ESTHER: Ontology of Rhetorical Figures in English

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

[1]*University of Passau, Innstrasse 43, 94032 Passau, Germany*
[2]*Institute for Artificial Intelligence Research and Development of Serbia, Fruškogorska 1 21000 Novi Sad, Serbia*

Abstract
ESTHER, the **E**nglish on**t**ology of r**he**torical figu**r**es, is a formal domain ontology of rhetorical figures in the English language. Rhetorical figures convey a subtle meaning or sentiment that often cannot be grasped by modern sentiment analyzers, machine translation systems, or text summarizers due to the lack of annotated datasets and machine-readable formats. Our ontology is a formal documentation of rhetorical figures to improve various natural language processing tasks. Existing ontologies in this domain mainly focus on figures in other languages, such as Serbian or German. Our ESTHER ontology fills this gap by modeling 85 English rhetorical figures. It aligns with existing ontologies to allow interoperability with other languages. In addition, the ESTHER ontology includes hierarchical relations and compositional features of figures to model their dependency relations. Besides identifying the most important steps of the good practice of ontology modeling, we also provide an overview of current tools and methods of ontological evaluation, validation, and documentation for the ESTHER ontology. The ontology does not only serve as a knowledge base and documentation of rhetorical figures but also has a practical use case. We developed a tool to show how the ontology helps annotators without linguistic understanding to annotate instances of rhetorical figures.

Keywords
Rhetorical Figures, Ontology Modeling, Annotation, Natural Language Processing

1. Introduction

Language is a means of communication and has the power to transmit emotions, influence the audience, or increase the memorability of a message. Non-literal language and rhetorical figures play an essential role in this process. The rhetorical figure “metaphor”,[1] for example, creates images in the reader’s mind to explain complex issues with elements from known domains in a non-literal way: “she has eagle eyes” or “he has a heart of gold”. However, to understand the actual meaning, a certain amount of cultural and background knowledge is always required. While it is easier for humans to spot rhetorical figures and grasp their meaning, it is difficult to computationally detect them due to several reasons:

- First, there is a lack of annotated datasets that are large enough for training machine learning models.

Ontology Showcase and Demonstrations Track, 9th Joint Ontology Workshops (JOWO 2023), co-located with FOIS 2023, 19-20 July, 2023, Sherbrooke, Québec, Canada.
*Corresponding author.
ramona.kuehn@uni-passau.de (R. Kühn); jelena.mitrovici@uni-passau.de (J. Mitrović); michael.granitzer@uni-passau.de (M. Granitzer)
0000-0002-9750-0305 (R. Kühn); 0000-0003-3220-8749 (J. Mitrović); 0000-0003-3566-5507 (M. Granitzer)

[1]A glossary of rhetorical figures and examples in natural language can be found in Appendix B.

- Second, there is a lack of formal descriptions of rhetorical figures as even the informal definitions are heterogeneous, and the same figure can have different names.
- Third, there is a lack of knowledgeable human annotators and domain experts on the one hand and a lack of formal rules for computer-assisted annotation on the other hand.

The detection of rhetorical figures is however relevant in the context of Natural Language Processing (NLP), as each rhetorical figure has a function [1, 2], e.g., make text more memorable, or fake news more credible. Understanding the function and meaning of rhetorical figures and being able to detect them automatically contributes to the improvement of several NLP tasks, such as sentiment analysis [3], argumentation mining [4], propaganda detection [5], hate speech detection [6], or text summarization [7].

We modeled a formal domain ontology of rhetorical figures in the English language, called ESTHER. It is a source of both human- and machine-readable knowledge. For example, it can support the annotation of rhetorical figures in a text. Furthermore, it helps to derive rules from the formal definitions, e.g., for the generation or rule-based detection of rhetorical figures. With more datasets of rhetorical figures, classifiers can be trained, leading to a major improvement of the above-mentioned NLP tasks.

Our methodology for the building process is guided by the good standards of ontology development. We examined existing ontologies in the domain of rhetorical figures and decided to align with the Serbian RetFig [8] and the German GRhOOT ontology [9] to guarantee interoperability between different languages. Additionally, we included the concept of hierarchical relations between figures (e.g., every text that contains figure A always contains figure B) from Harris et al. [10].

We also provide an overview of different methods and tools that are available for ontology evaluation, validation, and documentation. Furthermore, we show the relevance of the ontology in the context of NLP by developing a terminal-based tool called "FyF: Find your Figure!". It is based on the ontology and guides human annotators through the process of determining the right name for a rhetorical figure.

The remainder of the paper is structured as follows: Section 2 gives an overview of relevant ontologies in the same domain. In Section 3, the methodology, the structure of the ontology, and its properties are described. Section 4 describes the annotation assistant tool that demonstrates the practical use case of the created ontology. The conclusion in Section 5 summarizes our work and gives ideas for future applications of the ontology. The ontology, the code, and documentation are available online.[2]

2. Related Work

A common problem when working with rhetorical figures is their additional non-literal meaning [3, 11] and their varying definitions. Ontologies can help here as they put an end to "inconsistent descriptions", and "ambiguous term definitions" from different sources, and allow computational processing [12]. In the domain of rhetorical figures, different definitions, names,

[2]https://github.com/kuehnram/ESTHER-Ontology

and categorizations for the same figures emerged over the years, leading to several inconsistencies. A formal domain ontology unifies the descriptions and then allows computational detection.

