Hidden in Plain Sight: Can German Wiktionary and Wordnets Facilitate the Detection of Antithesis?

Ramona Kühn
Innstraße 43
94032 Passau
Germany

Jelena Mitrović
Innstraße 43
94032 Passau
Germany

Michael Granitzer
Innstraße 43
94032 Passau
Germany

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

Abstract

Existing wordnets mainly focus on synonyms, while antonyms have often been neglected, especially in wordnets in languages other than English. In this paper, we show how regular expressions are used to generate an antonym resource for German by using Wiktionary as a source. This resource contains antonyms for 45499 words. The antonyms can be used to extend existing wordnets. We show that this is important by comparing our antonym resource to the antonyms in OdeNet, the only freely available German wordnet that contains antonyms for 3059 words. We demonstrate that antonyms are relevant for the detection of the rhetorical figure antithesis. This figure has been known to influence the audience by creating contradiction and using a parallel sentence structure combined with antonyms. We first detect parallelism with part-of-speech tags and then apply our rule-based antithesis detection algorithm to a dataset of the messenger service Telegram. We evaluate our approach and achieve a precision of 57 % and a recall of 45 % thus overcoming the existing approaches.

1 Introduction

The goal of Natural Language Processing (NLP) is to enable computers to automatically analyze text and to understand its (sometimes subtle) underlying meaning. While this is a relatively easy task for most humans, it is difficult for computers, as good results depend on available language resources. Examples of such resources are lexicons, dictionaries, or databases. One of the most well-known lexical-semantic databases, or knowledge bases, is the Princeton WordNet (Fellbaum, 2010) for the English language.

For text understanding of non-English texts, wordnets in other languages have been developed. However, two problems are often encountered. The first problem is common in NLP: Good resources are only available in "popular" languages with many speakers (e.g., English), while less- or low-resourced languages are often neglected. The second problem is that some wordnets are not freely accessible or have poor quality (Nhut Lam et al., 2022).

In this paper, we want to focus on antonyms in the German language: While synonyms are easier to obtain from existing wordnets, antonyms only get marginal attention. So far, freely available German antonyms can only be found in the open German OdeNet (Siegel and Bond, 2021) and BabelNet, a multilingual encyclopedic dictionary and semantic network that combines different sources like WordNet, Wikipedia, Wiktionary, etc. (Navigli and Ponzetto, 2012; Navigli et al., 2021).

We aim to use a different antonym resource for the task of antithesis detection – a rhetorical figure that is constructed by using parallel phrases that contain a pair of antonyms. As rhetorical figures always have a function (Givón, 1985), this figure is used to express tension, to make a comparison, or to reveal contradicting behavior, ideas, or statements. For example, the biblical antithesis **"The spirit is willing, but the flesh is weak"** (Matthew 26:41) expresses the contrast between mind and body. Green (2021) presents a study on how antitheses are used in arguments in environmental science policy journal articles to show the contrast between the view of environmentalists and engineers. In politics, its usage can influence the audience and increase the scepticism towards the authorities by showing that their actions are contradictory, raising a feeling that they are unable to govern people successfully. By detecting antitheses in politically related texts, people's opinions can be better evaluated by understanding the hidden notions expressed by this figure: By using it for contrasting comparisons ("they are allowed to do this" vs. "we are not allowed to do that"), people are made jealous and incited to riot. It is important to prevent this by understanding the role of antithe-

sis, especially in crisis situations, e.g., a pandemic, war, or in energy crises.

We consider antithesis detection as a binary classification problem and tackle it with a rule-based approach. As a dataset, we use posts by a German journalist on the messenger service Telegram in which he criticizes the German government during the COVID-19 pandemic. The parallel structure of those posts is identified by using part-of-speech (POS) tags. For the antonym detection, we rely on a word-based comparison. However, we realized that both OdeNet and Babelnet cannot satisfy our requirements for this task: We expect the antonym resource to be as complete as possible, i.e., containing antonyms for as many words as possible. Furthermore, we are interested in achieving a high recall to find most of the antitheses. In addition, we want the resource to be freely available. For Babelnet, it was difficult to retrieve all words with their respective antonyms, as we could not locate a list of all words contained in Babelnet. Furthermore, the extraction is limited by so-called Babelcoins where one coin represents one query: To retrieve one antonym, at least three different queries are necessary. In addition, by manually checking Babelnet, we realized that only few words contain antonyms which are often not German. OdeNet offers more antonyms and it provides a file of every word. With this file, we were able to extract all antonyms from OdeNet. However, we were still not able to identify most of the relevant antitheses (see Section 5 for a detailed comparison, especially Table 2). To meet all of those requirements, we built our own dictionary for German antonyms. With regular expressions, we extracted antonyms using the semi-structured data from the German Wiktionary.[1]

Our contributions are as follows:

- We show how Wiktionary can be used for creating a language resource for German, which can be used for the extension of existing wordnets.
- Our resource covers more German antonyms than OdeNet, resulting in a better performance in the task of antithesis detection.
- We present an algorithm based on the antonym dictionary that not only detects parallelism but also antitheses.
- We create the first annotated German dataset of antitheses.

