Enhancing Rhetorical Figure Annotation: An Ontology-Based Web Application with RAG Integration

Ramona Kühn[1], Jelena Mitrović[1,2], Michael Granitzer[1]

[1]University of Passau [2]Institute for AI Research and Development of Serbia

Correspondence: {ramona.kuehn, jelena.mitrovic, michael.granitzer}@uni-passau.de

Abstract

Rhetorical figures play an important role in our communication. They are used to convey subtle, implicit meaning, or to emphasize statements. We notice them in hate speech, fake news, and propaganda. By improving the systems for computational detection of rhetorical figures, we can also improve tasks such as hate speech and fake news detection, sentiment analysis, opinion mining, or argument mining. Unfortunately, there is a lack of annotated data, as well as qualified annotators that would help us build large corpora to train machine learning models for the detection of rhetorical figures. The situation is particularly difficult in languages other than English, and for rhetorical figures other than metaphor, sarcasm, and irony. To overcome this issue, we develop a web application called "Find your Figure" that facilitates the identification and annotation of German rhetorical figures. The application is based on the German Rhetorical ontology GRhOOT which we have specially adapted for this purpose. In addition, we improve the user experience with Retrieval Augmented Generation (RAG). In this paper, we present the restructuring of the ontology, the development of the web application, and the built-in RAG pipeline. We also identify the optimal RAG settings for our application. Our approach is one of the first to practically use rhetorical ontologies in combination with RAG and shows promising results.

1 Introduction

The consideration of rhetorical figures from a computational perspective is important, as they can convey subtle, implicit meanings (e.g., metaphors), make texts more memorable, or add emphasis to the message (e.g., through the repetition of words). Their detection in text regularly leads to improved performance of various NLP applications, such as hate speech (Lemmens et al., 2021) or fake news detection (Dwivedi and Wankhade, 2021; Fang et al., 2019; Rubin et al., 2016; Troiano et al., 2018), sentiment analysis (Ranganath et al., 2018), or persuasive communication in general (Anzilotti, 1982; Gass and Seiter, 2022; Ranganath et al., 2018).

Unfortunately, most computational approaches for the detection of rhetorical figures struggle with lower performance than they could actually achieve, e.g., Bhattasali et al. (2015); Dubremetz and Nivre (2015); Zhu et al. (2022). Kühn and Mitrović (2024a) identify the major challenges for researchers in this domain and point out why their approaches often suffer from lower performance. One of the main reasons is the lack of data or unbalanced datasets. In these datasets, the number of instances without rhetorical figures is higher than those containing them, as shown in Adewumi et al. (2021); Bhattasali et al. (2015); Dubremetz and Nivre (2017); Kühn et al. (2023); Kühn et al. (2024b); Ranganath et al. (2018).

Another major problem is that most detection approaches focus on English. This means that (annotated) data in other languages are even scarcer. In addition, annotators qualified in this research field are not easy to find, and the quality of their annotations varies greatly due to the ambiguous nature of rhetorical figures (Strommer, 2011; Gavidia et al., 2022).

We overcome those two major challenges in the domain of rhetorical figure detection by developing an interactive web application for the collection of rhetorical figures in German. Users without linguistic knowledge are guided through an interactive process in which they can determine the name of a rhetorical figure in a text and annotate it. In addition to the guided input, we offer a chat-like feature, so users can interact with a large language model (LLM). The LLM obtains domain-specific knowledge about rhetorical figures through retrieval augmented generation (RAG) (Lewis et al., 2020). Although current LLMs are powerful, they often struggle even in English with generating and

Proceedings of the 31st International Conference on Computational Linguistics, pages 8774–8786
January 19–24, 2025.

detecting rhetorical figures, especially syntactic figures. An example is shown in Fig. 1, where ChatGPT is asked to generate sentences that contain an epiphora. Epiphora is a rhetorical figure, where the last word of each sentence is repeated at the end of the next sentence (e.g., "When I was a child, I spoke as a child, I understood as a child, I thought as a child"). Although the LLM in our example is aware of the definition of epiphora, it fails to apply it in a sentence. The situation is even worse in languages other than English.

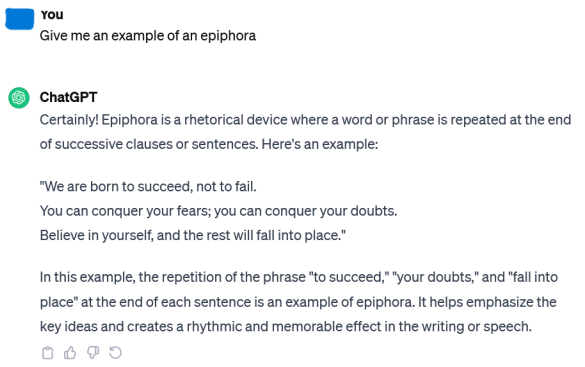

Figure 1: Powerful LLMs such as ChatGPT still fail when asked for examples of certain rhetorical figures. In this example, the LLM claims that the last words in the sentences are repeated.

RAG reduces hallucinations and enables LLMs to obtain domain-specific knowledge, making it especially useful in areas where fine-tuning is constrained by limited data. Rhetorical figures are such a field with scarce annotated data (Dubremetz and Nivre, 2017). RAG requires only an external knowledge source, often a document. This makes it easier to adapt RAG if the information in the document changes. We use an adapted version of the German GRhOOT ontology of rhetorical figures (Kühn et al., 2022) as an external source of knowledge. We experiment with different configurations and chunking methods to find the optimal setting for our purpose. We evaluate RAG's performance with the Ragas framework (Es et al., 2023) and a ground truth file based on ontological competency questions (Grüninger and Fox, 1995; Allemang and Hendler, 2011; Noy et al., 2001; Hristozova and Sterling, 2002). Competency questions are specifications of ontologies ensuring that they have enough knowledge to answer the questions of users. They are usually formulated during the ontology design process.

