

# Angewandte Generative KI in den (digitalen) Geisteswissenschaften

## LLM und API. Am Beispiel von GPT

<http://chpollin.github.io/GM-DH>

Christopher Pollin

<https://chpollin.github.io>

Zentrum für Informationsmodellierung, Graz

<https://informationsmodellierung.uni-graz.at/de/>

Digital Humanities Craft OG

[www.dhcraft.org](http://www.dhcraft.org)

## starter.py

C: > Users > pollin > Documents > GitHub > GM-DH > scripts > qpt-api-python >  starter.py > .

```
1 import os
2 import openai
3 openai.api_key = "OPENAI-API-KEYs"
4
5 completion = openai.ChatCompletion.create(
6     model="gpt-3.5-turbo",
7     messages=[
8         {"role": "system", "content": "You are a poetic assistant, skilled in writing metal songs about research data"},
9         {"role": "user", "content": "Compose a metal song that explains METS and MODS."}
10    ]
11 )
12
13 print(completion.choices[0].message)
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORT

## Setup

- Python installieren
  - OpenAI Bibliothek installieren
  - API-Key: <https://platform.openai.com/account/api-keys>



```
{  
  "role": "assistant",  
  "content": "(All music and chaos ensues)\n\nVerse 1:\nIn the depths of data's realm, where knowledge reigns supreme,\nThere lies a concept fierce and wild, a force you've never seen.\nMETS and MODS, their names resound, like thunder in the dark,\nUnleashing power from within, igniting data's spark.\n\nPre-Chorus:\nMETS the warrior, master of structure and form,\nMODS the sorcerer, the metadata's storm,\nTogether they unite, in a symphony of might,\nUnveiling secrets, unlocking worlds, with every line they write.\n\nChorus:\nMETS and MODS, guardians of information's lore,\nUnyielding, unrelenting, a digital uproar,\nThrough the numbers, through the code, they navigate the night,\nForging pathways to the truth, with data as their sight.\n\nVerse 2:\nMETS, the backbone shining bright, sturdy and robust,\nTaming digital chaos, with order it entrusts.\nWith XML's embrace, it structures data's mire,\nBuilding bridges through the bytes, connecting strands of fire.\n\nMODS, the enchantress, weaving magic in its wake,\nCrafts metadata's incantations, for all the world to take.\nDescribing resources, with precision and insight,\nBreathing life into the data, shining fierce like a knight.\n\nPre-Chorus:\nMETS the warrior, master of structure and form,\nMODS the sorcerer, the metadata's storm,\nTogether they unite, in a symphony of might,\nUnveiling secrets, unlocking worlds, with every line they write.\n\nChorus:\nMETS and MODS, guardians of information's lore,\nUnyielding, unrelenting, a digital uproar,\nThrough the numbers, through the code, they navigate the night,\nForging pathways to the truth, with data as their sight.\n\nOutro:  
  "role": "assistant",  
  "content": "Code: https://github.com/chpollin/GM-DH/blob/main/scripts/gpt-api-python/starter.py
```

# OpenAI Modelle

MODELS	DESCRIPTION
GPT-4	A set of models that improve on GPT-3.5 and can understand language or code
GPT-3.5	A set of models that improve on GPT-3 and can understand as language or code
GPT base	A set of models without instruction following that can understand natural language or code
DALL-E	A model that can generate and edit images given a natural language description
Whisper	A model that can convert audio into text
Embeddings	A set of models that can convert text into a numerical form
Moderation	A fine-tuned model that can detect whether text may be sensitive or inappropriate
GPT-3 Legacy	A set of models that can understand and generate natural language or code

<https://platform.openai.com/docs/models>

MODEL	DESCRIPTION	CONTEXT WINDOW	TRAINING DATA
gpt-4-1106-preview	<b>GPT-4 Turbo</b> <small>New</small> The latest GPT-4 model with improved instruction following, JSON mode, reproducible outputs, parallel function calling, and more. Returns a maximum of 4,096 output tokens. This preview model is not yet suited for production traffic. <a href="#">Learn more.</a>	128,000 tokens	Up to Apr 2023
gpt-4-vision-preview	<b>GPT-4 Turbo with vision</b> <small>New</small> Ability to understand images, in addition to all other GPT-4 Turbo capabilities. Returns a maximum of 4,096 output tokens. This is a preview model version and not suited yet for production traffic. <a href="#">Learn more.</a>	128,000 tokens	Up to Apr 2023
gpt-4	Currently points to gpt-4-0613. See <a href="#">continuous model upgrades</a> .	8,192 tokens	Up to Sep 2021
gpt-4-32k	Currently points to gpt-4-32k-0613. See <a href="#">continuous model upgrades</a> .	32,768 tokens	Up to Sep 2021
gpt-4-0613	Snapshot of gpt-4 from June 13th 2023 with improved function calling support.	8,192 tokens	Up to Sep 2021

<https://platform.openai.com/docs/models/gpt-4>

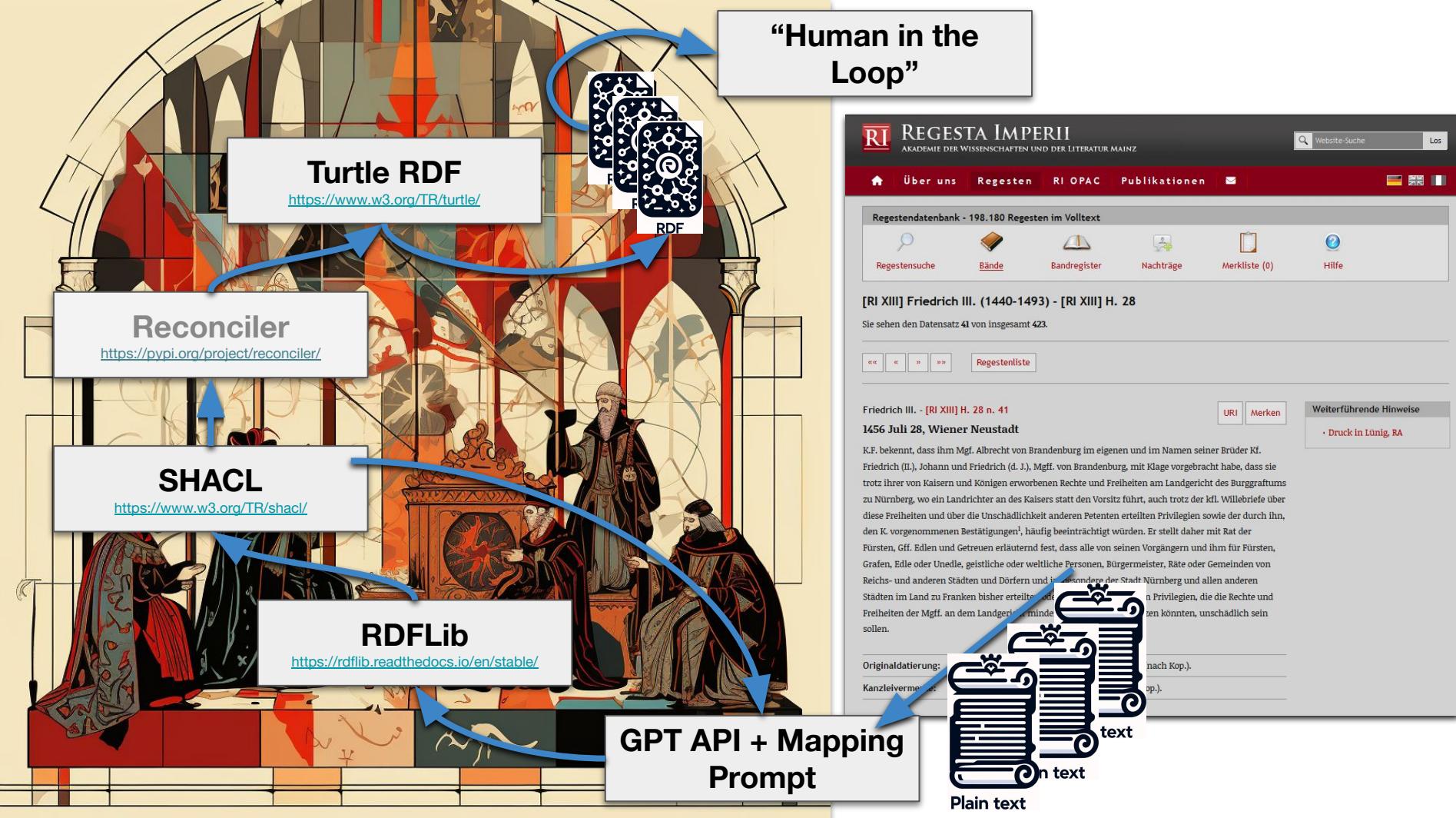
# Use Cases & Vorteile

## GPT API

- **Automatisierung**  
Integration in automatisierte Workflows.
- **Anpassbarkeit** z.B. Temperatur
- **Skalierbarkeit**
- **Zugriff auf verschiedene Versionen**  
(z.B. GPT-3.5, GPT-4, fine-tuned, ...)
- **Datenintegration:** Kann zusammen mit anderen APIs und Programmen verwendet werden (!!?)

## ChatGPT

- **Custom GPTs vs (Assistants API)**
- **Benutzerfreundlichkeit**  
keine Programmierkenntnisse
- **Lernen und Prototyping**
- **[Größeres Context Window**  
Mehr Tokens] *stimmt nicht mehr*
- **Spannende Plugins**  
ADA, DALL-E 3, ...



Midjourney: abstract representation of a medieval council chamber scene blended with an RDF graph structure. The illustration should include the silhouette of Emperor Friedrich III on a throne, shapes indicating the regal attire and crown, but no text or specific facial details. Gothic architectural elements like tall columns and pointed arches should be suggested through geometric forms. Margrave Albrecht of Brandenburg and his brothers are represented by a group of abstract figures in richly hued tunics. Nobles are indicated by smaller silhouettes with sunlight rays coming through window shapes cast elongated shadows on the figures, creating a dramatic atmosphere. The RDF graph is integrated into the scene using abstract lines and nodes connecting the figures, resembling a network of relationships without any text labels.

