Using Pre-Trained Language Models in an End-to-End Pipeline for Antithesis Detection

Ramona Kühn*, Khouloud Saadi*, Jelena Mitrović*†, Michael Granitzer*
*University of Passau, † Institute for Artificial Intelligence Research and Development of Serbia
Innstraße 43 94032 Passau, Germany; Fruškogorska 1 21000 Novi Sad, Serbia
{ramona.kuehn, khouloud.saadi, jelena.mitrovic, michael.granitzer}@uni-passau.de

Abstract

Rhetorical figures play an important role in influencing readers and listeners. Some of these word constructs that deviate from the usual language structure are known to be persuasive – antithesis is one of them. This figure combines parallel phrases with opposite ideas or words to highlight a contradiction. By identifying this figure, persuasive actors can be better identified. For this task, we create an annotated German dataset for antithesis detection. The dataset consists of posts from a Telegram channel criticizing the COVID-19 politics in Germany. Furthermore, we propose a three-block pipeline approach to detect the figure antithesis using large language models. Our pipeline splits the text into phrases, identifies phrases with a syntactically parallel structure, and detects if these parallel phrase pairs present opposing ideas by fine-tuning the German ELECTRA model, a state-of-the-art deep learning model for the German language. Furthermore, we compare the results with multilingual BERT and German BERT. Our novel approach outperforms the state-of-the-art methods (F1-score of 50.43 %) for antithesis detection by achieving an F1-score of 65.11 %.

Keywords: Language Models, ELECTRA, BERT, Antithesis, Rhetorical Figures, German, Persuasion

1. Introduction

Computational treatment and detection of rhetorical figures are tremendously promising but remain an undeveloped area (Lawrence et al., 2017). Rhetorical figures, e.g., metaphor, irony, or alliteration are a "departure from the normal usage" of language (Fahnestock, 2002). For example, metaphor is based on bridging concepts from two domains, e.g., "he has a heart of gold" or "she has eagle eyes." However, the actual meaning is implicit and can only be understood with sufficient background or context knowledge. Due to this implicit nature, efficient, automatic identification of rhetorical figures can improve text-processing results. Including features of rhetorical figures enhances the performance of several natural language processing (NLP) tasks. For example, the performance of hate speech detection models is improved when features of metaphors (Lemmens et al., 2021) or irony and sarcasm (Frenda et al., 2023) are included. Considering properties of figures of repetition improves text summarization (Alliheedi and Di Marco, 2014). To show how rhetorical figures influence arguments in the field of argument mining, Mitrović et al. (2017) included characteristics of metaphors and figures of repetition. Investigating the occurrence of rhetorical figures that are known to be polarizing or persuasive, e.g., the figure antithesis could greatly improve bias detection, propaganda detection, and even fake news detection.

Several challenges exist in the automatic detection of rhetorical figures. We argue that the following five obstacles should be addressed: (1) Although rhetorical figures have been a subject of research for hundreds of years from a linguistic perspective, their definitions are often inconsistent (Gavidia et al., 2022). Formalizing rhetorical figures is one approach to overcome these inconsistencies. However, recent advances in building such formal ontologies of rhetorical figures (Mladenović and Mitrović, 2013; Harris et al., 2017; Wang et al., 2021; Kühn et al., 2022; Wang. et al., 2022) still leave room for interpretation regarding the definitions. (2) Some rhetorical figures are rarely used, and it is difficult to find examples of them (Dubremetz and Nivre, 2015). (3) Existing research often focuses on the detection, rarely considering the cognitive effects of rhetorical figures (Mitrovic et al., 2020). Therefore, the potential of a computational analysis of rhetorical figures as a function of a deeper text understanding often remains unrecognized. (4) Only few annotated datasets of certain figures exist, and most of them only contain examples in the English language. This leads to the next obstacle. (5) As most of the language tools are limited to the English language, the detection of rhetorical figures in other languages is more challenging.

In this paper, we examine the figure antithesis that is known for its persuasive effects.[1] Fahnestock (2002) dedicates a complete chapter of her book "Rhetorical Figures in Science" to the figure an-

[1] Code is available here: `https://github.com/kuehnram/Pretrained_LM_Antithesis_Detection`

LREC-COLING 2024, pages 17310–17320
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

tithesis. The figure is described as “pleasing” and “persuasive” and defined as a verbal structure that places contrasted or opposed terms in parallel or balanced phrases. Therefore, both the opposed terms, as well as the parallel structure, are necessary for the creation of an antithesis. Furthermore, not only opposed terms are allowed, but also negating terms, e.g., “have” vs. “have not”. However, other definitions state that contrasting ideas (and not only contrasting words) also form an antithesis, where a parallel structure is not mandatory (McGuigan, 2011). In this paper, our approach will be guided by the following definition of an antithesis: An antithesis requires both syntactic **parallelism** (which does not have to be perfectly strict, i.e., not perfectly repeating part-of-speech (POS) tags) and **opposing words or ideas** which are used to express polarity, tension, or to emphasize contradictory facts.