Our contributions in this paper are as follows:

- **Web Application Development:** We develop a web application called "Find your Figure" to overcome the lack of annotated data for rhetorical figures in German.
- **Ontology Restructuring:** We restructure and simplify the German GRhOOT ontology. The simplification of relations and properties is called reification in the context of ontologies. The reified GRhOOT ontology serves as the basis for our web application.
- **LLM Integration with RAG:** We integrate an LLM with RAG to ensure natural interaction with the ontology, and test different settings and chunking methods to find the most effective configuration for our needs.
- **Performance Evaluation:** We evaluate the performance of the integrated RAG LLM using the competency questions of the GRhOOT ontology.

The code and supplementary material are available online.[1]

2 Related Work

The scarcity of annotated data and the high imbalance between classes with or without rhetorical figures is a well-known issue in the domain of computational detection of rhetorical figures (Dubremetz and Nivre, 2015; Bhattasali et al., 2015; Dubremetz and Nivre, 2017; Ranganath et al., 2018; Adewumi et al., 2021; Kühn et al., 2023; Kühn and Mitrović, 2024a). Unfortunately, efforts to overcome the problem are limited. Chakrabarty et al. (2022) use OpenAI's gpt-3 to generate text containing a rhetorical figure, but their method still requires three human annotators to oversee the output. Additionally, annotations of rhetorical figures often have a high variability, because annotators cannot agree on the existence of a figure in particular cases (Strommer, 2011; Dubremetz and Nivre, 2015; Troiano et al., 2018). This problem can be directly linked to the lack of consensus and multiple varying definitions, inconsistent names, and spellings of rhetorical figures, which is a well-known problem (Harris et al., 2018; Gavidia et al., 2022; Kühn and Mitrović, 2024b).

[1]The code is available on GitHub: https://github.com/kuehnram/FindYourFigure.

Rhetorical ontologies aim to address this issue by building formal models to standardize definitions and descriptions. Important ontologies in this domain are the English RhetFig (Kelly et al., 2010), Ploke (Wang et al., 2021), and ESTHER ontology (Kühn et al., 2023), the Serbian RetFig (Mladenović and Mitrović, 2013), and the German GRhOOT ontology (Kühn et al., 2022). However, none of those ontologies has yet been applied in a practical scenario for collecting or annotating rhetorical figures.

Our work addresses this gap by implementing a web application based on the German GRhOOT ontology. In addition, we use the ontology to enhance the context of an LLM through RAG. Lewis et al. (2020) show the effectiveness of RAG in different NLP tasks such as question answering and generation, while outperforming pre-trained models. Zhao et al. (2024) present more domains in which RAG is useful, such as the video, audio, or text domain.

3 Reification of the GRhOOT ontology

The German GRhOOT ontology was developed by (Kühn et al., 2022). It contains the formal description of 110 common German rhetorical figures. Each figure is specified based on the way it is constructed. The figure epiphora shall serve as an example here. In an epiphora, the last word of each sentence is repeated at the end of the next sentence. In the ontology, these properties are expressed by the relations

```
:Epiphora :isInPosition :Beginning .
:Epiphora :isInArea :Sentence .
:Epiphora :isRepeatableElementOfSameForm
                    :Word .
```

Rhetorical figures in the ontology also contain relations to express a textual definition, example sentences, and names of the figure in other languages. An example of the complete formal model of epiphora is shown in Fig. 6 in the Appendix in Section A. While building the web application and specifying user needs, we identify opportunities for further enhancement, particularly in simplifying the relations within the ontology. For this reason, we create an adapted version of the GRhOOT ontology. The main changes are the **reification** of relations and a more **fine-grained description** of definitions, authors, example sentences, and their sources. In addition, we model rhetorical figures as **classes** instead of individuals. Reification involves breaking down properties and relationships into more fine-grained components while offering a more detailed and flexible representation of the ontology. Although this approach increases complexity (Stevens and Lord, 2010), it allows for more precise querying and filtering of attributes. Consider for example the construction relationship `:isRepeatableElementOfSameForm :Word .`

When a user wants to list all figures that contain a repetition of a word, it would be a cumbersome operation to filter the relation names for the `isRepeatable` substring. We want to make the search more intuitive by breaking compound relations into smaller, fine-grained ones. For example, we split the repetition relation into three relations (RF denotes Rhetorical Figure):

```
:RF :hasOperation :Repetition .
:RF :affectedElement :Word .
:RF :operationalForm :SameForm .
```

A comparison between old and new relations is shown in Table 1 in Example (a). There were several relations of this form that we adapted accordingly.

This change allows users of the web application to generally filter for figures with the same operations, the same affected elements, or the same operational forms. Reification often makes relations more implicit, requiring an understanding of which relations belong together. However, users are guided through our web interface, so we do not see any drawbacks.

Moreover, we adapt the ontology to reflect the hierarchical structure of rhetorical figures. In the original GRhOOT, rhetorical figures are modeled as individuals, similar to the Serbian RetFig ontology (Mladenović and Mitrović, 2013). However, the English RhetFig (Kelly et al., 2010) and the English Ploke ontology (Wang et al., 2021) model figures as classes. We also decided to convert the rhetorical figures from individuals to classes to better align with established hierarchies in rhetorical theory (Harris and Di Marco, 2017; O'Reilly et al., 2018). This adjustment is particularly useful in annotation tasks, where it is important to recognize that one figure may be a more specific form of another figure, and both annotations can be valid. For example, the figure antimetabole is a more specific form of the figure chiasmus and can therefore be modeled as a subclass of chiasmus.