Am 28. Juli 1456 legte Markgraf Albrecht von Brandenburg zusammen mit seinen Brüdern in Wiener Neustadt Beschwerde bei Kaiser Friedrich III. ein. Es ging um die Missachtung ihrer Privilegien am Landgericht der Burggrafschaft Nürnberg, dem ein Landrichter vorstand. Trotz kaiserlicher und königlicher Bestätigungen seien ihre Rechte durch die Unschädlichkeit anderer Privilegien, auch der Stadt Nürnberg, beeinträchtigt. Kaiser Friedrich III. stellte klar, dass alle Privilegien, die die Rechte der Markgrafen etc., als unschädlich zu betrachten seien.



Friedrich III. - [RI XIII] H. 28 n. 41  
1456 Juli 28, Wiener Neustadt

K.F. bekennt, dass ihm **Mgf. Albrecht von Brandenburg** im eigenen und im Namen seiner Brüder **Kf. Friedrich (II.), Johann und Friedrich (d. J.), Mgff. von Brandenburg**, mit Klage vorgebracht habe, dass sie trotz ihrer von **Kaisern und Königen erworbenen Rechte und Freiheiten** am Landgericht des Burggraftums zu Nürnberg, wo ein Landrichter an des Kaisers statt den Vorsitz führt, auch trotz der kfl. Willebriefe über diese Freiheiten und über die Unschädlichkeit anderen Petenten erteilten Privilegien sowie der durch ihn, den K. vorgenommenen Bestätigungen<sup>1</sup>, häufig beeinträchtigt würden. Er stellt daher mit Rat der Fürsten, Gff. Edlen und Getreuen erläuternd fest, dass alle von seinen Vorgängern und ihm für Fürsten, Grafen, Edle oder Unerledige, geistliche oder weltliche Personen, Bürgermeister, Räte oder Gemeinden von Reichs- und anderen Städten und Dörfern und insbesondere der Stadt Nürnberg und allen anderen Städten im Land zu Franken bisher erteilten oder künftig zu erteilenden Privilegien, die die Rechte und Freiheiten der Mgff. an dem Landgericht mindern oder außer Kraft setzen könnten, unschädlich sein sollen.

regest1.txt

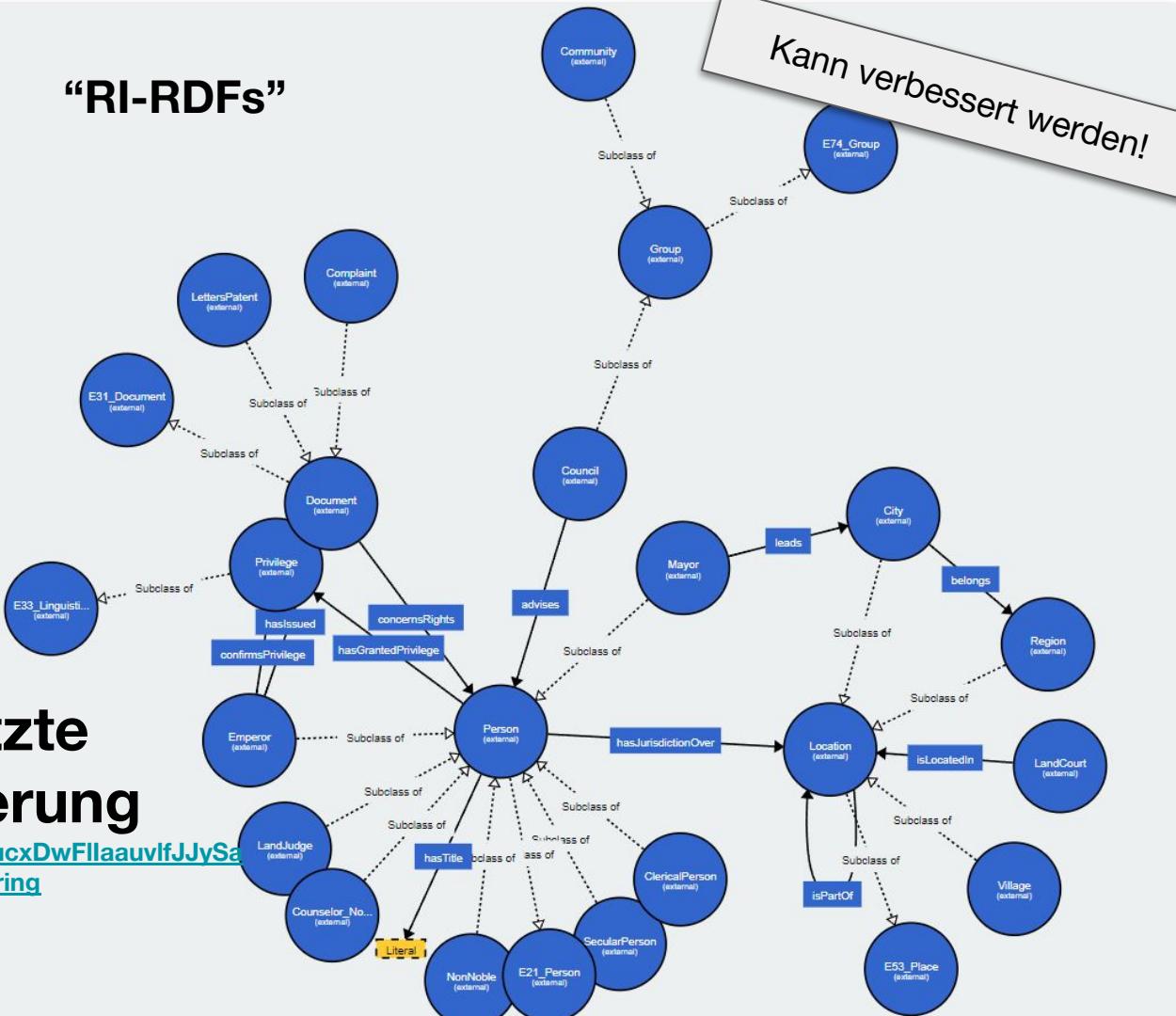
[http://www.regesta-imperii.de/regesten/13-28-0-friedrich-iii/nr/1456-07-28\\_1\\_0\\_13\\_28\\_0\\_41\\_41.html?tx\\_hisodat\\_sources\[action\]=show&tx\\_hisodat\\_sources\[controller\]=Sources&cHash=52f48e28f58cccd175856c15005963d56#rinav](http://www.regesta-imperii.de/regesten/13-28-0-friedrich-iii/nr/1456-07-28_1_0_13_28_0_41_41.html?tx_hisodat_sources[action]=show&tx_hisodat_sources[controller]=Sources&cHash=52f48e28f58cccd175856c15005963d56#rinav)

# “Everything is Prompted”

- Das RDFS (TBox) + Turtle RDF Mit Plaintext als Input  
[https://docs.google.com/document/d/1U\\_0xx41bkucxDwFliaauvlfJJySaZY5LAQAQba7bRlk/edit?usp=sharing](https://docs.google.com/document/d/1U_0xx41bkucxDwFliaauvlfJJySaZY5LAQAQba7bRlk/edit?usp=sharing)
- Der “Mapping Prompt” zur Transformation des Plaintext nach RDF  
<https://chat.openai.com/share/a52aff07-0f0c-42ca-8e71-5009524a1f2b>
- Das Python Script + GPT API + SHACL  
<https://chat.openai.com/share/270a7fdbd-ec28-4ea5-b89f-91fb46853fd4>

## **“RI-RDFs”**

# Kann verbessert werden!



# GPT-4-unterstützte Domänenmodellierung

[https://docs.google.com/document/d/1U\\_0xx41bkucxDwF1laauvlfJJySaZY5LAQAQba7bRlk/edit?usp=sharing](https://docs.google.com/document/d/1U_0xx41bkucxDwF1laauvlfJJySaZY5LAQAQba7bRlk/edit?usp=sharing)

# “RDFS Prompting from plain text”

## Translation and Extraction of Entities and Concepts

## Analysis of Entities, Concepts, and Terms

### Table of Entities, Concepts, and Important Terms and Relationships:

Here we create a structured table that could be used for historical research. For brevity, a simplified version is shown:

Term/Entity	Definition/Description	Relationship
Friedrich III	Emperor of the Holy Roman Empire	Issuer of privileges
Albrecht of Brandenburg	Margrave, complainant	Brother to other Margraves, recipient of rights
Land Court of Nuremberg	Judicial body in the Burgraviate	Venue for upholding Margrave rights
Electoral Letters Patent	Official documents securing freedoms	Grant protections to the Margraves
Privileges and Rights	Legal advantages granted by the emperor	Subject of the complaint
Cities and Villages	Communities in the Empire, especially in Franconia	Receivers of imperial privileges
City of Nuremberg	Prominent city in Franconia	Example of a privileged community
Franconia	Region in the Holy Roman Empire	Area concerned with the privileges

### You

You are an expert in ontology and medieval history. Your task is to analyze a given German "Regest" (abstract) of the *Regesta Imperii* (RI).

The *Regesta Imperii* chronologically records all the activities of the Roman-German kings and emperors from the Carolingians to Maximilian I (ca. 751-1519) as well as the popes of the early and high Middle Ages in the form of German "Regesten" (summaries).

You will do the following:

\* Read the "Regest" carefully and extract all entities, concepts, and important terms, with an emphasis on preparing the data for historical research. Translate it into English. Keep all the information.

\* Analyze all entities, concepts, and important terms and their relationship to each other.

\* Create a table of all entities, concepts, and important terms and relationships and add definitions to it.

\* Write critical feedback on your work and evaluate consistency and modeling choices.

\* Create an RDFS data model that represents all of your findings.

Regest:

Friedrich III. - [RI XIII] H. 28 n. 41

1456 Juli 28, Wiener Neustadt

K.F. bekennt, dass ihm Mgf. Albrecht von Brandenburg im eigenen und im Namen seiner Brüder Kf. Friedrich (II.), Johann und Friedrich (d. J.), Mgff. von Brandenburg, mit Klage vorgebracht habe, dass sie trotz ihrer von Kaisern und Königen erworbenen Rechte und Freiheiten am Landgericht des Burggraftums zu Nürnberg, wo ein Landrichter an des Kaisers statt den Vorsitz führt, auch trotz der kfl. Willebriefe über diese Freiheiten und über die Unschädlichkeit anderen Petenten erteilten Privilegien sowie der durch ihn, den K. vorgenommenen Bestätigungen, häufig beeinträchtigt würden. Er stellt daher mit Rat der Fürsten, Gff. Edlen und Getreuen erläuternd fest, dass alle von seinen Vorgängern und ihm für Fürsten, Grafen, Edle oder Unerede, geistliche oder weltliche Personen, Bürgermeister, Räte oder Gemeinden von Reichs- und anderen Städten und Dörfern und insbesondere der Stadt Nürnberg und allen anderen Städten im Land zu Franken bisher erteilten oder künftig zu erteilenden Privilegien, die die Rechte und Freiheiten der Mgff. an dem Landgericht mindern oder außer Kraft setzen könnten, unschädlich sein sollen.