An example of an antithesis is the following:

(1) [DET The] [ADJ stronger] [V leads],
[DET the] [ADJ weaker] [V follows].

We have a syntactic parallelism in this example as the POS tags show. The highlighted words form in fact, a double antithesis by using two opposing word pairs {stronger, weaker} and {leads, follows}. An extensive use of antithesis points out that the writer tries to convince the readers, often inciting them by showing contrasting sides. Obviously, the writer favors one side of the antithesis over another (Fahnestock, 2002). The usage of antitheses can also reveal persuasive actors. Persuasion per se is neither good nor bad, but readers should be made aware of it to not fall for fake news or populist ideologies.

Only few approaches exist that try to computationally detect antitheses (Lawrence et al., 2017; Green and Crotts, 2020; Kühn et al., 2023). They rely on rule-based algorithms that are not able to capture semantic relations, opposed concepts, ideas, or negation. In addition, these approaches focus on the English language and cannot achieve high metrics regarding precision and recall with the existing resources. Furthermore, they do not mention if the split into phrases is done manually or computationally. We want to fill this gap in the field of rhetorical figure detection. Our detection is based on a dataset from the messenger service Telegram[2] and translated examples from Green and Crotts (2020). We are able to automatically extract parallel phrases by relying on repeating POS tags. We then fine-tune the German ELECTRA (GELECTRA) model (Chan et al., 2020) on our dataset to detect if the pair of phrases contains a contradiction or not. In Section 5.3., we show that fine-tuning GELECTRA outperforms both multilingual BERT (mBERT) (Devlin et al., 2018) and the original deepset German BERT (GBERT) language models (Chan and Pietsch, 2019).

In this work, we are focusing on the German language as most research in the field of NLP is only directed towards the English language. However, it is “undesirable” that language technologies are only developed for one or two popular languages (Bender, 2019). Rogers and Augenstein (2020) criticize that reviewers often consider work in other languages than English as “narrow”. Furthermore, they state that models in other languages are equally generalizable as those in English. There is even the assumption that English models are less generalizable to certain domains, e.g., because it is not a grammatically gendered language (Ramesh et al., 2023). Søgaard (2022) highlights that the focus on English models in NLP in general and in conferences creates inequalities. He suggests banning English language models completely for one year to reduce this imbalance. This step seems to be too drastic, but we want to contribute to reducing this imbalance and therefore focus on the German language. Also, our end-to-end pipeline is generalizable and applicable to other languages. Antithesis is a figure that can be easily translated into other languages if the parallel structure can be maintained (Kühn et al., 2023).

The contributions of this paper are listed in the following:

- We propose a new approach for the antithesis detection task composed of three main blocks: text splitting, syntactic parallelism detection, and sentence-based contradiction detection.
- We propose a dataset of annotated German antitheses.
- As our detection algorithm focuses more on semantics, we are able to capture both contrasting ideas and negation, besides contrasting words.
- We test our proposed approach with various experimental setups (e.g., augmentation techniques) and different language models (GELECTRA, GBERT, mBERT) for comparison.
- To the best of our knowledge, our work is the first work that introduces an end-to-end pipeline with language models for antithesis detection.

2. Related Work

Research of rhetorical figures mainly focuses on the figure metaphor. Newer approaches fine-tune

[2] https://telegram.org/

pre-trained language models (Liu et al., 2020; Choi et al., 2021) for its detection. Pre-trained language models also proved to be successful for the detection of other figures, e.g., irony (Jiang et al., 2021; Zhang and Abdul-Mageed, 2019). The detection of less popular figures like chiasmus often relies on rule-based approaches or classical machine learning techniques due to the lack of training data for deep learning methods (Dubremetz and Nivre, 2017). Regarding the figure antithesis in English texts, Lawrence et al. (2017) first split a text into "constitutive dialogue units" and "associated propositional units". After removing common English stopwords, they used the Princeton WordNet (Fellbaum, 2010) to find antonyms that appear in the other part of the unit.

Another approach from Green and Crotts (2020) used the annotated antimetabole (rhetorical figure with a repetition of words in reverse order) corpus by Dubremetz and Nivre (2018) and extracted 120 antitheses out of it. For the detection, the authors relied on the algorithm of Lawrence et al. (2017). However, the relation of antonyms is defined more loosely, as also synonyms of antonyms are considered. Both of those approaches are not able to detect negative constructions (e.g., ask – ask not). To the best of our knowledge, there has been only one approach for antithesis detection in German (Kühn et al., 2023), which is rule-based. The authors used the translated dataset from Green and Crotts (2020), and a dataset from Telegram of a COVID-19 skeptic (Peter et al., 2022). They first extracted parallel phrases based on POS tags and then searched for a pair of antonyms in the parallel phrases. However, with this word-level comparison, they could not detect negations, opposing ideas, or differentiate between homonyms (same words with different senses).