Another major change is the adaption of textual definitions and example sentences that are no longer modeled as property relations. We convert

the actual definitions and examples into individuals, called e.g., `DefinitionAnaphora1`, or `Example1`. This way, we can add multiple definitions to a rhetorical figure, reflecting the great variety of definitions from different authors and perspectives. Furthermore, by not directly naming examples according to the figure, e.g., `ExampleAnaphora1` but only `Example1`, we can reuse the example for other figures, because multiple figures are often co-located. This means that an example can be assigned to the figure anaphora as well as to another related figure (symploke, epiphora, parallelism, etc.). In addition, the new structure reduces redundancy in the ontology. The new construction of the examples is shown in Table 1 in Example (b) for the definitions and Example (c) for the textual examples.

Fig. 7 in the Appendix in Section A shows an example of the figure epiphora in the reified GRhOOT ontology.

4 Ontology-Based Web Application for Rhetorical Figure Annotation

Our overall goal is to improve the computational detection of rhetorical figures by collecting more annotated instances of rhetorical figures. We demonstrated the important role of rhetorical figures and how their detection can improve many NLP systems in Section 1. The interaction with the ontology through the web application is as natural and intuitive as possible without the need for linguistic knowledge or knowledge about ontological details. When users encounter a sentence in which they suspect a rhetorical figure, they can use our web application to determine its name and function. The application is based on the Python Flask[2] framework and uses an SQLite [3] database. The Flask framework is suited for lightweight web applications such as ours.

4.1 Pages of the Web Application

The application encompasses the following five pages.

- `create.html`: On this page, users have the possibility to enter a sentence with a rhetorical figure without annotating it. Users submit a text or sentences, context (e.g., preceding sentences, description of the situation), author, and source of the text. The example is stored in the database for later annotation by other users who do not have an own example but choose a random one from the database.

- `FyF.html`: This is the main page of our "Find your Figure" application. It is shown in Fig. 3. Users choose to submit their text/sentence or choose a random one from the database previously submitted by users on the `create.html` page. The option to enter an own text also includes specifying context, author, and source. The users then select the properties of the text from a dropdown list that best describes the pattern in the submitted text. Properties are extracted relations from the ontology, such as `operation` (e.g., `repetition`), `affected element` (e.g., `word`). Users always have the possibility to choose `No idea` (`Keines davon/Weiß nicht`) if they are not sure about the property. After the users submit the information, the properties are translated into a SPARQL query in the backend and executed on the ontology. If matching figures are found, they are presented along with a definition and examples of the figure in the frontend. The users can choose one or more figures they consider appropriate as shown in Fig. 4. The text, context, author, source, and annotated figure are then written to the SQLite database. The database scheme is shown in Fig. 2.

- `llm.html`: As the annotation process in `FyF.html` still requires basic knowledge of linguistic concepts, we integrate a chatbot-like feature for a more natural interaction between users and ontology. Users simply submit the example text they want to annotate and describe its properties to the LLM. It offers a field for text, context, author, and source, or the possibility to load an example from the database. Instead of the drop-down list, a text field is presented for the LLM prompt. The answers are generated by the LLM with RAG extended context. The different setups to find the best RAG parameters are described in Section 5.

- `figure_info.html`: This page provides an informative overview of rhetorical figures. Users can select the name of a rhetorical figure from a dropdown list. The application then presents definitions of the figure and example sentences. As the elements of the list and

[2] https://flask.palletsprojects.com/en/3.0.x/
[3] https://www.sqlite.org/

Example	Original GRhOOT	Reified GRhOOT		
(a)	:RF :isRepeatableElementOfSameForm :Word	:RF	:hasOperation	:Repetition ;
			:affectedElement	:Word ;
			:hasOperationForm	:SameForm .
(b)	:RF rdfs:comment :Repetition of the first word [...]	:RF	:hasDefinition	:DefinitionRF1 .
		:DefinitionRF1	:hasAuthor	"Gerd Berner" ;
			:isDefinition	"Repetition of the first word [...]" .
(c)	:RF :isExample :The water [...]. The water [...] (J. W. Goethe, Der Zauberlehrling)	:RF	:hasExample	:Example1 .
		:Example1	:hasAuthor	"Johann Wolfgang von Goethe" ;
			:hasSource	"Der Zauberlehrling" ;
			:isExample	"The water [...]. The water [....] " .

Table 1: Example of reified relations that we changed in the new version of the GRhOOT ontology (RF = Rhetorical Figure).

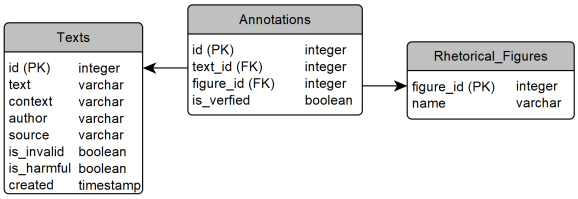

Figure 2: Scheme of the SQL Lite Database. The arrows indicate foreign key (FK) relations. PK denotes primary keys.

their information are retrieved from the ontology, it can be easily extended by adapting the ontology.

- `about.html` This page presents our research project and offers an imprint with contact details.

Annotated text submitted by the users can be added to the ontology as examples of the respective figure. However, we did not implement this functionality yet, as we first want to verify that the submitted examples are correct.

4.2 Verification of User Input

We need to verify that

1. users do not violate intellectual property rights when uploading examples,
2. the submitted text is valid and not "gibberish",
3. the assigned rhetorical figures is correct,
4. the submitted text is not harmful or violating, especially when presented to other people for annotation.

It is important to verify that no intellectual property rights are infringed, especially when we later train models on the obtained data. Researchers are aware of those challenges regarding intellectual property rights when training LLMs with large amounts of text (Smits and Borghuis, 2022). However, identifying unauthorized material or a violation of intellectual property is "notoriously difficult to prove" (Chesterman, 2024). To overcome the first challenge (1.), users must indicate at least an author or source of their submitted text. In addition, we raise awareness of the problem by displaying an informational text next to the author and source fields in the `create.html` and `FyF.html` pages. The LLM we are using in the `llm.html` page is used in a generative way, but we prompt it with information from the ontology in a RAG setting to ensure that it is more likely to only present examples from the ontology. Another possibility to make potential copyright infringements tractable is user authentication via login. However, this would increase the threshold to use the app, especially for younger users, e.g., school children.