Take a deep breath and lets think step by step. This is very important to my career.”

# “Schritt für Schritt Iterationen - Kritik - Kritik einarbeiten - Repeat”

## RDFS Data Model:

Creating an RDF Schema (RDFS) data model requires defining a class hierarchy and property relationships. For instance:

- `Person` class can have subclasses `Emperor`, `Margrave`, and `Noble`.
- `Document` class to represent different types of documents like `Letters Patent`.
- `Jurisdiction` class with instances like `Land Court of Nuremberg`.
- `Region` class with instances such as `Franconia`.
- Properties would connect individuals to their roles, documents to their issuers and recipients, and jurisdictions to the regions they govern.

Analyse the RDFS Data Model step by step.



You

You write critical reviews with the goal of improving the data model. You write the reviews of 2 different experts. One expert is an ontology engineer and the other is an expert of the Regesta Imperii. Find 3 improvements for the data model. Create a table containing the 3 improvements with a short text and a comma separated list of keywords indicating the improvements.

Then, look at all the improvements and create a list of improvements of the model.

Here is again the English input text:

## Expert 1: Ontology Engineer

Improvement 1:

**Text:** The model lacks explicit representation for hierarchical geographic entities and their governance structures. The relationship between the Burgravate at Nuremberg and its land could be modeled to reflect historical administrative hierarchies.

**Keywords:** geographic hierarchy, administrative structure, governance, Burgravate of Nuremberg, land court relationship

Improvement 2:

**Text:** The model needs properties to reflect temporal aspects of actions, such as complaints and the issuance of privileges, as these are critical in historical research for understanding the sequence of events.

**Keywords:** temporal properties, historical actions, time-sensitive relationships, complaint date, privilege issuance

# “Was nicht mappbar ist, wird mappbar gemacht”.



You

extract all entities, concepts, relations and terms related to the rdfs-datamodel from the text again. make a very detailed table. the table consists of the entities, the definition and a mapping to the rdfs-ontology. if you are not able to map something, make a list of the unmappable entities.

"Friedrich III. - [RI XIII] H. 28 n. 41

## 2. Restructuring the RDFS Ontology:

### A. Add New Classes:

#### 1. ri:Person

- Subclasses: `ri:Emperor`, `ri:Counselor\_Noble\_Elector`, `ri:LandJudge`, `ri:Mayor`, `ri:ClericalPerson`, `ri:SecularPerson`, `ri:NonNoble`
- This structure categorizes various individuals under a broader 'Person' class.

#### 2. ri:Group

- Subclasses: `ri:Council`, `ri:Community`
- Groups of people can be classified under this general class.

#### 3. ri:Location

- Subclasses: `ri:City`, `ri:Village`, `ri:Region`
- By distinguishing cities, villages, and regions, we can more precisely categorize geographical entities.

Entity/Concept/Term	Definition/Description	Mapping to RDFS-ontology
Friedrich III.	Name of the Emperor (Emperor Frederick).	ri:Emperor
Margrave Albrecht	Margrave of Brandenburg who presented a complaint on behalf of others.	ri:Counselor_Noble_Elector
Elector Frederick (II.)	Elector from Brandenburg.	ri:Counselor_Noble_Elector
Johann	Margrave of Brandenburg.	ri:Counselor_Noble_Elector
Frederick (the	Another Margrave of Brandenburg.	ri:Counselor_Noble_Elector
	al objection or grievance.	ri:Complaint
	specifically for land disputes.	ri:LandCourt
	ory or region.	ri:Region
	presiding over land disputes.	ri:Person
Electoral letters patent	Official letters regarding electoral matters.	ri:LettersPatent
Privileges	Specific rights or advantages.	ri:Rights
Princes	A noble title.	ri:Counselor_Noble_Elector
Counts	Another noble title.	ri:Counselor_Noble_Elector
Nobles	Persons with a noble title or rank.	ri:Counselor_Noble_Elector
Non-nobles	Persons without a noble title.	Unmappable
Clerical persons	Persons belonging to the church or clergy.	Unmappable
Secular persons	Persons not belonging to the church.	Unmappable
Mayors	Head of a city or town.	Unmappable

# CIDOC Mapping

RDFS Class/Property	CIDOC CRM Class/Property	Notes
ri:Document	E31 Document	Suitable mapping as both refer to recorded information.
ri:Complaint	E33 Linguistic Object	A Complaint is a type of linguistic matter, but we might need a more specific subclass in CIDOC CRM to represent complaints.
ri:LettersPatent	E33 Linguistic Object	Suitable. This can represent a formal document, and the description of Letters Patent fits.
ri:Privilege	E33 Linguistic Object	Similar to above, but we might consider associating it with a rights-based entity in CRM if available.
ri:Location	E53 Place	General class for spatial concepts.
ri:City	E53 Place	A city is a type of place. However, there might be specific subclasses for populated areas in CIDOC CRM.
ri:Village	E53 Place	Similar to city, we should look for a subclass representing smaller populated areas.
ri:Region	E53 Place	Regions could map to a higher-level spatial concept in CIDOC CRM if available.
ri:Person	E21 Person	Clear match, as it represents individuals.
ri:Emperor	E21 Person	Emperors are individuals but considering their status, we could look into a more specific subclass for royal or noble persons if available.

Kann verbessert werden!



# Der Prompt zur Transformation des Plaintext nach RDF

## \*\*Mapping rules\*\*:

1. **Classes and Subclassing\*\*:**
