



Auswertung Archivdokumente zu Kunstobjekten

Anwendung von Large-Language-Models
(LLMs) zur Strukturierung und Integration
heterogener Daten in relationale
Datenbanksysteme

Henry Rotzoll (DVZ MV GmbH)

1. Einführung und Fragestellung, Hintergrund
2. Datengrundlage und Umsetzung
3. Ergebnisse
4. Erkenntnisse und Empfehlungen
5. Fazit und Ausblick

Die Digitalisierung stellt Unternehmen und Organisationen vor große Herausforderungen

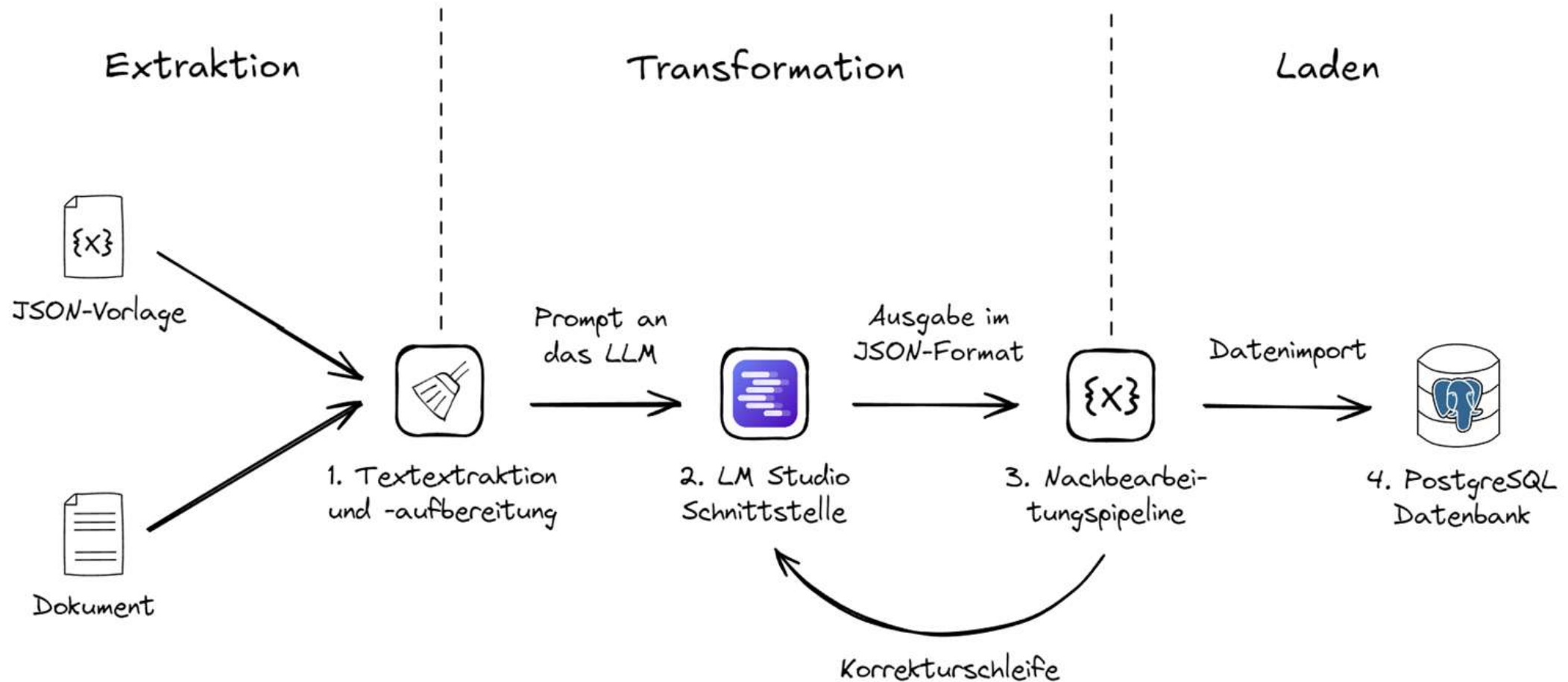
- sie verfügen über Archive mit großen Mengen an Dokumenten, die in eine gemeinsame Datenbank überführt werden sollen
- Dokumente wie Word- und PDF-Dateien enthalten unstrukturierte Daten, die vor der Datenintegration in ein strukturiertes Format überführt werden müssen
- Zusammenführung heterogener Daten ist ein zeitintensiver und fehleranfälliger Prozess, der mit hohem manuellen Aufwand verbunden ist

Kann der Prozess der Datenstrukturierung und -integration mithilfe von generativer KI automatisiert werden?