The remainder of this paper is structured as follows: In Section 2, we describe related work on wordnets, Wiktionary as data source, and antithesis detection. Section 3 gives insights into the process of building the antonym dictionary. The antithesis detection process is described in Section 4. The evaluation in Section 5 compares our antonym resource with the antonyms in OdeNet. We also evaluate the performance of the rule-based antithesis detection approach. The following discussion includes a critical review of our methods and results. The paper concludes in Section 6. The code and data relevant for this paper are available online.[2]

2 Related Work

We will first give an overview of different wordnets for the German language and the sources that can be used to construct a language resource. The second part briefly defines antithesis and presents approaches for its detection.

2.1 Wordnets and Language Resources

Some advances have already been made to develop a German wordnet: GermaNet (Hamp and Feldweg, 1997) tries to be the German counterpart of WordNet. However, it is not freely available, and its functionality is limited. The Leipzig corpora (Biemann et al., 2007) and OpenThesaurus (Naber, 2005), a dictionary for German, contain only synonyms. BabelNet (Navigli and Ponzetto, 2010) includes some German antonyms but is far from being complete. Only OdeNet (Siegel and Bond, 2021), which is based on WordNet, contains more antonyms.

To extend the list of antonyms, we rely on Wiktionary as a source: Wikipedia and Wiktionary have already proved to be useful for different tasks, such as building an n-gram corpus (Cacho et al., 2021) or extending the GermaNet with definitions (Henrich et al., 2011). As Wiktionary consists of semi-structured text, effective scrapers and parsers are required. Scrapers or parsers for Wikipedia and Wiktionary are mostly used to create English resources: wiktextract (Ylonen, 2022) is a great tool to parse the English Wiktionary as it even includes

[1] https://de.wiktionary.org/wiki/Wiktionary:Hauptseite

[2] GitHub repository:
Wiktionary Parser: https://github.com/kuehnram/Wiktionary_Parser_German_Antonyms
Antithesis Detection: https://github.com/kuehnram/Antithesis_Detection

English dictionaries. Unfortunately, this resource does not work for other languages yet. For the German language, the wiktionary-de-parser[3] extracts different elements of a Wiktionary page, but it does not parse the text that contains the actual antonyms. This is actually the difficult part, as the content of a Wiktionary page is just "ordinary text" (Krizhanovsky, 2010). We developed a parser that can not only be used to extract antonyms but also to extract other elements of a Wiktionary page, e.g., synonyms, idiomatic expressions, examples, etc. in a similar way.

2.2 Antithesis

We want to show the relevance of antonyms for the detection of the rhetorical figure antithesis. Fahnestock (2002) describes this figure as "pleasing" and "persuasive" and defines it as "a verbal structure that places contrasted or opposed terms in parallel or balanced [...] phrases". Both a parallel structure and predictable antonyms are required, e.g., "the night is long, the day is short". A taxonomy of antithesis and examples from the environmental domain is presented by Green (2021).

Despite the relevance of rhetorical figures nowadays, e.g., in argument mining (Mitrović et al., 2017), there is no common definition, making it even more important to formalize their properties. Mladenović and Mitrović (2013) formally described rhetorical figures by creating RetFig, a formal domain ontology for almost 100 figures in the Serbian language. GRhOOT (Kühn et al., 2022) is the adaption and extension of the Serbian RetFig for the German language. In the GRhOOT ontology, an antithesis is described by the properties (in bold) that it is a **semantic figure of thought** which appears over a **whole sentence** and affects a **word** or a **phrase**. An element of the **opposite meaning** is **added**, expressing the use of an antonymous pair. However, this formal description cannot be used for detection yet.

A detection algorithm was developed by Lawrence et al. (2017) who split a text into "constitutive dialogue units" and "associated propositional units". They use the Princeton WordNet to find antonyms that appear in the other part of the unit. However, polarity shifters and negation cues are problematic.

Green and Crotts (2020) also try to detect antitheses: They use the antimetabole dataset of Dubremetz and Nivre (2018), as Harris et al. (2018) state that antitheses often occur with antimetabole - a repetition of words in reverse order. Green and Crotts found 120 antitheses in this dataset. For the detection, they rely on the algorithm by Lawrence et al. (2017) but use a broader definition of antonyms: They do not only look for antonyms in WordNet and ConceptNet (Speer et al., 2017) but also consider synonyms of the antonyms. They criticize that the publicly available resources are limited, as the used wordnets are not complete and contain "wrong" antonyms.

As the language resources prove to be insufficient for the English language, it is even more challenging in other languages: For German, we will show that our antonym dictionary outperforms the antonyms from OdeNet. We also advance the detection of antitheses by identifying not only antonyms but also parallelism.

3 Antonym Resource Creation

The data on Wiktionary pages are only semi-structured and the structure even differs between languages (Krizhanovsky, 2010). Typically, a German Wiktionary page of a certain word contains subsections describing its pronunciation, meanings, synonyms, antonyms, idiomatic expressions, etc.[4] Below the section of the German entry are properties often displayed in other languages, e.g., English, Swedish, etc.

The German Wiktionary dumps (here: from 2021-11-21 17:50:44) that are created regularly and are available online[5] are used to create the antonym resource. The structure of such a file is shown in Appendix A.1. We parse the XML dump file with ElementTree[6] from the Python Standard Library, and search for the string "Gegenwörter" (*antonyms*). Antonyms are extracted by using regular expressions. We ensure that only German antonyms are extracted and not words from other languages that are also present on some pages.