The first approach to modeling ontologies of rhetorical figures in the English language was made by Harris and DiMarco [13], describing linguistic operations and cognitive effects of figures. Later, Kelly et al. [14] extended this ontology within the scope of the project "RhetFig" by adding formal linguistic and rhetorical classes that also serve as a basis for our ESTHER ontology. Harris et al. [10] modeled an ontology focusing on cognitive effects and on the hierarchical connectedness of different figures. They found out that some figures are co-located with others, e.g., the figure anaphora is always a more specific form of the figure ploke. The OWL ontology of O'Reilly et al. [15] focused on three rhetorical figures (gradatio, incrementum, and climax). They also take into account that figures are connected or composed. We will formalize those relations and also include them in our ontology. Wang et al. [16] developed an ontology of the figure ploke. However, those ontologies only model a few figures. Ontologies that model the most common figures in a language are the Serbian RetFig[3] [8] modeling 98 figures and the German GRhOOT ontology [9] with 110 figures common in the German language. The RetFig ontology is actually based on the classification of Kelly et al. [14], whereas the GRhOOT ontology is again based on the Serbian RetFig to guarantee interoperability between the two languages. The rhetorical figures are modeled as instances, whereas the linguistic specificities are defined as concepts. The authors expressed the desire to "build [...] an ontology of rhetorical figures in English and other languages." For figures of repetition, this was fulfilled by the multilingual ontology [17] that combines the English Ploke ontology, the Serbian RetFig, and the German GRhOOT.

Our ESTHER ontology aligns with the structure of RetFig and GRhOOT, additionally implementing hierarchical relations as presented by Harris et al. [10] and O'Reilly et al. [15]. To show a practical application of the ontology, a decision tree for figures of repetition was developed within the GRhOOT ontology. With our ESTHER ontology, we go a step further by implementing a first version of the user interface envisioned by Kühn and Mitrović [18]. It is realized by a terminal-based tool that assists human annotators in determining the names of figures based on the English ESTHER ontology.

3. Modelling ESTHER

In this section, we describe the modeling methodology, the ontology design process, and the structure of the ontology. The modeling of an ontology of rhetorical figures is a challenging task because all the multiple names, spellings, and definitions have to be included. Also, the categorization of figures is not consistent, but needs to be specified in the ontology. Figures can belong to multiple categories and multiple linguistic elements (words, phrases, verses) can be affected by a rhetorical figure. In addition, the relation between figures and the composition of figures has to be considered.

[3]Please note the spelling without "h" compared to the English RhetFig project [14].

3.1. State of the Art of Ontology Modeling

If the ontology development process follows a particular methodology, it has "normally higher quality" [19]. We examined different approaches to ontological modeling ("The working ontologist" [20], "Understanding ontological engineering" [21], "Methontology" [22], "Methodology for the Design and Evaluation of Ontologies" [23], "Ontology development 101" [24], "Ontology: A practical guide" [25], "The enterprise ontology" [26], "eXtreme" [27, 28], "NeOn" [29], "LOT" [30]) and found common approaches to building an ontology. We identified similar main steps that are listed below and described in their respective sections.

1. Plan main tasks, define resources and tools (Section 3.2).
2. Define purpose, scope, and end-users (Section 3.3).
3. Formalize the conceptual model while reusing and integrating parts of existing ontologies (Section 3.4).
4. Evaluate the ontology (Section 3.5).
5. Create documentation and maintain the ontology (Section 3.6).

Often, it is additionally recommended to use competency questions [20, 24, 27] that were developed by Grüninger and Fox [23]. Those questions serve in the beginning as a specification for the developers what the ontology should fulfill. They are specified by the users [27]. During the development process or afterwards, the developers can test if the ontology contains enough information by correctly answering those questions. However, drawbacks are that they cannot cover every functionality of the ontology and that they are often not available, as the modeler of the ontology already knows how the ontology will behave [20]. In our methodology, we also include competency questions. For the integration of other ontologies in step 3, the modeling approach of NeOn [29] helps by describing how existing ontologies can be reused or included. This approach focuses rather on scenarios (building ontology from scratch, reusing, re-engineering, merging) than on steps. For ontology evaluation, Gangemi et al. [31] suggest three dimensions: the structural, functional, and usability profile dimension. For example, they determine the depth of the ontology graph, cycle ratios, satisfaction of users, agreement of domain experts, and many more. Unfortunately, the assessment appears to be rather subjective.

For ontology validation, SPARQL queries are often used in this domain [8, 16]. We consider this to be rather an evaluation than a validation, as the queries are self-defined. However, there are different tools available for formal validation. We identified different ontology validators, such as the OOPS! ontology pitfall scanner [19], or the oQual ontology validator [31]. The OnToology system [32] provides a validator for ontology syntax, similar to the RDF validation of W3C.[4]

For the documentation of ontologies, different tools are available. For example, HTML documentation tools, ontology browsers, and visualizations [33]. We identified the following possible candidates: OnToology [32], LODE [34], Protégé [35], and Widoco [36], which is actually a combination of LODE, WebVowl visualization [37], and OOPS! validation. OnToology also provides documentation. In our case, the syntax checker did not work, and connecting it to GitHub is more complicated than using Widoco. LODE combined with the WebVowl graph is a good documentation approach.