Furthermore, we want to verify that users only submit valid text which is then written to the database (2.). We cannot rely (solely) on common grammar or spell checks, as odd grammatical constructs or omitted letters can be a feature of rhetorical figures. We use a combination of a language detector (`langdetect`) to identify if the submitted example is German, a measure of the text length ($10 \leq$ `text_length` ≤ 1000, and a grammar checker that supports German (`language-tool-python`[4]). If one of those three checks fails, we ask gpt-3.5-turbo to evaluate if the text is "gibberish". If the answer is positive, we show a notification to the users if they really want to submit the example, making them aware of potential problems. If they submit the example anyhow, we will flag the example in the database in the column `is_invalid`, such that an administrator can later check the validity.

Ensuring that the rhetorical figures assigned by the users are correct (3.) is a highly challenging

[4] https://pypi.org/project/language-tool-python/

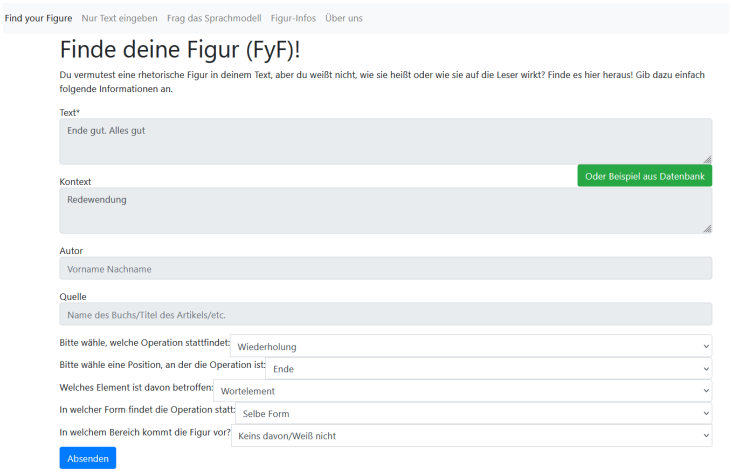

Figure 3: The page `FyF.html` helps users to find the name of a rhetorical figure hidden in a text. Properties can be selected from the dropdown lists.

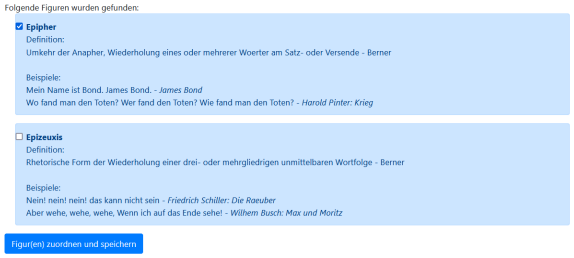

Figure 4: The result of the user's submission. Rhetorical figures fulfilling the properties are displayed along with the option to select suitable ones.

task. A different detection algorithm would be required for each figure. However, especially for figures other than metaphor, irony, and sarcasm, if any approaches exist, they often have lower performance (Kühn et al., 2024a). In addition, each approach is language dependent and often requires high manual efforts to achieve acceptable performance (Mladenović et al., 2017). Using existing lexical resources for rule-based approaches, such as German wordnets, is barely an option as they do not contain enough information for a reliable detection (Kühn et al., 2023) and are in general difficult to maintain (Mladenović et al., 2014). Even approaches based on language models are difficult to implement as data is too scarce or often too imbalanced for training models (Dubremetz and Nivre, 2015; Kühn et al., 2024b). However, we use a rule-based check if a figure of perfect lexical repetition is assigned and verify that at least two words are repeated in the same form. Nevertheless, we still have to rely on manual checks by an administrator for all figures. For this purpose, the column `is_verified` in table `Annotation` (see Fig. 2) is intended to mark if an example has already been approved.

Users can annotate random examples from the database on the `FyF.html` page. However, we want to ensure that potentially harmful content is not presented (4.), especially when the web application is used by young children who learn about rhetorical figures. We do not prevent per se the submission of harmful content, as certain rhetorical figures occur frequently in hate speech, e.g., sarcasm (Frenda et al., 2023). However, we exclude those examples to be shown to other users. We introduce the boolean field `is_harmful` to mark those examples and prevent their retrieval for annotation in the `FyF.html` page. We have not yet implemented any hate speech detection mechanism yet. However, we plan to run a daily check on the database.

5 RAG Integration: Parameter Testing and Evaluation

RAG uses an external knowledge source to enhance the context of an LLM. LLMs still struggle to generate rhetorical figures as we showed in Fig. 1. Another challenge in this domain are the many different and varying definitions of rhetorical figures. Though, we want the integrated LLM to respond to the web application users with the specific definitions we use in our ontology to ensure consistent annotation guidelines.