So far, we have identified five different variants of antonym representations on the Wiktionary page. There is no guarantee that this list is complete, and the different examples can also appear combined

[3] https://pypi.org/project/wiktionary-de-parser/

[4] Example for the word "gut" (*good*): https://de.wiktionary.org/wiki/gut

[5] https://dumps.wikimedia.org/dewiktionary/latest/

[6] https://docs.python.org/3/library/xml.etree.elementtree.html

on a single page:

1. :[1] [[antonym1]], [[antonym2]]/[[antonym3]]
2. :[1] [[antonym(plural s/n)]]
3. :[1] [[multi]] [[word]]
4. :[1] Explanatory text sometimes with [[link]]: [[antonym]]
5. :[1] [[Text: Antonym]] (e.g., [[*Substantive*: Antonym]]

Obviously, the first case is the easiest to extract, whereas the other cases induce more complexity to the parsing process. In the second case where the plural word is given within brackets, e.g., "Wolke(n)" *(cloud(s))*, we consider both the singular "Wolke" *(cloud)* and the plural "Wolken" *(clouds)* separately. If there are multiwords like in the third case, we concatenate them: e.g., [[darüber]] [[halten]] (literally: *over hold*) is concatenated to "darüber halten". The problem with the German language is that the position changes with inflexion, e.g., "ich halte darüber" (*I hold over*). Even lemmatization cannot resolve this conflict. In the fourth and fifth cases, additional free text is added as an explanation or further specification of the antonyms. In the fourth case, we are able to extract the antonym as it is normally provided within two square brackets. The fifth case, however, requires semantic understanding. Therefore, those few cases are ignored completely. We also ensure that the antonyms are implications: For example, the antonym of the German word "mother" is father, but "father" does not have any antonyms in the German Wiktionary. In our dictionary, the antonym relation is bidirectional: If x is an antonym of y, then y is also an antonym of x: A(x)=y → A(y)=x.

The approach of using regular expressions is simple yet effective: The final data structure is a Python key-value Dictionary consisting of 45,499 keys, where the key is the actual word/page title in lowercase, and the values are the set of antonyms, e.g., the antonyms of "woman" are "man, mister, exwife, husband", whereas for "freedom" it is "dependency, heteronomy":

```
{...
`frau`: {`mann`, `herr`, `exfrau`,
      `ehemann`},
`freiheit` :
      {`abhängigkeit`, `fremdbestimmung`}
...}
```

4 Antithesis Detection

To highlight the relevance of antonyms in NLP in general and especially in the context of rhetorical figure detection, we focus on the figure antithesis, a figure combining parallel phrases with antonyms. As definitions for rhetorical figures have never been precise or uniform, the detection of antitheses poses the following challenges:

1. How to define a relevant phrase?
2. How to define if phrases have a parallel structure?
3. How strict does the parallelism have to be to maintain the effect of an antithesis?
4. When is a word considered as an antonym of another word?

Those issues are tackled in the following way:

Challenge 1 - Phrases: We define relevant phrases by the occurrence of specific markers such as punctuation marks, the word "als" (*as/when*), or "und" (*and*). Considering quotation marks as such markers or removing them can also yield a parallel structure. A sentence is split at the occurrence of such markers into individual phrases. Only phrases that consist of more than one word are considered.

Challenge 2 & 3 - Parallelism: Parallel phrases do not have to be necessarily within one sentence, only within one post. We define parallelism by repeating POS tags, e.g.,

```
the/DET night/NOUN is/AUX long/ADV,
the/DET day/NOUN is/AUX short/ADV ./
    PUNCT
```

The spaCy POS tagger[7] also supports the German language. We use the trained pipeline `de_dep_news_trf` with the highest accuracy for POS tags (99%).[8] Despite the high accuracy, false labelled POS tags can occur, causing the algorithm not to recognize the parallel structure. Furthermore, we replace the POS tag "PROPN" for proper nouns with the tag "NOUN", as proper nouns are just a further specification of general nouns (e.g., (company) names, brands, etc.).

We do not use a strict definition of parallelism but accept some deviations: If a phrase consists of 3 or fewer words, perfect parallelism is required, i.e., perfectly repeating POS tags. If a phrase consists of

[7] https://spacy.io/models/de

[8] https://spacy.io/models/de#de_dep_news_trf

more than three words, we defined a Levenshtein distance: In our case, the number of POS tags between two phrases has to match at least to 75 %. To investigate parallelism further to find the optimal threshold for parallelism is considered future work.

Challenge 4 - Antonyms: With adequate language resources, finding antonyms should not be a challenge. However, as already mentioned, the functionality of existing resources is limited. With our generated dictionary from Wiktionary, we hope to cover most of the existing antonyms. Dependent on how strictly Fahnestock's definition is interpreted, it would be necessary to define for each antonym pair a distance function to determine the appropriateness of an antonym pair. Green and Crotts (2020) consider synonyms of antonyms in their antithesis detection. We will not include synonyms, not only because of the lack of resources but also because the function of the antithesis is weakened if the predictability of antonyms decreases. Another problem that both Green and Crotts (2020) and Lawrence et al. (2017) face are polarity shifters and negation cues: For example, "unethical" is considered to be the opposite of "ethical". However, "unethical" is semantically very close to "not ethical". As we are looking for antonyms on a word basis, we are not able to capture those negations. Another problem that was also identified by Green and Crotts (2020) is that opposing concepts cannot be recognized. A further challenge in the German language is the so-called tmesis with separable verbs: a particle is split from its core, or prefixes are inserted in the process of inflection, changing their position in a sentence. Even lemmatizers are not able to transform those words into their original lemma.