[4]https://www.w3.org/RDF/Validator/

We are aware that there is more than one solution to model a domain ontology [24], especially in the domain of rhetorical figures with a quite diverse knowledge base. Also, the modeling depends on how the ontology will be used later.

3.2. Main Tasks, Resources, and Tools

Our **main task** is to collect information about rhetorical figures in English and to structure them. Furthermore, we need to build the ontology and validate it. We are using **resources** that contain definitions and examples of rhetorical figures. The standard book in this field was written by Lausberg [38]. However, as it is highly interlinked with numerous subsections and mixes languages within entries (German, Greek, Latin, French, English), it is complicated to fully understand it. Therefore, we identified the book of McGuigan [39], which aims to explain rhetorical figures to students and therefore provides understandable definitions, explanations of the functions of a figure, and examples. In addition, Plett [40] offers an overview of several figures. Different online resources like the Silva Rhetoricae [41] and the RhetFig database[5] with its origins in the RhetFig project [14] are a great collection of definitions and examples. The Silva Rhetoricae contains 433 rhetorical figures and provides a definition, related figures, and examples of the usage for most of the figures. The RhetFig database contains 1489 figures and their definitions, synonyms, etymology, linguistic domain, and examples. Most of the definitions are rooted in the Silva Rhetoricae.

We also rely on existing ontologies from the domain of rhetorical figures. For the Serbian RetFig ontology that contains 98 figures, the authors retrieved information by searching novels, poems, and journal texts to find examples of the figures in the Serbian language. However, they do not state how they found the figures in the first step or resources for the Serbian definitions. The authors of the GRhOOT ontology indicate that they used a variety of German resources such as Lausberg's Handbook [38], and literature from Plett [42], besides others. In our ESTHER ontology, we want to focus only on the most common figures for now. Therefore, we rely on the 110 figures from the GRhOOT ontology and searched for English equivalents. As ontology development is an ongoing process, we want to add as many figures as possible to the ontology, to make resources like the Silva Rhetoricae or the RhetFig machine-readable.

As **tool** for modeling, we use the Protégé ontology modeling framework to build the ontology in combination with a text editor, as Pease recommends [25]. To validate the competency questions, we will use SPARQL queries, similar to the validation of other ontologies in the same domain [8, 9, 16]. The tools used for the validation and evaluation of the ontology are presented in Section 3.5.

3.3. Purpose, Scope, Users, and Competency Questions

The **purpose** of the ontology is to create a formal machine-readable model of the most common rhetorical figures in the English language. The **scope** of our ontology are the most known figures but the ultimate goal is to include as many figures as possible. The ontology can then assist its users in determining the proper name for a figure in the process of annotation. In

[5]http://rhetfig.appspot.com/

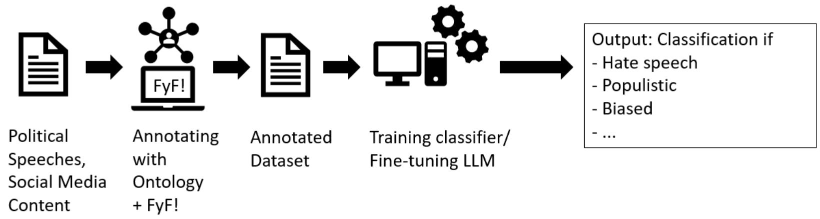

Figure 1: How the ontology helps with overcoming the lack of annotated datasets.

addition, it is possible to derive rules from the formal model that can be used for rule-based detection techniques or the generation of rhetorical figures.

Users of the ontology are people with linguistic or computational background, and people without any linguistic background. For those, we want to make the ontology accessible via a terminal-based tool called FyF! for determining figures (Section 4). Figure 1 illustrates the workflow of our vision of how the ontology can help annotate datasets, thus overcoming the problem of small datasets of rhetorical figures, and therefore enabling the training of classifiers or the fine-tuning of large language models (LLM). The process starts with textual data that could be political speeches or text from social media or messenger platforms. Together with the developed tool that is based on the ESTHER ontology, human annotators can reliably identify rhetorical figures in those texts, thus creating an annotated dataset. This dataset is then used to train a machine learning classifier or to train LLMs. With the help of those algorithms, hate speech, biased or populistic opinions or persuasive arguments can be better identified and revealed. This can be important in a context to warn users to read certain texts more attentively as it contains bias or persuasion methods. However, we want to highlight that the ontology is not only built to train a classifier. It also serves as a source of carefully hand-crafted knowledge about rhetorical figures and the first machine-readable format of them. Furthermore, the ontology reveals dependencies and similarities between different figures that can help in their future detection. For example, detection mechanisms for symploke can be used for anaphoras, as every symploke has also features of an anaphora.

