

Anreicherung

- [Einleitung OpenRefine](#)
- [Grundlegende OpenRefine Funktionen](#)
- [Reconciliation mit OpenRefine](#)
- [LLM Einbindung in OpenRefine](#)
- [Externe Ressourcen für OpenRefine](#)

Einleitung OpenRefine

Die Informationen auf dieser Seite beziehen sich auf die zum Zeitpunkt März 2026 aktuelle Version **OpenRefine 3.10.0**. Die beschriebenen Schritte wurden unter **Linux** ausgeführt. Betriebssystemspezifische Hinweise für **Windows** und **macOS** finden sich in der offiziellen [OpenRefine?Dokumentation](#).

Metadatenanreicherung beschreibt den Prozess, digitale Inhalte (Bilder, Dokumente, Datensätze) durch Hinzufügen strukturierter Informationen, wie IDs, Geodaten oder Beschreibungen, aufzuwerten. Diese Anreicherung verbessert die Auffindbarkeit, Verwaltung und Vernetzung der Daten.

Ein wichtiger Bestandteil dieses Prozesses ist die Anbindung an **kontrollierte Vokabulare** oder **Linked-Open-Data-Datenbanken** (z.?B. Wikidata, GND). OpenRefine bietet dafür integrierte Funktionen zur **Datenbereinigung** und zum **Abgleich (Reconciliation)** mit externen Quellen.

OpenRefine (ehemals Google Refine) ist eine OpenSource Software zur Bearbeitung tabellarischer Daten. Es wird als *"A power tool for working with messy data. - Ein leistungsstarkes Werkzeug für die Arbeit mit unstrukturierten Daten."* beschrieben.

Typische Anwendungsbereiche umfassen:

- Analysieren großer und "unordentlicher" Datensätze
- Korrigieren von Fehlern
- Anreichern von Daten mit externen Informationen
- Normalisieren und Vereinheitlichen von Daten
- Transformieren und Strukturieren von Daten

Obwohl OpenRefine im Webbrowser ausgeführt wird, benötigt die Anwendung grundsätzlich keine Internetverbindung. Alle Daten werden lokal auf dem jeweiligen Rechner verarbeitet und nicht online übertragen. Eine Ausnahme bilden Funktionen, die auf externe Dienste zugreifen – etwa Reconciliation Services oder bestimmte Extensions wie die LLM?Erweiterungen.

OpenRefine erstellt beim Import eine lokale Kopie der Daten in Form eines Projekts. Die Originaldateien bleiben dadurch unverändert. Soll dasselbe Projekt auf mehreren Rechnern weiterbearbeitet werden, muss es zunächst exportiert und anschließend auf die jeweiligen Geräte übertragen werden.

Daten bleiben lokal auf dem Rechner und werden nicht ins Internet übertragen. Nur Funktionen wie **Reconciliation Services** oder bestimmte **Extensions** (z.?B. LLM?Erweiterungen) benötigen eine Online?Verbindung.

Installation von OpenRefine

Die Installationsdateien für OpenRefine können über die [offizielle Webseite](#) heruntergeladen werden. Unter Linux wird in der Regel eine ZIP-Datei bereitgestellt, die am gewünschten Installationsort entpackt wird.

Zum Ausführen von OpenRefine wird ein **Java Runtime Environment (JRE)** benötigt. Hinweise zum Download und zur Installation finden sich unter: <https://www.java.com>

Um OpenRefine unter Linux zu starten, navigieren wir im Terminal in den entpackten OpenRefine-Ordner und führen den folgenden Befehl aus:

```
./refine
```

Dadurch öffnet sich ein neues Fenster im Standardbrowser, in dem OpenRefine ausgeführt wird. Empfohlen werden **WebKit-basierte Browser** wie *Opera*, *Google Chrome* oder *Microsoft Edge*, da es unter **Firefox** zu bekannten Performanceproblemen kommen kann. *Internet Explorer* wird nicht unterstützt.

Soll OpenRefine nicht im Standardbrowser geöffnet werden, kann es im Headless-Mode gestartet werden:

```
./refine -x refine.headless=true
```

Anschließend lässt sich OpenRefine in jedem beliebigen Browser über folgende Adresse öffnen:

```
http://127.0.0.1:3333
```

Nach dem Start zeigt das Hauptmenü die verwendete OpenRefine-Version und bietet verschiedene Optionen. Projekte können erstellt, geöffnet oder importiert werden. Zudem kann die Sprache geändert oder eine Übersicht der installierten Extensions aufgerufen werden. Wir belassen die Benutzeroberfläche in **Englisch**, um eine bessere Kompatibilität mit vorhandenen Lernmaterialien sicherzustellen. Daher werden auch in dieser Dokumentation überwiegend die englischen Bezeichnungen verwendet. OpenRefine wird durch die Tastenkombination **Strg + C** im Terminal gestoppt, dabei werden alle Änderungen automatisch gespeichert.

Installation von Extensions

OpenRefine kann über unterschiedliche Erweiterungen (Extensions) individuell an spezifische Arbeitsabläufe angepasst werden. Eine Übersicht der verfügbaren Extensions gibt es unter: <https://openrefine.org/extensions>

Es gibt zwei Möglichkeiten, Extensions zu installieren:

- **Installation im OpenRefine-Programmordner**
 - Die Erweiterung steht nur dieser konkreten OpenRefine-Installation zur Verfügung.
 - Bei Updates von OpenRefine muss sie erneut installiert werden.

- **Installation im persönlichen Workspace?Ordner**

- Die Erweiterung ist für alle OpenRefine?Installationen des Benutzers verfügbar.
- Updates von OpenRefine haben in der Regel keinen Einfluss auf die installierten Extensions.

Im Folgenden beschreiben wir die Installation im **Programmordner**.

Manche Erweiterungen verfügen über eigene, spezifische Installationsanleitungen, die unbedingt beachtet werden sollten. Grundsätzlich ist die Installation von Extensions in OpenRefine sehr einfach:

1. ZIP?Datei der gewünschten Extension herunterladen.
2. Den Ordner `webapp/extensions` im OpenRefine?Installationsverzeichnis lokalisieren.
3. Die heruntergeladene ZIP?Datei in diesem Ordner **entpacken**.
4. OpenRefine **starten**.

Eine Extension wurde erfolgreich installiert, wenn ihre zusätzlichen Funktionen in OpenRefine sichtbar bzw. nutzbar sind. Details dazu finden sich in der jeweiligen Dokumentation der Erweiterung.

Grundlegende OpenRefine Funktionen

Erstellen eines OpenRefine Projekts

Um Daten in OpenRefine bearbeiten zu können müssen diese als neues OpenRefine-Projekt importiert werden. Generell kann OpenRefine eine Vielzahl an möglichen Formaten importieren. Das [OpenRefine Handbuch](#) bietet hierfür eine genauere Beschreibung.

Um ein Projekt zu erstellen, öffnen wir auf der Startseite von OpenRefine das Menü **Create Project**.

Unter **Get data from** wählen wir die Option **This Computer**.

Alternativ können Daten auch über eine URL geladen, aus der Zwischenablage eingefügt oder aus einer Datenbank importiert werden.

Über **Choose Files** wählen wir anschließend unsere Datei aus und klicken auf **Next >>**, um die Import-Vorschau zu öffnen.

Im **Parsing-Menü** lassen sich verschiedene Einstellungen festlegen, die bestimmen, wie OpenRefine die Datei einliest.

Hier kann beispielsweise die Option **Trim leading & trailing whitespace from strings** aktiviert werden, um überflüssige Leerzeichen am Anfang oder Ende eines Textfeldes automatisch zu entfernen.