4.1 Dataset

We use two different sources for the dataset: First, we reuse the annotated antithesis dataset of Green and Crotts (2020). In this dataset, both parallelism and antonyms are loosely defined, as synonyms of antonyms are allowed.We translated it from English into German with Deepl[9] and manually checked the output. Some entries lost their parallel structure in the translation process, resulting in 106 out of 120 entries that can be considered as an antithesis in German.

As second source, we use 3433 posts from a German channel on the messenger service Telegram. The data was collected by Peter et al. (2022). We choose the channel of `reitschusterde`, which is operated by a German journalist criticizing the COVID-19 strategy of the German government. His posts are polarizing, so he is sometimes referred to as a right-wing populist (Bednarz, 2020). Populists are persons that "pit the pure, innocent, always hardworking people against a corrupt elite

[9] `deepl.com`

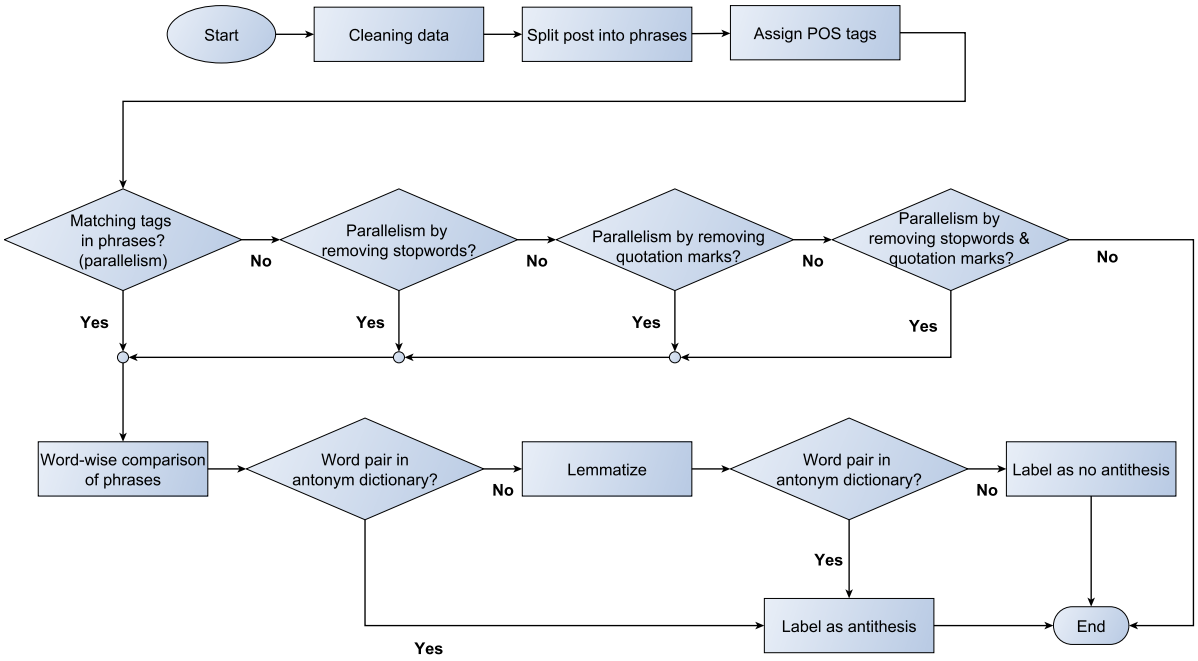


Figure 1: Steps of the parallelism and antonym detection algorithms.

who do not really work" (Müller, 2016), opposing normal people to the elites (Wodak, 2015). Simply said, they often display the world in binary schemes like "good" vs. "evil", "citizens" vs. "elites", which resembles the structure of antithesis. Furthermore, the political actions of the German government during the COVID-19 pandemic were actually contradictory, as even neutral newspapers reported (Suchy, 2021; Hierholzer, 2021; Gerd Antes, 2021). We therefore hope to find good examples of the figure antithesis in this data. In the following, we will refer to each post from Telegram and each instance of Green's dataset as "entry".

4.2 Annotation

It is widely known that annotation is a tedious task. Only one annotator was available who was introduced to the characteristics of detecting antithesis. To reduce the workload, the data was pre-filtered by only selecting entries where the parallelism algorithm based on POS tags could identify at least one pair of parallel phrases. This results in a reduction of the dataset. Overall, it consists of 954 distinct entries.

The structure of the overall annotated dataset is shown in Table 1. For readability, the entries are translated from German to English and the opposing words are highlighted in bold. For each entry, the parallelism algorithm first identifies all combinations of parallel phrases (column *Phrase 1/Phrase 2*. In the next step, the antithesis algorithm looks for opposing words in each phrase (column *Algorithm*; the steps of the algorithms are described in Subsection 4.3). The human annotator decides if the phrases are parallel and contain an antithesis (column *Human* is 1 if yes, 0 otherwise). There can be multiple parallel phrases for each post (cf. Table 1). This means that if a post contains an antithesis in general, it is possible that there may be parallel pairs of phrases that do not contain antonyms. This resulted in 1251 different annotated phrase pairs originating from the 954 initial entries.