Fig. 5 illustrates the RAG pipeline with its individual steps. It also shows the different parameters used in our experiment to determine the optimal settings for the web application. The input is the

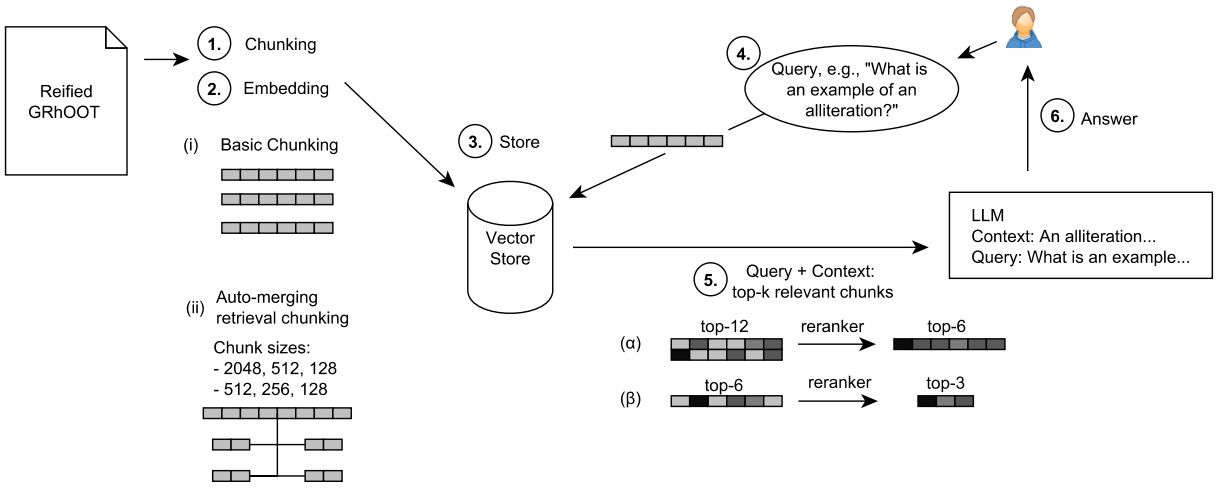

Figure 5: Overview of the integrated RAG Pipeline for the GRhOOT ontology.

reified GRhOOT ontology which is also the basis of the web application. After chunking and embedding the information (steps 1 and 2), it is stored in a vector store (step 3). When a user asks the system a question about rhetorical figures in step 4 (e.g., "What is an alliteration?"), the question is embedded and compared to the content in the vector store. The top-k relevant chunks are retrieved in step 5, often in combination with a reranker, and added as context to the query, e.g., actual examples of an alliteration. The LLM receives the context along with the original question. In step 6, the LLM answers the users' questions with a reduced probability of hallucinations and knowledge of the domain, which are rhetorical figures in our case.

To find the best setting, we experiment with different chunk sizes ([2048],[2048, 512, 128], and [512, 256, 128]). It is a known phenomenon called "lost in the middle" that content stored in the middle of a chunk is more difficult for the LLM to recall (Liu et al., 2024). Therefore, we test different sizes to avoid this problem. As chunking technique, we investigate basic chunking (i) and auto-merging retrieval chunking (ii) with the `HierarchicalNodeParser` by LlamaIndex.[5] It is an advanced chunking technique, where a smaller chunk that contains relevant information is merged into the parent chunk and provided as context.

We use the multilingual model bge-m3[6] for the embeddings.

Usually, the vectorized index is stored in a suitable vector database, which offers additional benefits, especially when working with multiple documents. However, as we only focus on a compact and relatively small ontology, we do not need a vector database. Instead, we store the index in a file. This approach reduces the number of variables that could affect the output, as we are not relying on optimization strategies of database vendors.

Furthermore, we use a reranker (BAAI/bge-reranker-large) in all settings to improve the results. In setting (α), we first retrieve the top-12 chunks, while the reranker selects the top-6 chunks. In setting (β), we retrieve the top-6 chunks, while the reranker selects the top-3 chunks (β).

We are using OpenAI's gpt-3.5-turbo as LLM, as it shows good results in German. We set the temperature to 0.1 to obtain stable responses.

5.1 Evaluation Setup for the RAG pipeline

RAG evaluation still poses a challenge. We use the Ragas[7] framework for evaluation. It requires a file of questions, answers generated by the LLM in the last step of the RAG pipeline, context information from the original document, and ground truth answers. Most approaches rely on LLM-generated ground truths that are then again evaluated by an LLM.

Ontologies in general and especially the GRhOOT ontology have a big advantage here. We can use ontological competency questions (CQ) (Grüninger and Fox, 1995; Noy et al., 2001; Hristozova and Sterling, 2002; Allemang and Hendler, 2011) their respective answers extracted

[5] https://docs.llamaindex.ai/en/stable/api_reference/node_parsers/hierarchical/

[6] https://huggingface.co/BAAI/bge-m3

[7] https://github.com/explodinggradients/ragas

Document	Chunk Sizes	Chunking Method	reranker	faithf.	c_precision	c_recall	a_correctn.	a_similarity	a_relevancy
Reified GRhOOT	2048	Basic	top-12/6	**0.9023**	0.5548	**0.9857**	0.7355	0.8655	**0.9489**
	2048	Basic	top-6/3	0.8342	0.8496	0.9098	0.5678	**0.9673**	0.7119
	2048, 512, 128	AMR	top-12/6	0.8760	0.5762	0.9714	0.6481	0.8542	0.8616
	2048, 512, 128	AMR	top-6/3	0.7612	0.9009	0.9254	**0.8619**	0.6625	0.8041
	512, 256, 128	AMR	top-12/6	0.8817	0.8446	0.6571	0.8311	0.7099	0.8889
	512, 256, 128	AMR	top-6/3	0.8622	**0.9230**	0.9190	0.5969	0.9608	0.5587

Table 2: Results of our RAG experiments on the reified ontology with different settings. AMR stands for automerging retrieval, an advanced chunking technique. The best scores per column are highlighted in bold.

from the ontology to generate the ground truth file for the Ragas evaluation. Unfortunately, the GRhOOT ontology comes only with five CQs. However, Alharbi et al. (2023) and Ciroku et al. (2024) demonstrate that OpenAI gpt-4 can generate competency questions after the ontology has been created. Therefore, we also use gpt-4 to generate further CQs for the reified GRhOOT ontology. In contrast to Alharbi et al. (2023), we skip the triple extraction and provide only the formalization of one rhetorical figure as context to the LLM. The LLM is then able to formulate appropriate questions. Additionally, we create template questions asking for properties of rhetorical figures, i.e., "What is <*property*> of the rhetorical figure <*figure_name*>", e.g., "What is an *example* of the rhetorical figure *anaphora*?". With those methods, we obtain 70 CQs. We formulate matching SPARQL queries to retrieve the answers from the ontology. The context is manually extracted from the ontology. From this information, we construct our ground truth file in the required format for the Ragas framework. The answers of the LLM still require post-processing before we can use them in the Ragas framework as the LLM tends to add additional quotation marks around figure names or examples but often does not close them. The post-processing step is only necessary for the evaluation. It is not required in the production mode of our web application.

