2015

Characterization of Prose by Rhetorical Structure for Machine Learning Classification

James Java
Nova Southeastern University, jj626@nova.edu

Characterization of Prose by Rhetorical Structure for Machine Learning Classification

by

James Java

A dissertation report submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in
Computer Science

College of Engineering and Computing
Nova Southeastern University

August 31, 2015

We hereby certify that this dissertation, submitted by James Java, conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

Michael J. Laszlo, Ph.D. Chairperson of Dissertation Committee	Date
Sumitra Mukherjee, Ph.D. Dissertation Committee Member	Date
Amon B. Seagull, Ph.D. Dissertation Committee Member	Date

Approved:

Amon B. Seagull, Ph.D. Interim Dean, College of Engineering and Computing	Date

College of Engineering and Computing
Nova Southeastern University

2015

An Abstract of a Dissertation Report Submitted to Nova Southeastern University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Characterization of Prose by Rhetorical Structure for Machine Learning Classification

by

James Java
August 31, 2015

Measures of classical rhetorical structure in text can improve accuracy in certain types of stylistic classification tasks such as authorship attribution. This research augments the relatively scarce work in the automated identification of rhetorical figures and uses the resulting statistics to characterize an author's rhetorical style. These characterizations of style can then become part of the feature set of various classification models.

Our Rhetorica software identifies 14 classical rhetorical figures in free English text, with generally good precision and recall, and provides summary measures to use in descriptive or classification tasks. Classification models trained on Rhetorica's rhetorical measures paired with lexical features typically performed better at authorship attribution than either set of features used individually. The rhetorical measures also provide new stylistic quantities for describing texts, authors, genres, etc.

Acknowledgments

For Mom & Dad and K. B., for all your love and support.

Thanks to Prof. Michael Laszlo for being my dissertation chair and guide, and to Profs. Sumitra Mukherjee and Amon Seagull (the rest of my dissertation committee) for their excellently pithy advice throughout the process.

Table of Contents

List of Tables

List of Algorithms

List of Figures

Chapter 1

Introduction

Background

The art and science of rhetoric was formulated inductively by the ancients, who studied its long-practiced techniques in action. The earliest known codification of the art of rhetoric as a set of "rules" came from Sicily, where Corax of Syracuse devised a system to help dispossessed citizens argue for the recovery of their property in court (Corbett, 1990). Corax's contribution to rhetorical theory is later acknowledged by the likes of Plato, Aristotle, and Cicero, and the study flourished, growing in complexity. In *Institutio oratoria*, Quintilian categorizes rhetorical practice into five "canons": *inventio* (invention), *dispositio* (arrangement), *elocutio* (style), *memoria* (memory), and *actio* (delivery).

This research in general concerns itself with the third canon, *elocutio*, which treats of style. More specifically, as adopted by Cicero and Quintilian from Theophrastus of Eresus, the style of any oration comprises four *virtues*: correctness of language, clarity, appropriateness, and ornament (Kirchner, 2007); the last is our primary concern. Ornament itself classically has three broad categories: figures of speech, figures of thought, and tropes ([Cicero], 1954). Corbett (1990) uses *figures of speech* to denote "any artful deviations from the ordinary mode of speaking or writing," with two main groups, *schemes* and *tropes*, the latter now subsumed as a figure of speech. Schemes involve deviation from the normal pattern of words; tropes involve deviation from the normal signification of words. Corbett's definition will hold here.

Fahnestock (1999) correctly observes that portraying figures of speech as devia-

tions from the norm demands some a priori definition of normal. Because figures occur as ordinary, acceptable uses of language in both formal and informal situations, they must be an intrinsic part of language; as Du Marsais (1804) says, "There is nothing so natural, so ordinary, and so common as figures in the language of men."[1] Fontanier (1977), also writing in the early nineteenth century, provides an alternative definition in which figures, assumed not to deviate from ordinary or common language, instead deviate from a "simple," more straightforward unfigured form, even though the simpler expression might occur less frequently in practice.

The common but possibly idiosyncratic nature of figures of speech motivates us to examine them for features useful in characterizing an author as similar or dissimilar to other authors. This sort of characterization informs text-classification tasks such as authorship attribution, gender identification, and genre detection. One complaint made of the character and lexical features[2] that often underlie these tasks is their classifiers' failure to explain *why* they work (Love, 2002); the classifiers act as ad-hoc black boxes without any connection to the psychology or neurobiology of writing. Although contemporary research rarely connects syntactic features such as figures of speech back to the authorial why of their discriminatory power, the potential exists to relate them to the stylistic tendencies or choices of the author, hence also giving them explanatory power.

For the purposes of our research, authorship attribution is the statistical or computational discrimination between texts written by different authors through the measurement of appropriate textual features (Stamatatos, 2009). We will begin with the axiomatic assumption that authors' rhetorical figurations are sufficiently idiosyncratic—probably along with other appropriate textual features—to distinguish them from other writers or group them among their writing peers through machine

[1] "[I]l n'y a rien de si naturel, de si ordinaire et de si comun que les figures dans le langage des homes."

[2] E.g. character n-grams, word frequencies, vocabulary "richness." See **Table 1** for more details.

learning. The rest of this paper describes the methods and problems that accompany the task of authorship attribution using classical rhetorical figures.

Problem Statement

Because of the historical prevalence of rhetorical structure in thought, writing, and speech, each of the three categories of classical ornament (see § *Background*) might be open to identification in text by natural language processing techniques, for the purpose of stylistic classification or authorship identification. However, not a lot of work has been done on text classification using such high-level stylometric features. One problem with using stylometric features for text classification is that the detailed analysis required to extract those features usually leads to less accurate and more noisy classification measures. Furthermore, complicated tasks such as full syntactic parsing or semantic analysis are not yet handled well by current NLP technology for unrestricted text. Therefore, few classification techniques make much use of high-level stylometric features (Stamatatos, 2009). Those that do must limit the feature set as much as possible without oversimplification.

Gawryjołek (2009) uses a large corpus of parsed sentences (the Penn Treebank; Marcus, Marcinkiewicz, and Santorini, 1993) with the Stanford Parser (Klein & Manning, 2003) to tag unknown text, and then to algorithmically identify a number of classical rhetorical figures, which can then provide a feature set for characterizing the text or comparing it to others. He reports good precision and recall for the discovery of rhetorical figures involving repetition, but only satisfactory results for other forms.

Linguists generally agree that we learn a language through examples of its use, and that the set of examples we are exposed to is individually unique; therefore, we each create our own unique forms of the language that are yet recognizable as a common tongue, because each form proceeds from an adequately similar set of examples, and because our brains and minds are, probably, not so dissimilar (Strozer, 1994; Rice,

1996; Gopnik, 1997; Chomsky, 1999; O'Grady, 1999; Wexler, 1999; Pinker, 2003). Van Halteren, Baayen, Tweedie, Haverkort, and Neijt (2005) have shown evidence of a human *stylome*, a "set of measurable traits of language products" derived from vocabulary and syntax, which could potentially identify individual authors with high probability. The rhetorical features of text, being derived from vocabulary and syntax, might prove useful as stylomic traits in authorship identification tasks. Syntactic patterns are considered more reliable authorial fingerprints than e.g. lexical information (Stamatatos, 2009); also, the success of function words in representing style indicates the usefulness of syntactic information in authorship identification (Mosteller and Wallace, 1964; Damerau, 1975; Burrows, 1987; Tweedie, Singh, and Holmes, 1996; Karlgren, 2000; D. Holmes, Robertson, and Paez, 2001; Baayen, van Halteren, Neijt, and Tweedie, 2002; Argamon, Koppel, Fine, and Shimoni, 2003; Argamon and Levitan, 2005; Juola and Baayen, 2005; Zhao and Zobel, 2005). The extraction of syntactic information from text, however, requires robust and accurate NLP tools able to perform syntactic analysis; and the resulting data sets are often noisy due to unavoidable errors in parsing (Stamatatos, 2009). Following Gawryjołek's lead, our research attempted to extend his identification techniques and avoid as much noise as possible.

Once an appropriate summary of the information extracted by rhetorical-figure identification has been determined, it can become part of the feature set of a classification task, which would also include lexical, character, and other types of features. Applied feature-selection algorithms could then reduce the dimensionality of the representation (Forman, 2003), making the classification algorithm less likely to overfit on the training data; in text classification, though, there are few irrelevant features, so the most effective classifiers need to use as much of the feature set as possible (Joachims, 1998).

Authorship attribution is of particular interest as a classification task.

Dissertation Goal

The primary goals of this research were, first, to adopt and extend the automatic discovery of classical rhetorical figures described by Gawryjołek (2009); then, second, to develop and test the utility of several summary measures of the discovered figures as a discriminant in authorship attribution tasks.

These goals were achieved in four stages. In the first stage, we developed software, called *Rhetorica*, to identify rhetorical figures in text. The second stage quantified Rhetorica's effectiveness through measures of precision and recall, and tweaked its performance. In the third stage, we sought useful summary statistics of the discovered figures that would, in the fourth stage, become part of the feature set in an authorship-attribution classification model, whose effectiveness could also be quantified through measures of model accuracy.

Research Questions

This research attempts to answer the following questions:

- Gawryjołek (2009) has shown that the identification of classical rhetorical figures of repetition is possible and often quite accurate. Can we improve upon any of the most difficult figures considered, isocolon, oxymoron, and polyptoton, enough to use them as discriminants in authorship attribution tasks?
- What are good summary measures of the discovered figures? Relative frequency? Sequence patterns that somehow represent the distribution of the figures in the text?
- Gamon (2004) used a syntactic parser to measure "syntactic production" (rewrite-rule) frequencies, and found that while the syntactic features alone performed worse than lexical features in an authorship-attribution task, their combined feature set improved the results. What other lexical or syntactic features, if any,

should be combined with rhetorical-figure statistics for authorship-attribution tasks?

- This a lesser question, but still relevant: Could measures of rhetorical figures in texts have other interesting classification uses besides authorship attribution? (See § *Other Work*)

Relevance and Significance

Several studies (e.g. Baayen, van Halteren, and Tweedie, 1996; Hirst and Feiguina, 2007) have shown that syntactic information measures can outperform lexical measures in authorship attribution. However, the complex process of syntactic parsing can lead to less accurate and noisier classification measures, which is why few classification techniques use high-level stylometric features.

However, classical rhetorical structure in spoken and written language is both historically ubiquitous and idiosyncratic, and its representation in classification models might allow for excellent discrimination of auctorial style. The problem is to identify appropriate rhetorical figures with high precision and recall, i.e. without too much noise detrimental to modeling. Gawryjołek (2009) has had some success in this identification, and our research builds on that success while trying to improve any deficits it encounters.

To identify figures of repetition (anadiplosis, anaphora, antimetabole, etc., described in § *Definition of Terms*) requires little syntactic knowledge of a text but can provide potentially interesting information about lexical or phrasal distribution, and has a good success rate.

Of particular interest are rhetorical figures of parallelism, whose identification requires syntactic information. *Isocolon* is a type of parallelism in which phrases of approximately equal length also have corresponding syntactic structure (Lanham, 1991), as in *Proverbs* 23:32, "[A]t the last it biteth like a serpent, and stingeth like an

adder" (KJV); and *Julius Caesar* 3.2.21–22, "Not that I loved Caesar less, but that I loved Rome more." The syntactic measures derived from figures of parallelism such as isocolon might provide excellent discrimination in classification models.

In general, we hope that this research will improve authorship attribution in domains where parsable text is available, with practical applications.

Other Work

Bennett's (1971) collection of essays on the history of English prose style is a redoubtable pre-computational contribution to the field of stylistic text analysis. Though the book is not particularly long, it manages to present a thorough, often quantified, overview of prose style in English (British and American) from the Anglo-Saxon of Ælfric through well into the twentieth century.

Though the book offers some comparison of various prose stylists by sentence length, frequency of functional words, and sentence diagramming, it is a work of neither statistics nor modern linguistics, preferring to rely on representative passages from authors rather than on corpora to justify its classification of the authors into one stylistic group or another. For example, some essays on the British Augustan writers (early eighteenth century) exemplify Samuel Johnson for his "parallelism," Joseph Addison for his "Neo-classicism," and Laurence Sterne for his "Senecan loose" style. While each of these styles is described and pointed out in the representative passages, one can hardly help wondering, without being excessively well-read in each author, whether a certain style obtains throughout his work, or whether it is mostly found in specific memorable passages.

One of the book's appendices, "A Contextual Method for the Description of Prose Style" (pp. 224–231), cites Halliday, 1967 as one inspiration for its taxonomy of style, which includes categories for function words and certain stylistic patterns that appear germinal for Halliday's later work on Systemic Functional Grammar (SFG,

Halliday, 1994). Nowadays, computational power has enabled expanded stylistic text analysis beyond the imagination of most of the scholars writing for Bennett's book forty-some years ago, and the modern reader of this still-useful book will feel the lack of computational analysis.

By examining the prevalence of specific rhetorical figures in the writings of certain authors, one might find some concordance between them and the qualitative categories in Bennett. For example, Addison, Dryden, and Swift are thought to have similar prose styles, but each with his own idiosyncratic touches; stylistic classification using these authors' works together as a corpus might help quantify both the similarities and the differences, and add some depth to the analyses in Bennett. Of course, such analyses should not be limited to Bennett's selections, and it remains a matter for discovery where their application would be most suitable.

Barriers and Issues

Many classification measures for authorship-attribution models are easy to obtain, such as sentence length and frequency of function words. Currently, that is not true of classical rhetorical figures; very few studies have sought to identify them in text, and none have used their summary measures for further classification studies. The work that has been done on figures whose identification requires syntactical information may be inadequate for such further studies, and needs augmentation. This research has provided some of that augmentation and begun to use rhetorical figures for authorship attribution.

As laid out in § *Dissertation Goal* and this chapter, the goals of this research demanded a large amount of study, coding, and testing in an attempt improve and add knowledge to the field of authorship attribution, in a sub-area (syntactic measures) where existing research is somewhat sparse.

Assumptions, Limitations, and Delimitations

We are working under the axiomatic assumption that authors' rhetorical figurations are sufficiently idiosyncratic to distinguish them from other writers or group them among their writing peers through machine learning. This assumption is not unreasonable given the long history of rhetoric as "artful" language that deviates from everyday expression, and its cultivation as a means of elevating discourse, but a rich cultural history may not in the end distinguish itself sufficiently in classification tasks.

Our discovery of rhetorical figures is limited by the abilities of Rhetorica and the NLP tools it incorporates. Because of the dearth of work on classical rhetorical figures in machine learning, it makes sense to begin with the simplest tropes and figures of repetition for classification instead of using noisier semantic figures whose difficult discovery would probably hinder classification tasks rather than enhance them.

Definition of Terms

The definitions of individual figures of speech derive from Gawryjołek (2009), Harris and DiMarco (2009), Quinn (1982), Lanham (1991), Corbett and Connors (1998), Fahnestock (1999), Crowley and Hawhee (2004), Burton (2007), and Farnsworth (2011).

Anadiplosis Repetition of the ending word or phrase from the previous clause at the beginning of the next. *Who has not the spirit of his age, of his age has all the unhappiness.* (Voltaire)

Anaphora Repetition of a word or phrase at the beginning of successive clauses; cf. **Epistrophe**. *He maketh me to lie down in green pastures: he leadeth me beside the still waters. He restoreth my soul: he leadeth me in the paths of righteousness for his name's sake.* (Ps. 23:2)

Antimetabole Repetition of words in reverse grammatical order. *Ask not what your country can do for you; ask what you can do for your country.* (John F. Kennedy)

Authorship Attribution Distinguishing texts written by different authors through statistical or computational techniques. (Stamatatos, 2009)

Chiasmus Repetition of grammatical structures in reverse order; cf. **Antimetabole**, **Isocolon**. *His time a moment, and a point his space.* (Pope)

Clause A short **Sentence** within a larger one; contains both a subject and predicate.

Conduplicatio The repetition of a word or phrase; broader than **Ploce**. *Then thou thy regal Sceptre shalt lay be, / For regal Sceptre then no more shall need, / God shall be All in All.* (Milton)

Corpus (*pl.* **Corpora**) A body of spoken or written words.

Epanalepsis Repetition at the end of a clause of the word or phrase that began it. *Once more unto the breach, dear friends, once more.* (*Henry V* 3.1)

Epistrophe Repetition of the same word or phrase at the end of successive clauses; cf. **Anaphora**. *Towards thee I roll, thou all-destroying but unconquering whale; to the last I grapple with thee; from hell's heart I stab at thee; for hate's sake I spit my last breath at thee.* (Melville, *Moby-Dick*)

Epizeuxis Repetition of a word or phrase with no others between. *To the swinging and the ringing / Of the bells, bells, bells— / Of the bells, bells, bells, bells, / Bells, bells, bells— / To the rhyming and the chiming of the bells!* (Poe)

Feature In machine learning, a measurable property of an object or event to be classified.

Feature Vector In machine learning, an n-dimensional vector whose elements are single **Features** of an object or event to be classified. The feature vector $\boldsymbol{x}$ is hence a point in an n-dimensional *feature space*, where

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

(Duda, Hart, & Stork, 2000)

Figure of Speech See **Rhetorical Figure**.

Grammar A set of rules in language for organizing meaningful parts into well-formed sentences; also the study of these rules. Grammar includes **Morphology** and **Syntax**.

Isocolon Repetition of grammatical structure in nearby phrases or clauses of approximately equal length; cf. **Chiasmus**. *The bigger they are, the harder they fall.*

Lexical Relating to words or vocabulary in language.

Morphology The analysis or study in language of the forms and inflections of words. Part of **Grammar**.

Oxymoron A terse paradox; the yoking of two contradictory terms. *Darkness visible.* (Milton)

Parser In natural language processing, a program that tries to determine the grammatical structure of sentences by grouping words into phrases and marking each word as a part of speech.

Phrase A unified group of words in a **Sentence** that does not include both a subject and predicate; a syntactic unit larger than a word but smaller than a **Clause** (OED Online, 2006a).

Ploce The repetition of word in a short span of text for rhetorical emphasis. *They are not all Israel, which are of Israel* (Rom. 9:6)

Polyptoton Repetition of a word in a different form; having cognate words in close proximity. *Who shall stand guard to the guards themselves?* (Juvenal)

Polysyndeton "Excessive" repetition of conjunctions between clauses. *The horizon narrowed and widened, and dipped and rose, and at all times its edge was jagged with waves that seemed thrust up in points like rocks.* (Crane)

Precision In text classification, the total number of documents correctly labeled,

divided by the total number of documents assigned the same label. Mathematically, the estimated precision $\text{prec}(h)$ of a classification rule h is

$$\text{prec}(h) = \frac{f_{++}}{f_{++} + f_{+-}},$$

where f_{++} is the total number of correctly labeled documents (true positives) and f_{+-} is the total number of documents *mislabeled* with the same label (false positives) (Joachims, 2002). *High* precision, a measure of exactness, means that more documents were correctly labeled than mislabeled. Cf. **Recall**.

Recall In text classification, the total number of documents correctly labeled, divided by the total number of documents that *should* have the same label. Mathematically, the estimated recall $\text{rec}(h)$ of a classification rule h is

$$\text{rec}(h) = \frac{f_{++}}{f_{++} + f_{-+}},$$

where f_{++} is the total number of correctly labeled documents (true positives) and f_{-+} is the total number of documents *mislabeled* with the other label (false negatives) (Joachims, 2002). *High* recall, a measure of completeness, means that most of the documents were correctly labeled. Cf. **Precision**.

Rhetorical Figure An artful deviation from the ordinary ways of speaking or writing. Rhetorical figures comprise two main groups, **Schemes** and **Tropes**.

Scheme Deviation from the normal *pattern* of words in speech or writing.

Sentence A group of words between two full stops that forms a grammatically complete expression; contains both a subject and predicate (OED Online, 2006b).

Stylometry Using measurable **Features** of a literary style for statistical or computational analysis. See **Table 1** for some stylometric features of text.

Support Vector Machines (SVM) A kind of machine learning often used in text-classification problems. The simplest linear form of a SVM is a hyperplane that

separates a set of positively classified items from a set of negatively classified ones based on the structural risk minimization procedure (Diederich, Kindermann, Leopold, & Paass, 2003). From appropriately weighted and transformed **Feature Vectors**, a SVM model can find an optimal discriminatory hyperplane, and thereby the best classification, given a particular feature set.

Symploce Repetition of a word or phrase at the beginning, and of another at the end, of successive clauses; the combination of **Anaphora** and **Epistrophe**. *Most true that I must fair Fidessa love, / Most true that I fair Fidessa cannot love.* (B. Griffin)

Syntax The analysis or study in language of the arrangement of words, phrases, and clauses in well-formed sentences. Part of **Grammar**.

Trope Deviation from the normal *signification* of words in speech or writing.

Summary

This introduction discusses the history of classical rhetoric and its systematization, defines terms, and puts the focus of our work on a particular subset of rhetorical figures, *schemes* and *tropes*. We assess the lack of research on the automatic discovery of rhetorical figures, think about how that sort of discovery might happen computationally, and plan to use counts of discovered rhetorical figures in text—or some other summary measure—for authorship-attribution tasks or possibly descriptive categorization.

Chapter 2

Review of the Literature

Koppel, Schler, and Argamon (2009) provides a history of methods in textual authorship attribution, discussing both the ineffective "unitary invariant" approach (Zipf, 1932; Yule, 1944), which searched for authorially unique properties of textual statistics; and the later, more effective multivariate-analysis approach, which essentially maps documents characterized by their features onto some multidimensional space, then assigns the most probable attribution of a questioned document to the author whose documents are "closest" in that space, according to some apt distance measure.

The features used in multivariate analyses of authorship attribution include *complexity measures* such as *hapax legomena*, word length in syllables (Fucks, 1952) or letters (Brinegar, 1963), the average number of words in a sentence (Morton, 1965), and vocabulary richness (e.g. Yule's (1944) K-measure, Sichel's (1975) S-measure, Honoré's (1979) R-measure); *function words*, words with little lexical meaning that express grammatical relationships, which are quite effective in various contexts (Morton, 1978; Burrows, 1987; Karlgren & Cutting, 1994; Kessler, Numberg, & Schütze, 1997; David I. Holmes, 1998; D. Holmes et al., 2001; Baayen et al., 2002; Binongo, 2003; Argamon & Levitan, 2005; Juola & Baayen, 2005; Koppel, Schler, & Zigdon, 2005; Zhao & Zobel, 2005; Koppel, Akiva, & Dagan, 2006); *character n-grams*, whose frequencies might reflect lexical preferences (Kjell, 1994a, 1994b; Ledger & Merriam, 1994; Kjell, Addison Woods, & Frieder, 1995; Clement & Sharp, 2003; Houvardas & Stamatatos, 2006; Stamatatos, 2008); and *syntactic* features (which are discussed

below), among others.