To the best of our knowledge, no deep-learning-based approach for antithesis detection in any language has been proposed so far.

Another line of research close to antithesis detection is contradiction detection. It plays an important role in our end-to-end pipeline for antithesis detection in the German text. Most of the available annotated datasets for the contradiction detection task are in English: the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015), the MultiNLI dataset (Williams et al., 2017), and the XNLI dataset (Conneau et al., 2018). Few works addressed this topic in other languages, e.g., German, Spanish, and Persian. A portion of the SNLI dataset was machine-translated to the German language (Sifa et al., 2019). The authors confirmed that the model built using the German version of the SNLI dataset and the model built using the English SNLI have similar results. Contradiction detection mainly uses either rule-based (Asmi and Ishaya, 2012; Pham et al., 2013) or machine learning-based approaches (Sifa et al., 2019). Contradiction detection in Spanish texts was also performed with a Spanish BERT model (Sepúlveda-Torres, 2021). Wangs (2016) used an LSTM architecture for the contradiction classification task. For contradiction detection in German, Sifa et al. (2019) used an RNN model.

Our antithesis detection pipeline can split text automatically into relevant phrases and identify both opposing words and ideas in those phrases. In Section 5.3., we show that our pipeline where the GELECTRA is employed outperforms its counterparts where the GBERT and the mBERT are alternatively used.

3. Dataset and Annotation

We are interested in the persuasive use of antithesis in the German language.

We use a dataset created by Peter et al. (2022) who collected posts from German COVID-19 skeptics on the messenger platform Telegram. The focus lies on the channel of Boris Reitschuster, a German journalist who became popular in the community of COVID-19 skeptics by criticizing and questioning the measures of the German government during the pandemic. He gained many followers and is considered to be a populist (Bednarz, 2020). Populists depict the world as a construct where hardworking people suffer from a lazy elite (Wodak, 2015; Müller, 2016). They often use contrasting schemes such as "good" vs. "evil", or "citizens" vs. "elites" that resemble the contrasting concepts of an antithesis. We assume that if it is possible to reliably identify the figure antithesis in a text, more information about the implicit message of that text can be obtained, e.g., if the author is a populist or persuasive. We are looking into the presence of antitheses in the context of persuasive argumentation, where the channel owner tries to convince readers that the German Corona strategy is contradictory, and where he tries to diminish the trust in politicians in general.

As antithesis is not a common rhetorical figure that is often used, we also include translated examples from the English antithesis dataset of Green and Crotts (2020) to decrease the imbalance of positive (post containing an antithesis) and negative (post without an antithesis) classes. To reduce the load on the annotators, we decided to prefilter the data by only extracting posts that contain parallel phrases (see Section 4.1.). We believe that detecting syntactic parallelism is suitable for a rule-based approach. Only for the detection of contrasting words, concepts, or ideas, a deep learning approach is more likely to deliver satisfactory results.

Two human annotators annotate the dataset. One is a linguistic expert and the other is a student without deep linguistic expertise. We prepared a codebook with annotation guidelines and explanations of the figure antithesis that instructs the two annotators. The annotators label each post with either `Antithesis` (1/positive class) or `No Antithesis` (0/negative class). For example, the post

(2) Die schwächeren Dinge werden
The weaker things are
wiederholt, die stärkeren Dinge werden
repeated, the stronger things are
unterdrückt, [...].
suppressed, [...].
The weaker things are repeated, the stronger things are suppressed, [...].

is split into the parallel phrases "The weaker things are repeated" and "the stronger things are suppressed" and is annotated to be an `Antithesis=1`. A counterexample is

(3) Man kennt sich, man braucht
You know each other, you need
sich, [...].
each other, [...].
You know each other, you need each other, [...].

which actually has a parallel structure but no opposed words or ideas, therefore is labeled as `Antithesis=0`.

We calculate the inter-annotator agreement with Cohen's Kappa on the dataset. In the first annotation round, we only achieve an agreement of $\kappa = 0.43$. After a discussion between the two annotators, we discovered that the student did not consider phrases parallel if another non-parallel sentence was in between. Furthermore, double antitheses and antitheses based on negation (e.g., do vs. do not) led to deviating labels. After a second round of annotation, we achieve an agreement of $\kappa = 0.71$. For the sentences where the annotators cannot agree, we select the label of the linguistic expert as gold label.