4.3 Antithesis Detection Algorithm

Fig. 1 shows the flowchart of the parallelism and antithesis detection algorithms. After cleaning the data, the post is split at specific markers into phrases. POS tags are assigned to each word while repeating POS tags in two phrases mean parallelism. If no parallelism is detected, common German stopwords are removed. If there is still no parallelism, we first remove quotation marks, which leads to another split of the phrases, and then try removing stopwords and quotation marks.

If parallelism was detected, we search for opposing words in the two phrases with the help of the created antonym dictionary, which we described in Section 3. If no antonym pair is found, we try lemmatizing each word.

5 Evaluation

We evaluate two aspects: (1) We want to quantitatively compare the created antonym resource with OdeNet's antonyms, which is so far the best wordnet for German antonyms. (2) We want to give

Entry	Phrase 1	Phrase 2	Algorithm	Human
'Many media focus on escalation and [...]. The police are focusing on de-escalation."	['Many','media', 'focus', 'on', **'escalation'**]"	['the', 'police', 'are', 'focusing', 'on', **'de-escalation'**]	1	1
"Who is a fascist here? "Antifa old" against "Antifa new": [...]. His thesis: The counter-protest is controlled. A search."	['antifa', '**old**']	['antifa', '**new**']	1	1
"Who is a fascist here? "Antifa old" against "Antifa new": [...]. His thesis: The counter-protest is controlled. A search."	['his', 'thesis']	['a', 'search']	0	0

Table 1: Three example entries in the dataset.

	Precision	Recall	Accuracy	F1-Score
OdeNet Antonyms	50.00 %	8.80 %	90.00 %	14.97 %
Wiktionary Antonym Dict	57.00 %	45.24 %	91.05 %	50.44 %

Table 2: Performance metrics for rule-based antithesis detection.

insights into how good the rule-based approach for antithesis detection performs both with OdeNet and our created antonym resource.

5.1 Comparison of Antonym Resources

We compare the antonyms from OdeNet and our created antonym resource. From OdeNet's lexical entries[10] we extracted all antonyms to build a dictionary that has the same structure as our antonyms dictionary. On average, OdeNet has more antonyms per word (16.43 vs. 2.45 in our resource), but it has antonyms for only 3,059 words, whereas our antonym resource contains antonyms for 45,499 words.

Another feature of OdeNet is that it contains several multiwords, idiomatic expressions, or tmeses. Due to the specialty of the German language, the word order is changed by inflection. As we perform a word-by-word comparison in our case, those multiwords cannot be detected, as lemmatizers are not yet able to respect those constructions.

5.2 Evaluation of Antithesis Detection

We evaluate the performance of OdeNet and our antonym resource in the task of detecting the rhetorical figure antithesis: We apply both resources to our annotated dataset and compare the results with those of the human annotator. In this step, we compare phrase-wise.

In those 1251 phrase pairs, 126 antitheses were identified by the human annotator. We are aware that the dataset is highly imbalanced, which can lead to problems regarding the evaluation metrics. The imbalance is often inherent in datasets with rhetorical figures. Our work is a step towards the creation of more datasets and enlarging existing ones. The confusion matrix in Fig. 2a shows that 57 antitheses are correctly identified (`Predicted label=1` and `True label=1`). As the dataset is unbalanced, Subfig. 2b on the right shows the normalized confusion matrix.

As orientation for our evaluation serves the result of Green and Crotts (2020): They achieved a precision of 41.1 % and a recall of 38.4 %. However, their approach was different and is therefore difficult to compare. Moreover, their dataset consists solely of antithesis, so they focused on detecting only antonyms and not on identifying parallelism in addition. Table 2 shows the metrics of the antithesis detection with OdeNet and our antonym dict. OdeNet is only able to find 8.8 % of relevant antithesis. This was too low for our requirements. With our antonym resource from Wiktionary, we achieve a precision of 57 % and a recall of 45.24 %. However, the accuracy has to be taken with a grain of salt due to the imbalanced dataset.

There is no antithesis that OdeNet finds that our antonym resource did not find. This means a combination of both resources would not improve the results here, but can be useful for other datasets. We also took a closer look when the antonym dictionary fails (see Table 3): Most cases were no "typical" or "proper" antonyms (see Appendix A.2 for details).

Not in antonym resource	40
Wrong lemmatization	15
Negation	4
Opposed concepts/ideas	10

