Automated Annotation and Visualization of Rhetorical Figures

by

Jakub J. Gawryjołek

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Linguistic annotation provides additional information asserted with a particular purpose in a document or other piece of information. It is widely used in various fields, from computing and bioinformatics, through imaging, to law and linguistics. There is also a clear distinction between *what* is communicated through the written/spoken natural language and *how* this is passed on. A new problem of linguistic annotation is the annotation of classical rhetorical figures — patterns of text in which a characteristic syntactic form modifies the standard meanings of words, and leads to a change or an extension of meaning. Rhetoric studies the effectiveness of language comprehensively, including its emotional impact, as much as its propositional content. The annotation of rhetorical figures is therefore important not only for the linguistic point of view, but also for discovering different styles of writing, purpose and effect of written documents, and for better natural language understanding in general.

The purpose of this thesis is the automated annotation of rhetorical figures. In the thesis we primarily focus on the figures of repetition, which include the repetition of words, phrases, and clauses. Additionally, we also describe the work we have done on the detection and annotation of figures of parallelism, as well as those that pertain more to the semantics than to the syntax, or positioning. We have developed a rhetorical figure annotation tool dubbed JANTOR (Java ANnotation Tool Of Rhetoric), which enables manual and automated annotation of files in HTML format. We have applied a lexicalized probabilistic context-free grammar parser for the recognition of the figures of repetition. We also describe a simple parse tree distance used for calculating the difference between similarly structured phrases, which is necessary for the recognition of some of the figures of parallelism. Moreover, we have applied the semantic relationships contained in the WordNet lexical database and extended Porter stemmer algorithm for finding derivationally related words. Finally, we present a method for finding pairs of words which are ordinarily contradictory, which is crucial for detecting the interesting figure of speech: *oxymoron*. For this purpose typed dependency grammars together with WordNet are used. The experiments we have conducted on the detection of selected subset of rhetorical figures have yielded very promising results.

Lastly, we present the visualization of the occurrences of the figures and comparison between 14 American presidents' inaugural addresses including the most recent one by President Barack Obama. The provocative results of this comparison show that a) automated analysis of meaningful rhetorical information is possible and tractable, and b) help us with understanding what creates a successful orator.

Acknowledgments

First, I would like to thank my supervisor, Chrysanne DiMarco, for all her support throughout my graduate studies and for all the insightful discussions that greatly helped me with the completion of this thesis.

I would also like to thank Randy Harris for many suggestions pertaining to the rhetorical part of the work, from which I have benefited in a large extent.

Many thanks to the members of the Inkpot Natural Language Research Group for their suggestions concerning the rhetorical background as well as the design of the annotation tool. Thanks to all of the following for their input:

Olga Gladkova, Elena Afros, Doug Mulholland, Matthew Skala, Claus Strommer, and Steve Banks.

For their assistance and willingness to accompany me remotely by inviting me to many Skype conferences during my stay in Waterloo, a special thanks to my brother Łukasz and his wife Dominika.

Additionally, I would like to thank my roommates for the willingness to participate in my sporadic and short breaks: Marta Sitek, Jodian Fairclough, Jakub Schmidtke, and Krzysztof Hebel. I am also grateful to all my current and previous office mates, who created the great atmosphere of both work and fun: Sharon Wulff, Abhinav Bahadur, Ting Liu, Tyler Lu, Ke Deng, Zhi Xu and Derek Wang.

I would especially like to thank Teresa Luu, who was patient and understanding enough to bear me being extremely busy over the last couple of months. You have been my greatest motivation and inspiration.

Dedication

I would like to dedicate this thesis to my parents. They have instilled in me the confidence and the determination to achieve set goals. Without their great support, understanding, and encouragement to reach further, over all the years of my education, completing this thesis would not have been possible.

Contents

List of Tables **ix**

List of Figures **xi**

1 Introduction **1**

1.1 What is Rhetoric? 1

1.1.1 Classical Rhetoric 2

1.1.2 What is a Rhetorical Figure? 3

1.2 Detecting Figures—Challenges 3

1.2.1 Classification of Figures 4

1.2.2 Other Problems 4

1.3 Annotation of Rhetorical Figures 5

1.4 Organization of the Thesis 6

2 Related Work **7**

2.1 Structure of Text 7

2.1.1 Rhetorical Structure Theory 7

2.1.2 RST Analysis and Annotation Tools 8

2.2 Linguistic Annotation 9

2.2.1 Annotation Tools 10

2.3 WordNet—Semantic Relatedness 10

2.4 Text Visualization 11

2.5 Citation Classification 12

3 Detection of Figures **13**

3.1 Detection Framework 13

3.1.1 Syntactic Units 13

3.1.2 Why a lexicalized PCFG parser? 14

3.1.3 Parsing problems 15

3.2 Schemes—Figures of Repetition 16

3.2.1 Detection of Repeating Words 16

3.2.2 Finding Rhetorical Figures 19

3.2.3 Repetition of Derivationally Related Words 26

3.2.4 Merging overlapping figures 30

3.3 Schemes—Figures of Parallelism 31

3.4 Tropes 37

3.5 Figures hierarchy 43

4 JANTOR—The Annotation Tool **45**

4.1 Annotation 45

4.1.1 Input and Output 46

4.1.2 Annotation Schema 46

4.2 Navigation 48

4.3 Detection 50

4.4 Visualization 51

4.4.1 Choice of colours 52

5 Results and Evaluation **54**

5.1 Testing documents 54

5.1.1 Experiments 54

5.2 Application: Annotation and Visualization of Presidential Speeches 75

5.2.1 Intensity—Washington versus Obama 76

5.2.2 Placement 78

5.2.3 Rhetorical Evolution? 82

5.2.4 General Observations 83

6 Conclusions and Future Work 84

6.1 Automated Annotation 85

6.2 Analysis 87

6.3 Future Work 87

6.3.1 Coach Notes Annotation 88

APPENDICES 91

A Glossary 91

B JANTOR - Step by Step 93

B.1 Graphical Components 93

B.2 Manual Annotation 95

B.3 Automated Annotation 96

B.4 Visualization 97

B.5 Configuration 97

B.6 Annotation File 98

C Penn Treebank Project Tag Set 99

References 101

List of Tables

3.1 Repeated phrases 18

3.2 Formalism for figures of repetition 18

3.3 Polyptoton examples 27

3.4 Different anaphoras example 31

3.5 Tags equivalence classes 34

3.6 Example oxymoron expressions 38

3.7 Possible grammatical relations between word pairs used for oxymoron detection 39

3.8 Word to word relation paths 41

4.1 Colours of the rhetorical figures 53

5.1 Repetitions of words in reverse grammatical order 67

5.2 Number of figures in the inaugural addresses 76

C.1 The Penn Treebank POS tagset[46] 100

List of Figures

3.1 Parse trees of two example phrases 15

3.2 Example phrase parse tree 33

3.3 Parse trees of two phrases 37

3.4 The tree of WordNet relations used for oxymoron detection 43

4.1 Annotation of anaphora 50

4.2 Visualization of rhetorical figures in text 51

5.1 Parse tree of a sentence with anaphora for minimum phrase length equal 2 58

5.2 Incorrect detection of anaphora 59

5.3 Example of incorrect parsing with respect to the detection of an epistrophe 61

5.4 Example of hard-to-detect epistrophe. 62

5.5 Example of wrongly categorized epanalepsis 64

5.6 Undetected example of anadiplosis 65

5.7 Example of missed anadiplosis 66

5.8 Parse trees of isocolon compound phrases 71

5.9 Positioning of polyptoton in Barack Obama's and George Washington's Inaugural Addresses 78

5.10 Positioning of rhetorical figures in Obama's and Washington's Inaugural Addresses 79

5.11 Comparison of isocolon positioning 80

5.12 Anaphora comparison at the end of the inaugural addresses by Bill Clinton . . . 82

5.13 Polyptoton, polysyndeton, and antimetabole placement in Clinton's first inaugural address . . . 83

B.1 JANTOR - Java ANnotation Tool Of Rhetoric . . . 94

B.2 Annotation panel . . . 95

B.3 Navigation panel . . . 95

B.4 Control panel . . . 96

B.5 New figure selection panel . . . 96

Chapter 1

Introduction

1.1 What is Rhetoric?

Rhetoric is the art of using language as a means to persuade, influence the behaviour or attitude of others. The word *rhetoric* itself originates from the Greek *rhētorikós*, "oratorical"—skilled in speaking[57]. More specifically, its etymology shows that it pertains to the notion of "words" and "speech"[16]. Craig Smith in his book about rhetoric and human consciousness observes that different strategies of persuasive communication are closely connected to our everyday life and are absolutely vital in innumerable situations[60]. We will have a look at many aspects of rhetoric that make it more persuasive and influential on the audience form of communication than the *informative* and *entertaining* forms.

Informative communication occurs when people share knowledge about the world in which they live. The main purpose of entertaining communication is to maintain our attention. Finally, rhetorical communication by its persuasion of ourselves and people around us reaches beyond the first two forms[60]. All three divisions might also be seen as dimensions of discourse.

According to Aristotle, the way we persuade ourselves is the way we persuade others. Very often our intrapersonal rhetoric transforms into the interpersonal swaying. Additionally, as Smith notices, it is closely related to epistemology—it contains a so-called "making known" function which explains what we discover[60]. It has the power of articulating ideas or delivering information in such a way that is sometimes not contained in any forms of logical reasoning. When logical discourse becomes constrained by the requirements imposed by its premises, rhetoric provides the means to communicate in a cohesive and sensible way.

Smith also observes that rhetoric is intrinsic to our existence as it deals with our emotional and psychological states. The way we perceive the surrounding world is to a very large extent influenced by our mood and current state of mind. Therefore, the form of communication that takes cognition into account is more effective and definitely more influential than just the raw transfer of information. Moreover, humans are decision-making creatures who persuade themselves all the time, who change their minds, make decisions, and debate. Mastering rhetoric is crucial for becoming a successful speaker and decision-maker.

1.1.1 Classical Rhetoric

Rhetoric has been with us since ancient times. In its classical form, as we mentioned earlier, it was associated mainly with persuasive discourse[16] and was divided into five parts: *inventio*, *dispositio*, *elocutio*, *memoria*, and *pronuntiatio*.

Inventio related to the ability of the orator to come up with intelligent argumentation in whatever speech he undertook, mainly by relying on his talent (according to Cicero). It was connected with a way of finding persuasive arguments. *Dispositio* is the arrangement of arguments in a written or spoken discourse. Cicero and Quitilian provided six parts to this arrangement[16]:

1. Introduction (exordium);
2. Statement or exposition of the case (narratio);
3. Outline of the points or steps in the argument (divisio);
4. Proof of the case (confirmatio);
5. Refutation of the opposing arguments (refutatio);
6. Conclusion (peroratio).

The third part is the *elocutio*. The word is associated with the act of speaking, but for a classical rhetorician it meant “style”. There are many different aspects of elocutio that form the way the text is delivered—e.g., classification of styles, choice of words. However, the most important from the point of view of this thesis is the *composition or arrangement of words* in phrases and clauses. The intrinsic part of *elocutio* is thus the rhetorical figures, which have gained a lot of attention during the study of rhetoric. We introduce the notion of a rhetorical figure (also known as a figure of speech) in the next section.

The second last part of a persuasive discourse is *memoria* which, as the name suggests, concerns memorizing speeches. There has been little attention paid to this part of rhetoric, as not much from the theoretical point of view can be said about the process of memorizing. However, the need to memorize speeches did influence the structure of discourse to some extent.

Finally, the last part of classical rhetoric was *pronuntiatio*, which concerned the way a text was delivered. As with *memoria*, *pronuntiatio* has been significantly neglected in the rhetoric textbooks, but all the experts unanimously agree that it is important in the persuasive process.

1.1.2 What is a Rhetorical Figure?

Figures of speech are the most revealing features of the third part of rhetoric—*elocutio*. The traditional definition of a rhetorical figure is provided by Corbett[16]. He defined it as an artful deviation from the normal or ordinary manner of expression. McQuarrie *et al.*[47] provide more formality to this deviation, stating that a rhetorical figure occurs when an expression deviates from expectation, but the expression is not rejected as nonsensical or faulty.

Depending on the situation there can be different ways of expressing a proposition which will be most persuasive for an audience[47]. More importantly, if "to sway" is the most significant goal of the speech or a written work, then the way one says something might have a more powerful effect than the actual content. In general, figures of speech constitute the characteristics of a person's linguistic style. If not used excessively, they remain apt and fresh, and can contribute to a large extent to the clarity, liveliness, and interest of one's style[16]. They are beautiful stylistic devices making any piece of written and spoken word more interesting and lively, and helping the rhetor direct the attention of the reader or listener.

1.2 Detecting Figures—Challenges

Why is the detection of rhetorical figures both important and difficult? A basic premise for rhetoric is the close relatedness between semantics and the means of communication. In other words, the way one says something conveys meaning as much as what one says. Rhetoric studies comprehensively how effective the language used is, including its emotional impact, as much as its propositional content. The annotation of rhetorical figures is therefore important not only for the linguistic purposes, but also for discovering different styles of writing, purpose, and effect of written documents, and in general for better natural language understanding. Additionally, this kind of annotation may greatly contribute to *genre* detection. Different types of writing make use of different kinds of rhetorical tools. For example, McQuarrie *et al.*[47] suggest that in advertising language certain rhetorical figures cause more favourable customer responses than others, making some ads more memorable and successful. The detection of figures of speech though is a very complex process due to numerous reasons, which we briefly outline below.

1.2.1 Classification of Figures

There is no definite classification of the figures of speech. Over time they have been organized in a variety of different ways in order to make sense of them and to learn their various qualities. Various kinds of groupings for the figures can be found; however, one of the first and at the same time the simplest categorization divides them into two broad categories, *schemes* and *tropes*[1]. *Trope* is the usage of a word in a different way than what is considered its literal principal form. *Scheme*, on the other hand, is a deliberate deviation from the ordinary arrangement of words. *Speech* and *thought* are other broad categories to which rhetoricians relate figures of speech. The former concerns the verbal expression, whereas the latter pertains to idea. Finally, the figures can be grouped according to function or strategy:

- Figures of amplification;
- Figures of grammar;
- Figures of omission: omit something, e.g., a word, words, phrases, or clauses from a sentence;
- Figures of repetition: repeat word, words, phrases, clauses or ideas;
- Figures of wordplay;
- and many more…

1.2.2 Other Problems

The problem of detecting figures of speech is challenging not only due to many possible classifications. First of all, it is impossible to annotate all the figures because of their abundance. Many of them (even from the *scheme* category) do not necessarily follow a specific pattern that could be easily used for simple detection. Furthermore, some of the figures, for example, *antithesis*, pertain to semantics (antithesis is the juxtaposition of opposing or contrasting ideas). Whenever the study of natural language meaning is involved, lots of new computer-related problems emerge.

Another set of problems lies on the computational level of the annotation. First, many figures are spread over not only numerous words but also phrases, clauses, or sentences. Thus modules enabling the discovery of boundaries of these syntactic units have to be devised. Figures of speech are very often composed of several words, syllables, or phrases. Additionally, these parts may overlap, may be included in one another, or occur separately. All of these issues have to be taken into account in order to depict the figures in a neat and understandable fashion in text. In general, the representation of textual content in a graphical form introduces issues concerning information visualization, such as: How many items should be displayed? What kind of representation of marked pieces should be chosen? Which colours should be selected so that everything is clear and unambiguous? And many others.

1.3 Annotation of Rhetorical Figures

Rhetoric is pervasive in language use and has been studied for millennia. The annotation of rhetorical features, and more specifically the annotation of rhetorical figures in an automated way, has not yet been extensively researched, if at all. The creation of a tool that would enable automated as well as manual annotation of figures of speech is the main aim of this thesis.

We have defined the concept of a rhetorical figure. Now, we need to answer the basic question: what is a linguistic annotation, and how might it pertain to rhetoric? In general, an annotation is an addition made to some information in a piece of writing, video, or other medium that provides some extra explanation. In linguistic and pragmatic analysis, annotations add information to raw language data about the linguistic form[10].

The primary aim of this thesis is the automated annotation of the figures of repetition. To accomplish this goal we have created a framework for finding word and phrase repetitions in certain fragments of text. With the use of a lexicalized probabilistic context-free grammar parser (PCFG) we have created a module that detects different figures at various positions in phrases, clauses, and sentences. We also present our initial work on the detection of figures of parallelism. We have created a method for the comparison of phrase and clause parse trees, which is a necessary step in the process of identification of similarly structured text fragments. We present several algorithms used to achieve this goal. Finally, we have taken the initial steps towards the automated detection of rhetorical figures which concern a deviation from the literal meaning rather than the modification of the arrangement of words. Our two-step process, including: (a) the extension of Porter stemming algorithms for finding derived forms of a word, and (b) the application of WordNet lexical database, produces satisfactory results and brings to light several problems that have to be taken into consideration. Additionally, we focus on how to present the discovered pragmatic evidence in an approachable and easily understandable way, which helps with the meaningful analysis of rhetorical information.

Finally, we perform the analysis of U.S. presidential inaugural addresses in terms of usage of rhetorical figures. More specifically, we try to investigate what kinds of figures of speech are most frequently used to achieve the communicative purpose of the speech. We also look at the positioning of figures in these texts. The results of this analysis help us with understanding why the first official speech of a president stands out from not only other written texts but also from other political speeches.

1.4 Organization of the Thesis

Chapter 2 reviews the related work concerning linguistic annotation tools, rhetorical structure theory, text visualization, and the applications of the WordNet lexical database to the determination of the semantic relatedness between words, which appear very useful for the detection of figures from the *trope* category.

Chapter 3 presents the methodology of our figure detection system. The general framework as well as the identification algorithms and procedures for the individual figures of speech are described.

Chapter 4 describes our implemented annotation tool. Various features of the system, its components and important design decisions, are briefly overviewed. In Chapter 5 the evaluation of the annotation tool is presented. The application of the tool to the prepared small corpora as well as real-life examples is described. Some possible ways for improvement of the tool are also highlighted.

Chapter 6 concludes the thesis and proposes directions for future research.

Chapter 2

Related Work

To the best of our knowledge (according to Google Scholar[29]), there has not been any work done specifically on the automated annotation of rhetorical figures. However, there has been substantial research conducted in some of the related areas in Computer Science and Rhetoric to which this thesis pertains. First, we briefly describe one of the most pervasive theories concerning the structure of text. We focus on some rhetorical structure annotation tools in this part. Then we overview the existing linguistic annotation tools followed by the research done on WordNet and semantic relatedness between words. Finally, we describe some of the approaches to text visualization.

2.1 Structure of Text

2.1.1 Rhetorical Structure Theory

Mann and Thompson

Rhetorical Structure Theory (RST) is a theory of text organization introduced in the 1980's, described by Mann and Thompson in [44]. One of the reasons why RST has been successful over many years is that it has been applied to many areas of natural language processing (NLP): discourse analysis, theoretical linguistics, psycholinguistics, computational linguistics[43]. The latter field uses RST for general NLP tasks such as parsing the structure of text and creating coherent texts[62].

The theory relies on identifying the rhetorical relations between parts of a text, which suffice to comprehensively analyze the discourse structure of most English texts. The main function of the relations is to present conceptual connections between two neighbouring parts of text. The two main parts of these relations are the fields that: (a) represent the effect to be achieved, and (b) provide constraints that have to be satisfied in order to achieve this effect[62]. Additionally, Mann and Thompson eliminate the necessity of linguistic devices as indicators of the relations, which differentiates them from other theories on text structure as, for example, Grosz and Sidner's (GST)[30]. Worth noticing is the notion of *nucleus* and *satellite*: spans of text, more and less central parts respectively. The majority of texts are structured through the relationships between these two components.

Very important to this thesis is Mann and Thompson's definition of a *text span*. They consider the clause as the minimal unit of text organization. The size of the unit is arbitrary, but it should be functionally independent. What distinguishes our approach from theirs is that we work with smaller spans than the clause. Additionally, at least as critically, we also focus on formal attributes, not (just) conceptual attributes. The definition of many rhetorical figures requires the thorough examination of not only clauses and sentences, but also phrases and words. Another important point is the requirement imposed on the relations with respect to the text spans, namely, they must exist between adjacent, non-overlapping units. In our approach, figures might occur in not necessarily adjacent units, or the units might overlap.

2.1.2 RST Analysis and Annotation Tools

O'Donnell's RSTTool

O'Donnell[50] created a tool, dubbed RST Tool, which is a graphical interface for marking up the structure of a text. The system can be used for both segmentation of text and graphical linkage between the segments and RST-tree.

Marcu's RSTTool

Marcu[20] added some useful features to O'Donnell's original project, but as his tool is an extension we describe them together. The tools are easy-to-use graphical interfaces that enable the indication of rhetorical dependencies between segments of texts. As they rely on RST, the main syntactical unit of text is a clause. Marcu's tool was used by a team of linguists at the Information Sciences Institute of the University of Southern California to create a corpus of annotated Wall Street Journal articles. The scope of this thesis on the other hand is the detection and annotation of figures of speech. We believe that the comparison of rhetorical relations between text spans and the usage of rhetorical figures may reveal some very interesting dependencies.

ConAno

Stede and Heintze[61] developed a rhetorical structure annotation tool which allows for efficient, interactive markup of relations, scopes, and connectives. They take a two-step approach towards the annotation. The first step is the annotation of connectives, their scopes (the two related text spans), and possibly the signalled relation. The second, more difficult, step is the determination of the relations between larger segments. The latter procedure is performed by a human annotator. Such an approach is a good way to tackle the usually hard and ambiguous problem of annotating pragmatic evidence. The lack of corpora with annotated figures of speech forced us to create a tool that, as in the case of ConAno, enables both automated and manual markup of rhetorical figures.

Stede and Heintze make a very important observation with respect to rhetorical analysis. They note that the 'ideal' discourse analysis proceeds incrementally from left to right. However, their research revealed that the strict left-to-right way of looking at a text is highly impractical, for we quite often learn a lot about the argumentative structure after examining a large piece of text. Although the discovery of rhetorical relations is not within the scope of this thesis, the above observation suggested that we should relax the constraints of the definitions of certain rhetorical figures. As we will see in the next section, some of the figures of repetition mention successive phrases, clauses, or sentences. However, a text containing figures of speech might have the same persuasive power even when these figures do not meet the exact restrictions imposed by the definition. Thus, looking at text in terms of rhetorical figures detection requires a broader perspective.

2.2 Linguistic Annotation

Linguistic annotation is any additional information provided for description or explanation to raw language data. As this information is for the text analyst, the annotations are intended to facilitate various analyses of text. Of course, here we only consider the textual annotations of textual data. There is a plethora of linguistic annotation tools available on the Internet and mentioned in the research literature. Below we present the ones that are most relevant to our task.

2.2.1 Annotation Tools

Ontobroker

Erdmann *et al.*[24] developed Ontobroker, a knowledge-base–supported annotation tool. Although designed for a different purpose, their annotation tool is similar in some aspects to our system. Essentially, Ontobroker is an ontology-based semi-automatic annotation system for natural language texts. The detection of figures of repetition does not require any ontological knowledge; however, in order to successfully identify *tropes* we will definitely have to incorporate an ontology of rhetorical figures[32] into our annotation tool. The fully-automated annotation of the semantically complex figures of speech requires both the creation of a large annotated training corpus and a module for learning domain ontologies from text.

Another important component of Ontobroker is the *lexical analyzer* which includes word and domain lexicons. Over 120,000 word entries and more than 12,000 subcategorization frames describe information used for lexical analysis. The lexicon of word forms would be extremely useful for the detection of rhetorical figures such as polyptoton. Furthermore, the notion of a subcategorization frame might be very useful for a successful determination of candidate oxymoron word pairs. In this sense, Ontobroker differs from our solution, as we concentrate on the grammatical relations of typed dependencies.

LinguaLinks

LinguaLinks is “an electronic productivity support system for language workers”, which among numerous utilities includes linguistics tools[41] to do word analysis. We focus only on the ones that pertain to morpheme analysis—the task salient to the efficient annotation of, for instance, polyptotons. However, as we will see in Chapter 5 different word forms occur in many rhetorical figures. Three tools—*Wordform Inventory* editor, *Analysis* editor and *Morphology Explorer*—facilitate the discovery of morphemes and the creation of a word forms and glosses database.