Out of the 1248 dataset entries, only 126 contain an antithesis (`Antithesis=1`). We present augmentation techniques in Section 5.1. to handle this imbalance. Data for antithesis is scarce as it is the case for most rhetorical figures other than metaphor. However, we take a first step to overcome this limitation with our dataset.

4. Proposed Approach for Antithesis Detection

In this section, we introduce our end-to-end pipeline for antithesis detection. The pipeline is composed of three blocks as illustrated in Figure 1. As input example in the first block, we use the German version of the well-known antithesis

(4) Der Geist ist willig, das Fleisch ist
The spirit is willing, the flesh is
schwach
weak.
The spirit is willing, the flesh is weak.

The first block shows how to split the text into phrases based on POS tags. The second block illustrates how to define the parallel structure, which we will describe in more detail in the following Section 4.1. In the third block, we check if there is a contradiction between the parallel sentences by fine-tuning the pre-trained GELECTRA model on our pre-processed dataset. Details on the model are described in Section 4.2. We built our pipeline based on our definition from the Introduction in Section 1. that requires both syntactic parallelism and opposing words or ideas for antithesis.

4.1. Phrase Splitting and Parallelism Detection

After data cleaning, i.e., removal of hashtags, URLs, links, and smileys, we split each post into phrases at the occurrence punctuation marks and the words "und" (*and*) and "als" (*as/when*), as they are often used as subordinate conjunctions in the German language. We use the trained pipeline `de_dep_news_trf` of spaCy to assign POS tags to each word in the phrase, as this pipeline promises the highest accuracy for POS tags (99 %).[3] SpaCy is also available in other language and as our approach is language agnostic, only this pipeline has to be replaced for other languages. One specificity of spaCy is that proper nouns are a subclass of nouns but are labeled differently by spaCy, so we replace all proper noun tags (`PROPN`) with noun tags (`NOUN`) to make no differentiation.

If there are repeating POS tags in two phrases, they are labeled as syntactically parallel. In most cases, we do not demand strict parallelism, it depends on the length of the phrase. First, we only consider phrases that contain more than two words. For phrases with three words, perfectly repeating POS tags are required. For longer phrases, we use a Levenshtein distance such that the POS tags must match at least 75 % to be considered parallel. Based on a short manual investigation, this threshold seemed to be the best, but it needs further evaluation in the future. If no parallelism can be achieved, stopwords are removed and the process starts anew. If parallelism cannot be achieved, we change the split point by removing quotation marks. If there is still no parallelism, both stopwords and quotation marks are removed.

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

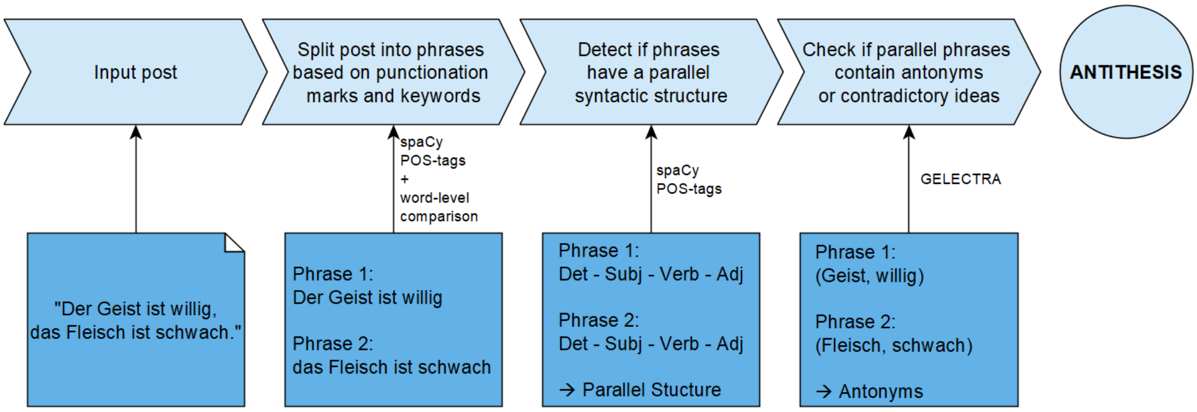
Input post
Split post into phrases based on punctionation marks and keywords
Detect if phrases have a parallel syntactic structure
Check if parallel phrases contain antonyms or contradictory ideas
ANTITHESIS
spaCy POS-tags + word-level comparison
spaCy POS-tags
GELECTRA
"Der Geist ist willig, das Fleisch ist schwach."
Phrase 1: Der Geist ist willig
Phrase 2: das Fleisch ist schwach
Phrase 1: Det - Subj - Verb - Adj
Phrase 2: Det - Subj - Verb - Adj
→ Parallel Stucture
Phrase 1: (Geist, willig)
Phrase 2: (Fleisch, schwach)
→ Antonyms

Figure 1: Antithesis Detection Scheme.