- Masterarbeit von Henrik Bongertmann (Universität Rostock)
- In Zusammenarbeit mit dem Sachgebiet GEO in der Zeit von April bis Oktober 2024
- Bestehendes Projekt mit der Nordkirche: Kunstguterfassung der Evangelisch-Lutherischen Kirche in Norddeutschland
 - Ziel: erfasste Kunstgüter in eine relationale Datenbank zu integrieren
 - Zweck: die in den Kirchengemeinden vorhandenen Kunstobjekte über eine interaktive Karte zugänglich zu machen
- 4 verschiedene Open Source Sprachmodelle verglichen
 - 30 Testdokumente verwendet
 - Gewinner: Llama-3.1-8B
- Laptop mit Intel Core i7, 32 GB Arbeitsspeicher und Nvidia GeForce GTX 1070 mit 8 GB VRAM für die Prozessierung verwendet

- 5

- Die Schritte im Implementierungsprozess



Prompt zur Strukturierung von Dokumenten in ein gültiges JSON-Format

In der folgenden Objektbeschreibung wurden Informationen über ein bestimmtes Kunstgut erfasst:

```
{document_text}
```

Aufgabe: Übertrage alle Informationen aus der Objektbeschreibung in ein gültiges JSON-Objekt anhand der folgenden Vorlage:

```
{json_template}
```

Hinweis: Die Informationen sollen ungekürzt und unverändert übertragen werden. Übertrage für jedes Merkmal den gesamten Text bis das nächste Merkmal beginnt. Es dürfen keine Informationen verloren gehen.

Verwende die vorgegebenen Schlüssel aus der Vorlage für das JSON-Objekt in der Antwort. Formatiere alle Werte im JSON-Objekt als Text-String. Setze die Schlüssel und Werte in doppelte Anführungszeichen, um sicherzustellen, dass die Antwort der korrekten JSON-Syntax entspricht. Die Antwort soll ausschließlich das JSON-Objekt und keinen zusätzlichen Text enthalten.

- Vorlage im JSON Format (29 Merkmale)

```

1 json_template = {
2   "Ortsname": "",
3   "Kirche": "",
4   "Ortskennzahl": "",
5   "Kirchengemeinde": "",
6   "Bezeichnung": "",
7   "Anzahl": "",
8   "Inventarnummer": "",
9   "Verweis auf zugehörige Objekte": "",
10  "Aufbewahrungsort/Standort": "",
11  "Material/Technik": "",
12  "Maße": "",
13  "Gewicht": "",
14  "Künstler/Werkstatt": "",
15  "Entstehungsort/Herkunft": "",
16  "Datierung": "",
17  "Beschreibung/Bemerkungen": "",
18  "Bildthema/Bildprogramm": "",
19  "Inschriften": "",
20  "Metallmarken und Auflösung": "",
21  "Zustand/Schaden": "",
22  "Fassung": "",
23  "Restaurierung": "",
24  "Leihvertrag": "",
25  "Literatur": "",
26  "Historische Bildquellen/Archivalische Quellen/Biografische Verweise": "",
27  "Stifter": "",
28  "Foto-Nummern": "",
29  "Erfasser": "",
30  "Erfassungsdatum/Letzte Änderung": ""
31 }

```



Ergebnisse

Temperatur-Parameter bestimmt die Modellkreativität bei der Antwortgenerierung

- Kreativität zwischen 0 und 2, wobei 0 möglichst deterministische Antworten erzeugt
- Testdurchlauf mit 30 Dokumenten ohne Nachbearbeitung der Ausgabe
- Niedrige Temperatur-Parameter (0,1 - 0,4) konnten höhere Erfolgsrate aufweisen

Parameter	Erfolgsrate	Antwortzeit	Antwortlänge	Token/sec
temp = 1.0	17 von 30 (57%)	72 s	1134 Token	16 t/s
temp = 0.7	18 von 30 (60%)	75 s	1187 Token	16 t/s
temp = 0.4	22 von 30 (73%)	75 s	1181 Token	16 t/s
temp = 0.1	21 von 30 (70%)	75 s	1175 Token	16 t/s

Tabelle 5: Vergleich der Erfolgsraten bei der Strukturierung von 30 Dokumenten in das Zielformat JSON mit unterschiedlichen Anpassungen der Modell-Kreativität