Welche Einstellungen sinnvoll sind, hängt vom jeweiligen Datensatz ab. Die Vorschau erleichtert es, die passenden Optionen auszuwählen.

Der **Projektname** kann individuell angepasst werden, und das Erstellen von **Tags** ermöglicht eine leichtere Navigation in der Projektübersicht.

Sind alle Einstellungen vorgenommen, wird das Projekt über **Create Project >>** erstellt.

OpenRefine Datentypen

OpenRefine bietet vier Datentypen:

- **string** (ein oder mehr Textzeichen)
- **number** (ein oder mehr Zahlen)
- **boolean** ("true" oder "false" Werte)

- **date** (ISO-8601-compliant extended format with time in UTC: YYYY-MM-DDTHH:MM:SSZ)

Die Typen **number**, **date**, und **boolean** werden grün angezeigt. **Strings** werden schwarz angezeigt. Sollten Werte, welche eigentlich eine Zahl repräsentieren schwarz angezeigt handelt es sich dabei um **strings** welche erst in das **number** Format konvertiert werden müssen. Funktionen dazu finden sich in *Column Dropdown* ? **Edit cells > Common transforms > To [...]**.

Zusätzlich hat OpenRefine noch die Datentypen **error** für Fehler und **null** für leere Zellen.

Wichtig: Die möglichen Facetten für eine Spalte hängen vom jeweiligen Datentyp ab.

Layout

Ähnlich wie die meisten Tabellenbearbeitungsprogramme strukturiert OpenRefine die Daten in Spalten (Columns) und Reihen (Rows). Die Spalten enthalten dabei die einzelnen Merkmale (Features), während die Reihen die einzelnen Dateneinträge repräsentieren.

Auf die meisten Funktionen in OpenRefine lässt sich über die Dropdown-Menüs zugreifen, die sich am oberen Rand jeder Spalte befinden.

OpenRefine bietet zwei Optionen zum Anzeigen der Daten.

- **Reihen-Ansicht** in der jede Reihe als eigenständiger Eintrag gesehen wird
- **Record-Ansicht** in der Reihen zu einem zusammenhängenden Eintrag zusammengefügt werden.

Unter **Show as:** kann durch auswählen von **rows** oder **records** zwischen den verschiedenen Ansichten gewechselt werden. Die ausgewählte Ansicht hat Einfluss auf das Verhalten von OpenRefine. Durch die Record-Ansicht ist bei Daten, die mehrere Zeilen pro Objekt enthalten eine konsistente Bearbeitung zusammenhängender Informationen möglich.

Records in OpenRefine

OpenRefine gruppiert mehrere Zeilen zu einem *Record*, wenn diese logisch zusammengehören, zum Beispiel, wenn ein Objekt mehrere Schlagwörter, mehrere Personen oder unterschiedliche Datenelemente besitzt. Jede Zeile innerhalb eines Records stellt dabei einen einzelnen Wert dar.

OpenRefine entscheidet anhand der Werte in der **linken Spalte**, ob eine neue Zeile zu einem bestehenden Record gehört oder einen neuen Record beginnt.

Steht in einer Zeile in einer der linken Spalten ein Wert, beginnt OpenRefine einen **neuen Record**. Bleiben diese Zellen leer, wird die Zeile dem **vorherigen Record** zugeordnet. Dabei ist die erste Spalte ausschlaggebend für die Gesamtstruktur. Soll jede Reihe im Dokument als eigener Records gesehen werden, muss die erste Spalte vollständig gefüllt sein.

Spalten ordnen und umbenennen

Es gibt zwei Möglichkeiten, Spalten in OpenRefine neu zu ordnen:

1. Über das Dropdown-Menü der Spalte „All“

Unter *All Dropdown* ? **Edit columns > Re-order / remove columns...** können alle Spalten gleichzeitig neu angeordnet oder entfernt werden.

Dabei öffnet sich ein separates Fenster, in dem die Spalten per Drag&Drop verschoben werden können. Durch das Verschieben einer Spalte in den rechten Bereich wird sie aus der Ansicht entfernt.

2. Über das Dropdown-Menü einer einzelnen Spalte

Jede Spalte kann auch individuell bewegt werden.

Dazu wählen wir *Column Dropdown* ? **Edit column** und anschließend eine der entsprechenden **Move column [...]** Optionen.

Diese Änderungen werden direkt auf das Datenset angewendet.

Über *Column Dropdown* ? **Edit column > Rename this column...** können Spalten umbenannt werden, und über *Column Dropdown* ? **Edit column > Remove this column** lassen sie sich löschen.

Die baumartige Record-Struktur in OpenRefine ergibt sich aus der Anordnung der Spalten von links nach rechts.

Eine Veränderung der Spaltenreihenfolge kann direkten Einfluss auf die Struktur und Gruppierung der Records haben.

Wird eine Spalte vorübergehend nicht benötigt, soll aber für später erhalten bleiben, kann sie ausgeblendet werden.

Über *Column Dropdown* ? **Hide / Show** stehen verschiedene Optionen zum Aus- und Einblenden zur Verfügung.

Spalten vereinigen und trennen

OpenRefine bietet außerdem leistungsstarke Funktionen, um Spalten aufzuteilen oder zusammenzuführen. Über *Column Dropdown* ? **Edit column** lassen sich sowohl **Split into several columns...** als auch **Join columns...** auswählen.

- **Split into several columns...:**

Diese Funktion eignet sich, wenn eine Zelle mehrere Informationen enthält, die durch ein Trennzeichen (z.B. Komma, Semikolon oder Leerzeichen) getrennt sind. OpenRefine erstellt daraus automatisch mehrere neue Spalten.

Zusätzlich kann festgelegt werden, wie viele neue Spalten maximal erstellt werden sollen.

- **Join columns...:**

Diese Option kann verwendet werden, wenn Informationen aus mehreren Spalten in einer einzigen Spalte zusammengeführt werden sollen. Dabei lässt sich ein optionales Trennzeichen definieren (z.B. Komma oder Semikolon), das zwischen die zusammengeführten Werte gesetzt wird. Die Spalten, die zusammengeführt werden sollen, können direkt im Menü ausgewählt werden. Dabei lässt sich festlegen, wie Null-Werte behandelt werden sollen. Außerdem kann bestimmt werden, ob die zusammengeführten Inhalte in einer neuen Spalte gespeichert oder in eine bestehende Spalte geschrieben werden sollen. Optional besteht auch die Möglichkeit, die ursprünglichen Spalten nach dem Zusammenführen zu löschen.

Diese Werkzeuge sind besonders hilfreich, um komplexe oder falsch strukturierte Daten in ein konsistentes und besser analysierbares Format zu bringen.

Zellen vereinigen und trennen

Ähnlich wie bei Spalten können auch Zellen innerhalb einer Spalte zusammengeführt oder aufgeteilt werden.

In *Column Dropdown* ? **Edit cells** finden wir die Optionen **Split multi-valued cells...** und **Join multi-valued cells...**. Dabei lässt sich jeweils ein gewünschtes Trennzeichen (Separator) festlegen.

Für diese Operationen ist die Record-Struktur von OpenRefine besonders wichtig.

- Wenn wir **Zellen aufteilen**, erzeugt OpenRefine zusätzliche Zeilen innerhalb desselben Records. Dadurch wird der Record *tiefer* und er enthält mehr Zeilen und damit mehr einzelne Werte.
- Wenn wir **Zellen zusammenführen**, fasst OpenRefine mehrere Zeilen bzw. Werte zu einer einzigen Zelle zusammen. Dadurch wird der Record *flacher*, weil mehrere Einträge zu einem Wert verschmelzen.

Diese Struktur beeinflusst direkt, wie Facets, Filter und Reconciliation-Prozesse funktionieren.