2.3 WordNet—Semantic Relatedness

Simone & Kazakov

Simone and Kazkov[63] propose a document search technique that uses the lexical database WordNet[32] to cluster search results according to the words that modify the original search term in the text. The main focus of their work is the examination of the importance of synonymy and antonymy semantic relations present in WordNet. The main idea behind their method is to group together these documents that share the same noun or verb modifiers—adjectives and adverbs, respectively. Additionally, they make use of a similarity relationship between these modifiers to extend the clusters and then antonymy for refinement. Although very loosely related to our topic, we apply the similar technique of extending word 'meaning' by using its synonyms, and then apply the antonymy relations in order to capture possible contradictions between pairs of words. This approach helps us with the identification of the trope figure oxymoron.

Marneffe et al.

Marneffe *et al.*[22] investigate the nature of contradictions occurring in natural language texts. Their research concerned different aspects and types of contradictions, such as: (a) those arising from antonymy, negation, and date/numeric discrepancies, and (b) those resulting from the use of factive or modal verbs, or from structural or lexical contrasts, as well as world knowledge. In this thesis, in order to identify oxymorons, we rely only on the first two types of contradictions: antonymy and negation. As Marneffe notes, the types of contradictions in the second category are much harder to detect because they require precise models of sentence meaning. Even though we do not operate on the level of higher syntactic units, like phrases or clauses, but only on words, we will have to take into consideration the broad influences arising from their properties. As we point out in Chapter 5, knowledge about the world is also necessary for the correct detection of some types of oxymorons, as not all of them result only from the contradictory meaning of the constituting words.

2.4 Text Visualization

Text visualization has been studies for some time now and many different approaches exist. Here, we focus on one specific method.

Wattenberg et al.

Wattenberg *et al.*[66] investigate editing activity on Wikipedia, the well-known online encyclopedia. The *chromogram* technique introduced in order to analyze the huge editing histories of the Wikipedia websites has particular relevance to us. First, the interesting idea of representing tokens by colours helps to avoid lengthy labels and therefore efficiently makes use of space. The authors address the subtle problem of mapping tokens to colours in the most informative way. The hue, saturation, and brightness components are determined by the first three letters of a string. The visualization of the position of rhetorical figures in text does not require such a precise scheme; however, we borrow some of the proposed ideas. By assigning different colours to individual figures and modifying the saturation we try to capture possible rhetorical patterns occurring in textual documents, more specifically in political speeches. Also, the graphical representation of rhetorical figures within a text in a linear form seems to be the most natural way to perceive text. Section 4.4 presents the details of our approach, and Section 5.2 describes the rhetorical analysis using our visual exploratory method.

2.5 Citation Classification

Radoulov

A citation annotation facility is a component of our Web-based document authoring tool, developed within our IN^3SCAPE project at the University of Waterloo. The scheme for annotating the rhetorical purpose of scholarly citations presented by Radoulov[56] was incorporated into the tool.

Citations in scholarly articles play an important role in creating relationships among mutually relevant articles within a research field by expressing semantic links between the documents. These inter-article semantic relationships represent aspect of the argumentation structure intrinsic to all scientific writing. Therefore, determining the nature of the exact relationship between a citing and cited paper requires an understanding of the rhetorical relations within the argumentation context in which a citation is placed. To determine these relations automatically in scientific articles, Radoulov proposed that associated pragmatic features within the context of a citation may be automatically determined by computational linguistic analysis. In his project, the goal was to automatically annotate the purpose of a citation, on the basis of these pragmatic features, using a combination of discourse analysis and machine learning techniques. The separate modules for automated annotation of the rhetorical purpose of scholarly citations, rhetorical figures, and possibly other pragmatic evidence might be usefully integrated into a system for comprehensive rhetorical analysis of different genres of text.

Chapter 3

Detection of Figures

3.1 Detection Framework

In this section we will briefly describe our general approach towards the detection of various kinds of figures. First we will give an overview of the syntactic units of a text that will be taken into consideration. Then we will describe the reasons for our choice of parser followed by a summary of its small imperfections and how they affect the detection of figures. Then we will move on to an overview of rhetorical figures discovery algorithms.

3.1.1 Syntactic Units

The initial step in our detection procedure is finding sentence boundaries. We will not be looking at the lines of text or larger syntactic units as, for example, paragraphs, although we are aware that for comprehensiveness these should also be taken into account in future. The sentence boundary detection (SBD) problem is broad itself and there has been substantial research already done. However, as it is not the main focus of this work, we believe the tools provided by Java API[2], and more specifically the *BreakIterator* class, should suffice. As we will see later in Section 3.2.2 and Chapter 5 it behaves correctly in most of our cases. The class is intended for use with natural languages only and the sentence boundary analysis provided allows selection with correct interpretation of periods within numbers and abbreviations, and trailing punctuation marks such as quotation marks and parentheses.

Figures of speech occur however among all possible syntactic units of sentences, starting at single-letter level, through words, to clauses and phrases. In order to detect all these our approach to parsing has to provide information about the boundaries of these units. Moreover, not only syntactic structure but also lexical dependency between units plays a significant role in the detection of rhetorical figures. With this in mind, we decided to use the lexicalized probabilistic context-free grammar (PCFG) parser created by the Stanford Natural Language Processing Group[3, 37, 38]. Klein and Manning[37] present a novel generative model for natural language tree structures. They provide two separate models for independent representation of lexical dependency and syntactic structures.

The detection of the figures which occur on the lowest syntactic level—letters—is not taken into consideration in this work. Possible extensions of the current system, which would include the annotation of rhetorical figures operating on letters and syllables, are suggested in Chapter 6.3.

3.1.2 Why a lexicalized PCFG parser?

Jurafsky and Martin[36] summarize the problems of plain PCFG parsers in modelling structural and lexical dependencies. First, PCFG parses rely on the independence assumption, meaning that the expansion of any non-terminal node is independent of the expansion of any other non-terminal. However, the desired approach that takes into account the statistics of English grammar is that the rule which is used for node expansion depends on its position in the parse tree. Another problem of plain PCFGs is that they do not take into consideration lexical information. According to many researchers ([27, 67, 33]), lexical dependencies play an important role in selecting the correct parse. Thus a model that keeps separate lexical dependency statistics for different parts of speech is more appropriate in most cases.

Stanford's parser is implemented as a product model of a plain PCFG parser and a lexicalized dependency parser. The separate PCFG phrase structure and lexical dependency experts' preferences are combined by efficient exact inference, using an A* algorithm[37]. Lexical dependencies are significant from the inter-word detection point-of-view. Correct phrase structure parses on the other hand are crucial for the accurate detection of figures of repetition. Below we review some of the imperfections of the parser followed by our methods for attempting to address these issues.

3.1.3 Parsing problems

Although performing very well in most of the cases, sometimes the parser does not choose the optimal correct parse. Let us consider the following example:

Example 3.1:
We shall fight in the fields and in the streets, we shall fight in the hills[1].

The parse-tree structure of the above sentence is presented in Figure 3.1. Marked in red are the nodes which have been incorrectly expanded from the main clause node. The correct parse should put these nodes under the S-clause located on the left side in the figure so that *We shall fight in the fields...* and *...and in the streets...* constitute one clause.

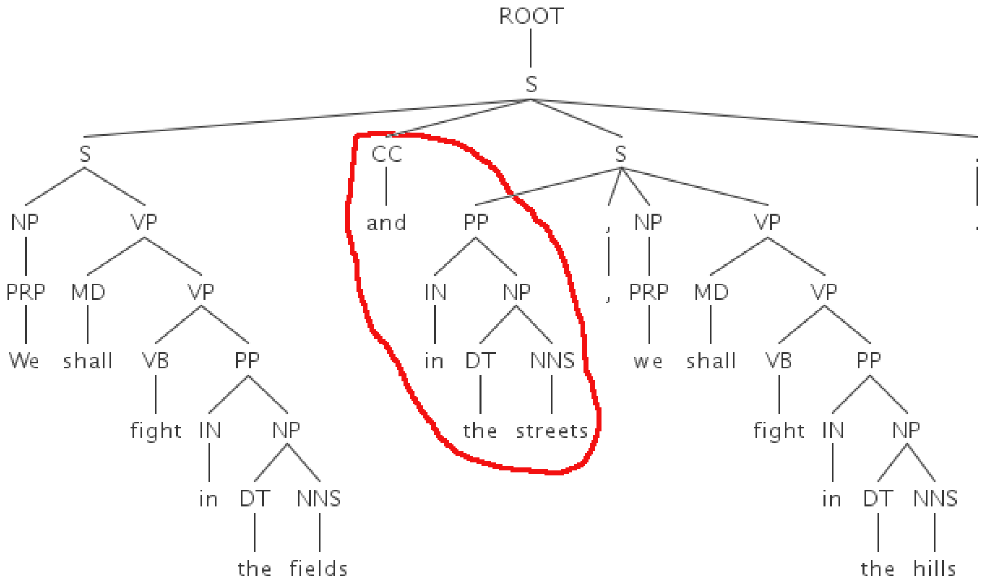


Figure 3.1: Parse trees of two example phrases

Such behaviour might cause problems for rhetorical figure detection, especially for those that pertain to the occurrences of repetitions of words at the certain location of phrases, clauses or that concern a number of successive phrases or clauses.

One way to deal with the above problem is to ignore the phrase structure parse and concentrate on the text itself. More precisely, we have developed a very simple

[1] Winston Churchill, excerpt from *Speech to the House of Commons* June 18, 1940

heuristic which creates phrases (additional to those coming from the parse tree) based on the punctuation markers. If there is more than one word between commas, colons, semicolons, etc., in a sentence, those words constitute a phrase. Of course, it is very hard to determine the nature of this phrase (verb, noun, adjective phrase), as we consider them separately and therefore lose some context information coming from the other, surrounding phrases. Nonetheless, as far as the recognition of figures of repetition is considered, it is not an influential factor. As the tool enables the manual annotation or correction of automatically detected pragmatic evidences, we assumed that it is better if the automated detection generates more false positives than false negatives. Therefore, relaxing the strict definition of the boundary of a phrase is not a very significant issue. However, it is definitely desirable to address this more formally in future work (see Chapter 6.3).

The rest of this chapter is organized as follows: Section 3.2 concentrates on the figures of repetition. The detection of figures of parallelism is described in Section 3.3. Lastly, our preliminary work on the discovery of *tropes* is described in Section 3.4.

3.2 Schemes—Figures of Repetition

3.2.1 Detection of Repeating Words

The major purpose of repetition is to produce emphasis, clarity, amplification, or emotional effect[1]. Repetitions are a good exemple of figures that occur on various levels of a text:

1. Letters, syllables, sounds – alliteration, assonance, paroemion;
2. Words – anadiplosis;
3. Clauses and phrases – isocolon;
4. Ideas – commoratio, disjunctio.

In this section we will concentrate on the second and third categories. We will be considering the repetitions of words, clauses, and phrases.

In order to find repeating words or phrases we have to first establish the range of text in which we will be looking for them. It is true that if two (or more) words repeat in three consecutive paragraphs, they might be considered as a repetition from a text-processing viewpoint. However, in most cases, such a repetition does not carry any deliberate rhetorical value and definitely does not change the audience's perception of the text. The only exception might be the case where a certain phrase starts or ends successive paragraphs. Therefore, we decided to look at a sliding window having the length of three sentences. We assumed that this is a range large enough to capture a deliberate repetition, but at the same time would omit others that should be ignored from rhetorical considerations. The value determining the size of the window is modifiable though.

Note:

> The length of the sentence sliding window may vary according to the figure under consideration. We will draw the reader's attention when describing individual figures.

It is important to note that the window moves forward one sentence at a time, which means that the repeated sequences might occur in three successive window frames. Let us consider the following example (numbers in the square brackets indicate the sentence number).

Example 3.2:
[1]*I have a dream that one day (...) are created equal.* [2]*I have a dream that one day (...) table of brotherhood.* [3]*I have a dream that one day (...) an oasis of freedom and justice.* [4]*I have a dream that (...) but by the content of their character*[2]*.*

For a sliding windows of length three, one of the repetitions found for the first three sentences is *I have a dream that one day.* For sentences 2, 3, and 4 another repetition is *I have a dream that.* The following question arises: should these two repetitions be merged or should they be presented as two separate entities? The issue of merging overlapping repetitions is addressed at the end of this section (Section 3.2.4 on page 30).

[2] Martin Luther King Jr., excerpt from *I Have a Dream* speech August 28, 1963

Finding repetitions

Let us consider the following example:

Example 3.3:
Believe not all you can hear, tell not all you believe[3].

Our algorithm looks for repeating expressions of all possible lengths. The result for the above example sentence is presented in Table 3.1.

Table 3.1: Repeated phrases

Phrase length	**Phrases**
1	not, all, believe, you
2	not all, all you
3	not all you

After all the repetitions in the text have been found we can move on to the detection of individual figures. First though, let us introduce the necessary formalism, introduced by Harris and DiMarco[32], that we will be using throughout this section. Table 3.2 presents the descriptive elements used the formalization of schemes.

Table 3.2: Formalism for figures of repetition

Symbol	**Meaning**
W	word
S	stem
M	morpheme
$\ldots$	arbitrary intervening material, possibly null, with some (as yet unspecified) upper limit (often shorthanded below as *proximal*)
$\{\ldots\}$	morpheme boundaries
$[\ldots]$	word boundaries
$< \cdots >$	phrase or clause boundaries (assuming, more or less, that clauses are just special types of phrases, aggregating other phrases)

[3] Native American proverb

Let us now move on to the description of the detection of individual figures. We start with those that we believe will be the easiest to find. First, we will look at the ones pertaining to the repetition of words regardless of their location in a sentence, clause, or phrase (epizeuxis, ploche, polysyndeton). Then we move on to the discovery of repeated phrases, or groups of words, which are distinguished by their position in a sentence or clause (anaphora, epistrophe, epanalepsis, antimetabole, anadiplosis). Lastly, we give an overview of the detection of a figure which concerns the repetition of a word but in different forms (polyptoton). The following algorithm 3.1 outlines the detection of figures of repetition. The condition in line 4 differs between figures but the main idea remains unchanged (except for polyptoton, which is described at the end of this section).

Algorithm 3.1 Outline of the detection of figures of repetition

```
1: create empty set for rhetorical figures $S_{rh}$
2: for all sliding windows $W_i$ do
3:    for all repetitions $R_j$ of the a sliding window $W_i$ do
4:       if conditions of the definition of a figure $F_k$ are met then
5:          add figure $F_k$ to $S_{rh}$
6:       end if
7:    end for
8: end for
9: return $S_{rh}$
```

3.2.2 Finding Rhetorical Figures

Epizeuxis

Definition 3.1. ***Epizeuxis***: *Repetition of words with no others between, for vehemence or emphasis[1].*

Formally:

$$[W]a[W]a$$

Example 3.4:
O horror, horror, horror.[4]

[4] William Shakespeare, *Macbeth*

Epizeuxis is very easy to detect. We need only look for two or more occurrences of the same word occurring next to each other. The only step of the procedure that should be stressed here is that some of the special characters between words are omitted. Our detection system operates on leaves of phrases coming from the parse trees. All the characters in Table C.1 starting from the pound sign to the right-closing double quote mark would also be leaves in a parse tree. The procedure would mistakenly not treat *horror, horror* as *epizeuxis* because a comma character and word *horror* are definitely not equal. Therefore, all of the above characters should not be taken into consideration. On the other hand, we cannot entirely rely on the word boundaries because characters that are not part of a word, such as symbols or punctuation marks, have word-breaks on both sides. As a result a phrase *can't can't* would not be considered an *epizeuxis* as the character *t* is not equal to the word *can.* Nor can we rely on the output of the parser in this case because the word *can't* is parsed as [\MD ca] and [\RB n't]. The neighbours *n't* and *ca* (from the second word) are not equal and the epizeuxis is missed. We handle this problem by removing all in-word symbols, concatenating the remaining parts and comparing those "compound" words.

Ploche

Definition 3.2. ***Ploche:*** *The repetition of a single word for rhetorical emphasis[1].*

Formally:

$$[W]_a ... [W]_a$$

Example 3.5:
We *must <u>all</u> hang together or assuredly* ***we*** *shall <u>all</u> hang separately.*[5]

Because a *ploche* is the repetition of a single word, *we*, *all*, and *hang* in the above example are three separate ploches. Another implication is that the occurrence of any of the other figures of repetition that require having the same word automatically causes the occurrence of a ploche.

The definition of a ploche does not mention specifically how far apart the repeated words should be. We decided that a sliding window consisting of two sentences is the maximum reasonable length. The closer the words are to each other,

[5] Benjamin Franklin, at the signing of Declaration of Independence

the stronger the emphasis of a ploche. We do not consider a double occurrence of a word a ploche if it happens to be within the scope of three successive sentences. In other words, if a word occurs in sentence 1 and then in sentence 3, it is not considered a ploche. On the other hand, if a word is repeated three or more times, or if it occurs twice in the same sentence, or in two neighbouring sentences, we consider it a ploche. Additionally, we do not take into consideration English stopwords.

Polysyndeton

Definition 3.3. ***Polysyndeton:*** *Employing many conjunctions between clauses, often slowing the tempo or rhythm[1].*

There are at least two reasons why *polysyndeton* is used:

1. To slow down the rhythm.

   Example 3.6:
   And *God made the beasts of the earth according to their kinds* ***and*** *the cattle according to their kinds* ***and*** *everything that creeps upon the ground according to its kinds.* ***And*** *God saw that it was good.*[6]

2. To produce special emphasis.

   Example 3.7:
   I have a midterm ***and*** *two projects* ***and*** *three assignments* ***and*** *two reports to write* ***and*** *I don't know what to begin with.*

The length of our sliding window for polysyndeton is reduced to two, for the reasons explained below. There are two cases which we take into consideration:

1. Polysyndeton occurs only in one sentence, or
2. The same conjunctions begin two consecutive sentences.

[6] *Genesis*, I:24-25

In order to capture the second case we have to look at two neighbouring sentences at a time. However, if several conjunctions occur in the middle of two (or more) sentences, they will be treated as separate polysyndetons.

Anaphora

Definition 3.4. ***Anaphora:*** *Repetition of the same word or group of words at the beginning of successive clauses, sentences, or lines[1].*

Formally:

$$< [W]_a ... >< [W]_a ... >$$
$$< [W]_a [W]_b ... >< [W]_a [W]_b ... >$$

Anaphora has a profound effect on the audience and we can be sure that if it occurs the author has used it deliberately. The repetition of words is very often used in order to establish a certain rhythm in the sequence of clauses, and therefore causes a strong emotional effect[16]. A famous example of anaphora is presented below:

Example 3.8:
We shall *not flag or fail.* ***We shall*** *go on to the end.* ***We shall*** *fight in France,* ***we shall*** *fight on the seas and oceans,* ***we shall*** *fight with growing confidence and growing strength in the air,* ***we shall*** *defend our island, whatever the cost may be,* ***we shall*** *fight on the beaches,* ***we shall*** *fight on the landing grounds,* ***we shall*** *fight in the fields and in the streets,* ***we shall*** *fight in the hills.* ***We shall*** *never surrender.*[7]

Anaphora is the first of the figures in our set that concerns not only the repetition of a group of words, but also specifies a certain position in a phrase or sentence in which it must be placed. We have relaxed the definition of anaphora a little bit and consider not only successive syntactic units but also phrases and clauses in close vicinity. At the beginning of this section we mentioned that before the detection of figures per se is performed, we find all the possible repetitions within the sliding window. Now, we only have to check whether some of them are located at the beginning of two or more clauses or phrases. There are two significant assumptions

[7] Winston Churchill, *Speech to the House of Commons* June 18, 1940

we make at this point. First, the length of a syntactic unit (phrase, clause, or sentence) in which we look for a anaphora, counted in the number of words, has to be equal or greater than some specified number *min_length*. Secondly, we do not take into consideration determiners, conjunctions, and prepositions that start a phrase or clause. The precision and recall results of anaphora detection according to the value of *min_length* are presented in Chapter 5.

Epistrophe

Definition 3.5. ***Epistrophe:*** *Ending a series of lines, phrases, clauses, or sentences with the same word or words[1].*

Formally:

$$< ...[W]_a >< ...[W]_a >$$
$$< ...[W]_a[W]_b >< ...[W]_a[W]_b >$$

Epistrophe creates emphasis not only by repeating a word or words but also by positioning them at the end of a clause or sentence, thus setting up a pronounced rhythm[16].

Example 3.9:
There is no Negro ***problem****. There is no Southern* ***problem****. There is no Northern* ***problem****. There is only an American* ***problem****.*[8]

The only difference in our detection algorithm between anaphora and epistrophe is that for the latter we look at the end of a phrase or clause instead of the beginning. In other words, only the condition in line 4 of algorithm 3.1 changes. The rest of the steps remains unmodified.

Epanalepsis

Definition 3.6. ***Epanalepsis:*** *Repetition at the end of a line, phrase, or clause of the word or words that occurred at the beginning of the same line, phrase, or clause[1].*

[8] Lyndon B. Johnson, *We Shall Overcome*

Formally:

$$< [W]_a ... [W]_a > < [W]_a [W]_b ... [W]_a [W]_b >$$

Example 3.10:
*In **times like these**, it is helpful to remember that there have always been **times like these**.*[9]

We know that anaphora and epistrophe are both figures located in positions of strong emphasis in a sentence, the beginning and the end, so, by having the same phrase in both places, the speaker calls special attention to it. According to Corbett [17], "epanalepsis is rare in prose, probably because when the emotional situation arises that can make such a scheme appropriate, poetry seems to be the only form that can adequately express the emotion."

The detection of epanalepsis is in a way a combined procedure of the algorithms for the two aforementioned figures. Once we have found the occurrence of a word or group of repeated words at the beginning of a clause or phrase we check whether the same group is also present at the end of the same syntactic unit.

Anadiplosis

Definition 3.7. ***Anadiplosis:*** *The repetition of the last word (or phrase) from the previous line, clause, or sentence at the beginning of the next[1].*

Formally:

$$< ...[W]_a >< [W]_a ... >$$
$$< ... < ... >_a >< < ... >_a ... >$$

Example 3.11:
*Information is not **knowledge**, **knowledge** is not **wisdom**, **wisdom** is not **truth**, **truth** is not **beauty**, **beauty** is not **love**, **love** is not **music** and **music** is the best.*[10]

In some sense epanalepsis is the reflection of anadiplosis. The difference in the detection between these two lies in the place that the algorithm commences. In

[9] Paul Harvey

[10] Frank Zappa

the latter case, we start by checking whether a group of words ends one syntactic unit and if the same group begins the next unit. Important to note is the distance between the end of one phrase and the beginning of the next one. We decided to omit determiners as well as conjunctions at the beginning of the second clause. Example 3.12 presents the case where the conjunction *and* is omitted and the word *hope* constitutes an anadiplosis.

Example 3.12:
We also rejoice in our sufferings, because we know that suffering produces ***perseverance; perseverance****, character; and character,* ***hope****. And* ***hope*** *does not disappoint us.*[11]

Antimetabole

Definition 3.8. ***Antimetabole:*** *Repetition of words, in successive clauses, in reverse grammatical order[1].*

Formally:

$$[W]_a ... [W]_b ... [W]_b ... [W]_a$$

Example 3.13:
Ask not what your ***country*** *can do for* ***you****; ask what* ***you*** *can do for your* ***country****.*[12]

Very often when *antimetabole* is used, the direct object in the first clause becomes the subject in the second. As in the above example, by applying this rhetorical figure, John Kennedy explicitly stressed not what a person will get but rather what they can give. The main purpose of such a mechanism was to emphasize the contribution Americans might make to the nation they live in. In general, antimetabole occurs very often in discourse, but can also introduce humour[71], as in the quote attributed to Samuel Johnson in *Boswell's Life of Johnson*[12] "This man I thought had been a Lord among wits, but, I find, he is only a wit among Lords."

The detection procedure in the case of anadiplosis is a little bit different. Once the repeating groups of words have been detected, we examine if they are "word palindromes'. What we mean by "word palindrome" is a sequence of words that

[11] Romans 5:3-5

[12] John F. Kennedy, *Inaugural Address* January 20, 1961

if read from either the beginning or the end is the same. Again, we relaxed the condition contained in the definition of the figure mentioning successive clauses. We look at word palindromes that occur in phrases situated in close vicinity but not necessarily neighbouring. The details are presented in algorithm 3.2.