For the evaluation, we use the pre-defined metrics from the Ragas framework. A detailed description can be found online, [8] but we describe them shortly here for clarification. We choose the following Ragas metrics:

- `Faithfulness`: Describes the extent to which the answer is grounded in the context.
- `Context precision`: Measures if relevant chunks are ranked higher.
- `Context recall`: Measures if the retrieved context is present in the ground truth.
- `Answer relevancy`: Measures the relevancy of the answer to the question by calculating mean cosine similarity between the actual question and artificial questions generated by an LLM based on the answer.
- `Answer correctness`: Measures how semantically and factually similar the answer is to the ground truth.
- `Answer (semantic) similarity`: Measures the semantic similarity between the ground truth and the LLM's answer based on the cosine similarity of the embeddings.

From a user's perspective, the performance of answer metrics, particularly `answer correctness`, is more important than the performance of context metrics. Therefore, we focus especially on these metrics.

5.2 Evaluation Results

In the first review of the answers, we notice that the LLM sometimes responds in English instead of German. For this reason, we add to the prompt the request to only answer in German (step 4 in Fig. 5): "Bitte antworte nur auf Deutsch!" ("Please answer only in German!").

Table 2 shows the result for the different settings. Surprisingly, advanced chunking techniques do not increase the performance. As best setting, we identify basic chunking with a chunk size of 2048, where the top-12 chunks are selected first and then filtered for the top-6 by the reranker. Only the context precision is low in this setting. However, as we mentioned, we focus more on answer metrics. When reviewing the results, we notice deviations in `answer correctness` and `answer similarity`,

[8] https://docs.ragas.io/en/latest/concepts/metrics/index.html

even when the LLM's answer is correct and semantically similar to the ground truth. We identify the LLM's circumscription of the answer as the problem. For example, consider the following case:

Question: "What is the name of the figure where the first letter of each word sounds the same?"
Ground truth: "Alliteration"
LLM's answer (translated from German): "The name of the figure where the first letters of each word sounds the same is 'Alliteration'"

This answer leads to a reduced `answer correctness` and `answer similarity` because of the wordiness of the LLM's answer. However, we do not consider this as an issue since the answer is correct and users would probably prefer a complete sentence over a single-word-response.

Other examples where the LLM is correct but creates reduced `answer correctness` and `answer similarity` are a different choice of words. For example, the LLM answers with this definition for a rhetorical figure

LLM's answer: "Eine Wiederholung des Anfangslautes benachbarter oder nah beieinanderstehender Worte in einem Satz oder Vers"
Ground truth: "Gleichklingender Anlaut der betonten Silben innerhalb einer Wortgruppe",

where "Gleichklingender Anlaut" means the same as "Wiederholung des Anfangslauts". We assume that it is caused by the definitions in the ontology taken from dictionaries and books, while the LLM uses more "modern" language and simpler expressions.

We see problems when the LLM is asked to answer questions that require aggregation of information (e.g., "What are the linguistic groups defined in the ontology?""), which requires reasoning over multiple chunks. This finding is in line with the experience of Alharbi et al. (2023) that LLMs generally tend to fail in such tasks. Unfortunately, RAG can do little to change this. Nevertheless, the high values of the metrics show the efficiency of RAG on ontologies. Even without special adaptions to the ontological structure, we achieve satisfying results.

6 Conclusion

The development of the web application to collect rhetorical figures is an important step to overcome the scarcity of annotated data in the field of the computational detection of rhetorical figures, especially in German. The web application allows users to specify the properties of a text and assists them to name and annotate the rhetorical figure hidden in it. Furthermore, the web application serves as an information collection about rhetorical figures, where users can learn the definitions and see examples. The web application, which is built on top of the GRhOOT ontology, is one of the first approaches to practically use a rhetorical figure ontology for figure annotation. It also contains verification functions for the user input. The integrated RAG pipeline allows users to use an LLM-powered chat for the interaction with the GRhOOT ontology. One of our main objectives for the future is to publish the web application and promote it to potential users. We can then evaluate the features of the application that are most beneficial to users and learn about their behavior, for example, if they prefer the chat function with the integrated RAG model or the structured drop-down fields.

In addition, we will observe the performance of the RAG pipeline. When we collect more examples from users through the web application, it is possible to add them to the ontology and update the vector store to improve the performance of the RAG pipeline. We also envision gamification elements and user sessions to store their achievements to keep them engaged. In addition, we will extend our verification methods for the user input. We also plan to show retrieved chunks to the users so they can compare the information from the ontology with the LLM's answer.

Nevertheless, the current version that combines annotation capabilities and educational resources makes our application a valuable tool in the domain of computational detection of rhetorical figures, as well as a possible interactive resource in education.

7 Limitations

Our web application for identifying rhetorical figures has some limitations. It is better suited to identify figures with obvious rhetorical features, e.g., figures with repeating elements, than for figures relying on transferred meanings, such as metaphors. However, we see this rather as a limitation on the side of the users. For most persons without lin-

guistic knowledge of rhetorical figures, it is easier to spot and describe obvious lexical patterns than figures with implicit, transferred meaning. Additionally, this app represents only the initial implementation of our envisioned tool. There are many possible features that can be implemented, enhanced, and improved in the future.