Stamatatos (2009) is a survey of recent advances of the automated approaches to attributing authorship, which examines their characteristics for both text representation and text classification. The focus of the survey is on computational requirements and settings rather than on linguistic or literary issues: "The main idea behind statistically or computationally supported authorship attribution is that by measuring some textual features, we can distinguish between texts written by different authors."

Current machine learning methods allow the consideration of many diverse, potentially relevant textual features without the threat of degraded accuracy if many of these features turn out to be irrelevant for classification. Feature sets based on the relative frequencies of syntactic structures have become possible through continual improvements in computational speed and the development of quick, reliable statistical NLP techniques (Koppel et al., 2009). Several studies relying on the syntactic output of chunkers and parsers to augment feature sets give classification results considerably better than studies using only word-based features, e.g. Baayen et al. (1996), Stamatatos, Fakotakis, and Kokkinakis (2000, 2001), van Halteren (2004), Gamon (2004), Chaski (2005), Uzuner and Katz (2005), and Hirst and Feiguina (2007). A number of other studies have used frequencies of POS sequences to approximate syntactic features (Argamon-Engelson, Koppel, and Avneri, 1998; de Vel, 2000; Kukushkina, Polikarpov, and Khmelev, 2001; Koppel, Argamon, and Shimoni, 2002; Koppel and Schler, 2003; Koppel et al., 2005, 2006; Zhao, Zobel, and Vines, 2006; Zheng, Li, Chen, and Huang, 2006).

There is little published work on the computational identification of rhetorical figures of speech, though some peripheral studies exist. Rhetorical Structure Theory (RST) (Mann & Thompson, 1988; Taboada & Mann, 2006) is a theory of the organization of discourse in a text which relies on hierarchical conceptual relationships between parts of the text; in practice it identifies semantic-based rhetorical devices

such as antithesis and restatement. Argamon et al. (2007) describes stylistic classification using mainly lexical features which resulted in improved discrimination among the texts in the evaluation corpora. Computational stylistic text analysis has been used for authorship attribution and profiling, genre-based text classification, sentiment analysis, spam filtering, criminal and national security forensics, text mining, and bolstering humanities scholarship (Argamon et al., 2007). Most past work in computational stylistics has been based on sets of content-independent features chosen by the researcher, such as function words (Mosteller & Wallace, 1964; Tweedie et al., 1996; Matthews & Merriam, 1997), clause-complexity measures (de Vel, 2000; Yule, 1944), and POS (part-of-speech) structures and syntax (Stamatatos et al., 2000).

Gawryjołek's thesis (2009) appears to present the first in-depth work attempting to find specific rhetorical figures in text and identify them as such. He reports good precision and recall for the discovery of rhetorical figures involving repetition, but only satisfactory results for other forms. Some of the rhetorical figures he discusses are *anaphora* (repetition of the same word or group of words at the beginning of successive clauses, sentences, or lines), *isocolon* (a series of similarly structured elements having approximately the same length), *oxymoron* (the yoking of two terms that are ordinarily contradictory), and *polyptoton* (using a cognate of a given word in close proximity), among others. The thesis is ambitious and well-considered. Gawryjołek makes use of sliding sentence "windows," parse trees, stemmers, and WordNet to try and meet his goals, and does so with decent success. A possible improvement on this approach, instead of using a fixed set of algorithms to find a small number of rhetorical figures, would be to use a more flexible, non-deterministic statistical method of discovery (possibly derived from the rhetorical frequencies in a suitably tagged corpus) that might detect very large or widely separated patterns.

Strommer (2011) describes a method of using "shallow" rhetorical figures, specifically tropes, to evaluate authorial intent. It distinguishes itself from Gawryjołek by

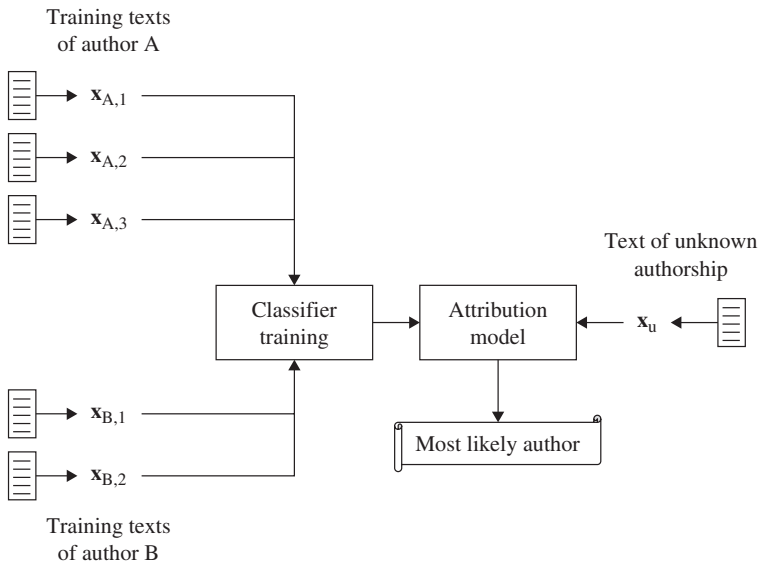
focusing on semantic rather than syntactic analysis.

The majority of modern authorship-identification approaches use multiple training instances for each class to extract a reliable attribution model, where each training text sample is considered a unit that contributes separately to the model:

- Each text sample of the training corpus is represented by a vector of attributes (x).
- A classification algorithm is trained to develop an attribution model.
- Such classification algorithms require multiple training instances per class for extracting a reliable model.

(See **Figure 1**.) Therefore, in the case of only one long training text for a particular candidate author (e.g. an entire book), it should be segmented into multiple parts, probably of equal length.

Figure 1: Typical architecture of instance-based approaches. (From Stamatatos, 2009.)



Because training sets are represented as multivariate collections of features, each text can be considered a vector situated in multivariate space. A variety of robust statistical and machine-learning algorithms can then be used for building a classification model, including discriminant analysis (Chaski, 2005; Stamatatos et al., 2000),

support vector machines (Joachims, 1998; de Vel, Anderson, Corney, & Mohay, 2001; Diederich et al., 2003; Li, Zheng, & Chen, 2006; Sanderson & Guenter, 2006; Koppel & Schler, 2004), decision trees (Uzuner & Katz, 2005; Zhao & Zobel, 2005; Zheng et al., 2006), neural networks (Matthews & Merriam, 1993; T. V. N. Merriam & Matthews, 1994; Tweedie et al., 1996; Matthews & Merriam, 1997; Zheng et al., 2006), and genetic algorithms (D. I. Holmes & Forsyth, 1995), *inter alia*.

Support vector machines (SVM) are a kind of machine learning often used in text-classification problems (Joachims, 2002). The simplest linear form of a SVM is a hyperplane that separates a set of positively classified items from a set of negatively classified ones, with a maximum *margin*, the interclass distance, based on the structural risk minimization procedure (Diederich et al., 2003). From appropriately weighted and transformed feature vectors, a SVM model can find an optimal discriminatory hyperplane, and thereby the best classification, given a particular feature set. SVMs are particularly well-suited to text-classification problems for the following reasons (Joachims, 1998):

- they can handle high-dimensional feature spaces and protect against overfitting;
- they work well with unreduced feature spaces, which is important because in text categorization very few features are irrelevant;
- they work with sparse feature vectors (i.e. those with few non-zero elements), which are typical of document vectors;
- most text-classification problems are linearly separable.

Chapter 3

Methodology

Overview

This project expanded upon Gawryjołek's (2009) discovery of rhetorical figures with good precision and recall, then used rhetorical-figure count and other summary statistics of rhetorical structure for authorship identification and stylistic classification.

Rhetorical Figures

We developed software called *Rhetorica* (see § *Rhetorica* for details) to extract rhetorical figures from text. Rhetorica attempts to find and summarize the figures defined briefly in § *Definition of Terms* and detailed later in this chapter using the formalism for representing rhetorical figures set down in Harris and DiMarco (2009) where possible.

Classification

For authorship-identification tasks we used multiple samples of each author's writing (by segmenting longer works if necessary) to train classification algorithms for the development of attribution models. Each sample of the training corpus is represented by a vector of attributes in multivariate space. As discussed previously, some summary measure of the Rhetorica software's output contributed to the attribute vector; but since syntactic features alone sometimes perform worse than lexical features in authorship-identification tasks (as in e.g. Gamon, 2004), we used vectors of

Table 1: Types of stylometric features. (Adapted from Stamatatos, 2009.)

Type	Examples
Lexical	Token-based (word length, sentence length, etc.)
	Vocabulary richness
	Word frequencies
	Word n-grams
	Errors
Character	Character types (letters, digits, etc.)
	Character n-grams (fixed length)
	Character n-grams (variable length)
	Compression methods
Syntactic	Part-of-speech (POS)
	Chunks
	Sentence and phrase structure
	Rewrite rules frequencies
	Errors
Semantic	Synonyms
	Semantic dependencies
Application-specific	Functional
	Structural
	Content-specific
	Language-specific

rhetorical features both alone and along with other stylometric features for developing attribution models. A list of basic stylometric features is presented in **Table 1**. We primarily considered only the first three categories—*lexical*, *character*, and *syntactic*—as adjunct elements of our rhetorical attribute vectors (though finding the rhetorical figure oxymoron does require semantic information), whose examples are found detailed practically in studies such as Graham, Hirst, and Marthi (2005) and Hirst and Feiguina (2007).

We developed attribution models using support vector machine (SVM; Joachims, 2002) libraries available for the R programming language and environment (§ *R*).

Corpora and Tasks

Juola, Sofko, and Brennan (2006) describe a shambolic state of affairs in which authorship attribution in particular and stylometry in general suffer from a lack of common practices and known error rates. Appealing to U.S. law for standards of admissibility of scientific evidence (which include empirical validation of techniques, an established body of practices, and known measures of accuracy), the authors argue that authorship attribution cannot at present meet those standards, and so they propose "some new methodological and practical developments in the field of authorship attribution," among which is a list of public-domain copora and specific problems based on them as an exemplary set of classification tasks; the corpora comprise short and long works in Middle English, modern English, and several other languages:

- *Problem A* (English) Fixed-topic essays written by thirteen Duquesne students during fall 2003.
- *Problem B* (English) Free-topic essays written by thirteen Duquesne students during fall 2003.
- *Problem C* (English) Novels by 19th century American authors (Cooper, Crane, Hawthorne, Irving, Twain, and 'none-of-the-above'), truncated to 100,000 characters.
- *Problem D* (English) First act of plays by Elizabethan/Jacobean playwrights (Johnson, Marlowe, Shakespeare, and 'none-of-the-above').
- *Problem E* (English) Plays in their entirety by Elizabethan/Jacobean playwrights (Johnson, Marlowe, Shakespeare, and 'none-of-the-above').
- *Problem F* ([Middle] English) Letters, specifically extracts from the Paston letters (by Margaret Paston, John Paston II, and John Paston III, and 'none-of-the-above' [Agnes Paston]).
- *Problem G* (English) Novels, by Edgar Rice Burroughs, divided into "early" (pre-1914) novels, and "late" (post-1920).
- *Problem H* (English) Transcripts of unrestricted speech gathered during committee meetings, taken from the *Corpus of Spoken Professional American-English.*
- *Problem I* (French) Novels by Hugo and Dumas (*père*).
- *Problem J* (French) Training set identical to previous problem. Testing set is one play by each, thus testing ability to deal with cross-genre data.
- *Problem K* (Serbian-Slavonic) Short excerpts from *The Lives of Kings and Archbishops*, attributed to Archbishop Danilo and two unnamed authors (A and B). Data was originally received from Alexsandar Kostic.
- *Problem L* (Latin) Elegaic poems from classical Latin authors (Catullus, Ovid, Propertius, and Tibullus).
- *Problem M* (Dutch) Fixed-topic essays written by Dutch college students, received from Hans van Halteren.

(From Juola et al., 2006.)

We had hoped to attempt these tasks in their entirety. Unfortunately, the discovery of some of the rhetorical figures we examined involved computing resources closely tied to the modern English language; for example, finding instances of isocolon required the Stanford PCFG (probabilistic context-free grammar) Parser (Klein & Manning, 2003), and oxymoron required the WordNet database (Miller, 1995). Since the English-only problem was not easily overcome here, we developed attribution models based only on the English-language corpora suggested by Juola et al. (2006) above.

Another standard problem in authorship identification is that of the *Federalist* papers. They were written in 1787–1788 by Alexander Hamilton, John Jay, and James Madison to convince the citizens of New York State to ratify the U.S. Constitution. These 85 short essays, each about 900–3500 words long, were published under the pseudonym "Publius"; 77 of them first appeared in several newspapers, and Hamilton later wrote the 8 complementary ones. Of the first 77, Nos. 2–5 and 64 were written by Jay; Nos. 10, 14, and 37–48 by Madison; Nos. 18–20 by both Hamilton and Madison; Nos. 49–58, 62, and 63 by either Madison or Hamilton—these are known as the "disputed papers"; and the rest by Hamilton. In the authorship identification problem, the author of the disputed papers is assumed to be either Hamilton or Madison, and the disputed papers are classified by an attribution model trained from the other *Federalist* papers of known authorship, and also some non-*Federalist* works by each writer (Mosteller & Wallace, 1964; Tweedie et al., 1996). This problem was a good test of the discriminatory power of our rhetorical models.

Hirst and Feiguina (2007) consider the problem of distinguishing the writings of Charlotte Brontë from those of her sister Anne. This problem is difficult, they say, because the sisters are "of the same era, same social and economic background, and same gender; they had similar educations; they strongly influenced one another in the

development of their writing; and their novels are similar in genre. Any differences can be attributed only to elements of individual style." The novels used are Charlotte's *Villette* (1853), and Anne's *Agnes Grey* (1847) and *The Tenant of Wildfell Hall* (1848). We also attempted this classification task with our rhetorical models.

Figure Detection

This section briefly describes our approach to the automatic detection of various rhetorical figures in English text.

Syntactic Units

The context for our detection of figures (which are made up of words) is phrases, clauses, and sentences; we do not consider as a whole any syntactic units larger than sentences. The first step, then, is to find sentence boundaries within a text. Although it is an important problem in natural language processing, *sentence boundary detection* (or *disambiguation*, SBD) seems underrepresented in the literature (Gillick, 2009). Nevertheless, several effective software implementations of SBD exist for resolving text into sentences; we have chosen the SBD functionality of the Apache OpenNLP framework (Baldridge, Morton, & Bierner, 2002), which uses the maximum entropy model proposed by Ratnaparkhi (1998).

Sentences contain phrases and sometimes clauses, both of which can provide a tighter search context for certain figures. We find phrases and clauses with the Stanford PCFG (probabilistic context-free grammar) Parser (Klein & Manning, 2003), discussed below in more detail. Once all the sentences have been detected, each sentence is parsed, tokenized, and then broken into phrases and clauses derived from the parse tree. In addition to the parser-derived phrases, any group of words between medial punctuation marks is included in the collection of phrases.

Parser

A natural language parser is a program that tries to determine the grammatical structure of sentences by grouping words into phrases and marking each word as a part of speech. *Probabilistic* parsers apply statistical knowledge of hand-parsed corpora (or sometimes knowledge induced from unannotated corpora) of sentences to produce a most-likely parse tree of new sentences. A probabilistic context-free grammar (PCFG) parser uses a CFG for which each production rule has a probability.

The Stanford PCFG (Klein & Manning, 2003) parser improves on the shortfalls of a plain PCFG parser (which are summarized in e.g. Jurafsky & Martin, 2009) by joining it with a lexical dependency parser to improve performance, since lexical dependencies can resolve otherwise ambiguous grammatical relations (Hindle & Rooth, 1993).The Stanford PCFG performs well in most cases, but sometimes fails to choose the correct parse, which can adversely affect our detection of rhetorical figures.

The detection of rhetorical figures follows sentence boundary detection and the resolution of each sentence into constituent parts by the Stanford parser, which defines subpart *phrases* and *clauses* according to the syntactic relations of the sentence's parse tree, and therefore—for computational utility—somewhat more specifically than in § *Definition of Terms*. For each sentence the Stanford parser produces a Penn Treebank (Marcus et al., 1993) style[1] general (or *unranked*) tree, a hierarchical collection of nodes having arbitrary numbers of subordinate child nodes, as in **Figure 2** (which is described below). A node without any children is called a *leaf*, and a *preterminal* node has only one child, itself a leaf. A *phrase* is then a node which is not a leaf or a preterminal, instead having two or more children, one of which is not a leaf; phrases are denoted by the ∗P-tags in the Penn Treebank syntactic tag set (**Table 11**). A *clause* typically comprises a noun phrase (Penn Treebank syntactic tag NP) as a

[1]The Penn Treebank part-of-speech (POS) and syntactic tag sets are summarized in **Appendix A**.

subject, and a verb phrase (VP) as a predicate, under the root node of a containing sentence, and is denoted by one of the S∗-tags in the Penn Treebank syntactic tag set.

Figure 2: Parse tree for the sentence "This is a test of the Emergency Broadcast System; this is only a test."

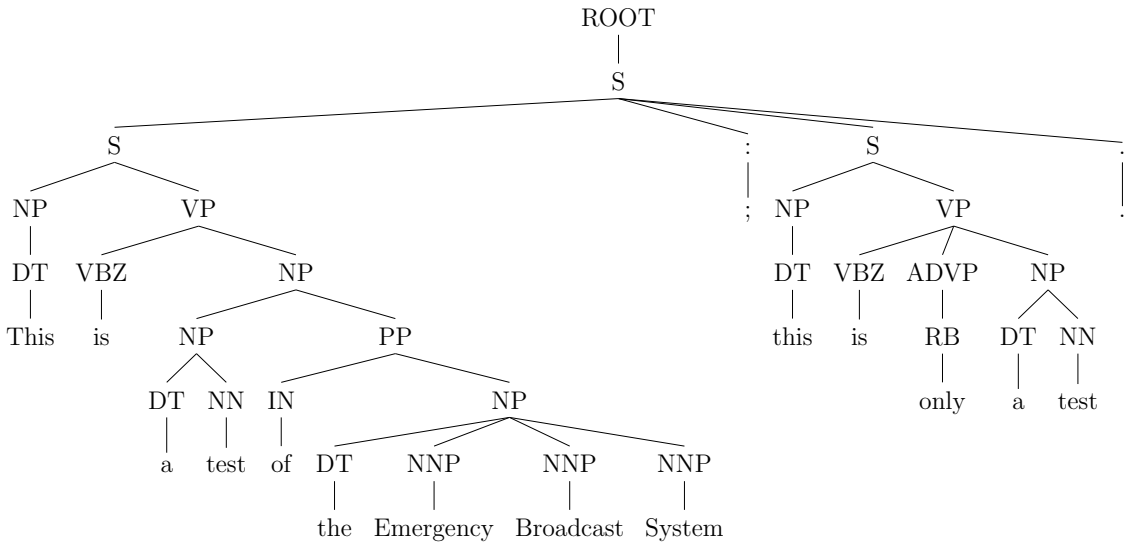


Figure 2 shows an example parse tree with several phrases and clauses. Each of the two S-clauses under the root contains a NP+VP subject-predicate pair, and those ∗P-phrases contain leaves and additional phrases. Our motivation for identifying both the constituent clauses and phrases in a sentence is to provide figure-specific search context for finding each rhetorical figure under consideration here. Besides the parser-derived phrases, any group of words between medial punctuation marks is considered a phrase, and is included with the other phrases to mitigate the effect of misparsing on figure discovery.

Phrases and clauses derived from the parser or from punctuation alone are stored without any punctuation tokens, since punctuation does not otherwise influence the discovery of rhetorical figures by our Rhetorica software.

Schemes—Figures of Repetition

Rhetorical repetition in language can produce rhythm, emphasis, humor, or strong emotional effect (Corbett, 1990). This section considers figures of speech in which words, phrases, clauses, or grammatical structures repeat.

Algorithm 1 Detecting figures of repetition.

```
1: function FindFigureOfRepetition
2:     Create empty collection C for instances of rhetorical figure
3:     for all sentences S_i do
4:         for all sliding windows W_j whose origin was S_i do
5:             if conditions of the figure definition are met in words F ⊆ W_j then
6:                 add figure F to C
7:             end if
8:         end for
9:     end for
10:    return C
11: end function
```

We search for figures of repetition within successive syntactic units no larger than the sentence. While repetitions could occur in successive instances of larger syntactic units such as paragraphs, such repetitions are rarely deliberate and carry little rhetorical value. We follow after Gawryjołek (2009) in searching for figures at the sentence level, but also allowing each sentence to become part of a sliding "window" of several sentences in which repetitions just outside the window can still become part of the original figure when the window is moved. **Algorithm 1** presents a generic outline for the discovery of figures of repetition.

In the sections that follow, we adopt the formalism of Harris and DiMarco (2009) for representing rhetorical figures as defined in **Table 2**. The definitions of individual figures of speech below and in succeeding sections derive from Gawryjołek (2009), Harris and DiMarco (2009), Quinn (1982), Lanham (1991), Corbett and Connors (1998), Fahnestock (1999), Crowley and Hawhee (2004), Burton (2007), and Farnsworth (2011).

Table 2: Formalism for representing rhetorical figures. (Adapted from Harris and DiMarco, 2009.)

Element	Meaning
P	phrase
W	word
S	stem
M	morpheme
$\ldots$	arbitrary intervening material*
$\{\ldots\}$	morpheme boundaries
$[\ldots]$	word boundaries
$\langle\ldots\rangle$	phrase or clause boundaries
$_{\mathrm{a}}, _{\mathrm{b}}, \ldots$	identity $_{\mathrm{a}=\mathrm{a}}$, nonidentity $_{\mathrm{a}\neq\mathrm{b}}$

* Possibly null, with some upper limit; the shorthand is *proximal*.

Epizeuxis

Definition 1 (Epizeuxis)**.** *Repetition of a word or phrase with no others between. Formally,*

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

Example 1. *Alone, alone, all, all alone, / Alone on a wide wide sea!*[2]

Example 2. *The horror! The horror!*[3]

Epizeuxis is often defined as the immediate repetition of a word only, with the term *epimone* covering contiguous phrases; however, because epimone is sometimes more loosely the "frequent repetition of a phrase" (Lanham, 1991) without the immediacy, and because enough standard rhetorical primers allow conflation of epizeuxis and epimone (e.g. Quinn, 1982; Farnsworth, 2011), we have decided to group both figures together here as *epizeuxis*.

Epizeuxis detection is straightforward. Within each search window we look for contiguous repetitions of the same word or phrase; letter case is ignored.

[2] Coleridge, *The Rime of the Ancient Mariner*.
[3] Conrad, *Heart of Darkness*.