Algorithm 3.2 Find inclusions of word pairs—pairs of words occurring in reverse grammatical order

```
 1: Vector $V$ contains all the repetitions found within a specified range
 2: initialize vector $S$ which contains the inclusions of repeated word pairs
 3: for $i = 0$ to $|V| - 1$ do
 4:     $j \leftarrow$ first next occurrence of word $W_i$ in $V$
 5:     $W_iNext \leftarrow$ word in $V$ at index $j$
 6:     $V_{sub} \leftarrow$ subvector $V(i+1, j)$ of words between $W_i$ and $W_iNext$
 7:     add $W_i$ and $W_iNext$ to $S$
 8:     repeat recursively lines 1 to 7 for $V_{sub}$
 9: end for
10: return $S$ — at the end of each step of recursion $S$ contains word pairs inclusions
```

While examining the sample antimetaboles we noticed that noun phrases containing a noun and a preceding determiner or pronoun are very often treated as one word with respect to reverse ordering. For example, in Example 3.13 a 2-word phrase *your country* is part of the antimetabole, and the swapping between *your* and *country* is not required, although it should be, if we want to follow exactly the definition of the figure.

3.2.3 Repetition of Derivationally Related Words

Polyptoton

Definition 3.9. ***Polyptoton:*** *Repeating a word, but in a different form. Using a cognate of a given word in close proximity[1].*

Formalism and examples:

Table 3.3: Polyptoton examples

Notation	Example
$[S_a\{M_a\}]...[S_a\{M_b\}]$	*Lovely lovers.*
$[\{M_a\}S_a]...[S_a\{M_b\}]$	*Unknown knowers.*
$[S_a\{M_a\}]...[\{M_b\}S_a]$	*Knowing unknowns.*
$[\{M\}S_a\{M\}]...[S_a]$	*Unfriendly friend.*
$[S_a\{M\}]...[S_a]$	*Friendly* to his *friend.*

WordNet

Before we move on to the polyptoton detection procedure itself we need to first introduce WordNet, as we refer to it several times in this chapter.

WordNet[14, 48] is one of the most important and widely used lexical databases for natural language processing tasks. English main parts-of-speech—nouns, verbs, adjectives and adverbs—are organized into so-called *synsets*, sets of near-synonyms. Unquestionably, the most important feature of WordNet for various NLP problems, including word sense disambiguation (WSD), is that the lexical information is arranged in terms of word meanings, rather than word forms[48]. Additionally, it records the various semantic relations between these synonym sets. Below we describe how we apply WordNet in polyptoton detection.

Finding Derivationally Related Forms of a Word

We have placed polyptoton as the last figure in this category because, although it concerns repetitions, not a word or group of words is repeated literally, but they occur in a different case, inflection, or voice, or are used in different parts-of-speech[23]. The term “polyptoton” derives from the Greek *poly*, many, and *ptosis*, (grammatical) case. In order to detect polyptotons we have implemented a method for finding the different forms of the same word. Algorithm 3.3 presents our procedure. We will be looking for the forms of an arbitrary word W. Each step of the algorithm is described in more detail below.

Algorithm 3.3 Find derivationally related forms of a word

```
1: create empty set of word candidate forms $S_{CF}$
2: add $W$ to $S_{CF}$
3: find stem $S_w$ (using Porter algorithm) of a word $W$
4: for each prefix $p$ from list of prefixes $L_p$ do
5:   if word $W$ starts with $p$ then
6:     word without prefix $W_{np} \leftarrow$ delete prefix $p$ from the beginning of $W$
7:     if $W_{np}$ exists in English (check against WordNet) then
8:       add $W_{np}$ to $S_{CF}$
9:     end if
10:  else
11:    word with prefix $W_{wp} \leftarrow$ add prefix $p$ in front of $W$
12:    if $W_{wp}$ exists in English (check against WordNet) then
13:      add $W_{wp}$ to $S_{CF}$
14:    end if
15:  end if
16: end for
17: repeat lines 4 to 16 for the stem $S_w$
18: for all candidate forms $C_i$ from $S_{CF}$ do
19:   find a derived form $D_C \leftarrow D_f(C_i)$ of $C_i$, where $D_f$ is the "Derived Forms"
      option in WordNet
20:   add $D_C$ to $S_{CF}$
21: end for
22: return $L_{CF}$
```

We use the traditional Porter algorithm[54] implementation[13] for finding the stem of a word (line 3). The stemmer performs quite well on suffixes, but its ability to handle prefixes is very limited. Our approach checks (line 5) if the word starts with a given prefix. The set of the 24 most popular English prefixes is used. If it does start, a word without the prefix is added to the candidate forms set (line 8). If not, we add the prefix to the word (line 11) and then add the new word to the candidate set (line 13). Before we add the new words to the set, we also make sure that they actually exist in the language by checking if there exists a corresponding entry in WordNet (lines 7 and 12). The procedure presented in algorithm 3.3 is also repeated for the predefined list of the most popular suffixes. The only difference is to add suffixes to the end of the word and check whether a new one exists in the

[13] Fotis Lazarinis implementation of Porter stemmer,
http://ftp.dcs.glasgow.ac.uk/idom/ir_resources/linguistic_utils/porter.java

lexical database. Although naive, the latter step produces some additional word forms. The final step is to add to the set of candidates the forms that WordNet itself might provide (line 19). One of the features of WordNet is the possibility to display derivational morphology links between noun and verb forms. Let us have a look at the forms inferred by the algorithm for the example word: *sensitive*:

Example 3.14:
Without WordNet derivative option: *[sensitize, oversensitive, sensitiveer, sensitise, sensitiveness, insensitive, sensitive]*

With WordNet derivative option: *[sensitize, insensitiveness, sensitivity, sensation, sensitiser, oversensitive, sense, sensitizer, sensitiveer, sensitisation, sensibleness, insensitive, sensing, sensor, sensible, insensitivity, oversensitiveness, sensibility, sensitization, sensitise, sensitiveness, sensitive]*

As we can see from the example, WordNet can provide a large extension of the derived forms of a word. The only form we are missing is *insensitively.* How does the above procedure help us with detecting polyptotons? The idea is simple. Imagine we have two words: *sensitive* and *insensitively.* First, we discover all the possible forms of those two words, and then, we check whether there is any overlap between the emerged sets. If so, we consider them as originating from the same stem and therefore constituting a polyptoton. Although the algorithm did not produce *insensitively* as a form of the word *sensitive*, running it on the second word results in the collection: *[insensitivity, insensitiveness, insensitively, sensitively, insensitive].* Four out of five words from the second set are also present in the first, which is a strong indication of the same origin of both words.

Unsolved problems

In every stemming procedure there are common issues that have to be taken into consideration. Natural languages are not completely regular constructs, and therefore stemmers operating on natural words inevitably make mistakes[51]. These are *understemming* and *overstemming* errors. The first type of error refers to the situation when words that ought to be merged together, for example, “insensitive” and “sensitively”, may remain separate after stemming. Above, we have shown one of the possible ways to reduce this kind of error. However, there are many cases in natural language that for now we are unable to handle correctly, especially when the detection of polyptotons is considered. One such case occurs when ‘the same' word significantly changes its form in different parts of speech. For example, the noun "blood" and the verb "bleed" should be marked as polyptoton, but the algorithm mistakenly omits these two. We hope to tackle this problem in the future development of the annotation tool (see Chapter 6.3).

Overstemming on the other hand occurs when words which are really distinct are wrongly conflated. Algorithm 3.3 produces many distinct forms of a word, and sometimes words that originate from different stems are put into the same set. A good example is the word *sentence*. The root form coming out of the stemmer is *sent*. When we apply the algorithm, many words, such as *unsent, dissent, presentation* etc., are wrongly included. However, for our analysis, producing false positives is more important than understemming, as the tool gives a human annotator the possibility to apply the corrections. Ultimately, we would like to create a comprehensive database of word forms using JANTOR and human annotators (see Chapter 6.3).

3.2.4 Merging overlapping figures

As we mentioned before in Section 3.2.1, the algorithm for detecting repeating words operates on sliding window of specified length. The following question arises: how should we treat the situation when the repetitions vary a little bit between the successive frames of the window? In other words, how should we deal with the situation presented in example 3.2? There are couple of cases we have to take into consideration.

First, if a group of repeated words is spread across more sentences than the size of the window, and these words constitute a rhetorical figure, then they certainly have to be merged into one. For example, if a word *we* starts four or more successive sentences all of *we*'s should be marked as one instance of anaphora spreading across those sentences.

Secondly, a smaller group of words might be covered by larger ones. In order to explain this situation, let us use have a look at example 3.8 on page 22 again (below) and let us consider the detection of anaphoras. All the phrases that start with the word *we* also start with the phrase *we shall*. Therefore, instead of creating two separate figures, we ignore the shorter one.

Example 3.8:
We shall not flag or fail. We shall go on to the end. We shall fight in France, we shall fight on the seas and oceans, we shall fight with growing confidence and

growing strength in the air, we shall defend our island, whatever the cost may be, we shall fight on the beaches, we shall fight on the landing grounds, we shall fight in the fields and in the streets, we shall fight in the hills. We shall never surrender.[14]

The previous two cases were trivial and obvious. However, if we have a closer look at example 3.8 there are many more smaller anaphoras and we think they should be treated separately. Table 3.4 presents all of them

Table 3.4: Different anaphoras example

Anaphora	# of starting phrases
we shall	10
we shall fight	7
we shall fight in	3
we shall fight on the	3
we shall fight in the	2

We decided to keep all of the above figures separate. Another solution would be to join, for example, *we shall fight in* and *we shall fight in the* and introduce some intra-figure similarity indicator. This indicator would represent the distance between the individual parts of a figure. For the aforementioned phrases, the distance between two *we shall fight in the* phrases would be 0, and the distance between these and *we shall fight in* would be 1, as the latter misses one word. Algorithm 3.4 on page 34 could be used for calculating this similarity. Our current approach does not take into consideration this solution as we decided to remain strict about the exactness of the repetition of the words, therefore all of the above figures are displayed separately.

3.3 Schemes—Figures of Parallelism

Definition 3.10. ***Parallelism:*** *Similarity of structure in a pair or series of related words, phrases, or clauses [16].*

In grammar and rhetoric, *parallelism* is one of the basic principles concerning the construction of sentences. The main requirement of the principle is that equivalent syntactic units are organized in co-ordinate grammatical structures. Therefore, as Corbett suggests[16], nouns must be yoked with nouns, prepositional phrases

[14] Winston Churchill, *Speech to the House of Commons* June 18, 1940

with prepositional phrases, adverb clauses with adverb clauses, etc. Ignoring these requirements leads to the violation of the grammar of co-ordination as well as the rhetoric of coherence. What is meant by the violation of the grammar of co-ordination, is, for example, joining two elements of different grammatical kind by the conjunction. This very often results in weakened communication and reflects confused thinking. Thus, the detection of figures of parallelism might be very useful for the identification of erroneous sentence constructs. One of the parallel figures the system annotates is *isocolon*, described in detail in the next subsection.

Isocolon

Definition 3.11. ***Isocolon:*** *A series of similarly structured elements having the same length[1].*

Example 3.15:
I speak Spanish to God, Italian to women, French to men, and German to my horse.[15]

Corbett[16] gives a more precise definition of *isocolon* indicating that parallel elements should be similar not only in structure but in length (meaning the same number of words, and even the same number of syllables). In our approach we take into consideration the number of words, but we currently ignore the number of syllables.

In this section we describe the detection of similarly structured phrases. First we propose an approach for finding grammatically alike phrases basing on the similarity of their parse trees, followed by an algorithm for detecting phrases with maximum word tags overlap.

Similarity of phrases parse-trees

Before we describe the algorithm for finding the similarity between two phrases, we must first provide the necessary formalization.

Leaf-Label element: A class constructed for each word in the phrase consisting of the following fields:

- Label: indicates the part-of-speech tag assigned to a word by the parser;

[15] Charles V

- Depth $Depth(C_{node})$: calculated as: $Depth(R_{node}) - Dist(R_{node}, C_{node})$, where R_{node} is the root node of the phrase, C_{node} is the POS (part-of-speech)-node of the current word, $Dist(R_{node}, C_{node})$ is the distance between two nodes, and finally the depth of a node $Depth(node)$ is the length of the longest path from this node to the lowest located leaf node. In other words, distance is the level on which the node is located, calculated from the bottom of the parse tree.

Figure 3.2 illustrates the procedure of constructing Leaf-Label elements. Let us consider the word *saw* and the assigned tag VBD (verb in a past tense form), circled in red. The parent of VBD is VP and is circled in green. Although we do not make use of them directly, we envisage that the comparison of parent nodes would introduce an additional criterion for spotting similarly, from the linguistic point-of-view, structured phrases.

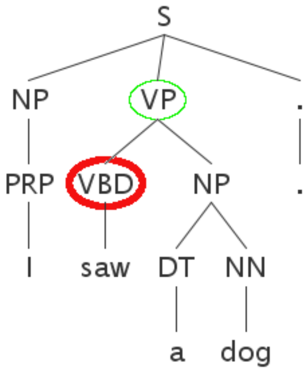


Figure 3.2: Example phrase parse tree

S is the root node of the phrase (the entire sentence in this case). The depth of the root node is equal to four (distance between S and leaf word *dog*). The distance between the root node and $C_{node} = VBD$ is two, therefore the depth of the node $C_{node} = VBD$ is equal to 2: $Depth(C_{node}) = Depth(R_{node}) - Dist(R_{node}, C_{node}) = 4 - 2 = 2$

The next part involves determining which tags can be considered as the same. For example, a verb in the future tense form and a verb in a past tense form are the same parts-of-speech and thus can be considered as the same with respect to the definition of isocolon. Table 3.5 on the next page presents the set of 'equivalent' tags and the corresponding part-of-speech. The detailed description of the Penn Treebank tag set can be found in [46] and in Table C.1.

Table 3.5: Tags equivalence classes

Tag	POS
JJ, JJR, JJS	adjective
NN, NNS, NNP, NNPS, NP-TMP	noun
RB, RBR, RBS, WRB	adverb
VB, VBD, VBG, VBN, VBP, VBZ	verb
WP, WP$, PRP, PRP$	pronoun

We also ignore the *n't* words coming from the parser. The parser treats this abbreviated part of modal-verb negations as an adverb, which we decided to omit. Algorithm 3.4 calculates the structural similarity between two phrases/clauses. The initial distance between the phrases is equal to the absolute value of the difference between the lengths of their word tag lists (line 2).

Algorithm 3.4 Find the difference between two phrases

```
 1: Construct list of Leaf-Label elements $L_{1,2}$ for first and second phrase respectively
 2: Initialize distance $D$ between phrases to $abs(|L_1| - |L_2|)$
 3: for $i = 0$ to $min(|L_1|, |L_2|)$ do
 4:    $E1_i \leftarrow ith$ element from $L_1$
 5:    $E2_i \leftarrow ith$ element from $L_2$
 6:    if $E1_i$ and $E2_i$ have the same labels and depth then
 7:       continue;
 8:    else
 9:       $D+ = min(|L_1|, |L_2|)-$ maximumWordTagOverlap($L_1$,$L_2$)
10:       return $D$
11:    end if
12: end for
13: return $D$
```

If all the elements in the lists are the same, meaning that their labels are equivalent and their depth is equal, then the lists are considered identical. If only one pair of the elements does not meet the identity criterion (line 6), we ignore the parse tree structure and calculate the maximum overlap of word tags between the lists (line 9). The distance is increased by the calculated value, and returned. Algorithm 3.5 finds the maximum overlap between the lists of Leaf-Label elements of two phrases/clauses.

Note:

> This approach should probably be extended by more advanced techniques for specifying distance between leaf-labelled trees. There are many solutions describing the similarity between labelled XML trees, which we may consider applying in the future[31, 5, 55, 28]. However, the current procedure gives satisfactory and promising results, which are presented in Chapter 5.

Finding maximum overlap between phrases' words tags

For simplicity, let us assume that first list (L_1) is shorter. In general, variable i in line 2 of algorithm 3.5 should vary from 0 to $min(|L_1|, |L_2|)$.

Let us have a look at the following excerpt from Winston Churchill's speech and observe how our algorithm works.

Example 3.16:
We shall fight on the beaches, we shall fight on the landing grounds, we shall fight in the fields and in the streets, we shall fight in the hills...[16]

Let us consider the following two phrases from the sentence. Their parse trees are presented in Figure 3.3 on page 37.

1. We shall fight on the beaches
2. we shall fight on the landing grounds

Their word tags lists are [PRP, MD, VB, IN, DT, NNS] and [PRP, MD, VB, IN, DT, NN, NNS] respectively. The result of algorithm 3.5 is in this case 6, the longest overlap list's size ([PRP, MD, VB, IN, DT, NNS]). Now, by varying the parameters, we determine how strict with the similarity of phrases the algorithm should be. Chapter 5 discusses precision and recall measures for the detection of isocolon (among other figures) according to different parameters.

[16] Winston Churchill, *Speech to the House of Commons* June 18, 1940

Algorithm 3.5 Find maximum overlap between word tags

```
create empty map $M(I, \Lambda(L(label)))$ mapping number of labels $I$ to the list of lists of labels having the length equal $I$
for $i = 0$ to $|L_1|$ do
  create empty vector of indices $V(j)$
  $E1_i \leftarrow ith$ element from $L_1$
  for $j = 0$ to $|L_2|$ do
    if $E1_i.label \equiv E1_j.label$ then
      add $j$ to $V(j)$
    end if
  end for
  if $M.isEmpty()$ then
    create an empty list $\Lambda$
    for all indices $j$ in $V$ do
      create a one-element list $L(j)$
      add $L(j)$ to $\Lambda$
    end for
    put $\Lambda$ to $M(1, \Lambda(L(j))$
  else
    for all indices $i$ from $V$ do
      for all number of elements $I$ from $\Lambda.keys()$ do
        for all lists $L_1$ where $L_1.length = I$ do
          $k \leftarrow$ last index stored in $L_1$
          if $k < i$ then
            $L_2 \leftarrow$ append $i$ to $L_1$
            add $L_2$ to $M's$ lists with length $I + 1$
          end if
        end for
      end for
    end for
  end if
end for
return the size of the longest list from $M$
```

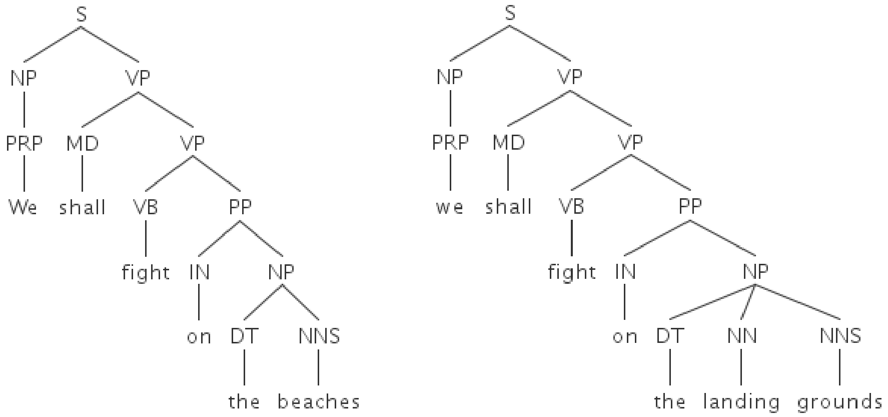


Figure 3.3: Parse trees of two phrases

3.4 Tropes

Trope is an artful deviation from the ordinary or principal signification of a word. In Greek it literally means "turn", and thus signifies when one turns a word or phrase from its conventional use to a novel one for rhetorical effect[1].

In this section we will describe a combined approach using WordNet and typed dependencies of Stanford's parser for the detection of one of the rhetorical figures from trope category, *oxymoron*.

Oxymoron

Definition 3.12. ***Oxymoron:*** *The yoking of two terms that are ordinarily contradictory[16].*

Examples of oxymoron expressions are presented below in Table 3.6.

As Corbett[16] notes, with this figure, as in most metaphorical language, there is a hidden ability to see similarities. By combining contradictions, speakers and writers might create a startling effect, gaining a reputation of good "word players". Interestingly, the word *oxymoron* is itself an oxymoron, as in Greek *oxy* means "sharp" or "pointed" and *moros* "dull'.

Table 3.6: Example oxymoron expressions

Oxymoron
open secret
clearly confused
act naturally
alone together
found missing
deafening silence

Typed Dependencies

The first step in our approach for the detection of oxymorons is determining the syntactic relationships between individual words. Marneffe *et al.*[21] propose a system for automated extraction of typed dependency parses of English sentences from phrase structure parses. There is a significant difference between these two types of parses. Phrase structure parses focus on capturing a nesting between multi-word constituents like clauses and phrases, whereas a dependency parse represents dependencies between individual words. Moreover, it assigns a grammatical relation to a dependency, such as subject, indirect object, adjectival modifier, etc.

Oxymoron is a type of rhetorical figure that concerns words located next to each other. In a text document consisting of N words there are $N - 1$ pairs of neighbouring words. If N is large, detection of all the oxymorons in a document might be computationally infeasible. Hence, we first have to determine grammatical relations of dependencies in which an oxymoron might appear. We examined 49 expressions containing oxymorons. Table 3.7 on the next page presents some examples with corresponding grammatical relations and their descriptions. The last row in the table presents the oxymoron that cannot be captured by only one grammatical relation. We will explain below how we address this problem. The next step of the detection is the semantic analysis of the chosen word pairs. More specifically, we will be looking at whether the two words meet the requirements imposed by the definition of the oxymoron. In other words, we want to find out whether they are contradictory. The application of WordNet, a lexical database, in finding contradictions is described in the next subsection.

Combining relations

Let us consider the expression *feather of lead*. There are two grammatical relations involved: a prepositional modifier between *feather* and *of* and a prepositional

Table 3.7: Possible grammatical relations between word pairs used for oxymoron detection

Example Sentence/Expression	Symbol	Relation description	Oxymoron
It is an *original copy.*	amod(copy, original)	adjectival modifier	original copy
almost exactly	advmod	adverbial modifier	almost exactly
icy hot	acomp	adjectival complement	icy hot
deafening silence	dobj	direct object	deafening silence
Plastic glasses	nsubj	nominal subject	plastic glasses

Feather of lead	prep(feather, of)
	pobj(of, lead)
		prepositional modifier
		prepositional object
			feather of lead

object between *of* and *lead*. In order to select the words *feather* and *lead* for further examination we have to somehow join those two relations. Here we present a naive approach:

1. For a dependent word W_1 in a grammatical relation $rel_1(W_1, W_2)$ we create a list of governors it occurred with (one governor for each relation);
2. We repeat step 1 for the governors, meaning for a governor word W_2 in $rel_1(W_1, W_2)$ we create a list of dependents;
3. If a new word W_3 of a new relation $rel_2(W_3, W_4)$ is the same word as W_1, we create a new pair (W_3, W_2) for examination, where W_2 is the second word in relation rel_1. Of course the number of new pairs might be greater than 1, provided that W_1 was part of more than one relation before rel_2.

Applying WordNet

Here we will concentrate only on those relations that are used in the detection procedure.

Synonymy
Most often two expressions are synonymous if the substitution of one for the other never changes the truth value of a sentence in which the substitution is made[48].

Antonymy
It is problematic to formally define the *antonymy* relation. As Miller *et al.*[48] note, the antonym of a word x is sometimes *not-x*, but not always. Generally, two words can be treated as antonyms if they are inherently incompatible[68].

Derived forms
The *derived forms* option of WordNet provides all the derivationally related forms of a given word according to WordNet.

Now let us move to the detection procedure itself.

Step 1:
Once the candidate word pairs have been identified using typed dependencies and grammatical relations, we find all the derived forms of both words the relation contains. The procedure for finding all the derived forms of a word has been previously described in Section 3.2.3.

Step 2:
We begin with the first word from the candidate pair and look for either its antonyms, synonyms, or derived forms. At this point it is important to describe exactly how we are applying the knowledge contained in WordNet. Let us have a look at Table 3.8.

Table 3.8: Word to word relation paths

Relation path			
Relation	Relation	Relation	Relation
Antonym	—	—	—
Antonym	Synonym	—	—
Synonym	Antonym	—	—
Antonym	Derivation	—	—
Derivation	Antonym	—	—
Antonym	Synonym	Derivation	—
Antonym	Derivation	Synonym	—
Synonym	Derivation	Antonym	—
Synonym	Antonym	Derivation	—
Derivation	Antonym	Synonym	—
Derivation	Synonym	Antonym	—
Synonym	Antonym	Synonym	—
Synonym	Antonym	Synonym	Synonym

We examined a set of the most popular oxymorons and came up with the WordNet-based detection approach. (W_1, W_2) is our candidate oxymoron. Given the first word W_1 in its original form we create a set $S(R_1, W_1)$ of all the words related to W_1 by relation R_1. We also keep the set $Der(W_2)$ of all the derived forms of the second word W_2 from the pair. R_1 is one of the possible values from the first column of Table 3.8. If the relations path already contained *antonymy* we check whether there is any overlap between words in $S(R_1, W_1)$ and $Der(W_2)$. If so, we treat (W_1, W_2) as an oxymoron and stop the algorithm. Of course, the more words overlap, the stronger the indication of a possible oxymoron. If not, we look for all the words $S(R_2(R_1, W_1))$ related to any word in $S(R_1, W_1)$ by relation R_2, where R_2 is a distinct value from the second column of Table 3.8. We check $S(R_2(R_1, W_1))$ against $Der(W_2)$, etc. If the algorithm fails to find the oxymoron after checking all the possible paths, it repeats the procedure starting with the second word W_2 and $Der(W_1)$.