We formulated **competency questions** as examples of what the ontology should be able to deliver in the end. Furthermore, we use them for consistency and sanity checks. For example, question three is expected to deliver the same results as question four as it is a negation of all other elements.
Q1: Which rhetorical figures have their defining element in a word?
Q2: In which rhetorical figures is a letter omitted?
Q3: What are figures that are neither figures of construction, nor figures of speech, nor figures of thought?
Q4: Which figures belong to the rhetorical group of tropes? *(Note: same output as Q3 expected)*
Q5: What is an example sentence for the figure alliteration?
Q6: What is the definition of the figure parallelism?
Q7: Which figures of speech or figures of thought have their defining element in the beginning?
Q8: Which figures repeat an element in the same form?

Q9: In which figures is a word in the same form repeated?
Q10: Which figures are the opposite of other figures?

3.4. Formalization of the Model

In ontology modeling, reusing parts from existing ontologies is considered good practice [20, 22]. Therefore, we align with the RetFig and GRhOOT ontologies to guarantee cross-language compatibility between Serbian, German, and English. Those ontologies suit our needs as they also model rhetorical figures. In the following, we describe the structure of the ESTHER ontology (Section 3.4.1) and the integration of hierarchical relations in Section 3.4.2.

3.4.1. Structure

We reuse the structure of the RetFig and GRhOOT ontology (which is actually based on RetFig). The linguistic and rhetorical entities are modeled as concepts/classes whereas the rhetorical figure itself is modeled as individual. We kept this architecture to guarantee interoperability between the English, Serbian, and German ontology. This means that applications like the FyF!-tool (see Section 4) can be easily adapted also to search the RetFig and GRhOOT ontologies. However, we are aware that there is more than one solution how to model such an ontology [24]. An ontology where the figures are modeled as classes and the example sentences as individuals would have also been possible, but we wanted to focus more on the aspect of interoperability.

We remove naming inconsistencies from RetFig and GRhOOT by strictly using lower CamelCase; a naming convention where words are joined together with a capitalized letter of each word, but the first letter is lowercase (e.g., isRhetoricalFigure, isOmitted).[6] As the terminology in the domain of rhetorical figures is quite diverse and some names originate from Greek, Latin, or are vernacular [43], we show other spellings, synonyms, and names in other languages such as German, Serbian, or Latin in the `seeAlso` property, also indicating the language. Fig. 2 shows an example of the entity graph in Protégé for the figure "isocolon", illustrating the connection between the classes and properties. The additional relations of the implied figure parallelism are left out for clarity.

Like in RetFig and GRhOOT, we model the textual definition of the figures in natural language as `rdfs:comment`, a property that provides a human-readable description of a resource [44]. Also, we use the self-defined data property called `isExample` with the RDF-datatype `string` to provide at least one example sentence for each figure. In addition to previous ontologies in this domain, we introduce hierarchical relations that are presented in the following Section 3.4.2.

3.4.2. Hierarchical Relations

Harris and DiMarco [1] mention "compound" figures. These are figures that are produced by the combination of other figures, e.g., a symploce (aka symploke) is the combination of an epanaphora (aka anaphora) and epistrophe (aka epiphora). In addition, hierarchical relations

[6] RetFig and GRhOOT use camel case (but switch between first letter lower- and uppercase) and hyphens to connect words.

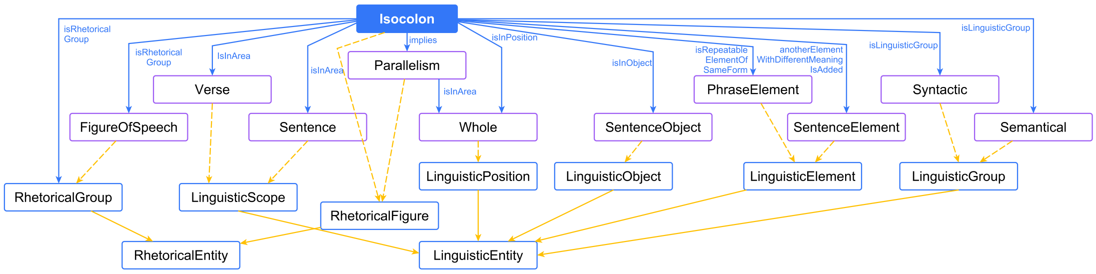

Figure 2: Entity graph in Protégé for the figure isocolon.

can exist such that some figures are a more specific form of other figures. Also, the combination of different figures yields the presence of other figures, as described by [15]. We tried to identify those relations and formally modeled them as the object property `implies`. The relation is similar to a subclass relation. However, as figures are modeled as individuals in both the Serbian RetFig and the German GRhOOT, we wanted to maintain the same structure to guarantee interoperability. That is the reason why the new property `implies` was introduced. The implication $\forall RF_A \sqsubseteq RF_B$ expresses that every figure RF_A also contains figure RF_B, as the figure RF_A includes figure RF_B. For example, each symploke is also an anaphora, while all figures with repetitions of the same form are a ploke (a figure of repetition without further constraints on the position):

$$\texttt{symploke} \sqsubseteq \texttt{anaphora} \quad \text{and} \quad \texttt{anaphora} \sqsubseteq \texttt{ploke}, \quad \text{respectively.}$$

Compound figures are also modeled with the `implies` relation. If a figure implies multiple other figures, it is a compound figure. For example, a symploke is the combination of anaphora and epiphora:

$$\texttt{symploke} \sqsubseteq \texttt{anaphora} \sqcup \texttt{epiphora}.$$

The figure climax is produced by a combination of following figures:

$$\texttt{climax} \sqsubseteq \texttt{gradatio} \sqcup \texttt{incrementum}\ [15, 45].$$

We also modeled the relation of opposition between figures with the symmetric property `isOppositeFigure`, e.g., anaphora `isOppositeFigure` epiphora, and vice versa respectively (repetition at the beginning of consecutive sentences vs. repetition at the end of consecutive sentences).