Ploce

Definition 2 (Ploce)**.** *The repetition of word in a short span of text for rhetorical emphasis. Formally,*

$$[W]_a \dots [W]_a$$

Example 3. *And my poor fool is hanged! No, no, no life!/ Why should a dog, a horse, a rat have life,/ And thou no breath at all?*[4]

Example 4. *Bloody Vikings. You can't have egg, bacon, Spam and sausage without the Spam.*[5]

By default, the scope of ploce is a search window of two sentences; the figure's emphasis is strengthened by proximity of the repetition. Ploce has some overlap with epizeuxis, insofar as immediate repetitions (the definition of epizeuxis) are also counted in the search for instances of ploce. However, we do ignore high-frequency stop words (v. **Appendix B**) while searching, which if counted would surely dilute the idiosyncrasy of ploce as a deviation from common language; note that in the first example only the repetition of *life* is labeled ploce, while the repeated stop words (*and*, *no*, *a*) are not.

Conduplicatio

Definition 3 (Conduplicatio)**.** *The repetition of a word or phrase; broader than ploce. Formally,*

$$[W]_a \dots [W]_a$$

$$\langle P \rangle_a \dots \langle P \rangle_a$$

[4] *King Lear* 5.3.
[5] Monty Python, "Spam" sketch.

Because ploce in most rhetorics very specifically refers to the repetition of only a single word, we have implemented conduplicatio, the non-contiguous repetition of a word or phrase, mostly to catch repeated phrases outside the scope of single-word ploce. The overlap of ploce and conduplicatio is more a matter of definition than of necessity, and the latter figure is included here for completeness. If the search for conduplicatio finds a repeated phrase made up entirely of stop words (v. **Appendix B**), the phrase is rejected as an instance of conduplicatio; also, contiguous repeated phrases are rejected to avoid making conduplicatio a proper superset of epizeuxis.

Polysyndeton

Definition 4 (Polysyndeton)**.** *"Excessive" repetition of conjunctions between clauses. Formally,*

$$and \ldots and \ldots and \ldots$$ [6]

Example 5. *And Joshua, and all Israel with him, took Achan the son of Zerah, and the silver, and the garment, and the wedge of gold, and his sons, and his daughters, and his oxen, and his asses, and his sheep, and his tent, and all that he had: and they brought them unto the valley of Achor. And Joshua said, Why hast thou troubled us?*[7]

Farnsworth (2011) lists the many uses of polysyndeton, the most prominent being:

- To create rhythm that might not otherwise exist.
- To regulate the pace of utterance; polysyndeton more commonly slows the pace, but it can also create a bouncy haste depending on the context.
- To emphasize singly the items in a list.

Following Gawryjołek (2009), our search window for polysyndeton is a single sen-

[6] More than two repetitions are possible.
[7] Josh. 7:24–25.

tence, but we also look for the same conjunction beginning two consecutive sentences. In the example above, in addition to the excessive (where *excessive* means more than two of the same word) repetition of *and* in the first sentence, the two underlined *and*s, each starting one of the sentences, also make an instance of polysyndeton.

Anaphora

Definition 5 (Anaphora)**.** *Repetition of a word or phrase at the beginning of successive clauses. Formally,*

$$\langle [W]_a \ldots \rangle \langle [W]_a \ldots \rangle$$

$$\langle \langle P \rangle_a \ldots \rangle \langle \langle P \rangle_a \ldots \rangle$$

Example 6. *Strike as I struck the foe! Strike as I would / Have struck those tyrants! Strike deep as my curse! / Strike!—and but once!*[8]

Anaphora is among the rhetorical figures whose definition includes not only repetition, but also syntactic position. We look for instances of anaphora in a (default) three-sentence window; while the figure is most effective when starting off truly successive clauses, we also allow instances of anaphora in which the repetitions occur in non-successive clauses within the search window. In searching for repetitions, we ignore leading determiners, conjunctions, and prepositions in the comparison subsequences.

In the example above, the single repetition of the phrase *Strike as* forms an anaphora, as does the triple repetition of the word *Strike*.

[8] Byron, *Marino Faliero, Doge of Venice*.

Epistrophe

Definition 6 (Epistrophe)**.** *Repetition of the same word or phrase at the end of successive clauses. Formally,*

$$\langle \dots [W]_a \rangle \langle \dots [W]_a \rangle$$
$$\langle \dots \langle P \rangle_a \rangle \langle \dots \langle P \rangle_a \rangle$$

Example 7. *"Business!" cried the Ghost, wringing its hands again. "Mankind was my business. The common welfare was my business; charity, mercy, forbearance, and benevolence, were, all, my business. The dealings of my trade were but a drop of water in the comprehensive ocean of my business!"*[9]

Epistrophe is similar to anaphora, but the repetition occurs at the end of clauses instead of the beginning. As in anaphora, we also allow repetitions in non-successive clauses within the search window (default three sentences).

The example contains two instances of epistrophe: *business* ($\times 5$) and *my business* ($\times 4$).

Symploce

Definition 7 (Symploce)**.** *Repetition of a word or phrase at the beginning, and of another at the end, of successive clauses. Formally,*

$$\langle [W]_a \dots [W]_b \rangle \langle [W]_a \dots [W]_b \rangle$$
$$\langle \langle P \rangle_a \dots \langle P \rangle_b \rangle \langle \langle P \rangle_a \dots \langle P \rangle_b \rangle$$

Example 8. *When I was a child, I spake as a child, I understood as a child, I thought as a child: but when I became a man, I put away childish things.*[10]

[9] Dickens, *A Christmas Carol*.
[10] 1 Cor. 13:11.

Symploce looks like a conflation of anaphora and epistrophe, with repeated words at both the start and the end of successive clauses; there is some leeway here in its composition, insofar as repetitions in non-successive clauses, but still within the search window, also count as instances of symploce. In searching for symploce, we ignore leading or trailing conjunctions in the comparison subsequences. Note that the searches for anaphora and epistrophe are separate from that of symploce, so Rhetorica should count all three figures once for each instance of symploce.

The example above contains two instances of symploce: *I . . . a child* ($\times 4$) and *I . . . as a child* ($\times 3$).

Epanalepsis

Definition 8 (Epanalepsis)**.** *Repetition at the end of a clause of the word or phrase that began it. Formally,*

$$\langle [W]_a \dots [W]_a \rangle$$

$$\langle \langle P \rangle_a \dots \langle P \rangle_a \rangle$$

Example 9. *Romans, countrymen, and lovers! hear me for my cause, and be silent, that you may hear: believe me for mine honour, and have respect to mine honour, that you may believe.*[11]

Farnsworth (2011) compares the effect of epanalepsis to "circuitry," in that the second instance of the word or phrase finishes an incomplete thought about it. Corbett (1990) notes that epanalepsis is "rare in prose," likely because its scheme of repetition typically results from such depth of emotion as only poetry can adequately hold.

In searching for epanalepsis, we ignore leading determiners, conjunctions, and prepositions in the comparison subsequences that start clauses.

[11] *Julius Caesar* 3.2.

The example has two instances of epanalepsis: the repetitions of *hear* and of *believe*.

Anadiplosis

Definition 9 (Anadiplosis)**.** *Repetition of the ending word or phrase from the previous clause at the beginning of the next. Formally,*

$$\langle \ldots [W]_a \rangle \langle [W]_a \ldots \rangle$$
$$\langle \ldots \langle P \rangle_a \rangle \langle \langle P \rangle_a \ldots \rangle$$

Example 10. *For this very reason, you must make every effort to support your faith with goodness, and goodness with knowledge, and knowledge with self-control, and self-control with endurance, and endurance with godliness, and godliness with mutual affection, and mutual affection with love.*[12]

Lanham (1991) observes that anadiplosis can also create *climax*, the "[m]ounting by degrees through linked words or phrases, usually of increasing weight and in parallel construction," as in the example.

In searching for anadiplosis, we ignore leading determiners, conjunctions, and prepositions in the comparison subsequences that start clauses.

Antimetabole

Definition 10 (Antimetabole)**.** *Repetition of words in reverse order. Formally,*

$$[W]_a \ldots [W]_b \ldots [W]_b \ldots [W]_a$$

Example 11. *"Beauty is truth, truth beauty,"—that is all Ye know on earth, and all*

[12] 2 Pet. 1:5–7, *NRSV*.

ye need to know.[13]

Example 12. *Woe unto them that call evil good, and good evil; that put darkness for light, and light for darkness; that put bitter for sweet, and sweet for bitter!*[14]

In some rhetorical primers, antimetabole is subsumed under *chiasmus* (q.v.; e.g. Farnsworth, 2011; Meynet, 2012) or synonymous with it (e.g. Lanham, 1991; Fahnestock, 1999; Murphy, Katula, Hill, & Ochs, 2003); but following Corbett (1990), we have separately defined antimetabole as the reversal of words, and chiasmus as the reversal of grammatical structure. While the two figures have some potential overlap, there are specific instances of reversal that would comprise only one figure or the other.

In searching for antimetabole, we consider only words that the parser has tagged as nouns, verbs, adjectives, or adverbs, within a default window of 1 sentence. The search includes two phases: first, we find all single-word repetitions in the search window; then we check whether or not each pair of repetitions $[W]_a \ldots [W]_a$, $[W]_b \ldots [W]_b$ forms the positional pattern $[W]_a \ldots [W]_b \ldots [W]_b \ldots [W]_a$ in the window (with any number of intervening words between the individual components).

[13] Keats.
[14] Isa. 5:20.

Polyptoton

Definition 11 (Polyptoton)**.** *Repetition of a word in a different form; having cognate words in close proximity. Formally,*

$$[S_a\{M_a\}]\ldots[S_a\{M_b\}]$$

$$[\{M_a\}S_a]\ldots[S_a\{M_b\}]$$

$$[S_a\{M_a\}]\ldots[\{M_b\}S_a]$$

$$[\{M\}S_a\{M\}]\ldots[S_a]$$

$$[S_a\{M\}]\ldots[S_a]$$

$$\textit{etc.}$$

Example 13. *Judge not, that ye be not judged.*[15]

Example 14. *The prophecy was that I should be dismembered; and—Aye! I lost this leg. I now prophesy that I will dismember my dismemberer.*[16]

Example 15. *The Greeks are strong, and skillful to their strength, fierce to their skill, and to their fierceness valiant;...*[17]

Polyptoton is the repetition of derivationally related forms of words; that is, it also allows for morphological similarity between words rather than just strict equality, as shown in the previous examples.

WordNet (Miller, 1995; Fellbaum, 1998, 2006) "is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory" (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990). Nouns, verbs, adjectives, and adverbs are grouped into so-called synonym sets or *synsets*, each having a common semantic concept similar to those typically found in thesauri; more important for the

[15] Mt. 7:1.
[16] *Moby-Dick*.
[17] *Troilus & Cressida* 1.1.

detection of polyptoton, WordNet also provides links from each word in a synset to others lexically related to it, and more specifically to those derivationally related.

Algorithm 2 presents an outline for finding derivationally related forms of a word. The algorithm combines WordNet's synsets and lexical relations, and *affix stemming*, to find a word's derivational forms. First, common prefixes and suffixes in English (**Appendix C**) are added to the word, and the existence of the resulting words is checked in the WordNet lexicon; existing ones get added to a collection of related word forms. Then the Porter stem (Porter, 1997) of the word is run through the same check, and existing forms added to the collection of related forms. The Porter stemmer does well with removing suffixes, less so with prefixes; we have attempted to improve its usefulness in this context by checking the stem for common prefixes that might have escaped stemming. Though the addition and removal of affixes to words—as well as the the Porter stemming itself—can create some false-positive related words, these do not adversely affect the algorithm results in any serious way. After collecting the related forms, the algorithm avails itself of WordNet's lexical relations by finding the derivational forms of every word in the collection, stores those, and then checks the existence of their affix-augmented versions in WordNet. All of the related and derivational words then become a single collection of words derivationally related to the original search word.

Example 16. *Derivationally related forms of the word* value *via* ***Algorithm 2****: value, devalue, overvalue, revalue, undervalue, valuable, valued, valuer, valuation, valueless, values, evaluate, valuate, devaluation, overvaluation, revaluation, undervaluation, valuableness, valuelessness, devalued, devaluate, invaluable, invaluableness, reevaluate, unvalued.*

Once the Rhetorica software has found all the derivationally related forms of a word, it can check for instances of those within the polyptoton search window (default 3 sentences); note that according to **Algorithm 2**, a repetition of the original search

Algorithm 2 Finding derivationally related forms of a word w with WordNet (Fellbaum, 2006).

```
 1: function FindDerivationallyRelatedForms(w)
 2:     Create empty collection R for related word forms
 3:     Add word w to R
 4:     for all common prefixes P_i (Appendix C) do
 5:         if word w starts with P_i then
 6:             if w − P_i exists in WordNet then
 7:                 add w − P_i to R
 8:             end if
 9:         else
10:             if P_i + w exists in WordNet then
11:                 add P_i + w to R
12:             end if
13:         end if
14:     end for
15:     Repeat ll. 4–14 for all common suffixes S_i (with w + S_i for P_i + w)
16:     Repeat ll. 4–15 for Porter stem (Porter, 1997) s in place of w
17:     Create empty collection D for derivationally related word forms
18:     for all R_i in R do
19:         Find derived forms of R_i in WordNet and add them to D
20:     end for
21:     for all D_i in D do
22:         for all common prefixes P_i do
23:             if P_i + D_i exists in WordNet then
24:                 add P_i + D_i to R
25:             end if
26:         end for
27:         for all common suffixes S_i do
28:             if D_i + S_i exists in WordNet then
29:                 add D_i + S_i to R
30:             end if
31:         end for
32:     end for
33:     return merge(R, D)
34: end function
```

word will count as an instance of polyptoton. In searching for polyptoton, we ignore stop words (**Appendix B**).

Schemes—Figures of Parallelism

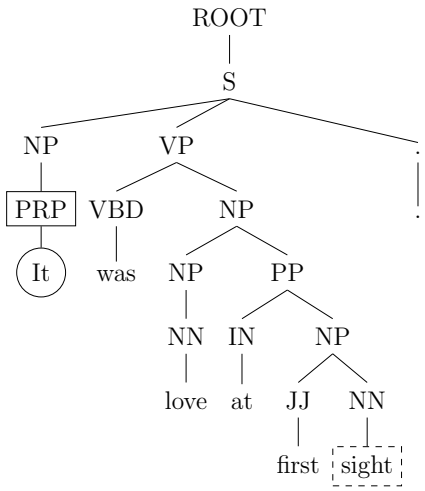
The principle of parallelism in grammar and rhetoric "demands that equivalent things be set forth in co-ordinate grammatical structures" (Corbett, 1990). Like parts of speech are matched with like for the sake of aesthetic coherence. A rhetorical use of parallelism is to specify or enumerate series of similar things.

Isocolon

Definition 12 (Isocolon)**.** *Repetition of grammatical structure in nearby phrases or clauses of approximately equal length.*

We searched for figures of parallelism within successive sentences as with figures of repetition; the figures were now matched, though, on their tokens' parts of speech as determined by the parser, and in the case of isocolon specifically, also adjudged equivalent on some minimal difference in measures of phrasal distance.

Figure 3: Parse tree for the sentence "It was love at first sight"[18].



[18]Heller, *Catch-22*.

In comparing two phrases, we consider their POS tags and their *height*. For a parse tree as in **Figure 3**, the sentence "It was love at first sight," the parser provides both the *depth* of the entire tree and the *distance* between any two nodes; the depth is the direct path—the count of intervening nodes + 1—from the root node to any of the most distant leaf nodes, and the distance between the root node and any other node is similarly calculated. In **Figure 3**, the (entire) tree depth from the root node S to the dashed box is 6; for the arbitrary circled token *It*, the distance to its preterminal POS node (solid boxed) is 2. We define the *height* of *It*'s POS node, *PRP*, as the difference between the tree depth and the root–*PRP* distance, i.e. $6 - 2 = 4$, which is the level of the *PRP* node above the bottom of the parse tree. Rhetorica uses this height calculation of POS nodes to compare the similarity of subtrees within a search window.

Table 3: POS Tag Equivalence Classes

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

As for the POS tags, instead of comparing them directly, we first assign them to broader *equivalence classes* (Gawryjołek, 2009), then compare those. **Table 3** lists sets of POS tags and their equivalence classes, which represent major parts of speech.

Gawryjołek's implementation of isocolon discovery uses POS equivalence classes and node height (though called *depth* in Gawryjołek 2009) within parse trees to calculate a *distance* or difference (**Algorithm 3**[19]) between every pair of phrases within a several-sentence search window, and we have adopted that methodology here. If

[19]The method `MaximumWordTagOverlap` is represented in Gawryjołek (2009) by a somewhat convoluted algorithm that appears to return the length of the longest common subsequence (Wikipedia, 2012) of POS equivalence classes between the two phrases. Our code uses a standard minimum-edit distance (Levenshtein, 1966) algorithm instead.

Algorithm 3 Find the difference between two phrases. (Adapted from Gawryjołek, 2009.)

```
1: function FindPhraseDifference(p1, p2)
2:     Construct list of leaf-label elements l1, l2 for phrases p1, p2
3:     Initialize distance d between phrases to abs (len l1 − len l2)
4:     for i = 0 to min (len l1, len l2) do
5:         e1_i ← ith element from l1
6:         e2_i ← ith element from l2
7:         if e1_i and e2_i have same labels and height then
8:             continue
9:         else
10:            d = max (len l1, len l2) − MaximumWordTagOverlap(l1, l2)
11:            return d
12:        end if
13:    end for
14:    return d
15: end function
```

all the equivalence classes and heights are the same between two phrases, then the phrases are equal with zero distance and comprise, along with any other mutually zero-distance phrases within the search window, an example of isocolon. If the phrases are not strictly equal, but the distance between them is considered small enough, they are effectively equal; otherwise, they are not equal structurally and cannot represent isocolon.

The method `MaximumWordTagOverlap` in **Algorithm 3** provides a switch to calculate either the minimum-edit distance (Levenshtein, 1966) or the longest common subsequence between the sets of POS equivalence classes representing a pair of phrases to compare; in practice, those two methods of finding overlap always returned the same answer.

Finally, in addition to ducking a maximum phrase-difference threshold, two phrases under comparison for isocolon must start with identical POS equivalence classes, and also end with identical classes. In testing, this requirement reduced the number of found isocolons that arguably looked like false positives.

Chiasmus

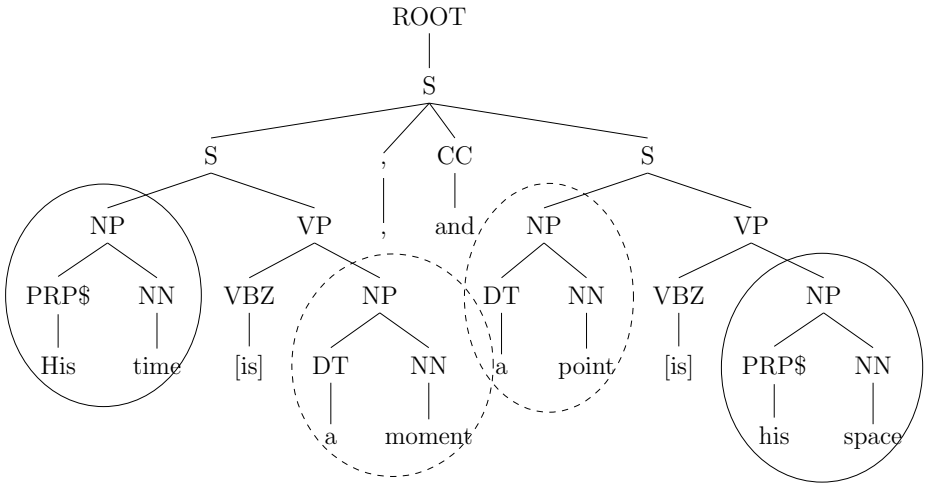
Definition 13 (Chiasmus)**.** *Repetition of grammatical structures in reverse order.*

Example 17. *[I]n his face, divine compassion visibly appeared: Love without end, and without measure, Grace,...*[20]

Example 18. *What counts is not necessarily the size of the dog in the fight—it's the size of the fight in the dog.*[21]

Chiasmus is similar to antimetabole (q.v.), but following Corbett (1990), we have separately defined antimetabole as the reversal of words, and chiasmus as the reversal of grammatical structure.

Figure 4: Parse tree for the phrase "His time a moment, and a point his space."[22]



The heart of chiasmus detection is the identification of *prepreterminal* phrases within sentence parse trees. A prepreterminal node has children that are all *preterminals*, whose children are in turn terminal leaf nodes. The reason we try to find prepreterminal phrases is that they represent atomic units vis-á-vis chiasmus; that is, in looking for the *ABBA* pattern of chiasmus, we do not want to work with collections

[20] *Paradise Lost.*
[21] Dwight D. Eisenhower.
[22] Pope, *An Essay on Man.*

any smaller than the prepreterminal phrase. The reason for this restriction becomes apparent in the example of **Figure 4**, the parse tree of the phrase "His time [is] a moment, and a point [is] his space." In looking for reversed subtrees of **Figure 4** within the search window, any collection of higher granularity than prepreterminal typically becomes nonsensical in English: e.g. the POS-tag pattern DT + NN—determiner + noun—of "a moment" reverses into the very low-probability pattern NN + DT, in which the determiner *follows* the noun. Therefore, to find instances of chiasmus, we first identify prepreterminals (with their subtrees intact), then check for reversal of pairs of them within the search window. **Figure 4** shows a pair of prepreterminal phrases, circled with solid and dashed lines respectively, and also their corresponding reversal, matched on the POS tags of the leaf nodes. In fact the matching is slightly looser than exact POS tags, instead comparing the POS-tag equivalence classes (v. **Table 3**) between phrases.

Tropes

A *trope* is a deviation from the normal signification of words in speech or writing. The discovery of most tropes is beyond the scope of our research, but a combination of WordNet (Miller, 1995) and the Stanford Parser's typed dependencies (De Marneffe, MacCartney, & Manning, 2006) allows detection of simple forms of oxymoron.

Oxymoron

Definition 14 (Oxymoron)**.** *A terse paradox; the yoking of two contradictory terms.*

Example 19. *Jumbo shrimp. Original copy. Open secret. Seriously funny. Foolish wisdom. Deafening silence.*

Example 20. *No light, but rather darkness visible / Served only to discover sights of woe,. . .*[23]

[23] *Paradise Lost.*

Example 21. *O miserable abundance, O beggarly riches!*[24]

Example 22. *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.*[25]

Corbett (1990) describes successful oxymoron as yolking contradictory ideas to "startling effect." While human minds can readily identify effective oxymoron, computationally the same identification is difficult. The problems include determining what syntactic collections of words can form an oxymoron, and how exactly to quantify the semantic discord of a "condensed paradox" (Lanham, 1991).