Eine saubere Record-Struktur ist daher entscheidend, um die Daten später korrekt analysieren und weiterverarbeiten zu können. Insbesondere bei hierarchischen oder gruppierten Daten sind *Column Dropdown* ? **Fill down** und *Column Dropdown* ? **Blank down** wichtige Werkzeuge.

- **Fill down:**

Diese Funktion füllt leere Zellen mit dem nächstoberhalb liegenden Wert.

Sie eignet sich vor allem, wenn Informationen in einer Spalte nur einmal pro Record angegeben sind

und auf die darunterliegenden Zeilen übertragen werden sollen.

- **Blank down:**

Das Gegenstück zu Fill Down. Damit lassen sich Zellen unterhalb eines Werts gezielt leeren, sodass nur der erste Eintrag einer Gruppe sichtbar bleibt.

Dies ist nützlich, wenn redundante Informationen entfernt oder die Datenstruktur für Record?Ansichten bereinigt werden soll.

Sortieren

Über die Option *Column Dropdown* ? **Sort...** können wir die Zeilen unserer Daten sortieren.

Dabei lassen sich verschiedene Kriterien festlegen, wie der OpenRefine?Datentyp, eine auf- oder absteigende Sortierung sowie die Position leerer oder fehlerhafter Zellen.

Nach dem Bestätigen der Auswahl erscheint oberhalb der Daten ein neues **Sort** ?Dropdown?Menü.

Dort können wir die Sortierung wieder entfernen oder die Reihenfolge umkehren.

Im Gegensatz zu anderen Tabellenbearbeitungsprogrammen wie Excel wird eine Sortierung in OpenRefine nicht automatisch dauerhaft übernommen.

Erst durch die Auswahl von **Sort > Reorder rows permanently** wird die aktuelle Sortierung tatsächlich im Datensatz gespeichert.

Undo/Redo

OpenRefine bietet ein unbegrenztes Undo/Redo-System. Das bedeutet, dass jederzeit sämtliche Bearbeitungsschritte zurückgenommen oder wiederhergestellt werden können.

Werden Schritte rückgängig gemacht und danach neue Schritte ausgeführt, werden die zuvor bestehenden Schritte dauerhaft überschrieben und können **nicht** wiederhergestellt werden.

Deshalb empfiehlt es sich, vor experimentellen Änderungen die bisherigen Bearbeitungsschritte zu exportieren und zu sichern.

Ein weiteres starkes Feature von OpenRefine ist die Möglichkeit, sämtliche durchgeführten Schritte als Workflow zu exportieren. Diese gespeicherten Workflows können anschließend auf andere Projekte angewendet werden. Hierbei gibt es jedoch Einschränkungen, da unter anderem Spaltennamen übereinstimmen müssen.

Filter

Manchmal möchten wir nur Zeilen anzeigen, die bestimmte Werte enthalten. Dafür bietet OpenRefine eine Filtermöglichkeit.

Wir wählen dazu *Column Dropdown* ? **Text filter**. Dadurch erscheint im Bereich **Facet / Filter** eine Eingabezeile, in die wir Suchbegriffe eingeben können.

Der Textfilter funktioniert ähnlich wie eine klassische Suchfunktion. Es kann nach einzelnen Buchstaben oder ganzen Wörtern gefiltert werden. Darüber hinaus lassen sich **Regular Expressions (Regex)** einsetzen oder die Suche mit **Groß- und Kleinschreibung** (case sensitive) beschränken.

Der Filter kann außerdem invertiert werden, sodass nur die Zeilen angezeigt werden, die **nicht** dem Suchkriterium entsprechen.

Wenn ein Filter aktiv ist, wirken sich alle weiteren Bearbeitungsschritte ausschließlich auf die aktuell gefilterten Zeilen aus. Zeilen außerhalb des Filters bleiben davon unberührt.

Facetten

Facetten gehören zu den zentralen Analyse- und Navigationswerkzeugen in OpenRefine. Sie ermöglichen es, die Werte einer Spalte strukturiert auszuwerten und gleichzeitig als Filter zu verwenden.

Eine Facette erstellt eine Übersicht aller vorkommenden Werte oder Wertbereiche und zeigt an, wie viele Zeilen zu jedem dieser Einträge gehören. Dadurch lassen sich Muster, Ausreißer, Tippfehler oder Inkonsistenzen besonders leicht erkennen.

OpenRefine stellt verschiedene Arten von Facetten zur Verfügung:

- **Text Facet:** listet alle eindeutigen Textwerte einer Spalte auf. Dadurch ist sie ideal für Kategorien, Schlagwörter oder Namen.
- **Numeric Facet:** gruppiert numerische Werte in übersichtliche Intervalle.
- **Timeline Facet:** ermöglicht die zeitliche Analyse, wenn Datumswerte vorliegen.
- **Scatterplot Facet:** erzeugt ein zweidimensionales Diagramm aus zwei numerischen Spalten und zeigt Zusammenhänge, Cluster oder Ausreißer visuell an. Im Diagramm können auch direkt Bereiche zum Anzeigen ausgewählt werden.
- **Custom Facet:** erlaubt eigene Filterlogik mithilfe von GREL-Ausdrücken.
- **Customized Facets:** bieten spezifische bereits implementierte Facet-Optionen, wie Wortfacetten, oder Facetten für leere Werte.

Sie lassen sich über *Column Dropdown* ? **Facet >** aufrufen. Durch auswählen einer Option erscheint ein neues Fenster in der **Facet \ Filter** Übersicht.

Mit **Reset all** können wir alle aktiven Facetten und Filter in den Originalzustand versetzen.

Mit **Remove all** können wir alle aktiven Facetten und Filter entfernen.

Neben der Analyse dienen Facetten auch als interaktive Filter:

Durch Anklicken eines Werts oder Intervalls wird nur der entsprechende Teil der Daten angezeigt und kann gezielt weiterbearbeitet werden.

Komplexere Analysen lassen sich durch die Kombination mehrerer Facetten durchführen, wie etwa das gleichzeitige Filtern nach einem bestimmten Schlagwort und einem bestimmten Datumsbereich.

Ein wesentlicher Vorteil von Facetten ist, dass **Werte direkt innerhalb der Facette bearbeitet werden können**.

Über die Option „**Edit**“ neben einem Facetteneintrag lassen sich einzelne Werte umbenennen, korrigieren oder zusammenführen (Cluster).

Diese Änderungen wirken sich unmittelbar auf alle Zeilen aus, die diesen Wert enthalten, zum Beispiel, um Tippfehler zu korrigieren oder ähnliche Begriffe zu vereinheitlichen.

Ähnlich wie bei Filter werden weiterführende Bearbeitungsschritte nur auf die derzeit ausgewählten Zeilen angewendet.

Facetten sind somit ein wesentliches Werkzeug, um Datenqualität sichtbar zu machen, Fehlerquellen aufzuspüren, Duplikate zu identifizieren und große Datensets effizient zu strukturieren.

Clustering

Clustering ist eines der wichtigsten Werkzeuge in OpenRefine, um ähnliche oder fehlerhaft geschriebene Werte automatisiert zu erkennen und zu bereinigen. Das Clustering-Menü erreichen wir über *Column Dropdown* ?

Edit cells > Cluster and edit.... Dort öffnet sich ein Dialogfenster, das verschiedene Algorithmen zur Gruppierung ähnlicher Werte anbietet. Besteht bereits eine Facette für eine Spalte, können wir auch einfach über den Button **Cluster** im Facetten-Fenster in das Menü gelangen.