In this way, we can achieve that our dataset consists only of posts with a parallel sentence structure and are, therefore, candidates to contain an antithesis.

4.2. Antithesis detection

Transfer learning with pre-trained language models has been a successful technique to deal with various NLP tasks (Han et al., 2021). Fine-tuning these big models on downstream tasks when the available task-specific data is limited is always accurate. It is also performing well compared to training these huge architectures, e.g., BERT, from scratch on a small dataset. In this work, we adopt the same technique and we insist on the usage of pre-trained language models for antithesis detection.

The ELECTRA language model (Clark et al., 2020) has the same architecture as BERT (Devlin et al., 2018) but is pre-trained differently. While BERT is pre-trained on Masked Language Modeling (MLM) and Next Sentence Prediction, ELECTRA is pre-trained on Replaced Token Detection. The pre-training method of ELECTRA is more efficient because, unlike BERT, the model learns from all the input tokens (Clark et al., 2020). Besides the original English version of ELECTRA (Clark et al., 2020), there exists the deepset German ELECTRA (GELECTRA) (Chan et al., 2020) which is pre-trained on 163.5 GB of German text from the German OSCAR corpus, Wikipedia dumps of German, OPUS corpus, and Open Legal Data.

Recently, fine-tuning pre-trained language models on downstream tasks, such as contradiction detection, has achieved good results (Reimers and Gurevych, 2019; Clark et al., 2020). Chan et al. (2020) showed that GELECTRA outperformed both the GBERT and mBERT models on multiple benchmark tasks like the GLUE natural language understanding benchmark. For our antithesis detection pipeline, we fine-tune $GELECTRA_{base}$ with 786 hidden units and 12 attention heads. We found out that GELECTRA outperforms both Google's mBERT and the original deepset GBERT on our task, which we show in Section 5.3.. We also explored both the bi-encoder and the cross-encoder architectures of the BERT and the ELECTRA models. The cross-encoder architecture is more accurate in the case of natural language inference (NLI) classification (Reimers and Gurevych, 2019). The running time for the cross-encoder is also much lower compared to the bi-encoder version.

A classification layer is added on top of $GELECTRA_{base}$ to accommodate our task. The model takes as input a pair of phrases separated by a comma and outputs a binary label i.e., `Antithesis` or `No Antithesis`. During the training, we minimize the binary cross entropy objective function $L(\theta)$.

$$L(\theta) = -1/N(\sum_{i=1}^{N} y_i log(p_i) + (1 - y_i) log(1 - p_i))$$

where x_i is an example in the data D, y_i is the label, and p_i is the predicted probability that the sample x_i is of class c_i presented by y_i.

5. Experimental Setup and Results

This section presents the experimental setup, including different augmentation techniques we applied to improve the results. In Section 5.2., we present the achieved accuracy, recall, precision, and F1-score. A comparison of different language models on the same task is presented in Section 5.3..

5.1. Experimental Setup

The input data for each parallel structure in this stage consists of the tuple

$(phrase_1, phrase_2, label)$ where the label determines whether the structure is an `Antithesis` or `No Antithesis`. The proposed approach is tested using different experimental setups. Similar to the work of Devlin et al. (2018); Reimers and Gurevych (2019); Caselli et al. (2020), the $GELECTRA_{base}$ model is fine-tuned with an Adam optimizer, and the learning rate, the number of epochs, and the batch size are set to 0.00001, 4, and 16 respectively; 5-fold cross-validation is used to report the performance metrics for all experiments.

To handle the imbalance problem of our dataset, the data is first randomly split into train (0.8 × total examples) and test (0.2 × total examples), then we apply the following techniques to the training set:

- Using weighted cross entropy loss (WCEL) during the model training. The weight of each class is computed as following:

  $$W_i = N/(CN_i)$$

  Where N is the total number of samples in the entire dataset. C is the number of classes in the dataset, in our case 2. N_i is the number of samples per class i.

- Applying the following augmentation techniques (AUG):
  - AUG1: Synonym replacement, for which we use bert-base-multilingual-uncased (Devlin et al., 2018) to replace a subset of words by their synonyms.
  - AUG2: Back translation, where the original text is translated to Arabic and English, then back-translated to German. In this work, for machine translation from German to English and the inverse, we use the Facebook WMT19 models (Ng et al., 2019). For machine translation from German to Arabic and the inverse, we use opus-mt from Helsinki-NLP (Tiedemann and Thottingal, 2020).

It is important to mention that, in this work, we did not investigate which models are the best to perform the augmentation techniques. As we have a non-balanced test set, we report the accuracy, recall, precision, and binary F1-score.