The latter problem sent us again to WordNet (Miller, 1995), which links its database of words together through the lexical relation of (inter alia) antonymy; this is cruder than actual *paradox*, but provides a simulacrum of it. Because the WordNet search is computationally costly, the former problem of allowed syntactic relationships in oxymoron now came to the forefront; if we could appropriately limit the search for oxymoron, we would both save computing time and minimize false-positive detection.

Table 4: Possible Typed Dependencies (De Marneffe, MacCartney, & Manning, 2006) Leading to Oxymoron

Example	Dependency(*gov*, *dev*)	Description
jumbo shrimp	amod(shrimp, jumbo)	adjectival modifier
seriously funny	advmod(funny, seriously)	adverb modifier
waxes small	acomp(waxes, small)	adjectival complement
even the odds	dobj(even, odds)	direct object
fair is foul	nsubj(foul, fair)	nominal subject
feather of lead	prep_*prep*(feather, lead)[26]	collapsed prepositional modifier

[24] John Donne.

[25] *Romeo & Juliet 1.1.*

[26] Where *prep* in this case is *of*, but generally comes from among common prepositions such as *by*, *for*, *from*, *to*, *with*, etc.

Gawryjołek (2009) lit upon the idea of using the Stanford Parser's *typed dependencies* to limit the number of word/phrase pairs searched for oxymoron. Typed dependencies "provide a simple description of the grammatical relationships in a sentence" (De Marneffe et al., 2006); the binary dependencies map onto a directed graph in which words are nodes and grammatical relations are edges. After evaluating 49 expressions containing de facto oxymorons, Gawryjołek narrowed down the list of possible grammatical relations between the *governing* and *dependent* parts of an oxymoron; these are summarized in **Table 4**, with examples. Our Rhetorica software's detection of oxymoron begins with finding all the dependencies in the one-sentence search window.

Once the dependencies have been collected, we can invoke WordNet's antonymy relations to check the pairs for the inherent contradiction necessary for oxymoron. However, WordNet's antonymy relations are somewhat sparse, insofar as antonym sets for particular words typically have few elements, without associated derivational forms; therefore, we must also incorporate other lexical relations to expand WordNet's antonymy relations somewhat.

To expand WordNet antonymy for use in oxymoron detection, we augment it by also following other lexical relations; all the relations we need are:

- *Antynomy.* Though the definition "The antonym of a word x is sometimes *not-x*, but not always" (Miller et al., 1990) perhaps too glibly describes a complex relationship between words, most English-speakers readily recognize the antonyms that they encounter as incompatible pairs. WordNet provides antonymy as a lexical relation between word forms.
- *Synonymy.* Miller et al. (1990) espouses a weak version of Leibniz's definition of synonymy: "[T]wo expressions are synonymous in a linguistic context C if the substitution of one for the other in C does not alter the truth value" of an utterance wherein the substitution is made. WordNet provides extensive

synonym sets (AKA *synsets*) as its most important lexical relation between words.

- *Derived Forms.* WordNet provides derivationally related forms of words, such as those for *value* in **Example 16**, e.g. *devalue*, *overvalue*, *evaluate*, etc. Properly filtered, these allow us to extend the antonymy relation.

Figure 5: Search tree to find an optimal collection of antonymy relations in WordNet for oxymoron detection, with actions at each node performed by the method `WordNetRelationVisitor`. Dashed nodes are preceded by an antonym search and must account for the reversal of lexical polarity.

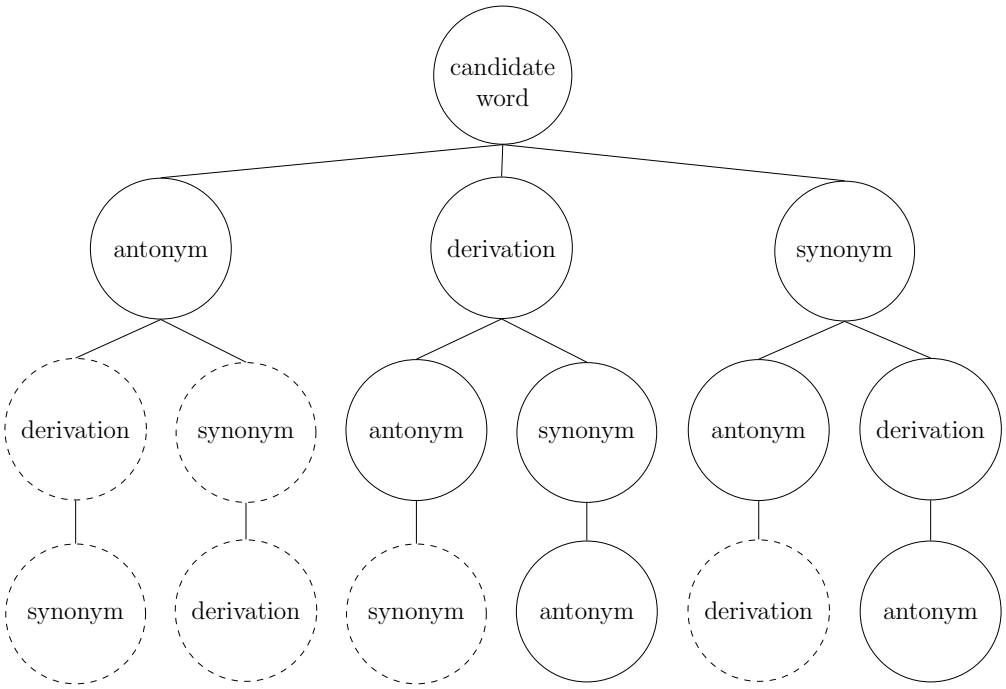


Figure 5 shows a binary search tree representing all the permutations of the set of lexical relations {antonym, derivation, synonym} to exploit. Starting at the top of the tree with a candidate word, we run a pre-order traversal of the tree; at each node, we search WordNet to collect the specified relation set for the candidate, then store it at the node. At nodes below the first level, the candidate word becomes a candidate set of the parent node's related words. The dashed circles in **Figure 5** represent nodes that must account for previous antonymy in their parent nodes. Generally that means not merging the parent node's related-word set with the current node's related

words; specifically for *derivation* nodes, it means adding negation prefixes (*anti-*, *de-*, *dis-*, *in-*, *im-*, *il-*, *ir-*, *mis-*, *non-*, *un-*) to the candidate word in hope of finding even more relevant antonymic derivational forms. We use the permutation ordering to squeeze as much antonymy out of WordNet as possible: traversing each branch of the tree results, in general, in different antonym sets for the candidate word. Once all the antonym sets for the candidate have been found, they are merged, and purged of duplicates.

Example 23. *Antonymically related forms of the word* cold *via the search tree of* ***Figure 5****: live, alive, living, near, conscious, enthusiastic, sentient, humane, loving, mild, fresh, original, warm, heat, passionate, vernal, summery, autumnal, equatorial, cooked, hotness, hot.*

Example 23 shows a collection of antonyms of the word *cold* that would result from the tree-traversal of **Figure 5**. In contrast, WordNet's lexical relations alone provide only the following antonyms of *cold*: *hot*, *hotness*.

Algorithm 4, which uses the antonym search tree of **Figure 5**, outlines oxymoron discovery for word pairs that are grammatically related according to the typed dependencies of **Table 4**. If the algorithm deems a word pair to be an oxymoron, then the pair and any intermediate words are finally stored as an example of oxymoron in the search window.

Resources

The only resources required for this research were time, moderate computing power, some development tools, and publicly available corpora for testing. All these resources were readily available.

Algorithm 4 Detection of oxymoron in a grammatically dependent word pair.

```
 1: function DetectOxymoron(p)
 2:     p_1 ← first element from pair p
 3:     p_2 ← second element from pair p
 4:     Create empty collection A for antonymically related word forms of p_2
 5:     for all permutations of lexical relations {antonym, derivation, synonym} do
 6:         for all nodes in corresponding Figure 5 tree-branch do
 7:             Store WordNet results from WordNetRelationVisitor(p_2) at node
 8:         end for
 9:         Add to A the antonymic relations resulting from branch traversal
10:     end for
11:     if intersection(p_1, A) is true then
12:         return true
13:     else
14:         if elements of pair p are in original order {p_1, p_2} then
15:             p' ← {p_2, p_1}
16:             Repeat ll. 2–18 for p'
17:         end if
18:         return false
19:     end if
20: end function
```

Rhetorica

Rhetorica is an application for finding rhetorical figures in text, with natural similarities to Gawryjołek's JANTOR (Java ANnotation Tool Of Rhetoric). Written in C# ($\sim$ 4000 lines of code) on top of Microsoft's .NET platform, it initializes itself with the following libraries and tools:

- IKVM (to allow Java libraries in .NET, http://www.ikvm.net/)
- OpenNLP (Baldridge et al., 2002; http://incubator.apache.org/opennlp/)
- The Stanford PCFG (probabilistic context-free grammar) Parser (Klein & Manning, 2003)
- An implementation of the Porter Stemmer (Porter, 1997)
- The WordNet database (Miller, 1995)

After initialization the text under analysis is loaded, all its sentences are detected; then each sentence is parsed, tokenized, and then broken into phrases derived from

the parse tree. In addition to the parser-derived phrases, any group of words between medial punctuation marks is considered a phrase and included.

Once the text's rhetorical figures are discovered and tagged in a separate CSV file, that file can provide relevant summary statistics to use in a SVM model.

R

R is an "integrated suite of software facilities for data manipulation, calculation and graphical display" designed around a true computer language (R Development Core Team, 2012). It is an open-source GNU (The Free Software Foundation, 2012) project which is similar to the S language and environment developed at Bell Laboratories. (S-PLUS is a commercial version of S.) R provides many functions and packages implementing statistical techniques, and in fact many of its users consider R to be primarily a statistics system.

Through its add-on packages, R provides interfaces for support vector machine libraries (packages **e1071** and **kernlab**) and Weka (Frank et al., 2010; package **RWeka**).

Corpora

The body of written material that we used in classification tasks is described in § *Corpora and Tasks*.

Summary

This chapter presents an outline of how we find rhetorical figures in text and what we use them for. Our Rhetorica software summarizes the textual figures it finds and uses that summary for several authorship-attribution tasks. The various rhetorical figures are presented formally and algorithmically, and the details of Rhetorica are discussed.

Chapter 4

Results

Figure Detection

For each of the 14 rhetorical figures described in § *Figure Detection*, we created a file with at least 25 examples of that figure as culled from the Bible, literature, political speeches, popular culture, and common sayings and clichés, with assistance from rhetoric texts and Web sites such as Quinn (1982), Lanham (1991), Corbett and Connors (1998), Fahnestock (1999), Crowley and Hawhee (2004), Burton (2007), and Farnsworth (2011). Whenever possible, the examples were left in the context of full sentences to more accurately simulate finding them *in situ*.

Table 5: Precision and Recall Tests of the Rhetorica Software

Figure	Total No.	f_{++}*	f_{+-}*	f_{-+}*	Misparse†	Prec. (%)	Recall (%)
Epizeuxis	42	42	0	0	0 (0)	100.0	100.0
Ploce	56	56	0	0	0 (0)	100.0	100.0
Conduplicatio	25	25	0	0	0 (0)	100.0	100.0
Polysyndeton	28	28	0	0	0 (0)	100.0	100.0
Anaphora	29	29	0	0	0 (0)	100.0	100.0
Epistrophe	42	42	2	0	0 (0)	95.0	100.0
Symploce	66	64	0	2	2 (2)	100.0	97.0
Epanalepsis	29	29	1	0	0 (0)	97.0	100.0
Anadiplosis	42	41	0	1	0 (0)	100.0	97.6
Antimetabole	25	25	10	0	0 (0)	71.0	100.0
Polyptoton	50	45	2	5	0 (0)	96.0	90.0
Isocolon	62	50	4	12	13 (12)	92.6	80.6
Chiasmus	33	14	14	19	19 (12)	50.0	42.4
Oxymoron	49	16	1	33	5 (5)	94.0	33.0

* f_{++}: true positives; f_{+-}: false positives; f_{-+}: false negatives.
† Total parser errors leading to false positives and negatives, with false negatives in parentheses.

The following sections describe the results of running these example files through our Rhetorica software, as summarized in **Table 5**. The ideas of precision/recall as

expressed there are similar to those described in § *Definition of Terms*, and derive from counts of *true positives*, *false positives*, and *false negatives* of potential figures among the examples:

- *Precision.* The total number of examples of rhetorical figures correctly identified, divided by the total number of figures tested. Mathematically, the estimated precision prec is

$$\text{prec} = \frac{f_{++}}{f_{++} + f_{+-}},$$

  where f_{++} is the total number of correctly identified figures (true positives) and f_{+-} is the total number of figures *misidentified* as the same figure (false positives). *High* precision, a measure of exactness, means that many more figures were correctly identified than misidentified.

- *Recall.* The total number of examples of rhetorical figures correctly identified, divided by the total number of figures that *should* have been identified. Mathematically, the estimated recall rec is

$$\text{rec} = \frac{f_{++}}{f_{++} + f_{-+}},$$

  where f_{++} is the total number of correctly identified figures (true positives) and f_{-+} is the total number of figures *not identified* as the same figure (false negatives). *High* recall, a measure of completeness, means that most of the figures were correctly identified.

We will also discuss errors in detection that happened.

Epizeuxis

Epizeuxis (to recall) is repetition of a word or phrase with no others between. The epizeuxis test file contained 42 examples that we had identified as true epizeuxis. Epizeuxis detection is straightforward (see § *Epizeuxis* for details), and Rhetorica

Table 6: Common Characteristics of Figure-Detection Methods in Rhetorica

Figure	Max. Unit*	Restrictions†	Window‡¶	Extra (*default*)§¶
Epizeuxis	sentence	—	2	—
Ploce	word	no stop words	2	—
Conduplicatio	sentence	not all stop words; not contiguous	2	minimum length (2)
Polysyndeton	word	—	2	consecutive starts (2)
Anaphora	sentence	no boundary determiners, conjunctions, prepositions	3	—
Epistrophe	sentence	no boundary determiners, conjunctions, prepositions	3	—
Symploce	sentence	no boundary conjunctions	3	minimum length (2)
Epanalepsis	phrase	no start determiners, conjunctions, prepositions	—	—
Anadiplosis	sentence	no determiners, conjunctions, prepositions	2	—
Antimetabole	words‖	only nouns, verbs, adjectives, adverbs	1	—
Polyptoton	word	no stop words	3	—
Isocolon	sentence	—	3	similarity threshold (1)
Chiasmus	prepreterminal phrase	—	3	minimum length (3)
Oxymoron	typed dependency phrase	only some typed dependencies	1	greedy (false)

* The largest (or only) syntactic unit that can be a repeated constituent of the rhetorical figure.
† These determine which POS types or grammatical relations are recognized or ignored in figure detection. (*Boundary*, *start*, and *end* refer to token positions within a syntactic unit.)
‡ The size of the sliding search window, in number of sentences, where the figure is searched for.
§ Extra parameter in the detection method for configurability; described in § *Figure Detection*.
¶ These method parameters can be changed from the default on the command line.
‖ These can be separated by any number of intervening words not composing the figure.

correctly identified all the examples, including an inadvertently augmented one:

Example 24. *Why should a dog, a horse, a rat have life, / And thou no breath at all? Thou'lt come no more, / Never, never, never, never!*[1]

[1] *King Lear* 5.3.

Example 25. *Never give in, never, never, never—in nothing, great or small, large or petty—never give in except to convictions of honour and good sense.*[2]

In the test file, **Example 24** was immediately followed by **Example 25**, so the terminal *never* of the former abutted on the leading *never* of the latter, which added another term to the already discovered *never*×4 epizeuxis of **Example 24**. Although this resulting epizeuxis was unexpected, Rhetorica nonetheless correctly found it; we do not consider it a false positive because of the linsey-woolsey nature of the test file, which gathered up disparate, self-contained examples of epizeuxis without considering possible rhetorical connections among them (unlike a more unified text).

The default search window size for epizeuxis is 2 sentences. (The search window mechanism is described in § *Schemes—Figures of Repetition*.)

Ploce

Ploce is the repetition of a word in a short span of text for rhetorical emphasis. The ploce test file had 56 examples of true ploce, and Rhetorica correctly identified all of them.

The default search window for ploce is 2 sentences. The configurable defaults and common attributes of all the figure-detection methods in Rhetorica are summarized in **Table 6**; the defaults can be changed through the command-line interface to Rhetorica. The relatively small default search window could miss legitimate instances of ploce in short, staccato sentences; while the default seems mostly appropriate for contemporary English prose and poetry, it should be tweaked for exceptionally short or long average sentence lengths.

[2] Churchill.

Conduplicatio

Conduplicatio is the repetition of a word or phrase; broader than ploce. The conduplicatio test file had 25 examples of true conduplicatio, and Rhetorica correctly identified all of them. One restriction we have on conduplicatio is that it cannot consist entirely of stop words (v. **Appendix 10**); another is that the figure's constituent phrases not be contiguous.

The default search window for conduplicatio is 2 sentences, which can be changed. The conduplicatio-detection method in Rhetorica also takes a "minimum length" parameter which puts a lower limit on the size of the constituent, repeated phrase in the figure; the default minimum length is 2 words. Lanham (1991) defines *epimone* as the "[f]requent repetition of a phrase or question, in order to dwell on a point"; historically, distinguishing our conduplicatio from epimone may not be easy, but we sense that epimone draws itself out a bit more than conduplicatio, so its detection in Rhetorica might happen by setting the minimum length to 3+.

Polysyndeton

Polysyndeton is overabundant repetition of conjunctions between clauses. The polysyndeton test file had 28 examples of true polysyndeton, and Rhetorica correctly identified all of them.

Example 26. 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.*[3]

As pointed out in § *Polysyndeton*, the search window for polysyndeton is a single sentence, but we also look for the same conjunction beginning two consecutive

[3]Gen. 1:24–25.

sentences. In the example above from the test file, Rhetorica found three instances of polysyndeton: the words in reverse italics (*and*×4) in the first sentence, those in boldface (*and*×4) in the second, and also the underlined, leading *and*s (×2) of each sentence. The polyptoton-detection method in Rhetorica takes a "consecutive starts" parameter specifying the minimum number of repeated sentence-leading conjunctions that can compose an instance of polysyndeton; the default is 2.

Anaphora

Anaphora is the repetition of a word or phrase at the beginning of successive clauses. The anaphora test file had 29 examples of true anaphora, and Rhetorica correctly identified all of them.

Example 27. *But madmen never meet. It is the only thing* they cannot *do.* *They can talk, they can inspire, they can fight, they can found religions;* ***but*** they cannot *meet.*[4]

To decrease the number of false-negative and incomplete anaphoras, Rhetorica ignores leading determiners, conjunctions, and prepositions in the comparison subsequences. In **Example 27**, the six underlined instances of *they can* compose one of the passage's anaphoras; note that *but* starts the clause containing the last instance, but ignoring this leading conjunction allows Rhetorica to find the complete anaphora. (Note that the tokenizer of the Stanford Parser splits the single word *cannot* into two separate tokens, *can* + *not*, to reflect its historical disjunction as modal + negative particle.) The other anaphora in the passage is *they cannot*×2, in reverse italics in the example.

Single-word phrases from which leading determiners, conjunctions, and prepositions have been removed, and which qualify more as epistrophe than anaphora,

[4]Chesterton, *A Miscellany of Men.*

sometimes enter the anaphora search window; Rhetorica recognizes these phrases and removes them from the search.

Example 28. *This royal throne of kings, this sceptred isle, / This earth of majesty, this seat of Mars, / This other Eden, demi-paradise, / This fortress built by Nature for herself / Against infection and the hand of war, / This happy breed of men, this little world, / This precious stone set in the silver sea, / Which serves it in the office of a wall, / Or as a moat defensive to a house, / Against the envy of less happier lands; / This blessed plot, this earth, this realm, this England, / This nurse, this teeming womb of royal kings / This land of such dear souls, this dear dear land, / Dear for her reputation through the world, / Is now leased out—I die pronouncing it— / Like to a tenement or pelting farm.*[5]

In **Example 28**, the underlined *this*×17 clearly qualifies as an anaphora, but gets rejected by Rhetorica's prohibition of certain leading stop words (as described above). The trade-off was difficult, but we feel that the prohibition defeats more false positives than it saves false negatives; we currently have no solution except perhaps to limit which leading determiners get rejected, or to make special allowances for words repeated excessively in a short span.

Epistrophe

Epistrophe is the repetition of the same word or phrase at the end of successive clauses. The epistrophe test file had 42 examples of true epistrophe, and Rhetorica correctly identified all of them, but also 2 false positives.

Example 29. *Fraud is indeed un-English; and dissimulation, and deception, and duplicity, and double-dealing, and promise-breaking, all, every vice akin to these vile things are indeed un-English; but tyranny, base, abominable tyranny, is un-English;*

[5] *Richard II* 2.1.

hard-hearted persecution of poor fanatic wretches is un-English;. . .[6]

The promiscuity of clause detection described in the anaphora results comes back to bite us here: in hope of picking up clauses left undetected by the parser, we also allow groups of words between some medial punctuation marks; this leads to some false clauses. In **Example 29**, Rhetorica has identified *but tyranny* and *abominable tyranny* as clauses, then picked out the underlined *tyranny*s as an epistrophe.

Single-word phrases from which trailing prepositions have been removed sometimes enter the epistrophe search window; Rhetorica recognizes these phrases, which qualify more as anaphora than epistrophe, and removes them from the search.

Symploce

Symploce is the repetition of a word or phrase at the beginning, and of another at the end, of successive clauses. The symploce test file had 66 examples of true symploce, and Rhetorica correctly identified 64 of them, with 2 false negatives.