3.5. Evaluation and Validation

To evaluate ESTHER, we translate the ten competency questions from Section 3.3 into SPARQL queries and execute them on the ontology to see if it behaves as expected. As mentioned, we use some of the queries of the RetFig [8] and GRhOOT [9] ontology to make it comparable. The results differ only for figures that either have different properties in the English language, that do not exist, or are not yet modeled in ESTHER. Otherwise, the results fulfill our expectations.

For example, competency question Q4 delivers all figures that belong to the rhetorical group of tropes. Its translation to SPARQL is shown below.

```
cq4 = """
   SELECT DISTINCT ?Figure
   WHERE {
      ?Figure rdf:type esther:RhetoricalFigure ;
              esther:isRhetoricalGroup esther:Trope .
      }
   """
```

Unfortunately, the oQual validator could not be found online. OnToology is a holistic system, but its usage requires additional effort with steps that are prone to failure, e.g., granting access to the GitHub repository. The OOPS! scanner is easy to use as users can provide a link or directly input the ontology to a text field on the website. It categorizes the errors into three levels of severity. The validator found only two minor and one medium issue within our ontology. The medium issue is that OOPS! suggests some classes or properties to be declared disjoint. According to the explanation of Noy et al. [24], we do not consider this issue, as e.g., an operation for a rhetorical figure can be both adding and omitting an element. The consistency of our ontology was verified by the HermiT reasoner version 1.4.3 [46].

3.6. Documentation and Maintenance

The last step is maintaining the ontology, which includes changes and updates as ontology development is an "iterative process" [24]. We want to iteratively extend the ontology with further rhetorical figures, and add examples to the existing figures. For code management, we made the ontology available in a version control system [25], namely GitHub,[7] which allows tracking and documenting all changes. This repository contains the LODE documentation, the OWL file of the ontology, the competency questions and the code of the FyF!-tool.

4. FyF: Find your Figure!

Identifying rhetorical figures e.g., in social media posts or political discussions, can help determine the intended meaning or to spot hidden notions. The ESTHER ontology can be used to support human annotators in determining the right name of a rhetorical figure and revealing its functions based on its properties. For example, a human can easily spot intentional repetitions of words or more complex structures, e.g., the semantic repetition of a word in a different form. However, it is difficult to name those figures. We present the ongoing work on our terminal-based tool written in Python. The tool asks the users to enter the characteristics they recognize in a text in which they suspect rhetorical figures. A SPARQL query is then built automatically that queries the ESTHER ontology for figures that fulfill those properties. If users are unsure, they can enter a "?" as blank property. An exemplary procedure is shown in Fig. 3 in Appendix A. If there is a match, the names of the figures, their definitions, and examples are shown to the users. If no figure matches their description, the users have the possibility to enter

[7]https://github.com/kuehnram/ESTHER-Ontology

comma-separated keywords as free text. We set up a small experiment to prove that the tool facilitates the annotation process for persons without any linguistic background. Our hypothesis is that a person using this tool is more likely to find the correct name for the rhetorical figure. We presented five sentences to three persons without any linguistic background, but with a proficient level of English. One person is allowed to use the FyF!-tool, one person uses a regular search engine on the Internet, and the other one uses ChatGPT [47], a large language model that provides an answer in the form of a conversation. The following five sentences with figures from all four linguistic categories were used (figure of speech: Sentence 1 and 5, trope: Sentence 4, figure of thought: Sentence 3, figure of construction/scheme: Sentence 2).

1. When there is talk of hatred, let us stand up and talk against it. When there is talk of violence, let us stand up and talk against it.
2. ’xactly!
3. The dog is black, the cat is white.
4. Boom! Meow! Woof!
5. When I was a child, I liked to watch TV. When I was a child, my mother loved me. when I was a child, I liked to watch TV. When I went to school, my mother went to work.

Table 1
Comparison of the annotation results. Green/✓means correct, red/× indicates false, and orange/∼ is not entirely correct.