For more clarity, the confusion matrices of the different variants are shown in Table 1. The recall is informative in our case as it reflects how well the model detects the positive class `Antithesis`. The performance of the classifier on the positive class is improving as we are applying different techniques to handle the data imbalance problem. In the beginning, the classifier identified only 66.32 % of the Antithesis examples with the $GELECTRA_{base}$ baseline setup fine-tuned on our proposed pre-processed dataset. Using WCEL, we achieve an improvement of 16.84 % in correctly detecting the Antithesis class compared to the baseline setup. Applying AUG1 and AUG2 combined helped to improve the results by 10.52 % compared to the baseline.

		Predicted	
		Antith.	No Antith.
Actual	Antith.	66.32	33.68
	No Antith.	4.16	95.84

$GELECTRA_{base}$

		Predicted	
		Antith.	No Antith.
Actual	Antith.	83.16	16.84
	No Antith.	6.06	93.94

$GELECTRA_{base}$ + WCEL

		Predicted	
		Antith.	No Antith.
Actual	Antith.	80.00	20.00
	No Antith.	8.31	91.69

$GELECTRA_{base}$ + AUG1 + WCEL

		Predicted	
		Antith.	No Antith.
Actual	Antith.	81.05	18.95
	No Antith.	6.06	93.94

$GELECTRA_{base}$ + AUG2 + WCEL

		Predicted	
		Antith.	No Antith.
Actual	Antith.	76.84	23.16
	No Antith.	5.28	94.72

$GELECTRA_{base}$ + AUG1 + AUG2

Table 1: Confusion Matrices (all values in %).

5.2. Results and Discussions

In Table 2, we note that the variant where we use the baseline setup $GELECTRA_{base}$ with WCEL achieves the best result in terms of a combination of recall and precision, which results in an F1-score of 65.11 %. In fact, 83.16 % of the antitheses examples are correctly identified by our model. Our approach outperforms the state-of-the-art rule-based approaches in antithesis detection by Green and Crotts (2020) and Kühn et al. (2023).

By adding AUG1 (synonym replacement) or AUG2 (back translation) on top of the setup of $GELECTRA_{base}$ + WCEL, the performance of the model in terms of accuracy, recall, and precision decreases. The results show also that in our case back translation as a data augmentation technique is more efficient than synonym replacement as it resulted in a better performance in terms of the three metrics. When we combine the two augmentation techniques and add them on top of the baseline, we obtain a better performance in terms of an F1-score of 63.78 % compared to the baseline where the F1-score is 61.28 %. The loss in F1-score when adding AUG1 or AUG2 on top of the setup $GELECTRA_{base}$ + WCEL also proves the limitations of the traditional data augmentation techniques in NLP, where one word could completely change the semantic meaning of a whole sentence. Thus, substituting words in the case of synonym replacement may result in a completely different sentence meaning. The same conclusion is applied to the back translation technique where the meaning could change after translating the sentence to another language.

As shown in Table 2, the standard mean error is not marginal. This can be explained by having a small number of positive examples i.e., `Antithesis` in the test set. Thus, even if there is a difference of one example between two different seeds, it results in a huge variation in the recall and precision metrics. For example, in the experiment where we combine both AUG techniques, we have in the first seed 14 true positives (TP) and 5 false negatives (FN), whereas in the second seed we have 15 TP and 4 FN, which results in a standard mean error of 2.64 % in terms of recall.

We also examined the Telegram posts in the dataset that GELECTRA did not classify correctly. In total, fifteen posts were misclassified, three posts were predicted not to be an antithesis, and twelve posts were predicted to be an antithesis when in fact they are not. We found that in the three posts where the model missed the antitheses, the parallelism was not obvious. Furthermore, those posts contain abstract contradictory ideas. One example is *"In the past, the party stood for the Bavarian way of life. Now they are destroying centuries-old traditions."*. The twelve posts that were misclassified as antitheses all contain an obvious parallel structure and/or repeating elements, but semantically they are not contradictory, e.g. *"You know each other, you need each other."*, or *"Life will always be strange, life doesn't have to be convincing."*.

5.3. Comparison of GELECTRA, mBERT, and GBERT

In this section, we replace the GELECTRA model by the GBERT and the mBERT language models in our end-to-end pipeline. In fact, instead of fine-tuning the GELECTRA model for the contradiction detection sub-task, we fine-tune the GBERT model or the mBERT model. We evaluate our pipeline using the three language models GBERT, mBERT, and GELECTRA and compare them with the best setup found in the previous experiments, i.e., using WCEL, as it achieves the highest F1-score. The accuracy, recall, precision, and F1-score metrics are averaged among 5 seeds.