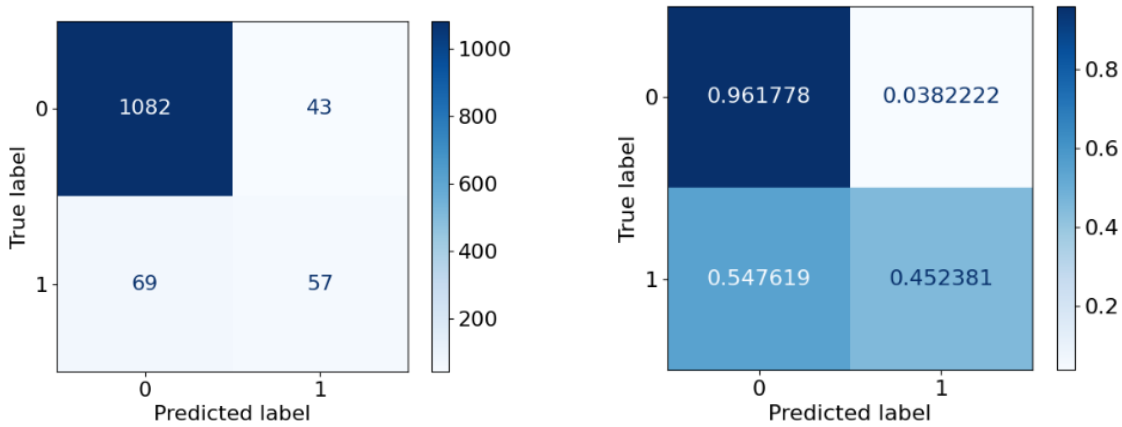
Table 3: Reasons for false negatives.

As already mentioned, an antithesis can evoke emotions of doubt by opposed comparison: We want to illustrate this by showing two examples the algorithm found. For readability, they are translated into English, and antonyms are in bold.

Example 1: **In former times**, the CSU (*German party*) used to stand for Bavarian lifestyle and culture. **Nowadays**, the CSU leader is destroying centuries-old traditions, [...].

Example 2: Good and bad demonstrators - well framed on ARD (*news channel*). Christopher Street Day **allowed** in Berlin, Corona Demo **banned** in Kassel.

[10] https://github.com/hdaSprachtechnologie/odenet



(a) Confusion Matrix.

(b) Normalized Confusion Matrix.

Figure 2: Confusion Matrices.

Example 3: Monday **morning**: Markus Söder (*German politician*) proposes to vaccinate politicians specifically with AstraZeneca's vaccine. Monday **afternoon**: Germany suspends Corona vaccinations with AstraZeneca as a precautionary measure on the recommendation of the Paul Ehrlich Institute.

5.3 Discussion

We want critically assess our approaches and discuss aspects that need improvement in the future. The detection of parallelism –another rhetorical figure– needs more attention in the future. The defined Levenshtein threshold needs more evaluation to discover at what distance the effect of parallelism is achieved. We also unveiled the property of lemmatizers that their precision increases with the number of words provided. As we performed lemmatization on single words, we obtained false lemmas of words, leading the algorithm to not find the antonym pair in the dictionary.

Although we cover more antonyms in our dictionary than OdeNet, we are only able to find opposing words, not separable words, multiword expressions, or contrasting concepts. Although OdeNet contains multiword expressions, it is not possible to reflect concepts based on a word-level comparison. This problem was already faced by Green and Crotts (2020) for English antonyms: "The current [...] resources [...] are incomplete in their coverage of opposite lexical concepts".

6 Conclusion

As wordnets mainly focus on synonyms, we constructed a resource for antonyms from the German Wiktionary. We highlighted the relevance of antonyms by using the created resource to detect the rhetorical figure antithesis, a persuasive figure that is often used in arguments.

Antithesis detection can enable the identification of bias and persuasion, which is helpful in a political context as our dataset demonstrated. With our rule-based approach, we were able to identify parallel phrases and achieved a recall of 45 %, whereas OdeNet was only able to identify 8.8 % of the antitheses. The limited availability of language resources, their functionality, and the need of datasets are still challenges that must be addressed.

In the future, we want to improve the detection of antithesis. With language models and deep learning, we assume to achieve higher precision and recall. Data augmentation techniques need to be considered to tackle the imbalance of the dataset. Wordnets can help here by replacing words with their synonyms.

Acknowledgment

The project on which this report is based was funded by the German Federal Ministry of Education and Research (BMBF) under the funding code 01lS20049. The author is responsible for the content of this publication.

We would like to thank Nancy L. Green very much for sharing her annotated dataset with us. We also thank the anonymous reviewers for their helpful comments.

References

Liane Bednarz. 2020. Lebensgefährliche "Lebensschützer". https://www.spiegel.de/politik/deutschland/christliche-corona-verharmloser-lebensgefaehrliche-lebensschuetzer-a-8c5ac68a-c030-414d-bb89-ed81cd992cf7. Accessed: 2022-07-29.

Chris Biemann, Gerhard Heyer, Uwe Quasthoff, and Matthias Richter. 2007. The leipzig corpora collection-monolingual corpora of standard size. *Proceedings of Corpus Linguistic*, 2007.

Jorge Ramón Fonseca Cacho, Ben Cisneros, and Kazem Taghva. 2021. Building a wikipedia n-gram corpus. In *Intelligent Systems and Applications: Proceedings of the 2020 Intelligent Systems Conference (IntelliSys) Volume 2*, pages 277–294. Springer.

Marie Dubremetz and Joakim Nivre. 2018. Rhetorical figure detection: chiasmus, epanaphora, epiphora. *Frontiers in Digital Humanities*, 5:10.

Jeanne Fahnestock. 2002. *Rhetorical figures in science*. Oxford University Press on Demand.

Christiane Fellbaum. 2010. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

Jürgen Zurheide Gerd Antes. 2021. Medizinstatistiker: Bevölkerung wird mit widersprüchlichen Aussagen konfrontiert. https://www.deutschlandfunk.de/corona-massnahmen-medizinstatistiker-bevoelkerung-wird-mit-100.html. Accessed: 2022-07-09.

