

Lehrstuhl für Data Science

Application of comprehensive and responsible Machine Learning methods to the detection of chiasmi's rhetorical salience

Masterarbeit von

Yohan Meyer

1. PRÜFER

2. PRÜFER

Dr. Jelena Mitrović Prof. Dr. Michael Granitzer

January 19, 2023

Contents

1	Introduction	1
1.1	Context	1
1.2	Definition of the Problem	1
1.2.1	Specifications	1
1.2.2	Research Questions	3
1.3	Contributions Preview	3
1.4	Structure of the Thesis	4
2	Background	6
2.1	Terminology	6
2.1.1	Chiasmus	6
2.1.2	Antimetabole	9
2.1.3	Rhetorical effect	12
2.2	State of the Art	14
2.2.1	Rhetorical Figures	14
2.2.2	Chiasmus and Antimetabole Detection	17
2.2.3	Stanza	30
3	Methods	32
3.1	Tools and Project Management	32
3.2	Automatic Extraction of Chiasmi	33
3.2.1	Use Case	33
3.2.2	Choices	35
3.2.3	Chronological Overview of the Implementation	37
3.2.4	Limitations	40
3.3	Data	41
3.3.1	Retrieval	41
3.3.2	Annotation	42

Contents

3.4	Detection of Antimetabole’s Saliency	43
3.4.1	Baselines	43
3.4.2	Models	44
3.4.3	Features	45
4	Results	47
4.1	Extraction Pipeline Evaluation	47
4.2	Models Evaluation and Comparison	48
4.2.1	General Presentation	48
4.2.2	Baselines	49
4.2.3	Novel Features	50
4.2.4	Different Types of Models	52
5	Discussion	57
5.1	Interpretation of the Baselines Results	57
5.1.1	The <i>Dubremetz</i> Baseline	57
5.1.2	The <i>Schneider</i> Baseline	59
5.1.3	All model types	60
5.2	Analysis of the Novel Features and Models	61
5.2.1	Parison	62
5.2.2	Isocolon	64
5.2.3	Nominal groups	67
5.2.4	Repetitions	69
5.2.5	Regression Tree Visualization	71
6	Conclusion and Future Work	74
6.1	Conclusion	74
6.2	Insights	76
6.2.1	Research Work	76
6.2.2	Application	77
6.2.3	Personal Assessment	78
6.3	Future Work	78
	Appendix A Glossary	80

Appendix B Topic for Master Thesis	82
B.1 Data Augmentation and Machine Learning for Rhetorical Figures	82
B.1.1 References	82
Appendix C Figures	84
C.1 Extraction pipeline	84
C.2 Dubremetz and Nivre (2015)	85
C.3 Dubremetz and Nivre (2016)	86
C.4 Dubremetz and Nivre (2017)	86
C.5 Schneider et al. (2021)	87
C.6 Dubremetz and Nivre (2018)	88
C.7 Partial Dependency Plots	90
C.7.1 Parison	90
C.7.2 Isocolon	93
C.8 Tree Visualization	97
Appendix D Code	100
D.1 Initialising a Stanza pipeline	100
Bibliography	101
Eidesstattliche Erklärung	107

Abstract

Rhetorical figures are an essential but well concealed part of our language. They subtly influence our perception and comprehension of speech, often working together to create appealing rhetorical effects. Due to their nature, they are difficult to detect for a human being and even more so for a computer. This work focuses on the automatic detection of two related figures: *chiasmi* and *antimetabole*. They are both defined by a inverse repetition, like in the famous phrase “All for one, one for all”, which makes them *schemes* (figures that rely on the structure of a text). The major hurdles for detecting salient chiasmi are its rarity in the English language, the abundance of uninteresting inverse repetitions, and the lack of research coupled with the lack of annotated data for it. The contributions of this thesis are threefold: (1) a novel and complete pipeline to extract chiasmi candidates from raw text based on the inverse repetitions of lemmas and of word embeddings, up to their annotation; (2) a manually retrieved and annotated dataset more than ten times the size of currently available ones, leaving us with 585 annotated antimetabole and almost one hundred remaining (which could not be extracted by our pipeline); (3) eventually, a comprehensive study of various Machine Learning models for detecting salient antimetabole, built upon baselines from the state-of-the-art and augmented with novel features, which manage to outperform the state-of-the-art.

Acknowledgments

First of all, I would like to thank my thesis advisors who would deserve their names cited on the title page: Jelena Mitrović and Ramona Kühn from the University of Passau and Diana Nurbakova from the INSA Lyon. Without your experience, your help and your precious insights, this thesis would certainly not be what it is (and very probably not be at all). Thank you for taking the time to attend our meetings (special gratitude for Ramona who managed to attend *every single one*), to answer our emails, and simply for bearing with me. I would also be very dishonest if I did not mention my cherished *project partner*, Guillaume Berthomet, who ~~suffered~~ worked side by side (sometimes literally) with me for the greatest part of this project. I would also like to thank Randy Allen Harris from the University of Waterloo, who, despite our timezone differences, took the time to meet with us and offer us his kind help.

I extend my gratitude to the University of Passau and its Faculty of Computer Science for welcoming me and making this double degree possible; the Computer Science Department of the INSA Lyon as well, for allowing me to benefit from this partnership. At the same time, this academic exchange would not have been possible without the help from the ERASMUS program and from the Franco-German University.

My most sincere thanks go to all the persons I did not nominally cite, of the administration especially, whose discrete yet essential help is rarely enough appreciated.

On a more personal note, my warmest thanks go to my PhD-Track roommates during my time in Passau, who made my life there brighter every day; to my dearest friends back in France who supported me nonetheless, Lucie and Adria, and all members of the most excellent association *l'ALIR*; and to Maëlle, who went so far as to come visit me in Passau. I wonder every day what I have done to deserve such greatness.

Of course, I finally need to thank my family, for their moral and financial support, who kept hearing me complain without fail, and who always bear with me, even at my worst. Last, but certainly not least, I thank you, my reader. Thank you for taking the time to read what I wrote, and may this thesis be up to your expectations.

List of Figures

3.1	Entire chiasmi candidates extraction pipeline, beginning with a raw text file (gray module).	33
5.1	Partial Dependency Plot for all parison features as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	64
5.2	Partial Dependency Plot for all isocolon features as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	66
5.3	Partial Dependency Plot for the <i>nominalGroups</i> feature as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	68
5.4	Partial Dependency Plot for the <i>repetitionsFront</i> feature as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	69
5.5	Partial Dependency Plot for the <i>repetitionsBack</i> feature as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	70
C.1	An example of Doccano’s graphical user interface for annotation.	84
C.2	An extract of an annotated XML file generated by the extraction pipeline.	85
C.3	Average precision, and precision at a given top rank, for each experiment, taken from Dubremetz and Nivre (2015).	85
C.4	Average precision for chiasmus detection (test set), taken from Dubremetz and Nivre (2016).	86
C.5	Average precision for chiasmus detection (Sherlock Holmes set), taken from Dubremetz and Nivre (2016).	86
C.6	Results for logistic regression model (Machine) with comparison to the hand-tuned models of Dubremetz and Nivre (2015; 2016) (Human), taken from Dubremetz and Nivre (2017).	86
C.7	Average precision for different feature combinations. D=Dubremetz features, L=lexical features, E=embedding features, taken from Schneider et al. (2021).	87

List of Figures

C.8	Number of correct examples among the top 100 ranked ones in unseen texts for the Dubremetz method baseline, the PoS inversions with Dubremetz features and the Dubremetz+lexical+embedding (DLE) features, taken from Schneider et al. (2021).	87
C.9	Annotation of 100 randomly selected chiasmus, epanaphora and epiphora candidates, taken from Dubremetz and Nivre (2018).	88
C.10	Choosing the best model for epanaphora, taken from Dubremetz and Nivre (2018).	88
C.11	Choosing the best model for epiphora, taken from Dubremetz and Nivre (2018).	89
C.12	Results for the epanaphora experiments, taken from Dubremetz and Nivre (2018).	89
C.13	Results for the epiphora experiments, taken from Dubremetz and Nivre (2018).	89
C.14	Partial Dependency Plot for the <i>parisonIntro</i> feature as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	90
C.15	Partial Dependency Plot for the <i>parisonBetween</i> feature as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	91
C.16	Partial Dependency Plot for the <i>parisonConclusion</i> feature as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	92
C.17	Partial Dependency Plot for the <i>isocolonInTerms</i> feature as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	93
C.18	Partial Dependency Plot for the <i>isocolonIntro</i> feature as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	94
C.19	Partial Dependency Plot for the <i>isocolonBetween</i> feature as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	95
C.20	Partial Dependency Plot for the <i>isocolonConclusion</i> feature as part of the logistic regression classifier based on <i>Dubremetz</i> with all novel features.	96
C.21	Display of the upper half of the regression tree based on <i>Dubremetz</i> with all novel features.	97
C.22	Display of the lower left half of the regression tree based on <i>Dubremetz</i> with all novel features.	98
C.23	Display of the lower right half of the regression tree based on <i>Dubremetz</i> with all novel features.	99

List of Tables

4.1	Composition of the dataset	47
4.2	Results of the extraction pipeline for antimetabole	48
4.3	Results of the baselines evaluation	49
4.4	Results of the standalone features evaluation	50
4.5	Results of the features gradation evaluation	51
4.6	Results of the features ablation experiment	52
4.7	Results of the baselines evaluation for different model types	53
4.8	Results of the standalone features evaluation for different model types . .	54
4.9	Results of the features ablation study for different model types	56

1 Introduction

1.1 Context

This project's subject was first introduced in October, 2021, as part of the PhDTrack program between the French engineering school INSA Lyon (Computer Science Department) and the German University of Passau (Faculty of Computer Science and Mathematics). This program allows its participants to study in both institutions, to have supervision from both sides and, eventually, to obtain a double master's degree. Several special courses and projects in Lyon completed with the courses in Passau, along with this research project as a master thesis, aim at supplementing the students' engineering toolkit with research-oriented skills and giving them another precious point of view for their career prospects.

A substantial share of this project in particular was carried out as a two-man team. Thus, I will try to focus as much as possible on my own work. When needed, I will also specify which parts were covered by my partner, Guillaume Berthomet (his complete work is covered by Berthomet (2023) [Ber23]), or by us both.

1.2 Definition of the Problem

1.2.1 Specifications

The subject for this thesis has evolved quite a lot since its presentation, and has especially been clarified. However, the original transcribed in the appendix B remains sufficiently relevant as a base:

1 Introduction

The goal of this thesis is to extend existing datasets by collecting examples [...] The next step is implementing different rule-based or machine learning algorithms (e.g., active learning for small datasets) and comparing their accuracy for the detection of chiasmus and antimetabole.

Therefore, this thesis places itself at the frontier of two complementary fields: *linguistics* which is necessary for the comprehension, the definition and the investigation of the aforementioned rhetorical figures (*chiasmus* and *antimetabole*¹), and, of course, *data science* which is crucial for the gathering, cleaning, and exploitation of the data, then for designing and building predictive models to allow proper detection of our figures. However, even if the goal of this thesis is to use linguistics only as a mean to enrich the data science, which is its central part, doing data science without the appropriate amount of linguistics merely adds up to playing in a sandbox.²

As stated, the ultimate **goal** of this project is to be able to automatically detect chiasmi - or rather, as we will see, antimetabole. In order to get there, many steps are required beforehand:

- Understanding the linguistic background of the problem (see Section 2.1);
- Exhaustive and thorough reviewing of the state-of-the-art (see Section 2.2.1);
- Analysis of the existing data and eventual new data collection (see Section 3.3);
- Data preprocessing (see Section 3.3.2) which includes two main parts:
 - The development of a tool used for preprocessing the data, notably by extracting chiasmi candidates;
 - The cleaning and annotation of the preprocessed data;
- Proposal and implementation of new models of automatic chiasmi detection allowing to distinguish the relevant candidates from the mass based on traditional Machine Learning models (see Section 3.4.2);
- Evaluation and comparison of the proposed models with the state-of-the-art (see Section 4) which also includes:

¹These two in particular, as well as other rhetorical figures, will be defined precisely later. In the meantime, a glossary of the rhetorical figures mentioned in this thesis can be found in Appendix A.

²This was indeed an antimetabole. One could not resist the temptation of surreptitiously inserting antimetabole in a thesis on antimetabole, even if the primary pedagogical purpose is obviously to get the reader used to the figure.

- Re-implementation and adjustment of the state-of-the-art models to work with the fresh data;
- Definition of the evaluation protocol.

1.2.2 Research Questions

Be that as it may, perhaps the most vital problem of this project was to define the scope of the problem itself: chiasmi or antimetabole? But **what is, even, a chiasmus? And an antimetabole? Which forms can they take? What makes them so particular and valuable?** [RQ1] Hopefully, all these questions should find their answer in the following chapter (Section 2.1).

Once the figures identified, the second question that arises³ concerns itself with where to find them: as we will see in the next chapter (Section 2.2), chiasmi are painfully rare figures along with having been very little researched, so that almost no data is available to this day. Put plainly: **how and where to find interesting chiasmi?** [RQ2]

Following ideas from the state-of-the-art, we tried to answer the second question with yet another one: **is it possible to efficiently⁴ extract chiasmi candidates from any given text?** [RQ3]

Last but not least, once all these issues are tackled, we eventually remember our original goal and come back to the automatic detection of chiasmi: **how can we improve the currently best performing algorithms for detecting salient chiasmi?** [RQ4]

1.3 Contributions Preview

This thesis and the hidden work behind it mainly contributes to this day’s state-of-the-art on three aspects.

First, we put together **a new dataset** of proportions that are incomparable with the existing and available ones (to this day) for chiasmi and antimetabole. By considering that the only current available state-of-the-art’s dataset (see Section 2.2) is comprised of

³The project’s chronological unfolding is overlooked in order to give what should be a more logical and easily understandable explanation.

⁴In this case, efficiency has to be a trade-off between recall - i.e. extracting as many relevant chiasmi as possible -, and the number of extracted candidates - i.e. extracting as few irrelevant candidates as possible.

31 antimetabole instances annotated as *True*, our new dataset thus expands this number by more than 1000%. As for chiasmi, we provided the first list of various types of chiasmi.

Concerning the automatic extraction of chiasmi and antimetabole candidates in plain text, we developed from scratch **the first up-to-date full pipeline for preprocessing and extracting said candidates**⁵. It makes their annotation as easy as possible, by allowing one to rank them with a proper detection model and annotate only the top hits with a compatible open-source annotation tool. The pipeline also permits the re-injection of the annotated data within the pipeline’s first output and within the original text in a readable format.

Lastly, we **re-implemented the state-of-the-art and several models of our own** for detecting salient antimetabole among uninteresting candidates⁶. By doing so, we manage to outperform the state-of-the-art by improving their features as well as introducing our own novel features. Further insights are proposed on which features work best and why.

The first two contributions were part of the collaborative work between this thesis and Guillaume Berthomet’s (see Section 1.1). The third one, however, was individual work (entirely and only done by the author of this thesis).

1.4 Structure of the Thesis

This thesis tries to follow the classic structure of a research article, a perfect fit for a research project.

The present introduction is followed by a chapter 2 presenting the necessary background for the rest of the project: linguistic considerations will be discussed there, by engraving in stone the most important definitions for the project; then, a full review of the state-of-the-art, which is most crucial for the subsequent chapters, will be presented.

The methodology chapter 3 traces back how and why the project has been managed, before disclosing in detail the different technical processes implemented to answer the aforementioned research questions. Particular attention will be given to everything concerning the automatic extraction of chiasmi candidates, the data and the resulting dataset, and the automatic detection of rhetorically salient chiasmi. The various Machine

⁵Available at <https://github.com/YohanMeyer/ChiasmusExtractor>, visited on 18/01/2023.

⁶Available at <https://github.com/YohanMeyer/AntimetaboleDetector>, visited on 18/01/2023.

1 Introduction

Learning models and features used for the detection part will be presented.

Afterward, the outcomes of various experiments concerning the candidates extraction tool and the detection models will be submitted in the results chapter 4. The detection phase will be broken down into three major parts: the evaluation of different versions of the baselines, the evaluation of some baselines augmented with our novel features and the evaluation of different models (with the baselines and the novel features).

Thereafter, a global interpretation of all these results will be presented (Chapter 5). The different baselines will be dissected and their results will be explained in detail on the main model, and the results of the main baselines will also be analysed when applied to different types of models. This will be followed by a thorough analysis of the novel features introduced in this thesis and their impacts on the predictions of the main model. Thereafter, a complete walk through a regression tree model will be proposed.

In the last chapter 6, we will take a step back and take a critical look on the project as a whole, whom we will dispute the strengths and weaknesses, and examine crucial choices made during the project. The potential applications of this project and further observations on public research will be discussed, before moving on to a more personal appraisal of this project and its ins and outs. It will then proceed on offering suggestions for different approaches or improvements for future research on the matters presented in this thesis.

2 Background

2.1 Terminology

2.1.1 Chiasmus

What is a *chiasmus*? Many contradictory or complementary answers can be found in dictionaries, articles and previous research. Let us hear what they all have to say. The simplest definition may be given by the Collins Dictionary [11a]:

rhetoric

[chiasmus:] reversal of the order of words in the second of two parallel phrases

he came in triumph and in defeat departs

The Oxford English Dictionary reports a very similar one, borrowed from A. S. Wilkins [Wil71a]:

Grammar.

A grammatical figure by which the order of words in one of two parallel clauses is inverted in the other.

From these first two definitions, we can keep the consensual idea that a chiasmus needs a reverse repetition of words. Nevertheless, some distinctions already arise with the confusion between a *phrase* and a *clause*, and whether a chiasmus is a *rhetorical* or a *grammatical* figure. But if we consider the example given by the Collins Dictionary,

2 Background

both definitions lack a crucial element: the inverted words do not have to (or must not?) be identical.

Finally, the Merriam-Webster [22] has a very different and thus peculiarly interesting formulation:

chiasmus: an inverted relationship between the syntactic elements of parallel phrases (as in Goldsmith's *to stop too fearful, and too faint to go*)

Here, the *words* become *syntactic elements*, and the inverted or reversed *order* becomes an inverted *relationship*. The new example from Goldsmith [Gol64] thus appears much more coherent, as the inverted elements {to stop, to go} and {too fearful, too faint} do present a clear relationship (respectively antonymous and synonymous).

However, we still lack many crucial details and some controversies need to be worked out; to that purpose, let us delve into linguistics and research, starting with Greene (2012) [Gre+12]. Its first attempt at defining chiasmus (p.225) is much more complete, more restricting in some ways and more open in others:

The repetition of a pair of sounds, words, phrases, or ideas in the reverse order, producing an abba structure [...]

Indeed, we now see our figure pinned down to two pairs of undefined terms in a strict ABBA pattern. But as the terms can virtually be anything, from sounds to ideas, it also appears virtually impossible to detect all of them automatically. However, we can find a slightly more restrictive definition, yet still very vague, from Kelly et al. (2010) [Kel+10]:

A related figure, chiasmus, is defined as “[r]epetition of grammatical structures in inverted order” [...]

On another hand, Greene presents us yet another proposition borrowed from Thompson (1995) [Tho95]:

If, as Thomson prefers, we construe chiasmus more broadly as the bilateral symmetry “of four or more elements around a central axis” [...]

Eventually, chiasmus may well be an inverted repetition of any number (still greater than two) of literally¹ anything. But such a definition will not get us anywhere. To add more complexity, the entry from Greene continues with:

Although chiasmus frequently describes the repetition of particular phonemes or the inversion of clauses, it is not uncommon to find that an entire poem or novel has a chiastic structure or that several kinds of chiasmus are at work simultaneously.

Now, this is definitely going too far for our purpose. For the sake of simplification and clarification, we shall prefer using the terms *Envelope* or *Ring Composition* (found respectively at pages 436 and 1201 of the same book) for figures that enclose material such as poems, texts, novels, etc.

Hopefully, we may be able to sort out everything with the help of Nordahl’s essay (1971) [Nor71], which has the ambition to “rehabilitate” the chiasmus along with defining it from top to bottom - and from bottom to top. Nordahl then distinguishes three categories of chiasmi, each with its own subcategories: the rhetorical chiasmi, the grammatical chiasmi, and the semantic chiasmi. Unfortunately, this classification does not help us in our quest for simplification, because the thirty different types of chiasmi are among others comprised of what we call *antimetabole* (see Section 2.1.2), and of others whose rhetorical effect (see Section 2.1.3) is more than dubious.

Let us try a final attempt with Dupriez (1980) [Dup80]. Its definition still includes an inverse repetition, but of “syntactically identical segments of groups of words”, which seems like a stricter and at the same time more permissive version of the Merriam-Webster’s.

Ultimately, the only two assumptions we can reasonably make concerning the definition of chiasmus are:

¹The word *literally* should not be used lightly. Otherwise, its use could lose its literal meaning.

- the reverse repetition of *something*;
- the rest is up to you.

In order to render our goal of automatically detecting chiasmi feasible, we need to pin down our own definition and focus on one specific type of chiasmus. Since we are interested as well in antimetabole, it seems reasonable to limit our interest in chiasmi to those closest to antimetabole, that is to say **lexical chiasmi**. Moreover, while preferring to keep as much characteristics as wide as possible, previous research motivates us to reduce the chiasmi terms to simple words for extraction purposes.

Hence our following definition:

A **chiasmus** consists of the repetition of two or more pairs of related words in the reverse order.

2.1.2 Antimetabole

Now that we have tackled the mighty chiasmus, we are allowed to hope that defining the *antimetabole* shall be easier. Nevertheless, this figure may be less well-known or less common than chiasmus, as some dictionaries (such as the Merriam-Webster) do not report it and other sources mark it, a little disdainfully, as a specific type of chiasmus. Interestingly enough, the Collins Dictionary [11b] does not seem to distinguish our two figures otherwise than by their name:

rhetoric

[antimetabole:] the repetition of words in reverse order for emphasis

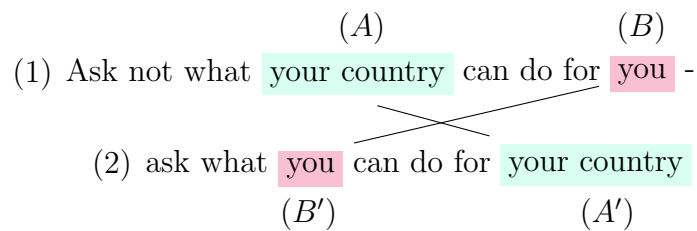
When compared to its definition of the chiasmus, only the specification of “two parallel phrases” disappears, but nothing is said about the terms composing the figures themselves. Fortunately, the Oxford English Dictionary [Wil71b] gives us more insight:

Rhetoric. A figure in which the same words or ideas are repeated in inverse order.

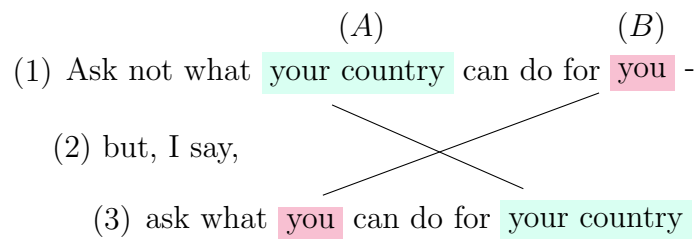
2 Background

We notice that while the chiasmus is considered a grammatical figure by some, the antimetabole would solely be a rhetorical figure. We can also note the similarity between this definition for an antimetabole and Greene’s first one for the chiasmus, mentioning a repetition of *ideas*, although the term remains very vague. Anyhow, we can already confirm that the antimetabole and the chiasmus are intertwined with each other and that both have the same default of being pretty troublesome to define.

To further investigate on the differences between chiasmus and antimetabole, Greene (2012) suggests that an antimetabole may indeed only be a special case of a chiasmus, that is a chiasmus where the paired elements are identical words - or almost identical, with “variation in case or tense”. Dupriez (1980) [Dup80] points in the same direction by bringing closer both figures, but specifies that the antimetabole consists of only two pairs of words. On another hand, even if Rabatel (2008) [Rab08] agrees with the former definition of chiasmus, its approach on the antimetabole appears even more restrictive: the repetition would have to be in two successive clauses. Kennedy’s famous quote

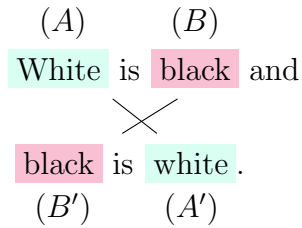


would therefore, according to this definition, stop being an antimetabole if we rephrase it as



As curious as it may seem, Rabatel carries on with saying that not all antimetabole are in fact antimetabole - and there starts a witch hunt against *false antimetabole*, whose terms would be interchangeable in the absence of any subordinate relationship between them, such as:

2 Background



It is also worth noting that Kelly et al. (2010) [Kel+10] seems to agree with Rabatel:

An antimetabole is a “[r]epetition of words, in successive clauses, in reverse grammatical order” [1]; an example of this figure: “I said what I meant, I meant what I said” [18]. It should be clear from the example that the structure of the phrase is important, as is the repetition of the same words.

[1] Burton (2007) [Bur07]

[18] Seuss (1968) [Seu68]

This article already hints at the rhetorical effect (see Section 2.1.3) which would be part of the antimetabole’s definition, while also giving us the notion that said rhetorical effect is as obscure as could be. “It should be clear”, but nothing really is.

So, in an effort to clear some things up, we will call up to Beauzée (1782) [BM82] with his *Encyclopédie Méthodique* (p.198) based on the famous one from Diderot and d’Alembert (1751-1772). This one surprisingly does not make any mention of the chiasmus, but dedicates an article to a resembling figure, the “*antimetalepse*”², and our antimetabole. According to the *Encyclopédie*, the antimetabole is a figure of repetition where the words of the second part change in order and function, where the antimetalepse is about ideas rather than words. It also explains how one can switch from the antimetalepse to the antimetabole (and from the antimetabole to the antimetalepse) by simply changing the used words:

We should eat to live, and not live to eat [antimetabole]

We should eat to live, but not use every moment of our life to gorge ourselves with food. [antimetalepse]

²A neologism used as no satisfying translation to English could be found for the French word *antimetalepse*.