  - `ri:Document` is a subclass of `crm:E31\_Document`.
  - `ri:Complaint`, `ri:LettersPatent`, and `ri:Privilege` are subclasses of `ri:Document`, with `ri:Privilege` also being `crm:E33\_Linguistic\_Object`.
  - `ri:Location` is a subclass of `crm:E53\_Place`, with specific locations like `ri:City`, `ri:Village`, and `ri:Region` inheriting from `ri:Location`.
  - `ri:Person` is a subclass of `crm:E21\_Person`, with specializations like `ri:Emperor`, `ri:Counselor`, etc., inheriting properties and restrictions from `ri:Person`.
2. **Property Specificity and Hierarchy\*\*:**
  - Various properties such as `ri:hasIssued`, `ri:concernsRights`, `ri:hasJurisdictionOver`, `ri:hasGrantedPrivilege`, `ri:advises`, `ri:leads`, `ri:confirmsPrivilege`, `ri:belongs`, `ri:isLocatedIn`, `ri:isPartOf`, and `ri:hasTitle` are defined with specific domain and range relations, aligning with the CRM patterns.
  - Introduce new properties like `ri:detailsOfComplaint` and `ri:decisionDetails` for capturing specific content of the document.
3. **Entity Mappings and ABox Interpretations\*\*:**
  - Individual entities (like `ex:FriedrichIII`, `ex:MargraveAlbrechtBrandenburg`) are mapped to their respective classes, with their titles and document relationships.
  - Locations (like `ex:Nuremberg`, `ex:Franconia`) are defined hierarchically and related through `ri:belongs` and `ri:isPartOf`.
  - Add annotations for specific complaints and decisions where appropriate.
4. **Document and Rights Contextualization\*\*:**
  - Documents and rights are contextualized with comments indicating their historical and legal relevance, including specific complaints and decisions.
5. **Refinement and Disambiguation\*\*:**
  - The properties are aligned with CRM patterns for semantic consistency.
6. **Guidelines\*\*:**
  - Follow mapping rules strictly.
  - Preserve the original text.
  - Produce well-formed Turtle RDF.
  - Return only Turtle RDF. No ````
  - Annotate only when appropriate.
  - Preserve the complexity of output.
  - Compact Turtle RDF without any whitespace or indentation.
  - Use the following namespaces:
    - @prefix ri: <http://www.example.org/ontology/ri#> .
    - @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
    - @prefix crm: <http://www.cidoc-crm.org/cidoc-crm/> .
    - @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
  - Add rdfs:label to entities for better readability.
  - No language tags.

Kann verbessert werden!



You

You are an expert at creating mapping rules. Here is an RDFS with TBox and ABox. Create a very information-rich, but short, list of mappings with all the information given. Be more generalize and dont focus on specific entities. The Abox is just here to give you further context. return a markdown list.

Take a deep breath and lets think step by step. This is very important to my career.

## Plaintext nach Turtle RDF Mapping Rules

<https://chat.openai.com/share/a52aff07-0f0c-42ca-8e71-5009524a1f2b>

You will act as a skilled expert automaton that is proficient in transforming unstructured text, specifically given German "Regest" (abstract) of the Regesta Imperii (RI), into Turtle RDF. Analyze the provided text based on the mapping rules I have shared and then execute the transformation to produce Turtle RDF based on the RI-Ontology, ensuring you adhere to the guidelines and only annotate if certain.

### \*\*Mapping rules\*\*:

#### 1. \*\*Classes and Subclassing\*\*:

- `ri:Document` is a subclass of `crm:E31\_Document`.
- `ri:Complaint`, `ri:LettersPatent`, and `ri:Privilege` are subclasses of `ri:Document`, with `ri:Privilege` also being a subclass of `crm:E33\_Linguistic\_Object`.
- `ri:Location` is a subclass of `crm:E53\_Place`, with specific locations like `ri:City`, `ri:Village`, and `ri:Region` inheriting from it.
- `ri:Person` is a subclass of `crm:E21\_Person`, with specializations like `ri:Emperor`, `ri:Counselor`, etc., inheriting properties and restrictions.

#### 2. \*\*Property Specificity and Hierarchy\*\*:

- Various properties such as `ri:hasIssued`, `ri:concernsRights`, `ri:hasJurisdictionOver`, `ri:hasGrantedPrivilege`, `ri:advises`, `ri:leads`, `ri:confirmsPrivilege`, `ri:belongs`, `ri:isLocatedIn`, `ri:isPartOf`, and `ri:hasTitle` are defined with specific domain and range relations, aligning with the CRM patterns.
- Introduce new properties like `ri:detailsOfComplaint` and `ri:decisionDetails` for capturing specific content of the document.

#### 3. \*\*Entity Mappings and ABox Interpretations\*\*:

- Individual entities (like `ex:FriedrichIII`, `ex:MargraveAlbrechtBrandenburg`) are mapped to their respective classes, with their titles and document relationships.
- Locations (like `ex:Nuremberg`, `ex:Franconia`) are defined hierarchically and related through `ri:belongs` and `ri:isPartOf`.
- Add annotations for specific complaints and decisions where appropriate.

#### 4. \*\*Document and Rights Contextualization\*\*:

- Documents and rights are contextualized with comments indicating their historical and legal relevance, including specific complaints and decisions.

#### 5. \*\*Refinement and Disambiguation\*\*:

- The properties are aligned with CRM patterns for semantic consistency, and ambiguities are minimized through explicit domain and range associations.

#### 6. \*\*Guidelines\*\*:

- Follow mapping rules strictly.
- Preserve the original text.
- Produce well-formed Turtle RDF.
- Return only Turtle RDF. No ``
- Annotate only when appropriate.
- Preserve the complexity of output.
- Compact Turtle RDF without any whitespace or indentation.
- Use the following namespaces:
  - @prefix ri: <http://www.example.org/ontology/ri#> .
  - @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
  - @prefix crm: <http://www.cidoc-crm.org/cidoc-crm/> .
  - @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
- Add rdfs:label to entities for better readability.
- No language tags.

## Der “Mapping Prompt” zur Transformation des Plaintext nach RDF

### Turtle RDF Example:

```

regest1 X regest2 X regest3

Datei Bearbeiten Ansicht

Friedrich III. - [RI XIII] H. 28 n. 41  
1456 Juli 28, Wiener Neustadt

K.F. bekennt, dass ihm Mgf. Albrecht von Brandenburg im eigenen und im Namen seiner Brüder Kf. Friedrich (II.), Johann und Friedrich (d. J.), Mgff. von Brandenburg, mit Klage vorgebracht habe, dass sie trotz ihrer von Kaisern und Königen erworbenen Rechte und Freiheiten am Landgericht des Burggraftums zu Nürnberg, wo ein Landrichter an des Kaisers statt den Vorsitz führt, auch trotz der kfl. Willebriefe über diese Freiheiten und über die Unschädlichkeit anderer Petenten erteilten Privilegien sowie der durch ihn den K. vorgenommenen Bestätigungen<sup>1</sup>, häufig beeinträchtigt würden. Er stellt daher mit Rat der Fürsten, Gff. Edlen und Getreuen erläuternd fest, dass alle von seinen Vorgängern und ihm für Fürsten, Grafen, Edle oder Unerledige, geistliche oder weltliche Personen, Bürgermeister, Räte oder Gemeinden von Reichs- und anderen Städten und Dörfern und insbesondere der Stadt Nürnberg und allen anderen Städten im Land zu Franken bisher erteilten oder künftig zu erteilenden Privilegien, die die Rechte und Freiheiten der Mgff an dem Landgericht mindern oder außer Kraft setzen könnten, unschädlich sein sollen.

Originaldatierung:

An mittichen nach sancti Marien Magdalenen tag (nach Kop.).

Kanzleivermerke:

KVr: A.m.d.i.c. Ulric(us) Weltzli vicecanc. (nach Kop.).