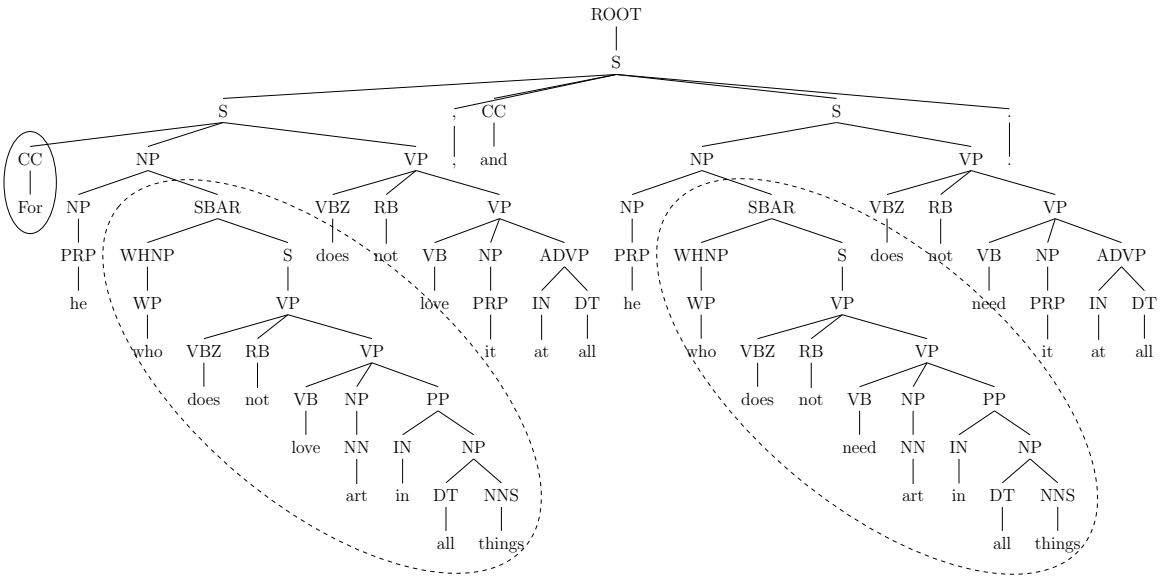
Example 30. *For he* ***who does not*** *love* ***art in all things*** *does not love it at all, and he* ***who does not*** *need* ***art in all things*** *does not need it at all.*[7]

Example 30 contains two symploces: "he who does not . . . it at all"×2 and "who does not . . . art in all things"×2, underlined and boldfaced, respectively, in the example text. Rhetorica, however, doesn't pick up either of them because of the parse tree for the sentence returned by the Stanford Parser.

Figure 6 shows the parse for **Example 30**, in which the leading *for* (solid ellipse), meaning *because*, has been correctly identified as a conjunction. With *for* as a conjunction, the parser thereafter identifies two S sentences, each with an SBAR subordinate clause (dashed ellipse), all of which types have exactly the same parse trees down to their preterminal nodes, and which become clauses that Rhetorica would

[6]Sheil, speech in the House of Commons (1843).
[7]Wilde, *The English Renaissance of Art.*

Figure 6: Correct parse tree for the sentence "For he who does not love art in all things . . . "



evaluate for symploce within its search window (default 3 sentences). Because each SBAR- and S-pair has the same grammatical structure and encompasses the same tags, Rhetorica would correctly identify its symploce.

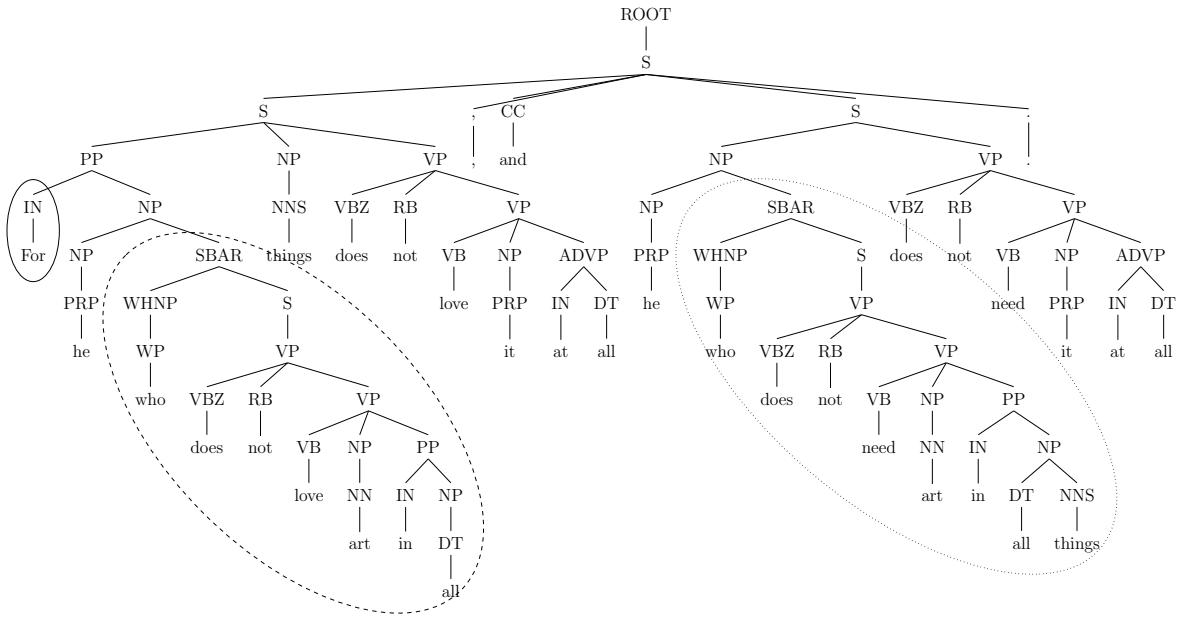
Figure 7 is, however, the parse actually returned by the Stanford Parser. It has incorrectly identified *for* as a preposition (IN, solid ellipse), which cascades into a misparsing of the first SBAR subordinate clause (dashed ellipse) such that its parse tree does not match that of the second subordinate clause (dotted ellipse), nor do the S sentences match each other. Therefore Rhetorica misses both symploces because of the misparsing. Fortunately this kind of misparsing happens infrequently; we can do little to prevent it, and we rely on the probabilistic nature of the parser.

Example 31. *Against yourself you are calling him, against the laws you are calling him, against the democratic constitution you are calling him.*[8]

Note that ignoring leading prepositions would not be a good solution, possibly causing excessive false negatives by missing symploces like that in **Example 31**.

[8] Aeschines, *Against Ctesiphon*

Figure 7: Incorrect parse tree for the sentence "For he who does not love art in all things . . . "



Epanalepsis

Epanalepsis is the repetition at the end of a clause of the word or phrase that began it. The epanalepsis test file had 29 examples of true epanalepsis, and Rhetorica correctly identified all of them, but also 1 false positive.

Example 32. *Don John Conmee walked and moved in times of yore. He was humane and honoured there. He bore in mind secrets confessed and he smiled at smiling noble faces in a beeswaxed drawingroom, ceiled with full fruit clusters. And the hands of a bride and bridegroom, noble to noble, were impalmed by Don John Conmee.*[9]

In **Example 32**, the repetition of *Don John Conmee* is sometimes considered epanalepsis on the paragraph- or verse- or pericope level. Since Rhetorica searches for epanalepsis only within individual clauses, we put the example in the epanalepsis test file with the expectation of its being passed over; however, the underlined words, *noble* ... *noble*, came out as a false-positive epanalepsis. The error derives from

[9]Joyce, *Ulysses*.

allowing groups of words between some medial punctuation marks to be clauses, which infrequently leads to false clauses, as in this case.

Anadiplosis

Anadiplosis is the repetition of the ending word or phrase from the previous clause at the beginning of the next. The anadiplosis test file had 42 examples of true anadiplosis, and Rhetorica correctly identified 41 of them, with 1 false negative.

Example 33. *The bill, therefore, was lost. It was lost in the House of Representatives. It died there, and there its remains are to be found.*[10]

Example 33 contains a pair of underlined phrases that would almost universally be considered an anadiplosis, but Rhetorica failed to find it. This is a consequence of trying to minimize false positives and false negatives simultaneously: Rhetorica does ignore all determiners, conjunctions, and prepositions in its search for anadiplosis (v. **Table 6**), but turning a blind eye on all stop words (v. **Appendix B**) ignores too much; because the pronoun *it* intervenes between the pair of *was lost*s that otherwise ends and starts the successive clauses, and Rhetorica does not ignore *it*, the anadiplosis itself is lost.

Antimetabole

Antimetabole is the repetition of words in reverse order. The antimetabole test file had 25 examples of true antimetabole, and Rhetorica correctly identified all 25 of them, but also 10 false positives.

Example 34. *Good judgment* comes *from experience and experience* comes *from bad judgment.*[11]

[10] Webster, speech in the Senate (1836).
[11] An old saw.

Example 34 underlines the antimetabole we want to identify, but in addition to that one, Rhetorica also finds:

judgment ... comes | comes ... judgment

comes ... experience | experience ... comes

The problem of false positives in antimetabole discovery derives primarily from a lack of sentiment analysis. Which repeated words are actually rhetorically emphatic? But sentiment analysis is beyond the scope of this study, so we found a more concrete way to limit false positives, considering only words that the parser has tagged as nouns, verbs, adjectives, or adverbs. That limitation helps, but not with **Example 34**, which has still provided us with three separate antimetaboles. A possible solution is to disallow word overlap among candidate antimetaboles; but then which candidate should have priority? Another possibility—should we only match verbs to verbs, nouns to nouns, etc.? No, because then we would miss gems like this one:

Example 35. *If you can't be with the one you love, love the one you're with.*[12]

Another idea is to provide a task-specific switch as the extra argument to the antimetabole-detection method in Rhetorica: for `specific:false`, the method could choose just one of overlapping antimetaboles to count for classification tasks where exact identification of the figure is not so important as its well-counted existence. For now, though, we accept the trade-off of high recall (most—*all* in our test file—figures correctly identified) offset by lower precision, i.e. less exactness because of the false positives.

Example 36. *"Beauty is truth, truth beauty,"—that is all Ye know on earth, and all ye need to know.*[13]

Example 36 contains a famous antimetabole, whose constituents are underlined. What Rhetorica found as an antimetabole, though, was—

[12]Stephen Stills.
[13]Keats.

Beauty is ... truth | truth ... beauty is

—which is a consequence of limiting the search to nouns, verbs, adjectives, and adverbs. Rhetorica ignored the intermediate *that* originally between the latter *beauty is* in the example, so the antimetabole was overdecorated with the copula. The unnecessary augmentation might be prevented by limiting antimetabole discovery to the reversal of word pairs only, but then we would miss *phrasal* antimetaboles like:

Example 37. *I can run faster than anyone who can run for longer, and I can run for longer than anyone who can run faster.*[14]

so finally we kept the phrasal discovery.

Polyptoton

Polyptoton is the repetition of a word in a different form; having cognate words in close proximity. The polyptoton test file had 50 examples of true polyptoton, and Rhetorica correctly identified 45 of them, with 5 false negatives, and 2 false positives.

Example 38. *Our knights are thinking only of the money they will make in ransoms: it is not kill or be killed with them, but* ***pay*** *or be* ***paid****.*[15]

Example 38 contains one of the false negatives, the boldface pair *pay ... paid*. Polyptoton discovery in Rhetorica works by running each element of the candidate pair through **Algorithm 2** to collect all its derivationally related forms; then the two collections are tested for intersection, the truth of which determines whether the candidate words comprise a polyptoton. For **Example 38**, the collections of derivationally related forms are as follows:

Example 38a. *Derivationally related forms of the word* pay *via* ***Algorithm 2****: pay, overpay, prepay, repay, underpay, payable, payer, paying, payment, payee, overpayment, prepayment, repayment, underpayment, repayable, payables, nonpayment ...*

[14]Internet example, provenance unknown (or *o.o.o.*—"of obscure origin").
[15]Shaw, *Saint Joan*.

Example 38b. *Derivationally related forms of the word* paid *via* ***Algorithm 2****: paid, prepaid, unpaid.*

It surprised us that WordNet did not provide overlapping sets of derivationally related words in this case; consequently, Rhetorica missed the *pay . . . paid* polyptoton. A possible augmentation of the search could include irregular verb forms, but we suspect that that might make the results noisier with false positives. On the other hand, **Algorithm 2** should be considered a mutable drop-in in Rhetorica, and a smarter stemmer with smarter prefix trimming might substantially improve polyptoton performance.

Example 39. *But here I only remark the interesting fact that the conquered almost always conquer. Sparta killed Athens with a final blow, and she was born again. Sparta went away victorious, and died slowly of her own wounds.*[16]

Such is the crime, and such is the criminal, which it is my duty in this debate to expose, and, by the blessing of God, this duty shall be done completely to the end.[17]

The puzzling polyptoton in **Example 39** requires a little more explanation.

Example 39a. *Derivationally related forms of the word* died *via* ***Algorithm 2****: died, dial, dior, dis, disable, diss, di.*

Example 39b. *Derivationally related forms of the word* end *via* ***Algorithm 2****: end, d, emend, ended, ending, endive, endless, endlessness, dal, dial, ded, emended, den, ding, emendation, dive, dative, endlessly, ds, des, dy, red, unended, unending.*

Example 39 shows the results of a low-probability confluence of spurious derivationally related forms to produce a false-positive polyptoton. **Example 39a** lists the derivationally related forms produced by running the word *died* through **Algorithm 2**; likewise **Example 39b** for the word *end*. Both lists share the derivation-

[16] Chesterton, *The Giant*.
[17] Sumner, speech in the Senate (1856).

ally *un*related—to either word—form *dial* (underlined), and both contain other words clearly unrelated to their sources.

In **Example 39a**, the problem of spurious derivationally related forms results from improper stemming of the word *died*. In § *Polyptoton*, we noted that the Porter stemmer (Porter, 1997) does well with removing suffixes, but less so with prefixes; for the preterite *died*, though, the stemmer lopped off the suffix -*ed* to leave the stem *di*, to which **Algorithm 2** then added the prefixes and suffixes of **Appendix C**, producing some legitmate words that are nevertheless completely unrelated to *died*, and among them, *dial*.

In **Example 39b**, the stemmer did just fine (*end* → *end*), but **Algorithm 2** had failed to recognize the atomicity of the word *end*, and removed the alleged prefix *en*- from it; then the algorithm tacked on the suffixes from **Appendix C** to the remainder *d* to produce *dial* and several other words derivationally unrelated to *end*, but which nevertheless exist in WordNet.

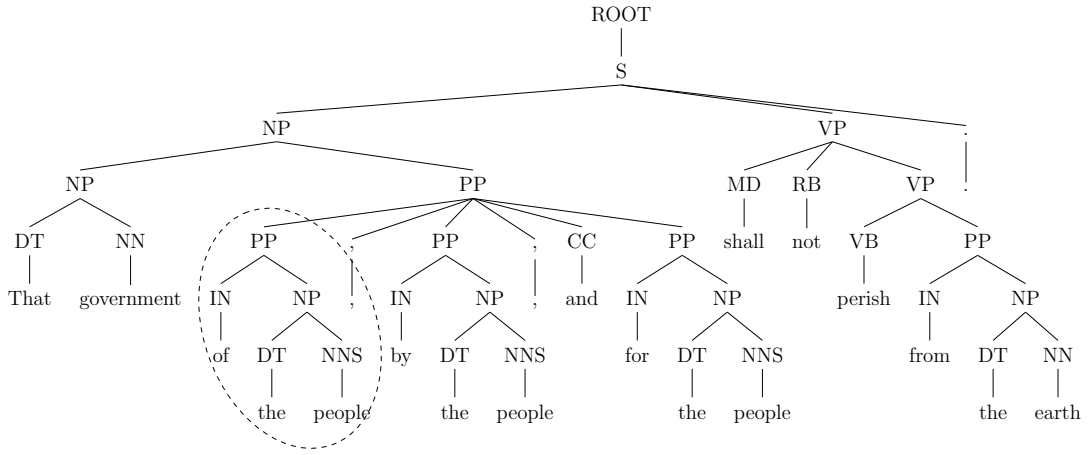
Because the lists of **Examples 39a and 39b** intersect at *dial*, Rhetorica incorrectly labeled *died* ... *end* as a polyptoton. One way to avoid this type of false positive is the rejection of any stems created by the Porter stemmer or **Algorithm 2** of less than 3 characters; but since the error did not happen much in the "real" texts processed by Rhetorica (probably because we disallow stop words as candidates), we opted to leave the potential of some false positives in hope of recognizing the more true positives.

Isocolon

Isocolon is the repetition of grammatical structure in nearby phrases or clauses of approximately equal length. The isocolon test file had 62 examples of true isocolon, and Rhetorica correctly identified 50 of them, with 12 false negatives and 4 false positives.

Example 40. *. . . that government of the people, by the people, and for the people shall not perish from the earth.*[18]

Figure 8: Correct parse tree for the sentence ". . . that government of the people, by the people, and for the people shall not perish from the earth."

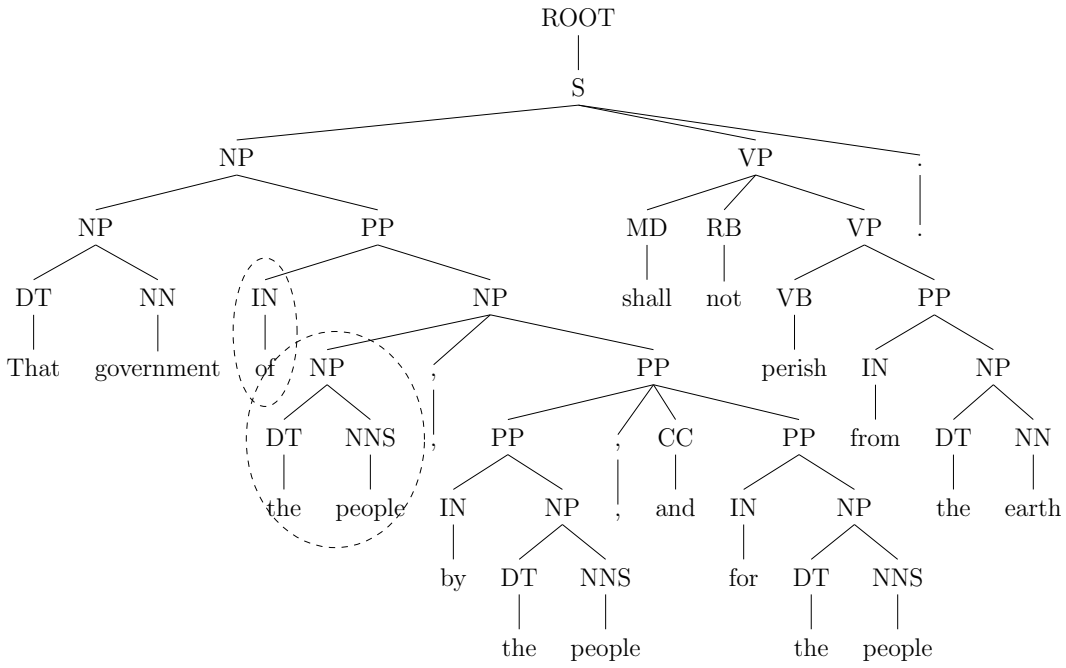


Rhetorica only partially identified the isocolon underlined in **Example 40**, missing the first underlined phrase. Rhetorica's isocolon detection depends heavily on the Stanford Parser's (Klein & Manning, 2003) parse trees, which here failed to return a correct parse. **Figure 8** shows the correct parse for **Example 40**; from it, Rhetorica could invoke **Algorithm 3** to detect all the underlined phrases as equaling one another within the appointed phrase-difference threshold. The missing phrase, "of the people," is within a dashed circle, and is clearly on par structurally with the other prepositional phrases (PP) within its encompassing compound PP, and also on par with the PP under the adjoining verb phrase that comprises the sentence predicate.

The actual parse returned by the Stanford Parser, though, is shown in **Figure 9**. It splits the missing phrase (dashed lines) under an inaccurate hierarchy that subordinates the phrase's coordinate PP's; hence, the discovered isocolon is incomplete because of the incorrect parse.

[18]Lincoln, Gettysburg Address (1863).

Figure 9: Incorrect parse tree for the sentence "... that government of the people, by the people, and for the people shall not perish from the earth."



Example 41. *The joy of battle comes after the first fear of death; the joy of reading Virgil comes after the bore of learning him; the glow of the sea-bather comes after the icy shock of the sea bath; and the success of the marriage comes after the failure of the honeymoon.*[19]

Example 42. *Then saith he unto them, Render therefore unto Caesar the things which are Caesar's; and unto God the things that are God's.*[20]

Underlined in **Examples 41 and 42** are two phrases comprising a false-positive isocolon. In this case the parser has incorrectly tagged *unto* as a VBG, a gerund/present participle, instead of as a preposition (IN); the parse of each phrase is shown in **Figure 10**, with subfigure **10b** showing the incorrect POS tag in a dashed box. Because each phrase has the same grammatical structure and tags, Rhetorica labeled the phrases together as an isocolon.

[19]Chesterton, *What's Wrong with the World* (1910).
[20]Mt. 22:21.

Figure 10: Example of false-positive isocolon detection.

(a) **Example 41** *(b)* **Example 42**

Parser-induced hiccups like those in **Figure 9** and **Figure 10b** can cause both false positives and false negatives as shown, but false positives are rarer, so Rhetorica's isocolon discovery has better precision than recall; i.e. Rhetorica infrequently misidentifies isocolons, but it misses some of them.

Chiasmus

Chiasmus is the repetition of grammatical structures in reverse order. The chiasmus test file had 33 examples of true chiasmus, and Rhetorica correctly identified 14 of them, with 19 false negatives and 14 false positives. No two ways about it, Rhetorica's chiasmus detection was somewhat bad, with just okay precision and poor recall. We will discuss why and consider future improvements.

Chiasmus detection is discussed in § *Chiasmus*: its basis is the identification of "atomic" prepreterminal nodes in the parse tree, whose reversal in a short span of text comprises chiasmus.

Example 43. *. . . Whom Joys with soft varieties invite,* / *By day the frolic, and the dance by night, . . .*[21]

Example 43 presents an augmented version of an ostensibly false-negative chiasmus from our test file; in fact we tested only the second line, "By day the frolic . . . ", which out of context prevented discovery of its own chiasmus, as we describe below.

[21] Johnson, *The Vanity of Human Wishes*.

Figure 11: Incorrect parse tree for the phrase "... By day the frolic, and the dance by night, ..."