And together, these two figures would form the *antimetathesis*. This distinction has the crucial advantage over the Oxford English Dictionary [Wil71b] and Rabatel (2008) [Rab08] to put the antimetabole within our automatic detection reach and to make it that much clearer. To put it in a more formal way, Harris et al. (2009) [HD09] proposes the following:

Antimetabole: Repetition of words in reverse order.

$[W]_a \dots [W]_b \dots [W]_b \dots [W]_a$ *Drake loves loons. Loons love Drake.*

As long as the matching words remain in Greene’s range of semi-open identicalness with “variation in case or tense”, this definition seems very promising for our use case. To make it our own, we would only need to add the idea (same as for the chiasmus) that the figure should not be limited to two pairs, to expand our range:

Antimetabole: Repetition of words in reverse order.

$[W]_1 \dots [W]_2 \dots [W]_n \dots [W]_n \dots [W]_2 \dots [W]_1$ *Drake loves loons. Loons love Drake.*

In order to mark the difference between the chiasmi and antimetabole from previous research and our own, we will settle on calling chiasmi (and antimetabole) with more than two pairs of terms *nested* chiasmi (and antimetabole).

2.1.3 Rhetorical effect

There still is a somewhat trivial but major problem with both definitions at which we arrived. Indeed, if taken literally, our definition of the antimetabole would cover *any* reverse repetition, even the most insignificant ones, and without any scope limit, the number of occurrences grows ridiculously large with the number of words. Any repetition of prepositions, conjunctions, pronouns, and articles, to cite only them, would be classified as an antimetabole.

From a linguistic point of view, this is not problematic *per se*, as argue Harris et al. (2018) [Har+18]. They take these two antimetabole to illustrate their point:

2 Background

[26] There are only two kinds of men:

(A)
(B)
 the righteous who think they are sinners
 and the sinners who think they are righteous.
(B')
(A')

(A)
(B)
 [27] You hear about constitutional rights, free speech
 and the free press. Every time I hear these words I say to myself [...]
(B')
(A')

The correct way to characterize 26 versus 27 in terms of Rhetorical Figures, then, does not involve the presence or absence or degree of antimetabole. [...] if antimetabole is reverse lexical repetition, as every definition ever formulated has it, then both 26 and 27 exhibit antimetabole.

If we take our figures definitions *stricto sensu*, it makes sense to put all repetitions, regardless of their meaning and context, on the same level. However, we are not interested in detecting *all* antimetabole and *all* chiasmi, only those we deem *interesting enough*. It then becomes apparent what our previous definitions are lacking: we still have to define what is a **rhetorical effect**, the same effect that makes our figures worth detecting, and which promotes the “grammatical” figure to the rank of “rhetorical” figure. In the end, we are not so much interested in simply extracting chiasmi and antimetabole from plain text, but above all in detecting the instances that detain a powerful rhetorical effect. Hence the title of this thesis: “detection of chiasmi’s rhetorical salience”.

Greene (2012) [Gre+12] and Ruan et al. (2016) [RDH16] give us hints on what this mysterious rhetorical effect may be about: the former suggests that antithesis mixes up well with antimetabole, and the latter goes further by naming it:

We call this phenomenon, when figures pile on other figures, *stacking*.

Antimetabole in particular would have a tendency to be stacked with *mesodiplosis* and *antithesis*³, for which Harris et al.'s [26] is a perfect example. But if this *stacking* explains how a figure's rhetorical effect can benefit from other figures, it does not suffice to express a standalone figure's effect, such as Samuel Johnson's very short antimetabole⁴, where no apparent mesodiplosis or antithesis supports the figure:

It made **Rich** gay and **Gay** rich.

The complication here lies in how rhetorical figures in general and our two in particular can serve many different purposes with each their own intent and effect.

To illustrate the former, Mitrović et al. (2017) [Mit+17]⁵ point out that rhetorical figures (and schemes in particular) can either “affect the audience's rational perception of a standpoint”, “constitute or emphasize arguments”, or even “epitomize arguments in their form”. Circling back to Appendix B, chiasmi's or antimetabole's effect could be measured by the way they seem appealing, convincing, or well put.

On another hand, Ruan et al. (2016) [RDH16] propose four different rhetorical functions to describe antimetabole's effects (Reciprocal Force, Reciprocal Specification, Comprehensiveness, Irrelevance of Order), which can also be practical means of separating the wheat from the chaff.

Nevertheless, when it will come down to annotation, these leads will not help us getting completely rid of the subjectivity underlying it all. In a sense, rhetorical effects can only be subjective in that they have different meanings and trigger different reactions in each of us. The subsequent problem of obtaining annotated data without spoiling it with our biases will be addressed in Section 3.3.

2.2 State of the Art

2.2.1 Rhetorical Figures

Let us now take a step back and wonder where our interest for our two figures, namely chiasmus and antimetabole, comes from. Here, we consider two groups of tasks linked

³A curious reader may find definitions for both of these figures in the Glossary (Appendix A).

⁴This was said about John Gay, the author of “The Beggar's Opera” that was rejected till John Rich, the theatre director, helped make it a success [Gro02].

⁵Any conflict of interest or any relation to the first examiner of this thesis is purely coincidental. This paper is only cited for its objective academic excellence and pertinence.

to rhetorical figures: data-driven detection / extraction and ontology-based modelling. The former (e.g. [Dub13; DN15; DN16; DN17; DN18; Sch+21]) should help deepening our understanding of them from an empirical point of view. The latter (e.g. [HD09; MM13; KMG22; Wan+22]) are based on a deeper linguistic expertise allowing to better exploit the obtained results. Thus, instead of only trying to *detect* rhetorical figures in pieces of text, these approaches aim at *modeling* these figures; instead of trying to extract rhetorical figures and some statistical *insights* about them from raw corpora, these ontologies rather focus on extracting *knowledge*. In the same way that these approaches may differ, they are also complementary - when they do not overlap.

Harris and Di Marco (2009) [HD09] are certainly pioneers in this domain, their article being followed by a series of other works of the authors pursuing the same goal: building “A Cognitive Ontology of Rhetorical Figures” (Harris et al., 2017) [Har+17]. Their ontology was built in OWL (Web Ontology Language) using a bottom-up approach, after having tested both top-down and middle-out techniques. The “cognitive” part comes from the ontology being organized around “cognitive affinities” (reported examples are repetition, similarity, and contrast) that are, as the authors argue, part of the essence of rhetorical figures. The ontology thus combines these different affinities in order to classify and define many rhetorical figures. However, as every figure may leverage these cognitive affinities differently (for instance, a repetition can have a semantic, lexical or syntactic nature), the ontology also maps every pair consisting of a rhetorical figure and of an affinity to a mode, representing how the former *uses* the latter.

Following the inspiration of Harris and Di Marco (2009), Mladenović and Mitrović (2013) [MM13] developed the Ontology of Rhetorical Figures for Serbian (why should the English language be the only centre of attention?). Their article allows them to introduce RetFig, “a formal domain ontology for rhetorical figures for Serbian”, with the principal motivation to enrich and develop the fields of argument and opinion mining, as well as semantic analysis. Their ontology was built upon a manually retrieved and annotated database of 98 different rhetorical figures, each corresponding to a *rhetorical type* and a *linguistic category*. Both like and unlike Harris et al. (2017), the ontology was constructed manually with OWL 2 and with a top-down methodology. Thus, while the bottom-up approach starts from specific instances of rhetorical figures (Harris et al.’s very first was an antimetabole!) to build their model because their “instances were not falling into neat categories”, Mladenović and Mitrović rather started by creating a taxonomy of linguistic and rhetorical concepts. From there, many elements and relationships taken from linguistics were manually inserted into the ontology, allowing for finer and

2 Background

finer definitions, and eventually developing a formal model of rhetorical figures. This model could then be exploited through SPARQL (SPARQL Protocol and RDF Query Language) queries, providing a precious help for the annotation of rhetorical figures, and for statistical studies (through rhetorical figures, obviously) of a text.

A much more recent work on an ontology of rhetorical figures, this time for the German language (each in its turn), was published last year by Kühn et al. (2022) [KMG22]. As one of the authors participated in the ontology for Serbian, GRhOOT (the **G**erman **R**het**O**rical **O**n**T**ology) was based on RetFig. Where RetFig handled 98 figures, GRhOOT took it a step further by managing 110 different figures. Furthermore, their approach “allows for an easy extension or translation into other languages”, leaving similar ontologies to be developed in different languages (as a French student, I certainly have a preference for one). Without surprise, the tools and methodology used to build GRhOOT are the same as for RetFig. However, the mapping of the modeled rhetorical figures in both languages remains challenging, as only 60 figures have an identical representation in both ontologies - and even then, they may have different properties. Therefore, even if the translation is rendered easier by design, building an ontology in another language remains time-consuming and still requires linguistic expertise. Lastly, Kühn et al. test their ontology with competency questions, the same way as Mladenović and Mitrović. In addition to thoroughly checking the validity of the ontology, it also allows for a direct comparison between ontologies, and thus for an analysis of the differences between languages when it comes to rhetorical figures. Eventually, as mentioned above, this article recalls that their approach may overlap with the one we follow in this thesis:

It can be used to develop a rule-based approach for rhetorical figure detection.

It can also be used to guide people to identify rhetorical figures. Therefore, it can support human annotators, too.

As we will see in subsequent sections and chapters, identifying and annotating rhetorical figures was a main challenge of this thesis, and rule-based approaches for antimetabole detection are also part of the models presented in Chapter 3 - although their construction differs a lot.

Now that we have a global overview of how to model rhetorical figures, we may go back to the two that interest us above all: chiasmus and antimetabole.

2.2.2 Chiasmus and Antimetabole Detection

In this Section, we overview the existing approaches of chiasmus/antimetabole automatic detection discussing them in chronological order.

Before 2013

To the best of our knowledge, Gawryjolek (2009) [Gaw09] was the first to grow interest in the automatic detection of antimetabole, for which he gives the same definition as Harris et al. (2009) - see Section 2.1.2). Gawryjolek starts simply with his antimetabole detection by choosing to take into consideration all matching repetitions he could find, and only those of exactly matching words. Although, using the same reasoning as we did in 2.1.3, he recognizes that this method “produces a lot of antimetabole that are not necessarily important from the rhetorical point of view”. Taking all this into consideration, Gawryjolek’s algorithm ends up being more accurately described as a *salient antimetabole candidates extraction tool* rather than as an *antimetabole detector*, since his ultimate goal is not to extract all antimetabole from a text, but only the “important” ones. The distinction has its importance, as it will likewise assist our own work. Thus extracting candidates, Gawryjolek is left with the task of selecting the best antimetabole out of his candidates, and developed to that end JANTOR (Java ANnotation Tool Of Rhetoric).

In an effort to make a similar but multilingual extraction tool, Hromada (2011) [Hro11] undertook the path of PERL-compatible regular expressions, applied to four schemes of repetition (anadiplosis, anaphora, epiphora and antimetabole). For the figure that interests us (if you thought about something else than antimetabole, you should consider reading more slowly), Hromada does not make matters easy by taking Gawryjolek formal description but rather adds his own to the list:

one can formalize it as follows:

$$\langle W_A W_B W_C \dots W_C W_B W_A \rangle$$

He then gives the example of “Alle wie einer, einer wie alle.”⁶ to illustrate his point. This definition is much stricter than our own given in Section 2.1.2, and actually appears to be mixing antimetabole with mesodiplosis while forcing the antimetabole terms to

⁶German for “All as one, one as all.”

be juxtaposed. This method still allows Hromada to find some instances in his dataset, composed of four corpora of famous writers (Shakespeare for English, Goethe for German, Molière for French, and Cicero for Latin). But still unlike Gawryjolek, he does not seem to be interested in his antimetabole’s rhetorical salience. The only mention of “false positives” concerns an epiphora that was matched on a character’s name (as a caption) in a play.

Dubremetz (2013)

In her first article, Dubremetz (2013) [Dub13] starts a long and productive series of experiments on chiasmi detection. Unlike her predecessors, Gawryjolek and Hromada, Dubremetz attempts to develop an actual *detection* rather than a mere *extraction* tool and filters out uninteresting chiasmi, which she calls “pseudo-chiasmi”.

She builds her algorithm for French language on top of Gawryjolek’s idea and adds three major improvements:

- the restriction of the length for chiasmi candidates to 30 tokens;
- the introduction of stopwords to filter out candidates in which one term is in the stopwords list;
- the analysis of punctuation, by filtering out a candidate if hard punctuation is found between two non-central terms of the antimetabole.

The algorithm is tested on a corpus put together by Dubremetz for the occasion, taking forty-three chiasmi in total (36 antimetabole and 7 chiasmi according to our definitions) from dictionaries, previous research or extracts of well-known poetry. To expand her algorithm beyond “strict antimetabole” (based on an exact word match), Dubremetz successfully makes use of a lemmatizer, and less successfully of a stemmer (due to the stemmer’s performance in French). Dubremetz uses the French lemmatizer *Flemm* by Namer (2000) [Nam00] coupled with the *TreeTagger* by Schmid (1994) and allows her algorithm to take plain text as input. Eventually, she even tries to detect (what we call) chiasmi and not only antimetabole, by searching for chiasmi whose terms are synonymous. However, her use of OpenOffice’s thesaurus yielded unsatisfactory but still interesting results. Moreover, it was also made apparent that the extraction of chiasmi candidates gives a disproportionate number of uninteresting ones due to the affinity between common words (so, like, such...)

Dubremetz & Nivre (2015)

It is regrettable to see that in Dubremetz and Nivre (2015) [DN15], Dubremetz seems to have cast aside the linguistic rigor that led her in 2013 to distinguish four variants of figures for antimetabole and chiasmi. This time, although she only focuses on antimetabole, she puts everything under the name of “chiasmus” for the sake of simplicity. However, this misuse is easily forgiven when considering the confusion ruling over the definition of the chiasmus (see Section 2.1). Back on the hunt for antimetabole, then, Dubremetz and Nivre first identify three major hurdles to overcome:

- The rareness of chiasmi, for which their proof is an analysis of a corpus of 130 000 words (and 66 000 extracted inversions) that would contain only 1 salient antimetabole;
- The number of false positives obtained through the extraction of candidates (1 true positive for 66 000 false positives in the same example);
- The lack of annotated data and the difficulty to provide said annotated data.

In the face of these issues, the authors decided to adopt a ranking strategy, inspired from information retrieval techniques. Instead of binary classifying candidates as *salient* or *not salient* (which we deem more accurate than “true” and “accidental” chiasmi), Dubremetz and Nivre’s algorithm now gives each candidate a *salience score*, calculated with four types of features.

The first type of features is simply taken from the previously seen article (Dubremetz, 2013); the second takes into account the number of words between terms of the antimetabole; the third one looks at N-grams for repetitions apart from the antimetabole itself, following Hromada’s interest for mesodiplosis; the last one reviews negations and conjunctions that would “underline the axial symmetry”, based on various definitions and research about chiasmi.

The algorithm used to extract candidates from text has not changed much, contrarily to the corpus: now working in English, the authors chose Europarl (4 million of words), a corpus of political discussions. And even though the core of the extraction program has not changed, it now only uses the TreeTagger by Schmid (1994) [Sch94], since no additional program specialized for French is needed. All extracted candidates are then fed to the new scoring function, whose weights were manually fitted.

Concerning the evaluation of the model, it is computed on the “top 200 hits given by the

2 Background

machine”. However, the algorithm output being based on a ranking, it is not possible to simply compute the recall and precision like for any classification task; or, to make it possible, a threshold would have to be defined over which a score would be considered sufficient to make the candidate tagged as “positive” and “negative” if the candidate’s score is below the threshold. This method presents the major disadvantage of being heavily dependent on the threshold’s value, which would have here to be empirically set, and reevaluated every time that the weights or features are revised. Moreover, its value would not have any meaning in itself, and could make the interpretation of the results as unreliable as impractical.

Accordingly, Dubremetz and Nivre decided to further draw inspiration from information retrieval methods in Croft et al. (2010) [CMS10]. In their book, Croft et al. describe several methods of evaluating search engines, based on a ranking system similar to Dubremetz and Nivre’s implementation:

The third method, and the most popular, is to summarize the ranking by averaging the precision values from the rank positions where a relevant document was retrieved (i.e., when recall increases). If a relevant document is not retrieved for some reason, the contribution of this document to the average is 0.0. [...] Average precision has a number of advantages. It is a single number that is based on the ranking of all the relevant documents, but the value depends heavily on the highly ranked relevant documents. This means it is an appropriate measure for evaluating the task of finding as many relevant documents as possible while still reflecting the intuition that the top-ranked documents are the most important.

Apart from the last point, i.e. that the candidates with the highest scores are most important, this method seems perfectly appropriate for the task at hand. In this case, a relevant document that would not be retrieved in the top 200 candidates (with regard to their score) would have a contribution of 0.0 to the average precision.

With all this in mind, Dubremetz and Nivre show that all their features improve the recall and the average precision, as can be seen in the figure C.3 (appendix C.2). In their gradation experiment, where they compare different versions by adding one feature after the other, it is difficult to guess which features contribute the most to the global average precision in this experiment, since the order for adding the features matters. However, all three features seem to improve the results substantially, suggesting that they are all relevant. But even the best results leave plenty of room for improvement, with a recall

of 90% (17 out of 19) and an average precision of only 61% for the top 200.

Moreover, the very limited dataset and the manual setting of the weights make the experiment quite unreliable. The previously identified issues are thus far from being tackled yet.

Dubremetz & Nivre (2016)

In their subsequent article, Dubremetz and Nivre (2016) [DN16] “start from the shallow feature-based algorithm introduced by Dubremetz and Nivre (2015) and extend it with features based on syntactic structure.” This work can be seen as a plain improvement of their own past research, whether it be on the data, on the model or on the results.

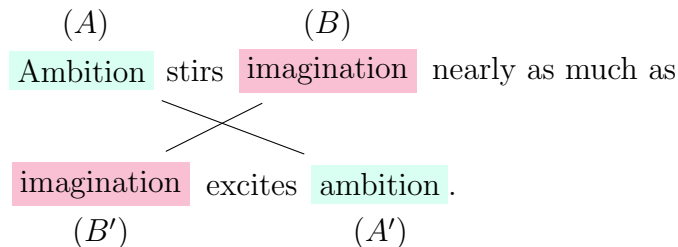
Concerning the datasets, Dubremetz and Nivre reuse their annotated corpus of four million words extracted from Europarl (see Section 2.2.2), but instead of splitting it in two for training and evaluating their ranking model, they dedicate it entirely to the training. Therefore, they annotate another extract from the same source consisting of two million words for the testing phase. Moreover, they also test their trained model on a completely different dataset coming from literature: the Sherlock Holmes novels and short stories by Conan Doyle. No precision on their choice or its relevance is given, but this new corpus is supposed to be an archetypical literary text, used for testing how well their model can generalize on another genre.

Their model needing additional input compared to its previous version (having only a lemmatizer), Dubremetz and Nivre replace the *TreeTagger* by the *Stanford CoreNLP toolkit* by Manning et al. (2014) [Man+14], which allows for Part-of-Speech tagging. However, the antimetabole candidates extraction tool remains the same - except for a new visual feature helpful for annotation. But this new addition does not come without a cost: at the time, it took them days to process 2 million words.

Regarding the rating model used for detecting the antimetabole, the authors use the exact same previously described methodology, and add to it two novel features based on Part-of-Speech tags and syntactic dependencies. In addition to sharing the same lemma, Dubremetz and Nivre thus hypothesize that the matching terms of an antimetabole should share the same grammatical category, and add an extra weight if all four terms share the same Part-of-Speech tag. However, this weight could also promote a lot of usual uninteresting antimetabole, notably composed of four common words such as articles, verbs or determiners; this effect could for instance counterbalance the stopwords feature. Following the same logic of chiasmic symmetry going beyond the semantics, this time

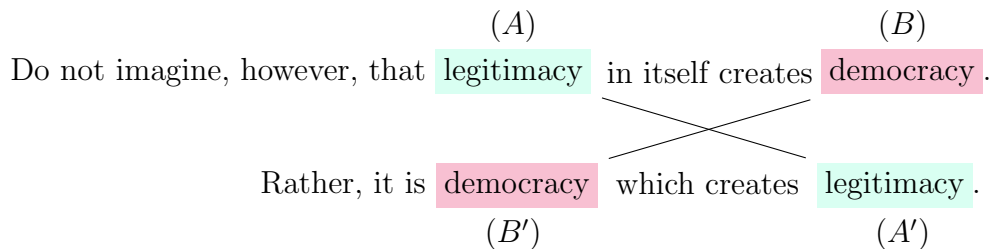
2 Background

focusing on repetition rather than inversion, Dubremetz and Nivre reward antimetabole whose dissimilar terms have the same syntactic dependency and penalize those whose matching terms have the same dependency. This hypothesis is motivated by the analysis of the following antimetabole by Winston Churchill:



Here, as the terms are repeated and inverted, they both apply the same force on each other (ambition *stirs* and imagination *excites*), serving the function Harris et al. (2018) [Har+18] call “Reciprocal Force”. In the second clause, the repeated words thus exchange their place and take the same syntactic dependency as their counterpart.

Their training corpus only contains 31 instances of what Dubremetz and Nivre call “true chiasmi”, and is therefore not suited for classical machine learning algorithms, which would not have enough to work with. As in their previous work, they chose to tune the new features’ weights manually. However, their testing set only contains 13 instances annotated as *True*, reiterating the issue of reliability that occurred in their previous article. Indeed, a considerable shift of **8%** in recall only translates the fact that the model was able to detect *one* more salient antimetabole. And it is exactly what happens - with both new features, the model now catches the following salient instance in its top 200-rated candidates pool:



Obviously, the two new features are “tailored” for this instance: the four terms share the same Part-of-Speech tag (noun) and their dependency structure is also a perfect fit

(same for the first and third, same for the second and fourth, and different for matching words). At this point, it should be clarified that we are not trying to imply anything about Dubremetz and Nivre’s work. However, it simply has to be pointed out that an increase in recall and (thus in) average precision, with a testing set of only 13 instances, should not mean much, if even anything.

All that being said, the progress made with only two new features seems remarkable: as we can see in the figure C.4 (appendix C.3), both features improve significantly the average precision when taken separately. The “tag features” improve the average precision by 17 points⁷ and the dependency features by 22 points. Although when all features are mixed, the average precision only increases by 25 points, which is only 3 more than with the dependency features. This may suggest that both features give generally a better score to similar instances, like the aforementioned example. Furthermore, in their generalization experience, Dubremetz and Nivre evaluate their model on the Sherlock Holmes set, which only contains 8 positive instances. Once again, the improvement of 17 points (see figure C.5, appendix C.3) for the average precision over the baseline should not lead to any hasty conclusion, as a test set of merely 8 instances may be so imbalanced that the evaluation becomes completely biased.

Moreover, the core of the previous example and the idea that motivated the addition of these features could well be what Harris et al. (2018) [Har+18] theorized two years later as the rhetorical function *Reject-Replace*:

For instance, when antimetabole collocates with mesodiplosis and antithesis, the combined function is primarily to reject the negated predication utterly and replace it with the positive predication.

Reject–Replace

We don’t build services to make money; we make money to build better services. (*services/make money; to; We don’t X, we X’*)

When looked closer at, this rhetorical function does directly point at these Part-of-Speech and dependency features: in order to “reject” the first predication and “replace” it, the four antimetabole terms *must* have the same Part-of-Speech tag and the dependency structure described earlier. Eventually, these new features do not so much improve the model’s overall efficiency, but rather allow it to detect a “new type” of antimetabole, should we classify them by their rhetorical function.

⁷and not by 14, as is indeed showed in the table taken from the article.

Dubremetz & Nivre (2017)

In the last article from their series about chiasmi (leading to a PhD thesis (Dubremetz, 2017) [Dub17]), Dubremetz and Nivre (2017) [DN17] eventually end up having enough annotated data to apply Machine Learning methods to their detection model. Using the same corpora and features as in 2016 for training, they only take the test set from Europarl and do not reuse the Sherlock Holmes corpus.

Their model consists of a simple binary logistic regressor: an instance is either considered *true* or *false*. They report trying support vector machines (SVM) with various kernels instead of a logistic regressor but with similar results. The implementation was done using scikit-learn (Pedregosa et al., 2011 [Ped+11]), a widely popular Python library for classical Machine Learning. In order to make the most out of the 31 positive instances for the training phase, it was conducted using two-fold cross-validation.

The main goal of this article was thus to compare the results of models using the same features two by two, one with manually hand-tuned weights and the other with computationally learned weights. As expected, the results in the figure C.6 (appendix C.4) show that the machine-learning-based model outperforms the hand-tuned ones. This difference is most significant for the baseline (Dubremetz et al., 2015), with an improvement of 15 points for the average precision, but shrinks considerably to 3 points when all features (Dubremetz et al., 2016) are used.

Concerning the weights of the features, small and expected differences between the hand-tuned and the learned values explain the 3% improvement in average precision, but a surprising result lies within a baseline feature: *#sameTrigrams*. This feature represents the number of trigrams that are identical in the intermediate spaces between the first and second terms (respectively A, B) and the third and fourth terms (respectively B', A') of the antimetabole. Intuitively, this feature was set as positive during the hand-tuning (like all features under the “Similarity” section that look for a sort of mesodiplosis), but the machine-based learning set this one in particular as negative. However, this sort of result is not exceptionally surprising considering the size of the training set (31 true instances).

Schneider et al. (2021)

To this day and to the best of our knowledge, the most recent English article concerning automatic chiasmi detection was written by Schneider et al. (2021) [Sch+21].

2 Background

The distinction made in the Terminology section 2.1 between chiasmi and antimetabole now becomes of the utmost importance, since Schneider et al. actually try to adapt Dubremetz and Nivre’s (2017)⁸ work on antimetabole to chiasmi. Their exact definition of a chiasmus, very similar to ours, was borrowed from Fauser (1994) [Fau94]:

Chiasmus is defined as an inversion of semantically or syntactically related words, phrases, or sentences in an A B B’ A’ pattern.