OpenRefine unterstützt mehrere eingebaute Cluster-Methoden, darunter:

Key-Collision-Methoden

Diese Methoden erzeugen aus jedem Wert einen „Schlüssel“, sodass ähnlich strukturierte Werte denselben Schlüssel erhalten.

- **Fingerprint:** Normalisiert Werte (Kleinschreibung, Entfernen von Sonderzeichen) und sortiert Wörter alphabetisch. Gut für typische Tippfehler.
- **N-gram Fingerprint:** Nutzt n-Gramme (Buchstabengruppen), um auch weniger offensichtlich ähnliche Wörter zu erkennen.

- **Cologne Phonetic:** Deutsches phonetisches Verfahren, bei dem Wörter nach ihrem Klang gruppiert werden.
- **Metaphone3:** Phonetische Verfahren für englische Wörter und internationale Begriffe.
- **Daitch-Mokotoff:** Phonetisches Matching für slawische und yiddische Begriffe.
- **Beider-Morse:** Komplexes Verfahren für matchen von phonetisch äquivalenten Begriffen.

Nearest-Neighbor-Methoden

Diese Methoden vergleichen Werte direkt anhand von Ähnlichkeitsmetriken:

- **Levenshtein Distance:** Misst Unterschiede zwischen zwei Zeichenketten (Einfügen/Löschen/Ersetzen).
- **PPM (Prediction by Partial Matching):** Modelliert Ähnlichkeiten probabilistisch; gut bei unstrukturierten Textvarianten.

Eigene Clustering-Methoden

OpenRefine kann um benutzerdefinierte Cluster-Algorithmen erweitert werden. Über Erweiterungen oder eigene Skripte (z. B. in **Python** oder Java) lassen sich zusätzliche Verfahren integrieren, insbesondere projektspezifische Ähnlichkeitsmetriken oder spezielle Sprachmodelle.

Im **Clustering-Dialog** können wir:

- erkannte Gruppen ähnlicher Werte anschauen
- Zielwerte bearbeiten oder vereinheitlichen
- Cluster zusammenführen
- Änderungen für einzelne oder mehrere Gruppen übernehmen

Dafür können folgende Arbeitsschritte durchgeführt werden:

1. Über *Column Dropdown* ? **Edit cells > Cluster and edit...** öffnen.
2. Oben im Dialog eine Cluster-Methode auswählen (z. B. „Fingerprint“ oder „Cologne Phonetic“).
3. OpenRefine zeigt nun Gruppen ähnlicher Werte an, die potenziell vereinheitlicht werden können.
4. Für jeden Cluster kann ein **neuer Zielwert** eingegeben werden (z. B. ein korrekter, vereinheitlichter Begriff).
5. Mit **„Merge Selected & re-cluster“** werden die gewählten Werte zusammengeführt.
6. Den Vorgang mit anderen Methoden wiederholen, um weitere Varianten zu entdecken.

Clustering eignet sich besonders gut, um **Tippfehler**, **Schreibvarianten**, **uneinheitliche Bezeichnungen**, **Groß-/Kleinschreibung** oder andere Inkonsistenzen effizient zu korrigieren und Daten für Reconciliation oder Analyse vorzubereiten.

Clustering ist häufig ein **iterativer Prozess**. Oft sind mehrere Clustering-Durchgänge mit verschiedenen Clustering-Methoden notwendig um alle ähnlichen Worte zu finden und zu vereinheitlichen.

Transformations

Transformationen ermöglichen es, Werte in einer Spalte gezielt anzupassen, umzuwandeln oder neu zu strukturieren.

In OpenRefine stehen dafür zwei zentrale Wege zur Verfügung: die allgemeine **Transform-Funktion** sowie die vordefinierten **Common Transforms**.

Transform-Funktion (freie Transformation)

Wir erreichen die freie Transformationsumgebung unter *Column Dropdown* ? **Edit cells > Transform...**. Hier können wir eigene Ausdrücke in **GREL (General Refine Expression Language)** verwenden, um Werte flexibel zu bearbeiten. Dadurch ist es möglich Texte umzuwandeln, Zeichen zu extrahieren, Listen zu erzeugen oder Datumsangaben zu normalisieren.

GREL unterstützt zusätzlich **Regex (Regular Expressions)**, mit denen sich komplexe Mustersuchen und Ersetzungen durchführen lassen. Dabei werden aber nicht alle Regex-Funktionen unterstützt und müssen an GREL adaptiert werden.

Beispiele für häufige GREL Ausdrücke finden sich [hier](#).

Common Transforms Unter *Column Dropdown* ? **Edit cells > Common transforms** befinden sich häufig benötigte Standardaktionen. Dazu gehören zum Beispiel:

- Entfernen von überflüssigen Leerzeichen
- Text in Groß-/Kleinschreibung umwandeln
- Zahlen oder Datumswerte parsen
- Sonderzeichen normalisieren

Diese Funktionen lassen sich mit einem Klick auf die gesamte Spalte anwenden und bieten einen schnellen Einstieg in die Datenbereinigung.

Nützliche Hinweise zur Arbeit mit Transformationen

- Transformationen wirken immer auf die ausgewählte Spalte.
- Komplexere Anpassungen entstehen häufig durch die Kombination von Common Transforms und GREL-Ausdrücken.
- Da GREL Ergebnisse sofort in einer Vorschau zeigt, können Transformationen gefahrlos ausprobiert werden.

Transformationen entfalten ihr volles Potenzial, wenn man einfache Schritte (Trim, Case-Conversion) mit gezielten GREL-Ausdrücken kombiniert. Dadurch lassen sich selbst komplexe Bereinigungsprozesse effizient lösen.

Daten exportieren

Nachdem wir unsere Daten bereinigt und erweitert haben, wollen wir sie für die Weiterverwendung **exportieren**. Dies können wir über die verschiedenen Optionen im **Export** Menü erledigen.

Originaldaten bleiben unverändert

OpenRefine lädt die Quelldaten als **lokale Projektkopie**.

Die **Originaldateien** werden während der Bearbeitung **nicht verändert** und müssen zur weiteren Verwendung exportiert werden.

Hinweis zur Google?Sheets?Unterstützung

Die OpenRefine?Dokumentation (Stand: Januar 2026) erwähnt eine Google?Sheets? Integration.

Diese Unterstützung wurde jedoch seit **OpenRefine 3.9** aus **Sicherheitsgründen entfernt**.

Sie kann über eine Extension wieder aktiviert werden.

Verfügbare Exportziele

OpenRefine stellt mehrere **Dateiformate** für den direkten Export bereit (insbesondere tabellarische Formate):

- **TSV/CSV, HTML, Excel (XLSX), ODF**
- Zusätzlich kann das **OpenRefine?Projekt** selbst exportiert werden (z.?B. zur Weitergabe).

Die meisten Exporte berücksichtigen die **aktuelle Ansicht**, also **aktive Facets und Filter**. Es werden in der Regel **nur die angezeigten Zeilen** exportiert.

Custom Tabular Export

Für erweiterte Anforderungen, insbesondere bei **reconcilierten Daten**, bietet OpenRefine den **Custom Tabular? Export**:

- Pro Spalte lässt sich wählen, ob **Originalwerte**, **gematchte Werte** oder **gematchte IDs** ausgegeben werden.
- Optional können **leere Zellen** für *no?match* erzeugt oder **Links zu Entitätsseiten** ausgegeben werden.

Wichtig: Diese Einstellungen werden **spaltenweise** konfiguriert.
Jede relevante Spalte muss separat korrekt eingestellt werden.

Über „**Download**“ und die „**Preview**“-Schaltfläche lässt sich eine **Vorschau der ersten Zeilen** anzeigen und feinjustieren.

Im Reiter „**Option Code**“ kann die **JSON?Konfiguration** der gewählten Einstellungen **kopiert** (für spätere

Exporte wiederverwendet) oder **importiert** werden.