Sentence	Correct Figure	ChaptGPT	FyF!	Search Engine
1	Symploke	Anaphora ∼	Symploke ✓	Symploke ✓
2	Aphaeresis	Exclamatory Emphasis ×	Aphaeresis ✓	Aphaeresis ✓
3	Parallelism	Parallelism ✓	Irony (??) ×	Antithesis ∼
4	Onomatopoeia	Onomatopoeia ✓	Ellipsis ∼	Onomatopoeia ✓
5	Anaphora	Parallelism ∼	Anaphora ✓	Anaphora ✓

Table 1 shows the results of the experiment. The cells in red with a ×-symbol show where the answer was wrong. A green cell with a ✓shows a correct answer. Orange cells with ∼ show answers that are at least partly correct. For example, annotating a symploke with “anaphora” is partly right, as symploke is a compound figure consisting of both anaphora and epiphora. The experiment confirmed our hypothesis. It showed that ChatGPT is powerful and fast for annotation, but does not spot compound figures and figures that are more salient for humans (cf. Sentence 5, where the dominant figure is anaphora, and parallelism is not so salient). Using the tool is slower than ChatGPT, but more precise. Still, it is faster than using a search engine (around 13 min. vs. around 20 min.). Using the tool also helps with avoiding spelling mistakes or using different spellings (cf. Sentence 2).

We are aware of the **limitations** in the functionality of the terminal-based tool, as it is still ongoing work. From the remarks of the FyF!-user, we found out that it is more difficult to detect tropes. Furthermore, we received input on how to improve the tool in the future. Nevertheless, we have shown that the ontology is helpful for annotators to determine rhetorical figures without extensive training. Researchers will benefit from more annotated datasets as they can be used for the training of algorithms or to improve language models.

5. Conclusion

We modeled a formal domain ontology of the most common rhetorical figures in the English language. We described our methodology, which is based on other approaches. Within this process, we aligned with the existing Serbian RetFig and German GRhOOT ontologies for rhetorical figures. In addition to resolving inconsistencies, we added hierarchical relationships between figures and defined compound figures. The ontology was evaluated to ensure consistency and sanity. We provided an overview of available ontology validation and documentation tools and used them for our ontology. The usage scenarios of this ontology are manifold. It serves as a knowledge base but it can also assist humans without prior knowledge in the process of annotating rhetorical figures. We proved this by developing a terminal-based annotation tool that helped to find the right names for figures. Furthermore, rules can be derived from the ontology for either detecting or generating examples of rhetorical figures. This is important as large annotated datasets lack in this domain.

Future work will include the extension of the ontology by modeling additional figures. We want to extend the terminal-based tool and build a graphical user interface for it, improving the application's usability. We want to distribute it and gather as many annotated examples as possible. Detecting rhetorical figures and generating rules based on the ontology is also considered future work. With more annotated data, models can be trained, which will then increase the efficiency and precision of various NLP tasks.

Acknowledgments

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 author is responsible for the content of this publication.

SPONSORED BY THE

Federal Ministry of Education and Research

References

[1] R. A. Harris, C. Di Marco, Rhetorical figures, arguments, computation, Argument & Computation 8 (2017) 211–231.

[2] R. A. Harris, C. Di Marco, S. Ruan, C. O'Reilly, An annotation scheme for rhetorical figures, Argument & Computation 9 (2018) 155–175.

[3] H. L. Nguyen, T. D. Nguyen, D. Hwang, J. J. Jung, Kelabteam: A statistical approach on figurative language sentiment analysis in twitter, in: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), 2015, pp. 679–683.

[4] J. Mitrović, C. O'Reilly, M. Mladenović, S. Handschuh, Ontological representations of rhetorical figures for argument mining, Argument & Computation 8 (2017) 267–287.
[5] K. Hamilton, Towards an ontology for propaganda detection in news articles, in: R. Verborgh, A. Dimou, A. Hogan, C. d'Amato, I. Tiddi, A. Bröring, S. Maier, F. Ongenae, R. Tommasini, M. Alam (Eds.), The Semantic Web: ESWC 2021 Satellite Events, Springer International Publishing, Cham, 2021, pp. 230–241.
[6] J. Lemmens, I. Markov, W. Daelemans, 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, 2021, pp. 7–16.
[7] M. Alliheedi, C. Di Marco, Rhetorical figuration as a metric in text summarization, in: Advances in Artificial Intelligence: 27th Canadian Conference on Artificial Intelligence, Canadian AI 2014, Montréal, QC, Canada, May 6-9, 2014. Proceedings 27, Springer, 2014, pp. 13–22.
[8] M. Mladenović, J. Mitrović, Ontology of Rhetorical Figures for Serbian, in: M. Habernal (Ed.), Text, Speech, and Dialogue, Springer, Berlin, Heidelberg, 2013, pp. 386–393.
[9] R. Kühn, J. Mitrović, M. Granitzer, GRhOOT: Ontology of Rhetorical Figures in German, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 4001–4010. URL: https://aclanthology.org/2022.lrec-1.426.
[10] R. A. Harris, C. Di Marco, A. R. Mehlenbacher, R. Clapperton, I. Choi, I. Li, S. Ruan, C. O'Reilly, A cognitive ontology of rhetorical figures, Cognition and Ontologies (2017) 18–21.
[11] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, A. Reyes, Semeval-2015 task 11: Sentiment analysis of figurative language in Twitter, in: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), 2015, pp. 470–478.
[12] X. Hu, Natural language processing and ontology-enhanced biomedical literature mining for systems biology, in: Computational systems biology, Elsevier, 2006, pp. 39–56.
[13] R. Harris, C. DiMarco, Constructing a rhetorical figuration ontology, in: Persuasive Technology and Digital Behaviour Intervention Symposium, Citeseer, 2009, pp. 47–52.
[14] A. R. Kelly, N. A. Abbott, R. A. Harris, C. DiMarco, D. R. Cheriton, Toward an ontology of rhetorical figures, in: Proceedings of the 28th ACM International Conference on Design of Communication, 2010, pp. 123–130.
[15] C. O'Reilly, Y. Wang, K. Tu, S. Bott, P. Pacheco, T. W. Black, R. A. Harris, Arguments in Gradatio, Incrementum and Climax; a Climax Ontology, in: Proceedings of the18th workshop on Computational Models of Natural Argument. Academic Press, 2018.
[16] Y. Wang, R. A. Harris, D. M. Berry, An Ontology for Ploke: Rhetorical Figures of Lexical Repetitions., in: JOWO, 2021.
[17] Y. Wang., R. Kühn., R. A. Harris., J. Mitrović., M. Granitzer., Towards a Unified Multilingual Ontology for Rhetorical Figures, in: Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KEOD,, INSTICC, SciTePress, 2022, pp. 117–127. doi:10.5220/0011524400003335.
[18] R. Kühn, J. Mitrovic, Multilingual domain ontologies of rhetorical figures and their applications, 2023.
[19] M. Poveda-Villalón, A. Gómez-Pérez, M. C. Suárez-Figueroa, OOPS! (OntOlogy Pitfall