Vergleich der Erfolgsraten mit und ohne Nachbearbeitung der Modellantworten

- Bereinigung der generierten Ausgabe soll gültiges JSON-Format sicherstellen
- Korrekturschleife bei fehlerhaften Modell-Ausgaben, die nicht korrigiert werden können
- Effektive Nachbearbeitung verbessert Erfolgsrate signifikant

Performance/ Strukturierung	Erfolgsrate Gesamt (in %)	Korrektur- läufe	Ungültige Antworten	Laufzeit gesamt
ohne Nachbearbeitung	22 von 30 (73%)	-	8 von 30 (27%)	37 min
mit Korrekturschleife	23 von 30 (77%)	8 (27%)	15 von 38 (40%)	44 min
mit Nachbearbeitung	29 von 30 (97%)	-	1 von 30 (3%)	38 min
beides zusammen	30 von 30 (100%)	1 (3%)	1 von 31 (3%)	39 min

Tabelle 6: Vergleich der Erfolgsraten bei der Strukturierung von 30 Dokumenten in das Zielformat JSON mit Korrekturschleife und Nachbearbeitung der Ausgabe

5443 Dokumente konnten mit einer Erfolgsrate von 96% importiert werden

Quell- Ordner	Anzahl Dokumente	Erfolgreich importiert	Korrektur- läufe	Antwort- länge	Antwort- zeit	Laufzeit gesamt
A	842	829 (98%)	55 (7%)	698 Token	38 s	9h 46min
B	1729	1642 (95%)	79 (5%)	651 Token	36 s	ca. 18h
C	1199	1124 (94%)	69 (6%)	786 Token	48 s	ca. 17h
D	1900	1848 (97%)	85 (5%)	817 Token	49 s	ca. 27h
Gesamt	5670	5443 (96%)	288 (5%)	742 Token	43s	ca. 71h
Schnitt	100	96 (96%)	5 (5%)	742 Token	43s	1h 15min

Tabelle 8: Performance des LLM-basierten ETL-Prozesses für verschiedene Datenquellen

515 Fehlversuche auf Dokumente (24%) und LLM-Antworten (76%) zurückzuführen

Quell- Ordner	Durch- läufe	Fehl- versuche	Leere Datei	Token -limit	JSON ungültig	Spalte ungültig
A	897	68 (7%)	0 (0%)	2 (3%)	22 (32%)	44 (65%)
B	1808	166 (9%)	60 (36%)	2 (1%)	59 (36%)	45 (27%)
C	1268	144 (11%)	47 (33%)	3 (2%)	59 (41%)	35 (24%)
D	1985	137 (7%)	6 (5%)	3 (2%)	36 (26%)	92 (67%)
Gesamt	5958	515 (9%)	113 (22%)	10 (2%)	176 (34%)	216 (42%)

Tabelle 9: Fehlerursachen bei der LLM-basierten Strukturierung von Dokumenten



Erkenntnisse



Erkenntnisse

Die Antwortqualität von LLMs wird durch viele Faktoren beeinflusst

- Viele Stellschrauben, die optimiert werden können
 - Textaufbereitung, Datenmodell, Prompt-Engineering und Modell-Parameter
 - Auswirkungen sind nicht vorhersehbar, da die Funktionsweise eine Black-Box ist
 - Optimierungsprozess nach dem Trial-and-Error-Prinzip
- Nachbearbeitungspipeline konnte die Erfolgsrate signifikant erhöhen
 - LLMs sind keine Alleskönner, sondern nur ein Werkzeug
 - Eine effektive Nachbearbeitung der Modellausgabe ist daher unerlässlich
- Optimierungen müssen individuell auf den Anwendungsfall abgestimmt sein

Ergebnisse zeigen Potenzial von LLMs bei der automatisierten Datenstrukturierung

- Erfolgreiche Fallstudie: Über 5000 Dokumente wurden in die Datenbank importiert
- Open-Source-LLMs können unstrukturierte Texte in strukturierte Daten überführen
- LLM-basierter ETL-Prozess kann manuellen Aufwand erheblich reduzieren
- Optimierungsmaßnahmen konnten Ergebnisqualität signifikant steigern
- Ansatz ist auf andere Anwendungsfälle übertragbar (mit kleineren Anpassungen)
- Es können Empfehlungen für ein allgemeines Vorgehen abgeleitet werden



Wer?
Wo?
Was?
Wie?
Wann?
Warum?

Fragen?





Kontakt



Henry Rotzoll

+ 49 385 4800 532

h.rotzoll@dvz-mv.de

IHR ANSPRECHPARTNER



www.dvz-mv.de



www.dvz-mv.de/facebook



www.dvz-mv.de/twitter



www.dvz-mv.de/xing