Weitere Exportoptionen

- **SQL?Export** (zum Import in Datenbanken)
- Mit der **RDF?Extension** sind Exporte als **RDF/XML** oder **Turtle** möglich.

Export als JSON (Templating Exporter)

OpenRefine besitzt **keine direkte „Export ? JSON“**-Funktion. Stattdessen verwenden wir **Export > Templating...**.

Der **Templating Exporter** bietet maximale Flexibilität:

- Zugriff auf **Reconciliation?Felder**
- **Spalten aufteilen/zusammenführen**
- **GREL?Ausdrücke** anwenden (Ausdrücke in **doppelten geschweiften Klammern** `{{ ... }}` wird als GREL interpretiert, alles außerhalb als Text)
- Optional sind auch **Jython** oder **Clojure** möglich, z.?B.: `{{jython:return cells['ColumnName'].value}}`

Verschiedene **Vorlagen** für, zum Beispiel, YAML oder geojson finden sich im [OpenRefine Github](#).

Praxis?Tipp zum JSON?Templating

Records werden beim Export **nicht automatisch** als geschachtelte Strukturen ausgegeben.

Komplexe Strukturen (z.?B. Arrays mit Bedingungen) erfordern **GREL?**

Kontrollstrukturen wie `forEach`, `filter`, `if(...)`, etc. Für erste Schritte lohnt es sich, die **Verschachtelung gering zu halten** und **Feineinstellungen** anschließend in einem externen Editor vorzunehmen.

Reconciliation mit OpenRefine

Für die Erstellung der Beispiele wurde eine adaptierte Version von Metadaten von Kulturpool-Objekten verwendet. Die Inhalte wurden so verändert, dass sie typische Anwendungsfälle und Bearbeitungsschritte in OpenRefine besser veranschaulichen. Sie entsprechen daher nicht den Originaldaten.

Reconciliation ermöglicht es, Dateneinträge mit externen Wissensquellen abzugleichen und anzureichern. Es wird die Reconciliation anhand von existierenden IDs (z.B.: Wikidata QID, GND, AAT, ...) gezeigt, als auch die Reconciliation über Textwerte.

Reconciliation Service hinzufügen

Zugriff auf die Reconciliation-Funktionen erhalten wir über *Column Dropdown* ? **Reconcile**. Das Reconciliation-Menü finden wir dabei unter *Column Dropdown* ? **Reconcile > Start reconciling...**. Dabei öffnet sich nun ein Fenster in welchem wir die Services verwalten können. Standardmäßig ist bereits die englische Version der Wikidata-API konfiguriert. Um einen anderen Service hinzuzufügen wählen wir **Add standard service...**. Dort fügen wir die URL des Service ein um ihn hinzuzufügen. Über **Discover services...** gelangen wir zur [Reconciliation service test bench](#), wo wir nicht nur eine Auswahl an verfügbaren Services finden, sondern auch eine Möglichkeit um APIs zu testen.

Auswahl an Services

- **Wikidata Deutsch:** `https://wikidata.reconci.link/de/api`
- **GND:** `https://lobid.org/gnd/reconcile` oder `https://reconcile.gnd.network`
- **Getty Vocabularies Reconciliation Service:** `https://services.getty.edu/vocab/reconcile/`
- **Bionomia:** `https://api.bionomia.net/reconcile`
- **GeoNames from FornPunkt:** `https://fornpunkt.se/apis/reconciliation/geonames`

Conciliator und RDF Extension

Für manche Reconciliation-Workflows in OpenRefine reicht ein einzelner externer Dienst nicht aus oder es fehlt eine direkte Schnittstelle. In solchen Fällen kann **Conciliator** eine hilfreiche Ergänzung sein. Dabei handelt es sich um einen lokal laufenden Reconciliation-Server, der verschiedene Quellen unterstützt und als flexible Middleware zwischen OpenRefine und externen Datenquellen dient.

Conciliator bietet **out-of-the-box Unterstützung** für Dienste wie **VIAF**, **ORCID**, **Open Library** sowie für **beliebige Apache-Solr-Collections**. Darüber hinaus stellt das Projekt ein Java-Framework zur Verfügung, mit dem sich eigene Reconciliation-Services entwickeln lassen, wenn institutionelle Daten oder Spezialvokabulare

angebunden werden sollen.

Der Dienst wird in einem **Docker?Container lokal ausgeführt**. Dazu wird das GitHub?Repository heruntergeladen und der Container entsprechend der Anleitung aufgebaut. Nach dem Start kann ein neuer Reconciliation?Service in OpenRefine konfiguriert werden, beispielsweise:

- **VIAF:** `http://localhost:8080/reconcile/viaf`
- **ORCID - Smart Names:** `http://localhost:8080/reconcile/orcid/smartnames`

Weitere Informationen zur Installation, zur Nutzung zusätzlicher Services sowie zur Entwicklung eigener Reconciliation?Endpunkte finden sich im offiziellen [Repository](#)

Mit der [RDF Extension](#) können wir aber auch SPARQL-Endpoints für die Reconciliation verwenden. Eine Anleitung wie die Reconciliation API für **DBpedia** konfiguriert werden kann findet man [hier](#).

Reconciliation über Zellenwerte

Das standardmäßige Vorgehen beim Abgleich von Daten mit externen Quellen besteht darin, die vorhandenen Zellenwerte über eine Textsuche mit passenden Einträgen der jeweiligen Datenbank zu matchen. Reconciliation?Services analysieren dabei den Inhalt einer Zelle, senden diesen als Suchanfrage an eine externe Wissensquelle und bewerten anschließend, welche der zurückgelieferten Treffer am wahrscheinlichsten zu dem Eintrag passen.

Reconciliation mit Wikidata

Ein häufig genutzter Service hierfür ist **Wikidata**, eine offene, kollaborativ gepflegte Wissensdatenbank, die strukturierte Informationen zu Personen, Orten, Werken, Konzepten und vielen weiteren Entitäten enthält. Durch seine Vielzahl verknüpfter IDs eignet sich Wikidata besonders gut, um Daten aus unterschiedlichen Systemen zusammenzuführen.

Die Reconciliation überschreibt den sichtbaren Zellwert. Eine Spaltenkopie erleichtert spätere Vergleiche.

Um eine Spalte mit Wikidata abzugleichen können wir folgende Schritte durchführen:

1. *Column Dropdown ?* **Start reconciling...**
2. Wikidata reconcil.link (de) wählen. Falls dieser noch nicht vorhanden ist, über **Add standard service..** mit dem URL `https://wikidata.reconcil.link/de/api` hinzufügen.
3. Den passenden Wikidata Typ auswählen.
4. **Auto-match candidates with high confidence** auswählen.
5. Mit **Start reconciling...** bestätigen.

Wikidata Typen

Es wird empfohlen für die Reconciliation mit Wikidata zuerst einen detaillierteren Typen zu wählen und ihn über mehrere Durchgänge an Abgleichen zu erweitern. Es besteht auch die Möglichkeit nach einem bestimmten Typ zu suchen oder gegen keinen bestimmten Typ abzugleichen.

OpenRefine kennzeichnet gematchte Entitäten durch eine dunkelblaue Schriftfarbe und der Option `Choose new match` unterhalb. Wurde für einen Wert noch kein Match ausgewählt erscheint der Wert in schwarz mit einer Liste an möglichen Begriffen. Stimmt einer der Vorschläge überein kann dieser durch klicken auf das Feld mit einem Haken nur auf die jeweilige Zelle angewendet werden. Durch Auswahl der zwei Haken wird das Matching für die gesamte Spalte übernommen.