Scanner!): An On-line Tool for Ontology Evaluation, International Journal on Semantic Web and Information Systems (IJSWIS) 10 (2014) 7–34.
[20] D. Allemang, J. Hendler, Semantic web for the working ontologist: effective modeling in RDFS and OWL, Elsevier, 2011.
[21] V. Devedzić, Understanding ontological engineering, Communications of the ACM 45 (2002) 136–144.
[22] M. Fernández-López, A. Gómez-Pérez, N. Juristo, Methontology: from ontological art towards ontological engineering (1997).
[23] M. Grüninger, M. Fox, Methodology for the Design and Evaluation of Ontologies, Proc. IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing (1995).
[24] N. F. Noy, D. L. McGuinness, et al., Ontology development 101: A guide to creating your first ontology, 2001.
[25] A. Pease, Ontology: A practical guide, Articulate Software Press, 2011.
[26] M. Uschold, M. King, S. Moralee, Y. Zorgios, The enterprise ontology, The knowledge engineering review 13 (1998) 31–89.
[27] M. Hristozova, L. Sterling, 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), 2002.
[28] V. Presutti, E. Daga, A. Gangemi, E. Blomqvist, extreme design with content ontology design patterns, in: Proc. Workshop on Ontology Patterns, 2009, pp. 83–97.
[29] M. C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, The NeOn methodology for ontology engineering, in: Ontology engineering in a networked world, Springer, 2011, pp. 9–34.
[30] M. Poveda-Villalón, A. Fernández-Izquierdo, M. Fernández-López, R. García-Castro, Lot: An industrial oriented ontology engineering framework, Engineering Applications of Artificial Intelligence 111 (2022) 104755. URL: https://www.sciencedirect.com/science/article/pii/S0952197622000525. doi:https://doi.org/10.1016/j.engappai.2022.104755.
[31] A. Gangemi, C. Catenacci, M. Ciaramita, J. Lehmann, Modelling ontology evaluation and validation, in: European Semantic Web Conference, Springer, 2006, pp. 140–154.
[32] A. Alobaid, D. Garijo, M. Poveda-Villalón, I. Santana-Perez, A. Fernández-Izquierdo, O. Corcho, Automating ontology engineering support activities with OnToology, Journal of Web Semantics 57 (2019) 100472. URL: https://www.sciencedirect.com/science/article/pii/S1570826818300465. doi:https://doi.org/10.1016/j.websem.2018.09.003.
[33] S. Peroni, D. Shotton, F. Vitali, Tools for the automatic generation of ontology documentation: A task-based evaluation, International Journal on Semantic Web and Information Systems (IJSWIS) 9 (2013) 21–44.
[34] S. Peroni, D. M. Shotton, F. Vitali, Making Ontology Documentation with LODE., in: I-SEMANTICS (Posters & Demos), 2012, pp. 63–67.
[35] M. A. Musen, The protégé project: a look back and a look forward, AI Matters 1 (2015) 4–12. URL: https://doi.org/10.1145/2757001.2757003. doi:10.1145/2757001.2757003.
[36] D. Garijo, WIDOCO: a wizard for documenting ontologies, in: International Semantic Web Conference, Springer, Cham, 2017, pp. 94–102. URL: http://dgarijo.com/papers/widoco-iswc2017.pdf. doi:10.1007/978-3-319-68204-4_9.
[37] S. Lohmann, V. Link, E. Marbach, S. Negru, WebVOWL: Web-based visualization of

ontologies, in: International Conference on Knowledge Engineering and Knowledge Management, Springer, 2014, pp. 154–158.

[38] H. Lausberg, D. E. Orton, R. D. Anderson, Handbook of literary rhetoric: A foundation for literary study, Brill, 1998.

[39] B. McGuigan, Rhetorical devices: A handbook and activities for student writers, Prestwick House Inc, 2011.