The procedure of finding the derived forms of a word for oxymoron candidates differs from the one for polyptoton. The difference lies in the addition of words starting with negation prefixes. By negation prefixes we mean the following list: [anti-, de-, dis-, in-, im-, il-, ir-, mis-, non-, un-]. According to Marneffe[22], there are various types of possible contradictions, such as: antonymy, negation, numeric, factive, structural, linguistic, etc. We consider the creation of a negation of the word by adding to it one of the negative prefixes mentioned above. Therefore, we include the words forms created by such a negation only before the antonymy relation occurred in our relation paths. This prevents us from an undesirable double-negation. Similarly, before the antonymy relation we include the negated words.

Note:

> It is important to note that we rely on the *antonymy* relation to a large extent. Oxymoron is all about combining contradictions, and antonymy is the primary relation that pertains to terms opposite in meaning. Therefore, it has to be included exactly once in each of the relation paths.

We implemented the above algorithm using a depth-first search approach on a tree which is presented in Figure 3.4.

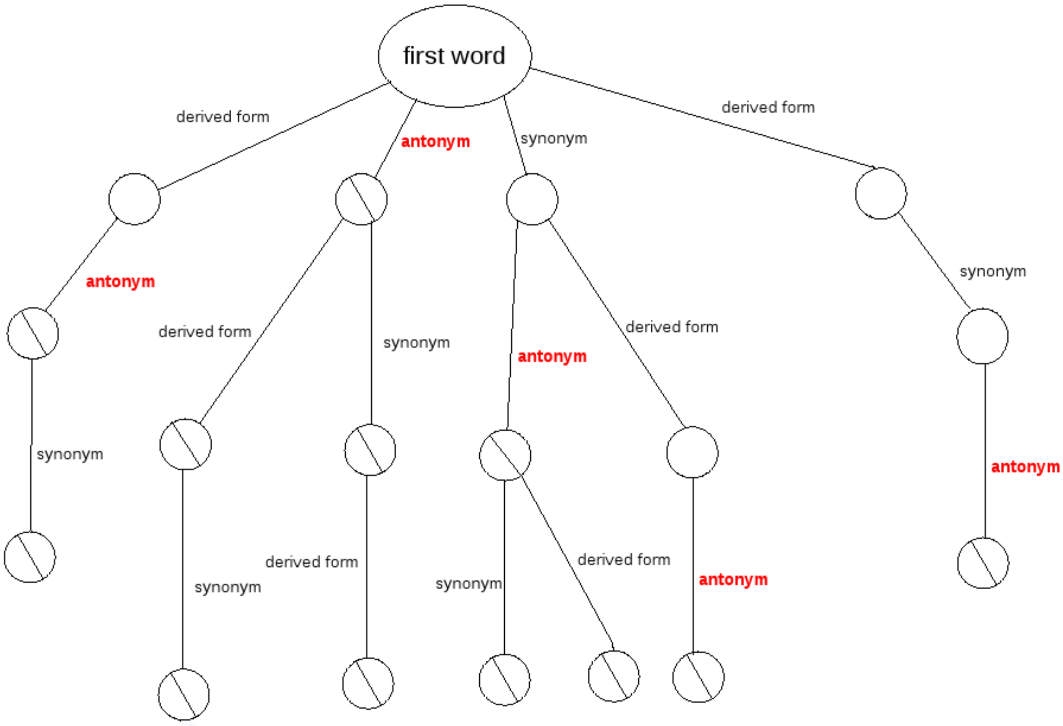


Figure 3.4: The tree of WordNet relations used for oxymoron detection

The circles with a diagonal bar indicate the nodes in which we check $S(R_n$ $(...(R1, W_1)$ against $Der(W_2)$, where $n \in \{1, 2, 3, 4\}$. The relation paths are not hardcoded, but come from the configuration file. Thus if we would like to add another path, for example, *derivation* → *synonym* → *antonym* → *synonym*, a line *[derivation, synonym, antonym, synonym]* would have to be added to the file. Section B.5 describes the details of the configuration.

3.5 Figures hierarchy

The occurrence of certain rhetorical figures trivially implicates the existence of the others. For example, a ploche is implied by antimetabole, epizeuxis, and possibly isocolon, if it contains the same words. On the other hand, isocolon is trivially implied by antimetabole, etc. Such a hierarchy is very important from the rhetorical viewpoint. A figure implied by other figures does not carry the same rhetorical power compared to the situation when it is deliberately used. Therefore, in our annotation procedure these implications have to be taken into account in order to rule out some of the 'weak' pragmatic evidences. However, creation of such a conceptual framework is extremely hard as the implications of figures strongly relies on the analyzed text. For example, anaphora in general does not trivially imply ploche. However, if anaphoric phrases are very short, then the starting words should probably be considered as an example of a ploche. Harris and DiMarco[32] propose the organization of basic *scheme* concepts. In the future, we will need to extend their taxonomy in order to successfully exclude the implied figures from the annotation.

Chapter 4

JANTOR—The Annotation Tool

Our annotation tool for figures of speech is called JANTOR—Java ANnotation Tool Of Rhetoric. It has been implemented entirely in version 6 of the JAVA programming language[2]. Apart from Stanford's parser API[], we also use two APIs for WordNet searching: JAWS[1] and MIT Java WordNet Interface JWI[2]. In this section we will present some of the most important features of the tool as well as the significant design decisions. The extended user guide (see Appendix B) provides a full description of its capabilities.

4.1 Annotation

JANTOR supports two working modes simultaneously. The first is the *annotation* mode. One of the most important features of the tool is the possibility of entering annotations that comprise an infinite number of parts, which can overlap, be embedded within each other or be placed completely separately. A user performs the marking of text either from scratch or opens an existing annotation file and continues editing. At any time she can delete the whole figure, meaning that all constituent parts will be erased. At any time the type of any annotated rhetorical figure can be changed. Additionally, each rhetorical figure that has been marked in the text can be associated with the name of the annotator and a *pragmatic cue*, which is a small piece of information explaining, for example, the purpose or meaning of the particular annotation. This information might be very useful for more accurate analysis of the rhetorical features and effect of the text.

[1] Java API for WordNet Searching (JAWS), http://lyle.smu.edu/ tspell/jaws/index.html

[2] MIT Java Wordnet Interface (JWI), http://projects.csail.mit.edu/jwi/

4.1.1 Input and Output

Currently, JANTOR supports two formats of the input file: HTML or XML annotation file. If HTML is loaded, the annotation starts from scratch. An XML file on the other hand consists of all the annotation information, which is described in more detail in the next section, together with the name of the corresponding HTML file. Thus every XML file should be associated with the HTML on which the annotation was performed. Please refer to Appendix B for more details. One of the nice features of the tool is its modularity. It supports multiple annotations at the same time. We can open many files, annotate rhetorical figures in one file, citations in the other (mentioned in related work chapter 2), or mark the same type of pragmatic evidence in all of them.

4.1.2 Annotation Schema

A significant implementation feature of the annotation tool is that all information concerning annotations is saved in a separate XML file. This concept is known as a *stand-off annotation* and enables multiple markup of the same document because the original text (in our case of the HTML document) remains unchanged. We are heading towards creating an annotation schema which would be in basic compliance with UIMA—IBM's Unstructured Information Management Architecture[26]. Let us have a look at the individual parts of the annotation schema. Each element has its own ID attribute called $xmi : id$ used for cross-referencing. We will be using the sentence presented in example 3.16 on page 35.

Annotator

$< rhe : Annotator\ xmi : id = "1"\ name = "Jakub"/ >$

There might be as many annotators as there are figures in the document, provided that each of them has been marked by different person. In the above example the annotator is called *Jakub*.

Document

$< rhe : Document\, xmi : id = "2"\ sha1 = "d3558b86a7b211f852c9eeada6eb1aecd40b8007"$ $sofaUri = "path\_to\_some\_html\_document"/ >$

Tag *rhe:Document* has two additional (apart from id) attributes. $sha1$ is the SHA-1 digest of the text included in the HTML document, and $sofaUri$ indicates the path to the HTML document on which the annotation was performed. A “sofa” is a “Subject Of Analysis”—UIMA terminology for the object that we annotate.

Figure of Speech

$< rhe : Figure\ annotator = ”1”\ xmi : id = ”4”\ sofa = ”24, 25”\ type = ”Isocolon” / >$

Tag *rhe:Figure* describes one rhetorical figure marked in the text. The attribute *annotator* refers to the annotator who marked this figure. $sofa$ indicates the IDs of all the *range* tags (described below) of that figure. Attribute *type* indicates the name of the figure (in the above example it is an isocolon).

Range

$< rhe : Range\ beginChar = ”149”\ endChar = ”178”\ xmi : id = ”24”\ sofaFeature = ”text”\ sofaObject = ”2”\ surface = ”\text{We shall fight on the beaches}” / >$

$< rhe : Range\ beginChar = ”183”\ endChar = ”217”\ xmi : id = ”25”\ sofaFeature = ”text”\ sofaObject = ”2”\ surface = ”\text{shall fight on the landing grounds}” / >$

Textual content is the feature of the annotation—the value of the attribute *sofaFeature*. *sofaObject* refers to the object of annotation, which is the HTML document. Attribute *surface* is the annotated text itself. Finally *beginChar* and *endChar* attributes indicate the starting and ending byte offset of this particular annotation in the original HTML document. Finding these offsets turned out not to be trivial and therefore we would like to devote a short section to the offset detection procedure below.

Aligning Algorithm

We use Java’s JEditorPane text component to display HTML documents. The problem we encountered is that the text that is displayed appears as in a web browser, so all the tags are removed. Additionally, JEditorPane tries to do some corrections to the existing HTML code, like adding closing tags if they are missing. As a result the character offsets in the rendered text do not correspond to the offsets in the original HTML code. The pseudocode in Algorithm 4.1 finds the offsets of the characters provided by JEditorPane in the source HTML document. The algorithm was initially implemented in Perl by Matthew Skala, a member of the Inkpot Natural Language Research Group, and then incorporated in the annotation tool.

The only preprocessing step we perform is the removal of the content of the $< HEAD >$ element from the original HTML document as whatever is within $HEAD$ tag is not displayed in JEditorPane.

The result of Algorithm 4.1 is the offset translation map M_{trans} (line 25). If we want to look up the offset $newOffset$ from the new file in the old file, we only need to perform the following two steps:

1. Find the closest index i in M_{trans} such that $M_{trans}.getNewOffsets(i) >= newOffset$
2. $oldOffset \leftarrow newOffset - M_{trans}.getNewOffset(i) + M_{trans}.getOldOffset(i)$

oldOffset is the byte offset in the original HTML of the offset *newOffset* in the file coming from the JEditorPane. The algorithm can be applied to the 'alignment' of any two files, where one is in some way a formatted version of the other.

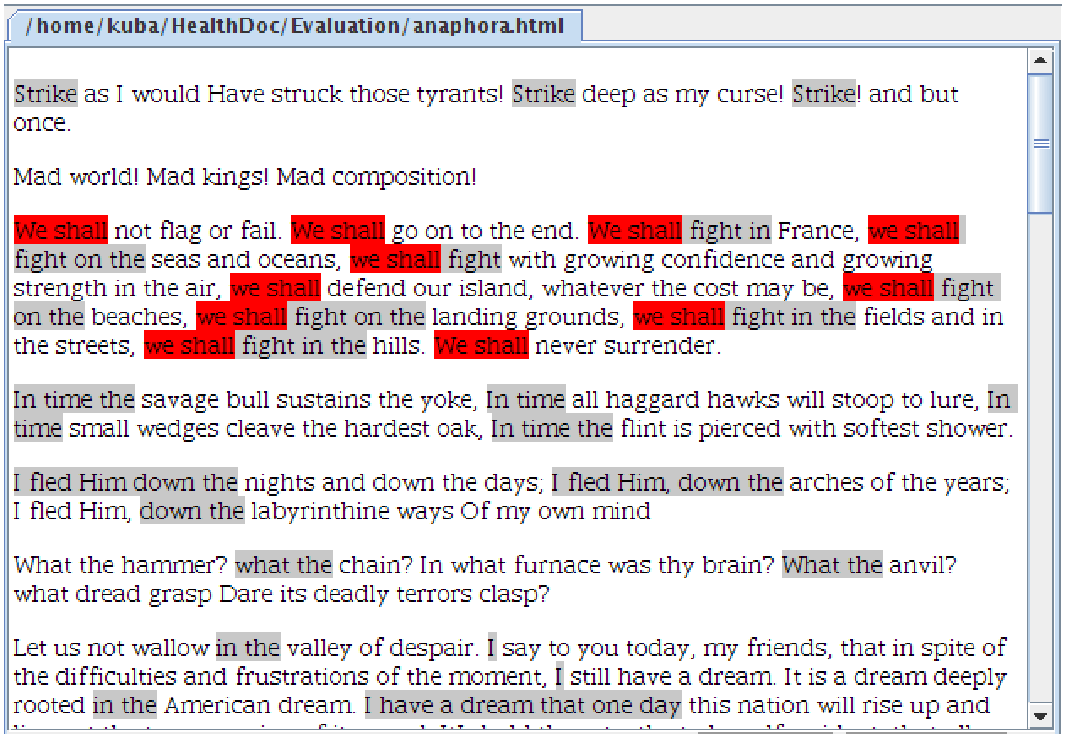
We decided to store the annotations in a separate file for various reasons. First, as we mentioned before, the original HTML text is not touched. Therefore, we do not introduce unnecessary marking characters in the source we work on, which makes it possible for any number of annotators to work on the same document. Secondly, such a solution enables full serialization. Once the HTML source has been annotated, it can be modified at any time. An example of a annotated file is presented in Figure 4.1 on page 50. The entire annotation procedure and other possibilities offered by JANTOR are presented in Appendix B.

4.2 Navigation

The other working mode is the *navigation mode*. There are a couple of important features of the navigation panel (see Figure B.3) that enable the user to move among the marked rhetorical figures. First, he can choose an arbitrary subset of figures that should be displayed. Once the type of figures has been selected the user can navigate through all the annotations in the text. At any time, the user can select/deselect the type figures to be shown. Additionally, when a new figure is added, its type is automatically selected to be displayed. Figure 4.1 presents the sample annotation of a text. The currently selected figure is marked in red, and all the others that should be displayed are highlighted in grey. More details about the navigation can be found in Appendix B on page 93.

Algorithm 4.1 Find offsets

```
create empty maps $M_{old}(offset, token)$ and $M_{new}(offset, token)$
while there is token (word or tag) $T$ in the original HTML document do
    put ($T$'s starting_offset, $T$) in $M_{old}$
end while
while there is token (word or tag) $T$ in the new HTML document do
    put ($T$'s starting_offset, $T$) in $M_{new}$
end while
$skip = 0$
create empty offset translation map $M\_trans(offset, offset)$
while true do
    if $M_{old}.empty()$ or $M_{new}.empty()$ then
        break
    end if
    if $|M_{old}| + |M_{new}| < skip$ then
        break
    end if
    $matched\_flag = 0$
    for $i = 0$ to $i < skip$ do
        $j \leftarrow skip - i$
        token $x \leftarrow M_{old}.get(i)$
        token $y \leftarrow M_{new}.get(j)$
        if tokens $x$ and $y$ are the same then
            $skip \leftarrow 0$
            $matched\_flag \leftarrow 1$
            put ($x.stating\_offset()$, $y.starting\_offset()$) to $M_{trans}$
            $M_{old} \leftarrow M_{old}.subMap(i + 1)$
            $M_{new} \leftarrow M_{new}.subMap(j + 1)$
        end if
    end for
    if token not matched: $matched\_flag = 0$ then
        $skip \leftarrow skip + 1$
    end if
end while
```



Strike as I would Have struck those tyrants! Strike deep as my curse! Strike! and but once.

Mad world! Mad kings! Mad composition!

We shall not flag or fail. We shall go on to the end. We shall fight in France, we shall fight on the seas and oceans, we shall fight with growing confidence and growing strength in the air, we shall defend our island, whatever the cost may be, we shall fight on the beaches, we shall fight on the landing grounds, we shall fight in the fields and in the streets, we shall fight in the hills. We shall never surrender.

In time the savage bull sustains the yoke, In time all haggard hawks will stoop to lure, In time small wedges cleave the hardest oak, In time the flint is pierced with softest shower.

I fled Him down the nights and down the days; I fled Him, down the arches of the years; I fled Him, down the labyrinthine ways Of my own mind

What the hammer? what the chain? In what furnace was thy brain? What the anvil? what dread grasp Dare its deadly terrors clasp?

Let us not wallow in the valley of despair. I say to you today, my friends, that in spite of the difficulties and frustrations of the moment, I still have a dream. It is a dream deeply rooted in the American dream. I have a dream that one day this nation will rise up and

Figure 4.1: Annotation of anaphora

4.3 Detection

Running the detection procedure is very straightforward. Once the XML or HTML file has been loaded the user can select which figures should be detected. After the identification is finished all of the detected stylistic devices are presented as shown in Figure 4.1. The most time-consuming part is segmenting and parsing the sentences. However, it is only required once, during the first detection. After the first run the execution time of the figure identification procedures is significantly faster. Additionally, we do not call the derived-forms finder algorithm until it is necessary, which is when the user asks for the discovery of polyptotons. Once the derived-form repetitions of a word have been found, they are also kept in memory. A walkthrough of the detection feature of the tool is described in detail in Appendix B on page 93.

4.4 Visualization

The last feature of JANTOR we present is a means of graphically representing the placement of rhetorical figures in text. The rhetorical visualization of a text should represent its content and meaning to the analyst without their having to read through it in the normal manner[70]. The main factors that determined the type of the visualization was the structure of a text with marked annotations—it is simply a string of characters, some of which are distinguished by being part of a rhetorical figure. Such an approach limits the presentation possibilities to the linear representation with some indicators of pragmatic evidence. Figure 4.2 presents an example of a visualization of a HTML document text with selected figures of speech marked in different colours.

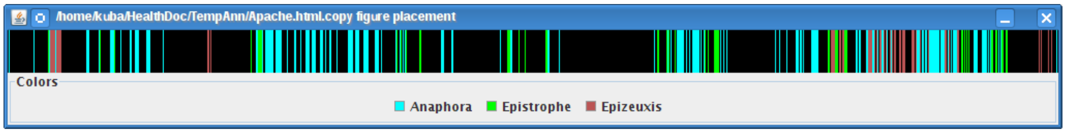

/home/kuba/HealthDoc/TempAnn/Apache.html.copy figure placement
Colors
Anaphora
Epistrophe
Epizeuxis

Figure 4.2: Visualization of rhetorical figures in text

In the above example, anaphoras are marked in light blue, epistrophes in green, and epizeuxis in light red. Obviously the image can become very cluttered if there is an abundance of figures of different types in the text. Therefore, it is more desirable to visualize only up to three figures at a time, unless they are well-spread over the document. However, a very significant solution provided in JANTOR is the option to turn on and off any of the types. So even if there are very many annotations of various types in a single visualization of a document, and the user wishes to display all of them, she can still choose an arbitrary subset of the figures to be shown.

This sort of visualization can provide us with answers to numerous questions, such as:

1. Do certain types of figures usually occur in specific places in a text?
2. Does the positioning somehow depend on the genre?
3. Which kind of figure is the most common for a certain genre?
4. Does the existence of certain figures imply the occurrence of the others?
5. Is there any observable pattern concerning figures of speech in the writings of a particular author?

We use visualization to help us in investigating a couple of these questions when we examine some of the inaugural speeches of the American Presidents in Section 5.2.

4.4.1 Choice of colours

"A picture is worth a thousand words" is a saying/cliché that comes from an old Chinese proverb "A Picture's Meaning Can Express Ten Thousand Words". In order for an image to be informative enough though it is extremely crucial to choose a good set of colours so that different parts of a picture are easily distinguishable. If the presentation of the position of figures in a text is the overriding goal of our visualization, then it is not important which colour we choose—it just has to be in contrast with the background. However, not only the placement but also the nature of a figure of speech is the significant part of our examination. JANTOR handles the detection of eleven figures so far, and we discovered that selecting a good set of well-contrasting colours, even for such a small number, is almost impossible. Therefore, if a text consists many different figures, we should depict a maximum of two or three types at a time. We also decided to use the *alpha* parameter[59]. In computer graphics, the alpha parameter introduces partial transparency to the appearance by combining an image with its background. When colours overlap, the intensity of the intersecting area is increased—in other words, the colours are additive. This approach enables us to spot any overlapping between figures because they are not covered by each other. Instead, if we apply colour composition theory, we can conclude from the composed colour which figures overlapped with each other. Table 4.1 presents the colours we have assigned to different figures.

Table 4.1: Colours of the rhetorical figures

Figure	Description	Colour
anadiplosis	brown	🟧
anaphora	cyan	🟦
antimetabole	blue	🟦
epanalepsis	bright magenta	🟪
epistrophe	green	🟩
epizeuxis	bright red	🟫
isocolon	light grey	⬜
oxymoron	red	🟥
ploche	yellow	🟨
polysyndeton	orange	🟧
polyptoton	magenta	🟪

Chapter 5

Results and Evaluation

5.1 Testing documents

We created a small collection of eleven HTML documents for testing the precision and recall of our rhetorical-figure detection system. Each of the files consists of the examples of one particular type of figure. The examples were taken from various sources—excerpts from political speeches, commercials, the Bible, poetry, prose, lyrics, film dialogues, etc. As sources we used the Wikipedia[69], About.com[7], American Rhetoric[8], Silva Rhetoricae[1] websites, and the well-known Corbett text on classical rhetoric[16].

Each of the following sections presents the experiments conducted for individual figures. Additionally, in order to thoroughly check the performance of JANTOR, we applied the tool to detect all the possible figures for each of the files. Although individual text documents from the collection were initially prepared to check only one type of figure, other figures also exist in them and therefore should be annotated. We comment on the results of precision and recall, and discuss the errors of the detection. Additionally, we provide some ideas for improvement and extensions of JANTOR's detection system.

5.1.1 Experiments

Epizeuxis

Let us start with the evaluation of *epizeuxis* discovery. Our HTML document consisted of 37 examples of this figure. As we mentioned before in Section 3.2.2, the detection of epizeuxis is straightforward—our system was able to correctly find all 37 figures. However, there was one part missing in one of them. Let us have a look at example 5.1.

Example 5.1:
. . . You need to hear the most important message thus far in the third millennium. You need to hear a maxim so simple, so clear, and so evocative that no one could misconstrue its meaning or miss its weighty issue–so, here goes. It's not a statement, but it's a request, It's not a bit of advice, but it's a plea: ***Help***. ***HELP***. ***HEEEELLLLLLPP***.[1]

Our system was unable to detect the last part, namely "HEEEELLLLLLPP", of the bolded epizeuxis above. From a pragmatic point of view this part should have been annotated because it is part of the epizeuxis, too. The definition of epizeuxis in Section 3.2.2 mentions the repetition of words. "HEEEELLLLLLPP" is not equal to "HELP", thus it has been omitted. However, in future such a situation should be taken into consideration as it introduces a huge emphasis on what is being said and is important from the rhetorical perspective. The system needs to be attuned to orthographic variables.

We relaxed the aforementioned definition of epizeuxis and also take into account not only repeating words but also phrases with no others between. Example 5.2 shows such a situation. Bolded phrases were identified by our system.

Example 5.2:
All around me are familiar faces
Worn out places, worn out faces
Bright and early for their daily races
***Going nowhere**, **going nowhere**.*
And their tears are filling up their glasses
No expression**, **no expression
Hide my head I want to drown my sorrow
***No tomorrow**, **no tomorrow**.*[2]

Ploche

The next figure that we examined was ploche. As for epizeuxis, all of the ploches in the text were identified. Still questionable though is the distance to use between

[1] Tom Hanks, excerpt from *Commencement Address at Vassar College*

[2] "Mad World", *Tears for Fears*

occurrences of words. As we mentioned in Section 3.2.2, we consider a ploche, a repetition of words, if and only if the word is repeated at least twice in the same sentence or two neigbouring sentences. Now, it might happen that the sentences are very short—two-word or three-word expressions. Perhaps in this case the author deliberately repeated a word in these expressions and this should be recorded as a usage of a ploche, even though the word did not occur in neighouring sentences. Let us have a look at the following example.