Just like Dubremetz and Nivre, Schneider et al. restrict their definition of a chiasmus to four words (since they disregard phrases and sentences). What’s more, they also make the distinction between extracting (or “searching for”) chiasmi candidates and detecting salient chiasmi (or “filtering the candidates”). Indeed, these two phases are harder to merge since the method proposed by Schneider et al. for extracting candidates is very different from Dubremetz and Nivre’s.

Indeed, tasked with finding chiasmi rather than antimetabole, the authors of this article chose to search not for identity of lemmas, but rather for “inversions of part-of-speech (PoS) tags in an A B B’ A’ pattern”. Therefore, they chose to focus on the *syntactical* part of their own definition rather than on the *semantic* part. With this method, they announce finding much more candidates (and thus “false positives”) than Dubremetz and Nivre, with among them chiasmi as well as antimetabole.

Their filtering model uses Dubremetz and Nivre’s features as a baseline. Since the extraction algorithm does not rely on lemmas anymore, they add them as a lexical feature for their model. A binary feature thus captures identities of lemmas between all six possible pairs of the chiasmi’s terms. Moreover, since the baseline and the new feature do not allow them “to distinguish true chiasmi from random PoS tag inversions”, they add an embedding feature to capture, as per their definition, any semantic relation between the terms:

For each pair of supporting tokens, we add an embedding feature equaling the cosine similarity (Salton et al., 1975) [SYY75] of the word embeddings of the two tokens.

⁸The authors rather cite Dubremetz and Nivre (2018), but as far as antimetabole are concerned, nothing changed since Dubremetz and Nivre (2017).

At this point, one can be wondering why Schneider et al. chose to extract their candidates with PoS inversions and to filter them with embedding features, and did not extract candidates with embedding inversions and filter them with PoS features (easy but tempting). Indeed, even if it is legitimate to base a definition of a chiasmus on a syntactical inversion (see Section 2.1.1), in the context of detecting chiasmi with an interesting rhetorical effect, it does not appear to be relevant at first glance. The inversion of semantically related words, however, seems to make more sense, since it is much easier to recognize for a human brain. Furthermore, they use this second approach to explain their first example:

[Chiasmus] can be used, for example, to emphasize contrasts. One example for it is:

Eng ist die **Welt**,
und das **Gehirn** ist **weit**
(Narrow is the **world**,
and the **brain** is **wide**)
Wallensteins Tod (Schiller, 1799)

The semantically related words are narrow and wide, as well as world and brain.

As we can see, even if the chiasmus' terms present a PoS inversion, it is indeed the semantic aspect that gives the chiasmus its rhetorical salience.

To evaluate their model, Schneider et al. use the same strategy and the same implementation tools as Dubremetz and Nivre. Their results can be found in the figure C.7 (appendix C.5). At first glance, it is interesting to note that even with their additional features, Schneider et al.'s model does not manage to outperform Dubremetz and Nivre's on their own dataset when detecting antimetabole (though they do not underperform either), because their features do not really bring anything new to the model when all candidates were already extracted based on a lemma inversion. In this experiment, the average precision was calculated with the top 100 rated candidates (and not the top 200 like Dubremetz and Nivre used to do). However, on the new corpus consisting of four Schiller dramas, Dubremetz and Nivre's model is clearly outperformed, even for plain antimetabole detection: an average precision of only 21% is achieved whereas Schneider et al. manage to get up to 49%. That being said, it is impossible to know if the difference is due to the addition of the new features or to the fact that Dubremetz and Nivre's

model performs less well on a German corpus, since the four Schiller dramas used for the experiment are indeed in German.

The reasoning also holds for the article’s second experiment, which is about evaluating the models’ generalization. The results, reported in the figure C.8 (appendix C.5), show that the extraction method based on PoS inversion coupled with the new features described earlier enable a much better recall for finding chiasmi as well as antimetabole... in German corpora. Nonetheless, it is still worth noting that both approaches hold their ground when confronted with unseen texts, considering the number of salient instances found in the GerDraCor corpus when trained only on Schiller dramas. But since the annotation is based on the best model’s output, we still have no way of knowing if the GerDraCor corpus (or even the Schiller dramas, for that matter) contains many more undetected chiasmi, and thus of knowing how well both approaches *actually* perform.

From Chiasmus to Epanaphora and Epiphora

In their last published article, Dubremetz and Nivre (2018) [DN18] try to adapt their work on antimetabole to other repetitive rhetorical figures: the epanaphora (or anaphora) and the epiphora (or epistrophe). Similarly to the translation of ontologies for rhetorical figures in different languages (see Section 2.2.1), it is of particular interest to us to see whether all of our work done for the detection of chiasmi and antimetabole can be “translated” for other rhetorical figures, and to determine how easy and efficient it may be. For their experiment, Dubremetz and Nivre thus chose two schemes based on repetition rather than symmetry to try and exploit their findings on antimetabole. For the sake of simplification, we will only focus on their definition of epanaphora and epiphora:

Epanaphora is defined as the repetition of a word or a group of words at the beginning of successive sequences of language [...] In this paper, we limit the scope to epanaphora of sentences, exemplified in Example 1.

(1) **I am** an actor. **I am** a writer. **I am** a producer. **I am** a director. **I am** a magician.

At the opposite end, epiphora is the figure of speech of repetition at the end of a sequence (see Example 2).

(2) I’m so **gullible**. I’m so damn **gullible**. And I am so sick of me being **gullible**.

2 Background

Fortunately, the definitions of epanaphora and epiphora seem to be simpler and more consensual than those of chiasmi and antimetabole. Moreover, this simplicity is also reflected in the problem of their detection: in 2011, when antimetabole detection was tackled with PERL-compatible regular expressions (Hromada, 2011 [Hro11]), epanaphora was already being detected with Machine Learning techniques by Strommer (2011) [Str11], whose work inspired several ideas in Dubremetz and Nivre’s latest article.

As their chapter about chiasmus detection presents the same methodology and results as their precedent article, we will solely focus on their chapters about epanaphora and epiphora. Likewise, the ranking approach, the model and its evaluation method are essentially the same for the three figures. However, the features used for detecting antimetabole undoubtedly have to be different if we mean to detect different rhetorical features, even if they do present similarities.

Concerning the candidates extraction, only the repetition of a lemma at the beginning or end of successive sentences are considered. As can be expected, this condition is much stronger than the one for antimetabole extraction and results in much fewer candidates as well as a much higher concentration of salient instances, as can be seen in the figure C.9 (appendix C.6). These preliminary results also suggest that epanaphora is more frequent than epiphora (four times more candidates), while having a smaller concentration of salient instances. This is partially explained by the fact that half of the candidates are due to the presence of *The* at the beginning of sentences. Because of such others complications, the authors decided to eventually restrict their extraction of epanaphora to the repetition not of at least a lemma, but of at least two identical words, bringing their number of candidates to be comparable with that of epiphora. The number of antimetabole candidates, on another hand, completely dwarfs that of the two others (200 times more candidates).

Dubremetz and Nivre use the same features for detecting epanaphora and epiphora, for a total of eight features:

1. Sentence count: [...] the number of sentences exhibiting a repetition ;
2. Strong punctuation: [...] counts the number of sentences that end with a “strong” punctuation mark (! or ?) ;
3. Sentence length: [...] measures the average number of tokens per sentence in the sequence ;

2 Background

- 4/5. End [and start] similarity: counts the number of successive identical lemmas at the end [and beginning] of adjacent sentences, averaging over all such pairs in the sequence ;
6. End tag similarity: [...] analogous to the end similarity feature but looks at part-of-speech tags instead of lemmas ;
7. Same strict: [...] a binary feature that is 1 if the last word of the sentences in a sequence has the same form as well as the same lemma ;
8. Diff on end [or start] similarity: [...] it counts the number of identical lemmas at the end [or start] of sentences but then divides it by the number of lemmas that do not reappear in the other sentence.

We can observe that the first three features look very similar to some features used for antimetabole detection, and form the baseline used for the evaluation experiment; concerning the other features, they also seem to have been adapted from antimetabole detection to take into account the specifics of the two new figures, with the exception of the last one (“Diff on End” for epanaphora or “Diff on Start” for epiphora) that is truly specific to these two figures. Surprisingly (or not), it is the latter feature coupled with the baseline (though with the “Length” feature removed) that brings the best results for epanaphora, as can be seen in the figure C.10 (appendix C.6). This feature was also the most important one for epiphora, although by a much smaller margin: the best detection model for epiphora indeed used all the previously cited features, as can be seen in the figure C.11 (appendix C.6).

Furthermore, these results show that even though their work carried out on antimetabole did not prove to be entirely satisfying for related rhetorical figures, it still allowed the authors to efficiently build a strong baseline for future research. As can be seen in the figures C.12 and C.13 (appendix C.6), the difference in average precision between antimetabole and both epanaphora and epiphora remains considerable: 13 points more than the former and 23 points more than the latter. However, these results are comparable to those obtained for earlier detection of antimetabole: in 2015, Dubremetz and Nivre had achieved an average precision of 61% for antimetabole, compared to an average precision of 58% for epanaphora in this experiment.

2.2.3 Stanza

From 2016 to 2018, Dubremetz and Nivre ([DN16], [DN17], [DN18]) used the *Stanford CoreNLP toolkit* by Manning et al. (2014) [Man+14] in their extraction tool as a means of pre-processing their data. They used it for parsing, lemmatizing and getting Part-of-Speech tags from their raw text.

In 2020, the Stanford NLP Group released a new tool called *Stanza* (Qi et al., 2020 [Qi+20]), “an open-source Python natural language processing toolkit supporting 66 human languages”⁹. Essentially, *Stanza* has the same basic functionalities as the *Stanford CoreNLP toolkit* with better performances, adapted to more languages, and with further additional features. Therefore, we reasonably decided to use *Stanza* rather than the *Stanford CoreNLP toolkit* to develop our chiasmi candidates extraction tool. Let us introduce its features that will be of use later:

Tokenization and Sentence Splitting. When presented raw text, Stanza tokenizes it and groups tokens into sentences as the first step of processing. [...]

POS and Morphological Feature Tagging. For each word in a sentence, Stanza assigns it a part- of-speech (POS), and analyzes its universal morphological features (UFeats, e.g., singular/plural, 1st/2nd/3rd person, etc.). [...]

Lemmatization. Stanza also lemmatizes each word in a sentence to recover its canonical form (e.g., *did*→*do*). [...]

Dependency Parsing. Stanza parses each sentence for its syntactic structure, where each word in the sentence is assigned a syntactic head that is either another word in the sentence, or in the case of the root word, an artificial root symbol. [...]

The usage of Stanza is made easy thanks to its intuitive implementation and complete documentation¹⁰. To initialise a pipeline, it is not necessary to download the provided pre-trained models at every run, once is enough. A code extract is available in the appendix D.1. With only ten lines of code, it is possible to get an iterable object of *Words*

⁹Available at <https://github.com/stanfordnlp/stanza>, visited on 04/12/2022.

¹⁰Available at <https://stanfordnlp.github.io/stanza/index.html>, visited on 04/12/2022.

2 Background

containing a whole pre-processed text. The details of each processed word (lemma, PoS tag, syntactic dependency...) is then directly accessible through each *Word* object.

3 Methods

3.1 Tools and Project Management

The main part of this project was conducted using *our own and very special variant* of the Agile methodology (Beck et al., 2001 [Bec+01]). The developments were made in sprints of one week each, punctuated by general meetings where progress reports were made and the goals for the next sprint were defined. This allowed us to keep in touch regularly with our supervisors, seek their help and guidance when needed, and always work toward a concrete goal. It also pushed us to get to a stable version of our code each week, which reduced time losses in case of reverting to earlier working version.

Holding these weekly meetings also helped a lot mentally, to keep us focused and motivated, as it may be hard to do when working autonomously and remotely.

Concerning the software development phase of this project, the tools that were used are the following:

- GitHub for sharing the code and working collaboratively;
- Atom 1.63.1¹ as an Integrated Development Environment and Python 3.10.5 as the programming language;
- Doccano 1.8.0 and Docker 20.10.20 were used for annotating the dataset;
- Discord, emails and Zoom were used for communication and meetings.

The code can be found in the following GitHub repositories: ChiasmusExtractor for the extraction pipeline and AntimetaboleDetector for the ranking models.

Ultimately, this thesis itself was written using LaTeX (Overleaf).

¹It was announced that Atom would be sunset and all projects under the organization archived on December 15, 2022. Being an Atom user for many years, this is a tough goodbye.

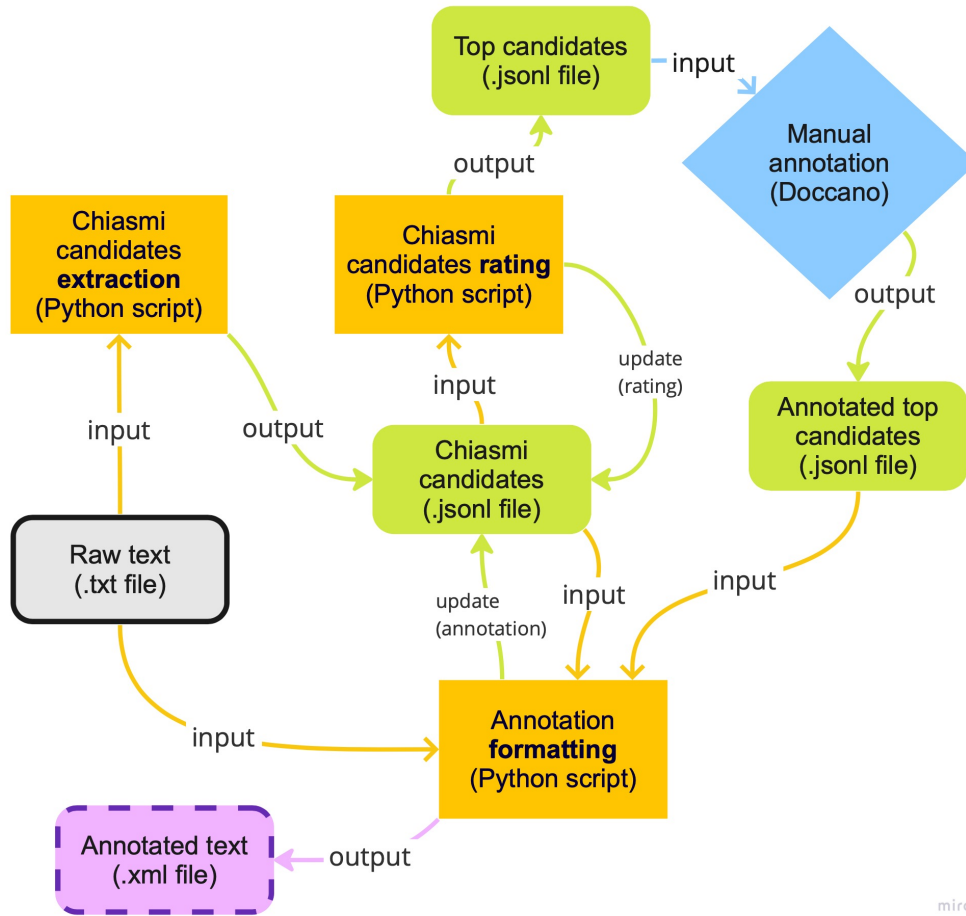


Figure 3.1: Entire chiasmi candidates extraction pipeline, beginning with a raw text file (gray module).

3.2 Automatic Extraction of Chiasmi

3.2.1 Use Case

As can be seen in the figure 3.1, our chiasmi candidates extraction pipeline is comprised of several steps, scripts and output files. Let us try to make sense out of it with a complete use case.

Everything starts with a .txt file containing a raw text, from which we wish to extract our chiasmi candidates. We feed it to the first script (*Chiasmi candidates extraction*), which will pre-process the text, parse through it, retrieve all the candidates it can find and place them in a JSON Lines file (*Chiasmi candidates*). Each line contains a candidate,

and each candidate contains a lot of data:

- the 30-token context in which the candidate was found (plain text),
- the positions of the candidate’s terms,
- an annotation indicating the candidate’s type (see section 3.3.2 for more details),
- the position of the 30-token context in the original file,
- the syntactic dependency of each word in the whole context,
- every word as is in the whole context,
- and the lemma of each word in the context.

Even if everything does not seem to be of use right now, it will be necessary for subsequent steps.

Once we have our full list of chiasmi candidates, we would like to annotate them; but first, since the number of candidates is generally much too substantial for a human annotator to handle manually, we have to pre-filter them. To that end, we feed our candidates file to the second script (*Chiasmi candidates rating*), which will assign a rating to each candidate based on a pre-trained predictive model. By doing so, it will update the candidates file by adding another entry to each JSON line: its candidate’s rating. The script will also generate another JSON Lines file (*Top candidates*) containing the best-rated candidates, up to a number that we can specify.

Once we have our file ready to be annotated, we may launch our annotation tool (*Manual annotation (Doccano)*) and feed it our filtered file. A glimpse at the resulting graphical user interface (GUI) used for the annotation can be found in the figure C.1 (Appendix C.1). Once annotation is finished, Doccano (Nakayama et al., 2018 [Nak+18]) will give back the same file as it was given as input with the annotation entry updated for each candidate.

Last, but not least, the third script (*Annotation formatting*) will take three files as input and perform two actions. First, it will use the candidates file and merge the annotated file into it, so that the final candidates file will contain all relevant information (rating and annotation). Second, it will use the original raw text and the candidates file to produce a final readable output (*Annotated text*) as an XML file. Its format is a variant of the specification by Harris et al. (2018) [Har+18]. An extract of such a generated file containing an instance of an antimetabole can be found in the figure C.2 (Appendix

C.1).

Eventually, the final outputs of this extraction pipeline are:

- the candidates file containing all relevant information for the subsequent detection part,
- and an XML file readable by a human containing all annotated chiasmi in the original text.

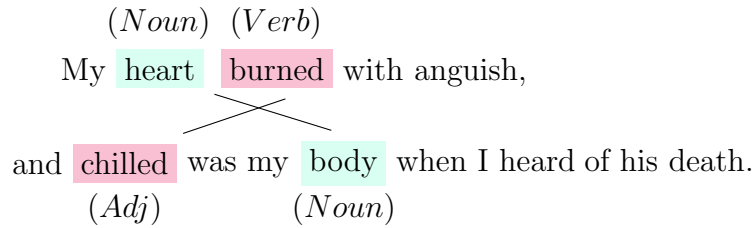
3.2.2 Choices

During this project, we made a lot of implementation choices - for instance, the definitions of a chiasmus and of an antimetabole presented in section 2.1, or the tool Stanza presented in Section 2.2.3. Let us review and explain the most important ones.

First of all, concerning the chiasmi and antimetabole definitions, we chose a looser one compared to previous research, as we decided to try and extract candidates with more than two pairs of terms. This choice was motivated by the fact that several definitions found during our research and prototypical examples of chiasmi pointed in that direction. As it can only allow us to detect more candidates, it was an attempt at improving the state of the art.

Out of the two latest approaches for extracting antimetabole and chiasmi, we chose to only keep Dubremetz and Nivre's (for antimetabole) but not Schneider et al.'s (for chiasmi). The former choice is motivated by the facts that we could not find any satisfying alternative than searching for inverse repetitions of lemmas to detect antimetabole, and that the present solution had been thoroughly researched and tested. On another hand, the latter choice was motivated by our finding conceptual flaws in the approach of searching for inverse repetitions of Part-of-Speech tags to extract chiasmi. Indeed, it does not make any practical sense with regard to our definition of a chiasmus, and through proper testing, we confirmed that it disregarded many prototypical antimetabole and chiasmi instances:

(Noun) (Verb)
 Never let a fool kiss you
 ×
 or a kiss fool you.
 (Noun) (Verb)



As we can see, both these examples do not present an inverse repetition of PoS tags. Since this intrinsically flawed approach discarded numerous instances of salient chiasmi, we decided to discard the approach. Instead, we experimented what seemed like a more logical approach with regard to our definition: we tried to extract candidates based on the semantic relations of their terms (either synonymous or antonymous), in a similar fashion as Dubremetz (2013) [Dub13].

Concerning the size limit of a chiasmus candidate, we chose to follow the example of past research and set it to 30 tokens.

Furthermore, in the light of Dubremetz and Nivre’s work, we decided to use stopwords not as a detection feature but as an extraction filter, since their feature did not allow them to detect any antimetabole with a stopword anyway. Doing that, we also removed some stopwords from Dubremetz and Nivre’s list to prevent our tool from missing a lot of salient chiasmi. Our motivation for this choice was to greatly reduce the number of uninteresting candidates, thus rendering annotation easier. However, this parameter can just as easily be removed.

Eventually, we chose to use Harris et al.’s (2018) [Har+18] scheme for producing a readable annotated output for the simple reasons that it was already thought through and thought out for supporting multiple types of rhetorical figures. By doing so, we hoped to make our work easier to improve and expand. Our only addition to this scheme is the introduction of an incremental numerical identifier for the instances of a same rhetorical figure throughout a document, e.g.:

```

<antimetabole-0>[...]</antimetabole-0>
[...]
<antimetabole-1>[...]</antimetabole-1>
[...]
<antimetabole-20>[...]</antimetabole-20>

```

3.2.3 Chronological Overview of the Implementation

When we first began to implement our extraction algorithm, we had no other choice than to start from scratch. What we considered as the state-of-the-art, i.e. Schneider et al.’s implementation, was not available to us, even after a direct contact with the main author. Dubremetz and Nivre’s code, on another hand, was available² along with a sample of their data. However, the code had not been updated since 2017, was written in Python 2, and we eventually estimated that it would have taken us more time to rework it than to write our own solution, so we went with the latter.

Our first steps were to learn how to use Stanza, which was luckily quite straightforward, and to implement our sliding window to iterate over the text. We wanted to start as simple as possible, and gradually upgrade our tool with additional features up to the final result presented in section 3.2.1. With that in mind, we decided to begin with extracting only antimetabole, and then to reuse the algorithm and adapt it for chiasmi. We thus began by coding the initialisation of our sliding window algorithm, in order to provide it with its 30 tokens. This first step was simpler than the main algorithm in the sense that it did not have to manage the window’s rear, and allowed us to debug efficiently. This step is depicted in Algorithm 1. The sliding window came soon after, and essentially consists of the same algorithm - with the difference that lemmas are removed from all tables when they reach the end of the window.

In order to make our extraction pipeline compatible with the annotation tool *Doccano*, we then had to work on the output of our algorithm to match the input format of Doccano’s. We settled for the .jsonl format because it is language-independent, highly common and well suited for our needs. As *Doccano* is an open-source tool, some features are not quite perfect yet: for instance, we had the surprise to be forced to name our JSON keys for the candidates annotation labels “cats”. And even if it did not require any implementation from our part, learning how to use Doccano and setting up our own annotation project was still a requirement for the pipeline. Luckily, the documentation³ was straightforward and very helpful for that matter.

Once the pipeline was more or less functional up to the annotation part, we needed a means to visualize the resulting data and to be able to share our annotated data easily, based on a common format. Having read Harris et al. (2018), we rapidly decided to use their XML annotation scheme for these purposes, and developed the third script that

²Available at <https://github.com/mardub1635/chiasmusDetector>, visited on 07/12/2022.

³Available at <https://doccano.github.io/doccano/>, visited on 07/12/2022.

Algorithm 1 candidates_extraction_initialisation(**file** *f*)

```

hash_table lemma_table;
hash_table match_table;
list candidates;
integer num_tokens  $\leftarrow$  0;

while not end-of-file of f and num_tokens  $\neq$  30 do
  read the next word w from f;
  get lemma from w;
  if not is_punctuation_or_stopword(w) then
    is_a_match, current_pairs  $\leftarrow$  search_lemma_in_window(lemma, lemma_table);
    if is_a_match then ▷ We found new pairs of matching lemmas
      old_matches  $\leftarrow$  search_old_matches(match_table, current_pairs); ▷ We
search for old pairs of matching lemmas inside the new pairs
      candidates  $\leftarrow$  update_candidates(current_pairs, old_matches, candidates);
      append current_pairs to match_table;
    end if
    append lemma to lemma_table
    num_tokens  $\leftarrow$  num_tokens + 1
  end if
end while

return candidates

```

allowed us to automatically produce such readable XML files directly from our annotated data.

Now that we were close to having a stable version of our antimetabole extraction tool, we tried to refactor and clean the code as much as we could, as well as fixing all bugs and abnormal behaviours we could find (such as candidates ending up as duplicates after extraction). Only then did we implement the last two main features of our tool: the extraction of chiasmi through embedding and the extraction of what we call *nested* antimetabole and chiasmi (see section 2.1.2).

Our algorithm for the extraction of chiasmi through embedding is not very different from our algorithm for antimetabole: instead of storing and comparing lemmas, we store embedding vectors provided by GloVe (Pennington et al., 2014 [PSM14]) and compare them with the *cosine_similarity* function provided by PyTorch (Paszke et al., 2019 [Pas+19]). Two embedding vectors (representing two words) are considered to be a match if their cosine similarity is above or below an empirically defined threshold. The newly found candidates are then treated the same way as the antimetabole candidates, all the way

to the extraction output.

Eventually, the extraction of nested chiasmi candidates is made possible quite easily thanks to the construction of our main extraction algorithm. In the algorithm 1, when a new pair of matching lemmas is found, the function to search for nested chiasmi (algorithm 2) is called and the possibly found nested chiasmi are appended to the candidates list.

Algorithm 2 `nested_candidates_search(word first_term, word second_term, list candidates_list)`

list *nested_candidates*;

for *candidate* in *candidates_list* **do**

if `start_position(candidate) > position(first_term)` and `end_position(candidate) < position(second_term)` **then**

 append *candidate* to *nested_candidates*

end if

end for

return *nested_candidates*

Afterwards, we received an update from the main author of Schneider et al. (2021) [Sch+21] with a link to a fresh public GitHub repository containing their code⁴. After some mandatory fixes and cleaning, we were able to adapt their implementation of a detection model to English, used to rate the extracted candidates. However, we also had to revise our extraction algorithm so that it could provide the detection model with additional data, required for its different features. That way, we were eventually able to retrieve the top candidates from our extracted list, which made annotation easier, even with a quite poor performance from the model (as the required training data was not provided).