8 Ethical Considerations

Regarding the web application, the main concern is the violation of intellectual property rights. Users may submit text from sources they do not have the right to use. Furthermore, the text is then stored in the database and used to train models, even if the original authors did not agree on the distribution of their text. This is not an easy task to solve both from a computational and legal perspective. However, we established methods to encourage users to indicate an author or source of the examples.

Regarding the RAG pipeline, users should be aware that the LLM may produce incorrect answers. We hope to support the users in assessing the truth with our web application, as they can browse the figures to learn about them and with the planned feature to show the retrieved chunks along with the answer of the LLM. This way, users can get a clearer picture if the LLM's answer is correct.

In conclusion, while RAG on rhetorical ontologies holds significant potential for advancing NLP, addressing these ethical concerns is important to ensure their responsible and fair application.

Acknowledgements

The project on which this report is based was funded by the German Federal Ministry of Education and Research (BMBF) under the funding code 01|S20049. The authors are responsible for the content of this publication.

References

Tosin P Adewumi, Roshanak Vadoodi, Aparajita Tripathy, Konstantina Nikolaidou, Foteini Liwicki, and Marcus Liwicki. 2021. Potential idiomatic expression (PIE)-english: Corpus for classes of idioms. *arXiv preprint arXiv:2105.03280*.

Reham Alharbi, Valentina Tamma, Floriana Grasso, and Terry Payne. 2023. An experiment in retrofitting competency questions for existing ontologies. *arXiv preprint arXiv:2311.05662*.

Dean Allemang and James Hendler. 2011. *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier.

Gloria Italiano Anzilotti. 1982. The rhetorical question as an indirect speech device in English and Italian. *Canadian Modern Language Review*, 38(2):290–302. Publisher: University of Toronto Press.

Shohini Bhattasali, Jeremy Cytryn, Elana Feldman, and Joonsuk Park. 2015. Automatic identification of rhetorical questions. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)*, pages 743–749.

Tuhin Chakrabarty, Arkadiy Saakyan, Debanjan Ghosh, and Smaranda Muresan. 2022. Flute: Figurative language understanding through textual explanations. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 7139–7159.

Simon Chesterman. 2024. Good models borrow, great models steal: intellectual property rights and generative ai. *Policy and Society*, page puae006.

Fiorela Ciroku, Jacopo de Berardinis, Jongmo Kim, Albert Merono Penuela, Valentina Presutti, and Elena Simperl. 2024. Revont: Reverse engineering of competency questions from knowledge graphs via language models. *Journal of Web Semantics*.

Marie Dubremetz and Joakim Nivre. 2015. Rhetorical figure detection: The case of chiasmus. In *Proceedings of the fourth workshop on computational linguistics for literature*, pages 23–31.

Marie Dubremetz and Joakim Nivre. 2017. Machine learning for rhetorical figure detection: More chiasmus with less annotation. In *Proceedings of the 21st nordic conference on computational linguistics*, pages 37–45, Gothenburg, Sweden. Association for Computational Linguistics.

Sanjeev M Dwivedi and Sunil B Wankhade. 2021. Survey on fake news detection techniques. In *Image processing and capsule networks: ICIPCN 2020*, pages 342–348. Springer.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Yong Fang, Jian Gao, Cheng Huang, Hua Peng, and Runpu Wu. 2019. Self multi-head attention-based convolutional neural networks for fake news detection. *PloS one*, 14(9):e0222713. Publisher: Public Library of Science San Francisco, CA USA.

Simona Frenda, Viviana Patti, and Paolo Rosso. 2023. When sarcasm hurts: Irony-aware models for abusive language detection. In *International conference of the cross-language evaluation forum for european languages*, pages 34–47. Springer.

Robert H Gass and John S Seiter. 2022. *Persuasion: Social influence and compliance gaining*. Routledge.

Martha Gavidia, Patrick Lee, Anna Feldman, and Jing Peng. 2022. Cats are fuzzy pets: A corpus and analysis of potentially euphemistic terms. *arXiv preprint arXiv:2205.02728*.

Michael Grüninger and Mark S Fox. 1995. Methodology for the design and evaluation of ontologies. Publisher: Citeseer.

Randy Allen Harris and Chrysanne Di Marco. 2017. Rhetorical figures, arguments, computation. *Argument & Computation*, 8(3):211–231. Publisher: IOS Press.

Randy Allen Harris, Chrysanne Di Marco, Sebastian Ruan, and Cliff O'Reilly. 2018. An annotation scheme for rhetorical figures. *Argument & Computation*, 9(2):155–175.

Maia Hristozova and Leon Sterling. 2002. An eXtreme method for developing lightweight ontologies. In *Workshop on Ontologies in Agent Systems, 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems,(Bologna, Italy, 2002)*.

Ashley R Kelly, Nike A Abbott, Randy Allen Harris, Chrysanne DiMarco, and David R Cheriton. 2010. Toward an ontology of rhetorical figures. In *Proceedings of the 28th ACM international conference on design of communication*, pages 123–130.

Ramona Kühn and Jelena Mitrović. 2024a. The elephant in the room: Ten challenges of computational detection of rhetorical figures. In *Proceedings of the 4th Workshop on Figurative Language Processing (FigLang 2024)*, pages 45–52.

Ramona Kühn and Jelena Mitrović. 2024b. Status Quo der Entwicklungen von Ontologien Rhetorischer Figuren in Englisch, Deutsch und Serbisch. In *Book of Abstracts - DHd2024*. Zenodo.

Ramona Kühn, Jelena Mitrović, and Michael Granitzer. 2023. Esther: Ontology of rhetorical figures in english. In *Proceedings of the Joint Ontology Workshops 2023 Episode IX: The Quebec Summer of Ontology co-located with the 13th International Conference on Formal Ontology in Information Systems (FOIS 2023)*.