Example 5.3:
[1]*He* ***speaks****.* [2]*She* <u>*writes*</u>*.* [3]*Then she* ***speaks****.* [4]*He* <u>*writes*</u>*.*

Although the words *speaks* and *writes* occur in sentences 1, 3 and 2, 4 respectively, and our detection system, due to its definition "sentence distance" omits them, they still constitute a ploche. Therefore, as well as the word-distance measure for sentences we need to introduce a maximum number of characters allowed between repeating words. Although the situation in example 5.3 was created only for presentation purposes, it can definitely happen in real texts, and thus the use of character distance between repeating words should be implemented in future versions of JANTOR.

Polysyndeton

Polysyndeton is the next figure in our list. Our system was able to correctly recognize all 20 examples, but it missed a couple of conjunctions in one figure. Below we provide the example that caused the error and discuss the reason for the mistake.

Example 5.4:
And she pushed St. Peter aside ***and*** *took a peek in,* ***and*** *there was God–with a plague in one hand* ***and*** *a war* ***and*** *a thunderbolt in the other* ***and*** *the Christ in glory with the angels bowing,* ***and*** *a scraping* ***and*** *banging of harps* ***and*** *drums, ministers thick as a swarm of blue-bottles, no sight of Jim [her husband]* ***and*** *no sight of Jesus, only the Christ,* ***and*** *she wasn't impressed. And she said to St. Peter This is no place for me* <u>*and*</u> *turned* <u>*and*</u> *went striding into the mists* <u>*and*</u> *across the fire-tipped clouds to her home.*[3]

The bolded conjunctions constitute one polysyndeton and the underlined ones the second. Two *and*'s that have been omitted by the system are marked in red.

[3] Ma Cleghorn in *Lewis Grassic Gibbon's Grey Granite, 1934*

The reason for such behaviour is that Java's *BreakIterator* did not correctly split the paragraph into sentences. *And she pushed St.* as well as *And she said to St.* are mistakenly considered as sentences, which should not be the case. This error in sentence boundary detection caused the wrong behaviour of our algorithm. In Section 3.2.2 we mentioned that conjunctions must either be repeated in one sentence or begin two or more successive sentences in order to be considered a polysyndeton. Because of the wrong text segmentation neither of the conditions were met. In the following example, on the other hand, both cases were correctly identified.

Example 5.5:
And *God said, Let the earth bring forth the living creature after his kind, cattle,* ***and*** *creeping thing,* ***and*** *beast of the earth after his kind:* ***and*** *it was so. And God made the beast of the earth after his kind, and cattle after their kind, and every thing that creepeth upon the earth after his kind: and God saw that it was good.*[4]

In the sample passage 5.5 there were three polysyndetons discovered. The first one is bolded, the second one is underlined, and the third one is coloured in red. The last one is the example of two conjunctions beginning two (or more) consecutive sentences. Our algorithm is designed to recognize all three of these cases. However, the above case might also have been merged into two figures by simply excluding the last case of polysyndeton. In general though, it might not be the case that the conjunctions starting the sentences are also repeated within them.

Anaphora

Let us now have a look at the first figure that concerns not only the repetition of a group of words, but which also specifies a certain position in a phrase or sentence within which it has to be placed—*anaphora*. There were a couple of issues pertaining to the discovery of anaphora that had to be thoroughly examined and discussed. First, we had to decide the minimum length of phrases or clauses which should be taken into consideration. Initially, we thought that a length of four would be a reasonable choice. However, there were cases in the prepared examples for which an anaphora was evidently present in two-word phrases. Let us have a look at example 5.6.

Example 5.6:
Mad *world!* ***Mad*** *kings!* ***Mad*** *composition!*[5]

[4] Genesis 1:24-25 (KJV)

[5] William Shakespeare, *King John, Act II, Scene 1*

The example above is "The Bastard's speech" with which Shakespeare ends Act II of King John. The three phrases shown in example 5.6 do not form sentences, yet still contain an anaphora which has been deliberately used for emphasis. When the minimum phrase length was decreased to two, it was detected, but as a result the precision of the system has fallen too. Let us have a look at the following example.

Example 5.7:
*Never shall I forget those moments which murdered **my** God and **my** soul and turned **my** dreams to dust.*[6]

The parse tree of the sentence presented in example 5.7 is shown in Figure 5.1.

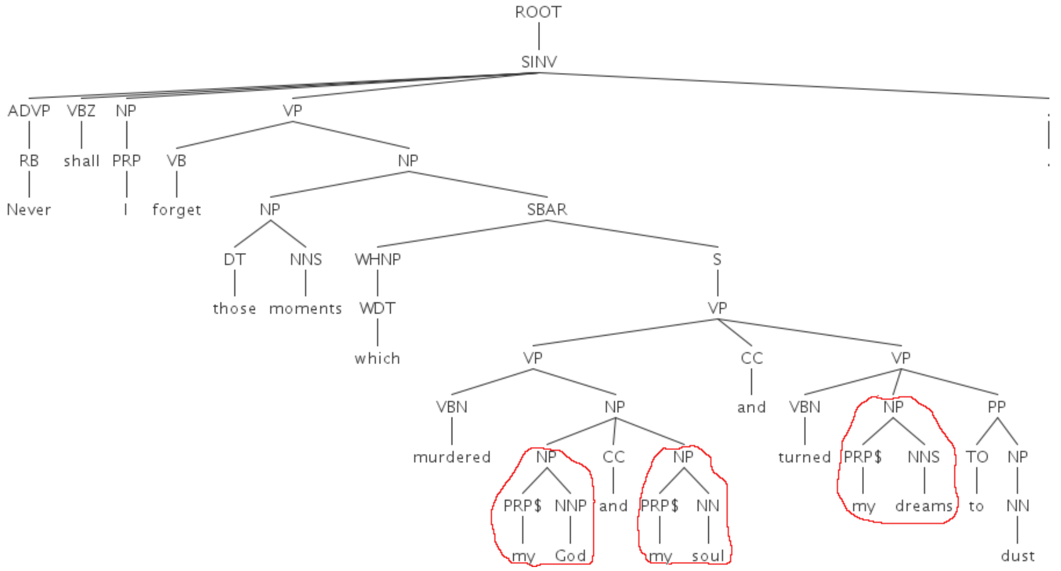


Figure 5.1: Parse tree of a sentence with anaphora for minimum phrase length equal 2

From the technical viewpoint the situation presented in Figure 5.1 should be considered as an example of anaphora. The three two-word noun phrases circled in red start with the same word—*we*. However, from the rhetorical viewpoint this is not necessarily an example of an anaphora. Larger syntactic units (for example, the verb phrases placed above the mentioned noun phrases) should be taken into consideration. Even more controversial is the situation presented in example 5.8 below.

[6] Elie Wiesel, excerpt from *Night*

Example 5.8:
It was the best of ***times****, it was the worst of* ***times****...*[7]

The parse structure of the above text segment is presented in Figure 5.2.

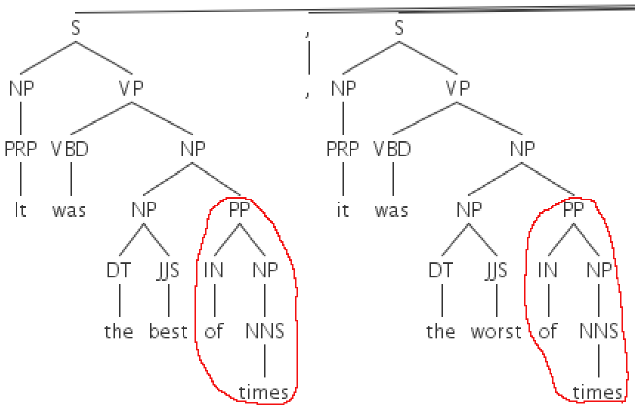


Figure 5.2: Incorrect detection of anaphora

As we mentioned in Section 3.2.2, we ignore all the determiners, conjunctions, and prepositions starting a phrase or clause. In the example shown in Figure 5.2, the two two-word phrases circled in red start with a preposition *of*. When those are omitted, we are left with the word *times*, which should be marked as an epistrophe rather than an anaphora. We address this problem by ignoring a situation where we are left with only one word in a phrase after the removal of initial determiners, conjunctions and prepositions. Ideally, a phrase beginning some other phrase or clause should not be treated as part of an anaphora, if it is placed at the end of a bigger syntactic unit.

Lastly, we would like to focus attention on an anaphora consisting of different forms of a word originating from the same stem. Example 5.9 presents such a situation.

Example 5.9:
Strike *as I would*
Have ***struck*** *those tyrants!*
Strike *deep as my curse!*
Strike*! and but once.*[8]

[7] Charles Dickens, excerpt from *A Tale of Two Cities*
[8] Byron

Our system was able to detect the three *strikes*, but omitted the word *struck*. The reason for this is that we decided to remain strict with the definition of repetition of words in the same form, as far as figures of repetition are considered. We will come back to this issue later in this section when performing the evaluation of polyptoton detection.

Epistrophe

A figure symmetrical to anaphora with respect to position in a sentence is epistrophe. We can report very high recall and precision for detection of this figure judging from the performance on our examples. Here we would like to emphasize one of the problems with lexicalized probabilistic context-free grammar parsers we pointed out in Section 3.1.3. Let us have a look at the following example.

Example 5.10:
Caesar has his province: there are laws which govern property, maritime ***law****, fiscal* ***law****, theological* ***law****—which determines the lengths of robes and so forth.*[9]

The parse tree of the excerpt from example 5.10 is shown in Figure 5.3 on the following page.

In the red circles we have marked phrases with an epistrophe: *maritime law*, *fiscal law*, *theological law*, all finish with the word *law*. However, when we look at the parse structure presented in Figure 5.3, we notice that all three of the above expressions do not constitute separate noun phrases. Such a case is an exemplification of the situation in which we cannot rely on the parser. Therefore, we have come up with a very simple, but successful in many cases, heuristic, which divides the sentence into phrases placed between punctuation markers. In the above example, commas helped us with finding the epistrophe. In future, we will have to develop a more reliable approach towards phrase boundary detection, as punctuation markers are not always existent, and even if they are, are not necessarily good indicators of the beginning or ending of a phrase.

The following example is exactly such a case where the parser did not expand the nodes correctly, and there is no indication that the repetition of words is positioned at the end of a phrase or clause.

Example 5.11:
What lies behind ***us*** *and what lies before* ***us*** *are tiny compared to what lies within*

[9] Timothy Leary, *LSD: Methods of Control*

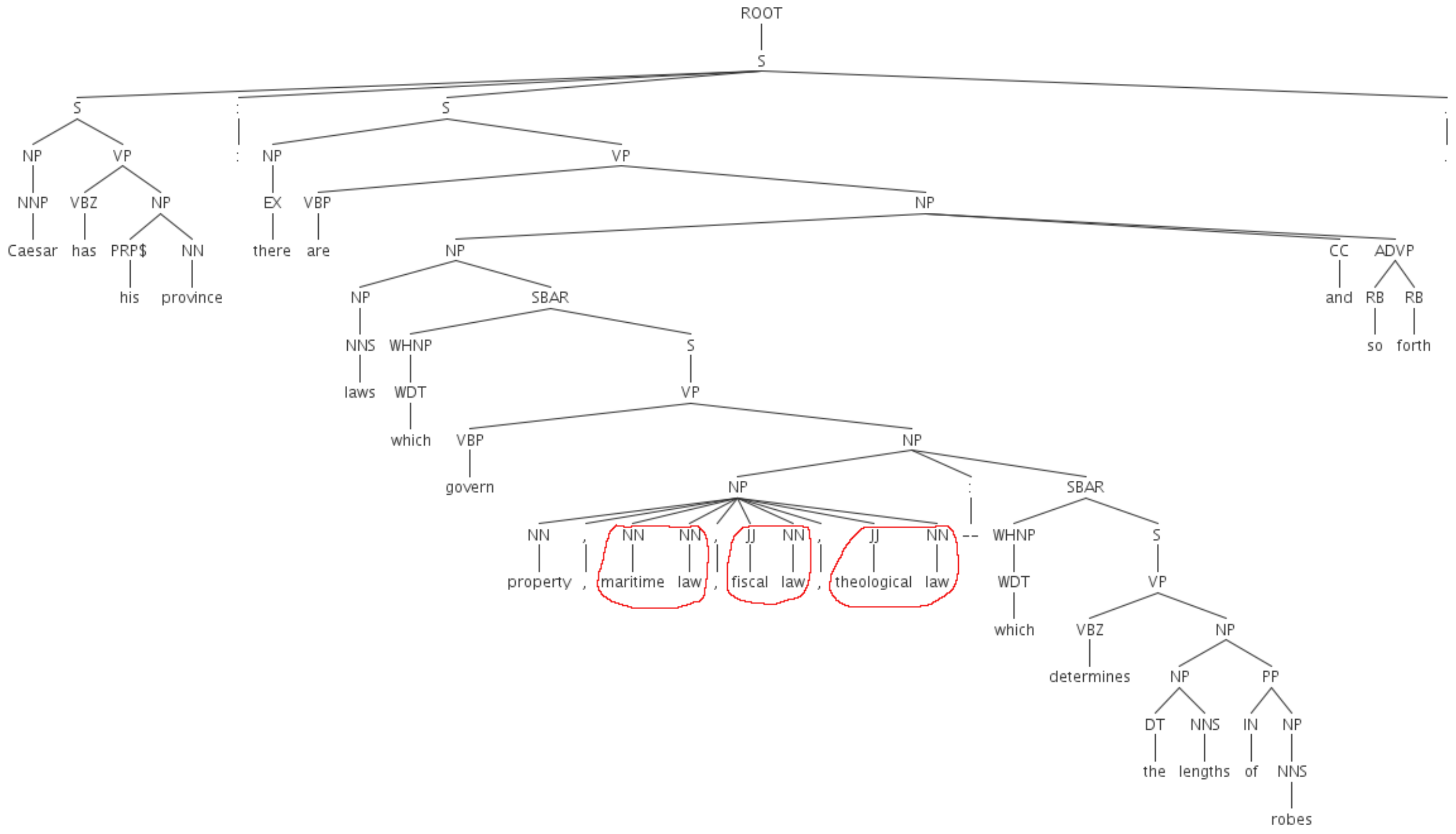

ROOT
S
S : S .
NP VP NP VP
NNP VBZ NP EX VBP NP
Caesar has PRP$ NN : there are NP CC ADVP
his province NP SBAR and RB RB
NNS WHNP S so forth
laws WDT VP
which VBP NP
govern NP : SBAR
NN , NN NN , JJ NN , JJ NN -- WHNP S
property , maritime law , fiscal law , theological law which VP
VBZ NP
determines NP PP
DT NNS IN NP
the lengths of NNS
robes

Figure 5.3: Example of incorrect parsing with respect to the detection of an epistrophe

us.[10]

The parse tree of the above sentence is presented in Figure 5.4. The erroneously omitted word *us* is circled in red. The parser not only did not put it at the end of a phrase, but there is no specific indication, such as a punctuation marker, that this word might be the ending of some phrase. Below we present one of the possible ways to tackle this problem.

Let W be the word we are testing whether it is placed at the end of a phrase. We assume that at this point our system failed to find the word at the end of a phrase using parse-tree structure as well as punctuation markers.

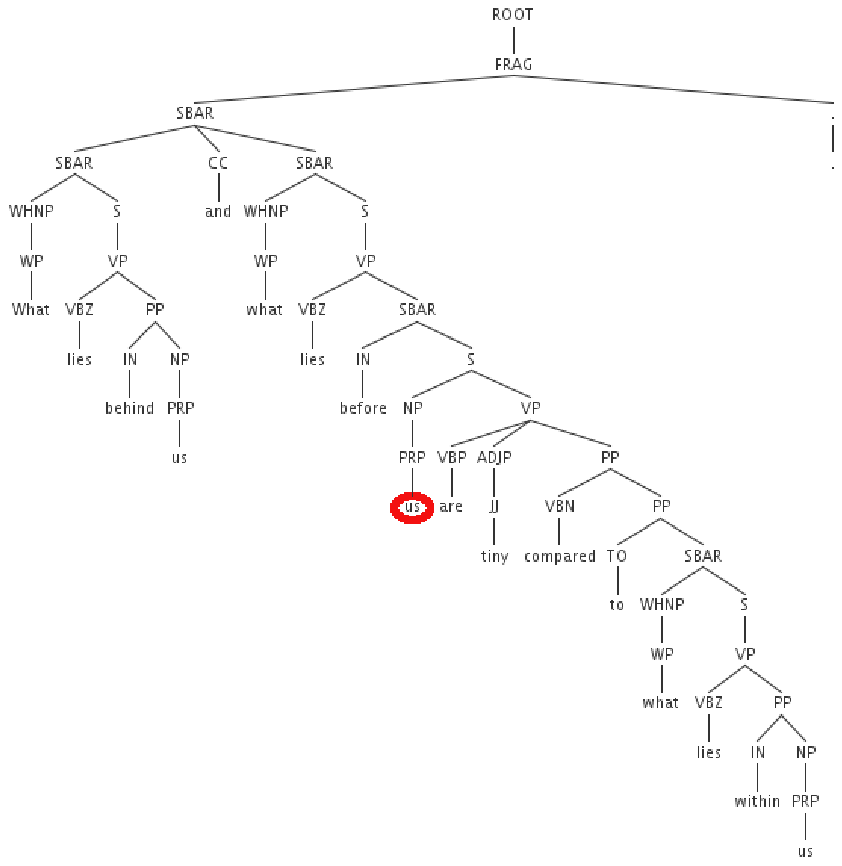


Figure 5.4: Example of hard-to-detect epistrophe.

1. Check whether W was previously recorded at the end of some other phrase.

[10] Ralph Waldo Emerson

2. If condition in step 1 is met, check whether the next word in the sentence is at the beginning of another phrase.

3. If so, W is very probably part of a epistrophe

The problem with the above approach is that it produces a lot of false positives. In future a more sophisticated algorithm needs to be devised. However, in most of the examples we examined, parse structures and punctuation markers were sufficient to correctly identify the boundaries of the syntactic units.

Epanalepsis

The detection of epanalepsis was quite successful on most of the prepared examples, so here we would like to highlight only one important case, where the system failed to find it.

Example 5.12:
The man who did the waking buys the man who was sleeping a drink; the man who was sleeping drinks it while listening to a proposition from ***the man who did the waking****.*[11]

Again, let us have a look at the parse-tree structure of the above sentence in Figure 5.5, with the phrases of interest circled in red, which were mistakenly recognized as the parts of the epanalepsis. In the same figure we marked in green the phrases that indeed constituted this figure.

Here it occurred accidentally that two groups of the same words—*the man who was sleeping*—begin and end the same noun phrase (circled in blue). In the previous examples we were mainly concentrating on the false negatives. Here we can see the example of a false positive of an epanalepsis, which is actually an example of anadiplosis (described below). The undesired behaviour of the system resulted from the incorrect parse of the sentence. We are not sure at this point how to address this problem. Including the detection of, for example, anadiplosis, and implying that the same group of words cannot be epanalepsis and anadiplosis at the same time would not work because we would not be able to determine which figure is the correct one. And what if anadiplosis is not existent? In short, it is much easier to handle false negatives than ignoring false positives. In future we will have to devise more sophisticated pruning algorithms.

[11] Jack Sparrow, *The Pirates of the Caribbean*

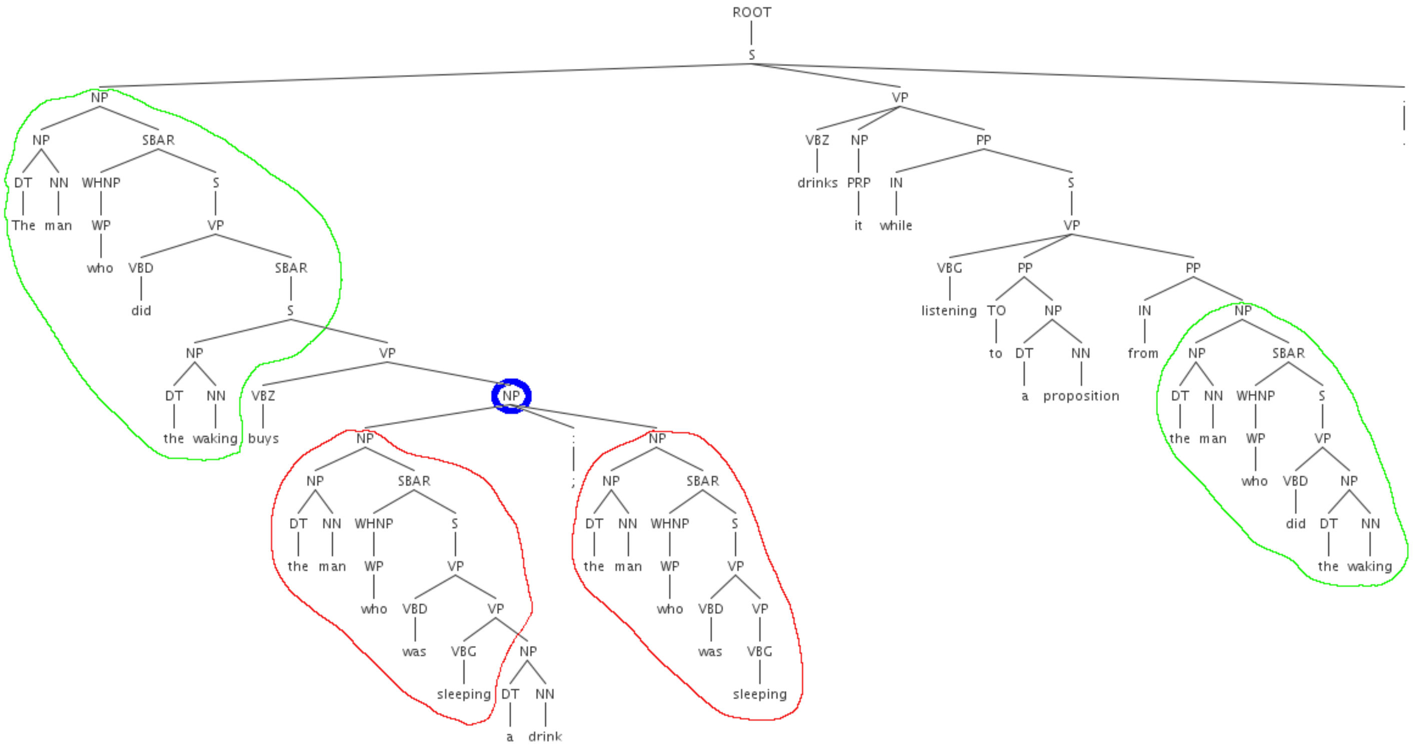

ROOT
S
NP VP .
NP SBAR VBZ NP PP
DT NN WHNP S drinks PRP IN S
The man WP VP it while VP
who VBD SBAR VBG PP PP
did S listening TO NP IN NP
NP VP to DT NN from NP SBAR
DT NN VBZ NP a proposition DT NN WHNP S
the waking buys NP : NP the man WP VP
NP SBAR NP SBAR who VBD NP
DT NN WHNP S DT NN WHNP S did DT NN
the man WP VP the man WP VP the waking
who VBD VP who VBD VP
was VBG NP was VBG
sleeping DT NN sleeping
a drink
;

Figure 5.5: Example of wrongly categorized epanalepsis

Anadiplosis

Out of 49 examples of anadiplosis in our test file our system was able to correctly identify 47. Below we present the two cases where it failed to detect this figure.

Example 5.13:
Only the brave deserve ***the fair*** *and* ***the fair*** *deserve Jaeger.*[12]

The parse tree of the sentence from example 5.13 is presented in Figure 5.6.