Eventually, the last step was to enable our pre-filtered annotated candidates to be merged back with their non-filtered peers, to have a complete file containing all annotated candidates. Following past research (see Section 2.2), we may consider all candidates outside of our filtered pool to be rhetorically not salient. However, this practice implies to adapt the size of said pool to the size of the dataset that we want to annotate.

⁴Available at <https://github.com/cvjena/chiasmus-detector>, visited on 07/12/2022.

3.2.4 Limitations

In this section, we will discuss the limitations of our chiasmi (or rather antimetabole, since, as we will see just now, the chiasmi extracting part did not work out) candidates extraction pipeline, whether they result from our implementation or from external factors.

First of all, the automatic extraction of chiasmi through semantic relationships, with the help of word embeddings, did not provide satisfactory results. It is difficult to say if today’s embedding tools are not mature enough yet to permit such uses, or if we should, like Schneider et al. (2021), accept having a disproportionate number of chiasmi candidates compared to the state-of-the-art. Nevertheless, we were not able to find a compromise between a threshold low enough to allow us to detect semantically related terms of chiasmi while still high enough (or the reverse for antonyms) for it to avoid matching completely unrelated words. Therefore, our extraction tool did not succeed in extracting chiasmi, and our work eventually shifted its entire focus to antimetabole.

Secondly, our pre-processing tool *Stanza* also had a role to play in our extraction tool’s efficiency. Indeed, as the latter relies on the words’ lemmas to find matches, it is naturally limited by the lemmatizer’s efficiency, for which further insights will be given in the Chapter 4. As explained earlier, we also chose to use stopwords as a “hard filter” during extraction rather than as a detection feature, unlike Dubremetz and Nivre (2017). While this leads our tool to give much fewer uninteresting candidates, which makes annotation much easier, it also makes us miss salient antimetabole candidates (again, see chapter 4 for more insights). However, this limitation is mitigated by the fact that the stopwords filter can easily be removed if necessary.

The last two major limitations of this work actually concern the data annotation. The first one is inherited from past research and from the idea to use information retrieval methods. As explained, only the top candidates given by the detection algorithms are used to evaluate the predictive models, as well as for annotation. However, this may induce overfitting in the predictive models while potentially leaving out salient antimetabole annotated as the contrary if they do not appear in the top.

Lastly, introducing nested antimetabole in our extraction pipeline raises the same issue as for *binary* antimetabole: how can we establish which nested antimetabole candidates should actually be considered as nested, or as the stacking of an antimetabole and a mesodiplosis? Obviously, one could argue that *all* nested antimetabole *are* nested antimetabole. However, if we consider our candidates only for their rhetorical salience, the

dilemma remains far from trivial.

3.3 Data

3.3.1 Retrieval

Considering the very limited size of the available data to this day (31 annotated salient antimetabole from Dubremetz and Nivre and none from the others), we decided to try a completely different approach from the previous research and expand our dataset. Instead of analysing huge pieces of text chosen intuitively but without any guarantee of containing a decent number of antimetabole, which would take a serious amount of time to annotate while presenting the risk to end up with “another *The River War*” (a long book from Winston Churchill containing only one salient antimetabole, see Section 2.2.2).

Therefore, we decided to adopt a different strategy. If chiasmi and antimetabole were so rare, we might as well go straight for the light rather than keep searching in the dark. And that is why we ended up roaming the vast Internet to find as much salient instances of chiasmi and antimetabole as we could find. At the same time, thanks to one of our supervisors (Diana Nurbakova), we discovered the book “Never Let a Fool Kiss You or a Kiss Fool You” by Dr. Mardy Grothe [Gro02], a truly great source for chiasmi and antimetabole. The examples from it were manually extracted by her and added to our dataset.

In total, the book gave us around 543 instances of chiasmi and antimetabole, while our additional search for additional sources gave 243 more instances, which all amount to 788 examples. Our additional sources are very diverse, and although most of them are websites, some examples also come from books, songs, movies, series, or even real-life discussions.

However, such a diversity of sources and examples inevitably lead to the presence of duplicates in our dataset. An algorithm-based cleaning work allowed us to take care of the most obvious duplicates, but some could have been trickier to find, or even unclear as to whether they really are duplicates.

Eventually, in order to add more uninteresting instances and to attempt to avoid biases in our heavily imbalanced dataset, my project partner had the idea to take a random sample of the Corpus of Contemporary American English (COCA) dataset (Davies, 2008

[Dav08]). This way, around 12000 new instances, automatically annotated as negative, were added to the dataset.

3.3.2 Annotation

The annotation of our dataset was made easier, even if not less tedious, by the fact that we did not really have to *think* about whether an instance extracted by our pipeline actually *was* a salient instance or not, since our dataset was already fully comprised of salient instances. However, we still had to pass our dataset into the pipeline in order to be able to use Doccano and to have all the data our models would subsequently need. As hinted in the previous section, the only problem we still had to resolve was about the nested chiasmi and antimetabole: how would we handle them along with their “binary” equivalents (e.g. the ABBA and ACCA and BCCB in ABCCBA)⁵, and how would we decide which of the chiasmus with more terms or with less terms is the more salient one?

First of all, since we are not interested in chiasmi anymore (see Section 3.2.4), our categories for annotation only needed to take antimetabole into account. Thus the following categories were used:

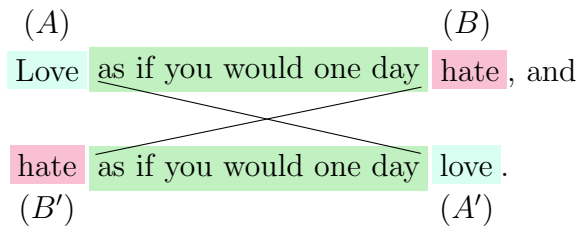
- Antimetabole
- Nested antimetabole
- Duplicate
- Not salient

The first two should be self-explanatory; as explained in Section 2, we do not agree with the distinction between “true chiasmi” and “false chiasmi” in the past research. We rather separate “rhetorically salient” chiasmi from “not salient” chiasmi, thus the last category. The *Duplicate* category was inspired from Dubremetz and Nivre (2015, 2016, 2017) but remains different from theirs. We use it to answer the questions formulated in the previous paragraph, i.e. to handle nested antimetabole: when a nested antimetabole is annotated as such, the corresponding binary antimetabole is marked as a *Duplicate*

⁵A passionate reader could be tempted to calculate how many “sub-chiasmi”, or “chiasmi of lesser order”, are covered by a chiasmus with n pairs of terms. The answer we found ourselves is the following (subtract 1 to avoid counting the chiasmus with n terms): $\sum_{k=0}^{n-2} [(n-k) * (n-k-1)/2]$. For $n = 2, 3, 4, 5$, it respectively gives 1, 4, 10, 20 chiasmi.

- only the most exterior one is considered and all other are discarded. This choice is motivated by the idea that in an ABCCBA or even ABCDDCBA pattern, the most interesting binary antimetabole is the ABBA one, since the C and eventual D pairs remain inside its boundaries, and often (if not always) act as a mesodiplosis.

However, as hinted previously, we do not consider all nested antimetabole as salient - or, at least, more salient than their binary counterparts. We left this choice to the annotator's discretion, though with two simple rules to help us. First, any nested antimetabole that would be too hard to annotate has to be considered as less salient, for simplicity's sake. Let us take a clear example:



As we can see, the binary antimetabole is very easy to spot (ABB'A'). On the contrary, the perfect mesodiplosis is comprised of six words, from whose it is impossible to choose which is the correct one to form a nested antimetabole. In these cases, we thus mark the nested ones as *Not Salient* and the exterior binary one as *Antimetabole*.

Second, we chose to disregard nested antimetabole with very common mesodiplosis, and especially whose central term is a stopword (such as “without”, which is extremely frequent).

3.4 Detection of Antimetabole's Saliency

Once that we managed to get a sturdy antimetabole extraction pipeline with a fully annotated dataset, we could finally get to the last part: actually detecting our salient antimetabole.

3.4.1 Baselines

For this experiment, we will use the current state-of-the-art as our baselines. Therefore, our two baselines will be:

1. Dubremetz and Nivre (2017), with a logistic regression model and all features described in Section 2.2.2.
2. Schneider et al. (2021), with a logistic regression model and all features described in Section 2.2.2.

Since Schneider et al.’s algorithm uses Dubremetz and Nivre (2017) as a baseline itself, we could have limited ourselves to one baseline. However, since both approaches remain different on several crucial aspects (e.g. detection on antimetabole against chiasm, made for English against German), we still deem it relevant and interesting to compare both on a larger dataset.

The implementation for both models was helped by Schneider et al.’s implementation, but still had to be reworked to match our data specifications, along with minor fixes. Moreover, all features from Dubremetz and Nivre and Schneider et al. were reworked a second time to adapt to the detection of nested antimetabole. For their binary antimetabole models, the instances considered as *true* will be annotated with *Antimetabole* or *Duplicate*, as specified in Section 3.3.2.

3.4.2 Models

Several models will be compared against the baselines. First of all, two modest attempts at directly improving the state-of-the-art, which we will call *improved Dubremetz* and *improved Schneider*, will be tried against their original implementations (respectively *Dubremetz* and *Schneider*).

Secondly, like Dubremetz and Nivre (2017) did to some extent, we will implement different types of models with the same features to compare their performances and the results of their training. The four different models were chosen for their interpretability and are:

- A classic logistic regression classifier, to be directly compared with past research;
- A Support-Vector Machine (SVM) with a Radial Basis Function (RBF) kernel (like Dubremetz and Nivre, 2017);
- A regression tree;
- A random forest.

The two first models are thus inspired from the state of the art, while the two others are an additional experiment. Even though the random forest is expected to perform better than the “simpler” regression tree, the latter is still included for its interesting interpretative properties.

All models are implemented using scikit-learn (Pedregosa et al., 2011 [Ped+11]).

3.4.3 Features

In order to make our models competitive against the baselines, we introduce four new features in addition to those of the state-of-the-art. The first two were inspired by Harris and Di Marco (2017) [HD17]:

20. [T]ous pour un, un pour tous. ([Dum49]:129)
All for one, one for all. ([Dum10]:80)

[...] Example 20, in particular, features not just the central member of the chiasmic suite, antimetabole, but also isocolon (prosodic repetition), parison (syntactic-structure repetition) and mesodiplosis (as above, medial lexical repetition). [...]

Figures of parallelism, in particular, like isocolon and parison, seem to have a special role in augmenting other figures [Fah03] [...]

Hence the two following features:

1. Detection of a parison stacking with the antimetabole (with Part-of-Speech tags).
2. Detection of an isocolon stacking with the antimetabole (with the *prosodic* Python library).

In order to shed lights over these two mysterious new rhetorical figures, Harris and Di Marco (2017) give us the following definition:

Parison: A scheme of syntactic repetition, often referred to as syntactic parallelism: the proximal repetition of the same syntactic pattern. (“My life is spent alone, without wealth, without status, without love, and without hope” ([Cix06]: p. 354.)

For isocolon, though, we will have once again to call Greene (2012, p.734) [Gre+12] for help with his double entry about isocolon and parison:

ISOCOLON AND PARISON. *Parison* (Gr., “almost equal”) describes syntactic members (phrases, clauses, sentences or lines of verse) showing parallelism of structure. In short, they are identical in grammar or form. *Isocolon* (Gr., “equal length”) denotes members that are identical in number of syllables or in scansion. [...]

Isocolon is particularly of interest because Aristotle mentions it in the *Rhetoric* as the figure that produces symmetry and balance in speech and, thus, creates rhythmical prose or even measures in verse; cf. Quintilian 9.3.76.

He gives the following example for isocolon:

“Was ever woman in this humour woo’d? / Was ever woman in this humour won?” (*Richard III* 1.2.227).

The last two novel features are partially inspired from already existing ones, and from experimental observations made during the retrieval of our dataset:

3. Detection of nominal groups attached to the antimetabole terms (terms underlined, **nominal group** in bold):

“Some have an idea that the reason we in this country discard things so readily is because **we have so much**. The facts are exactly opposite – the reason **we have so much** is simply because we discard things so readily.”

In fact, one could argue that this feature only captures the simplification made during the extraction of antimetabole, which reduces the antimetabole’s terms to only one word.

4. Detection of repetitions before or after the antimetabole (terms underlined, **repetition** in bold):

“My job is not **to represent** Washington to you, but **to represent** you to Washington.”

“Being deeply loved by someone **gives you** strength, while loving someone deeply gives you courage.”

4 Results

4.1 Extraction Pipeline Evaluation

Unfortunately, we cannot directly compare our antimetabole extraction pipeline to the state-of-the-art. Indeed, the method of Schneider et al. (2021) [Sch+21] is much too different, and Dubremetz and Nivre’s (2017) [DN17] implementation, as explained previously, is not easily available for comparison. In a sense, our extraction pipeline is the first one to be fully implemented, working and specialized for antimetabole.

However, we can still evaluate its performances on our dataset and understand its limitations to facilitate further improvements left for future work.

As can be seen in the Table 4.1, our dataset is comprised of 680 antimetabole in total (a very attentive reader may have noticed the discrepancy with the total number of 788 chiasmi announced in the chapter 3 - this is due to one instance in Latin that was left out, being impossible to classify). The 107 chiasmi instances, whether they could have been possible or not to extract with our embedding method (implied and wordplay fall into the latter category), do not interest us.

As can be seen in the Table 4.2, the recall of our extraction pipeline is estimated to 86%. That being said, we also mentioned that the stopwords filter can easily be switched off if necessary, bringing the recall up to 90%. One small improvement could be the expansion of the window size, fixed to 30 tokens as per previous research, which could

Table 4.1: Composition of the dataset

Antimetabole	Implied chiasmi or wordplay	Chiasmi	Total
680	50	57	787

Table 4.2: Results of the extraction pipeline for antimetabole

Extracted	Unextracted		
	Lemmas	Stopwords	Window size
585	60	24	11
86%	8.8%	3.5%	1.6%

add another 1% to our recall, but the main hurdle remains the lemmatizer. Because of its performances, we lose almost 9% of our antimetabole.

4.2 Models Evaluation and Comparison

4.2.1 General Presentation

In order to make things clearer as well as simpler, we have to make a few considerations before diving into this section. First of all, we will only focus on the most popular type of model in previous research, so that direct comparisons with the baselines will be possible. Thus, all of our models will be logistic regression classifiers. Implemented with scikit-learn (see chapter 3), we chose the L2 penalty term for training because of the relatively small number of our features, leading us to not need the L1 penalty for feature selection with many zero coefficients. Moreover, we chose the *liblinear* optimisation algorithm because it is supposed to be ideal for small datasets.

In order to evaluate our models, we have them score all of our candidates in the test sets, and follow previous research by computing the average precision of the ranked set. Because of our small dataset and the will to prevent overfitting as well as getting more reliable results, we chose to train and test the models with 5-fold cross-validation, just like Schneider et al. (2021). Following the previous section, we then have 472 positive instances for training, and 118 positive instances in each testing set.

Where Dubremetz and Nivre (2017) had less than 50 positive instances to evaluate their models on, and chose to compute their average precision on the first 200 candidates according to their ranking scores, we logically chose to expand that range due to our greater number of positive instances. The experimental number of a top 400 was chosen accordingly. The average precision is thus computed as the average of the 5 average

precisions, computed for each fold. To calculate the recall, we simply look at how much of the 118 positive instances were indeed found in this top 400. Due to the nature of our task, these two metrics will solely be used throughout this chapter.

4.2.2 Baselines

We can expect from the Chapter 2 to have at least two baselines (from Dubremetz and Nivre (2017) and Schneider et al. (2021)). However, the situation is even a bit more complicated than that, since these two baselines were tampered with and improved. Hence the following distinction:

- *Dubremetz* and *Schneider* will be the simple, raw baselines;
- *Dubremetz nested* and *Schneider nested* will be the baselines whose features were adapted to take better advantage of the additional information given by the nested antimetabole;
- *Improved Dubremetz* and *Improved Schneider* are straight improvements of the baselines, following ideas of our own to gain possibly better results without changing anything fundamental.

The results of this first experiment are reported in the Table 4.3. Following the results from Schneider et al. (2021) concerning antimetabole, we expected both models to perform on an even scale: even if *Schneider* only adds features to *Dubremetz*, these features were found to be effective for chiasmi but not for antimetabole. Seeing how small the differences are between *Dubremetz* and *Schneider* on our dataset, we may confirm this result. However, we can also notice that the two improvements (taking the nested antimetabole into account and our own improvements) indeed increase significantly the average precision compared to the baselines. *Nested Dubremetz* clearly outperforms its

Table 4.3: Results of the baselines evaluation

Metric	Simple		Nested		Improved	
	Dub.	Schn.	Dub.	Schn.	Dub.	Schn.
Ave. Precision	68.0%	69.0%	74.3%	72.3%	77.5%	77.5%
Recall	96.3%	96.4%	94.9%	96.4%	94.6%	95.4%

baseline by 6%, although its recall decreases, while *Nested Schneider* keeps a stable recall but increases its average precision by 3%. The *Improved* versions, on another hand, both decrease their recall but improve greatly the average precision (+9.5% and +8.5%), ending with exactly similar results.

In an additional experiment, we tried to combine both *Improved* versions, since the *Schneider* models are based on the *Dubremetz* baseline. Thus, combining the *Improved Dubremetz* with the *Improved Schneider* gave an average precision of 77.5% and a recall of 94.7%. In fact, these results match perfectly *Improved Dubremetz*.

In conclusion, what we can consider the best version of the baselines is *Improved Dubremetz*, since it achieves the same performances as when augmented with *Schneider* features, but with a simpler model.

4.2.3 Novel Features

In order to test the four new features presented in Section 3.4.3, we first conducted an experiment by adding each feature to the baseline to see what effect it had on the results. The results of this experiment can be found in the Table 4.4. At first glance, the recall seems to be very stable and close to the baseline (between 95.9% and 96.4%) for features based on *Dubremetz*, and the average precision only significantly increases when the parison feature is added - and by 7%. Isocolon and repetitions do not seem to benefit the model at all, but do not damage its performances either; nominal groups, on another hand, do make the average precision go up by 1%. However, when we add our features to *Improved Dubremetz*, whose average precision was already 9% higher than *Dubremetz*, the increase is greatly mitigated: from 7% to 0.6% for parison, and comparable results for the three other features. Concerning recall, it is systematically

Table 4.4: Results of the standalone features evaluation

Metric	+ Parison		+ Isocolon		+ Nominal gr.		+ Repetitions	
	Dub.	Impr. Dub.	Dub.	Impr. Dub.	Dub.	Impr. Dub.	Dub.	Impr. Dub.
Ave. Precision	75.1%	78.1%	68.0%	77.6%	69.1%	78.0%	68.2%	77.7%
Recall	95.9%	95.4%	96.3%	94.7%	95.9%	94.2%	96.4%	94.4%

Table 4.5: Results of the features gradation evaluation

Metric	+ Parison		+ Isocolon		+ Nominal gr.		+ Repetitions	
	Dub.	Impr. Dub.	Dub.	Impr. Dub.	Dub.	Impr. Dub.	Dub.	Impr. Dub.
Ave. Precision	75.1%	78.1%	75.3%	78.5%	75.7%	78.5%	76.5%	79.1%
Recall	95.9%	95.4%	95.9%	95.4%	95.9%	95.8%	95.9%	95.8%

lower (between 1.5 and 2%) when we combine our new features with *Improved Dubremetz* than with *Dubremetz*, suggesting a trade-off for the improved average precision.

By doing the same experiment with *Schneider* as a baseline rather than *Dubremetz*, we found comparable increases in average precision, with a slight increase in recall.

To confirm these results, we also conducted a gradation experiment, by adding one feature after the other and stacking them, instead of adding them only one by one on their own. The results of this experiment are shown in the Table 4.5. Interestingly, we do not get the exact same results as in the previous experiment. Concerning *Dubremetz*, although adding the parison feature still improves the average precision by 7.1%, adding the isocolon on top of it gives 0.2% more: even if it is very small, it is still better than having no effect at all. The nominal groups, however, now only add 0.4%, but the repetitions add yet another 0.8%, compared to the 0.2% in the standalone experiment. It is also worth noting that even with all these new features, improving their baseline by 8.5%, the baseline *Improved Dubremetz* still outperforms it by 1% (even though we still gain 1.3% in recall).

The gradation based on *Improved Dubremetz* yield comparable results, although the nominal groups now do not improve the average precision at all. On another hand, the isocolon and the repetitions bring similar improvements, for a total of +1.6%, since the parison only improved it by 0.6%.

Adding features does not worsen the recall, although it does not improve it greatly either (not at all for the *Dubremetz*-based features and by 0.4% for the *Improved Dubremetz*-based ones).

Once again, we conducted the same experiment with features whose baselines are *Schneider* and *Improved Schneider*. Compared to the gradation for *Dubremetz*, the one for *Schneider* still yields comparable results, although with a smaller improvement with the parison features and slightly better improvements for the other three, for a final

Table 4.6: Results of the features ablation experiment

Metric	- Parison		- Isocolon		- Nominal gr.		- Repetitions	
	Dub.	Impr. Dub.	Dub.	Impr. Dub.	Dub.	Impr. Dub.	Dub.	Impr. Dub.
Ave. Precision	-7.2%	-0.7%	-0.2%	-0.0%	-0.2%	+0.1%	-0.7%	-0.6%
Recall	+0.5%	-1%	+0.2%	-0.5%	-0.3%	-0.5%	-0.0%	-0.0%

average precision of 77.8% (+8.8% compared to the baseline) with all features. The experiment for *Improved Schneider* gives almost the exact same results as for *Improved Dubremetz*, with a final average precision of 79.3% (+1.8% compared to the baseline) with all features. Finally, combining *Improved Schneider* with a base consisting of *Improved Dubremetz* does not improve it at all.

Last but not least, we finally conducted an ablation experiment on the models combining all new features, to see which ones improved the most the whole model. The results of this experiment are shown in the Table 4.6, and are really not surprising. We confirm that the parison feature has the most impact on the average precision, and that the isocolon has very little of no impact at all. The repetitions features seem to perform better when combined with the others, contrarily to the nominal groups, which even worsen a tiny bit the performances of the whole model based on *Improved Dubremetz*. On another hand, the variations in the recall measures seem to be consistent with the standalone features experiment for both approaches.

Compared to the ablation experiment for features combined with *Schneider* and *Improved Schneider*, the former seems to give more importance to the nominal groups features compared to *Dubremetz*, and the latter gives more importance to all features compared to *Improved Dubremetz*.

4.2.4 Different Types of Models

Now that we gained a lot of insights on our baselines and features thanks to the logistic regression classifiers, we can try different types of models and compare them to see which one performs best. To that end, we will evaluate four types of models with several experiments:

Table 4.7: Results of the baselines evaluation for different model types

Model type	Metric	Simple Dub.	Nested Dub.	Improved Dub.
Log.	Ave. Precision	68.0%	74.3%	77.5%
Reg.	Recall	96.3%	94.9%	94.6%
SVM	Ave. Precision	65.4%	68.8%	77.0%
RBF	Recall	92.9%	89.8%	92.9%
Reg.	Ave. Precision	66.8%	89.4%	65.0%
Tree	Recall	21.4%	12.4%	8.8%
Random	Ave. Precision	60.9%	71.7%	67.5%
Forest	Recall	77.8%	75.2%	81.9%

- A logistic regression classifier (log. reg.);
- A Support Vector Machine with a RBF kernel (SVM RBF);
- A regression tree (reg. tree);
- A random forest (rand. forest).

First of all, let us compare our baselines with the Table 4.7. The first obvious result is that the logistic regression model seems to outperform all others on both recall and average precision, by a small margin for the SVM and by a much larger one for the regression tree, as well as the random forest. Another outstanding result is the poor performance of the regression tree, even if it achieves an average precision of 89.4% for *Nested Dubremetz*: with a recall of only 12.4%, the average precision does actually not mean much, if anything at all. The random forest achieves a much better recall than the regression tree, and also satisfying average precisions, but both still cannot compare to the logistic regressor and the SVM: even with a recall lower by 15% to 20%, its average precision remains much lower for *Dubremetz* and *Improved Dubremetz*. Just like the regression tree, it seemingly outperforms the SVM for *Nested Dubremetz* (though still not the logistic regressor) in terms of average precision, but once again, its recall lower by 15 points does not allow us to jump to any conclusion.

It is also worth noting that for the three main models (excluding the regression tree), the performance gap is narrowing for *Improved Dubremetz*: the SVM has an average precision only 0.5% lower than the logistic regressor and a recall only 1.7% behind, for

Table 4.8: Results of the standalone features evaluation for different model types

Model type	Metric	+ Pari-son	+ Iso-colon	+ Nomi-nal gr.	+ Repeti-tions
Log.	Ave. Precision	75.1%	68.0%	69.1%	68.2%
Reg.	Recall	95.9%	96.3%	95.9%	96.4%
SVM	Ave. Precision	78.0%	64.1%	66.5%	63.3%
RBF	Recall	90.8%	92.4%	92.7%	91.7%
Reg.	Ave. Precision	74.7%	68.1%	71.2%	71.7%
Tree	Recall	41.9%	15.3%	24.1%	9.7%
Random	Ave. Precision	72.5%	60.5%	63.7%	60.7%
Forest	Recall	80.7%	76.1%	72.0%	70.5%

a gap of 5.5% and 5.1% respectively when it comes to *Nested Dubremetz*. The random forest, on another hand, remains 10 points lower for its average precision, but “only” 11 to 13 points behind the two first models in terms of recall. At the same time, the regression tree performs worst with *Improved Dubremetz*, with a recall and an average precision lower than ever (to get an idea, its recall of 8.8% means that it scored only 10 salient instances out of 118 in its top 400).