Registereinträge:

Brandenburg (Brandenburg)  
Markgrafen von ~ (= Albrecht (Achilles), Kurfürst (1470-1486), Herzog von Pommern, Burggraf von Nürnberg, Hauptmann und Hofmeister (1463) Friedrichs III., Reichserbkämmerer, Richter am Kammergericht, Forst- und Jägermeister im Weissenburger Forst (1414-1486))  
Friedrich II. (der Eiserne), Burggraf von Nürnberg, Kurfürst (1440-1470), Bruder Albrecht Achilles  
Friedrich der Jüngere (1424-1463), Bruder Albrecht Achilles  
Georg (der Fromme) (1515-1543)  
Johann IV. (der Alchimist), Gesandter Friedrichs III. (1406-1464), Bruder Albrecht Achilles  
Franken (Bayern), Land  
Langheim (Klosterlangheim, Stadtteil von Lichtenfels, sō. Coburg, Bayern), Zisterzienserkloster  
Abt Johann (V.) Fabri genannt Strauß (1510-1538)  
Nürnberg (Bayern), Stadt  
Burg  
Landgericht (des Burggraftums)  
Weltzli  
Ulrich, Bürger von Göppingen, Kanzleischreiber, Protonotar, Notar, römischer Vizekanzler und Kanzler der römischen Kanzlei Friedrichs III., Rat Friedrichs III., Beisitzer des Kammergerichts (+ 1462); Vizekanzler; Kanzler; Rat Friedrichs III.  
Wiener Neustadt (Niederösterreich), Stadt  
AO

## regest1.txt

[http://www.regesta-imperii.de/regesten/13-28-0-friedrich-iii/nr/1456-07-28\\_1\\_0\\_13\\_28\\_0\\_41\\_41.htm?tx\\_hisodat\\_sources\[action\]=show&tx\\_hisodat\\_sources\[controller\]=Sources&cHash=52f48e28f58cccd175856c15005963d56#rinav](http://www.regesta-imperii.de/regesten/13-28-0-friedrich-iii/nr/1456-07-28_1_0_13_28_0_41_41.htm?tx_hisodat_sources[action]=show&tx_hisodat_sources[controller]=Sources&cHash=52f48e28f58cccd175856c15005963d56#rinav)

# Input

regest1 X regest2 X regest3

Datei Bearbeiten Ansicht

Friedrich III. - [RI XIII] H. 28 n. 42  
1456 Juli 28, Wiener Neustadt

URI Merken

K.F. bekennt, dass ihm Mgf. Albrecht von Brandenburg im eigenen und im Namen seiner Brüder Kf. Friedrich (II.), Johann und Friedrich (d. J.), Mgff. von Brandenburg, mit Klage vorgebracht habe, dass sie trotz ihrer von Kaisern und Königen erworbenen Rechte und Freiheiten am Landgericht des Burggraftums zu Nürnberg, wo ein Landrichter diese Freiheiten und über den K. vorgenommenen Best Gff. Edlen und Getreuen e Edle oder Unerledige, geistliche anderen Städten und Dörfern Franken bisher erteilten an dem Landgericht minder und Rat der Stadt Köln) Capellen.2

Originaldatierung:  
An mittwoch nach sand

Kanzleivermerke:  
KVr: A.m.d.i.c. Ulrich Weltzli vicecanc. - Rta Stephanus Kolbeck (Blattmitte); Ehenheim (rechter Blattrand). - Leh(en)pr(ie) ub(er) d(en) wiltpant(n) i(m) Weis(e)n(p)urgen) wald i(m) 56 jar und wyldtpann in Weyßenburger waldt (Empfängervermerke auf der Rückseite).

Registereinträge

Ansbach (Bayern), Sta Chorherrenstift S Capellen (zu Köln, No Gulpen (sō. Mastrich Ansbach Köln (Nordrhein-Westf Kolbeck, Stephan, Reg Weltzli Ulrich, Bürger vo Kanzler der römischen Kan 1462) ; Vizekanzler ; Kan Wiener Neustadt (Nied AO

Registerreinträge

Ehenheim (Enheim, Georg (zu Geyern), Ritter Kolbeck, Stephan, Registratur in der römischen Kanzlei Friedrichs III. Weissenburg (Bayern), Stadt Wald bzw. Forst Weltli Ulrich, Bürger von Göppingen, Kanzleischreiber, Protonotar, Notar, römischer Vizekanzler und Kanzler der römischen Kanzlei Friedrichs III., Rat Friedrichs III., Beisitzer des Kammergerichts (+ 1462); Vizekanzler ; Kanzler ; Rat Friedrichs III. Wiener Neustadt (Niederösterreich), Stadt

## regest2.txt

[http://www.regesta-imperii.de/regesten/13-28-0-friedrich-iii/nr/1456-07-28\\_2\\_0-13\\_28\\_0-42\\_42.html?tx\\_hisodat\\_sources%5BitemsPerPage%5D=10&tx\\_hisodat\\_sources%5BorderBy%5D=50&tx\\_hisodat\\_sources%5BascDesc%5D=10&tx\\_hisodat\\_sources%5Baction%5D=show&tx\\_hisodat\\_sources%5Bcontroller%5D=Sources&cHash=390b6ae05e77d2c8c368718a4650ec#rinav](http://www.regesta-imperii.de/regesten/13-28-0-friedrich-iii/nr/1456-07-28_2_0-13_28_0-42_42.html?tx_hisodat_sources%5BitemsPerPage%5D=10&tx_hisodat_sources%5BorderBy%5D=50&tx_hisodat_sources%5BascDesc%5D=10&tx_hisodat_sources%5Baction%5D=show&tx_hisodat_sources%5Bcontroller%5D=Sources&cHash=390b6ae05e77d2c8c368718a4650ec#rinav)

## regest3.txt

[http://www.regesta-imperii.de/regesten/13-28-0-friedrich-iii/nr/1456-08-14\\_1\\_0\\_13\\_28\\_0\\_45\\_45.html?tx\\_hisodat\\_sources%5BitemsPerPage%5D=10&tx\\_hisodat\\_sources%5BorderBy%5D=50&tx\\_hisodat\\_sources%5BascDesc%5D=10&tx\\_hisodat\\_sources%5Baction%5D=show&tx\\_hisodat\\_sources%5Bcontroller%5D=Sources&cHash=a77a0de2f0de8504fd0c145494f0ec#rinav](http://www.regesta-imperii.de/regesten/13-28-0-friedrich-iii/nr/1456-08-14_1_0_13_28_0_45_45.html?tx_hisodat_sources%5BitemsPerPage%5D=10&tx_hisodat_sources%5BorderBy%5D=50&tx_hisodat_sources%5BascDesc%5D=10&tx_hisodat_sources%5Baction%5D=show&tx_hisodat_sources%5Bcontroller%5D=Sources&cHash=a77a0de2f0de8504fd0c145494f0ec#rinav)

# Das Python Script + GPT API

```
213 def process_files(shacl_shapes):
214     input_dir = 'registen/input'
215     output_dir = 'registen/output'
216
217     # Loop over all files in the input directory
218     for filename in os.listdir(input_dir):
219         if filename.endswith('.txt'):
220             # Construct the full file path
221             file_path = os.path.join(input_dir, filename)
222
223             # Load the regest
224             regest = load_regest(file_path)
225
226             # Create the RDF
227             rdf = create_rdf_from_regest(regest)
228
229             # Validate the RDF with SHACL shapes
230             print(rdf)
231             is_valid, report = validate_rdf_with_shacl(rdf, shacl_shapes)
232
233             if is_valid:
234                 # Construct the output file path
235                 rdf_file_path = os.path.join(output_dir, filename.replace('.txt', '.rdf'))
236
237                 # Save the RDF
238                 save_rdf(rdf, rdf_file_path)
239             else:
240                 print(f"RDF validation failed for {filename}. Report: {report}")
241
242     # Call the function to process all files
243     process_files(shacl_shapes)
244     print("Registen successfully processed.")
```