Generell ist bei der Reconciliation auf potenzielle Namensgleichheit und damit einhergehende **False Positives** zu achten. **False Negatives** durch unterschiedliche Schreibweisen und Fehler sind ebenfalls möglich.

Der Wikidata Reconciliation Service ermöglicht es durch überfahren des Begriffs mit der Maus weitere Infos in einem Pop-up Fenster aufscheinen zu lassen. Stimmt keiner der Vorschläge überein, oder wurden keine potenziellen Kandidaten gefunden, kann über `Search for match` auch nach Kandidaten gesucht werden.

Da Wikidata eine offene und kollaborative Wissensdatenbank ist, besteht hier generell auch die Möglichkeit neue Begriffe hinzuzufügen, falls kein Match für einen relevanten Begriff gefunden wird. Hierfür muss aber zuerst festgestellt werden, dass der Begriff tatsächlich noch nicht existiert um Duplikate zu vermeiden.

Relevante Informationen nutzen

Zusätzlich zur Suche mit Textabgleich können wir auch andere relevante Informationen aus unseren Daten für den Abgleich verwenden. Im Reconciliation Menü besteht dafür die Option *Columns* als *As property* zu übergeben. Wollen wir, zum Beispiel, Künstler mit Wikidata abgleichen und haben zusätzlich zu den Namen auch eine Spalte mit Geburtsdaten, können wir diese Spalte als Property **P569** (Geburtsdatum) an Wikidata übergeben.

Mögliche Properties können über das jeweilige `As property`-Feld oder in den jeweiligen Reconciliation Service Dokumentation gefunden werden. In Wikidata ist es hilfreich ähnliche Entitäten auf Wikidata zu suchen und die vorhandenen Properties zu erkunden.

Property Paths

In Wikidata ist es auch möglich sogenannte `Property Paths` zu übergeben. Sind für die Künstler nur die Geburtsjahre verfügbar können wir diese als `SPARQL: P569@year` übergeben. Dadurch werden die Werte direkt

auf die relevante Wikidata Information gematched.

Property Paths können die Dauer einer Reconciliation-Anfrage erheblich verlängern. Daher sollten Property Paths **sparsam** und nur dann eingesetzt werden, wenn sie für das gewünschte Matching wirklich notwendig sind.

Reconciliation über vorhandene IDs

IDs aus vorhandenen URLs extrahieren

In manchen Daten befinden sich bereits URLs, die auf externe Identifikatoren verweisen (z.B. GND, AAT, VIAF). Wir wollen daraus zunächst konsistente ID-Werte extrahieren. Dazu können wir die Transform-Funktion über *Column Dropdown* ? **Edit cells > Transform...** nutzen. Haben wir z.B. URLs in der Form `http://vocab.getty.edu/aat/300264737` können wir mit folgendem GREL Ausdruck, die IDs inklusive jeweiliger Prefixes erhalten.

```
value.replace(/^.*?(gnd|aat|ulan|tgn|ulan|viaf|wikidata)\\[^\|/?#\]+).*$/,"$1/$2")
```

AAT-IDs können wir nun direkt mit dem Getty Vocabularies Reconciliation Service abgleichen:

1. Über einen Textfilter „aat“ selektieren
2. *Column Dropdown* ? **Reconcile > Use values as identifiers...**
3. Reconciliation?Service: *Getty Vocabularies Reconciliation Service*
4. Bestätigen mit **OK**

Für GND-IDs müssen wir zuerst das Prefix „gnd/“ entfernen.

Dies kann über Filter + Transformation erfolgen. Danach können wir analog zu den AAT-IDs mit dem GND Reconciliation Service den abgleich durchführen.

Externe IDs mit Wikidata abgleichen

Wikidata bietet die Möglichkeit, eine Vielzahl unterschiedlicher Identifikatoren über einen einzigen Reconciliation ?Service abzugleichen. Dadurch lassen sich Daten aus verschiedenen Quellen miteinander verknüpfen und zu einem zusammenhängenden Netzwerk von Werten verbinden. Dieser Ansatz ermöglicht es, bestehende IDs wie GND, VIAF oder AAT gemeinsam zu nutzen, um präzisere Matches zu erzielen und Datensätze effizient miteinander in Beziehung zu setzen. Dies funktioniert über sogenannte **Wikidata-Properties**. Damit können wir Werte gezielt gegen Properties von Wikidata Entitäten abgleichen und somit bessere Ergebnisse erzielen.

Mithilfe von **Wikidata-Properties** können wir verschiedene IDs gegen einen einzigen Service abgleichen und somit eine bessere Verbindung der Werte herstellen.

Da AAT und GND in Wikidata unterschiedliche Properties nutzen, trennen wir die IDs in eigene Spalten.

Zuerst erstellen wir eine neue Spalte für die **AAT-IDs**. Dies machen wir über *Column Dropdown ?* **Edit column**

> **Add column based on this column...** und folgendem GREL-Ausdruck:

```
if(value.startsWith("aat/"), value, null)
```

Anschließend geben wir der Spalte einen passenden neuen Namen. Da für unsere **GND-IDs** im vorherigen Schritt das Prefix bereits entfernt wurde können wir nun folgenden Ausdruck verwenden um eine neue Spalte für die GND-IDs zu erstellen:

```
if(value.contains("/"), "", value)
```

Um die AAT Property in Wikidata verwenden zu können, müssen zuerst der *aat/* Prefix vor den IDs entfernt werden. Dazu verwenden wir eine Transformation und den GREL-Ausdruck:

```
value.replace("aat/", "")
```

Nachdem die Daten nun entsprechend vorbereitet wurden, kann die Reconciliation mit Wikidata durchgeführt werden:

1. *Column Dropdown ?* **Start reconciling...**
2. Wikidata reconcili.link (de) wählen.
3. Den passenden Wikidata Typ auswählen
4. **Auto-match candidates with high confidence** auswählen
5. Zusätzliche Properties übergeben. Dazu wählen können wir in der Spalte **Also user relevant details from other columns:** folgende Properties auswählen:

Column	As property
GND-ID	P227
AAT-ID	P1014

6. Mit **Start reconciling...** bestätigen.

Nun können wir wie zuvor die Matches und Vorschläge kontrollieren und ergänzen oder ändern.

Reconciliation Values nutzen

Wir führen Abgleiche mit externen Wissensdatenbanken durch um unsere Daten mit zusätzlichen Informationen anzureichern. OpenRefine bietet dafür verschiedene Möglichkeiten um die Ergebnisse der Reconciliation zu nutzen. Sobald eine Zelle erfolgreich reconciliert wurde, stehen neben dem eigentlichen Wert zusätzliche Informationen zur Verfügung — etwa die eindeutige Identifikationsnummer (Q-ID bei Wikidata), der bevorzugte Label-Name, der Matching-Score oder alternative Kandidaten. Diese angereicherten Informationen können wir auf unterschiedliche Weise verwenden:

- **Entitäten-IDs hinzufügen:** Mit *Column Dropdown* ? **Reconcile > Add entity identifiers column...** lässt sich eine neue Spalte mit den IDs der gematchten Entität erzeugen, die sich ideal für Weiterverarbeitung oder Verknüpfungen eignet.
- **Weitere Eigenschaften abrufen:** Über *Column Dropdown* ? **Edit column > Add columns from reconciled values...** können zusätzliche Metadaten (z.B. Geburtsjahr, Standort, Typen, Bezeichnungen) aus der externen Datenquelle in neuen Spalten importiert werden. Dabei kann über **Add property** nach verfügbaren Werten gesucht werden.
Wurden Werte mit **Wikidata** reconciliert, können beim Import zusätzlicher Informationen sowohl **Property-IDs** (z.B. **P569** für Geburtsdatum) als auch **Property-Paths** verwendet werden, um gezielt bestimmte Bestandteile eines Werts abzurufen (z.B. nur das Jahr mit **SPARQL: P569@year**). Über die jeweiligen Properties können somit auch die IDs von anderen Reconciliation Services bezogen werden.
Verlinkte Werte, die über Eigenschaften wie z.B. **has part(s)** (**P527**) referenziert sind, werden bei erfolgreicher Reconciliation **automatisch** mit dem jeweiligen Service verbunden. Solche verknüpften Entitäten können nun wie andere angereicherte Werte behandelt werden, inklusive des Imports ihrer eigenen Eigenschaften (Labels, IDs, zusätzliche Metadaten).
- **URLs hinzufügen:** Über *Column Dropdown* ? **Reconcile > Add column with URLs of matched entities...** kann auch direkt eine Spalte mit den URLs zu den jeweiligen Werten hinzugefügt werden.
- **Badges und Status nutzen:** Reconciliation-Ergebnisse liefern Informationen darüber, ob ein Wert automatisch gematcht, manuell bestätigt oder noch unentschieden ist. Über *Column Dropdown* ? **Reconcile > Facets > By judgment** lassen sich diese Statuswerte filtern, um fehlende oder zweifelhafte Matches zu korrigieren.

Wichtiger Hinweis zur Performance & Verfügbarkeit von Eigenschaften

Die **Leistung (Antwortzeiten, Stabilität)** sowie die **Verfügbarkeit von Properties/Values** hängen ausschließlich vom **jeweiligen Reconciliation-Service** ab und **nicht** von OpenRefine.

Je nach Dienst (z.B. Wikidata, GND, VIAF, Getty) können Umfang der bereitgestellten Eigenschaften, unterstützte Property-Paths, Drosselungen (Rate Limits) und Wartungsfenster variieren.

Bei langsamen Antworten, fehlenden Properties oder temporären Ausfällen liegt die Ursache in der Regel beim **externen Service**, nicht in OpenRefine.

Weiters unterscheiden sich die **Score-Werte** je nach verwendetem Reconciliation-Service.

Während bei **Wikidata** ein Score von **100** einem perfekten Match entspricht, gelten bei **Getty** bereits Werte ab **40** als sehr gute Übereinstimmung.

Welche Score-Bereiche genau als „gut“ oder „akzeptabel“ gelten, ist **dienstspezifisch** und sollte in der jeweiligen Service-Dokumentation nachgelesen werden.

LLM Einbindung in OpenRefine

Die **LLM-Extension** erweitert OpenRefine um Funktionen, die große Sprachmodelle (LLMs) direkt in Datenbereinigungs- und Anreicherungs-Workflows integrieren. Typische Einsatzszenarien sind das Extrahieren von Informationen aus Freitext, das Normalisieren uneinheitlicher Schreibweisen oder das Generieren neuer strukturierter Felder.

Installation

Die Installation erfolgt wie in der [Einleitung](#) beschrieben.

Repository: <https://github.com/sunilnatraj/llm-extension>

1. **ZIP herunterladen** (vom GitHub-Release oder Repo).
 2. Inhalt des ZIP-Archivs in den OpenRefine-Ordner `webapp/extensions` entpacken.
 - In diesem Verzeichnis liegen bereits einige mitgelieferte Erweiterungen.
 - Nach dem Entpacken sollte ein Ordner `llm-extension` vorhanden sein.
 3. **OpenRefine starten** und ein Projekt öffnen.
 - Ist die Installation erfolgreich, erscheint in der Menüleiste ein „AI“-Button neben **Extensions**.
-

LLM-Provider hinzufügen

Eine detaillierte **Dokumentation** zum hinzufügen von verschiedenen LLM-Providern und eine Erklärungen der möglichen Parameter findet sich in der [Github Dokumentation](#). Wir wollen hier aber die wichtigsten Schritte zusammenfassen:

1. In OpenRefine den **AI-Button** öffnen ? **Manage LLM Providers**.
 - Hier sieht man die vorhandenen Modelle sowie Links zu *Setup Guide* und *Feature Guide*.
2. **Add LLM Provider** auswählen und die Felder ausfüllen:
 - **Label:** interne Bezeichnung für das Modell.
 - **Endpoint-URL:** `/chat/completions` ?kompatibel (siehe Provider-Guide, falls kein direkter Endpoint vorliegt).
 - **Model name:** z.B. `gpt-...`, `claude-...`, `llama-...` etc.
 - **API Key:** je nach Anbieter (Erstellung siehe jeweilige Provider-/Anbieter-Dokumentation).
 - **Parameter:**
 - **Temperature & Top-P** (steuern Kreativität/Determinismus).
 - **Seed** (optional, für reproduzierbare Ergebnisse).

- **Max Tokens** und **Rate?Limit?Wartezeit** (zur Steuerung von Kosten/Last).

3. **Test service:**

- Ein kurzer Test ist möglich. Auch wenn gelegentlich eine Fehlermeldung erscheint, kann die Modellnutzung im eigentlichen Datenworkflow dennoch funktionieren.

4. **Save** – der Provider steht nun in der Extension zur Verfügung.

Bei der Option **Test service** im **LLM Provider** Menü kommt es regelmäßig zu Fehlermeldungen. Diese sind auch möglich wenn der Service korrekt konfiguriert wurde. Um ein sachgemäßes Arbeiten des Models zu testen, sollte daher direkt eine Anwendung auf den Daten durchgeführt werden.

KI-gestützte Extraktion/Transformation anwenden

Neue Prompts anwenden

Zum Anwendungsdialog für die AI-Extension gelangen wir über *Column Dropdown* ? **Extract using AI**. Auch wenn der Name der Funktion darauf hindeutet, dass hier nur Daten mithilfe von KI-Anwendungen extrahiert werden können, ist die tatsächliche Funktionalität durch den jeweiligen Prompt gegeben.

Im Dialog kann konfiguriert werden:

- **Column:**
 - *Update existing* (in dieselbe Spalte schreiben) **oder**
 - *Create new* (eine neue Ergebnis?Spalte erzeugen).
- **LLM Provider:** zuvor eingerichteten Dienst auswählen.
- **Response format:** in vielen Fällen **Text** (weitere Formate je nach Extension?Stand).
- **Prompt:** klare Anweisung, Eingabeformat und gewünschtes Ausgabeformat definieren.

Beispiel?Prompt (Namen normalisieren):

```
I will show you names in different formats. Normalize each to "GivenName Surname".  
Return only "GivenName Surname". If the cell is empty, return an empty string.
```

Im Reiter **Preview** kann mit der Auswahl **Generate Preview** eine Vorschau des Ergebnisses des Prompts auf den ersten Wert in der Spalte erstellt werden. Anhand des Verhaltens hier kann der Prompt anschließend weiter bearbeitet werden.

Prompts werden **pro Zelle** als **separate Anfrage** ausgeführt.

Das bedeutet: Werte/Zellen können **nicht** innerhalb einer einzelnen AI-Abfrage **miteinander verknüpft** oder **über mehrere Zeilen/Spalten hinweg** gemeinsam verarbeitet werden.

Typische Anwendungsfälle:

- Uneinheitliche Namensformate harmonisieren (z.B. „Surname, GivenName“ ? „GivenName Surname“).
- Geburts-/Todesjahr (und ggf. -ort) aus biografischem Freitext extrahieren.
- Kurze Kategorisierung/Tagging aus Beschreibungen erzeugen.
- Adressen/Orte vormalisieren (mit klaren Felddefinitionen im Prompt).