Table 3 shows that accuracy, recall, precision, and F1-score of GELECTRA outperforms both GBERT and mBERT on the antithesis detection task, although GELECTRA has only 110 million parameters compared to the 172 million parameters of mBERT, as shown in column **Params**. GELECTRA identifies correctly 83.16 % of the Antithesis examples. It also has the best accuracy and precision values with 93.12 % and 54.02 %, respectively. mBERT achieves around 3.15 %, 0.64 %, and 2.25 % improvement in recall, accuracy, and precision, respectively compared to the original deepset GBERT.

In future work, we will focus on further exploring the GELECTRA pre-trained language model to tackle new research directions in the German language, e.g., Natural Language Inference and Rhetorical Figure Detection.

6. Limitations

We are aware of points of criticism in our approach. As antitheses are scarce, our dataset is not big enough to train a language model. This problem is inherent to datasets of rhetorical figures other than the popular figure metaphor. Due to this limitation, we focused on fine-tuning pre-trained language models. We made the first important steps towards antithesis detection with deep learning models, and we showed the effectiveness of our approach through the conducted experiments. We want to encourage other researchers both to investigate less known rhetorical figures and their effects, and to try the models on domains where only a few annotated datasets exist. For example, antithesis is co-located with the figure parallelism. Identifying parallelism also requires more atten-

Experimental setup	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
$GELECTRA_{base}$	93.60±0.66	66.32±4.27	57.56±4.21	61.28±3.46
$GELECTRA_{base}$ + WCEL	93.12±0.74	83.16±1.05	54.02±3.48	**65.11±2.26**
$GELECTRA_{base}$ + AUG1 + WCEL	90.80±0.55	80.00±1.05	44.51±1.71	57.08±1.25
$GELECTRA_{base}$ + AUG2 + WCEL	93.12±0.38	81.05±1.28	52.68±2.02	63.76±1.54
$GELECTRA_{base}$ + AUG1 + AUG2	93.36±0.27	76.84±2.10	54.69±1.58	63.78±1.03
Rule-based (Green and Crotts, 2020)	–	38.4	41.1	39.70
Rule-Based (Kühn et al., 2023)	91.05	45.24	57.00	50.43

Table 2: Evaluation of our antithesis detection approach on our test data. Each setup is trained with 5 random seeds. We report the averaged accuracy, recall, and precision with the standard mean error. WCEL = weighted cross entropy loss.

Model	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)	Params (M)
$GBERT_{base}$	90.48±0.95	67.37±3.06	43.25±2.66	52.68±1.63	110
$mBERT_{base}$	91.12±0.93	70.52±3.15	45.25±3.50	55.13±3.03	172
$GELECTRA_{base}$	**93.12±0.74**	**83.16±1.05**	**54.02±3.48**	**65.11±2.26**	**110**

Table 3: Comparison between $GBERT_{base}$, $mBERT_{base}$, and $GELECTRA_{base}$. M refers to millions.

tion in the future. For our purposes, we consider phrases to be parallel if they match at least 75 %. This appeared to us as a reasonable threshold after a manual investigation. However, more experiments are necessary on this topic. One could criticize that we could have used further datasets of COVID-19 skeptics that are available from Telegram (Peter et al., 2022). However, this task included a lot of manual effort for the annotation and those resources are limited. In the future, we hope that antithesis detection becomes a more popular research area with an increased number of annotated datasets in different languages. We do not consider our focus on the German language a limitation, as we clearly stated in Section 1. that our approach is generalizable for many other languages.

7. Conclusion and Future Work

In the presented work, we have made an important step towards deeper text understanding and persuasive text identification. To the best of our knowledge, this paper is the first end-to-end pipeline that uses pre-trained language models for antithesis detection. We focused on the German language, but our approach is also generalizable and applicable to other languages. We evaluated the performance of German ELECTRA and conducted further experiments with different augmentation techniques. The results prove the effectiveness of our approach as 83.16 % of the antithesis examples are correctly detected, thus outperforming existing antithesis detection approaches that are rule-based. In the future, we want to enlarge the existing dataset. With more annotated data, we will be able to retrain our model, increasing further its performance. Moreover, future work includes evaluating the proposed approach on other languages, such as English or French. In addition, we think that the study of the figure antithesis and its role in convincing the audience could also be interesting for other research disciplines, such as sociology or politics. Furthermore, we will continue to investigate other rhetorical figures, for example parallelism, which is part of every antithesis.

Ethics Statement

The Telegram data we use come from publicly accessible channels. We do not access any personal data. With our antithesis detection pipeline, it is possible to identify how many antitheses are used in a text. Although the antithesis is considered a persuasive rhetorical figure, it does not necessarily imply that someone using antitheses is a populist or tries to be persuasive. One should be aware that labeling texts as persuasive without further manual checking might damage the reputation of the authors of a text.