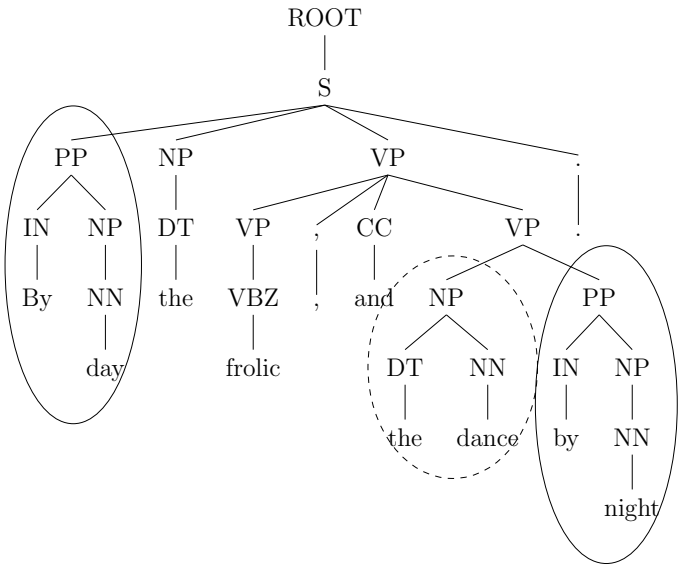
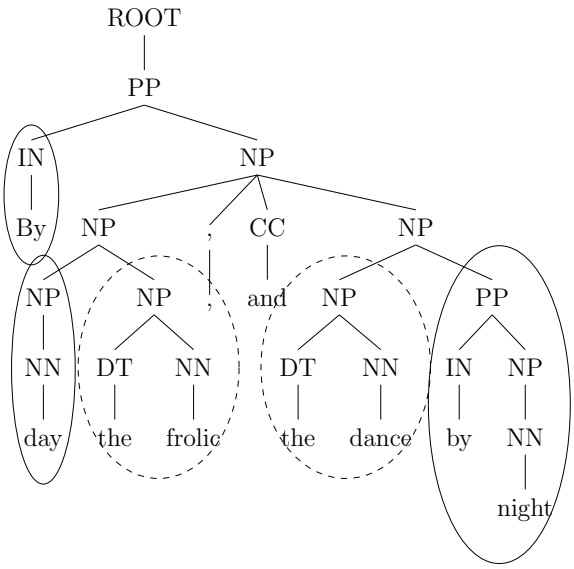


Figure 11 shows a botched parse of our example phrase, the parse returned as Rhetorica worked its way through our test file. The internal phrase *the frolic* no longer matches its prepreterminal counterpart *by night*, and the chiasmus is lost. Our habit here has been to frankly acknowledge the shortcomings of the Stanford Parser that stymie Rhetorica's figure detection; but this time the blame is mostly ours, not the parser's. In the gallimaufry of the chiasmus test file, we had to fit in the single line from Doctor Johnson while keeping it separate from nearby, unrelated material, so we ended it with a period; that was a mistake, because our efficient, Procrustean sentence detection took the line out of its original poetic context as an appositive phrase, and forced it into a (thereafter badly parsed) sentence.

One solution is to end such a fragment with an ellipsis, like so: *By day the frolic, and the dance by night...*, and the parser will handle it better; an even stronger solution is to keep more surrounding material from the same text to provide grammatical context—which we uncharacteristically neglected to do here. This sort of error is much less likely to happen in "real" texts, which our chiasmus test file is not (it contains numerous quotes and bon mots whose original context is lost), but shows

Figure 12: Another incorrect parse tree for the phrase "... By day the frolic, and the dance by night, ..."



the utter reliance of Rhetorica on correct parsing for chiasmus detection.

When we had prevented the parser, as just described, from parsing **Example 43** incorrectly as a full sentence, and then Rhetorica identified its chiasmus, we declared victory—but a short-lived one when we saw the new parse tree of **Figure 12**. In this case, Rhetorica is enjoined to join lone prepositions (IN) or infinitival *to*s (TO) to the immediately following phrase—so *By day* (each word separately solidly circled) becomes a single prepreterminal candidate phrase, and the whole chiasmus includes the other circles, too. But the chiasmus is fortuitous, because we consider **Figure 12** a bad parse that nevertheless led to a correct identification.

Figure 13 shows the correct parse for the second line of **Example 43**. From it Rhetorica would find the chiasmus; but our parser was not trained on poetic inversions like this one, and found them difficult to parse correctly.

Example 44. *Some* ***have an idea*** *that the reason we in this country discard things so readily is because we have so much.* ***The facts are*** *exactly opposite—the reason we have so much is* simply *because we discard things so readily.*[22]

[22] Alfred P. Sloan, attributed.

Figure 13: Correct parse tree for the phrase "... By day the frolic, and the dance by night, ..."

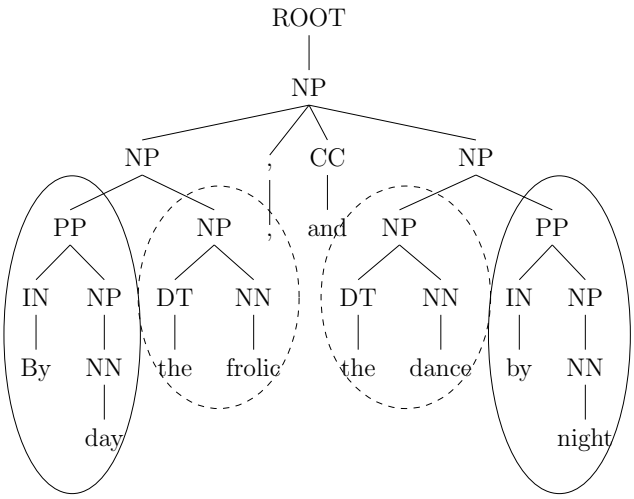
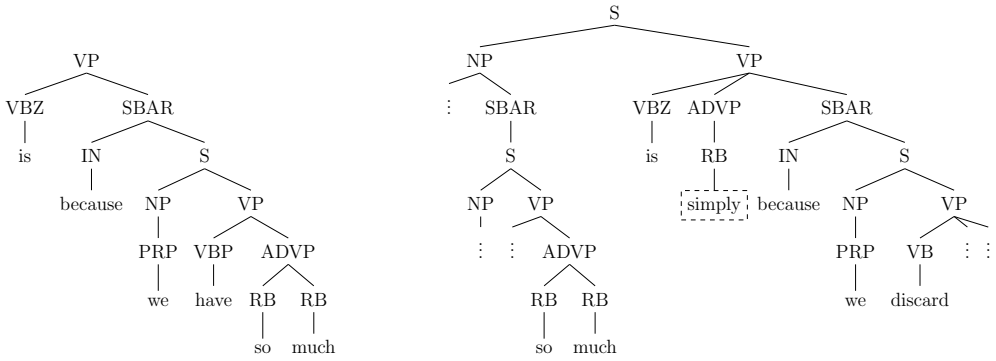


Figure 14: Example of chiasmus missed by Rhetorica.



(a) **Example 44**, left phrase

(b) **Example 44**, right phrase

Some chiasmi were missed because of misparsing, but others because of Rhetorica's inflexibility in matching tag sequences. **Example 44** provides a quote from which we would hope to minimally extract the underlined chiasmus, and **Figure 14** shows the parse trees for the constituent phrases. The catch is *simply* (dashed box) in subfigure **14b**, which interferes with the exact reversal of the prepreterminal phrases in the *is because ... so much* | *so much ... we discard* pair of POS-tag collections; without *simply*, Rhetorica finds the chiasmus. This example, as well as others wherein e.g. a simple verb negation prevented chiasmus detection, argues for allowing some

fuzziness in phrase "equality," as in isocolon's **Algorithm 3** for comparing phrases with non-identical POS tags.

The false positives in chiasmus detection—undeliberate chiasmus—in both the test file and in unsynthetic texts come mainly from an overreach of the search window, or from accidental grammatical reversal, as in the boldfaced phrases in **Example 44**, which we display here with each word's POS tag:

have/VBP an/DT idea/NN
The/DT facts/NNS are/VBP

From our aforementioned "rules" about prepreterminal phrases, and by equating POS tags in the same equivalence class (**Table 3**), the phrases *have an idea* and *The facts are* comprise an instance of chiasmus; but is it a chiasmus in spirit? Farnsworth (2011) suggests some characteristics of an effective chiasmus:

- The reversal of grammatical structure reinforces an actual reversal or reciprocity of meaning.
- The chiasmus sounds *convincing* by creating a definitive, finished language edifice.
- The reversal of sound and structure is attractive and memorable.

Frankly, our so-called chiasmus *have an idea ... The facts are* has none of those characteristics, which is why we deem it a false positive despite its grammatical reversal. We have no ready solution for rejecting such false positives; perhaps creating a new chiasmus + antimetabole (*structural antimetabole*?) figure in Rhetorica, with allowance for derivational forms, would help trim false positives, but would it still allow too many false negatives?

Dubremetz and Nivre (2015), hot off the presses, proposes and tests a method of evaluating chiasmus as a "graded phenomenon" through techniques from information retrieval. The authors report an enviable 61% precision, with recall similar to Rhetorica's; their success may suggest future upgrades to Rhetorica's chiasmus detection.

Also, we could try using a different grammar with the Stanford Parser. For this study, we used the file `englishPCFG.ser.gz`, an unlexicalized PCFG grammar. In general, PCFG grammars use less memory and run faster than the alternative factored grammars, which include lexicalization, but the factored grammars usually have better accuracy; in English specifically, though, both kinds have comparable accuracy, so we preferred `englishPCFG.ser.gz` for its speed. But we could instead try the factored grammar `englishFactored.ser.gz` for its marginal improvements; or, given more copious computing time, we could try the RNN (recurrent neural network) model `englishRNN.ser.gz`, which provides maximum accuracy (Lu, 2014).

It is even possible to train the parser using collections of syntactically annotated data (Stanford NLP Group, 2014) that specifically suit the corpora to be parsed (e.g. poetry, newspaper, tweets, etc.), but such training is outside the scope of this study.

Oxymoron

Oxymoron is a terse paradox; the yoking of two contradictory terms. The oxymoron test file had 49 examples of true oxymoron, and Rhetorica correctly identified 16 of them, with 33 false negatives and 1 false positive.

Example 45. *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.*[23]

Example 45 (recycled from § *Tropes*) is perfect for showing the successes and failures of Rhetorica's oxymoron detection. All the candidate oxymorons are underlined. Only one tweak is available through the `FindOxymoron` method, `greedy:false` (the default) or `true`, which tells WordNet to perform a shallow or deep search, respectively, of derivational forms at each node of **Figure 5**; as ever, the risk of expanded

[23] *Romeo & Juliet 1.1.*

lexical relations is more potential false positives. For `greedy:false`, Rhetorica finds the following oxymorons in **Example 45**:

loving hate, heavy lightness, cold fire, sick health

or 4/11 of the candidates, without any false positives. For `greedy:true`, Rhetorica finds:

*loving hate, *first create, heavy lightness, serious vanity, bright smoke, cold fire, sick health, *love feel, *feel love*

or 6/11 of the candidates, but with 3 asterisked false positives. In other words, the greedy search produces more true positives, but at the cost of more false positives. As with polyptoton, the success of oxymoron detection depends on the ability of **Algorithm 2** to return accurate derivational forms.

Example 46. *The shackles of an old love straitened him, / His honour rooted in dishonour stood, / And faith unfaithful kept him falsely true.*[24]

Failure of the Stanford Parser's typed-dependency relationships also creates the possibility of missed oxymorons, i.e. false negatives. Recall that **Table 4** lists the allowed binary dependencies that mitigate a brute-force search of all word pairs; if the parser misidentifies a pair's dependency, then the pair might go untested for oxymoron. In **Example 46**, we begin with two known oxymorons, which are underlined. Rhetorica has no trouble finding the latter one, *falsely true*, but it misses the former, *faith/NN unfaithful/JJ*—where we have also shown each word's correct POS tag. According to **Table 4**, the pair should show up among found dependencies as the *amod*(faith, unfaithful) relationship. Unfortunately, though, the parser actually tagged the phrase thus:

faith/NN unfaithful/NN

[24]Tennyson, *Idylls of the King: Lancelot and Elaine.*

so the correct dependency was lost, and Rhetorica never tested the pair as a candidate oxymoron.

Overall, Rhetorica's oxymoron discovery has much better precision than recall; i.e. Rhetorica infrequently misidentifies oxymorons, but it misses a lot of them.

Authorship Attribution

In **Chapter 3**, we described some corpora and our plans for performing authorship-attribution tasks using Rhetorica's output from them (§ *Corpora and Tasks*). This section summarizes the results of those tasks.

We began by breaking up large documents in each corpus into manageable chunks. Hirst and Feiguina (2007) has identified a block size of 1000 words as minimally optimal for training classifiers on English prose, so we aimed for approximately that length while still preserving sentence boundaries, which are essential to Rhetorica's figure detection.

For each corpus, we trained SVM (support vector machine) attribution models to discriminate the authorship of test documents from among those in the training set, using as feature sets lexical vectors, rhetorical vectors, and combined lexical + rhetorical vectors.

Each lexical vector held the frequency (/1000 words) of a particular word in the training set, for every separate document in the set. The words considered were the 12 most frequent in the training set; unsurprisingly, they were always function words, whose frequencies rank among the best features for author discrimination (see e.g. Burrows, 1987; Argamon & Levitan, 2005).

Each rhetorical vector held the frequency (/1000 words) of a particular rhetorical figure among the 14 identified by our Rhetorica software. The figures considered were 12 of the 14: anadiplosis, anaphora, antimetabole, conduplicatio, epanalepsis, epistrophe, epizeuxis, isocolon, ploce, polyptoton, polysyndeton, and symploce. Be-

cause chiasmus and oxymoron have poor recall (and chiasmus also poor precision; v. **Table 5**), we left both of them out of the attribution tasks.

The lexical + rhetorical vectors simply combined the two collections of vectors just described, to see whether their combination performed better as a feature set than each individual collection.

Table 7 summarizes the results of our authorship-attribution tasks, which we discuss in greater detail in the following sections. The table shows the test documents for each corpus, the actual author of each document, and the author classified by each of the three attribution models. More important, though, are the model accuracies for each corpus listed in the rightmost columns of the table. The accuracies come from ten-fold cross-validation to assess model performance, which partitions the training data into subsets that successively validate the model trained on the other, complementary subsets. Cross-validation estimates the accuracy of a predictive model without "wasting" any of its training data on validation (Harrell, 2001).

Note that generally the model accuracies are similar for the lexical and rhetorical models separately, with some improvement when those feature sets are combined. In the following discussions we will use abbreviations of the various models: $\mathcal{L}$ = attribution model trained on lexical features only; $\mathcal{R}$ = attribution model trained on rhetorical features only; $\mathcal{LR}$ = model trained on combined lexical + rhetorical features.

The Federalist *Papers*

Of the 85 *Federalist* papers, 12 of them are "disputed" because of contradictory authorship claims by both Hamilton and Madison. Beginning with Mosteller and Wallace (1964), computational authorship identification has typically chosen Madison as the author of all 12 disputed papers, though sometimes doubt about individual papers arises (see e.g. Savoy, 2013).

Table 7: Results of Authorship Attribution Based on Lexical and Rhetorical Counts

Corpus			Pred. Author[‡]			Model Accuracy[§]		
	training set[*]	author[†]	lex.	rhet.	lex.+rhet.	lex.	rhet.	lex.+rhet.
Federalist papers	126					84.0%	82.1%	92.5%
No. 49		Madison	Mad.	Ham.	Ham.			
No. 50		Madison	Mad.	Ham.	Mad.			
No. 51		Madison	Mad.	Mad.	Mad.			
No. 52		Madison	Mad.	Mad.	Mad.			
No. 53		Madison	Mad.	Mad.	Mad.			
No. 54		Madison	Mad.	Mad.	Mad.			
No. 55		Madison	Ham.	Ham.	Ham.			
No. 56		Madison	Ham.	Mad.	Mad.			
No. 57		Madison	Mad.	Ham.	Ham.			
No. 58		Madison	Mad.	Ham.	Ham.			
No. 62		Madison	Mad.	Mad.	Mad.			
No. 63		Madison	Mad.	Mad.	Mad.			
Juola C	296					74.0%	65.5%	79.7%
Sample 01		C3	C3	C3	C3			
Sample 02		C1	C1	C1	C1			
Sample 03		C1	C1	C1	C1			
Sample 04		C4	C4	C4	C4			
Sample 05		C5	C5	C2	C5			
Sample 06		C2	C2	C1	C2			
Sample 07		C4	C4	C4	C4			
Sample 08		C5	C2	C5	C5			
Sample 09		C2	C2	C2	C2			
Juola D	328					74.1%	75.9%	80.8%
Sample 01		D1	D1	D1	D1			
Sample 02		D2	D3	D3	D3			
Sample 03		—	D1	D1	D1			
Sample 04		D3	D3	D1	D3			
Juola E	328					75.0%	75.6%	80.8%
Sample 01		E3	E3	E3	E3			
Sample 02		E1	E3	E1	E1			
Sample 03		—	E3	E3	E3			
Sample 04		E2	E3	E3	E3			
Juola G	436					71.6%	60.1%	68.1%
Sample 01		G2	G2	G1	G2			
Sample 02		G1	G2	G2	G2			
Sample 03		G2	G2	G2	G2			
Sample 04		G1	G2	G2	G1			
Juola H	13					76.9%	76.9%	76.9%
Sample 01		H3	H2	H2	H2			
Sample 02		H1	H2	H2	H2			
Sample 03		H2	H2	H2	H2			
Brontës	439					86.8%	85.2%	88.1%

[*] The number of documents in the training set. Each document is $\sim$ 1000 words or less.
[†] The true (or assumed true) author of the test file. Note that *Ham.* abbreviates "Hamilton."
[‡] Test file author predicted by each feature-set model. *lex.* = "lexical"; *rhet.* = "rhetorical."
[§] Average accuracy of each training model derived from ten-fold cross-validation.

Table 7 shows how well the attribution models performed at choosing the author of each disputed paper: $\mathcal{L}$ assigned 2 papers to Hamilton and the rest to Madison; $\mathcal{R}$ assigned 5 to Hamilton; and $\mathcal{LR}$, 4 to Hamilton. It might seem prima facie that neither $\mathcal{R}$ nor $\mathcal{LR}$ did particularly well at this attribution task, but the model accuracies suggest that on average $\mathcal{L}$ and $\mathcal{R}$ perform similarly well; and $\mathcal{LR}$ has an average success rate of 93%, which suggests that authorship of the disputed papers should be quite predictable from the authors' use of function words and rhetorical figures combined.

Other tests of $\mathcal{L}$ using—instead of just the 12—the 100 and 200 most-frequent words in the training set, improved the model's accuracy to the level of $\mathcal{LR}$; but at $N_{fw} = 200$, $\mathcal{L}$ assigned all the disputed papers to Madison (whereas $\mathcal{LR}$ assigned some to Hamilton). Although exploring the sources of the discrepancy interests us, it is beyond the scope of this study; furthermore, the black-box nature of SVM models makes their interpretation difficult and obscures what drives the classification (Baayen, 2008).

Exemplary Corpora of Juola et al. (2006)

We describe the Juola corpora and attribution tasks in § *Corpora and Tasks*. Of the given Problems A–M, our ability to complete them was limited by a modern-English-only requirement coming from the Stanford Parser's integration into Rhetorica—which cut the problem set down to A–E, G, H. Even some of these corpora resisted the creation of SVM classification models, though. **Table 7** summarizes the results of the finally remaining Juola corpora and tasks, and we provide further commentary below.

- *Problem A* (Fixed-topic student essays) The training set of 38 short documents provided feature sets inadequate to train SVM classification models to return anything but random results, so we dropped this problem.
- *Problem B* (Free-topic student essays) This problem used the same training set

as Problem A, resulting in the same inadequate models, so it was also dropped.

- *Problem C* (19th-century American novels) Both $\mathcal{L}$ and $\mathcal{R}$ did reasonably well at the attribution tasks (1 and 2 misclassifications, respectively), and $\mathcal{LR}$ matched all the authors correctly, with a model accuracy of 80%.

- *Problems D & E* (Elizabethan/Jacobean plays) These problems used the same training set but different test sets, with the twist that each third test document came from an author outside the training set (and who should be so identified). For both problems, $\mathcal{LR}$ improved slightly on the other models by correctly matching 2 of 4 authors, with a model accuracy of 81%. The Stanford Parser, trained on Modern English prose, had some trouble with the Early Modern English prose + verse of these plays, which inflated both the false positives and negatives in figure discovery, and led to poorer rhetorical feature sets.

- *Problem G* (E. R. Burroughs' early and late novels) Though $\mathcal{LR}$ correctly matched 3 test documents as *early* or *late*, all three models were afflicted with overfitting, as shown by their low accuracy rates. $\mathcal{L}$ had quite a poor accuracy of 60% for this training corpus; and the rhetorical feature set, for the only time in all our attribution tasks, gave $\mathcal{LR}$ worse accuracy than $\mathcal{L}$. We suppose that Burroughs must have written similarly in all phases of his career, making this a particularly difficult attribution task.

- *Problem H* (Unrestricted corporate speech) Like Problems A and B, this one probably had too small a training set to train an effective SVM classification model, but all the model accuracies were around 77%.

Works of Charlotte and Anne Brontë

Because Hirst and Feiguina (2007) did not suggest any specific authorship-attribution tasks related to the novels of Charlotte and Anne Brontë, but instead concerned themselves only with model accuracy, we have followed their lead. $\mathcal{LR}$, with a model accuracy of 88%, improves slightly upon $\mathcal{L}$ and $\mathcal{R}$. As with the *Federalist* and Juola C, $\mathcal{LR}$ once again performed very well on 19th-century prose.

Text Characterization

In § *Epanalepsis*, we quoted Corbett (1990) as saying that epanalepsis is "rare in prose," likely because its scheme of repetition typically results from such depth of

emotion as only poetry can adequately hold. Corbett implies that epanalepsis should appear more frequently in poetry than in prose.

Without being able to prove it, we could still test this assertion in reasonably large corpora of temporally similar prose and poetry from the Brontë sisters. In § *Works of Charlotte and Anne Brontë* we looked at a model to distinguish the novels of Charlotte Brontë from those of her sister Anne; now we considered the two sisters' novels together as a Brontë "prose corpus." As the Brontë "poetry corpus," we used Charlotte, Emily, and Anne's 1846 collection of poems (Brontë, Brontë, & Brontë, 1846), which they published under the pen names of Currer, Ellis, and Acton Bell, respectively.

Example 47. *Thought followed thought, star followed star, / Through boundless regions, on; / While one sweet influence, near and far, / Thrilled through, and proved us one!*[25]

Example 48. *Her reply–not given till after a pause–evinced one of those unexpected turns of temper peculiar to her.*[26]

Table 8: Prevalence of Epanalepsis in the Brontë Corpora

Corpus	Count (per 1000 words)
prose	1.39
poetry	0.88

Example 47 and **Example 48** show instances of epanalepsis that Rhetorica found in the Brontë poetry and prose corpora, respectively. Although epanalepsis seemed rare in both corpora, the poetry initially appeared to contain *fewer* instances of the figure (**Table 8**). However, a tally of epanalepsis in the corpora is count data,

[25] Charlotte Brontë, "Stars."
[26] Charlotte Brontë, *Villette*.

often well-modeled by the Poisson distribution, so we wanted to confirm a statistical difference in the rates of **Table 8**.

Table 9: Poisson Regression of Epanalepsis Counts between Poetry and Prose Corpora

	Dependent variable:
	Epanalepsis Counts
Corpus Type = prose	0.469* (0.090, 0.848)
Constant	−0.134 (−0.504, 0.237)
Observations	471
Log Likelihood	−729.582
Akaike Inf. Crit.	1,463.164
Note:	*p<0.05; **p<0.01; ***p<0.001

Table 9 summarizes the results of a regular Poisson regression model[27] of epanalepsis counts per document as a function of corpus type. The relevant result here is the statistically significant (at the 0.05 level) coefficient of the *prose* corpus type, whose value is 0.469 (95% confidence interval: 0.090–0.848); from that, the value $\exp(0.469) = 1.598$ shows mean epanalepsis count increasing by a factor of 1.6 (similar to that of **Table 8**) in prose documents compared to poetry from our corpora. In other words, in the Brontë corpora, epanalepsis is statistically more prevalent in the prose than the poetry—contrary to Corbett's assertion.