Talmy Givón. 1985. Iconicity, isomorphism and non-arbitrary coding in syntax. *Iconicity in syntax*, pages 187–219.

Nancy L Green. 2021. Some argumentative uses of the rhetorical figure of antithesis in environmental science policy articles. *Computational Models of Natural*.

Nancy L Green and L Joshua Crotts. 2020. Towards automatic detection of antithesis. In *CMNA@ COMMA*, pages 69–73.

Birgit Hamp and Helmut Feldweg. 1997. Germanet-a lexical-semantic net for german. In *Automatic information extraction and building of lexical semantic resources for NLP applications*.

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.

Verena Henrich, Erhard Hinrichs, and Tatiana Vodolazova. 2011. Semi-automatic extension of germanet with sense definitions from wiktionary. In *Proceedings of the 5th Language and Technology Conference (LTC 2011)*, pages 126–130.

Michael Hierholzer. 2021. Ohne Kompass durch die Krise. https://www.faz.net/aktuell/rhein-main/kultur/widerspruechliche-corona-politik-ohne-kompass-durch-die-krise-17261870.html. Accessed: 2022-07-09.

AA Krizhanovsky. 2010. Transformation of Wiktionary entry structure into tables and relations in a relational database schema. *arXiv preprint arXiv:1011.1368*.

Ramona Kühn, Jelena Mitrovic, and Michael Granitzer. 2022. GRhOOT: Ontology of Rhetorical Figures in German. *LREC. Marseille, France*.

John Lawrence, Jacky Visser, and Chris Reed. 2017. Harnessing rhetorical figures for argument mining. *Argument & Computation*, 8(3):289–310.

Jelena Mitrović, Cliff O'Reilly, Miljana Mladenović, and Siegfried Handschuh. 2017. Ontological representations of rhetorical figures for argument mining. *Argument & Computation*, 8(3):267–287.

Miljana Mladenović and Jelena Mitrović. 2013. Ontology of rhetorical figures for serbian. In *Text, Speech, and Dialogue: 16th International Conference, TSD 2013, Pilsen, Czech Republic, September 1-5, 2013. Proceedings 16*, pages 386–393. Springer.

Jan-Werner Müller. 2016. *What is populism?* University of Pennsylvania Press.

Daniel Naber. 2005. OpenThesaurus: ein offenes deutsches Wortnetz. *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen: Beiträge zur GLDV-Tagung, Bonn, Germany*, pages 422–433.

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of babelnet: A survey. In *IJCAI*, pages 4559–4567.

Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial intelligence*, 193:217–250.

Khang Nhut Lam, Feras Al Tarouti, and Jugal Kalita. 2022. Automatically constructing Wordnet synsets. *arXiv e-prints*, page arXiv:2208.03870.

Valentin Peter, Ramona Kühn, Jelena Mitrović, Michael Granitzer, and Hannah Schmid-Petri. 2022. Network analysis of german covid-19 related discussions on telegram. In *Natural Language Processing and Information Systems: 27th International Conference on Applications of Natural Language to Information Systems, NLDB 2022, Valencia, Spain, June 15–17, 2022, Proceedings*, pages 25–32. Springer.

Melanie Siegel and Francis Bond. 2021. OdeNet: Compiling a GermanWordNet from other Resources. In *Proceedings of the 11th Global Wordnet Conference*, pages 192–198.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Clara Suchy. 2021. Herr Spahn, was ist Ihr Plan? https://www.n-tv.de/politik/politik_kommentare/Herr-Spahn-was-ist-Ihr-Plan-article22896673.html. Accessed: 2022-07-09.

Ruth Wodak. 2015. The politics of fear: What right-wing populist discourses mean. *The Politics of Fear*, pages 1–256.

Tatu Ylonen. 2022. Wiktextract: Wiktionary as machine-readable structured data. *LREC. Marseille, France*.

A Appendix

A.1 Structure of Wiktionary

Fig. 3 shows the structure (xml tags) of the dump file of the German Wiktionary. We are interested in the "'text" part, as it contains both the visible text of the Wiktionary website and the antonyms. Unfortunately, it is a semi-structured text string, making it more difficult to parse.

For example, for the word "gut" (*"good"*), an excerpt of the text tag's content is shown in Fig. 4.

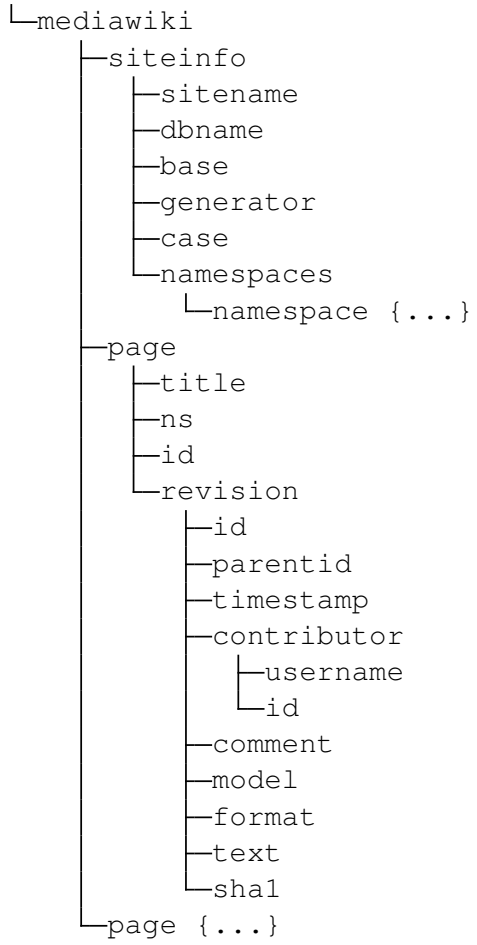


Figure 3: XML structure of German Wiktionary.