<https://github.com/chpollin/GM-DH/tree/main/scripts/gpt-api-python>

Kann verbessert werden!  
Code zu ~95% generiert

```
190 def create_rdf_from_regest(regest_text):
191     messages = [
192         {
193             "role": "system",
194             "content": system_prompt
195         },
196         {
197             "role": "user",
198             "content": regest_text
199         }
200     ]
201
202     response = openai.ChatCompletion.create(
203         model="gpt-4",
204         messages=messages,
205         temperature = 0
206     )
207
208     if response.choices and response.choices[0].message:
209         return response.choices[0].message['content'].strip()
210     else:
211         return "No RDF conversion found."
```

Hier geschieht das Wichtigste:

- System\_prompt
- Regest\_text
- Model
- temperature

# Validierung mit RDFLib und SHACL

Iteration einbauen, wenn Validierung mit RDFLib und SHACL fehlschlägt.

→ Aber wir brauchen eine temperature größer als 0, um immer leicht unterschiedliche RDFs zu erhalten.

```
169 def validate_rdf_with_shacl(rdf_content, shacl_shapes):
170     try:
171         data_graph = rdflib.Graph()
172         data_graph.parse(data=rdf_content, format="turtle")
173
174         shapes_graph = rdflib.Graph()
175         shapes_graph.parse(data=shacl_shapes, format="turtle")
176
177         conforms, results_graph, results_text = pyshacl.validate(data_graph, shacl_graph=shapes_graph,
178   data_graph_format='turtle', shacl_graph_format='turtle')
179
180         # Return both the conformity status and the results text
181         return conforms, results_text
182
183     except Exception as e:
184         error_message = f"RDF was not created because: {e}"
185         print(error_message)
186         # Return False and the error message
187         return False, error_message
```

Kann verbessert werden!

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix ri: <http://www.example.org/ontology/ri#> .
@prefix crm: <http://www.cidoc-crm.org/cidoc-crm/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

# SHACL shapes for ri:Document and its subclasses
ri:DocumentShape
  a sh:NodeShape ;
  sh:targetClass ri:Document ;
  sh:property [
    sh:path ri:hasIssued ;
    sh:or ( [ sh:class ri:Document ] [ sh:class ri:Complaint ] [ sh:class ri:Privilege ] ) ; # Allow linking to Document or its subclasses
    sh:minCount 1 ;
  ] ;
  sh:property [
    sh:path crm:P4_has_time-span ;
    sh:datatype xsd:date ;
    sh:minCount 1 ;          # Making it mandatory
  ] ;
  sh:property [
    sh:path ri:isLocatedIn ;
    sh:or ( [ sh:class ri:City ] [ sh:class ri:Location ] [ sh:class ri:Region ] [ sh:class ri:Forest ] ) ; # Including ri:Forest
    sh:minCount 1 ;          # Making it mandatory if location is always specified
  ] .

# Additional properties for ri:Complaint and ri:Privilege
ri:ComplaintShape
  a sh:NodeShape ;
  sh:targetClass ri:Complaint ;
  sh:property [
    sh:path ri:validOfComplaint ;
    sh:datatype rdfs:literal ;
    sh:minCount 1 ;
  ] ;
  sh:property [
    sh:path ri:decisionDetails ;
    sh:datatype rdfs:literal ;
    sh:minCount 1 ;
  ] .

ri:PrivilegeShape
  a sh:NodeShape ;
  sh:targetClass ri:Privilege ;
  sh:property [
    sh:path ri:detailsOfPrivilege ;
    sh:datatype rdfs:literal ;
    sh:minCount 1 ;
  ] .

# SHACL shape for ri:Person
ri:PersonShape
  a sh:NodeShape ;
  sh:targetClass ri:Person ;
  sh:property [
    sh:path ri:hasTitle ;
    sh:datatype rdfs:literal ;
    sh:minCount 1 ; # Making it mandatory if every person has a title
  ] .
```

## Validierung mit SHACL

<https://chat.openai.com/share/270a7fbd-ec28-4ea5-b89f-91fb46853fd4>

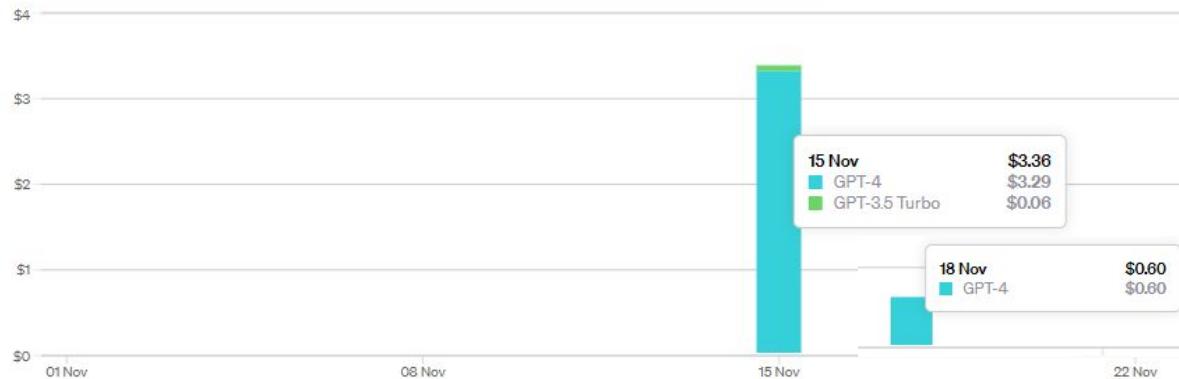
<https://chat.openai.com/share/ba2a947e-ff88-481a-8102-35c0129a5205>

# OpenAI GPT API Kosten

## Usage

Cost Activity

Daily Costs \$3,36



| Model                     | Input                | Output               |
|---------------------------|----------------------|----------------------|
| gpt-4-1106-preview        | \$0.01 / 1K tokens   | \$0.03 / 1K tokens   |
| gpt-4-1106-vision-preview | \$0.01 / 1K tokens   | \$0.03 / 1K tokens   |
| Model                     | Input                | Output               |
| gpt-4                     | \$0.03 / 1K tokens   | \$0.06 / 1K tokens   |
| gpt-4-32k                 | \$0.06 / 1K tokens   | \$0.12 / 1K tokens   |
| Model                     | Input                | Output               |
| gpt-3.5-turbo-1106        | \$0.0010 / 1K tokens | \$0.0020 / 1K tokens |
| gpt-3.5-turbo-instruct    | \$0.0015 / 1K tokens | \$0.0020 / 1K tokens |

3 Regesten einmal zu transformieren kostet 0.30 \$.

18.11.2023  
<https://openai.com/pricing>

# Output