This analysis shows that Corbett's assertion is not universally true, but it could be true *generally*; for example, maybe the Brontës' novels were exceptionally poetic in their prose, and therefore made a poor exemplar for this epanalepsis test. We should really consider much more expansive corpora, both poetry and prose, in order to pronounce authoritatively on Corbett's assertion.

[27]We compared the model of **Table 9** with a (more complicated) zero-inflated Poisson model to account for documents with no epanalepsis, but the Vuong test (Vuong, 1989) showed the two models to be statistically indistinguishable.

Chapter 5

Conclusions, Implications, Recommendations, and Summary

The long history of the art and science of rhetoric and the ubiquity of rhetorical figuration in language inspired this study, and sparse work on the computational detection of rhetorical figures in text motivated it. Our research expanded on previous automatic figure detection by tweaking and improving detection algorithms, and adding new figures (cf. **Table 5** and **Appendix D**); it developed and tested authorship-attribution models based on a combination of lexical and rhetorical features; and it explored other possible uses of rhetorical-figure summary data. The key contribution of this study is the development of a set of programs and methods for incorporating rhetorical figures into NLP research, derived from relevant concepts in the literature and based on current technology.

The focus of this study is the NLP-based detection and application of classical rhetorical figures. The study aimed to answer four research questions:

1. Can previous work on the automatic detection of rhetorical figures be augmented enough to make their summary statistics useful in NLP?

2. What measures of the detected figures are useful?

3. Although syntactic measures alone tend to perform worse than lexical measures in classification tasks, do combined lexical and rhetorical features improve classification models?

4. What are other potentially fruitful uses of measuring rhetorical figures in text?

To answer these questions we developed our Rhetorica software from historical and NLP sources, and applied it to real texts, then took the results to classification models and other tasks. Our approaches to these challenges are described in **Chapter 3**, and their results in **Chapter 4**. Next, we summarize our answers.

Question 1. Can previous work on the automatic detection of rhetorical figures be augmented enough to make their summary statistics useful in NLP?

Inspired by Gawryjołek (2009), we developed Rhetorica not only to find that study's 11 rhetorical figures, but also 3 more for a total of 14: anadiplosis, anaphora, antimetabole, *chiasmus*, *conduplicatio*, epanalepsis, epistrophe, epizeuxis, isocolon, oxymoron, ploce, polyptoton, polysyndeton, and *symploce*—with the new figures emphasized. We also tried to minimize false positives and negatives, and achieved some success relative to Gawryjołek's JANTOR (cf. **Table 5** and **Appendix D**), which had posted good results. Although Rhetorica's discovery of chiasmus and of oxymoron still need improvement, we deemed the other 12 figures accurate enough for NLP tasks, and used their statistics from real texts to answer the other questions.

Question 2. What measures of the detected figures are useful?

This wide question was circumscribed by the demands of the other questions, and eventually became, "Are frequencies of rhetorical figures sufficiently useful in NLP tasks?" The answer to the smaller question seems to be yes, and is taken up in the other answers below. The wider, more general question must regretfully ☹ be relegated to § *Future Work*, where we want to consider the distribution of rhetorical figures throughout clauses, sentences, paragraphs, and whole texts as auctorial markers.

Question 3. Although syntactic measures alone tend to perform worse than lexical measures in classification tasks, do combined lexical and rhetorical features improve classification models?

We took up this question in § *Authorship Attribution* with the development of authorship-classification models based on diverse corpora. In general (v. **Table 7**), models based separately on lexical measures (function words) and on rhetorical measures perform similarly, while models combining lexical and rhetorical measures together perform just a little bit better. We had hoped for razor-sharp classifications from rhetorical features but, as expected, they only helped modestly.

Question 4. What are other potentially fruitful uses of measuring rhetorical figures in text?

§ *Other Work* hints at some ambitious applications of rhetorical-figure detection, but once again the study's scope limited how far we could pursue them. We settled on a fairly concrete distinction asserted between prose and poetry: Does the depth of emotion inherent in epanalepsis make it more prevalent in poetic language? (See § *Text Characterization.*) Using distinct prose and poetry corpora of the Brontë sisters, we adduced contrary evidence, that epanalepsis might *not* obtain in poetry, because it did in their prose with statistically higher frequency—but we also acknowledged that any more general assertion about the prevalence of epanalepsis should come from rhetorical analyses of much larger corpora.

Other analyses are the domain of § *Future Work*, no matter how much we wish to consider them here.

Implications

Our research provides some exploration of the automatic discovery of rhetorical figures beyond the sparse extant literature on the topic, and looks first at the performance of rhetorical-figure features in classification models. The research might only lightly impact the study of syntactic measures in NLP because of its limitations; for example, Rhetorica's reliance on English-language parsing makes it unsuitable for language-agnostic authorship attribution (the Holy Grail of AA), and its contribution

to classification models when paired with lexical measures is modest, considering that lexical features come more easily, copiously, and flawlessly from text than rhetorical ones. On the other hand, summary measures of rhetorical figures can provide useful new quantifications of English-language texts, where we have only just started; that work, and the future work described below, hold promise that our research might find a place in NLP.

Future Work

If the automatic detection of rhetorical figures and their interpretation are interesting, then Rhetorica and this study have only scratched the surface. We can consider improvements by figure:

- *Anaphora:* Allow looser prohibition on stop words in the case of their "excessive" repetition. Consider matching repetitions on derivationally related word forms (here, and for other figures of repetition).
- *Anadiplosis:* This (and some other figures) might benefit from a fuzzier equality of phrases.
- *Epanalepsis* etc.: Clauses identified between medial punctuation (outside the parser), should be parsed out of context to confirm them as clauses.
- *Antimetabole:* Perhaps combine the word/phrase reversal of antimetabole with the structural reversal of chiasmus to optimize the rhetorical relevance of both figures.
- *Polyptoton, Oxymoron:* Limit Porter stemming of words to stems of > 2 letters to reduce false positives; but also consider altogether different, potentially more accurate techniques such as lemmatization (a kind of word-form normalization; see e.g. Korenius, Laurikkala, Järvelin, & Juhola, 2004), or matching algorithms that use a database to test candidate stems against.
- *Isocolon:* Future work might include testing some more robust measures of

phrasal distance such as those used to describe the differences between labeled XML trees (De Meyer, De Baets, & Janssens, 2001; Nierman & Jagadish, 2002; Xing, Guo, & Xia, 2007), for example, to improve accuracy.

- *Chiasmus:* An initial winnowing of candidate chiasmi by typed dependency as in § *Oxymoron* might reduce false positives.

And also general improvements:

- To test Rhetorica's discovery of figures, we used artificial test files composed of many small stretches of text containing the putative figures. While these files seemed to work well without underestimating false positives, we would rather have large, "real," figure-annotated texts for testing. The preparation of such non-artificial texts was too onerous for this study, but it remains a goal that could lead to more flexible, accurate implementations of figure discovery.
- In this study we have limited the subjectivity in each figure's definition as much as possible by using formal notation (**Table 2**). Where the combination of strict definition and figure-specific POS restrictions (**Table 6**) seemed to over-circumscribe Rhetorica's figure discovery, we have proposed just above some potential enhancements of Rhetorica to expand its discovery a little, with the intent of further minimizing both false positives and negatives.

  In general, the subjectivity is small for the figures with exact repetition of words or phrases, but larger for the syntacic (chiasmus) and semantic (oxymoron) figures, which also have the requirement of being emphatic in context (whereas the simpler figures of repetition are emphatic in the repetition itself). It might benefit future revisions of Rhetorica to have human annotators pronounce on the truth or strength of examples, within otherwise free texts, of the more subjective figures; afterwards, some measure of inter-rater agreement such as raw-agreement indices (Fleiss, 1971; Uebersax, 1983) or latent class models (Uebersax & Grove, 1990) could be evaluated and could guide development of

better algorithms for figure identification.

- Perhaps Rhetorica's figure-specific algorithms could be generalized into the discovery of *all* repetitions and inversions of a text's (or text window's) constituent words (and their lexically related forms) and phrases, followed by matching, or characterizing, patterns of their distribution within appropriate search windows defined by sentence disambiguation or parsing. While this approach might hinder labeling the repetitions as this or that classical rhetorical figure, it could yet provide broader and more relevant statistical details of repetitions within texts or corpora than Rhetorica can in its current form. Dubremetz and Nivre (2015) attempts this sort of expansion with chiasmus: the authors discover all word pairs in a text, then use various weighted features of the candidate pairs and their contextual sentences in a succession of linear models to score the pairs as chiasmus; they then identify some subset of the highest-ranking candidate pairs as true or false chiasmi. Their research suggests a potential direction for future work on Rhetorica.
- We did not explore summary measures of rhetorical figures beyond their frequencies. It is possible that their distributions within and among the language hierarchies contain author-specific information not expressed in the frequencies.
- The promise of § *Other Work* was barely realized in this study. We hope that the whole enterprise of Digital Humanities might show some interest in automatically detected rhetorical figures because of the quantification they offer.

Summary

This research has shown that the automatic detection of classical rhetorical figures, though nascent, has potential value in natural language processing. Summary statistics derived from the Rhetorica software's figure detection can characterize the style of texts and provide uncommon stylistic feature sets for classification tasks; and

while the figure frequencies that we tested seem to be only excellently adequate for authorship attribution, other measures (such as figure distributions) might perform better.

Further enhancements of Rhetorica's figure detection will improve the feature sets it provides. The success of figures of straight-up repetition (epizeuxis, anaphora, etc.) in precision and recall tests was offset somewhat by the poorer performance of the syntactic (chiasmus) and semantic (oxymoron) figures; but cannier use of Rhetorica's adjunct NLP tools (WordNet, Stanford Parser), or adopting more suitable ones, might close the gap. We continually learn from the past.

Appendix A

Penn Treebank Tag Sets

Table 10: The Penn Treebank POS Tag Set (Marcus, Marcinkiewicz, & Santorini, 1993).

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

Table 11: The Penn Treebank Syntactic Tag Set (Marcus, Marcinkiewicz, & Santorini, 1993).

Tag	Description
ADJP	Adjective phrase
ADVP	Adverb phrase
NP	Noun phrase
PP	Prepositional phrase
S	Simple declarative clause
SBAR	Clause introduced by subordinating conjunction or 0 (see below)
SBARQ	Direct question introduced by *wh*-word or *wh*-phrase
SINV	Declarative sentence with subject-aux inversion
SQ	Subconstituent of SBARQ excluding *wh*-word or *wh*-phrase
VP	Verb phrase
WHADVP	*wh*-adverb phrase
WHNP	*wh*-noun phrase
WHPP	*wh*-prepositional phrase
X	Constituent of unknown or uncertain category
*	"Understood" subject of infinitive or imperative
0	Zero variant of *that* in subordinate clauses
T	Trace—marks position where moved *wh*-constituent is interpreted
NIL	Marks position where preposition is interpreted in pied-piping contexts

Appendix B

Stop Words in English

Table 12: A list of English stop words used by the Rhetorica software.

I	theirs	should	by	here
me	themselves	could	for	there
my	what	ought	with	when
myself	which	'm	about	where
we	who	're	against	why
us	whom	's	between	how
our	this	've	into	all
ours	that	'd	through	any
ourselves	these	'll	during	both
you	those	n't	before	each
your	am	wo*	after	few
yours	is	sha†	above	more
yourself	are	ca‡	below	most
he	was	cannot	to	other
him	were	a	from	some
his	be	an	up	such
himself	been	the	down	no
she	being	and	in	nor
her	have	but	out	not
hers	has	if	on	only
herself	had	or	off	own
it	having	because	over	same
its	do	as	under	so
itself	does	until	again	than
they	did	while	further	too
them	doing	of	then	very
their	would	at	once	

*Part of *won't*, which the parser resolves into *wo+n't*.
†Part of *sha+n't*.
‡Part of *ca+n't*.

Appendix C

Prefixes and Suffixes in English

Table 13: A list of common English prefixes used by the Rhetorica software.

anti-	in-	mis-	sub-
de-	im-	non-	super-
dis-	il-	over-	trans-
en-	ir-	pre-	un-
em-	inter-	re-	under-
fore-	mid-	semi-	

Table 14: A list of common English suffixes used by the Rhetorica software.

-able	-est	-ity	-ness
-ible	-ful	-ty	-ous
-al	-ic	-ive	-eous
-ial	-ing	-ative	-ious
-ed	-ion	-itive	-s
-en	-tion	-less	-es
-er	-ation	-ly	-y
-or	-ition	-ment	

Appendix D

Precision and Recall Tests Reported in Gawryjołek (2009)

Table 15: Precision and Recall Tests of Gawryjołek's JANTOR Software.

Figure	Total No.	f_{++}*	f_{+-}*	f_{-+}*	Precision (%)	Recall (%)
Epizeuxis	37	36	0	1	100.0	97.2
Ploce	—	—	0	0	100.0	100.0
Polysyndeton	20	19	—	1	—	95.0
Anaphora	—	—	—	—	—	—
Epistrophe	—	—	—	—	†	†
Epanalepsis	—	—	—	—	‡	‡
Anadiplosis	49	47	—	2	—	95.9
Antimetabole	—	—	—	—	§	—
Polyptoton	28	24	—	4	—	85.7
Isocolon	27	23	—	4	$\sim$ 50.0	85.2
Oxymoron	52	42	9	10	82.4	80.8¶

* f_{++}: true positives; f_{+-}: false positives; f_{-+}: false negatives.
† "high recall and precision"
‡ "quite successful on most of the prepared examples"
§ Excessive false positives noted.
¶ Gawryjołek reports very high oxymoron recall compared to Rhetorica, but in the common **Example 45** (*Romeo & Juliet 1.1*), JANTOR found 3/11 oxymorons; Rhetorica found 4/11 oxymorons for setting `greedy:false`, 6/11 for `greedy:true`.

Appendix E

Getting and Using the Rhetorica Software

Rhetorica runs from the Windows (64-bit only) command line. The source code (Visual Studio 2013 solution) resides in a GitHub repository:

https://github.com/priscian/rhetorica

As do the executable files alone:

https://github.com/priscian/rhetorica/raw/master/bin/x64/Debug.zip

Both the VS 2013 solution and the executable rely on external NLP tools:

https://github.com/priscian/nlp

The executable file `Rhetorica.exe` (Windows 64-bit with .NET Framework 4.5.1 installed) requires that the NLP tools repository, which contains files used by the Stanford Parser, OpenNLP, and WordNet, be installed to the root `C:\` directory, so that its path is `C:\NLP\`. If this location is not optimal or possible, then these fields in the file `Rhetorica.exe.config` can be changed from their default values:

```
RootDrive: "C:\"
NlpFolder: "NLP\"
```

If `Rhetorica.exe` is run from the command line without any arguments, it will automatically read in the file `Obama - Inaugural Address (2009).txt`, parse its sentences and find all 14 rhetorical figures. There are two other ways to send a document into Rhetorica for processing:

`Rhetorica.exe [`*`drive`*`:][`*`path`*`][`*`filename`*`]`

`Rhetorica.exe [`*`filename`*`]`

If only the filename is given without an absolute path or one relative to `Rhetorica.exe`, then Rhetorica will look for the file in the directory `C:\NLP\texts\`. For example, the NLP repository's `texts` directory contains the file `obama_2009.txt`, so running the following command will also process President Obama's 2009 Inaugural Address for rhetorical figures:

```
Rhetorica.exe "obama_2009.txt"
```

The command-line interface also allows a second argument with JSON notation for limiting the figures discovered and tweaking the search settings (described in **Table 6**) for any or all the figures; e.g.

```
Rhetorica.exe "Stevens - Farewell to Florida.txt" ^
"{^
  Anadiplosis: { windowSize: 2 },^
  Epizeuxis: { windowSize: 2 },^
  Polysyndeton: { windowSize: 1, extra: 2 },^
  Isocolon: { windowSize: 3, extra: 1 },^
  Oxymoron: { extra: false },^
  All: {}^
}" "stevens"
```

where `"stevens"` is the base filename for the Rhetorica output files. Generally, `Rhetorica.exe` takes three arguments:

```
Rhetorica.exe source_file search_params output_pathbase
```

1. *`source_file`*: Path and filename of a text file to process for rhetorical figures.
2. *`search_params`*: (Optional) JSON object with names of the rhetorical figures to find and optional search settings for each figure.
3. *`output_pathbase`*: (Optional) path and partial filename for storing results. *`output_pathbase`* + `.doc.csv` describes each token in the source document, and *`output_pathbase`* + `.csv` describes each figure discovered, in the context of the source document.

Some further examples follow.

Example 1. *Search for all figures in the file* `test.txt`*, then save the results to* `out.doc.csv` *and* `out.csv` *in the current directory.*

```
Rhetorica.exe "test.txt" "" "out"
```

or

```
Rhetorica.exe "test.txt" "{ All: {} }" "out"
```

Example 2. *Search only for isocolon in the file* `test.txt`*.*

```
Rhetorica.exe "test.txt" "{ Isocolon: {} }"
```

Example 3. *Search only for isocolon with a search window of 2 sentences (default 3) and a similarity threshold of 1 (default 0; see § Isocolon for details).*

```
Rhetorica.exe "test.txt" "{ Isocolon: { windowSize: 2, extra: 1 } }"
```

Example 4. *Search for isocolon with tweaked search settings as in the previous example, but then also search for all the remaining figures with their default settings.*

```
Rhetorica.exe "test.txt" ^
"{^
  Isocolon: { windowSize: 2, extra: 1 },^
  All: {}^
}"
```

Example 5. *Search for isocolon with tweaked search settings, and oxymoron with* `greedy:true` *(see § Oxymoron for details), then save the results to* `out.doc.csv` *and* `out.csv` *in the current directory.*

```
Rhetorica.exe "test.txt" ^
"{^
  Isocolon: { windowSize: 2, extra: 1 },^
  Oxymoron: { extra: true }^
}" "out"
```

Example 6. *Similar search to that of the previous example but with tweaked polysyndeton (minimum consecutive sentence-leading conjunctions comprising a polysyndeton = 3 instead of the default 2; see § Polysyndeton), then save the results to* `out.doc.csv` *and* `out.csv` *in the directory* `C:\NLP\texts\`.

```
Rhetorica.exe "test.txt" ^
"{^
  Isocolon: { windowSize: 2, extra: 1 },^
  Oxymoron: { extra: true },^
  Polysyndeton: { extra: 3 }
}" "C:\NLP\texts\out"
```

References

Argamon, S., Koppel, M., Fine, J., & Shimoni, A. R. (2003). Gender, genre, and writing style in formal written texts. *Text*, *23*(3), 321–346. (Cit. on p. 4).

Argamon, S. & Levitan, S. (2005, June). Measuring the usefulness of function words for authorship attribution. In *Proceedings of the 2005 ACH/ALLC conference*. Victoria, BC, Canada. (Cit. on pp. 4, 14, 73).

Argamon, S., Whitelaw, C., Chase, P., Hota, S. R., Garg, N., & Levitan, S. (2007, April). Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology*, *58*(6), 802–822. doi:10.1002/asi.v58:6. (Cit. on p. 16)

Argamon-Engelson, S., Koppel, M., & Avneri, G. (1998). Style-based text categorization: what newspaper am I reading? In *Proceedings of the AAAI Workshop on Text Categorization* (pp. 1–4). Menlo Park, CA: AAAI Press. (Cit. on p. 15).

Baayen, H. (2008). *Analyzing linguistic data: a practical introduction to statistics using R*. Cambridge University Press. (Cit. on p. 76).

Baayen, H., van Halteren, H., Neijt, A., & Tweedie, F. (2002). An experiment in authorship attribution. In *6th JADT* (pp. 29–37). (Cit. on pp. 4, 14).

Baayen, H., van Halteren, H., & Tweedie, F. (1996, September). Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. *Literary and Linguist Computing*, *11*(3), 121–132. doi:10.1093/llc/11.3.121. (Cit. on pp. 6, 15)

Baldridge, J., Morton, T., & Bierner, G. (2002). *The OpenNLP maximum entropy package*. Sourceforge. Apache. (Cit. on pp. 23, 47).

Bennett, J. R. (Ed.). (1971). *Prose style: a historical approach through studies*. San Francisco: Chandler Publishing Company. (Cit. on p. 7).

Binongo, J. N. G. (2003). Who wrote the 15th Book of Oz? An application of multivariate analysis to authorship attribution. *Chance*, *16*(2), 9–17. doi:10.1080/09332480.2003.10554843. (Cit. on p. 14)

Brinegar, C. S. (1963). Mark Twain and the Quintus Curtius Snodgrass letters: a statistical test of authorship. *Journal of the American Statistical Association*, *58*(301), 85–96. doi:10.1080/01621459.1963.10500834. (Cit. on p. 14)

Brontë, C., Brontë, E., & Brontë, A. (1846). Poems by Currer, Ellis, and Acton Bell. (Cit. on p. 78).

Burrows, J. F. (1987). Word-patterns and story-shapes: the statistical analysis of narrative style. *Literary and Linguistic Computing*, *2*(2), 61–70. (Cit. on pp. 4, 14, 73).

Burton, G. (2007). The Forest of Rhetoric (Silva Rhetoricae). http://humanities.byu.edu/rhetoric/silva.htm. From Brigham Young University. (Cit. on pp. 9, 26, 49).

Chaski, C. E. (2005). Who's at the keyboard: authorship attribution in digital evidence investigations. *International Journal of Digital Evidence*, *4*(1), 1–13. (Cit. on pp. 15, 17).

Chomsky, N. (1999). On the nature, use, and acquisition of language. In W. Ritchie & T. Bhatia (Eds.), *Handbook of child language acquisition* (pp. 33–54). San Diego: Academic Press. (Cit. on p. 4).

[Cicero]. (1954). *Ad C. Herennium: de ratione dicendi (Rhetorica ad Herennium).* Translated by Harry Caplan. Cambridge, MA: Havard University Press. (Cit. on p. 1).

Clement, R. & Sharp, D. (2003). Ngram and bayesian classification of documents for topic and authorship. *Literary and Linguistic Computing*, *18*(4), 423–447. doi:10.1093/llc/18.4.423. (Cit. on p. 14)

Corbett, E. P. J. (1990). *Classical rhetoric for the modern student* (3d). USA: Oxford University Press. (Cit. on pp. 1, 26, 32, 34, 38, 41, 43, 77).

Corbett, E. P. J. & Connors, R. J. (1998). *Classical rhetoric for the modern student* (4th). USA: Oxford University Press. (Cit. on pp. 9, 26, 49).