A.2 Details on False Negatives

We want to show some example sentence where the antithesis detection algorithm fails.

Not in antonym resource: Most cases fail because their antonym pairs are not in the antonym dictionary. Surprisingly, the antonym pair "mehr – weniger" ("*more – less*") is not in the dictionary. Another example from the dataset is "**Föderalimus** ade - **Zentralstaat** hurra" (*"**federalism** goodbye - **centralstate** hooray"*. The antonym of federalism is centralism in the antonym resource, but not centralstate. Another example is "**Aufregung** über Rassismus, **Wegsehen** bei Islamismus" (*"**agitation** over racism, **look away** from islamism"*): Agitation is contrasting to look away but more in a transferred sense. A further example uses numbers as antonym pairs to express the contrast that left-wing demonstations are more dangerous than Covid-19 demonstrations that are often considered to be led by right-wing activists: "**43** Verletzte bei der linksextremen demo, **7** bei der Corona Demo" (*"**43** injured at the extreme left demo, **7** at the Corona demo"*).

Wrong lemmatization: In the sentence "**Deeskalierende** Polizei, **eskalierende** Presse. ("***de-escalating** police, **escalating** press."*, the words are not correctly lemmatized by spaCy, leading to the non-detection of the pair escalate – de-escalate. In Table 1, the words are in their basic form and are therefore correctly detected.

Negation: "Erfolg **ist nicht der Schlüssel** zum Glück, Glücklichsein **ist der Schlüssel** zum Erfolg." ("*Success is **not the key** to happiness, happiness **is the key** to success.*"): The negation cannot be detected yet by the algorithm.

Opposed concepts and ideas: The following sentence shows the contrast between restrictions in two states. It is however not expressed by an antonym pair, therefore the algorithm fails to detect this antithesis. Please note that "north" and "south" is not a German antonym pair as

they are no German words: “In South Dakota gab es **fast keine Einschränkungen**, Schulen und Restaurants blieben offen. North Dakota setzte auf **Maskenpflicht und Restriktionen**.” (*South Dakota had **almost no restrictions**, schools and restaurants remained open. North Dakota relied on **mandatory masks and restrictions**.*)

```
{{Siehe auch|[[Gut]], [[güt]]}}
{{Wort der Woche|30|2008}}
== gut ({{Sprache|Deutsch}}) ==
=== {{Wortart|Adjektiv|Deutsch}} ===

{{Deutsch Adjektiv Übersicht
|Positiv=gut
|Komparativ=besser
|Superlativ=besten
}}

{{Worttrennung}}
:gut, {{Komp.}} bes·ser, {{Sup.}} am bes·ten

{{Aussprache}}
:{{IPA}} {{Lautschrift|gu:t}}
:{{Hörbeispiele}} {{Audio|De-gut.ogg}}, {{Audio|De-gut2.ogg}}, {{Audio|De-at-gut.ogg|spr=at}}
:{{Reime}} {{Reim|u:t|Deutsch}}

{{Bedeutungen}}
:[1] vom Menschen her [[positiv]] [[bewerten|bewertet]], [[empfinden|empfunden]], [[fühlen|gefühlt]] und dergleichen
::[a] ''prädikativ oder attributiv gebraucht''
::[b] ''adverbiell gebraucht''
:[2] eine Schulnote
::[a] ''(Deutschland und Österreich<ref>In Österreich großgeschrieben: „Mit „Gut“ sind Leistungen zu beurteilen, mit denen der Schüler …“
([https://www.ris.bka.gv.at/Dokumente/Bundesnormen/NOR12119641/NOR12119641.pdf Leistungsbeurteilungsverordnung, § 14. (3))]</ref>)''
entspricht der Note 2
::[b] ''(Schweiz)'' entspricht der Note 5
:[3] mit Zahl- oder Maßangaben: reichlich bemessen, etwas mehr als angegeben
:[4] jemandem [[freundlich]] gesinnt, jemandem zugetan
:[5] für besonders [[feierlich]]e Anlässe gedacht
:[6] ohne größere Mühen zu erledigen, leicht machbar

{{Herkunft}}
:[[mittelhochdeutsch]] und [[althochdeutsch]] ''guot'', ursprünglich „[[passend]]“<ref>{{Ref-Duden|gut}}</ref><ref>{{Ref-DWDS|gut}}</ref>

{{Synonyme}}
:[1] [[schön]], [[fein]], [[O. K.]], [[okay]]
:[1b] [[wohl]]
:[2] (österreichisch) [[Gut]]

{{Gegenwörter}}
:[1] [[schlecht]], [[negativ]], [[böse]], [[übel]]
:[3] [[knapp]]

{{Beispiele}}
:[1] Es geht mir ''gut.''
```

Figure 4: Semi-structured content of the Wiktionary text-tag. The heading {{Gegenwörter}} contains the antonyms.