Ramona Kühn, Jelena Mitrović, and Michael Granitzer. 2024a. Computational approaches to the detection of lesser-known rhetorical figures: A systematic survey and research challenges. *arXiv preprint arXiv:2406.16674*.

Ramona Kühn, Khouloud Saadi, Jelena Mitrović, and Michael Granitzer. 2024b. Using pre-trained language models in an end-to-end pipeline for antithesis detection. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17310–17320.

Ramona Kühn, Jelena Mitrovic, and Michael Granitzer. 2023. Hidden in plain sight: Can german wiktionary and wordnets facilitate the detection of antithesis? In *Proceedings of the 12th global wordnet conference*, pages 106–116.

Ramona Kühn, Jelena Mitrović, and Michael Granitzer. 2022. GRhOOT: Ontology of rhetorical figures in German. In *Proceedings of the thirteenth language resources and evaluation conference*, pages 4001–4010, Marseille, France. European Language Resources Association.

Jens Lemmens, Ilia Markov, and Walter Daelemans. 2021. Improving hate speech type and target detection with hateful metaphor features. In *Proceedings of the fourth workshop on NLP for internet freedom: Censorship, disinformation, and propaganda*, pages 7–16.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Miljana Mladenović, Cvetana Krstev, Jelena Mitrović, and Ranka Stanković. 2017. Using lexical resources for irony and sarcasm classification. In *Proceedings of the 8th Balkan Conference in Informatics*, pages 1–8.

Miljana Mladenović, Jelena Mitrović, and Cvetana Krstev. 2014. Developing and maintaining a wordnet: Procedures and tools. In *Proceedings of the Seventh Global Wordnet Conference*, pages 55–62.

Miljana Mladenović and Jelena Mitrović. 2013. Ontology of rhetorical figures for serbian. In *Text, speech, and dialogue*, pages 386–393, Berlin, Heidelberg. Springer.

Natalya F Noy, Deborah L McGuinness, and others. 2001. Ontology development 101: A guide to creating your first ontology.

Cliff O'Reilly, Yetian Wang, Katherine Tu, Sarah Bott, Paulo Pacheco, Tyler William Black, and Randy Allen Harris. 2018. Arguments in gradatio, incrementum and climax; a climax ontology. In *Proceedings of the18th workshop on computational models of natural argument. Academic press*.

Suhas Ranganath, Xia Hu, Jiliang Tang, Suhang Wang, and Huan Liu. 2018. Understanding and identifying rhetorical questions in social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(2):1–22.

Victoria L Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the second workshop on computational approaches to deception detection*, pages 7–17.

Jan Smits and Tijn Borghuis. 2022. Generative ai and intellectual property rights. In *Law and artificial intelligence: regulating AI and applying AI in legal practice*, pages 323–344. Springer.

Robert Stevens and Phillip Lord. 2010. Reification of properties in an ontology. *Ontogenesis, An Ontology Tutorial*.

Claus Walter Strommer. 2011. Using rhetorical figures and shallow attributes as a metric of intent in text. Publisher: University of Waterloo.

Enrica Troiano, Carlo Strapparava, Gözde Özbal, and Serra Sinem Tekiroğlu. 2018. A computational exploration of exaggeration. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3296–3304.

Yetian Wang, Randy Allen Harris, and Daniel M Berry. 2021. An ontology for ploke: Rhetorical figures of lexical repetitions. In *JOWO*.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.

Dawei Zhu, Qiusi Zhan, Zhejian Zhou, Yifan Song, Jiebin Zhang, and Sujian Li. 2022. ConFiguRe: Exploring discourse-level chinese figures of speech. *arXiv preprint arXiv:2209.07678*.

A Appendix

Fig. 6 shows the graphical illustration of the figure epiphora in the original GRhOOT ontology by Kühn et al. (2022). Classes are shown in blue boxes, individuals in purple boxes, and the blue arrows represent relations. The adapted, reified version of an epiphora is shown in Fig. 7. Individuals are now modeled as classes, while definitions are specific instances. In addition, we simplified the long relations with compound semantics to make them more explicit.

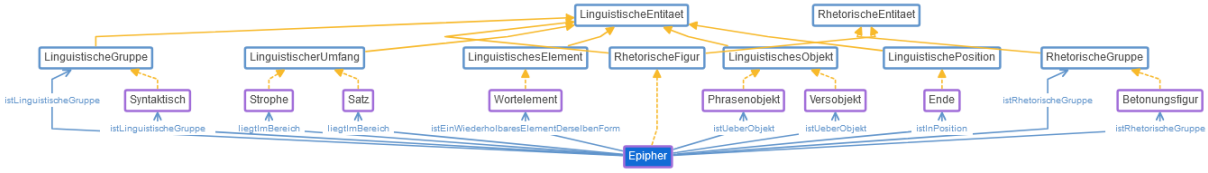

Figure 6: The formal model of an epiphora in the GRhOOT ontology, illustrating the relations between the construction properties.

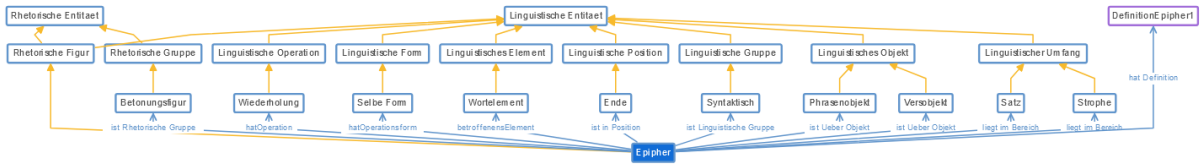

Figure 7: The formal model of an epiphora in the reified GRhOOT ontology. Compared to the model in the original GRhOOT in Fig.6, relations are simplified. In addition, the concepts are now modeled as classes, illustrated by blue boxes instead of purple boxes that represent individuals.