Best Practice: Stellen Sie sicher, dass der Prompt eindeutig das gewünschte Ziel? Schema beschreibt (Feldreihenfolge, Trennzeichen, erlaubte/unerlaubte Werte), um konsistente Ergebnisse zu erhalten.

Prompts wiederverwenden

Im Anwendungsdialog unter *Column Dropdown* ? **Extract using AI** befinden sich auch die Reiter **History** und **Starred**.

- **History:** bietet eine Übersicht über all bereits verwendeten Prompts mit Beschreibungen zu den jeweiligen Konfigurationen. Hier können Prompts nun entweder als Favoriten mit Stern gekennzeichnet werden oder direkt wieder geladen werden.
- **Starred:** zeigt die Übersicht an Prompts welche im Reiter **History** mit einem Stern als Favoriten gekennzeichnet wurden. Gekennzeichnete Prompts können entfernt oder direkt geladen werden.

Tipps und weitere Informationen

Prompt-Tipps

- **Explizit sein:** gewünschtes Format, Beispiele, leere Fälle definieren.
- **Determinismus:** Temperature niedrig (z.B. 0.0–0.2) für reproduzierbare, regelnahe Outputs.

- **Validierung:** Antwort auf ein Feld oder ein JSON-Snippet begrenzen (wenn unterstützt).
- **Kleine Batches:** erst an Teilmengen testen, dann auf den gesamten Datensatz anwenden.

Fehlerbilder & Troubleshooting

- **„Test service“ meldet Fehler, aber Batch läuft:** Test ist nicht immer repräsentativ; maßgeblich ist das Verhalten beim tatsächlichen Spaltenlauf.
- **Uneinheitliche Outputs:** Prompt präzisieren (Ausgabeformat strikt festlegen), Temperature senken, Beispiele beifügen.
- **Leere Antworten:** Edge-Cases im Prompt definieren (z.B. „If input is empty, return empty string“).
- **Zeitüberschreitungen:** Max Tokens reduzieren, Batches verkleinern, Wartezeit erhöhen.

Qualitätssicherung & Wiederholbarkeit

- **Vorher/Nachher:** Bei zerstörungsfreier Arbeit neue Spalten erstellen (nicht „Update existing“), um Vergleiche zu ermöglichen.
- **Facets:** Nach einem AI-Durchlauf Text Facets nutzen, um Ausreißer schnell zu erkennen.
- **Undo/Redo:** OpenRefine speichert jeden Schritt; misslungene Läufe können rückgängig gemacht werden.
- **Rate Limits & Kosten:** Große Datensätze in Batches verarbeiten; Pausen konfigurieren (Provider-Einstellungen).

Tipp: Mit 20–50 Zeilen starten, Format & Qualität verifizieren, Prompt/Parameter anpassen und erst dann auf den gesamten Datensatz skalieren.

Datenschutz & Governance

Wichtig: Inhalt, Kosten und Datenschutz hängen vom **jeweiligen LLM-Provider** ab und nicht von OpenRefine. Prüfen Sie Datenklassifizierung, Nutzungsbedingungen, Logging und Speicherorte der Anbieter. Für sensible Daten ggf. **on-prem** bzw. **Self-Hosted-Modelle** oder lokal betriebene Gateways verwenden.

Externe Ressourcen für OpenRefine

Durch eine aktive Community rund um OpenRefine gibt es zahlreiche externe Lern- und Wissensressourcen. Für die Erstellung dieses Artikels verwendete Ressourcen wurden mit (*) gekennzeichnet. Alle verlinkten Ressourcen waren verfügbar mit Stand 23.03.2026.

In der [OpenRefine Recommendation List](#) findet sich auch noch eine Auswahl an weiteren Ressourcen.

Handbücher, Tutorials und Wissensressourcen

- **OpenRefine User Manual:** <https://openrefine.org/docs> *
- **FDMLab Workshops:** <https://fdmlab.landesarchiv-bw.de/workshops/> *
- **Getty Vocabularies OpenRefine Reconciliation:**
<https://www.getty.edu/research/tools/vocabularies/obtain/openrefine.html> *
- **British Library Introduction to OpenRefine by Owen Stephens:**
https://drive.google.com/drive/folders/11I29TjBgyWzNDGMnFjT01stCl_VJaY0C *
- **OpenRefine Workshop Hebrew University - Yael Gretzer:**
<https://docs.google.com/presentation/d/1dYjfMO3ACoepD59H8bL2qm8CFvhJfszAM8cZAr0Y7dc/edit?slide=id.p#slide=id.p>
- **OpenRefine - Data Science Training for Librarians - Tom Morris:**
<https://www.slideshare.net/slideshow/openrefine-data-science-training-for-librarians/18151288#1> *
- **Named Entity Recognition mit OpenRefine und spaCy:** <https://fdmlab.landesarchiv-bw.de/post/2021-08-ner-mit-openrefine-und-spacy/>
- **Gemeinsam an OpenRefine-Projekten arbeiten:** <https://fdmlab.landesarchiv-bw.de/post/2021-12-gemeinsam-an-openrefine-projekten-arbeiten/>
- **Library Carpentry: OpenRefine:** <https://librarycarpentry.github.io/lc-open-refine/> *
- **Data Wrangling and Cleaning with LLMs and OpenRefine:**
https://www.eosc.cz/media/3854440/letni_skola-christian_steiner-prezentace.pdf
- **Essential Spreadsheet Data Cleaning with OpenRefine:**
<https://libguides.galter.northwestern.edu/c.php?g=1075027&p=7829462>
- **Turning flat data into semantic data with Open Refine and the RDF extension:**
<https://networkedplanet.com/blog/2015/07/13/turning-flat-data-into-semantic-data-with-open-refine-and-the-rdf-extension.html>
- **Reconciliation in OpenRefine: Part 1 [Video]:** <https://www.youtube.com/watch?v=q8ffvdeyuNQ> *

- Reconciliation in OpenRefine: Part 2 [Video]: <https://www.youtube.com/watch?v=0tQPmfb6IFk> *
- Reconciling Metadata with OpenRefine [Video]: <https://www.youtube.com/watch?v=zWUdwN6s0ZU> *
- OpenRefine YouTube Channel: <https://www.youtube.com/@OpenRefineProject>

Extension und andere Erweiterungen

- OpenRefine Extension Overview: <https://openrefine.org/extensions> *
- Conciliator: <https://github.com/codeforkjeff/conciliator> *
- PyRefine: <https://github.com/jezcope/pyrefine>
- LLM Extension: <https://github.com/sunilnatraj/llm-extension> *
- RDF Extension: <https://github.com/AtesComp/rdf-transform>

Weiterführende Ressourcen

- Reconciliation Service Test Bench: <https://reconciliation-api.github.io/testbench/0.2/#/>
- Useful OpenRefine GREL functions - Yael Gretzer:
<https://docs.google.com/document/d/1sLkIZKeWxD6AbyLQz7d4o7JLyfxEMkTjgPd5OVS9jSk/edit?tab=t.0#heading=h.ielr0i6hygmb>
- OpenRefine GREL Recipes: <https://github.com/OpenRefine/OpenRefine/wiki/Recipes>
- Wikidata for authority control: sharing museum knowledge with the world - Alicia Fagervang:
<https://journals.uio.no/dhnbpub/article/view/10665>
- OpenRefine Wikibase Scoring mechanism: <https://openrefine-wikibase.readthedocs.io/en/latest/scoring.html>
- OpenRefine Forum with Helpdesk: <https://forum.openrefine.org/>
- OpenRefine Wiki FAQ: <https://github.com/OpenRefine/OpenRefine/wiki/FAQ>
- OpenRefine Recommendation List - External Resources:
<https://github.com/OpenRefine/OpenRefine/wiki/External-Resources> *