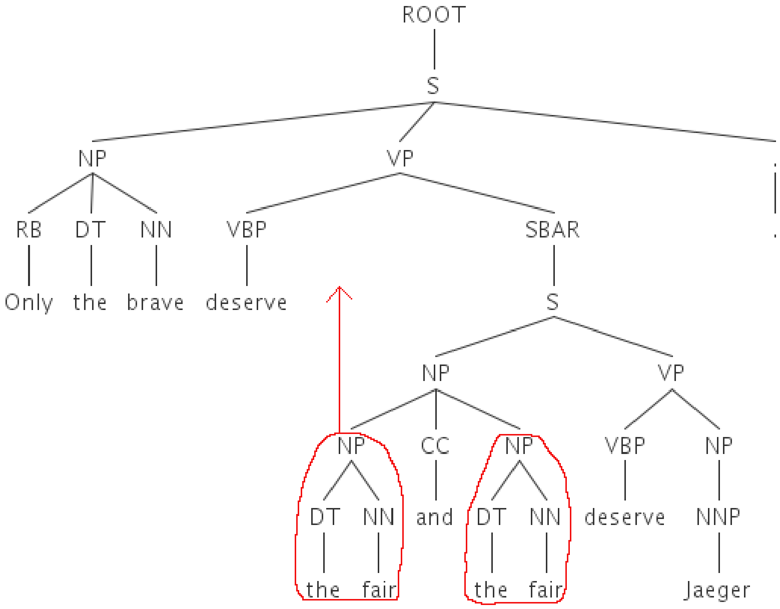


Figure 5.6: Undetected example of anadiplosis

In red we have marked a group of words (part of the missed anadiplosis) that was incorrectly put under the SBAR clause in the parse tree. Because the first *the fair* phrase was not placed at the end of any other phrase or clause, the second *the fair* was omitted. By accident, it just happens that these two words begin and finish (at the same time) the neighbouring phrases so an anadiplosis was detected, but in general this parsing error would cause the omission of the figure.

The second example concerns more the definition of anadiplosis. Let us have a look at the following sentence.

Example 5.14:
Somehow, with the benefit of little formal education, my grandparents recognized the

[12] advertising slogan for Jaeger Sportswear

inexorable downward spiral of conduct outside the guardrails: If you lie, you will cheat; if you cheat, you will steal; if you steal, you will kill.[13]

Red and green are two instances of anadiplosis that were omitted by JANTOR. We decided to skip the determiners, conjunctions and prepositions if they start a repeated group of words. In the above example, however, the word *will* is a modal verb, and the word *you* is a pronoun. Therefore, if we do not remove them the only repetitions that will be found are the words *cheat* and *steal*, but they are not placed at the end and the beginning of phrases in close proximity (see Figure 5.7), thus are not taken into account.

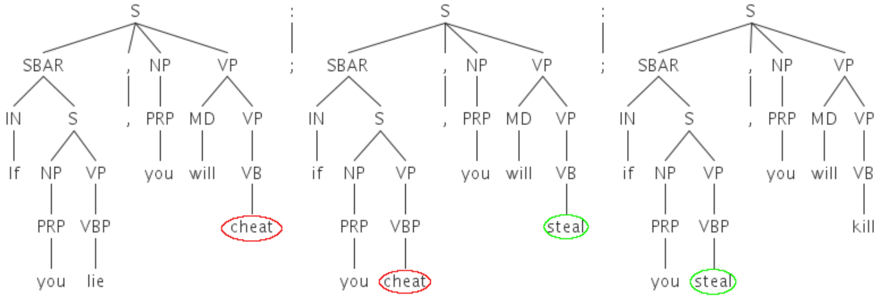


Figure 5.7: Example of missed anadiplosis

Antimetabole

The definition of antimetabole we provided in Section 3.2.2 states that it is the repetition of words in successive clauses in reverse grammatical order. However, the definition does not precisely explain whether all kinds of words should be taken into account, nor whether different forms of a word should still be considered the same word. We address these two points in the following way:

1. We take into consideration all kinds of words, including determiners, conjunctions, and prepositions.

2. We do not look at different forms of a word, but only at repetitions of exactly the same words.

The approach described in the first point one produces a lot of antimetaboles that are not necessarily important from the rhetorical point of view, but is necessary in order not to decrease the recall value. The example shown below explain both assumptions in more detail.

[13] USSC Justice Clarence Thomas, *1993 Mercer Law School Address*

Example 5.15:
I can write better than anybody who can write faster, and I can write faster than anybody who can write better.[14]

In the above passage there are about 20 repetitions of words in reverse grammatical order. Some of them are shown in Table 5.1.

Table 5.1: Repetitions of words in reverse grammatical order

No	Repetition
1	better. . . faster. . . faster. . . better
2	better. . . write. . . write. . . better
3	better. . . can. . . can. . . better
4	better. . . who. . . can. . . can. . . who. . . better
5	better. . . anybody. . . faster. . . faster. . . anybody. . . better
6	better. . . than. . . faster. . . faster. . . than. . . better
7	anybody. . . can. . . can. . . anybody
8	who. . . faster. . . faster. . . who

All the repetitions (many more were omitted) listed in Table 5.1 meet the constraints imposed by the definition of antimetabole. However, only a few of them are actually genuine instances of antimetabole—repetitions used for emphasis. As we mentioned before, JANTOR enables a human annotator to delete all of the false positives. Nevertheless, in future, a more precise definition of antimetabole will be necessary for it to be correctly automatically detectable by our system.

Another example is shown below.

Example 5.16:
*You can take the **gorilla out** of the **jungle**, but you can't take the **jungle out** of the **gorilla**.*

It is important that all three words be marked as parts of the same antimetabole. In other words, the saying asks, at what point does a gorilla stop being a gorilla, if at all? None of the combination of the two-word antimetaboles—*gorilla. . . out. . . out. . . gorilla*, *out. . . jungle. . . jungle. . . out*, or *gorilla. . . jungle. . . jungle. . . gorilla*—would have the same effect on the audience as the one marked in example 5.16. Therefore, it is crucial to be able to determine which of the words meeting the loose constraints imposed by the definition of the figure should actually be included in the antimetabole.

[14] A.J. Liebling

It is especially important because, as Corbett notes[16], antimetaboles very often have the air of the "neatly turned phrase"—a kind of phrasing often met in memorable aphorisms.

Now, let us have a look at a sentence that is the exemplification of the second point mentioned above.

Example 5.17:
*To be **kissed** by a **fool** is stupid; To be **fooled** by a **kiss** is worse.*[15]

There is here a non-negligible number of antimetaboles which include different forms of a word. Algorithm 3.3 on page 28, which finds different forms of a word, can be applied to finding more complicated antimetaboles, but currently we decided to remain strict with the definition of the repetition. Thus, all the expressions similar to the one presented in example 5.17 are omitted.

Finally, we would present some memorable aphorisms that were correctly identified by the system despite their relative complexity.

Example 5.18:
***Integrity without knowledge** is weak and useless, and **knowledge without integrity** is dangerous and dreadful.*[16]

*It is not the **consciousness** of men that **determines** their **being**, but, on the contrary, their social **being** that **determines** their **consciousness**.*[17]

The first expression, by English author, critic, and lexicographer Samuel Johnson is a very neat example of a powerful antimetabole, which stresses how important is the interaction between integrity and knowledge. The effect of the quotation would definitely be diminished if written instead as: *Integrity without knowledge is weak and useless, and it is crucial for integrity to come with knowledge too because otherwise it may be dangerous and dreadful.*. Fresh and apt antimetaboles undoubtedly reveal rhetoricians of great intelligence and wit, therefore it is desirable to precisely recognize them in texts.

We present the second sentence in example 5.18 not only due to its undoubted rhetorical effect, but also because it is more complex. Algorithm 3.2 is able to detect all kinds of word-pair inclusions and the above expression is a perfect example.

[15] Ambrose Redmoon

[16] Samuel Johnson, *Rasselas*

[17] Karl Mark's, *Preface to A Contribution to the Critique of Political Economy*

Even though there are other noisy words between the repetitions, the antimetabole is correctly detected.

Polyptoton

Polyptoton is the first figure in our detection set which requires the discovery of different forms of a word. In the discussion of polyptoton in Section 3.2.2 we described the algorithm for finding derived forms of a word, which uses a knowledge of the lexicon—WordNet—and the extended Porter stemmer. Out of 28 examples of polyptoton our system correctly found 24. It skipped four due to the issues described below. The following example shows the situation which for now we are unable to efficiently solve.

Example 5.19:
With eager ***feeding food*** *doth choke the* ***feeder****.*

Our system detected the words *feeding* and *feeder* but was unable to spot the word *food*. As we mentioned earlier in Section 3.2.2, at this point our system fails to detect words that appear in different parts of speech with a modified stem. Examples of such words are the verb *to feed* and the noun *food*. None of the forms produced by our algorithm for the former overlapped with any of the forms of the latter. One possible solution to this problem is using *glosses* of words. The words in WordNet come with an accompanying explanation or definition. Very often in the definition of a word its other forms are mentioned in some context. As an example, let us look at the glosses of the first seven most popular senses of the verb *feed*:

1. feed: (provide as **food**; "Feed the guests the nuts")
2. feed, give: (give **food** to; "Feed the starving children in India"; "don't give the child this tough meat) "
3. feed: (feed into; supply; "Her success feeds her vanity")
4. feed, feed in: (introduce continuously; "feed carrots into a food processor")
5. feed: (support or promote; "His admiration fed her vanity")
6. feed, eat: (take in **food**; used of animals only; "This dog doesn't eat certain kinds of meat"; "What do whales eat?")
7. feed — (serve as **food** for; be the **food** for; "This dish feeds six")

In four of these glosses, the noun *food* appeared in some context. The overlap between the glosses of words and the second word provides an important indication that they might be related[9]. We have not yet tested this approach, but we believe that in future it might significantly improve our algorithm for finding derivationally related words. One of the problems we will be faced with though is the complexity of this algorithm. Even for short sentences and after the removal of stop words, the comparison of the glosses of all the words might be unacceptably time-consuming.

In the case where the root of the word is not modified, our detection algorithm works quite well. Below we present some of the identified polyptotons.

Example 5.20:
*A good ad should be like a good sermon: it must not only **comfort** the afflicted; it also must afflict the **comfortable**.*[18]

Divine Master, grant that I may not so much seek to be consoled as to console;
To be understood as to understand;
To be loved as to love;
For it is in giving that we receive;
It is in pardoning that we are pardoned;
And it is in dying that we are born to eternal life.[19]

Our system also produced three false positives due to the issues we described earlier in Section 3.2.2. More specifically, this problem is due to the not always very precisely performing algorithm for finding derived forms of a word forms.

Isocolon

Isocolon was one of the most difficult to detect figures in our test set. As it is a series of similarly structured elements having the same length, we must rely to a large extent not only on the word count, but also on the parse-tree structures. However, it is not always entirely clear what the similarly structured phrases are. We would like to remind the reader of the two assumptions we made in the isocolon part of Section 3.2.2. We treat two phrases as similarly structured when:

1. Either they have exactly the same parse-tree structures, meaning they only differ in words, but parts of speech as well as the depth and order of the tree nodes are the same; or

[18] Bernice Fitzgibbon

[19] Prayer of St. Francis of Assisi

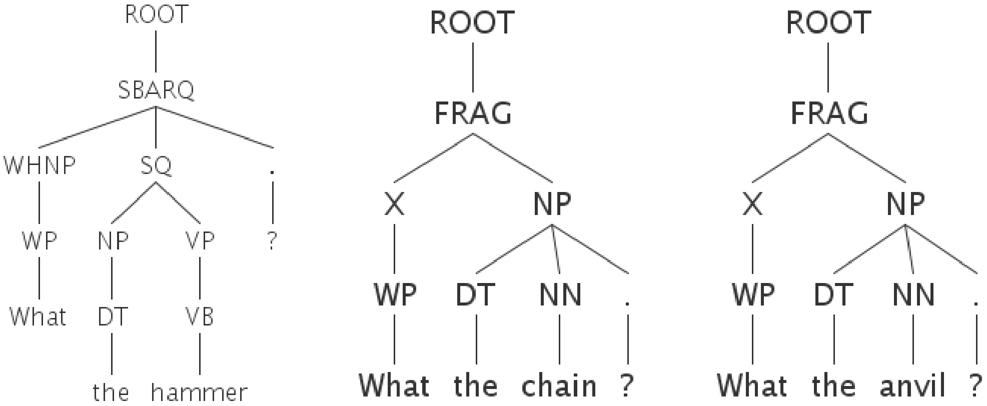
2. If condition 1 is not met, the difference between part-of-speech labels of the words in the phrases is smaller than some set threshold, usually 1.

Our isocolon detection approach performed quite well with respect to recall—only 4 out of 27 isocolons in the test set were missed. However, our precision values are only satisfactory; we explain the reasons of poor performance with respect to both precision and recall with the following examples.

Missed isocolons

Example 5.21:
What the hammer*? **What the chain**?*
In what furnace was thy brain?
What the anvil*? What dread grasp*
Dare its deadly terrors clasp?[20]

The parse trees of the bolded phrases are presented in Figure 5.8.



(a) Isocolon part A (b) Isocolon part B (c) Isocolon part C

Figure 5.8: Parse trees of isocolon compound phrases

Our detection system correctly identified phrases in 5.8b and 5.8c, but it missed the phrase in 5.8a. The reason for this is the incorrect expansion of the word *hammer* in the first phrase. The parser treated it as a verb in a base form and thus

[20] William Blake, *The Tyger*

assigned it a VB tag instead of NN (noun, singular form). As a result our tree-similarity approach did not treat these phrases as sufficiently structurally close. Additionally, for such short phrases, algorithm 3.5 requires no difference between the tags of words, and one-tag introduces a 33% discrepancy in a three-word expression. Another issue is the distance between these similar phrases, which we have already discussed before. In the text, expressions 5.8b and 5.8c are separated by *In what furnace was thy brain?*, which interferes with the notion of a sequence of parallel phrases. However, if our sliding sentence window is sufficiently large, it ignores the in-between 'noises'.

Another example phrase, taken from a commercial, in which we missed an isocolon is shown below.

Example 5.22:
It takes a licking, *but* ***it keeps on ticking!***[21]

The problem here is with the tags assigned to the words *licking* and *ticking*. The former was treated as a noun (NN) whereas the latter as a verb (VBG). Together with the difference between the words *a* (determiner) and *on* (preposition) the algorithm did not treat these two phrases as structurally similar, although having the same length.

False positives

Most of our false positives of isocolons resulted from not having a definite range within which we should look for this figure. On the one hand example 5.21 clearly encourages having the range at least two-sentences or three-sentences long. If smaller, we are prone to miss many parallel expressions which occur in successive sentences or are placed in proximity. On the other hand, if the sentences are very large, as the one from example 3.8 on page 22, then the algorithms find phrases close in structure, but positioned so far apart that the symmetry of length and structural similarity do not establish any kind of rhythm, which is one of the main purposes of using isocolon. We plan to vary the range within which we look for certain pragmatic evidence, according to both the type of the evidence (figure of speech, in particular) and the length of the syntactic units, so that the detection algorithm would adjust dynamically while scanning the document.

Oxymoron

[21] advertising slogan of Timex watches

Oxymoron is the last figure in our set, the detection of which is the most problematic mainly because it pertains more to semantics than to syntax. The main challenges we have to face in the approach are:

1. Which candidate words should be selected for the examination?
2. How to establish the notion of contradiction between the chosen words?

We address the first problem by examining grammatical relations between pairs of words, and we try to find the opposition of meaning between them using the semantic knowledge contained in the WordNet database (see Section 3.4). Most of our detection errors resulted from the inaccuracy of the procedure for finding contradictions. However, we also missed several oxymorons because, according to the typed-dependencies module of the parser there were no grammatical relations between a pair of words, which in human understanding of natural language text constitutes an oxymoron. Additionally, we obtained some false positives, mainly due to the imperfections of our algorithm for finding derived forms of a word. As we mentioned in the discussions of polyptotons in Section 3.2.2, the algorithm is prone to sometimes considering two words as derived from the same stem when they actually are not. This misleads our WordNet-based approach in some cases. Out of 52 oxymorons collected in our test file we were able to identify 42. We provide examples of the incorrect behaviour below and discuss the probable causes.

Consider the following example:

Example 5.23:
O ***brawling love****! O* ***loving hate****!*
O ***anything*** *of* ***nothing*** *first create! O* ***heavy lightness! serious vanity!***
Misshapen chaos *of* ***well-seeming forms!***
Feather of lead, bright smoke, cold fire, sick health!
Still-waking sleep*, that is not what it is!*
This love feel I, that feel no love in this.[22]

Our system correctly recognized only three oxymorons in the above passage: *heavy lightness*, *cold fire*, *sick health.* The reasons why the others were omitted vary. We explained in the oxymoron discussion of Section 3.4 how we deal with

[22] William Shakespeare, *Romeo and Juliet, Act I, Scene 1*

the situation where there is no direct grammatical relation between two words, but there exist two separate relations between each of these two words and a third word. Our approach produces a lot of word pairs that are totally unrelated, and therefore leads to unnecessary computation and sometimes incorrect detection of oxymorons. We need a better heuristic for initial selection of candidate contradictory words. Again, if having more false positives than false negatives is more important, then such an approach might be worth considering. We have applied it, for example, to the expression *feather of lead* from Section 3.4, and words *feather* and *lead* were detected as oxymoron. Similarly, when we changed the phrase from *feather of lead* to *leaded feather*, JANTOR also did not make a mistake.

Another type of error we encountered pertains to our second phase—the WordNet-based detection procedure. For a given pair of words we were not always able to reach one word starting from the other using the semantic relations of WordNet. *Serious vanity* or *bright smoke* are examples of such oxymorons. When examining the excerpt from Example 5.23 JANTOR also came up with some false positives. There are at least three reasons why it happens. First, as we mentioned in Section 3.4, we rely to a large extent on the overlap between the words connected to the first word through the semantic relations and the derivative forms of the second word. We do not apply word-disambiguation algorithms, therefore we take into consideration all the possible senses of a given word. Secondly, if we combine this fact with the derivative-form procedure, which is prone to produce false positives, it is inevitable that the precision of the oxymoron-finding algorithm decreases. Nine expressions in our test file were mistakenly treated as oxymorons. Finally, we observed that the semantic relationship between two words is sometimes misleading. For example, applying the following relations to the word *love*, we reached the word *feeling* (derived from *feel*), which should not be the case.

$$\text{love} \xrightarrow[antonymy]{} \text{hate} \xrightarrow[synonymy]{} \text{emotion} \xrightarrow[synonymy]{} \text{feeling}$$

The problem lies in the synonymy relation between the words *hate* and *emotion.* Synonyms, if substituted for one another, should not change the truth value of a sentence in which the substitution is made. This would simply lead us to the conclusion that every emotion is hatred, which is certainly not true.

Finally, at the current stage of development, our system is not able to detect expressions sometimes referred to as "cruel oxymorons". Some of them are: *punk music*, *Microsoft works*, *political co-operation*, etc. These expressions are deeply embedded in certain communities, so that their meaning depends on the context. For example, in the Linux community, the expression *Microsoft works* might be used exactly as an oxymoron whereas among people closely related to the company from Redmond this is an absolutely valid statement. We hope to extend our approach in the future to include both context and community-related information.

5.2 Application: Annotation and Visualization of Presidential Speeches

We applied JANTOR to the annotation of figures of speech in the inaugural addresses of 14 American Presidents. The address, an example of political speech, shares many characteristics with this genre in general. But because an inaugural speech is special in the sense that it is supposed to be the first 'dialogue' between the new president and the citizens of the country, it has to stand out. The communicative function as well as the rhetorical strategies have to be developed and thought through in minute detail in order to successfully have the intended effect on the audience's minds, and to convey an important message to the nation. As Anna Trosborg from the Aarhus School of Business notes[64], there is a huge effort put into the preparation of any inaugural address to create the most communicative, enthralling, and memorable piece of text. Thus the rhetorical features of the address are absolutely salient. Additionally, although any text could be the subject of a rhetorical analysis, the research conducted by rhetoricians has mainly focused on public and professional texts and formal talks such as presidential speeches because they attempt to address real-world issues and create emotional reactions by directing the words to specific audiences who have decision-making power[42].

As a source we have used the archives of The American Presidency Project[4], which contains 86,001 documents related to the study of the Presidency. The speeches we examine in this section were delivered by: Barack Obama (2009), George W. Bush (2005), William Jefferson Clinton (both 1993 and 1997), Ronald Reagan (1981), Jimmy Carter (1977), Richard Milhous Nixon (1969), John Fitzgerald Kennedy (1961), Harry Truman (1949), Franklin Delano Roosevelt (1933), Woodrow Wilson (1913), Theodore Roosevelt (1905), Abraham Lincoln (1861), Thomas Jefferson (1801), and George Washington (1789). Altogether, we analyzed and compared 15 speeches. However, here we present only the most interesting and provocative results.

The organization of this section is as follows. First, we present a statistical analysis of the speeches by comparing the inaugural addresses delivered by Presidents George Washington and Barack Obama. Then we move on the overview of the importance of the placement of rhetorical figures in text. Lastly, we present our initial approach towards the detection of rhetorical patterns among texts written by the same author. At each stage we briefly describe the configuration of JANTOR used for certain analyses.

5.2.1 Intensity—Washington versus Obama

Our first analysis concentrates on the intensity of usage of various rhetorical figures. In this part we do not take into account where exactly in the text these figures occur, but only count the number of occurrences. Obviously, the volume of pragmatic evidence increases proportionally to the length of a speech. Therefore, this examination is meant to give just a general overview of the 'rhetorical maturity' of speakers. The next step towards more precise analysis would be the introduction of a percentage ratio indicating, for example, how many words in a text are parts of some rhetorical figures.

Note:

> The numbers of figures presented in this section are the estimates. Due to various reasons concerning precision of the annotation tool described in the first part of this section the exact determination of the number of figures is dubious. Therefore, we have to manually verify whether the annotations found are correct, and thus provide more accurate analysis.

Let us first consider the inaugural addresses of the current and first presidents, Barack Obama and George Washington, respectively. Table 5.2 presents the number of certain figures of speech in their speeches. We have set the minimum length of a repeating-word sequence to two for anaphora and epistrophe.

Table 5.2: Number of figures in the inaugural addresses

Figure	**Barack Obama**	**George Washington**
anaphora	40	10
epistrophe	1	0
antimetabole	6	1
epanalepsis	0	0
anadiplosis	0	0
polyptoton	10	9
polysyndeton	15	6

As we can observe, Obama used many more anaphoras and polysyndetons in his address. Also, just because the minimum length of a repeating sequence was set to two, the number of his epistrophes is 1. The system correctly identified the following epistrophe:

Our challenges ***may be new****. The instruments with which we meet them* ***may be new****.*

When we changed the minimum length of a sequence to one, JANTOR also found the following, unforgettable, expression:

All this we can ***do****. All this we will* ***do****.*

The lengths of the speeches measured in the number of words were 2406 and 1435 for Obama and Washington respectively (using the Linux "wc" command). Even though the first is roughly 1000 words longer, we can definitely conclude that Obama uses the power of anaphora to a larger extent. He also uses more polysyndetons, therefore the pace of his speech is sometimes slower, but certain fragments are more emphasized. We also observed that the sentences used by Washington were significantly longer on average. Apparently though, the length of a sentence does not force a speaker to excessively use conjunctions. Consider the following example:

Example 5.24:
Time ***and*** *again, these men* ***and*** *women struggled* ***and*** *sacrificed* ***and*** *worked until their hands were raw so that we might live a better life.*

And *we will transform our schools* ***and*** *colleges* ***and*** *universities to meet the demands of a new age.*

One last observation concerns polyptotons—both presidents used words in different forms almost the same number of times. Keeping in mind that Washington's speech is significantly shorter, we might say he 'wins' in this category. However, the individual words in this figure were much further apart than the ones in polyptotons used by Obama. Example 5.25 below presents one of them.

Example 5.25:
The Nation cannot ***prosper*** *long when it favors only the* ***prosperous****. The success*

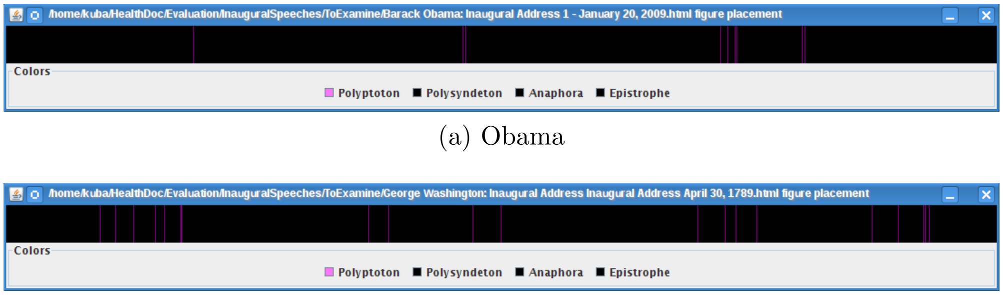
of our economy has always depended not just on the size of our gross domestic product, but on the reach of our ***prosperity****, on our ability to extend opportunity to every willing heart, not out of charity, but because it is the surest route to our common good.*

Even though Obama's speech is considered to be very pragmatic, not poetic[58] and deliberately not flowery[34], rhetorically it seems well-developed. Reiterations allow the audience to thoroughly think over and understand his words. The usage of many stylistic devices helps make his audience truly understand the assertiveness of his language.