For the sake of simplicity, since we now have four different models to evaluate, the standalone features experiment will only be conducted on the basis of *Dubremetz*, for which the novel features had most impact with the logistic regression classifier. We thus hope to be able to see a similarly great impact on the performances of the other models if we choose a weaker baseline (as compared to *Improved Dubremetz*), which should be more interesting to analyse. The results of this new experiment are reported in the Table 4.8.

First of all (since we already presented the logistic regression model in the Section 4.2.3), let us be impressed by the improvement induced by the introduction of the parison feature. Admittedly, we expected it to work well since it was the most effective one for the logistic regression, but its effects are even more impressive with the SVM, with an increase of 12.6% in average precision. However, this result may be slightly mitigated by the associated decrease of 2.1% in recall. Furthermore, it truly works wonders for the regression tree and random forest. The poor regression tree with its ridiculous recall now sees it jump to 41.9%, almost doubling it with just one additional feature, while increasing its average precision by 7.9% ! Eventually, the random forest also presents a

stunning increase in average precision (+11.6%), but contrarily to the SVM, its recall also increases by 2.9%. So far, this new feature undoubtedly seems to perform much better than the other three ones, but let us hear what these have to say.

The isocolon that was already a poor student in our preliminary study with the logistic regressor, since its addition did not improve the model’s performances one bit; however, for our new models, the isocolon... is even worse. Unsatisfied with doing nothing for the logistic regression, it now damages the SVM’s performances by decreasing its average precision (by 1.3%) **and** its recall (by 0.5%). For the regression tree, which already had a very low recall, it further lowers it by one quarter, which tempers its slight increase in average precision. The random forest suffers the same fate as the SVM and sees both its average precision and recall decrease (by 0.4% and 1.7% respectively). Therefore, where the parison was already hinted to be the most promising feature, the isocolon was hinted to be the worst... and still does nothing to convince us otherwise. However, the isocolon does not stand alone in the corner of shame. Not unlike it, the repetitions feature did not improve the logistic regression performances when added alone; and for the other three models, its effect is the exact same as the isocolon (understand: just as bad), if not even worse.

Hopefully, our last “nominal groups” feature will perform better than its two neighbours, as can be expected from the slight improvement it gave the logistic regressor. True to its promises, it indeed improves our SVM’s average precision, if only by 1.1%, while the recall practically does not vary (-0.2%). The effect is more visible on the regression tree, whose average precision rises by almost 5 points, while its recall also increases, but only by 2.7% (and thus remains... awful). The random forest finally shows modest results, as the improvement in average precision (+2.8%) is once again mitigated by its recall deterioration (-5.8%).

Eventually, even if the parison seems much more useful than the other features, all hope is not lost: even if a feature performs poorly on its own, it can always improve a model when combined with others. To confirm this, we thus run a final feature ablation study, whose results are reported in the Table 4.9.

Without any surprise, this experiment proves (if it was still needed at this point) that the parison feature has the most positive impact, whatever the model and the features already used. Although it worsens the logistic regressor’s recall by 0.5%, it improves the same model’s average precision by 7.2% and simply improves everything for all other models. It is most notable for the regression tree, for which it improves the average precision by 13.7% as well as the recall by 29.2% !

Table 4.9: Results of the features ablation study for different model types

Model type	Metric	All features	- Parison	- Iso-colon	- Nominal gr.	- Repetitions
Log. Reg.	Ave. Precision	76.5%	-7.2%	-0.2%	-0.2%	-0.7%
	Recall	95.9%	+0.5%	+0.2%	-0.3%	-0.0%
SVM	Ave. Precision	75.9%	-12.8%	+0.9%	-0.0%	+1.3%
RBF	Recall	92.9%	-0.2%	-1.4%	-0.9%	-1.0%
Reg. Tree	Ave. Precision	78.8%	-13.7%	-0.3%	-5.3%	-5.5%
	Recall	41.4%	-29.2%	+1.1%	-7.7%	-0.7%
Random Forest	Ave. Precision	73.5%	-11.4%	+0.5%	-0.3%	-0.4%
	Recall	79.2%	-2.3%	-1.4%	-0.2%	+1.6%

We also expected the nominal groups to perform correctly, which this study confirms: it is the only feature that improves all models on both their metrics (maybe with the exception of the SVM’s average precision, but an absence of degradation is still an improvement), even if by more modest margins.

Obviously, the results are a bit more complicated for the two resulting features (and “bad students”): the isocolon and the repetitions. Indeed, the isocolon has the interesting effect to improve each model’s average precision or recall only by undermining the other. In that sense, it is difficult to say whether it has a positive, negative or neutral impact; even measuring the difference between the positive and negative impact does not help much, since this difference also varies from one model to another. At least, the repetitions feature offers a somewhat clearer result: it has an undeniably positive impact on the performances of the logistic regression classifier and of the regression tree, but an evenhanded if not negative impact for the SVM and the random forest.

In conclusion, it would seem that the parison and the nominal groups features are very effective, that the repetitions feature is to handle with care and maybe to fine-tune to make it interesting, while the isocolon feature is an utter disappointment.

5 Discussion

5.1 Interpretation of the Baselines Results

5.1.1 The Dubremetz Baseline

In the Section 4.2.2, it was made plainly clear that the modified versions of the *Dubremetz* baseline features, i.e. *Nested Dubremetz* and *Improved Dubremetz*, improved substantially the model performances. Let us attempt at explaining these discrepancies.

First of all, the differences between *Dubremetz* and *Nested Dubremetz* are quite simple: the latter is merely an adapted version of the former to take into account all specificities of antimetabole candidates annotated as *Nested*, where the original *Dubremetz* only computes its feature values based on binary antimetabole, composed of the most exterior terms of a nested antimetabole. Therefore, as the *Nested* version is obviously better able to discriminate and rate the nested candidates, which add up to a bit more than one quarter (27.5%) of the whole dataset, and a bit less than one quarter (22.5%) of the candidates annotated as salient antimetabole. These numbers being far from negligible, it is easily understandable that analysing the nested candidates more in detail allows for a much better average precision.

On another hand, while the *Improved Dubremetz* was expected to perform better than its simpler counterpart for similar reasons, it was less obvious to expect such improvements with regard to the model's performances. Indeed, even if the features are adapted to be able to conduct a finer analysis, they are not computed with more data like the *Nested Dubremetz* did: to allow for a fair comparison, this version was built upon *Dubremetz* rather than *Nested Dubremetz* with only the exterior terms of the nested candidates being kept. For starters, we improved the lists used for detecting conjunctions, negation and punctuation in the candidates as follows:

- The “conjunction list” was extended with five new conjunctions and adverbs: “although”, “before”, “once”, “though” and “while”. This extension was not based on any statistical study of the data, to avoid any bias, and was rather a continuation of the idea expressed by Dubremetz and Nivre (2015) [DN15] for their first implementation of the feature, which was supposed to “underline the axial symmetry” of the antimetabole.
- The “negation list” was extended with two new words: “neither” and “none”. We simply considered that these two words expressed as much a negation as the ones originally implemented by Dubremetz and that there was no reason to not add them to the list.
- Inspired by the implementation of Schneider et al. (2021) [Sch+21], we extended the “hard punctuation list” with six new signs: “-”, “_”, “{”, “}”, “'” and “ ””. The thought process that led to this improvement was similar to the one for the negation list.

With this, the new model was supposed to only perform better, as the features were plainly improved to integrate more possibilities than before. However, the following modifications may be more subtle or subject to discussion, even though we also consider them to be improvements:

- The “centralPunct” feature, which captured the “number of hard punctuation marks and parentheses in C_{bb} ” (C_{bb} designating the central part of the antimetabole between the B term and the B' term) as explained in Dubremetz and Nivre (2015), was duplicated to capture not only the hard punctuation marks, but also the soft punctuation marks. Indeed, the former was thought to allow the model to discriminate more easily “false positives” from salient antimetabole, as it was hypothesized that hard punctuation marks should not (or rarely) appear in the middle of an antimetabole. By joining this idea with the one behind the “conjunction list” feature, we emitted the hypothesis that soft punctuation marks (i.e. commas) could become an indicator of an interesting candidate by “underlining the symmetry” just like a conjunction.
- We modified one of the “similarity” feature (exactMatch) to match it with two other features of the same category (sameTok and simScore). Originally, the feature is “true if C_{ab} and C_{ba} are identical” (C_{ab} and C_{ba} being composed of the

words between each of the antimetabole pairs of terms $A-B$ and $B'-A'$), but said identity is based on the exact words in the sentence. We switched it to lemmas, with the idea to capture more examples while keeping the same fundamental idea, which is in fact to detect a mesodiplosis stacking with our antimetabole.

- Eventually, we extended the “hasConj” feature beyond its list by adding two features to capture the presence of conjunctions (even if the list is now more heterogeneous) not only in the central part of the antimetabole, but also in C_{ab} and C_{ba} . While the former is trying to capture an axial symmetry, the latter rather tries to capture an axial dissymmetry.

Even if we could do it, we will not conduct any further experiment to try to determine which combination of these improvements and additions gives the best performances: we are satisfied enough to confirm our intuition that the modifications as a whole indeed improved the model’s performances, and trying to find the ideal combination, down to which additions to the various lists actually improve the ratings and which do not, would very probably end up with our model overfitting on our current dataset. Even if we managed to substantially increase the size of our pool of antimetabole, the number of salient instances remains objectively quite low for such a fine-grained optimisation.

5.1.2 The Schneider Baseline

From Schneider et al (2021), we expected the models based on the *Schneider* baseline to perform as well as the *Dubremetz* ones, since their article showed no improvement with regard to the detection of antimetabole. This result was somewhat confirmed by the very small difference in performance with the addition of the *Schneider features* in the section 4.2.2: 1% in average precision and 0.1% in recall. However, we can notice that the *Nested* version for *Dubremetz* logically performs better, since the *Schneider features* benefit less from the additional data for the nested candidates compared to the *Dubremetz* ones, which are much more complete. It is still very interesting to watch how both *Improved Dubremetz* and *Improved Schneider* end up coincidentally having the exact same average precision (although the latter presents a slightly better recall). Just like the *Nested* version, we could have expected the *Dubremetz features* to benefit more from their improvements (which are also more numerous). Indeed, the modifications for the *Schneider features* were not as straightforward as some for the *Dubremetz* ones, and

were even sometimes aimed at simplifying the features rather than expanding them. With regard to the embedding features (even if they are not supposed to matter much when it comes to antimetabole detection), Schneider et al. (2021) took into account the cosine distance between the embedding vectors of all pairs of terms in the antimetabole (six in total for a simple ABB'A' pattern). As we did not deem it useful to compare the embedding vectors of non-matching pairs, we restricted these computations to the matching pairs only (thus from six to only two). In addition to that, in order to prevent any data bias concerning the importance of both pairs, we average the distances for each candidate, resulting in only one value for this feature.

Following the same idea, the lexical features suffered a similar fate and were restricted to only the matching pairs of terms in the antimetabole, and were as well averaged over one value. It is also worth noting that, contrarily to *Improved Dubremetz*, the modifications made to the *Schneider* features are based on its *Nested* version, since they are much simpler.

All that being said, as explained by Schneider et al. (2021), the lexical features do not bring anything useful for the detection of antimetabole; thus, the difference in performance between *Nested Schneider* and *Improved Schneider* should only be due to the improvements made to the embedding features. Although the simplification of the lexical features may have had a positive impact as well, since they do not bring any new information. Indeed, if we remember that a logistic regression model trains not only weights for its features but also models the interactions between them, the fewer features, the less irrelevant interactions the model has to be trained with. And, in the end, the *Improved Schneider* model was only able to compete with *Improved Dubremetz* because it was built upon the exploitation of nested candidates information.

5.1.3 All model types

As far as the baselines models are concerned, the relative performances depicted in the Section 4.2.4 were unsurprising for several reasons. First of all, we anticipated the comparable results of the logistic regression model and of the Support-Vector Machine thanks to Dubremetz and Nivre (2017), who reported getting similar results with both models, if not slightly better ones with the logistic regression. We are at least able to confirm these results on a larger dataset. It can be explained through several factors: an SVM is less suited for a pure regression task than a logistic regressor, and even conceptually, separating a space of antimetabole between salient and not salient instances can

be a very tricky assignment, as data points can be very close while having a different annotation.

We also expected the regression tree to perform worse than all other models, even if maybe not to such an extent; indeed, the model itself is intrinsically much simpler, while much harder to optimise (“constructing optimal binary decision trees is NP-complete”, Laurent et al., 1976 [LR76]). Furthermore, the maximal depth of our regression tree was arbitrarily set to 8 to easily prevent overfitting, but above all to be able to take advantage of its natural interpretability, as a very large tree is impossible to understand for a human eye. This is why we included a random forest (Ho, 1995 [Ho95]) for comparison, which was logically supposed to perform better than the regression tree while keeping the same structure. Indeed, the very principle of a random forest is to learn from multiple trained trees generated with a random factor and to compute their average, thus learning from several trees rather than only one. Among others, it allows the random forest to generalise better on unseen data: this is unmistakable if we compare the results of both models with the same depth (see Section 4.2.4), as the random forest very clearly outperforms the regression tree whatever the features.

However, it was more difficult to anticipate the relative performances between the random forest and our first two models; even with our results, it remains unclear whether the difference comes from the models themselves, from the restrictive maximal depth of the random forest (identical to the regression tree for a fair comparison) or from both. Lastly, we can notice how inadequate the regression tree is for our task by how its performances decrease drastically with the *Nested* and *Improved* baselines, where the results of all other models follow the reverse tendency. The regression tree is indeed the only one who sees both its average precision and its recall drop with *Improved Dubremetz* compared to *Dubremetz*.

5.2 Analysis of the Novel Features and Models

Since the *Dubremetz* features were thoroughly analysed by their creators (Dubremetz and Nivre, 2015, 2016, 2017), we will focus on the novel features proposed in this thesis (see Section 3.4.3).

5.2.1 Parison

The most startling singularity about parison is how well this feature seems to work on our dataset to discriminate salient antimetabole and uninteresting inverse repetitions. This result was confirmed through and through, with different experiments and on different models: its presence systematically and considerably increased the average precision of the associated predictive model; although with disparate results for the recall, but even then, these were never alarming enough to question its overall efficiency.

Our parison feature is in fact declined in three different features weighing on the different parts on an antimetabole:

1. the “introductions” (what Dubremetz and Nivre called C_{left} and C_{bb}) are comprised of the words coming before the very first term of the antimetabole and of the words in between the central terms of the antimetabole. The associated *parisonIntro* feature computes the number of corresponding Part-of-Speech tags by mapping the words of the two previously defined intervals.
2. the “middle intervals” (what Dubremetz and Nivre called C_{ab} and C_{ba}) are comprised of the words between each pair of terms of the antimetabole, except for the central ones. The associated *parisonBetweenIntervals* feature computes the average number of corresponding Part-of-Speech tags by mapping symmetrically the words of the previously defined intervals (C_{ab} with C_{ba} for a four-term antimetabole, C_{ab} with C_{cb} and C_{bc} with C_{ba} for a six-term antimetabole, etc.).
3. the “conclusions” (what Dubremetz and Nivre called and could have called C_{bb} and C_{right}) are comprised of the words in between the central terms of the antimetabole and of the words coming after the very last term of the antimetabole. The associated *parisonConclusion* feature computes the number of corresponding Part-of-Speech tags by mapping the words of the two previously defined intervals.

The fact that the parison features improve the average precision more than the recall is still to be nuanced by the results of the tree-based models. Indeed, if the parison improves very marginally or deteriorates the recall of the logistic regressor and of the SVM, it may well be that their recall without it is already extremely high (respectively 96.3% and 92.9%) ! With the standalone features evaluation, we showed that models with much lower recalls (21.4% for the regression tree and 77.8% for the random forest) see it improve substantially along with their average precision with only the addition of

the parison features.

In order to gain more insight on these three sub-features, let us take a look at their Partial Dependency Plots, computed for the best model where their impact is the most significant: the logistic regression model based on *Dubremetz* and with all novel features combined. This model, because of its optimal trade-off between simplicity and performance, will serve as a reference for the future features analyses.

Put simply, the Partial Dependency Plots (Friedman, 2001 [Fri01]) show the empirical influence of a specific feature on the predictions of a model and can be very useful to determine whether a feature helps the model to discriminate its candidates positively or negatively, and the extent of the feature’s impact. The Partial Dependency Plots for the three sub-features of parison can be found in the Appendix Section C.7.1, while the summary plot of all three can be found in the Figure 5.1.

The first interesting result is that the two last sub-features, *parisonBetweenIntervals* (Figure C.15) and *parisonConclusion* (Figure C.16), indicate a positive effect for the predictions of the model: in other words, the higher the value of the feature, the higher the score of the candidate. This was completely expected, since we know thanks to the Chapter 2 that the stacking of rhetorical figures increase their rhetorical saliency. On another hand, the fact that the first sub-feature, *parisonIntro* (Figure C.14), shows an inverse tendency, is quite surprising. It means that depending on the location of the parison in the candidate, the model either uses it to classify said candidate as a salient or as a non-salient antimetabole.

However, this conclusion is nuanced by two other evidences. The first one is the actual relative impact of the three features on the predictions: if we take a look at the y-axes, or if we consider the three plots at the same time (Figure 5.1), we may notice that the scale is very different between the first one and the two other ones, suggesting that the *parisonIntro* feature is actually much less important than its complementary sub-features. Thus, the surprising result for *parisonIntro* is attenuated by the fact that the sub-feature does not matter much by itself. Furthermore, if we dig deeper, we can partially explain this observation by noticing that *parisonIntro* concerns (i.e. has a non-zero value for) around 500 candidates, while *parisonBetweenIntervals* and *parisonConclusion* respectively concern around 4000 and 900 candidates. Lastly, the former has a value lower than 1 (indicating a moderate if at all existent parison) for almost 90% of its concerned candidates. For comparison, the much more effective *parisonBetweenIntervals* feature has a value lower than 1 for 67% of its concerned candidates, in addition to being much more widespread.

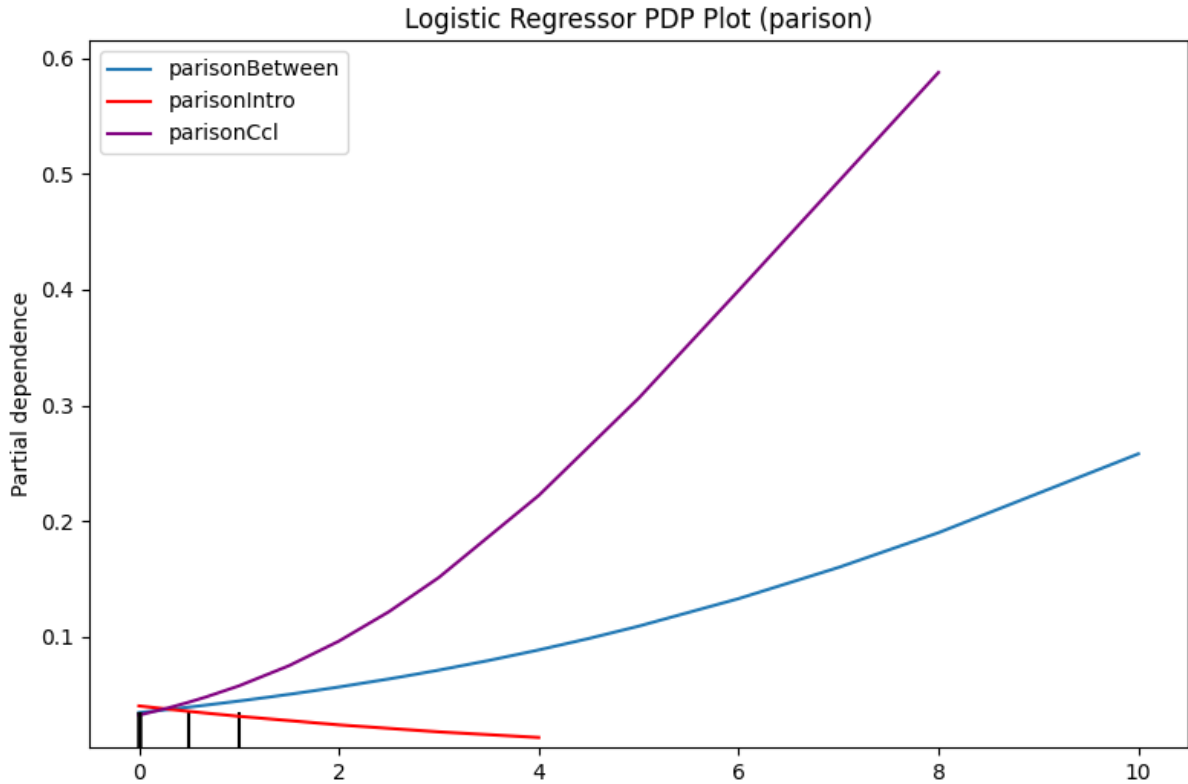


Figure 5.1: Partial Dependency Plot for all parison features as part of the logistic regression classifier based on *Dubremetz* with all novel features.

In conclusion, we may confirm the intuition of Schneider et al. (2021), who based their whole chiasmi candidate extraction algorithm on Part-of-Speech tag, and the assertions of Harris and Di Marco (2017), that the syntactic structure of the antimetabole is of high importance with regard to its rhetorical saliency. Nevertheless, such an affirmation is quite bold considering the objectively limited size of our dataset, which may well be biased and mislead our conclusions. If anything, this only further emphasizes the need for bigger, more complete and more diversified datasets for the computational study of rhetorical figures.

5.2.2 Isocolon

On the contrary, the most startling singularity about isocolon is how... useless it seems to be as a feature - in the best scenarios. In the worst cases, it is even counterproductive: it doesn't have any effect on the performances of the logistic regression model alone, when

added separately or with all other features, and seriously damages the performances of our three other models. Unsatisfied with a mitigated result such as raising the average precision while deteriorating the recall (which they do to the regression tree), the isocolon features go as far as undermining both for the SVM and the random forest if we add them to the baseline. Luckily, these effects are heavily limited when the models are already equipped with all of our other features, making them more robust, as the results of the ablation study showed us.

Before diving again into Partial Dependency Plots, let us precise what the features are exactly:

1. First of all, the *isocolonInTerms* binary feature checks the presence of an isocolon within the terms of the antimetabole. As an identity of lemmas often leads to an equal number of syllables, we decided to only check all terms at the same time by setting this feature’s value to 1 if all terms have the same number of syllables, and 0 in the opposite case.

Similarly to the parison, the three next sub-features check the presence of isocolons in the introductions, between the terms and in the conclusions of the antimetabole.

2. The *isocolonIntro* binary feature is set to 1 if the total number of syllables in both introductory intervals is identical, and otherwise to 0.
3. The *isocolonBetweenIntervals* feature computes the average number of intervals that have the same total number of syllables.
4. The *isocolonConclusion* binary feature is set to 1 if the total number of syllables in both concluding intervals is identical, and otherwise to 0.

At first glance, the definitions of these features may seem more restrictive than those for parisons, and could indeed begin to explain why they gave such poor results. In any case, we will not stop at a first impression, and will see what the numbers have to say: let us begin by analysing the four sub-features Partial Dependency Plots (in the Appendix Section C.7.2), computed in similar conditions to the parison. A summary of all four plots is also given in Figure 5.2.

Interestingly enough, this time, we do not only have one off-setting result, but two. The ugly ducklings of the isocolon family thus are *isocolonInTerms* (Figure C.17) and *isocolonConclusion* (Figure C.20), which, contrarily to our intuition, prove to us that the model learned to negatively discriminate instances that show an isocolon within

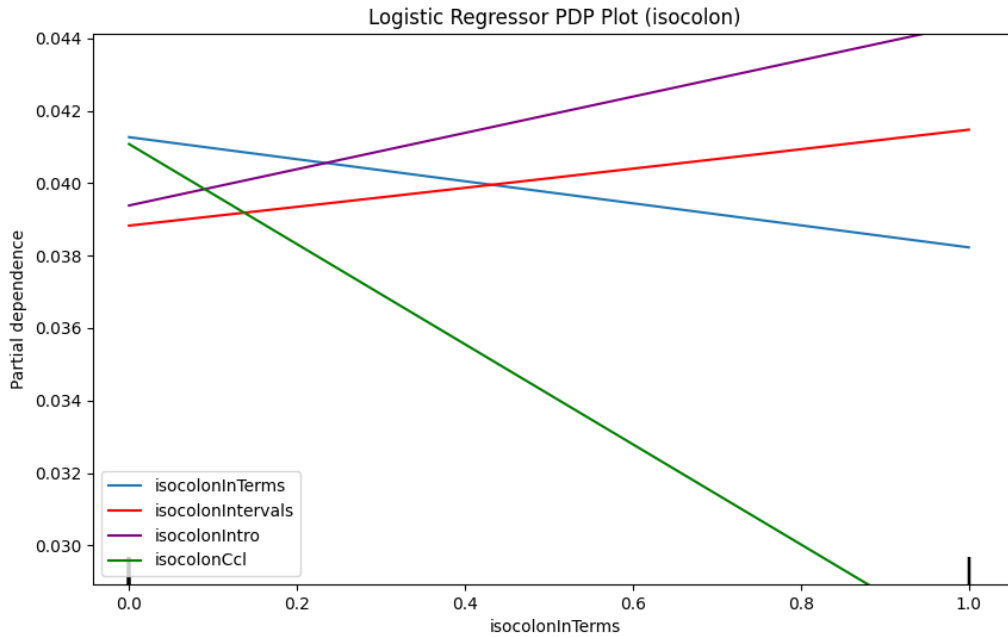


Figure 5.2: Partial Dependency Plot for all isocolon features as part of the logistic regression classifier based on *Dubremetz* with all novel features.

their terms or in their conclusions. Fortunately, the two other sub-features reassure us by showing a the inverse behavior, consistent with our expectation. On another hand, the summarising graph (Figure 5.2) presents us two other interesting facts: contrarily to the parison, all sub-features here seem to have a relatively equivalent importance with regard to the model’s predictions, and, compared to the parison’s sub-features, the isocolon ones have an impact similar to the parison’s least important one. Once again, this somehow softens the counter-intuitive results of the sub-features having a negative impact on the candidates ratings.