Crowley, S. & Hawhee, D. (2004). *Ancient rhetorics for contemporary students* (3d). USA: Pearson Education, Inc. (Cit. on pp. 9, 26, 49).

Damerau, F. J. (1975). The use of function word frequencies as indicators of style. *Computers and the Humanities*, *9*(6), 271–280. (Cit. on p. 4).

De Marneffe, M.-C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)* (Vol. 6, pp. 449–454). (Cit. on pp. 42–44).

De Meyer, H., De Baets, B., & Janssens, S. (2001). Similarity measurement on leaf-labelled trees. In *Proceedings of the 2nd EUSFLAT Conference (Leicester, UK)* (pp. 253–256). (Cit. on p. 84).

de Vel, O. (2000). Mining e-mail authorship. Paper presented at workshop on text mining. In: ACM International Conference on Knowledge Discovery and Data Mining (KDD). (Cit. on pp. 15, 16).

de Vel, O., Anderson, A., Corney, M., & Mohay, G. (2001, December). Mining e-mail content for author identification forensics. *SIGMOD Record*, *30*(4), 55–64. doi:10.1145/604264.604272. (Cit. on p. 18)

Diederich, J., Kindermann, J., Leopold, E., & Paass, G. (2003, May). Authorship attribution with support vector machines. *Applied Intelligence*, *19*(1-2), 109–123. doi:10.1023/A:1023824908771. (Cit. on pp. 13, 18)

Du Marsais, C. C. (1804). *Des tropes, ou des différens sens dans lesquels on peut prendre un même mot dans une même langue* (nouvelle édition). À Lyon: Chez Tournachon–Molin. (Cit. on p. 2).

Dubremetz, M. & Nivre, J. (2015). Rhetorical figure detection: the case of chiasmus. In *Proceedings of NAACL-HLT Fourth Workshop on Computational Linguistics for Literature* (pp. 23–31). Association for Computational Linguistics. (Cit. on pp. 70, 85).

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification* (2nd). Wiley-Interscience. (Cit. on p. 11).

Fahnestock, J. (1999). *Rhetorical figures in science*. USA: Oxford University Press. (Cit. on pp. 1, 9, 26, 34, 49).

Farnsworth, W. (2011). *Farnsworth's classical English rhetoric*. Jaffrey, New Hampshire, USA: David R. Godine. (Cit. on pp. 9, 26, 27, 29, 32, 34, 49, 70).

Fellbaum, C. (Ed.). (1998). *WordNet: an electronic lexical database*. Cambridge, Massachusetts: MIT Press. (Cit. on p. 35).

Fellbaum, C. (2006). WordNet(s). In K. Brown (Ed.), *Encyclopedia of language & linguistics* (2nd ed., Vol. 14, pp. 665–670). Elsevier Science. (Cit. on pp. 35, 37).

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, *76*(5), 378. (Cit. on p. 84).

Fontanier, P. (1977). *Les figures du discours*. Paris: Flammarion. (Cit. on p. 2).

Forman, G. (2003, March). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, *3*, 1289–1305. Retrieved from http://dl.acm.org/citation.cfm?id=944919.944974. (Cit. on p. 4)

Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H., & Trigg, L. (2010). Weka-A Machine Learning Workbench for Data Mining. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (2nd, Chap. 66, pp. 1269–1277). Boston, MA: Springer US. doi:10.1007/978-0-387-09823-4_66. (Cit. on p. 48)

Fucks, W. (1952). On mathematical analysis of style. *Biometrika*, *39*(1/2), 122–129. (Cit. on p. 14).

Gamon, M. (2004, August). Linguistic correlates of style: authorship classification with deep linguistic analysis features. In *Proceedings of the 20th International Conference of Computational Linguistics* (pp. 611–617). Geneva, Switzerland: COLING. (Cit. on pp. 5, 15, 19).

Gawryjołek, J. J. (2009). *Automated annotation and visualization of rhetorical figures* (Master's thesis, University of Waterloo, Waterloo, Ontario, Canada). (Cit. on pp. 3–6, 9, 16, 19, 26, 29, 39, 40, 44, 47, 81, 91).

Gillick, D. (2009). Sentence boundary detection and the problem with the U.S. In *Proceedings of human language technologies: the 2009 annual conference of the North American Chapter of the Association for Computational Linguistics, companion volume: short papers* (pp. 241–244). Association for Computational Linguistics. (Cit. on p. 23).

Gopnik, M. (Ed.). (1997). *The inheritance and innateness of grammars*. Vancouver Studies in Cognitive Sciences. Oxford University Press, USA. (Cit. on p. 4).

Graham, N., Hirst, G., & Marthi, B. (2005, November). Segmenting documents by stylistic character. *Natural Language Engineering*, *11*(4), 397–415. doi:10.1017/S1351324905003694. (Cit. on p. 20)

Halliday, M. A. K. (1967). The linguistic study of literary texts. In S. Chatman & S. R. Levin (Eds.), *Essays on the language of literature* (pp. 217–223). Boston: Houghton Miffin. (Cit. on p. 7).

Halliday, M. A. K. (1994). *An introduction to functional grammar* (2nd). London: Hodder Arnold. (Cit. on p. 7).

Harrell, F. E. (2001). *Regression modeling strategies, with applications to linear models, survival analysis and logistic regression*. Springer. (Cit. on p. 74).

Harris, R. & DiMarco, C. (2009). Constructing a rhetorical figuration ontology. In *Proceedings of the Persuasive Technology and Digital Behaviour Intervention Symposium: A symposium at the AISB 2009 Convention (6-9 April 2009) Heriot–Watt University, Edinburgh, Scotland* (pp. 47–52). The Society for the Study of Artificial Intelligence and the Simulation of Behaviour. (Cit. on pp. 9, 19, 26, 27).

Hindle, D. & Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics*, *19*(1), 103–120. (Cit. on p. 24).

Hirst, G. & Feiguina, O. (2007). Bigrams of syntactic labels for authorship discrimination of short texts. *Literary and Linguistic Computing*, *22*(4), 405–417. doi:10.1093/llc/fqm023. (Cit. on pp. 6, 15, 20, 22, 73, 77)

Holmes, D. I. [D. I.] & Forsyth, R. S. (1995). The Federalist revisited: new directions in authorship attribution. *Literary and Linguistic Computing*, *10*(2), 111–127. doi:10.1093/llc/10.2.111. (Cit. on p. 18)

Holmes, D. I. [David I.]. (1998). The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing*, *13*(3), 111–117. doi:10.1093/llc/13.3.111. (Cit. on p. 14)

Holmes, D., Robertson, M., & Paez, R. (2001). Stephen Crane and the *New-York Tribune*: a case study in traditional and non-traditional authorship attribution. *Computers and the Humanities*, *35*(3), 315–331. (Cit. on pp. 4, 14).

Honoré, A. (1979). Some simple measures of richness of vocabulary. *Association for Literary and Linguistic Computing Bulletin*, *7*(2), 172–177. (Cit. on p. 14).

Houvardas, J. & Stamatatos, E. (2006). N-gram feature selection for authorship identification. In J. Euzenat & J. Domingue (Eds.), *Artificial intelligence: methodology, systems, and applications* (Vol. 4183, pp. 77–86). Lecture Notes in Computer Science. Springer Berlin Heidelberg. doi:10.1007/11861461_10. (Cit. on p. 14)

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In C. Nédellec & C. Rouveirol (Eds.), *Machine learning: ECML-98* (Chap. 19, Vol. 1398, pp. 137–142). Lecture Notes in Computer Science. Berlin/Heidelberg: Springer Berlin/Heidelberg. doi:10.1007/bfb0026683. (Cit. on pp. 4, 18)

Joachims, T. (2002). *Learning to classify text using support vector machines*. The Kluwer International Series in Engineering and Computer Science. Norwell, Massachusetts, USA: Kluwer Academic Publishers. (Cit. on pp. 12, 18, 20).

Juola, P. & Baayen, H. (2005). A controlled-corpus experiment in authorship identification by cross-entropy. *Literary and Linguistic Computing*, *20*(Supplement), 59–67. doi:10.1093/llc/fqi024. (Cit. on pp. 4, 14)

Juola, P., Sofko, J., & Brennan, P. (2006). A prototype for authorship attribution studies. *Literary and Linguistic Computing*, *21*(2), 169–178. (Cit. on pp. 21, 22, 76).

Jurafsky, D. & Martin, J. H. (2009). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition* (2nd). USA: Pearson Education, Inc. (Cit. on p. 24).

Karlgren, J. (2000). *Stylistic experiments for information retrieval* (Doctoral dissertation, Swedish Institute of Computer Science, Kista, Sweden). (Cit. on p. 4).

Karlgren, J. & Cutting, D. (1994). Recognizing text genres with simple metrics using discriminant analysis. In *Proceedings of the 15th conference on computational*

linguistics (Vol. 2, pp. 1071–1075). Association for Computational Linguistics. (Cit. on p. 14).

Kessler, B., Numberg, G., & Schütze, H. (1997). Automatic detection of text genre. In *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics* (pp. 32–38). EACL '97. Madrid, Spain: Association for Computational Linguistics. doi:10.3115/979617.979622. (Cit. on p. 14)

Kirchner, R. (2007). *Elocutio*: Latin prose style. In W. Dominik & J. Hall (Eds.), *A companion to Roman rhetoric* (pp. 181–194). Malden, MA, USA: Blackwell Publishing. (Cit. on p. 1).

Kjell, B. (1994a, October). Authorship attribution of text samples using neural networks and bayesian classifiers. In *1994 IEEE International Conference on Systems, Man, and Cybernetics: humans, information and technology* (Vol. 2, pp. 1660–1664). doi:10.1109/ICSMC.1994.400086. (Cit. on p. 14)

Kjell, B. (1994b). Authorship determination using letter pair frequency features with neural network classifiers. *Literary and Linguistic Computing*, *9*(2), 119–124. doi:10.1093/llc/9.2.119. (Cit. on p. 14)

Kjell, B., Addison Woods, W., & Frieder, O. (1995, October). Information retrieval using letter tuples with neural network and nearest neighbor classifiers. In *IEEE International Conference on Systems, Man, and Cybernetics: intelligent systems for the 21st century* (Vol. 2, pp. 1222–1226). doi:10.1109/ICSMC.1995.537938. (Cit. on p. 14)

Klein, D. & Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, volume 1* (pp. 423–430). ACL '03. Sapporo, Japan: Association for Computational Linguistics. doi:10.3115/1075096.1075150. (Cit. on pp. 3, 22–24, 47, 64)

Koppel, M., Akiva, N., & Dagan, I. (2006). Feature instability as a criterion for selecting potential style markers. *Journal of the American Society for Information Science and Technology*, *57*(11), 1519–1525. doi:10.1002/asi.20428. (Cit. on pp. 14, 15)

Koppel, M., Argamon, S., & Shimoni, A. R. (2002). Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, *17*(4), 401–412. doi:10.1093/llc/17.4.401. (Cit. on p. 15)

Koppel, M. & Schler, J. (2003). Exploiting stylistic idiosyncrasies for authorship attribution. In *IJCAI '03 workshop on computational approaches to style analysis and synthesis* (pp. 69–72). (Cit. on p. 15).

Koppel, M. & Schler, J. (2004). Authorship verification as a one-class classification problem. In *Proceedings of the twenty-first International Conference on Machine*

Learning (pp. 62–68). ICML '04. Banff, Alberta, Canada: ACM. doi:10.1145/1015330.1015448. (Cit. on p. 18)

Koppel, M., Schler, J., & Argamon, S. (2009). Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, *60*(1), 9–26. doi:10.1002/asi.20961. (Cit. on pp. 14, 15)

Koppel, M., Schler, J., & Zigdon, K. (2005). Determining an author's native language by mining a text for errors. In *Proceedings of KDD 2005* (pp. 624–628). Chicago, IL. (Cit. on pp. 14, 15).

Korenius, T., Laurikkala, J., Järvelin, K., & Juhola, M. (2004). Stemming and lemmatization in the clustering of Finnish text documents. In *Proceedings of the thirteenth ACM International Conference on Information and Knowledge Management* (pp. 625–633). ACM. (Cit. on p. 83).

Kukushkina, O. V., Polikarpov, A. A., & Khmelev, D. V. (2001, April). Using literal and grammatical statistics for authorship attribution. *Problems of Information Transmission*, *37*(2), 172–184. doi:10.1023/A:1010478226705. (Cit. on p. 15)

Lanham, R. A. (1991). *A handlist of rhetorical terms* (2nd). Berkley: University of California Press. (Cit. on pp. 6, 9, 26, 27, 33, 34, 43, 49, 53).

Ledger, G. & Merriam, T. (1994). Shakespeare, Fletcher, and the Two Noble Kinsmen. *Literary and Linguistic Computing*, *9*(3), 235–248. doi:10.1093/llc/9.3.235. (Cit. on p. 14)

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady* (Vol. 10, p. 707). (Cit. on pp. 39, 40).

Li, J., Zheng, R., & Chen, H. (2006, April). From fingerprint to writeprint. *Communications of the ACM*, *49*(4), 76–82. doi:10.1145/1121949.1121951. (Cit. on p. 18)

Love, H. (2002). *Attributing authorship: an introduction*. Cambridge University Press. (Cit. on p. 2).

Lu, X. (2014). *Computational methods for corpus annotation and analysis*. Springer. (Cit. on p. 71).

Mann, W. C. & Thompson, S. A. (1988). Rhetorical structure theory: toward a functional theory of text organization. *Text*, *8*(3), 243–281. (Cit. on p. 15).

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993, June). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, *19*(2), 313–330. Retrieved from http://dl.acm.org/citation.cfm?id=972470.972475. (Cit. on pp. 3, 24, 87, 88)

Matthews, R. A. J. & Merriam, T. V. N. (1993). Neural computation in stylometry i: an application to the works of Shakespeare and Fletcher. *Literary and Linguistic Computing*, *8*(4), 203–209. doi:10.1093/llc/8.4.203. (Cit. on p. 18)

Matthews, R. A. J. & Merriam, T. V. N. (1997). Distinguishing literary styles using neural networks. In E. Fiesler & R. Beale (Eds.), *Handbook of neural computation*. IOP Publishing and Oxford University Press. (Cit. on pp. 16, 18).

Merriam, T. V. N. & Matthews, R. A. J. (1994). Neural computation in stylometry ii: an application to the works of Shakespeare and Marlowe. *Literary and Linguistic Computing*, *9*(1), 1–6. doi:10.1093/llc/9.1.1. (Cit. on p. 18)

Meynet, R. (2012). *Treatise on biblical rhetoric*. Brill. (Cit. on p. 34).

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*, 39–41. (Cit. on pp. 22, 35, 42, 43, 47).

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, *3*(4), 235–244. doi:10.1093/ijl/3.4.235. (Cit. on pp. 35, 44)

Morton, A. Q. (1965). The authorship of Greek prose. *Journal of the Royal Statistical Society. Series A (General)*, *128*(2), 169–233. (Cit. on p. 14).

Morton, A. Q. (1978). *Literary detection: how to prove authorship and fraud in literature and documents*. Bowker London. (Cit. on p. 14).

Mosteller, F. & Wallace, D. L. (1964). *Inference and disputed authorship: the Federalist*. Reading, MA, USA: Addison-Wesley. (Cit. on pp. 4, 16, 22, 74).

Murphy, J. J., Katula, R. A., Hill, F. I., & Ochs, D. J. (2003). *A synoptic history of classical rhetoric* (3d). Hermagoras Press. (Cit. on p. 34).

Nierman, A. & Jagadish, H. V. (2002). Evaluating structural similarity in XML documents. In *Proceedings of the 5th International Workshop on the Web and Databases (WebDB 2002), Madison, Wisconsin, USA* (pp. 61–66). (Cit. on p. 84).

OED Online. (2006a). phrase, n. Retrieved June 2014, from http://www.oed.com/. (Cit. on p. 11)

OED Online. (2006b). sentence, n. Retrieved June 2014, from http://www.oed.com/. (Cit. on p. 12)

O'Grady, W. (1999). The acquisition of syntactic representations: a general nativist approach. In W. Ritchie & T. Bhatia (Eds.), *Handbook of child language acquisition* (pp. 157–194). San Diego: Academic Press. (Cit. on p. 4).

Pinker, S. (2003). *The blank slate: the modern denial of human nature*. Penguin. (Cit. on p. 4).

Porter, M. F. (1997). Readings in information retrieval. In K. Sparck Jones & P. Willett (Eds.), (Chap. An algorithm for suffix stripping, pp. 313–316). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. (Cit. on pp. 36, 37, 47, 63).

Quinn, A. (1982). *Figures of speech: 60 ways to turn a phrase*. Layton, Utah: Gibbs. M. Smith • Inc. (Cit. on pp. 9, 26, 27, 49).

R Development Core Team. (2012). *R: a language and environment for statistical computing*. ISBN 3-900051-07-0. R Foundation for Statistical Computing. Vienna, Austria. Retrieved from http://www.R-project.org/. (Cit. on p. 48)

Ratnaparkhi, A. (1998). *Maximum entropy models for natural language ambiguity resolution* (Doctoral dissertation, University of Pennsylvania, Philadelphia, PA, USA). (Cit. on p. 23).

Rice, M. L. (Ed.). (1996). *Toward a genetics of language*. Mahwah, NJ, USA: Lawrence Erbaum Associates. (Cit. on p. 3).

Sanderson, C. & Guenter, S. (2006). Short text authorship attribution via sequence kernels, Markov chains and author unmasking: an investigation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (pp. 482–491). EMNLP '06. Sydney, Australia: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1610075.1610142. (Cit. on p. 18)

Savoy, J. (2013). The Federalist papers revisited: a collaborative attribution scheme. *Proceedings of the American Society for Information Science and Technology*, *50*(1), 1–8. (Cit. on p. 74).

Sichel, H. S. (1975). On a distribution law for word frequencies. *Journal of the American Statistical Association*, *70*(351), 542–547. (Cit. on p. 14).

Stamatatos, E. (2008). Author identification: using text sampling to handle the class imbalance problem. *Information Processing & Management*, *44*(2), 790–799. doi:10.1016/j.ipm.2007.05.012. (Cit. on p. 14)

Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, *60*(3), 538–556. doi:10.1002/asi.v60:3. (Cit. on pp. 2–4, 10, 15, 17, 20)

Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (2000). Automatic text categorization in terms of genre and author. *Computational Linguistics*, *26*(4), 471–495. (Cit. on pp. 15–17).

Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (2001). Computer-based authorship attribution without lexical measures. *Computers and the Humanities*, *35*(2), 193–214. (Cit. on p. 15).

Stanford NLP Group. (2014, January). Stanford Parser FAQ. Retrieved from http://nlp.stanford.edu/software/parser-faq.shtml. (Cit. on p. 71)

Strommer, C. W. (2011). *Using rhetorical figures and shallow attributes as a metric of intent in text* (Doctoral dissertation, University of Waterloo, Waterloo, Ontario, Canada). (Cit. on p. 16).

Strozer, J. R. (1994). *Language acquisition after puberty*. Georgetown Studies in Romance linguistics. Washington, D.C.: Georgetown University Press. (Cit. on p. 3).

Taboada, M. & Mann, W. C. (2006). Rhetorical structure theory: looking back and moving ahead. *Discourse Studies*, *8*, 423–459. (Cit. on p. 15).

The Free Software Foundation. (2012). The GNU project. Web site. http://www.gnu.org/. (Cit. on p. 48).

Tweedie, F. J., Singh, S., & Holmes, D. I. (1996). Neural network applications in stylometry: the Federalist Papers. *Computers and the Humanities*, *30*(1), 1–10. (Cit. on pp. 4, 16, 18, 22).

Uebersax, J. S. (1983). A design-independent method for measuring the reliability of psychiatric diagnosis. *Journal of Psychiatric Research*, *17*(4), 335–342. (Cit. on p. 84).

Uebersax, J. S. & Grove, W. M. (1990). Latent class analysis of diagnostic agreement. *Statistics in Medicine*, *9*(5), 559–572. (Cit. on p. 84).

Uzuner, Ö. & Katz, B. (2005). A comparative study of language models for book and author recognition. In R. Dale, K.-F. Wong, J. Su, & O. Y. Kwong (Eds.), *Proceedings of the 2nd International Joint Conference on Natural Language Processing* (Vol. 3651, pp. 969–980). Lecture Notes in Computer Science. Springer. doi:10.1007/11562214_84. (Cit. on pp. 15, 18)

van Halteren, H. (2004, July). Linguistic profiling for authorship recognition and verification. In *Proceedings of the 42nd Conference of the ACL* (pp. 199–206). East Stroudsburg, PA: ACL. (Cit. on p. 15).

van Halteren, H., Baayen, H., Tweedie, F., Haverkort, M., & Neijt, A. (2005). New machine learning methods demonstrate the existence of a human stylome. *Journal of Quantitative Linguistics*, *12*(1), 65–77. doi:10.1080/09296170500055350. (Cit. on p. 4)

Vuong, Q. H. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica: Journal of the Econometric Society*, 307–333. (Cit. on p. 79).

Wexler, K. (1999). Maturation and growth of grammar. In W. Ritchie & T. Bhatia (Eds.), *Handbook of child language acquisition* (pp. 55–110). San Diego: Academic Press. (Cit. on p. 4).

Wikipedia. (2012, December). Longest common subsequence problem. Retrieved from http://en.wikipedia.org/wiki/Longest_common_subsequence_problem. (Cit. on p. 39)

Xing, G., Guo, J., & Xia, Z. (2007). Classifying XML documents based on structure/content similarity. *Comparative Evaluation of XML Information Retrieval Systems*, 444–457. (Cit. on p. 84).

Yule, G. U. (1944). *The statistical study of literary vocabulary*. Cambridge, UK: Cambridge University Press. (Cit. on pp. 14, 16).

Zhao, Y. & Zobel, J. (2005). Effective and scalable authorship attribution using function words. In G. Lee, A. Yamada, H. Meng, & S. Myaeng (Eds.), *Information retrieval technology* (Vol. 3689, pp. 174–189). Lecture Notes in Computer Science. Springer Berlin/Heidelberg. doi:10.1007/11562382_14. (Cit. on pp. 4, 14, 18)

Zhao, Y., Zobel, J., & Vines, P. (2006). Using relative entropy for authorship attribution. In *Proceedings of the 3rd AIRS Asian Information Retrieval Symposium* (pp. 92–105). NY: Springer. (Cit. on p. 15).

Zheng, R., Li, J., Chen, H., & Huang, Z. (2006). A framework for authorship identification of online messages: writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, *57*(3), 378–393. doi:10.1002/asi.20316. (Cit. on pp. 15, 18)

Zipf, G. K. (1932). *Selected studies of the principle of relative frequency in language*. MA: Harvard University Press. (Cit. on p. 14).