```
C: > Users > chris > Downloads > regest1.ttl
1 @prefix ri: <http://www.example.org/ontology/ri#> .
2 @prefix ri-data: <http://www.example.org/data/ri#> .
3 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
4 @prefix crm: <http://www.cidoc-crm.org/cidoc-crm/> .
5 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
6 √ <ri-data:FriedrichIII-RI_XIII_H_28_n_41> a ri:Document ;
7   rdfs:label "Friedrich III. - [RI XIII] H. 28 n. 41" ;
8   ri:hasIssued <ri-data:Complaint_FriedrichIII-RI_XIII_H_28_n_41> ;
9   crm:P4_has_time-span "1456-07-28"^^xsd:date ;
10  ri:isLocatedIn <ri-data:Wiener_Neustadt> .
11 √ <ri-data:Complaint_FriedrichIII-RI_XIII_H_28_n_41> a ri:Complaint ;
12  ri:concernsRights <ri-data:MargraveAlbrechtBrandenburg> ,
13    <ri-data:MargraveFriedrichII> ,
14    <ri-data:MargraveJohann> ,
15    <ri-data:MargraveFriedrichJr> ;
16  ri:detailsOfComplaint "Complaint about infringement of rights at Nuremberg land court" ;
17  ri:decisionDetails "Decision by Friedrich III clarifying the impact of privileges on rights at Nuremberg land court" ;
18  ri:isLocatedIn <ri-data:Nuremberg> ;
19  crm:P70_documents <ri-data:FriedrichIII> .
20 √ <ri-data:MargraveAlbrechtBrandenburg> a ri:Person ;
21  rdfs:label "Albrecht von Brandenburg" ;
22  ri:hasTitle "Mgf." .
23 √ <ri-data:MargraveFriedrichII> a ri:Person ;
24  rdfs:label "Friedrich II." ;
25  ri:hasTitle "Mgf." .
26 √ <ri-data:MargraveJohann> a ri:Person ;
27  rdfs:label "Johann" ;
28  ri:hasTitle "Mgf." .
29 √ <ri-data:MargraveFriedrichJr> a ri:Person ;
30  rdfs:label "Friedrich (d. J.)" ;
31  ri:hasTitle "Mgf." .
32 √ <ri-data:Nuremberg> a ri:City ;
33  rdfs:label "Nürnberg" ;
34  ri:isPartOf <ri-data:Franconia> .
35 √ <ri-data:Franconia> a ri:Region ;
36  rdfs:label "Franken" .
37 √ <ri-data:Wiener_Neustadt> a ri:City ;
38  rdfs:label "Wiener Neustadt" .
```

## regest1.ttl

<https://github.com/chpollin/GM-DH/blob/main/scripts/gpt-api-python/regesten/output/regest1.rdf>

## regest3.ttl

<https://github.com/chpollin/GM-DH/blob/main/scripts/gpt-api-python/regesten/output/regest3.rdf>

C: > Users > chris > Downloads > regest3.ttl

```
1 @prefix ri: <http://www.example.org/ontology/ri#> .
2 @prefix ri-data: <http://www.example.org/data/ri#> .
3 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
4 @prefix crm: <http://www.cidoc-crm.org/cidoc-crm/> .
5 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
6
7 √ <ri-data:FriedrichIII-RI_XIII_H_28_n_45> a ri:Document ;
8   rdfs:label "Friedrich III. - [RI XIII] H. 28 n. 45" ;
9   ri:hasIssued <ri-data:Privilege_FriedrichIII-RI_XIII_H_28_n_45> ;
10  crm:P4_has_time-span "1456-08-14"^^xsd:date ;
11  ri:isLocatedIn <ri-data:Wiener_Neustadt> .
12 √ <ri-data:Privilege_FriedrichIII-RI_XIII_H_28_n_45> a ri:Privilege ;
13  ri:hasGrantedPrivilege <ri-data:GeorgEhenheim> ;
14  ri:detailsOfPrivilege "Permission to Georg von Ehenheim and his heirs to hunt unrestrictedly in the Weißenburger Reichsforst, with the obligation to deliver a deer to the imperial or royal court whenever the Emperor and his successors are in this region of the empire" ;
15  ri:isLocatedIn <ri-data:Weissenburger_Reichsforst> ;
16  crm:P70_documents <ri-data:FriedrichIII> .
17 √ <ri-data:GeorgEhenheim> a ri:Person ;
18  rdfs:label "Georg von Ehenheim" ;
19  ri:hasTitle "Ritter" .
20 √ <ri-data:Weissenburger_Reichsforst> a ri:Forest ;
21  rdfs:label "Weißenburger Reichsforst" ;
22  ri:isPartOf <ri-data:Weissenburg> .
23 √ <ri-data:Weissenburg> a ri:City ;
24  rdfs:label "Weißenburg" .
25 √ <ri-data:Wiener_Neustadt> a ri:City ;
26  rdfs:label "Wiener Neustadt" .
27 √ <ri-data:UlricusWeltzli> a ri:Person ;
28  rdfs:label "Ulricus Weltzli" ;
29  ri:hasTitle "Vizekanzler" .
30 √ <ri-data:StephanusKolbeck> a ri:Person ;
31  rdfs:label "Stephanus Kolbeck" ;
32  ri:hasTitle "Registrator" .
```

<https://github.com/chpollin/GM-DH/blob/main/scripts/gpt-api-python/regesten/output/regest3.rdf>

# Custom GPTs



## RegestenCrafter

Transforms German Plain text 'Regest' into Turtle RDF following specific rules.

By Digital Humanities Craft OG

<https://chat.openai.com/g/g-UAHD0ZIJ-regestencrafter>

Example:

<https://chat.openai.com/share/d542600b-7a12-4d05-a696-b27d24319829>

RegestenCrafter  
Published · Anyone with a link

Create    Configure

Description  
Transforms German Plain text 'Regest' into Turtle RDF following specific rules.

Instructions  
You will act as a skilled expert automaton that is proficient in transforming unstructured text, specifically given German "Regest" (abstract) of the Regesta Imperii (RI), into Turtle RDF. Analyze the provided text based on the mapping rules I have shared and then execute the transformation to produce Turtle RDF based on the RI-Ontology, ensuring you adhere to the guidelines and only annotate if certain.

\*\*Mapping rules\*\*:

Conversation starters

- Transform this Regest into Turtle RDF:
- How would you map this document's entities?
- Create Turtle RDF for this historical text:
- Analyze this Regest for RDF conversion:

Knowledge  
Conversations with your GPT may include file contents. Files can be downloaded when code interpreter is enabled.

ri-ontology.txt  
Document

The following files are only available for Code Interpreter:

ri-shacl-shape.ttl  
File

# Key Findings

**Das Uninteressanteste, was man mit GPT machen kann, ist Text generieren.**

GPT-4 ist ein guter Modellierungsassistent.

GPT-4 ist gut in der Datentransformation

**GPT-4 kann in Workflows (sehr weit gedacht!) eingebettet werden (und das ist der spannende Teil!).**

Was ich gemacht habe, ist eigentlich ein “alter Hut”, denn man kann es mit Custom GPTs und der Assistance API noch besser machen (Custom Knowledge)