Last but not least, we can also dismiss the preliminary objections to the construction of these features by seeing how many candidates they concerned. Obviously, the most widespread feature remains *isocolonInTerms* with 65% of candidates giving it a value of 1. The two other binary features, *isocolonIntro* and *isocolonConclusion*, both show a range between 9% and 10% of instances having positive values, while *isocolonBetween-Intervals* has around 20% of non-zero values.

In conclusion, the failure of the isocolon features is not due to their rarity, but solely to the fact that the model did not consider them to be effective discriminating factors

during its training.

5.2.3 Nominal groups

The *nominalGroups* feature is easier to analyse since it is not declined in several sub-features. For each pair of matching terms in an antimetabole, it simply computes the average number of matching words coming right before and after said terms, as depicted in the Section 3.4.3. Leaving the field of stacking rhetorical figures, this feature was intuitively inspired by the fact that our candidates are extracted based on the repetitions of single words, and not nominal groups, although the latter would make more sense in numerous cases. Therefore, this *nominalGroups* feature should have been able to capture such cases of antimetabole and to help the model to positively discriminate them. But as we have seen with our first two features, our intuition is rarely entirely proven right. Let us then take a look at the Partial Dependency Plot for our feature, found in the Figure 5.3. As “expected”, the feature does not match our expectations: instead of signaling salient antimetabole, the graph shows us that the bigger the nominal groups attached to an antimetabole terms, the lower its given score will be. And, this time, we do not have any other sub-feature to put it into perspective: the *nominalGroups* feature is clearly an indicator for uninteresting instances. At this point, we are allowed to say: “Why, *nominalGroups*? You were the chosen one. It was said that you would compensate for the bad features, not join them ! Bring balance to the rhetorical force, not leave it in darkness...”

Anyway, since 66% of the candidates have a non-zero value and 28% have a value above 1 for this feature, it still means that the feature should be interesting for our model. So, where does the fault lie ? If we take a closer look at the Partial Dependency Plot, we can see that the biggest values obtained for the feature are higher than 10, which is absolutely huge: it means that some instances have both their pairs of terms each inside a nominal group of ten words (in average). How is that even possible ? Let us see what these instances look like:

1. We know you 're a little slut . No , I 'm not ! I 'm not a slut ! I 'm not a slut ! I 'm not a slut ! I 'm not a slut ! I 'm not a slut ! I 'm not a slut ! I ain't no slut !
2. Get up , get up ” - ” Ooh , yeah ” ” Get up , get up , get up ” - ” Get up , get up , get up ” - ” Get up ” - ” Getting

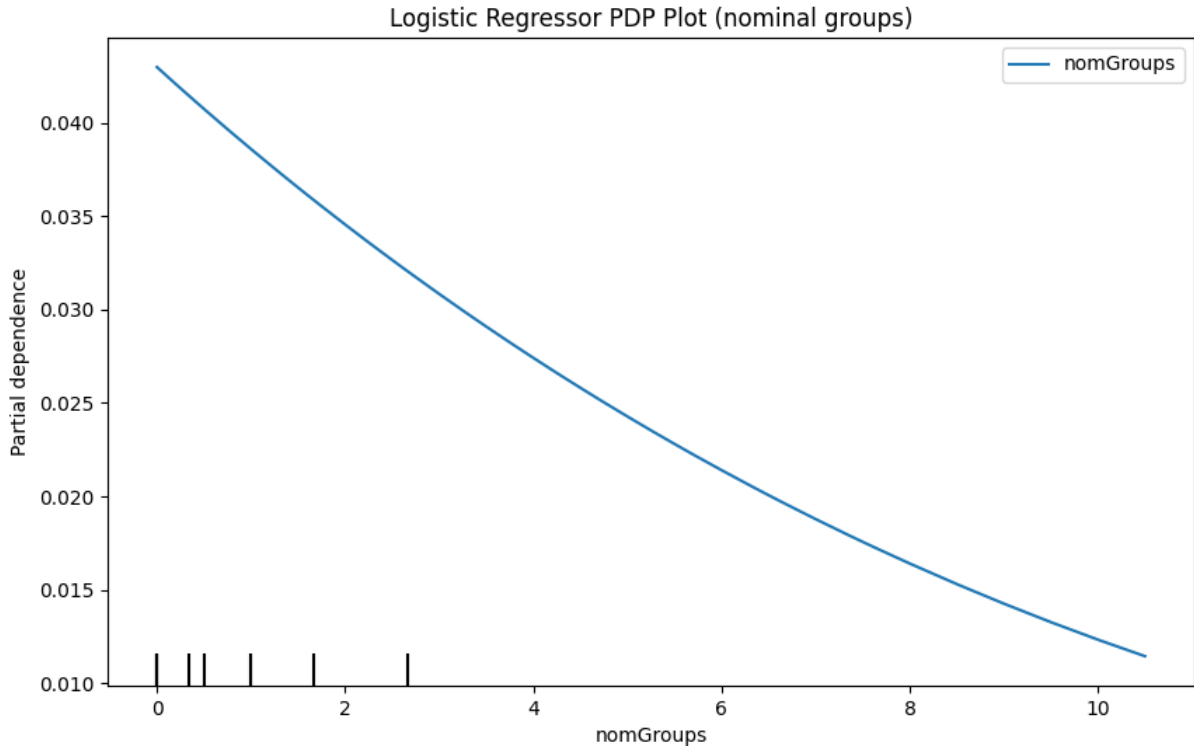


Figure 5.3: Partial Dependency Plot for the *nominalGroups* feature as part of the logistic regression classifier based on *Dubremetz* with all novel features.

3. Oh , dear God , please make this crazy kid go away . Go away . Go away . Go away . Dear God , please make this crazy kid go away . What do you want
4. guitar strings , then pausing to sing . ” Never gon na get old , Never gon na die . Never gon na get old , Never gon na die . ” Guiseppi Scapellini
5. walk away ? ? Or you ’ll crash ? ? Crash , crash ? ? Oh , you ’ll crash ? ? Crash , crash ? ? Oh , you ’ll crash ? ?

As you may guess, these instances are not part of the original and manually retrieved dataset. They come from the additional set of supposed false instances taken randomly from the Corpus of Contemporary American English (COCA). The first three quoted examples make up for all the instances that have a value over 7 for the *nominalGroups* feature, and the five examples make up for all instances that have a value over 5. Therefore, it is easily understandable that, with such borderline cases, our feature did not behave according to our expectations !

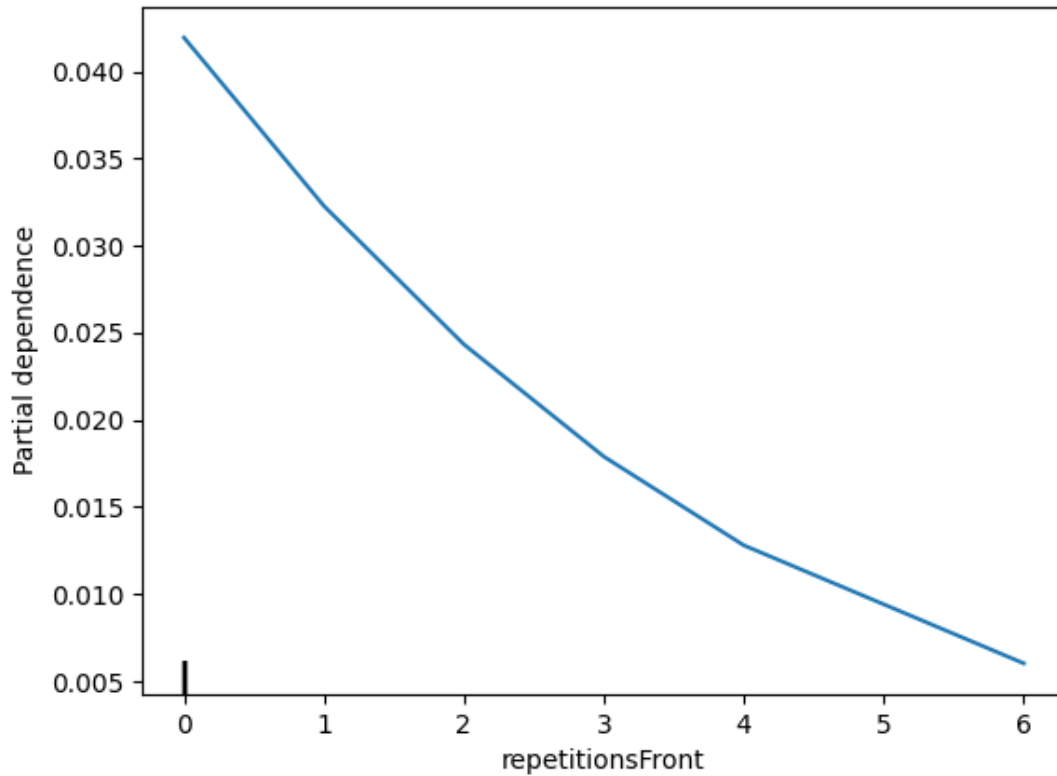


Figure 5.4: Partial Dependency Plot for the *repetitionsFront* feature as part of the logistic regression classifier based on *Dubremetz* with all novel features.

5.2.4 Repetitions

Eventually, the last novel feature that we tried to add to the state of the art was, in a sense, a continuation of the same state of the art’s ideas. Indeed, many of Dubremetz and Nivre’s features rely on the idea that an antimetabole is built on an axial symmetry, beyond the obvious one formed by the terms of the antimetabole itself (AB / B’A’). Therefore, the *repetitions* feature try once again to assimilate that sense of symmetry, by trying to capture some sort of epiphora or of epanaphora (see Section 2.2.2) present directly around (at the beginning or at the end) the antimetabole’s terms, rather than at the beginning or at the end of a sentence or of a clause. The *repetitions* features are then naturally divided in two:

1. The *repetitionsFront* feature computes the number of repeated words before the

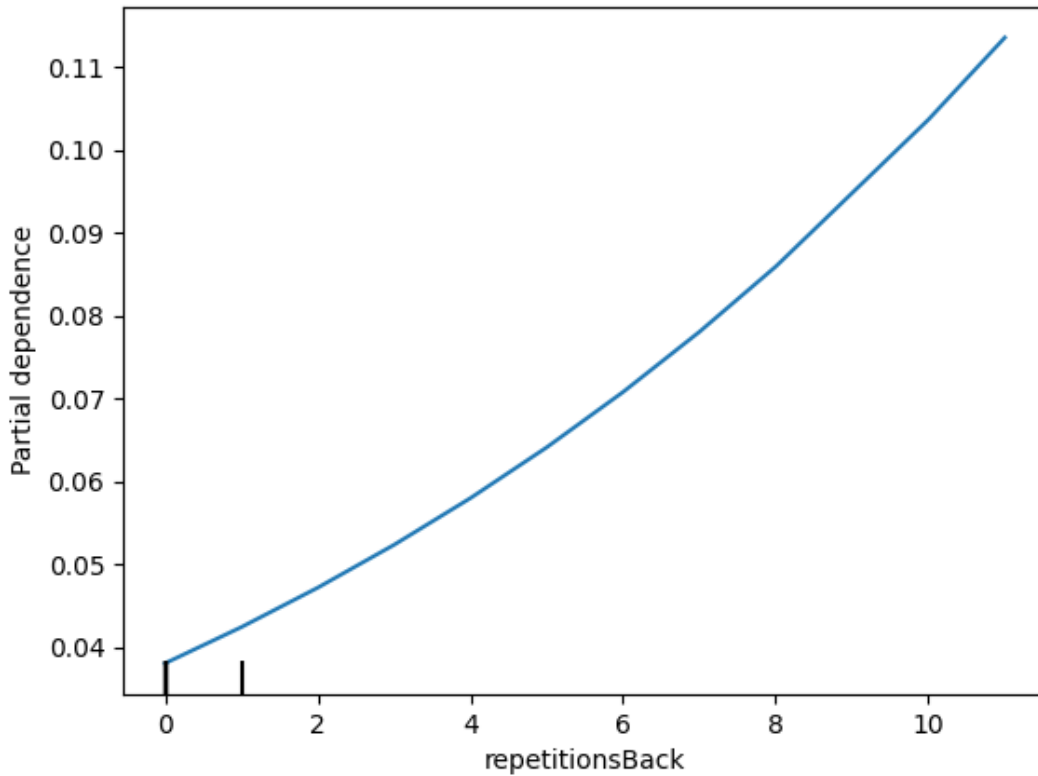


Figure 5.5: Partial Dependency Plot for the *repetitionsBack* feature as part of the logistic regression classifier based on *Dubremetz* with all novel features.

first term of each half of the antimetabole (in C_{left} and C_{bb} to borrow Dubremetz and Nivre’s notation).

2. The *repetitionsBack* feature computes the number of repeated words after the last term of each half of the antimetabole (in C_{bb} and what we could call C_{right}).

Not unlike the *nominalGroups* features, the *repetitions* features lead to shy or nonexistent results improvements for every model it was tested with. One last time (I swear), we will thus call on two magical Partial Dependency Plots (Figures 5.4 and 5.5) to help us understand how these features influenced the logistic regression model’s behavior.

Once again, the results are... as surprising as interesting. First of all, the two sub-features show a contrary influence on the model’s predictions: *repetitionsFront* indicates

a non-salient antimetabole when its values get higher while *repetitionsBack* points towards a higher score, except for the difference that *repetitionsBack* has a quite greater impact in terms of scale. This may be explained by the fact that *repetitionsBack* also presents a broader range of values, going up to 11 repeated words, while *repetitionsFront* stops at a maximum of 6. In addition to that, a further analysis of the features values shows us that *repetitionsBack* is given a non-zero value for 1600 candidates, compared to only 340 for *repetitionsFront*.

Although, surprisingly enough, the borderline cases encountered in the (previous) Section 5.2.3, especially the second one, are found in the highest values of *repetitionsBack*, which is yet the “good” feature out of the two. If anything, this suggests that the *repetitionsBack* feature has an even greater potential than its Partial Dependency Plot shows! As to why *repetitionsBack* works better than *repetitionsFront*, our data points to the former being much more widespread and meaningful than the latter; however, we still lack the necessary linguistic hindsight and a sufficiently large dataset to draw solid conclusions, instead of being misled by biased data.

5.2.5 Regression Tree Visualization

The simplicity of the regression tree, which served it poorly up until now seeing how it performed along with our three other models, becomes its only advantage when it comes to interpretation. Indeed, one of the specificities of this kind of model is that it is possible for a human to directly interpret its prediction process, instead of having to resort to complementary tools such as Partial Dependency Plots. In our case, as our regression tree did not fit on one page, it can be found in the Appendix Figures C.21 (upper part of the tree), C.22 (lower left part) and C.23 (lower right part). The resulting model, following the idea of this Section, was trained with the *Dubremetz* baseline features augmented with all of our novel features.

The first part of the tree, which in optimal cases should try to maximize its information gain by finding the most interesting splits of the data, already looks here very bad. Indeed, the first three splits shown in the Figure C.21 only allow the tree to discriminate three instances (one per split) out of 15009 while already “consuming” 37.5% of its height.

The fourth split, leading to the remaining two parts of the tree, looks much more interesting: it uses a *Dubremetz* dependency feature, *sameDepWbWa'*, to split the data into two sets containing respectively 12164 (lower left part of the tree) and 2842 instances

(lower right). Let us go with the biggest split, found in the Figure C.22.

The next split seems once again sub-optimal, since it only classifies seven more instances, before leaving the hand to another one of *Dubremetz* features, *simScore*, taking 1300 instances out of the main split. The smallest one is then split in a weird way, leaving 1250 instances to be rated as non-salient, while the remaining 50 eventually give 20 salient instances. We can already notice that the tree’s recall will probably be very low, since at this point, it classified about 1300 instances as “false” and only 27 as “true”, for a ratio of 2% while the proportion of salient instances in the whole dataset is around 4.5%. If we go back to the main split of this part of the tree, still containing more than 10800 instances, we can shake our heads by seeing another somewhat useless split to classify a single instance as positive; while the last split on this part will not give anything interesting. Thus, from 2%, we are suddenly at 0.2% of instances rated as “true”. Hopefully, the lower right part of the tree (containing 2842 instances, remember ?) in the Figure C.23 will do better than its symmetrical counterpart... Interestingly, its first split uses the same feature as the lower left part, *parisonConclusion*, but much more efficiently: 2500 instances on one side and 340 on the other. On the biggest side, however, the next two splits are quite disappointing: the first one will send 65 instances to the “non-salient graveyard” with the help of the *sameBigram* feature from Dubremetz and Nivre, while the second one classifies 2400 instances as false with *mainRep*. The last 10 remaining will finally be split by *sameTok*, and give us 9 instances rated as “true”. In the end, the 2500-instances split only gave these 9 interesting instances, keeping the recall as low as ever with a total of 37 “true” for 14668 “false”.

The last split in which we may place all of our remaining hope is the furthest right branch of the tree, with only 341 instances. Its next split looks promising, with 207 instances on the left side and 134 on the right; however, the latter will only give 6 additional positives. The former, on another hand, manages to bring us 200 positives samples at once with two splits using *sameTrigram* and *punct*, and a last single positive on its other branch.

Eventually, we are thus left with 244 positive instances out of 15009, whose 585 instances were annotated as true, giving us the expected recall of 42% (see Section 4.2.4). Out of those 244, 200 were roughly discriminated thanks to 5 features: 4 from the baseline, and one from the parison features (*parisonConclusion*), which is not surprising at all given that it was the most interesting set of features within all of the novel ones. Furthermore, this specific sub-feature was already seen in the Section 5.2.1 as the most impactful one in terms of rating with regard to its Partial Dependency Plot. In addition to that, the

parison features are the most represented novel features in this regression tree, since they account for 5 splits (and only one discriminating a single instance), the repetitions for 3 splits (all of which discriminate a single instance), the nominal groups for 1 split and the isocolon is downright missing.

6 Conclusion and Future Work

6.1 Conclusion

This thesis aimed at contributing to the state of the art concerning the automatic extraction and detection of two related rhetorical figures: chiasmi and antimetabole.

First of all, we investigated many different sources, from dictionaries to past research, including linguistics, in order to settle on a definition for our two figures. This step was indeed essential insofar as said sources very often contradicted themselves, and no consensus could be found for either figure - especially for the chiasmus. Eventually, we decided to restrict both figures' definitions in order to make them match our needs and our computational capacities, although we specifically decided to make them more wide-ranging than the state of the art's.

Then, we conducted an extensive study of the state of the art in terms of antimetabole and chiasmi detection. For the former, the works of Dubremetz and Nivre (2013-2018) completed previous pioneering work and were very insightful, as well as gave us our first baseline for future experiments; for the latter, Schneider et al. (2021) inspired us to venture into chiasmi detection as well, and try not to stop at antimetabole. Throughout our whole research, the work of Harris et al. was also a beacon in the night and helped us greatly to grasp the linguistics fundamentals behind our rhetorical figures.

In the next chapter, we detailed the first contribution of this thesis: a complete pipeline for extracting and annotating chiasmi and antimetabole from raw text. Two similar algorithms had been developed by Dubremetz and Nivre and Schneider et al., respectively for antimetabole and chiasmi, but the former is to this date unusable and the latter is flawed by design. In any case, both did not go as far as we did in proposing a tool that took everything in charge from start to finish - from the raw text to an annotated dataset. While our extracting of antimetabole proved to be perfectly sound, our method for extracting chiasmi did not yield any satisfactory results, due to the limitations of

current embedding models. In the same manner, the principal limitation of our antimetabole extractor was the external lemmatizer it used.

Afterward, we presented the second big contribution of this thesis, which consists of a brand-new and more than 10 times bigger dataset than the only available one from the state of the art to this date. Even if our dataset contains chiasmi, our extraction tool only permitted us to annotate its antimetabole. For the annotation, we introduced the concept of "nested" chiasmi and antimetabole, resulting from our new definition of both figures.

Eventually, we presented the baselines used for our antimetabole detection experiments, the four different types of traditional Machine Learning models and the four new sets of features we introduced in our attempt at improving the state of the art's baselines.

In the Results section, we briefly evaluated our antimetabole extraction algorithm's performances. Most importantly, we thoroughly presented the numerous results from our various experiments with our different baselines, models and features for detecting rhetorically salient antimetabole among the set of candidates retrieved by our extraction pipeline. In summary, we managed to substantially improve the state of the art's performances by enhancing their own features. On another hand, we also improved the baselines by adding our new features, and observed that one feature in particular gave impressive results, one other surprisingly bad results, while the two remaining gave mitigated outcomes. We confirmed these findings through different experiments on a logistic regression model, before corroborating them with four different models.

The model's relative performances broadly matched our expectations, with the logistic regressor closely followed by the SVM, the random forest staying a good step behind and the regression tree being completely out of the race.

In the last section, we further investigated the results of our experiments. First of all, we proposed explanations as to how our improvements of the baselines managed to get such remarkable results, and clarified the effect that the nested antimetabole candidates had on the model's predictions, and then justified our expectations for the model performances: in particular, the regression tree and random forest were purposefully restrained by their maximal depth, in order to allow a direct interpretation. Thereafter, we conducted an extensive analysis of our four novel features. We disclosed their particular characteristics of the sub-features for each set, and explained their respective influence on a specific model's predictions with the help of Partial Dependency Plots. We completed this study with additional statistical evidences, and eventually found out borderline instances in our dataset that partially misled the models training for some

of the features.

Eventually, we proposed a comprehensive examination of our full regression tree trained with all of our features.

6.2 Insights

I deem it important to warn the reader that this section contains a lot of, if not only, personal opinions. These opinions commit only the author of this thesis at the moment he wrote it.

6.2.1 Research Work

Working on this thesis, and especially on the state of the art, has definitely been a crucial lesson for me with regard to the teachings I received about scientific work and scientific research. In computer science in particular¹, one aspect of this "scientific research work" is often neglected, and I have had the misfortune to witness it by myself while working on this thesis: the reproducibility of a scientific experiment.

As trivial as it may seem, I consider this point to be absolutely crucial for any valid scientific work. One of the pillars of modern science is, according to me, peer-reviewing: being able to double-check any assertion made by any colleague, to reproduce any experiment, and finally, to validate or refute any result and its associated hypothesis or conclusion. We live in an era where information flows and "fake news" have never been so important, and at the same time where scientific communities are sometimes ignored, if not laughed at, and struggle to be heard even if they manage to reach a global consensus exempt of any credible scientific contradiction. I am thinking here about the IPCC (Intergovernmental Panel on Climate Change), which was created in 1988, whose first report was published in 1990, and yet, more than 30 years later, their reports are still not taken *seriously enough*. If anything, this example tells me that scientific research should at the very least be taken seriously by its researchers. Otherwise, I feel like it is unfair to expect others to give them any credit.

I have no intention of casting stones, or let alone, as a master student, of having the

¹If this assertion seems very presumptuous due to my lack of experience in domains other than computer science, and even in this one, I base it on the previously mentioned teachings.

audacity of criticizing scientific work that has been published and peer-reviewed. However, as I mentioned it, my view of peer-reviewing in computer science is not limited to the published article, but also - if not especially - encompasses the implementation. And as I tried to work on the state of the art of this thesis, I found myself more than once utterly unable to replicate (and *a fortiori* reproduce) any experiment described in published articles. Unmaintained, inaccessible or non-functional code, unavailable data - the issues have been diverse and I will not speculate further on their causes. I simply felt that I could not avoid, as part of a fundamental research project, attracting attention on these matters; at least in a self-directed (understand: directed at myself) pedagogical approach.

6.2.2 Application

As incongruous as it may seem to discuss the potential applications of this thesis' work at its very end, I considered this choice to be more coherent with regard to the personal conclusion I will add to it.

First of all, the most recurrent application in previous articles on the matter (see Chapter 2) concerns stylometric analyses. Indeed, the computational study and especially detection of rhetorical figures immediately gives way to imagining a complete tool for judging, sorting, analyzing, and classifying various texts: the presence or absence of rhetorical figures may, according to the context, be an indicator of quality, style, thought, or may on the contrary suggest superficiality, manipulation or fallacies, especially in speeches or argumentative texts.

On another hand, one idea inspired from Dubremetz (2013) stroke a chord with my writing passion: instead of detecting rhetorical figures, large enough datasets could very well help complex models to automatically generate instances of such figures, and propose them as a writing assistance, or help to enrich already existing figures. This could find an application in many various domains, from pedagogy (teaching poetry at school could become much easier and more fun) to translation (as rhetorical figures are often held as major hurdles in translating texts) as well as tools for (everyday or professional) writing assistance.

From a linguistic point of view, any large study on rhetorical figures can also be very insightful for various research fields, from classifying to analyzing specific figures. As we saw in the Section 2.1, giving a proper definition of a rhetorical figure can sometimes be

much trickier than it seems, and would benefit much from statistical as well as computational insights.

6.2.3 Personal Assessment

All of that being said, I would not be true to myself if I did not try to contradict it. As I am fond of asking myself questions, I could not help but wondering, while working on this thesis: do we *need* - do we even *want* - computers to help us understanding rhetorical figures? Do we deem it important enough to carry on such experimental work, which could only find its purpose in 20, or 30 years? As both a writer and a (future) computer engineer, I could not find a clear answer to these questions.

However, my ecologist conscience tells me otherwise: in the face of both worldwide and personal crises, from the climate crisis (going hand in hand with the biodiversity collapse and the exceeding of many irreversible planetary limits) to personal enquiries about the kind of world I would like to live in tomorrow, my personal answer to the two aforementioned questions, at this time, is negative.

Before calling it "the naive heat of youth" and brushing it aside, let me clarify that I certainly do not intend to "save or change the world". My personal approach to this is a simple matter of coherence with myself and my values. And, as much as I may find working on rhetorical figures interesting and entertaining, staying in denial of the previously cited crises (and more uncited) builds up too much cognitive dissonance for me to pursue it as is.