[40] H. F. Plett, Literary rhetoric: concepts-structures-analyses, volume 2, Brill, 2010.

[41] G. O. Burton, Silva rhetoricae, Brigham Young University 14 (2007).

[42] H. F. Plett, Einführung in die rhetorische Textanalyse, Buske, 1991.

[43] H. F. Plett, Systematische Rhetorik: Konzepte und Analysen, volume 2127, Fink, 2000.

[44] W. W. W. Consortium, et al., RDF Schema 1.1 W3C Recommendation 25 February 2014 (2014).

[45] L. A. Black, K. Tu, C. O'Reilly, Y. Wang, P. Pacheco, R. A. Harris, An ontological approach to meaning making through PATH and gestalt foregrounding in climax, The American Journal of Semiotics 35 (2019) 217–249.

[46] R. Shearer, B. Motik, I. Horrocks, HermiT: A Highly-Efficient OWL Reasoner., in: Owled, volume 432, 2008, p. 91.

[47] OpenAI, ChatGPT: Optimizing Language Models for Dialogue, 2022. URL: https://openai.com/blog/chatgpt/.

[48] J. Fahnestock, Rhetorical figures in science, Oxford University Press on Demand, 2002.

A. Find your Figure User Interfaces

Fig. 3 shows how users can interact with the terminal-based tool to find the names of rhetorical figures, based on certain properties.

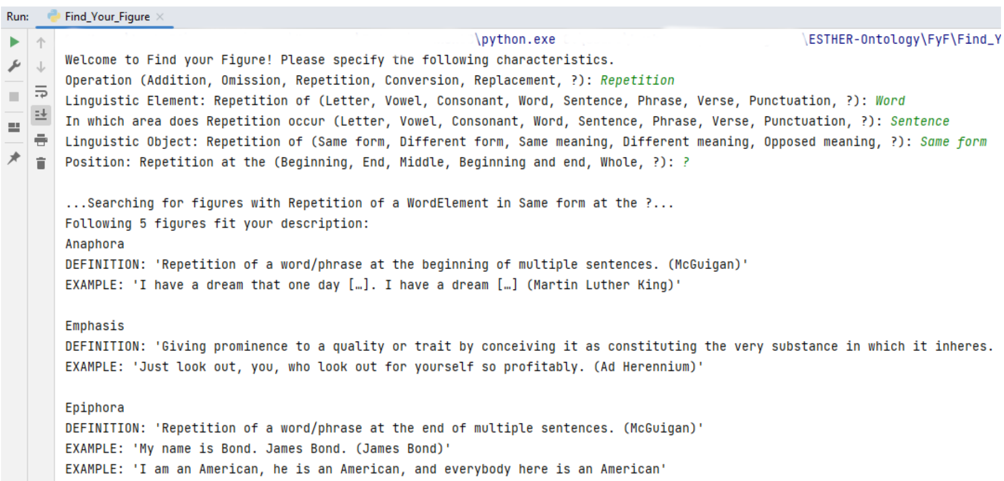

Figure 3: User specifies properties (at the end of the line in green).

B. Glossary

alliteration Repetition of same sound of nearby words/within a group of words. Example: “Mail. Message. More” (Advertisement). 6

anadiplosis Anadiplosis is the repetition of a word/phrase of the end of a sentence at the beginning of the next sentence. Example: “Strength through purity, purity through faith.” (Chancellor Adam Susan, V for Vendetta). 15

anaphora Phrase/word repetition at the beginning of multiple sentences [39]. Example: “I have a dream that one day [...]. I have a dream [...]” (Martin Luther King). 3, 6–8, 15

climax Generally, the arrangement of words, phrases, or clauses in an order of increasing importance, often in parallel structure.[41]. Example: “First we lost the village, then the city, then the world”. 3, 8

epiphora Repetition of a word/phrase at the end of multiple sentences [39]. Example: “My name is Bond. James Bond.” (James Bond). 7, 8, 15

gradatio A Gradatio is a series whose members overlap but need not possess the same quality[48]. Gradatio is a progressive anadiplosis [38]. Example: “He is my friend, my angel, my god” (Friedrich Schiller: Die Raeuber). 3, 8

incrementum A series of words in the same domain where subsequent words mark an increase on a semantic scale [15]. Example: “Thus a proud man is called Lucifer, a drunkard a swine, an angry man mad”. 3, 8

isocolon A series of similarly structured elements having the same length. A kind of parallelism [41]. Example: “No pain, no gain”. 8

metaphor A metaphor compares two things from a different domain. It can convey literal, emotional, or psychological truths [39]. Example: “Flood of refugees”, “Heart of gold”. 1

parallelism Parallelism uses the same structure for multiple parts of a sentence to link them. It makes a text easier to read and keeps long lists understandable. Often, parallelism refers to the syntax of the elements, but not necessarily [39]. Example: “In the name of the father, in the name of the son, and in the name of the holy ghost”. 6

ploke Perfect lexical repetition of a word/phrase. 3, 8

symploke Combination of anaphora and epiphora. Example: “All through industry, all for industry” (Claude-Henry des Saint-Simon). 6–8