeinschaft (DFG) für die finanzielle Unterstützung — Projektnummer 501609550.