6.3 Future Work

The possibilities for carrying on with this work are plentiful for each phase. First of all, the extraction pipeline, as explained in Section 3.2, only works for antimetabole. In addition to that, the only chiasmi extraction algorithm that exists to this date is from Schneider et al. (2021), but cannot identify all chiasmi according to our definition given in Section 2.1.1. Thus, either an improvement of Schneider et al.'s algorithm or a redesign of ours is necessary to carry out chiasmi detection. Furthermore, we saw that the main limitation of our antimetabole extractor was the lemmatizer it used: as

a consequence, finding a better lemmatizer or a manual way to counterbalance its defects would be a great improvement for the automatic extraction part. Lastly, Section 4.1 demonstrated that the empirical size of 30 tokens for the extraction introduced by Dubremetz and Nivre (2015) does not hold anymore and should be revised.

Concerning the data itself and its annotation, we managed to greatly contribute to expanding the current available datasets, but some consolidation work still needs to be done. We saw in Section 5.2 that, contrarily to Dubremetz and Nivre’s intuition in 2017, we cannot completely disregard any unseen text as ”False“; moreover, these borderline cases raise another annotation issue: dealing with sentences that bring a lot of candidates with very few words, while only one of them can be annotated as *the* antimetabole. These similar candidates with different annotations throw the model training out of balance, similarly to the distinction between the nested and binary candidates. In order to be able to make the most out of our newfound data, these intricate matters need to be addressed. On another hand, the binary annotation between ”salient“ and ”not salient“ may be improved with regard to the desired prediction outcomes, which are discrete ratings. To that end, one solution worth exploring could be to annotate the candidates in pairs rather than giving them an absolute label, as is often done in automatic argument quality assessment; deciding which of two candidates is *more* salient could indeed make annotation easier and more intuitive, as well as removing the need for nonsensical labels such as ”Borderline antimetabole“, all the while resulting in a more nuanced annotation scheme. The associated increase in resources needed for annotation could be counterbalanced by crowd-sourcing.

Nonetheless, it goes without saying that any further development of this work should be preceded by a thorough reflection on whether said work *should* be developed.

A Glossary

The following definitions are expressly summarized or simplified for brevity. For more details, especially about the sources of these definitions, please refer to the Sections 2.1, 2.2 or 3.4.3 where most of these definitions are presented and discussed. The main source for these definitions is Harris and Di Marco (2017) [HD17].

- **Anaphora:** see Epanaphora.
- **Antimetabole:** repetition of words in reverse order.
 $[W]_1 \dots [W]_2 \dots [W]_n \dots [W]_n \dots [W]_2 \dots [W]_1$ *Drake loves loons. Loons love Drake.*
- **Antithesis:** proximally opposed predications, through antonyms or affirmatives and negations.
*They wanted **peace**? Let's bring them **war**.*
- **Chiasmus:** consists of the repetition of two or more pairs of related words in the reverse order.
*He **came** in **triumph** and in **defeat** departs.*
- **Epanaphora:** repetition of a word or a group of words at the beginning of successive sequences of language (e.g. sentences).
***I am** an actor. **I am** a writer. **I am** a producer. **I am** a director. **I am** a magician.*
- **Epiphora:** repetition of a word or a group of words at the end of successive sequences of language (e.g. sentences).
*I'm so **gullible**. I'm so damn **gullible**. And I am so sick of me being **gullible**.*
- **Epistrophe:** see Epiphora.
- **Isocolon:** denotes members that are identical in number of syllables or in scansion.
Was ever woman in this humour woo'd? / Was ever woman in this humour won?

- **Mesodiplosis:** the occurrence of the same word or word sequence in the middle of proximal clauses or phrases.

*All **for** one, one **for** all.*

- **Parison:** A scheme of syntactic repetition, often referred to as syntactic parallelism: the proximal repetition of the same syntactic pattern.

*My life is spent alone, **without wealth, without status, without love, and without hope.***

B Topic for Master Thesis

B.1 Data Augmentation and Machine Learning for Rhetorical Figures

Rhetorical figures are not only used in advertisements or political speeches, but also in fake news, hate speech or argumentations. Detecting those figures can help improving the overall text understanding and detect subtle meanings.

The thesis focuses on rhetorical figures that are especially used for persuasion in arguments. Fahnestock [1] describes as examples for these figures with “argumentative potential” the figures antimetabole and chiasmus. Both show a crisscross pattern with antimetabole using the same words (e.g., “eat to live, not live to eat”) and chiasmus using contrary words (“the spirit is willing but weak is the flesh”).

However, those figures are rather rare, causing a decrease of accuracy in the detection with the use of statistical methods [2]. In [3] and [4], the authors were not able to use machine learning for the detection of chiasmus as the dataset was too small. The goal of this thesis is to extend existing datasets by collecting examples and developing suitable data augmentation techniques. The next step is implementing different rule-based or machine learning algorithms (e.g., active learning for small datasets) and comparing their accuracy for the detection of chiasmus and antimetabole.

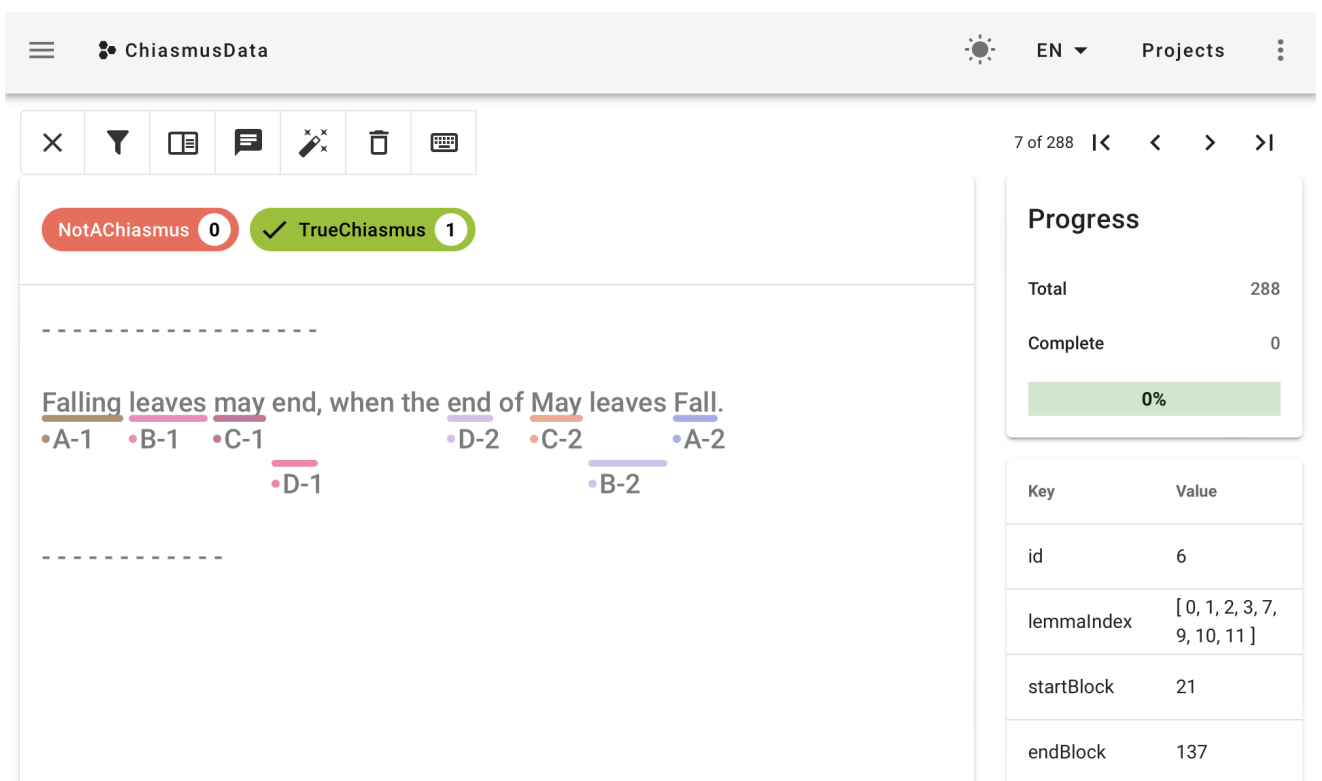
B.1.1 References

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

2. Dubremetz, Marie, and Joakim Nivre. “Machine learning for rhetorical figure detection: More chiasmus with less annotation.” Proceedings of the 21st Nordic Conference on Computational Linguistics. 2017.
3. Dubremetz, Marie, and Joakim Nivre. “Rhetorical figure detection: The case of chiasmus.” Proceedings of the Fourth Workshop on Computational Linguistics for Literature. 2015.
4. Dubremetz, Marie, and Joakim Nivre. “Syntax matters for rhetorical structure: The case of chiasmus.” Proceedings of the Fifth Workshop on Computational Linguistics for Literature. 2016.

C Figures

C.1 Extraction pipeline



The screenshot displays the ChiasmusData web application interface. At the top, there is a navigation bar with a hamburger menu, the text 'ChiasmusData', a sun icon, 'EN', and 'Projects'. Below this is a toolbar with icons for close, filter, list, chat, edit, delete, and keyboard. The main area shows a text snippet: 'Falling leaves may end, when the end of May leaves Fall.' with annotations: 'A-1' (under 'Falling'), 'B-1' (under 'leaves'), 'C-1' (under 'may'), 'D-1' (under 'end'), 'D-2' (under 'end'), 'C-2' (under 'of'), 'B-2' (under 'May'), and 'A-2' (under 'leaves'). Above the text are two buttons: 'NotAChiasmus 0' and 'TrueChiasmus 1'. On the right, a 'Progress' panel shows 'Total 288' and 'Complete 0' with a 0% progress bar. Below that is a table with the following data:

Key	Value
id	6
lemmaIndex	[0, 1, 2, 3, 7, 9, 10, 11]
startBlock	21
endBlock	137

Figure C.1: An example of Doccano's graphical user interface for annotation.

```

2  <document>
3  It was not so much a matter of having power to
4  <antimetabole-0>
5  <antimetabole-A-1>do</antimetabole-A-1>
6  a
7  <antimetabole-B-1>thing</antimetabole-B-1>
8  as it was having the power to stop
9  <antimetabole-B-2>things</antimetabole-B-2>
10 from being
11 <antimetabole-A-2>done</antimetabole-A-2>
12 </antimetabole-0>
13 to you.

```

Figure C.2: An extract of an annotated XML file generated by the extraction pipeline.

C.2 Dubremetz and Nivre (2015)

Precision at candidate	+Size	+Ngram	+Lex. clues
10	40	70	70
50	18	24	28
100	12	17	17
200	7	9	9
Ave. P.	34	52	61

Figure C.3: Average precision, and precision at a given top rank, for each experiment, taken from Dubremetz and Nivre (2015).

C.3 Dubremetz and Nivre (2016)

Model	Average Precision	Compared to Baseline
Baseline	42.54	NA
Tag features	59.48	+14
Negative dependency features	40.36	-2.2
Pos dep features	62.40	+20
All dependency features	64.27	+22
All features	67.65	+25

Figure C.4: Average precision for chiasmus detection (test set), taken from Dubremetz and Nivre (2016).

Model	Average Precision	Diference
Baseline	53.00	NA
All features	70.35	+17

Figure C.5: Average precision for chiasmus detection (Sherlock Holmes set), taken from Dubremetz and Nivre (2016).

C.4 Dubremetz and Nivre (2017)

Model		Avg Precision	Precision	Recall	F1-score
Machine	Base	57.1	80.0	30.8	44.4
Machine	All features	70.8	90	69.2	78.3
Human	Base	42.5	–	–	–
Human	All features	67.7	–	–	–

Figure C.6: Results for logistic regression model (Machine) with comparison to the hand-tuned models of Dubremetz and Nivre (2015; 2016) (Human), taken from Dubremetz and Nivre (2017).

C.5 Schneider et al. (2021)

	Schiller dramas			Dubremetz data
	Antimetaboles	Chiasmi	Combined	Antimetaboles
<i>(baseline)</i> D	0.21 ± 0.18	0.15 ± 0.07	0.17 ± 0.04	0.73 ± 0.24
L	0.07 ± 0.08	0.01 ± 0.00	0.02 ± 0.00	0.06 ± 0.01
E	0.10 ± 0.07	0.06 ± 0.03	0.09 ± 0.07	0.25 ± 0.13
LE	0.11 ± 0.06	0.05 ± 0.02	0.08 ± 0.04	0.25 ± 0.13
DL	0.49 ± 0.32	0.14 ± 0.09	0.22 ± 0.07	0.72 ± 0.24
DE	0.48 ± 0.30	0.23 ± 0.13	0.28 ± 0.01	0.73 ± 0.24
DLE	0.48 ± 0.30	0.19 ± 0.09	0.28 ± 0.08	0.73 ± 0.24

Figure C.7: Average precision for different feature combinations. D=Dubremetz features, L=lexical features, E=embedding features, taken from Schneider et al. (2021).

	Schiller dramas			Rest of the GerDraCor corpus		
	D method	D features	DLE	D method	D features	DLE
antimetaboles	6	2	8	7	2	25
chiasmi	5	5	10	6	9	10
combined	11	7	18	13	11	35

Figure C.8: Number of correct examples among the top 100 ranked ones in unseen texts for the Dubremetz method baseline, the PoS inversions with Dubremetz features and the Dubremetz+lexical+embedding (DLE) features, taken from Schneider et al. (2021).

C.6 Dubremetz and Nivre (2018)

Type of instance	True	Borderline	False	Number of candidates
Chiasmus	0	0	100	2,097,583
Epanaphora	1 ± 1.94	3 ± 3.33	96 ± 3.82	10,249
Epiphora	4 ± 3.77	7 ± 4.91	89 ± 6.02	2,723

The corpus is 4M words of parliamentary discourses (159,056 sentences).

Figure C.9: Annotation of 100 randomly selected chiasmus, epanaphora and epiphora candidates, taken from Dubremetz and Nivre (2018).

Epanaphora	F-Score	Δ Baseline
	Av. P.	
Baseline	35.96%	—
	31.74%	—
Full Features	62.25%	+26.29
	54.60%	+22.86
Baseline + DoS	61.18%	+25.22
	54.75%	+23.01
Baseline - Length + DoS	63.73%	+27.77
	55.63%	+23.89

Bold values indicate the most important differences between baselines and experiments.

Figure C.10: Choosing the best model for epanaphora, taken from Dubremetz and Nivre (2018).

Epiphora	F-Score	Δ Baseline
	Av. P.	
Baseline	35.11%	–
	41.91%	–
Full Features	51.80%	+16.69
	60.53%	+18.62
Baseline + DoE	48.48%	+13.37
	56.29%	+14.38

Bold values indicate the most important differences between baselines and experiments.

Figure C.11: Choosing the best model for epiphora, taken from Dubremetz and Nivre (2018).

Experiment	Recall	Precision	F-Score	Av. Prec.
Baseline	30.19	29.09	29.63	19.97
Baseline -Length + DoS	45.28	53.33	48.97	57.92
Δ	+15	+14	+24	+38

Inter annotator agreement Cohen's $\kappa = 0.85$.

Figure C.12: Results for the epanaphora experiments, taken from Dubremetz and Nivre (2018).

Experiment	Recall	Precision	F-Score	Av. Prec.
Baseline	25.71	42.86	32.14	26.78
Full Features	45.71	64.00	53.33	47.90
Δ	+20	+21	+21	+21

Inter annotator agreement Cohen's $\kappa = 0.88$.

Figure C.13: Results for the epiphora experiments, taken from Dubremetz and Nivre (2018).

C.7 Partial Dependency Plots

C.7.1 Parison

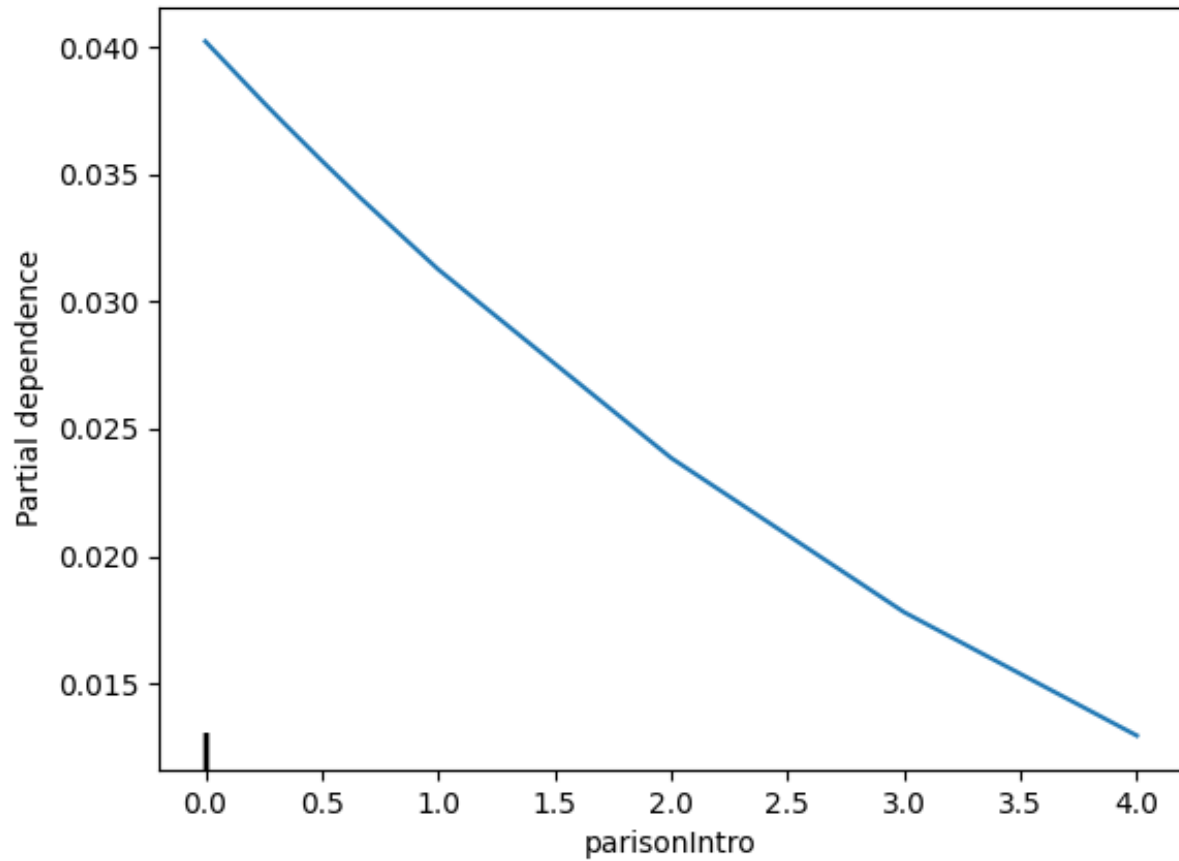


Figure C.14: Partial Dependency Plot for the *parisonIntro* feature as part of the logistic regression classifier based on *Dubremetz* with all novel features.

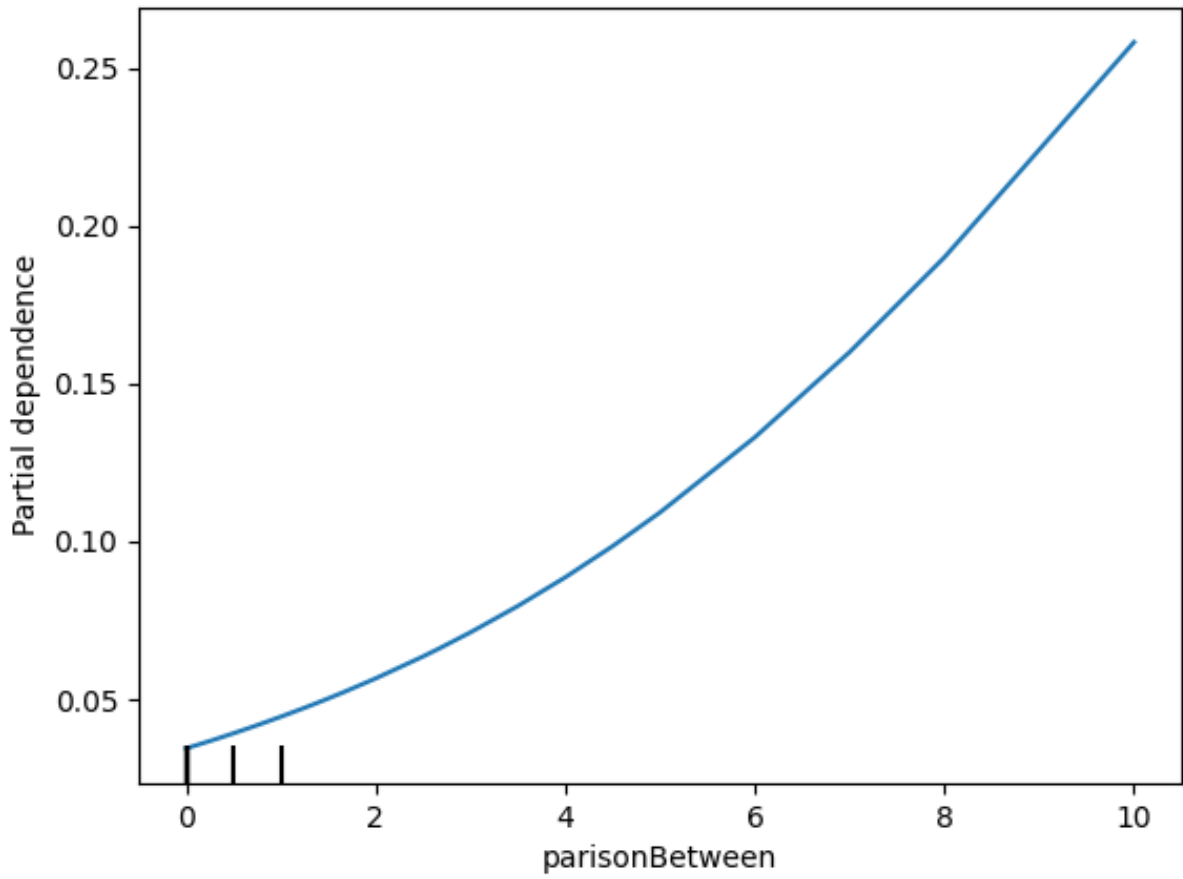


Figure C.15: Partial Dependency Plot for the *parisonBetween* feature as part of the logistic regression classifier based on *Dubremetz* with all novel features.

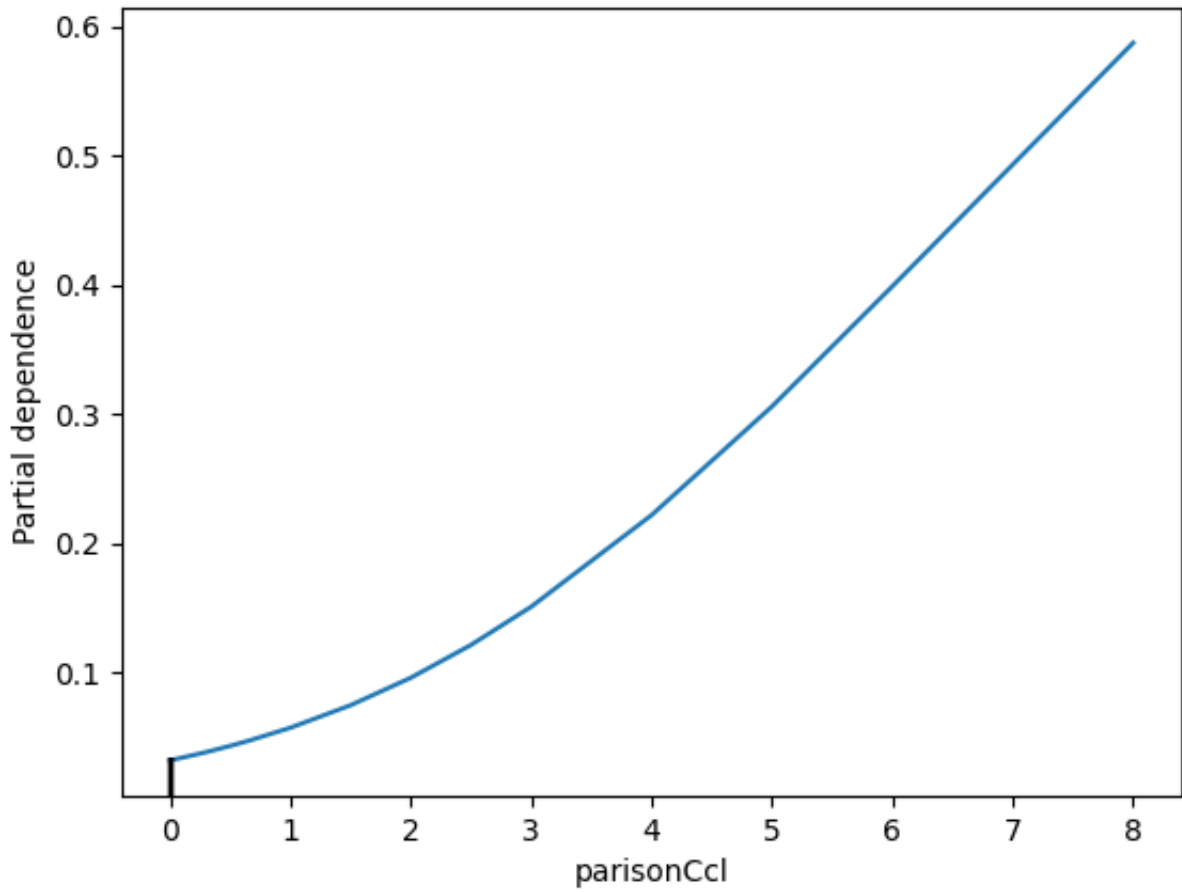


Figure C.16: Partial Dependency Plot for the *parisonConclusion* feature as part of the logistic regression classifier based on *Dubremetz* with all novel features.