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 01|S20049. The author is responsible for the content of this publication.

SPONSORED BY THE
Federal Ministry
of Education
and Research

8. Bibliographical References

References

Mohammed Alliheedi and Chrysanne Di Marco. 2014. 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*, pages 13–22. Springer.

Amna Asmi and Tanko Ishaya. 2012. Negation identification and calculation in sentiment analysis. In *The second international conference on advances in information mining and management*, pages 1–7.

Liane Bednarz. 2020. Lebensgefährliche ”Lebensschützer”. https://tinyurl.com/4sv8743x. Accessed: 2023-10-01.

Emily Bender. 2019. The #benderrule: On naming the languages we study and why it matters. *The Gradient*, 14.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2020. Hatebert: Retraining bert for abusive language detection in english. *arXiv preprint arXiv:2010.12472*.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German’s next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Möller Chan and Soni Pietsch. 2019. bert-base-german-cased. https://www.deepset.ai/german-bert.

Minjin Choi, Sunkyung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. 2021. Melbert: metaphor detection via contextualized late interaction using metaphorical identification theories. *arXiv preprint arXiv:2104.13615*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

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. pages 231–243. Springer.

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.

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*.

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

Wenjuan Han, Bo Pang, and Yingnian Wu. 2021. Robust transfer learning with pretrained language models through adapters. *arXiv preprint arXiv:2108.02340*.

Randy Allen Harris, Chrysanne Di Marco, Ashley Rose Mehlenbacher, Robert Clapperton, Insun Choi, Isabel Li, Sebastian Ruan, and Cliff O’Reilly. 2017. A cognitive ontology of rhetorical figures. *Cognition and Ontologies*, pages 18–21.

Shengyi Jiang, Chuwei Chen, Nankai Lin, Zhuolin Chen, and Jinyi Chen. 2021. Irony detection in the portuguese language using bert. In *IberLEF@ SEPLN*, pages 891–897.

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

Ramona Kühn, Jelena Mitrovic, and Michael Granitzer. 2023. Hidden in plain sight: Can german wiktionary and wordnets facilitate the detection of antithesis? *Global Wordnet Conference*.

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

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.

Jerry Liu, Nathan O'Hara, Alexander Rubin, Rachel Draelos, and Cynthia Rudin. 2020. Metaphor detection using contextual word embeddings from transformers. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 250–255.

Brendan McGuigan. 2011. *Rhetorical devices: A handbook and activities for student writers*. Prestwick House Inc.

Jelena Mitrovic, Cliff O'Reilly, Randy Allen Harris, and Michael Granitzer. 2020. Cognitive modeling in computational rhetoric: Litotes, containment and the unexcluded middle. In *ICAART (2)*, pages 806–813.

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. pages 386–393, Berlin, Heidelberg. Springer.

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

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Minh Quang Nhat Pham, Minh Le Nguyen, and Akira Shimazu. 2013. Using shallow semantic parsing and relation extraction for finding contradiction in text.

Krithika Ramesh, Sunayana Sitaram, and Monojit Choudhury. 2023. Fairness in language models beyond english: Gaps and challenges. *arXiv preprint arXiv:2302.12578*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Anna Rogers and Isabelle Augenstein. 2020. What can we do to improve peer review in nlp? *arXiv preprint arXiv:2010.03863*.

Robiert Sepúlveda-Torres. 2021. Automatic contradiction detection in spanish.

Rafet Sifa, Maren Pielka, Rajkumar Ramamurthy, Anna Ladi, Lars Hillebrand, and Christian Bauckhage. 2019. Towards contradiction detection in german: a translation-driven approach. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2497–2505. IEEE.

Anders Søgaard. 2022. Should we ban english nlp for a year? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5260.

Jörg Tiedemann and Santhosh Thottingal. 2020. Opus-mt–building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. European Association for Machine Translation.

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

Yetian Wang., Ramona Kühn., Randy Harris., Jelena Mitrović., and Michael Granitzer. 2022. 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,*, pages 117–127. INSTICC, SciTePress.

Jiangj Wangs. 2016. Learning natural language inference with LSTM. *Proceedings of the Human Language Technologies: The 2016 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Ruth Wodak. 2015. *The politics of fear: What right-wing populist discourses mean*. Sage.

Chiyu Zhang and Muhammad Abdul-Mageed. 2019. Multi-task bidirectional transformer representations for irony detection. *arXiv preprint arXiv:1909.03526*.

9. Language Resource References

Peter, Valentin and Kühn, Ramona and Mitrović, Jelena and Granitzer, Michael and Schmid-Petri, Hannah. 2022. *Network Analysis of German COVID-19 Related Discussions on Telegram*. Springer.