5.2.2 Placement

It might seem inaccurate to conclude that one person is more 'rhetorically mature' than another relying only on statistics. Therefore, here we provide the analysis of the placement of figures of speech, which is relative to the length of texts.

First, let us have a look at the two speeches analyzed in the previous discussion. Figure 5.9 present the positioning of polyptotons in both texts.

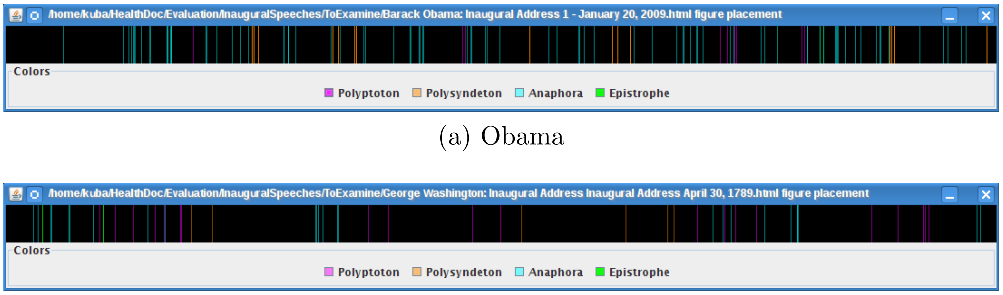

/home/kuba/HealthDoc/Evaluation/InauguralSpeeches/ToExamine/Barack Obama: Inaugural Address 1 - January 20, 2009.html figure placement
Colors
🟪 Polyptoton
■ Polysyndeton
■ Anaphora
■ Epistrophe
/home/kuba/HealthDoc/Evaluation/InauguralSpeeches/ToExamine/George Washington: Inaugural Address Inaugural Address April 30, 1789.html figure placement
Colors
🟪 Polyptoton
■ Polysyndeton
■ Anaphora
■ Epistrophe

(a) Obama

(b) Washington

Figure 5.9: Positioning of polyptoton in Barack Obama's and George Washington's Inaugural Addresses

Our first observation is that some parts of polyptotons in Obama's text are missing. This is because certain words such as *our, ours* were too short with respect to the length of the text to be captured in the visualization. This observation suggests that in future developments of the tool, a 'zoom' facility would be very useful. Figure 5.9 confirms though the observation that Washington's parts of polyptoton seem to be more far apart. Now let us look at all the figures mentioned in Section 5.2.1 (see Figure 5.10).


/home/kuba/HealthDoc/Evaluation/InauguralSpeeches/ToExamine/Barack Obama: Inaugural Address 1 - January 20, 2009.html figure placement
Colors
Polyptoton
Polysyndeton
Anaphora
Epistrophe

(a) Obama

/home/kuba/HealthDoc/Evaluation/InauguralSpeeches/ToExamine/George Washington: Inaugural Address Inaugural Address April 30, 1789.html figure placement
Colors
Polyptoton
Polysyndeton
Anaphora
Epistrophe

(b) Washington

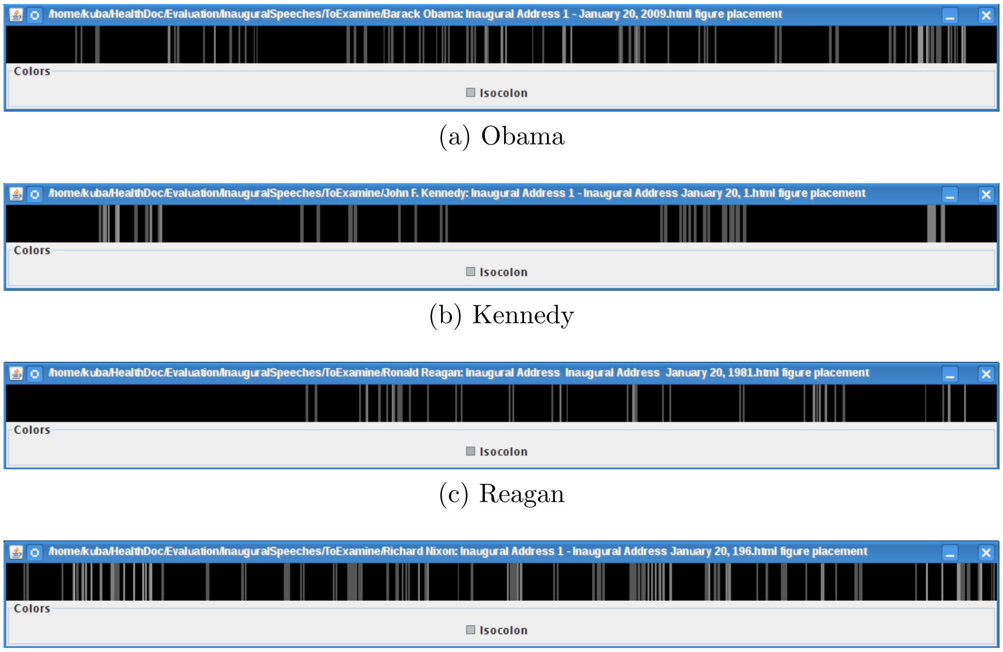
Figure 5.10: Positioning of rhetorical figures in Obama's and Washington's Inaugural Addresses

The immediate conclusion that can be drawn from the examples shown in Figure 5.10 pertains to the distribution of figures. Mainly due to the abundant use of anaphora, the figures in Obama's speech occur more uniformly. When we excluded anaphora from the visualization, the distributions did not vary so much. However, what is noticeable is the difference in the usage of figures in the opening paragraph (left side of both figures). Obama did not use any of the rhetorical devices, whereas Washington seems to have bombarded the audience with rhetorical figures in the commencing sentences. This might suggest different styles the two gentlemen use in their speeches.

Let us now compare the use of isocolons between the addresses of Barack Obama and another accomplished political orator, John Fitzgerald Kennedy. We also include Ronald Reagan and Richard Nixon (Figure 5.11).

We manually deleted some of the falsely discovered isocolons and added those that were missed. The system performed very well in terms of recall (over 80% of isocolons were detected), but the precision can definitely be improved (roughly 50%).

What can be inferred from the above comparison? The examination and visualization of the placement of isocolon lead us to a very provocative and important, but perhaps expected, conclusion. Thoroughly structured texts, good examples of which inaugural addresses usually are, are rhythmic not only within the scope of individual paragraphs, but also throughout the whole speech. In the examples in Figure 5.11 we can easily spot the areas of increased intensity of usage of isocolons. Kennedy's address is exceptional in that sense. Not only these areas are clearly visible, but also repeat in roughly the same intervals. Isocolons are most widely used by Nixon and Obama, and although they appear in the entire text, zones of increased usage are still visible. Although Reagan's speech does not seem so


/home/kuba/HealthDoc/Evaluation/InauguralSpeeches/ToExamine/Barack Obama: Inaugural Address 1 - January 20, 2009.html figure placement
Colors
▨ Isocolon
/home/kuba/HealthDoc/Evaluation/InauguralSpeeches/ToExamine/John F. Kennedy: Inaugural Address 1 - Inaugural Address January 20, 1.html figure placement
Colors
▨ Isocolon
/home/kuba/HealthDoc/Evaluation/InauguralSpeeches/ToExamine/Ronald Reagan: Inaugural Address Inaugural Address January 20, 1981.html figure placement
Colors
▨ Isocolon
/home/kuba/HealthDoc/Evaluation/InauguralSpeeches/ToExamine/Richard Nixon: Inaugural Address 1 - Inaugural Address January 20, 196.html figure placement
Colors
▨ Isocolon

(a) Obama

(b) Kennedy

(c) Reagan

(d) Nixon

Figure 5.11: Comparison of isocolon positioning

structured in terms of the usage of this figure, the same zones can also be spotted in his address. Finally, all four presidents emphasized the end of their speeches by the use of parallelism. Below we present some of the most memorable examples of isocolon in presidential speeches, which were all detected by JANTOR.

Example 5.26:
Our security emanates from ***the justness of our cause****,* ***the force of our example****, the tempering qualities of humility and restraint.*[23]

If a free society ***cannot help the many who are poor****,* ***it cannot save the few who are rich****. (. . .) And so, my fellow Americans: ask not what your country can do for you—ask what you can do for your country.*[24]

Government can and must ***provide opportunity, not smother it; foster productivity, not stifle it****.*[25]

[23] Barack Obama

[24] John Kennedy

[25] Ronald Reagan

***We are caught in war, wanting peace**. **We are torn by division, wanting unity**.*[26]

Trosborg[64] notes that parallel structures and balanced constructions are also typical features of other inaugural speeches (e.g., Bush's from 1998). The above analysis confirms that parallelism is an intrinsic rhetorical characteristic of many presidential addresses. It sets the rhythm of the speech and also makes the words memorable.

The Rule of Three

"The Rule of Three"[11], which is beloved by rhetoricians, very often appears in the form of isocolons. The rule itself comes from ancient times and is generally considered as a mathematical law of proportion. As far as rhetoric is concerned, the number three is truly magical because people usually are more inclined to memorize something if it is said three different times[52]. Below we present some of the detected examples of usage of this rule in the form of an isocolon.

Example 5.27:
*The time has come **to reaffirm our enduring spirit**, **to choose our better history**, **to carry forward that precious gift**, that noble idea passed on from generation to generation.*[27]

*. . . the belief that **the rights of man** come not from **the generosity of the state** but from **the hand of God**. . .*[28]

*It is time **to reawaken this industrial giant**, **to get government back within its means**, and **to lighten our punitive tax burden**.*[29]

*Let us take as our goal: **Where peace is unknown, make it welcome**; **where peace is fragile, make it strong**; **where peace is temporary, make it permanent**.*[30]

[26] Richard Nixon

[27] Barack Obama

[28] John Kennedy

[29] Ronald Reagan

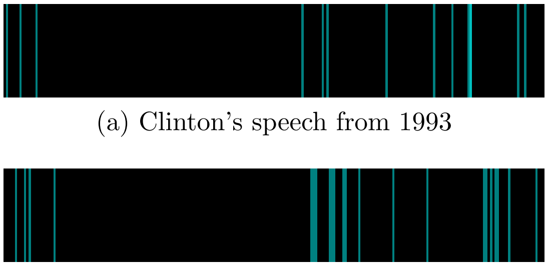
[30] Richard Nixon

5.2.3 Rhetorical Evolution?

The final stage in our study of presidential speeches was to search for any rhetorical trends within the inaugural speeches of the same president. For this purpose we analyzed the two addressed delivered by President Bill Clinton. The figures we included in this comparison were: anadiplosis, anaphora, antimetabole, epanalepsis, epistrophe, epizeuxis, polyptoton, and polysyndeton. Below we describe several regularities we were able to observe.

First, the figures of epanalepsis, epizeuxis, and anadiplosis were very rare in both speeches. Due to the small number of occurrences of these figures, it is not possible to draw any definite conclusions. Thus we will focus on more numerous figures in this analysis. We will begin anaphora.

The first characteristic we may observe is the absence followed by the accumulation of this figure at the end of both speeches. This observation is shown in Figure 5.12.



(a) Clinton's speech from 1993

(b) Clinton's speech from 1997

Figure 5.12: Anaphora comparison at the end of the inaugural addresses by Bill Clinton

It is well-established in rhetoric that the closing paragraph of a speech plays an important role in affecting the audience. In most of the addresses we examined, the speakers emphasized the concluding sentences, and the two Clinton speeches are good examples of this. The blue bars on the furthest right side indicate anaphora at the end of the articles. The sentences preceding the closing paragraph (black space before the blue bars) are not really filled with many figures (anaphora in this case), which accentuates the final words even more.

As Trosborg observes[64], the typical rhetorical feature of Clinton's 1993 speech is *iconic linkage*. What she is referring to is a repetitive pattern that attracts attention and joins the parts of the text. This linkage, established by extensive use of anaphora, is not only very characteristic in Clinton's first speech, but in his second as well, and in many inaugural addresses in general.

In terms of the other remaining figures, the 1993 Clinton speech seems to be more rhetorically developed. First of all, antimetaboles, polysyndetons and polyptotons tend to form groups, mostly in the first half of the speech, which sets up a certain rhythm in this part of the speech. We illustrate this situation in Figure 5.13—the figure groupings are indicated by red rectangles.

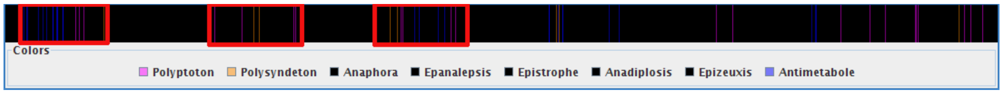


Figure 5.13: Polyptoton, polysyndeton, and antimetabole placement in Clinton's first inaugural address

Trosborg[64] also highlights the dominant cohesive features of Clinton's speech. These features concern the notions of *change* and *renewal*. These words in many variations occur quite often in the text, and are depicted by many purple bars (polyptoton) in Figure 5.13. Polyptotons are also present in Clinton's 1997 address, but have a different nature. Only once does Clinton use *new, renew*, and what is very interesting, he then refers to his previous (1993) inauguration speech.

Overall, we can conclude that, although different in nature, rhetorically both speeches do not vary that much. The words used in individual figures of speech are different, but in general both addresses set a good rhetorical standard.

5.2.4 General Observations

Our analysis of a subset of U.S. presidential inaugural addresses leads us to several general conclusions. The most frequently used figures of speech are anaphora, isocolon, polyptoton, and polysyndeton. Anaphora is used not only to enumerate items, but also to bind the parts of the text together. This use of anaphora helps create a fluent flow of ideas and, in general, a more cohesive speech. Secondly, the patterns of isocolon establish a good rhythm in the texts and makes them easier to remember, partially by applying the aforementioned rule of three. Finally, the significant number of polysyndetons is used to stress specific fragments by periodically occurring throughout the text.

Chapter 6

Conclusions and Future Work

In this work, we created an annotation tool that enables both the manual and automated markup of rhetorical figures. To our knowledge (based on Google Scholar[29]) this is the first implemented system for the automated annotation of figures of speech. The tool, JANTOR, in its current stage of development supports the detection of 11 figures of speech. Apart from the figures of repetition, more specifically the repetition of words, the annotation tool enables the detection of a figure of parallelism, isocolon, and also a figure of semantic contradiction between words, oxymoron. Our results for precision and recall are very promising as far as figures of repetition are concerned. The satisfactory results for the detection of the other forms of rhetorical devices suggest that improvement is necessary, but also provide helpful observations and indicate possible directions for future research.

This first step towards the automated annotation of figures of speech can be used to characterize and classify rhetorical patterning. Tasks might involve the recognition of rhetorical strategies such as persuasion and argumentation[32], or improving communication and creating more appealing texts. The system we have created combines the ancient theory of rhetoric with modern computational linguistic technology to facilitate more efficient and accurate identification of pragmatic evidence in natural language texts. In general, the automated annotation of figures provides a computationally efficient method of pragmatic analysis which can help in understanding how to speak and write effectively, compose messages in the most informative way, and reach audiences in the most appealing way.

6.1 Automated Annotation

There were many issues related to rhetorical-figure annotation that influenced this thesis. First of all, it was crucial to capture different cases of intra- and inter-figure positioning of the individual annotations. Different genres of text contain different figures, and the intensity also varies. Very often they overlap with each other or occur in the same place. Additionally, almost all the figures consist of many parts. JANTOR supports all of the aforementioned cases.

Secondly, we believe that it is crucial to store the annotations separately from the object of annotation itself. It enables multiple markup of the same text without human annotators having to create many instances of the same documents. For this purpose we have created the stand-off schema to handle the nuances of rhetorical-figures annotation.

We identified many problems pertaining to the annotation of rhetorical figures, some of which we have presented a solution for, but some still remain unsolved. The first set of problems concerns the identification of syntactic units in text on many different levels. In Section 3.1 we have presented our approach for the detection of sentences, phrases, and clauses. The identification of boundaries of these syntactic units is absolutely crucial for the correct detection of figures of repetition. Our solution using *BreakIterator* together with a lexicalized probabilistic context-free grammar parser performed very well on the examples we examined. However, more advanced solutions are needed. In Chapter 5 we showed numerous cases when the parsing of sentences was not correct, which often resulted in missed or wrongly classified rhetorical figures. We tried to alleviate these problems not by trying to fix erroneous parses, but by finding solutions that satisfied the constraints imposed by the definitions of individual figures. For example, we add the phrases between punctuation markers to whatever comes out of the parser and then operate on this extended set of phrases. However, as we mentioned in Chapter 5, a better heuristic for phrase boundary detection is definitely needed. Generally, punctuation plays a significant role in the analysis of discourse structure[18, 19, 35], but other markers such as lexical, including cue words, or graphical, including the use of paragraphs, might also significantly improve the detection of rhetorical figures. As far as clause identification is concerned we hope to extend our approach to one based on conditional random fields[39] or on specialized Hidden Markov Models presented by Molina *et. al* in[49].

The observations we made in Chapter 5 also led us to the following conclusions concerning the detection of individual figures. For isocolon, we should probably also take into consideration the number of syllables. So far we have looked at the number of words, parse structures, and the number of overlapping part-of-speech tags between phrases. This strategy enabled us to obtain satisfactory recall results, but the values for precision definitely have to be improved. We believe that imposing some constraints that relate to small syntactic units, like syllables, and that are actually mentioned in the precise definitions of isocolon might improve the precision for detection of this figure and still keep the recall at the current level.

Other figures we definitely plan to investigate further are polyptoton and oxymoron. For polyptoton, an approach producing less false positives of the word forms has to be devised. The use of both the Porter stemmer and WordNet "derived forms" option (see Algorithm 3.3) performs reasonably well on many words, but sometimes produces words totally unrelated to the original one. Additionally, cases when the stem of a word varies depending on the part of speech are also not handled yet. We feel that measuring semantic relatedness between words using WordNet might be very useful for completing this task[9, 13, 53].

Finally, there is much to be done in the detection of *tropes*. In this thesis we have shown a way for detecting oxymorons. The major problems we encountered pertained to: (a) selection of candidate words contradictory in meaning; and (b) determination whether these words are actually in semantic opposition. We have addressed the first problem by making use of the grammatical relations of typed dependencies between words stems, and this method seems to be the right one to follow. One of the possible extensions would be using some sort of database of *subcategorization frames*[24]. A subcategorization frame is a set of arguments with which a particular verb can appear[45]. Subcategorization frames capture syntactic regularities about complements of verbs, therefore a comprehensive collection of such frames might significantly improve candidate-words selection for the examination of the meaning contradiction in oxymoron.

The second issue definitely requires more attention. In order to improve the detection procedure of not only oxymorons, but semantics-related figures in general, word sense disambiguation techniques have to be applied. Our current approach did not distinguish between senses of words, which is one of the causes of low precision. There exists a substantial amount of research on word sense disambiguation using WordNet[15, 65, 40]. We also believe that extending our current search by the *hypernymy* and *hyponymy* relations may prove helpful. Additionally, oxymoron is an example of a figure that sometimes relies on the contextual information specific for a community in which it is used. Incorporating this community-related knowledge is crucial, especially for capturing figures that only occur in certain conditions.

There are additional problems that we observed. First, the range within which we look for figures of repetition significantly impacts the precision of the detection.

The identification procedure for polysyndeton, for example, should usually be performed within one, or at most two sentences. This number varies though according to the figure. The same words sometimes begin successive paragraphs, each having five or more sentences, and they are still considered occurrences of anaphora. On the other hand, sentences can be 15 or more words in length, so that then even three words can constitute a large chunk of text, in which word reoccurrences might not have a strong effect. More sophisticated measures of distance between repeating words will have to developed in future.

Finally, even the manual annotation of rhetorical figures cannot always be definite and depends on many factors, some of which we have mentioned above. Therefore, the certainty dependent on the aforementioned and other factors has to be assigned by the automated detection system to the identified pragmatic evidence.

6.2 Analysis

The analysis presented in Section 5.2 provides some evidence that the automated analysis of meaningful rhetorical information in real-life texts is possible and tractable. The visualization module of JANTOR run on the set of presidential speeches indicates that the extensive use of rhetorical figures makes inaugural addresses seem both outstanding and communicative. According to Trosborg, the communicative purpose and the usage of many rhetorical features set off these figures among other political texts[64]. We hope to perform an appropriate experimental analysis in our future work to confirm this hypothesis.

6.3 Future Work

There are several possible directions in which this work could be developed. First, the scope of the rhetorical-figure annotations supported by JANTOR should be extended. There are different kinds of repetitions that need to be handled—letters, syllables, sounds (alliteration, assonance, consonance, etc.), and ideas (commoratio, disjunctio, palilogia, and many others). Additionally, we hope that the improved approach used for the detection of oxymoron could be extended to *antithesis*—the juxtaposition of contrasting words or ideas (often, although not always, in parallel structure)[1]. A new strategy for selecting candidate words or phrases that could possibly constitute this figure would have to worked out. SentiWordNet[25]—the lexical resource for opinion mining—could be applied. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity. If we could capture the general sentiment of two phrases, one of which would be negative and the other would be positive, and locate contradictory word pairs in them, we could possibly spot some antitheses. Additionally, the substantial research conducted by Marneffe *et al.*[22] on the detection of contradictions in text might be extremely helpful.

We believe that our system could also be used for the identification of genres depending on the type, number, and position of rhetorical figures used. This could provide additional dimensions to well-established fields in machine learning—text clustering and classification.

While the analysis presented in Section 5.2.3 is not conclusive, it provides some evidence that the discovery of rhetorical trends is possible. This is another direction of research that should be considered in the future. It would be extremely helpful, provocative, and interesting to see whether a speaker 'develops' rhetorically and oratorically over time.

There are two broad problems we would like to address in the future. First, we hope to create a corpus or database of annotated rhetorical figures. To our knowledge, a collection of annotated figures of speech for any kind of texts does not exist. Creating such corpora would be extremely valuable for various reasons. For example, it would facilitate the creation of an ontology of rhetorical figures for use in Natural Language systems[32]. As another potential use, from the linguistic research viewpoint, a queryable database of pragmatic evidence, with connections to the authors and source documents, could provide a significant improvement to the rhetorical analyses of specific writers and help with tailoring texts to specific audiences.

Finally, we hope to create a comprehensive collection of word forms. For this purpose more advanced algorithms than the one we presented in polyptoton part of Section 3.2.2 have to be designed and implemented. Such a database would definitely be valuable to many text-related tasks in natural language processing, including information retrieval and information extraction.

6.3.1 Coach Notes Annotation

Our tool can be extended to enable various kinds of annotation. Currently, the scope of JANTOR's annotation has been broadened to handle the annotation of a rhetorical model of health coach-patient interaction. The markup of seven different 'dimensions' was made possible: social/situational, interaction/interactive mode, emotional tone, agency, accountability/responsibility, cost, and chronos. The values for each dimension describe a spectrum of different strategies that a coach may use during the interaction with a patient[6].

APPENDICES

Appendix A

Glossary

anadiplosis the repetition of the last word (or phrase) from the previous line, clause, or sentence at the beginning of the next

anaphora repetition of the same word or group of words at the beginning of successive clauses, sentences, or lines

antimetabole repetition of words, in successive clauses, in reverse grammatical order

epanalepsis repetition at the end of a line, phrase, or clause of the word or words that occurred at the beginning of the same line, phrase, or clause

epistrophe ending a series of lines, phrases, clauses, or sentences with the same word or words

epizeuxis repetition of words with no others between, for vehemence or emphasis

isocolon a series of similarly structured elements having the same length

oxymoron the yoking of two terms that are ordinarily contradictory

ploche the repetition of a single word for rhetorical emphasis

polyptoton repeating a word, but in a different form; using a cognate of a given word in close proximity

polysyndeton employing many conjunctions between clauses, often slowing the tempo or rhythm

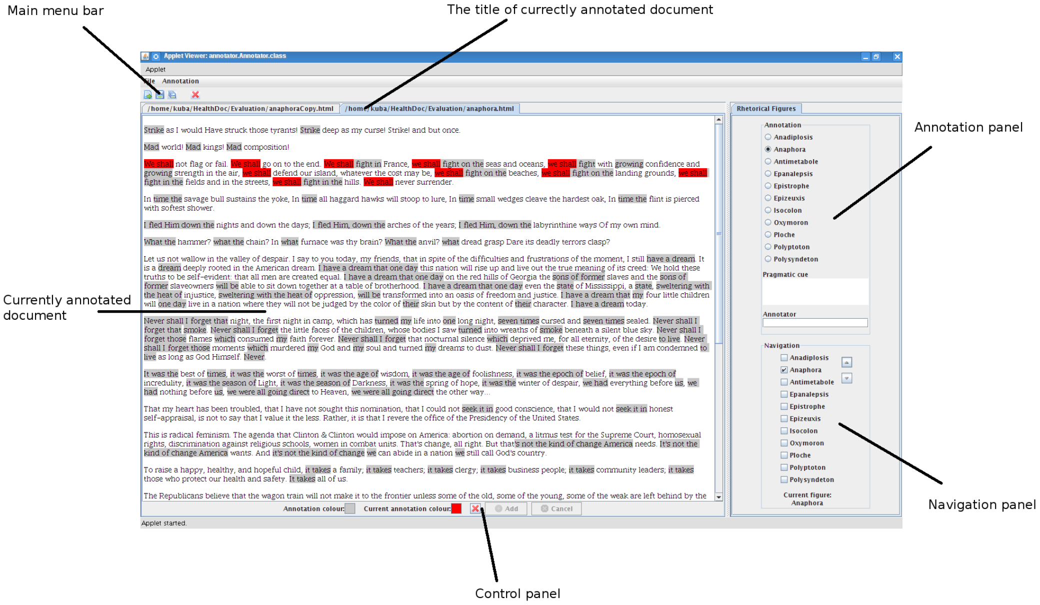
Appendix B

JANTOR - Step by Step

The purpose of this appendix is to guide the user through the options and features of JANTOR. First, we describe the general inferface of the tool.