C.7.2 Isocolon

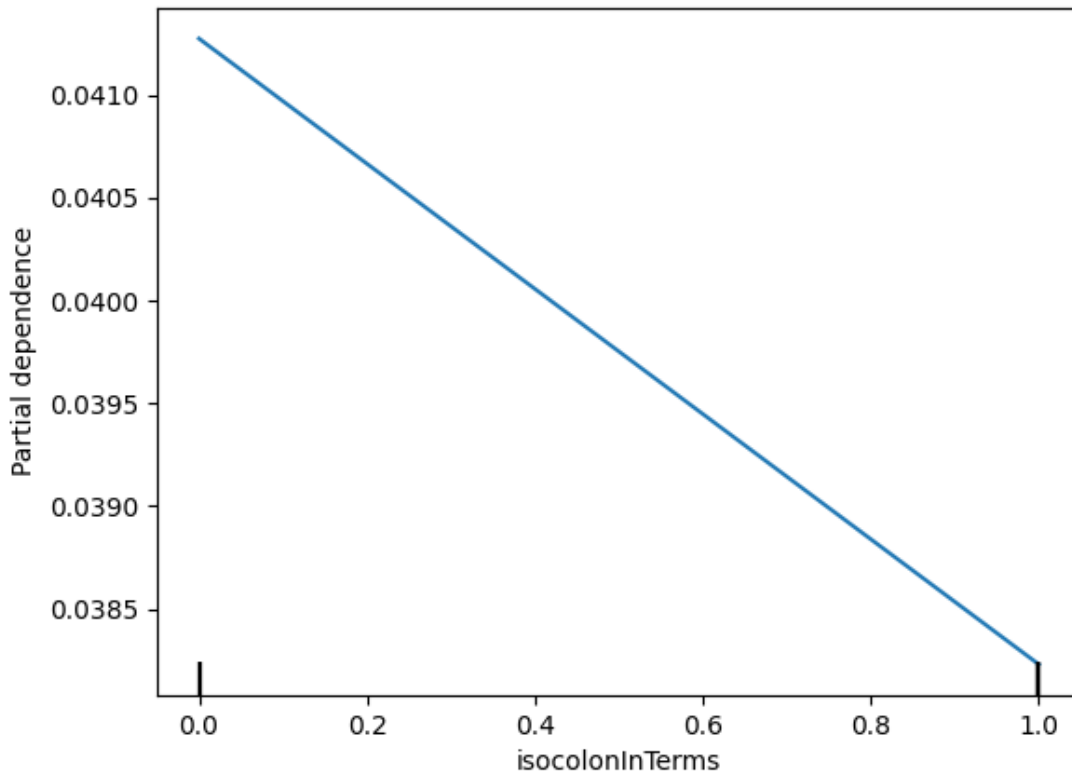


Figure C.17: Partial Dependency Plot for the *isocolonInTerms* feature as part of the logistic regression classifier based on *Dubremetz* with all novel features.

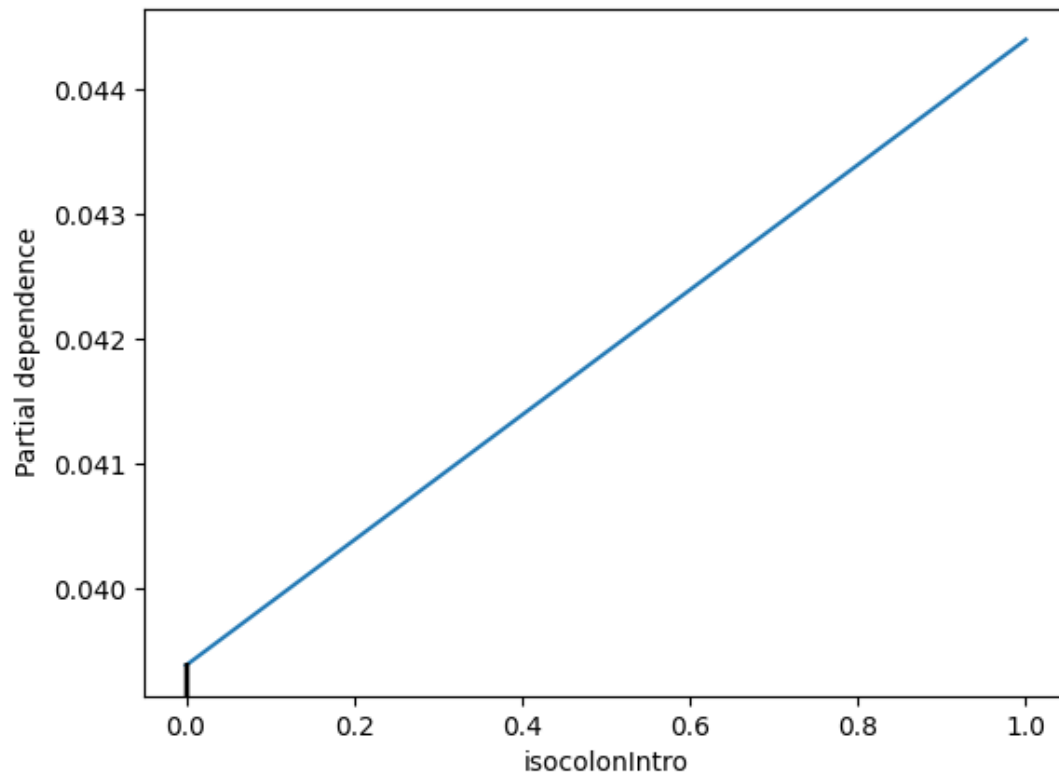


Figure C.18: Partial Dependency Plot for the *isocolonIntro* feature as part of the logistic regression classifier based on *Dubremetz* with all novel features.

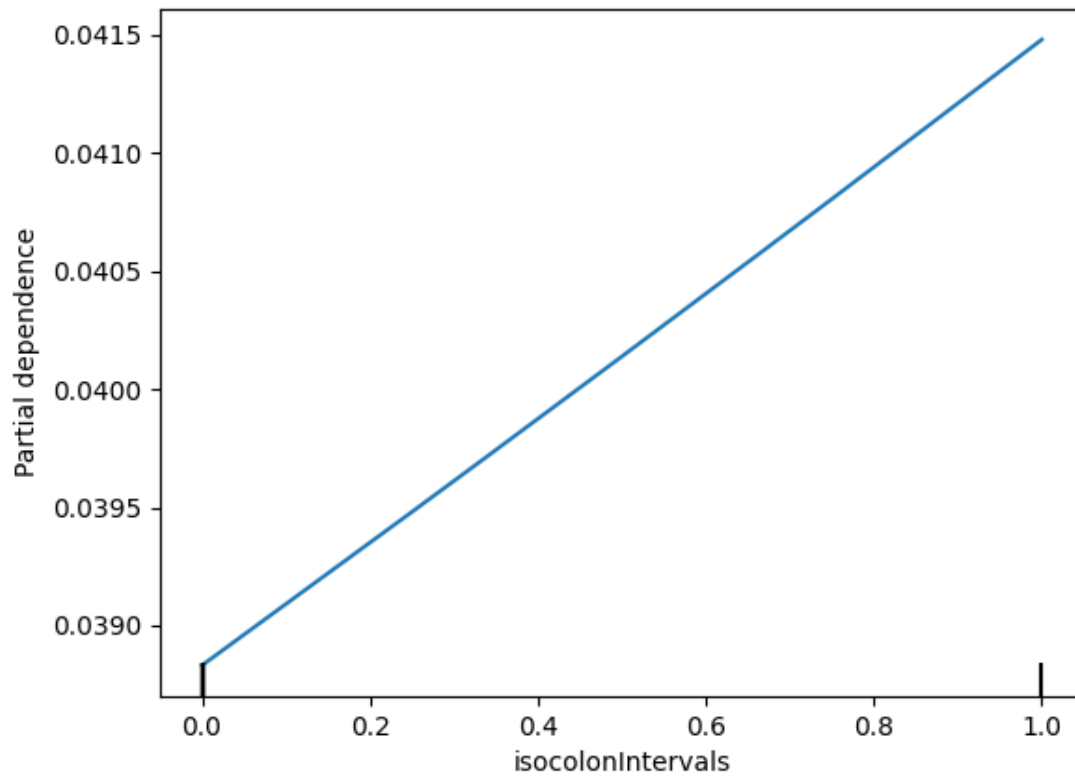


Figure C.19: Partial Dependency Plot for the *isocolonBetween* feature as part of the logistic regression classifier based on *Dubremetz* with all novel features.

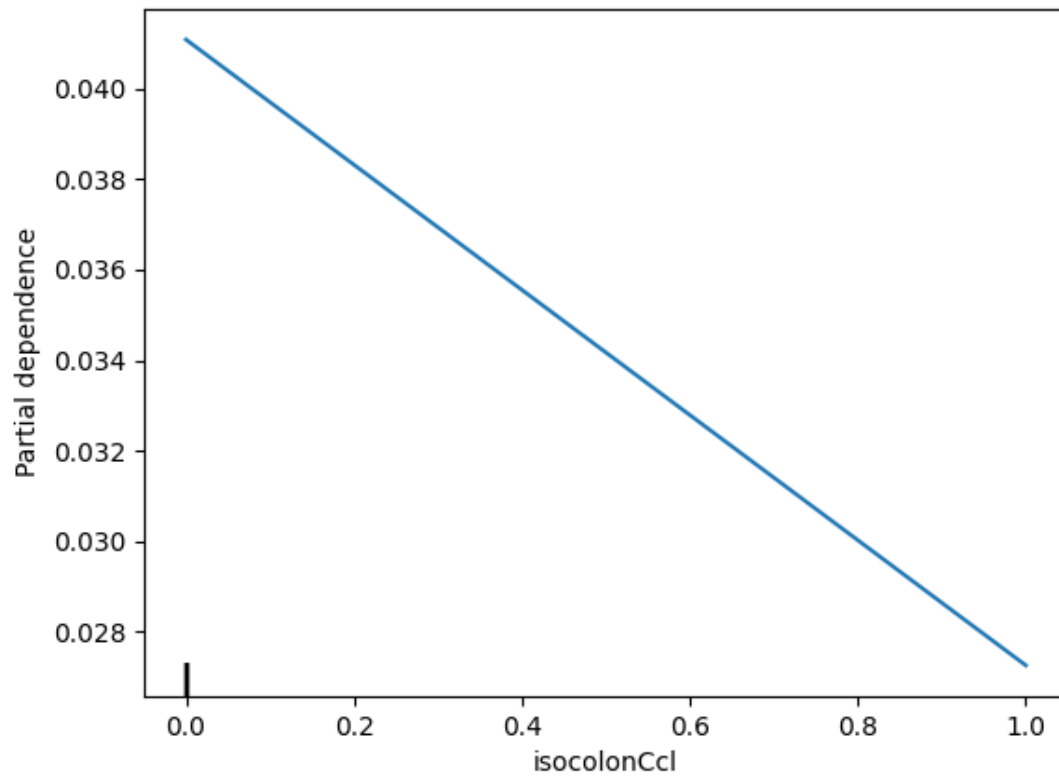


Figure C.20: Partial Dependency Plot for the *isocolonConclusion* feature as part of the logistic regression classifier based on *Dubremetz* with all novel features.

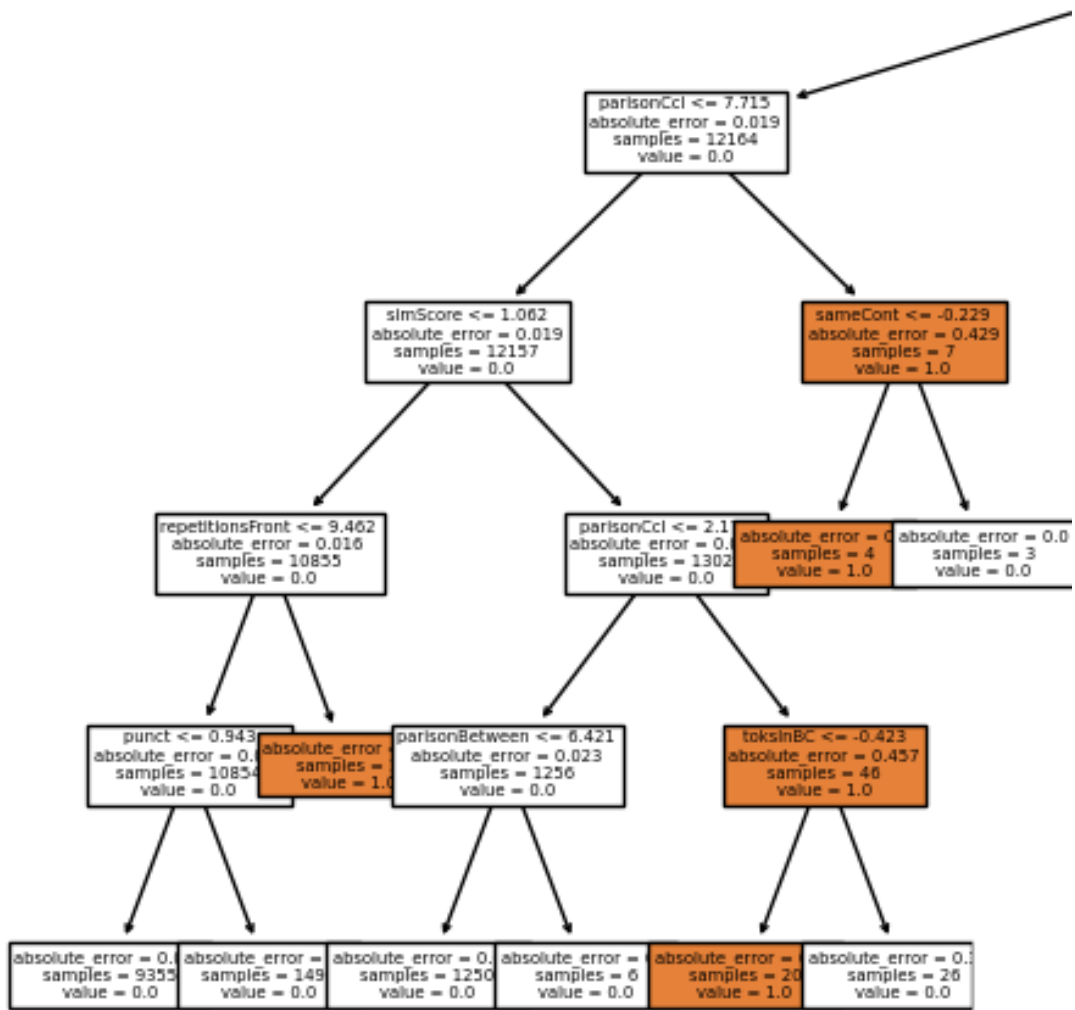


Figure C.22: Display of the lower left half of the regression tree based on *Dubremetz* with all novel features.

D Code

D.1 Initialising a Stanza pipeline

```
1 import stanza
2 from stanza.pipeline.core import DownloadMethod
3
4 stanza.download('en', processors='tokenize, lemma, pos, depparse')
5 processingPipeline = stanza.Pipeline('en', processors='tokenize, lemma,
  ↪ pos, depparse', download_method=DownloadMethod.REUSE_RESOURCES)
6
7 # fileContent contains the raw text
8 document = processingPipeline(fileContent)
9 documentWords = document.iter_words()
10 # documentWords is an Iterable containing Words
```

Bibliography

- [BM82] Nicolas Beauzée and Jean-François Marmontel. “Encyclopédie Méthodique. Grammaire & Littérature”. In: *Paris, Panckoucke* (1782) (cit. on p. 11).
- [Bec+01] Kent Beck et al. “Manifesto for agile software development”. In: (2001) (cit. on p. 32).
- [Ber23] Guillaume Berthomet. “Detecting Salient Antimetaboles in English Texts using Deep and Transfer Learning”. MA thesis. University of Passau & INSA Lyon, 2023 (cit. on p. 1).
- [Bur07] Gideon O Burton. “Silva rhetoricae”. In: *Brigham Young University* 14 (2007). URL: <http://rhetoric.byu.edu> (cit. on p. 11).
- [Cix06] Liu Cixin. *The Three-Body Problem*. Head of Zeus, 2006 (cit. on p. 45).
- [11a] *Collins English Dictionary*. 2011. URL: <https://www.collinsdictionary.com/dictionary/english/chiasmus> (visited on 11/26/2022) (cit. on p. 6).
- [11b] *Collins English Dictionary*. 2011. URL: <https://www.collinsdictionary.com/dictionary/english/antimetabole> (visited on 11/26/2022) (cit. on p. 9).
- [CMS10] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*. Vol. 520. Addison-Wesley Reading, 2010 (cit. on p. 20).
- [Dav08] Mark Davies. *The corpus of contemporary American English (COCA): 560 million words, 1990-present*. 2008. URL: <https://www.english-corpora.org/coca/> (visited on 11/26/2022) (cit. on p. 42).
- [Dub13] Marie Dubremetz. “Towards an automatic identification of chiasmus of words (Vers une identification automatique du chiasme de mots)[in French]”. In: *Proceedings of RECITAL 2013*. 2013, pp. 150–163 (cit. on pp. 15, 18, 36).

Bibliography

- [Dub17] Marie Dubremetz. “Detecting Rhetorical Figures based on repetition of words: Chiasmus, epanaphora, epiphora”. PhD thesis. Acta Universitatis Upsaliensis, 2017 (cit. on p. 24).
- [DN15] Marie Dubremetz and Joakim Nivre. “Rhetorical figure detection: The case of chiasmus”. In: *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*. 2015, pp. 23–31 (cit. on pp. 15, 19, 58).
- [DN16] Marie Dubremetz and Joakim Nivre. “Syntax matters for rhetorical structure: The case of chiasmus”. In: *Proceedings of the Fifth Workshop on Computational Linguistics for Literature*. 2016, pp. 47–53 (cit. on pp. 15, 21, 30).
- [DN17] Marie Dubremetz and Joakim Nivre. “Machine learning for rhetorical figure detection: More chiasmus with less annotation”. In: *Proceedings of the 21st Nordic Conference on Computational Linguistics*. 2017, pp. 37–45 (cit. on pp. 15, 24, 30, 47).
- [DN18] Marie Dubremetz and Joakim Nivre. “Rhetorical figure detection: Chiasmus, epanaphora, epiphora”. In: *Frontiers in Digital Humanities* 5 (2018), p. 10 (cit. on pp. 15, 27, 30).
- [Dum49] Alexandre Dumas. *Les trois mousquetaires*. Dufour & Mulat, 1849 (cit. on p. 45).
- [Dum10] Alexandre Dumas. *The Three Musketeers*. Translation anonymous. Simon & Schuster, 2010 (cit. on p. 45).
- [Dup80] Bernard Dupriez. *Gradus: les procédés littéraires: dictionnaire*. Union générale d’éditions, 1980. URL: https://archive.org/stream/BernardDupriezGradusLesProcedesLitteraires/%5BBernard_Dupriez%5D_Gradus__Les_procedes_litteraires_djvu.txt (visited on 11/26/2022) (cit. on pp. 8, 10).
- [Fah03] Jeanne Fahnestock. “Verbal and visual parallelism”. In: *Written Communication* 20.2 (2003), pp. 123–152 (cit. on p. 45).
- [Fau94] Markus Fauser. “Chiasmus”. In: *Ueding, Gert (Hg.): Historisches Wörterbuch der Rhetorik, Tübingen: Max Niemeyer* (1994), pp. 171–173 (cit. on p. 25).
- [Fri01] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232 (cit. on p. 63).

Bibliography

- [Gaw09] Jakub Jan Gawryjolek. “Automated annotation and visualization of rhetorical figures”. MA thesis. University of Waterloo, 2009 (cit. on p. 17).
- [Gol64] Oliver Goldsmith. *The Traveller: or a prospect of society*. 1764 (cit. on p. 7).
- [Gre+12] R. Greene et al. *The Princeton Encyclopedia of Poetry and Poetics: Fourth Edition*. Princeton Reference. Princeton University Press, 2012. ISBN: 9781400841424. URL: <https://books.google.fr/books?id=uKiC6IeFR2UC> (cit. on pp. 7, 13, 46).
- [Gro02] Mardy Grothe. *Never let a fool kiss you or a kiss fool you*. New York: Penguin Books, 2002. ISBN: 9781440621260. URL: <https://www.drmardy.com/chiasmus/book> (cit. on pp. 14, 41).
- [HD09] Randy Harris and Chrysanne DiMarco. “Constructing a rhetorical figuration ontology”. In: *Persuasive Technology and Digital Behaviour Intervention Symposium*. Citeseer. 2009, pp. 47–52 (cit. on pp. 12, 15).
- [HD17] Randy Allen Harris and Chrysanne Di Marco. “Rhetorical figures, arguments, computation”. In: *Argument & Computation 8.3* (2017), pp. 211–231 (cit. on pp. 45, 80).
- [Har+17] Randy Allen Harris et al. “A cognitive ontology of rhetorical figures”. In: *Cognition and Ontologies* (2017), pp. 18–21 (cit. on p. 15).
- [Har+18] Randy Allen Harris et al. “An annotation scheme for rhetorical figures”. In: *Argument & Computation 9.2* (2018), pp. 155–175 (cit. on pp. 12, 22, 23, 34, 36).
- [Ho95] Tin Kam Ho. “Random decision forests”. In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282 (cit. on p. 61).
- [Hro11] Daniel Hromada. “Initial experiments with multilingual extraction of rhetoric figures by means of PERL-compatible regular expressions”. In: *Proceedings of the Second Student Research Workshop associated with RANLP 2011*. 2011, pp. 85–90 (cit. on pp. 17, 28).
- [Kel+10] Ashley R Kelly et al. “Toward an ontology of rhetorical figures”. In: *Proceedings of the 28th ACM International Conference on Design of Communication*. 2010, pp. 123–130 (cit. on pp. 7, 11).

Bibliography

- [KMG22] Ramona Kühn, Jelena Mitrović, and Michael Granitzer. “GRhOOT: Ontology of Rhetorical Figures in German”. In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. 2022, pp. 4001–4010 (cit. on pp. 15, 16).
- [LR76] Hyafil Laurent and Ronald L Rivest. “Constructing optimal binary decision trees is NP-complete”. In: *Information processing letters* 5.1 (1976), pp. 15–17 (cit. on p. 61).
- [Man+14] Christopher D Manning et al. “The Stanford CoreNLP natural language processing toolkit”. In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 2014, pp. 55–60 (cit. on pp. 21, 30).
- [22] *Merriam-Webster.com Dictionary*. 2022. URL: <https://www.merriam-webster.com/dictionary/chiasmus> (visited on 11/26/2022) (cit. on p. 7).
- [Mit+17] Jelena Mitrović et al. “Ontological representations of rhetorical figures for argument mining”. In: *Argument & Computation* 8.3 (2017), pp. 267–287 (cit. on p. 14).
- [MM13] Miljana Mladenović and Jelena Mitrović. “Ontology of rhetorical figures for Serbian”. In: *International Conference on Text, Speech and Dialogue*. Springer. 2013, pp. 386–393 (cit. on p. 15).
- [Nak+18] Hiroki Nakayama et al. *doccano: Text Annotation Tool for Human*. Software available from <https://github.com/doccano/doccano>. 2018. URL: <https://github.com/doccano/doccano> (cit. on p. 34).
- [Nam00] Fiammetta Namer. “FLEMM: un analyseur flexionnel du français à base de règles”. In: *Traitement automatique des langues* 41.2 (2000), pp. 523–547 (cit. on p. 18).
- [Nor71] Helge Nordahl. “Variantes chiasmiques. Essai de description formelle”. In: *Revue Romane* (1971) (cit. on p. 8).
- [Pas+19] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019) (cit. on p. 38).
- [Ped+11] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830 (cit. on pp. 24, 45).

Bibliography

- [PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543 (cit. on p. 38).
- [Qi+20] Peng Qi et al. “Stanza: A Python Natural Language Processing Toolkit for Many Human Languages”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2020. URL: <https://nlp.stanford.edu/pubs/qi2020stanza.pdf> (cit. on p. 30).
- [Rab08] Alain Rabatel. “Points de vue en confrontation dans les antimétaboles PLUS et MOINS”. In: *Langue française* 4.160 (2008), pp. 20–35 (cit. on pp. 10, 12).
- [RDH16] Sebastian Ruan, Chrysanne Di Marco, and Randy Allen Harris. “Rhetorical Figure Annotation with XML.” In: *CMNA@ IJCAI*. 2016, pp. 23–33 (cit. on pp. 13, 14).
- [SYY75] Gerard Salton, Chung-Shu Yang, and CLEMENT T Yu. “A theory of term importance in automatic text analysis”. In: *Journal of the American society for Information Science* 26.1 (1975), pp. 33–44 (cit. on p. 25).
- [Sch94] Helmut Schmid. “Part-of-speech tagging with neural networks”. In: *arXiv preprint cmp-lg/9410018* (1994) (cit. on p. 19).
- [Sch+21] Felix Schneider et al. “Data-Driven Detection of General Chiasmi Using Lexical and Semantic Features”. In: *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. 2021, pp. 96–100 (cit. on pp. 15, 24, 39, 47, 58).
- [Seu68] Seuss. *Horton hatches the egg*. Vol. 1. Random House Books for Young Readers, 1968 (cit. on p. 11).
- [Str11] Claus Walter Strommer. “Using rhetorical figures and shallow attributes as a metric of intent in text”. In: (2011) (cit. on p. 28).
- [Tho95] Ian H Thomson. *Chiasmus in the Pauline letters*. Vol. 11100. Sheffield Academy Press, 1995 (cit. on p. 7).

Bibliography

- [Wan+22] Yetian Wang et al. “Towards a Unified Multilingual Ontology for Rhetorical Figures:” in: *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. Valletta, Malta: SCITEPRESS - Science and Technology Publications, 2022, pp. 117–127. ISBN: 9789897586149. DOI: 10.5220/0011524400003335. URL: <https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0011524400003335> (visited on 01/18/2023) (cit. on p. 15).
- [Wil71a] Augustus Samuel Wilkins. *Oxford English Dictionary*. 1871. URL: <https://www.oed.com/view/Entry/31503> (visited on 11/26/2022) (cit. on p. 6).
- [Wil71b] Augustus Samuel Wilkins. *Oxford English Dictionary*. 1871. URL: <https://www.oed.com/view/Entry/8673> (visited on 11/26/2022) (cit. on pp. 9, 12).

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich diese Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe und alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind, sowie, dass ich die Masterarbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.

Passau, January 19, 2023



Yohan Meyer