B.1 Graphical Components

The main components: *Annotation Panel*, *Navigation Panel* and *Control Panel* are described in details later in this section. After a user selects the file for annotation, by either choosing a HTML file to annotate from scratch or a XML file with already annotated content, the main window of JANTOR is presented to them (see Figure B.1). The left-hand side of the window — the *Document View* — contains the text of the document being annotated together with the marked figures. On the right-hand side the *Annotation* and *Navigation* panels are situated. At the bottom one can find the *Control Panel*. We overview each of these components in more detail below.



Strike as I would Have struck those tyrants! Strike deep as my curse! Strike! and but once.

Mad world! Mad kings! Mad composition!

We shall not flag or fail. We shall go on to the end. We shall fight in France, we shall fight on the seas and oceans, we shall fight with growing confidence and growing strength in the air, we shall defend our island, whatever the cost may be, we shall fight on the beaches, we shall fight on the landing grounds, we shall fight in the fields and in the streets, we shall fight in the hills. We shall never surrender.

In time the savage bull sustains the yoke, In time all haggard hawks will stoop to lure, In time small wedges cleave the hardest oak, In time the flint is pierced with softest shower.

I fled Him down the nights and down the days; I fled Him, down the arches of the years; I fled Him, down the labyrinthine ways Of my own mind.

What the hammer? what the chain? In what furnace was thy brain? What the anvil? what dread grasp Dare its deadly terrors clasp?

Let us not wallow in the valley of despair. I say to you today, my friends, that in spite of the difficulties and frustrations of the moment, I still have a dream. It is a dream deeply rooted in the American dream. I have a dream that one day this nation will rise up and live out the true meaning of its creed: We hold these truths to be self-evident: that all men are created equal. I have a dream that one day on the red hills of Georgia the sons of former slaves and the sons of former slaveowners will be able to sit down together at a table of brotherhood. I have a dream that one day even the state of Mississippi, a state, sweltering with the heat of injustice, sweltering with the heat of oppression, will be transformed into an oasis of freedom and justice. I have a dream that my four little children will one day live in a nation where they will not be judged by the color of their skin but by the content of their character. I have a dream today.

Never shall I forget that night, the first night in camp, which has turned my life into one long night, seven times cursed and seven times sealed. Never shall I forget that smoke. Never shall I forget the little faces of the children, whose bodies I saw turned into wreaths of smoke beneath a silent blue sky. Never shall I forget those flames which consumed my faith forever. Never shall I forget that nocturnal silence which deprived me, for all eternity, of the desire to live. Never shall I forget those moments which murdered my God and my soul and turned my dreams to dust. Never shall I forget these things, even if I am condemned to live as long as God Himself. Never.

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way...

That my heart has been troubled, that I have not sought this nomination, that I could not seek it in good conscience, that I would not seek it in honest self-appraisal, is not to say that I value it the less. Rather, it is that I revere the office of the Presidency of the United States.

This is radical feminism. The agenda that Clinton & Clinton would impose on America: abortion on demand, a litmus test for the Supreme Court, homosexual rights, discrimination against religious schools, women in combat units. That's change, all right. But that's not the kind of change America needs. It's not the kind of change America wants. And it's not the kind of change we can abide in a nation we still call God's country.

To raise a happy, healthy, and hopeful child, it takes a family; it takes teachers; it takes clergy; it takes business people; it takes community leaders; it takes those who protect our health and safety. It takes all of us.

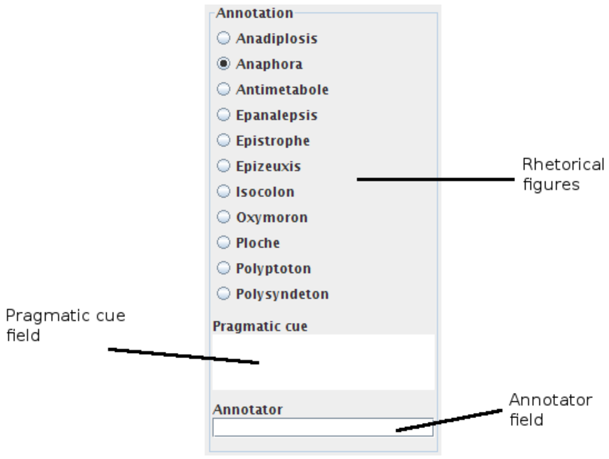
The Republicans believe that the wagon train will not make it to the frontier unless some of the old, some of the young, some of the weak are left behind by the

Figure B.1: JANTOR - Java ANnotation Tool Of Rhetoric

The *Annotation Panel* is used for a) choosing the type of currently selected figure, b) adding pragmatic cue to the figure, and c) providing the name (or any other identification) of the annotator. The multiple assignment of types of figures to one annotation is not allowed. Therefore, a user can select one of the radio buttons corresponding to figures of speech at a time. However, at any time of the annotation process they can change the type of figure. The Annotation Panel is presented in Figure B.2. The type of currently selected figure is *anaphora*.

The next component of JANTOR is the *Navigation Panel*. This panel enables you to select the type of figures that should be displayed. You can display many different figures at a time. Navigation Panel contains two more components — navigation buttons and current figure indicator. The former one is used to move up and down (backward and forward with respect to the position in the text) through the annotated figures. The latter indicates the currently selected figure type. The current figure is marked in red in the *Document View*, whereas all the other annotations are highlighted in grey.

The next component is the *Control Panel* (see Figure B.4). Once the parts of the text have been marked in *Document View*, you can either select the type of

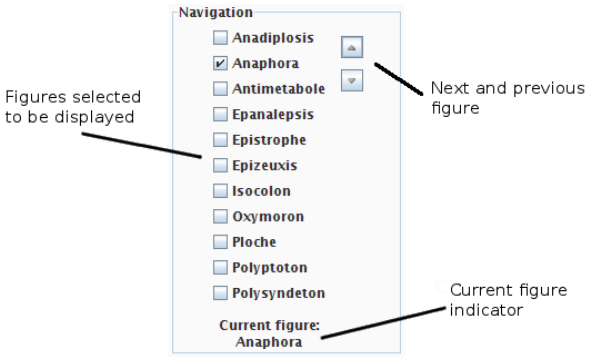

Annotation

○ Anadiplosis
● Anaphora
○ Antimetabole
○ Epanalepsis
○ Epistrophe
○ Epizeuxis
○ Isocolon
○ Oxymoron
○ Ploche
○ Polyptoton
○ Polysyndeton

Pragmatic cue

Annotator

Figure B.2: Annotation panel


Navigation

□ Anadiplosis
☑ Anaphora
□ Antimetabole
□ Epanalepsis
□ Epistrophe
□ Epizeuxis
□ Isocolon
□ Oxymoron
□ Ploche
□ Polyptoton
□ Polysyndeton

Current figure:
Anaphora

Figure B.3: Navigation panel

figure for the current annotation by clicking the *Add* button. The panel for selection of the figure type is presented in Figure B.5; or abort the current annotation by clicking the *Cancel* button. You can also delete currently selected figure by clicking the red 'X' button.

B.2 Manual Annotation

This part shortly describes the steps needed for manual annotation of text.

1. Select parts of text that should constitute one figure

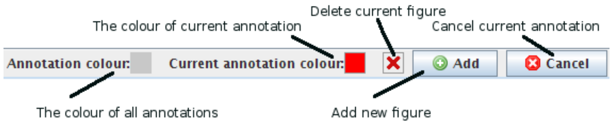
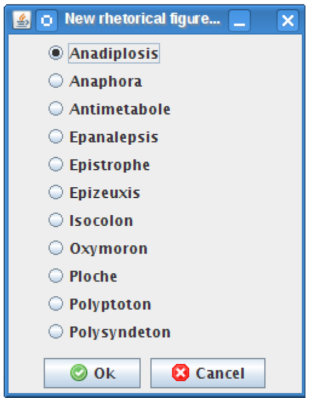

Delete current figure
The colour of current annotation
Cancel current annotation
Annotation colour:
Current annotation colour:
Add
Cancel
The colour of all annotations
Add new figure

Figure B.4: Control panel



● Anadiplosis
○ Anaphora
○ Antimetabole
○ Epanalepsis
○ Epistrophe
○ Epizeuxis
○ Isocolon
○ Oxymoron
○ Ploche
○ Polyptoton
○ Polysyndeton

Figure B.5: New figure selection panel

2. Click *Add* in the *Control Panel*

3. Select the type of figure that should assigned to the annotations

Once the figure has been added, you can put your name in the *Annotator* field; change the type of the figure by selecting appropriate radio button in the *Annotation Panel*; or add the pragmatic cue. In order to save the annotated figures click *File→Save/Save As* or press corresponding button in the *Main Menu* toolbar.

B.3 Automated Annotation

In order to automatically detect certain type of figures you have to click *Figure→Detect...*. The figure type selection dialog window will pop up. The first run of the detection takes some time, as all the sentences have to be parsed and the word repetitions have to be founds. Once completed, all the successive runs take much shorter amount of time. The exception is the detection of polyptoton, which calls the word form finder module. However, this module is also required only once per document, as the different forms of the 'same' word are kept in memory. When the identification is finished, the figures found are displayed in the *Document View* window.

B.4 Visualization

Finally, to visualize the annotated figures, one has to select *Figure*→*Visualize* option from the main menu. All the figures currently selected for display in the Navigation Panel will be visualized. One of the neat options of the visualization module is the possibility to turn on and off any type of figures. Thus, one can select all of them to be shown and then, if the picture appears too cluttered, they can choose which should actually be displayed.

B.5 Configuration

The only configuration supported at the current stage is the specification of oxymoron relation paths, described in Section 3.4. Below we present a sample **oxymoron.config** file.

```
[Antonym]
[Antonym, Synonym]
[Synonym, Antonym]
[Antonym, Derivation]
[Derivation, Antonym]
[Antonym, Synonym, Derivation]
[Antonym, Derivation, Synonym]
[Synonym, Derivation, Antonym]
#[Synonym, Antonym, Synonym, Derivation]
#[Hyponym, Derivation, Antonym]
#[Hyponym, Derivation, Synonym, Antonym]
```

JANTOR reads all the lines not starting with '#' and creates corresponding relations paths used for the WordNet-based detection of oxymoron.

B.6 Annotation File

Example annotation file generated by JANTOR indicating anaphoras is presented below.

```
<xml:XMI>
    <rhe:Annotator xmi:id="1" name=""/>
    <rhe:Document xmi:id="2" sha1="6ced665a65683fbfe73db4a81a51a134685d18ab"
     sofaUri="/home/kuba/HealthDoc/Evaluation/anaphora.html"/>
    <rhe:Figure annotator="1" xmi:id="3" sofa="74,75,76" type="Anaphora"/>
    <rhe:Figure annotator="1" xmi:id="4" sofa="77,78,79" type="Anaphora"/>
    <rhe:Range beginChar="95" endChar="101" xmi:id="74" sofaFeature="text"
                                            sofaObject="2" surface="Strike"/>
    <rhe:Range beginChar="140" endChar="146" xmi:id="75" sofaFeature="text"
                                             sofaObject="2" surface="Strike"/>
    <rhe:Range beginChar="165" endChar="171" xmi:id="76" sofaFeature="text"
                                             sofaObject="2" surface="Strike"/>
    <rhe:Range beginChar="214" endChar="217" xmi:id="77" sofaFeature="text"
                                             sofaObject="2" surface="Mad"/>
    <rhe:Range beginChar="225" endChar="228" xmi:id="78" sofaFeature="text"
                                             sofaObject="2" surface="Mad"/>
    <rhe:Range beginChar="236" endChar="239" xmi:id="79" sofaFeature="text"
                                             sofaObject="2" surface="Mad"/>
</xml:XMI>
```

Appendix C

Penn Treebank Project Tag Set

Table C.1: The Penn Treebank POS tagset[46]

Tag	Description	Tag	Description
CC	Coordinating conjunction	TO	to
CD	Cardinal number	UH	Interjection
DT	Determiner	VB	Verb, base form
EX	Existential there	VBD	Verb, past tense
FW	Foreign word	VBG	Verb, gerund/present participle
IN	Preposition/subordinating conjunction	VBN	Verb, past participle
JJ	Adjective	VBP	Verb, non-3rd ps. sing. present
JJR	Adjective, comparative	VBZ	Verb, 3rd ps. sing. present
JJS	Adjective, superlative	WDT	wh-determiner
LS	List item marker	WP	wh-pronoun
MD	Modal	WP$	Possessive wh-pronoun
NN	Noun, singular or mass	WRB	wh-adverb
NNS	Noun, plural	#	Pound sign
NNP	Proper noun, singular	$	Dollar sign
NNPS	Proper noun, plural	.	Sentence-final punctuation
PDT	Predeterminer	,	Comma
POS	Possessive ending	:	Colon, semi-colon
PRP	Personal pronoun	(	Left bracket character
PP$	Possessive pronoun	)	Right bracket character
RB	Adverb	"	Straight double quote
RBR	Adverb, comparative	‘	Left open single quote
RBS	Adverb, superlative	“	Left open double quote
RP	Particle	’	Right close single quote
SYM	Symbol (mathematical or scientific)	”	Right close double quote

References

[1] Silva Rhetoricae website, http://humanities.byu.edu/rhetoric/Silva.htm. 4, 16, 19, 20, 21, 22, 23, 24, 25, 26, 32, 37, 54, 87

[2] Sun Java 6 Platform website, http://java.sun.com/javase/6. 13, 45

[3] Stanford NLP Group website, http://nlp.stanford.edu. 14

[4] The American Presidency Project website, http://www.presidency.ucsb.edu. 75

[5] A. Nierman, H. V. Jagadish. Evaluating structural similarity in XML documents. In *Proceedings of the Fifth International Workshop on the Web and Databases (WebDB 2002)*, Madison, Wisconsin, USA, June 2002. 35

[6] N. Abbot, A. Kelly, and A. McDougall. Rhetorical Appraisal Model. Technical report, University of Waterloo, 2009. 89

[7] About.com on grammar and composition, http://grammar.about.com. 54

[8] American Rhetoric web-site, http://www.americanrhetoric.com/. 54

[9] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810, 2003. 70, 86

[10] S. Bird and M. Liberman. A formal framework for linguistic annotation. *Speech Communication*, 33(1-2):23–60, 2001. 5

[11] C. Booker. *The Seven Basic Plots: Why We Tell Stories*. Continuum International Publishing Group, 2004. 81

[12] J. Boswell. *Boswell's Life of Johnson*. Signet-New American Library, 1968. 25

[13] A. Budanitsky. Semantic Distance in WordNet: An Experimental, Application-oriented Evaluation of Five Measures. In *Proceedings of the NACCL 2001 Workshop: on WordNet and other lexical resources: Applications, extensions, and customizations*, pages 29–34, 2001. 86

[14] C. Fellbaum, editor. *WordNet: An Electronic Lexical database.* MIT Press, 1998. 27

[15] A. J. Cañas, A. Valerio, J. Lalinde-Pulido, M. M. Carvalho, and M. Arguedas. Using WordNet for Word Sense Disambiguation to Support Concept Map Construction. In *SPIRE*, pages 350–359, 2003. 86

[16] E. P. J. Corbett. *Classical Rhetoric for the Modern Student.* New York: Oxford University Press, 1990. 1, 2, 3, 22, 23, 31, 32, 37, 54, 68

[17] E. P. J. Corbett and R. J. Connors. *Classical Rhetoric for the Modern Student.* New York: Oxford University Press, 1999. 24

[18] R. Dale. Exploring the role of punctuation in the signalling of discourse structure. In *Proceedings of a Workshop on Text Representation and Domain Modelling: Ideas from Linguistics and AI*, pages 110–120. Technical University Berlin, 1991. 85

[19] R. Dale. The Role of Punctuation in Discourse Structure. In *Working Notes for the AAAI Fall Symposium on Discourse Structure in Natural Language Understanding and Generation*, Menlo Park, California, 1991. AAAI, American Association for Artificial Intelligence. 85

[20] Daniel Marcu's modification of RSTTool http://www.isi.edu/licensed-sw/RSTTool/. 8

[21] M.-C. de Marneffe, B. MacCartney, and C. Manning. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of LREC-06*, pages 449–454, 2006. 38

[22] M.-C. de Marneffe, A. Rafferty, and C. D. Manning. Finding contradictions in text. 11, 42, 88

[23] Encyclopedia Britannica web site on polyptoton, http://www.britannica.com/EBchecked/topic/469083/polyptoton. 27

[24] M. Erdmann, A. Maedche, H. P. Schnurr, and S. Staab. From Manual to Semi-automatic Semantic Annotation: About Ontology-Based Text Annotation Tools. In *Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content*, 2000. 10, 86

[25] A. Esuli and F. Sebastiani. SENTIWORDNET: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06)*, pages 417–422, 2006. 87

[26] D. Ferrucci, A. Lally, D. Gruhl, E. Epstein, M. Schor, J. W. Murdock, A. Frenkiel, E. W. Brown, T. Hampp, Y. Doganata, C. Welty, L. Amini, G. Kofman, L. Kozakov, and Y. Mass. Towards an Interoperability Standard for Text and Multi-Modal Analytics. Technical report, IBM Research, 2006. 46

[27] M. Ford, J. Bresnan, and R. M. Kaplan. A Competence-Based Theory of Syntactic Closure. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 727–796. MIT Press, Cambridge, MA, 1982. 14

[28] G. Xing, J. Guo, Z. Xia. Classifying XML Documents Based on Structure/Content Similarity. In *INEX*, pages 444–457, 2006. 35

[29] Google Scholar, http://scholar.google.com. 7, 84

[30] B. J. Grosz and C. L. Sidner. ATTENTION, INTENTIONS, AND THE STRUCTURE OF DISCOURSE, 1986. 8

[31] H. De Meyer, B. De Baets and S. Janssens. Similarity measurement on leaf-labelled trees. 2001. 35

[32] R. Harris and C. DiMarco. Constructing a Rhetorical Figuration Ontology. In *Symposium on Persuasive Technology and Digital Behaviour Intervention (AISB 2009)*. 10, 18, 44, 84, 88

[33] D. Hindle and M. Rooth. Structural ambiguity and lexical relations. In *HLT '90: Proceedings of the workshop on Speech and Natural Language*, pages 257–262, Morristown, NJ, USA, 1990. Association for Computational Linguistics. 14

[34] Jeff Greenfield video on President Obama's Inaugural Address, http://www.cbsnews.com/video/watch/?id=4738702n. 78

[35] B. E. M. Jones. Exploring the role of punctuation in parsing natural text. In *Proceedings of the 15th conference on Computational linguistics*, pages 421–425, Morristown, NJ, USA, 1994. Association for Computational Linguistics. 85

[36] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000. 14

[37] D. Klein and C. D. Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems*, pages 3–10. MIT Press, 2002. 14

[38] D. Klein and C. D. Manning. Accurate Unlexicalized Parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA, 2003. Association for Computational Linguistics. 14

[39] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. 85

[40] X. Li, S. Szpakowicz, and S. Matwin. A WordNet-based Algorithm for Word Sense Disambiguation. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1368–1374, 1995. 86

[41] LinkaLinks tools web-site, http://www.sil.org/lingualinks/LingTool.html. 10

[42] J. L. Lucaites, C. M. Condit, and S. Caudill. *Contermporary Rhetorical Theory: A Reader*. Guilford Press, 1998. 75

[43] Maite Taboada and William C. Mann. Applications of Rhetorical Structure Theory. *Discourse Studies*, 8(4):567–588, 2006. 7

[44] W. Mann and S. Thompson. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 3, 1988. 7

[45] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, sixth edition, 2003. 86

[46] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993. ix, 33, 100

[47] E. F. McQuarrie and D. G. Mick. Figures of Rhetoric in Advertising Language. *Journal of Consumer Research: An Interdisciplinary Quarterly*, 22(4):424–38, March 1996. 3, 4

[48] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–244, January 1990. 27, 40

[49] A. Molina and F. Pla. Clause detection using HMM. In *ConLL '01: Proceedings of the 2001 workshop on Computational Natural Language Learning*, page 1, Morristown, NJ, USA, 2001. Association for Computational Linguistics. 85

[50] M. O'Donnell. RSTTool 2.4: a markup tool for Rhetorical Structure Theory. In *INLG '00: Proceedings of the first international conference on Natural language generation*, pages 253–256, Morristown, NJ, USA, 2000. Association for Computational Linguistics. 8

[51] Official web-site for the Lancaster (Paice/Husk) stemming algorithm, http://www.comp.lancs.ac.uk/computing/research/stemming/index.htm. 29

[52] Patricia Fripp, The Rule of Three, http://www.fripp.com/art.rulethree.html. 81

[53] T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet::Similarity - Measuring the Relatedness of Concepts. In *AAAI*, pages 1024–1025, 2004. 86

[54] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980. 28

[55] R. Nayak, T. Tran. A Progressive Clustering Algorithm to Group the XML Data by Structural and Semantic Similarity. *International Journal of Pattern Recognition and Artificial Intelligence*, 2006. 35

[56] R. Radoulov. Exploring Automatic Citation Classification. Master's thesis, Universtiy of Waterloo, 2008. 12

[57] Rhetorikos, Henry George Liddell, Robert Scott, *A Greek-English Lexicon*, at Perseus. 1

[58] F. Rich. No Time for Poetry. *NY Times*, January 2009. http://www.nytimes.com/2009/01/25/opinion/25rich.html?_r=1&ref=opinion. 78

[59] A. R. Smith. Image compositing fundamentals. Microsoft Technical Memo 4, August 1995. 52

[60] C. R. Smith. *Rhetoric and Human Consciousness: A History.* Waveland Press, Inc, 2003. 1

[61] M. Stede and S. Heintze. Machine-assisted rhetorical structure annotation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 425, Morristown, NJ, USA, 2004. Association for Computational Linguistics. 9

[62] M. Taboada and W. C. Mann. Rhetorical Structure Theory: looking back and moving ahead. *Discourse Studies*, 8:423–459, 2006. 7, 8

[63] Thomas De Simone and Dimitar Kazakov. Using WordNet Similarity and Antonymy Relations to Aid Document Retrieval. In *Recent Advances in Natural Language Processing (RANLP*, 2005. 10

[64] A. Trosborg. *Analysing Professional Genres.* John Benjamins, 2000. 75, 81, 82, 83, 87

[65] E. M. Voorhees. Using WordNet to disambiguate word senses for text retrieval. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 171–180, New York, NY, USA, 1993. ACM. 86

[66] M. Wattenberg, F. Viegas, and K. Hollenbach. Visualizing Activity on Wikipedia with Chromograms. In *Human-Computer Interaction - INTERACT 2007*, pages 272–287. Springer Berlin / Heidelberg, 2007. 11

[67] G. Whittemore, K. Ferrara, and H. Brunner. Empirical study of predictive powers of simple attachment schemes for post-modifier prepositional phrases. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, pages 23–30, Morristown, NJ, USA, 1990. Association for Computational Linguistics. 14

[68] Wikipedia site on antonymy, http://en.wikipedia.org/wiki/Antonymy, (Revised 14 March 2009). 40

[69] Wikipedia web-site, http://en.wikipedia.org. 54

[70] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *INFOVIS '95: Proceedings of the 1995 IEEE Symposium on Information Visualization*, page 51, Washington, DC, USA, 1995. IEEE Computer Society. 51

[71] WiseGeek web site on antimetabole, http://www.wisegeek.com/what-is-antimetabole.htm. 25