Syntactic Stylometry:
Using Sentence Structure for Authorship Attribution

by

Charles Dale Hollingsworth

(Under the direction of Michael Covington)

Abstract

Most approaches to statistical stylometry have concentrated on lexical features, such as relative word frequencies or type-token ratios. Syntactic features have been largely ignored. This work attempts to fill that void by introducing a technique for authorship attribution based on dependency grammar. Syntactic features are extracted from texts using a common dependency parser, and those features are used to train a classifier to identify texts by author. While the method described does not outperform existing methods on most tasks, it does demonstrate that purely syntactic features carry information which could be useful for stylometric analysis.

Index words: stylometry, authorship attribution, dependency grammar, machine learning

Syntactic Stylometry:
Using Sentence Structure for Authorship Attribution

by

Charles Dale Hollingsworth

B.A., Mississippi State University, 2003

M.A., Louisiana State University, 2005

A Thesis Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Master of Science

Athens, Georgia

2012

Syntactic Stylometry:

Using Sentence Structure for Authorship Attribution

by

Charles Dale Hollingsworth

Approved:

Major Professor:	Michael Covington
Committee:	William Kretzschmar
	L. Chad Howe

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2012

Syntactic Stylometry: Using Sentence Structure for Authorship Attribution

Charles Dale Hollingsworth

July 22, 2012

Acknowledgments

I would like to thank my major professor, Dr. Michael Covington, and my committee members, William Kretzschmar and L. Chad Howe, for their help and support. I would also like to thank my fellow students David Robinson and Shayi Zhang, for helping me prepare and proofread this document. Finally, I wish to thank my friends Donald and Amy Goodman-Wilson, for their constant support throughout my academic career.

This thesis is dedicated to my mother, Maureen Hollingsworth, and my goddaughter, Acadia Goodman-Wilson.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Attempts to determine the authorship of unknown texts have traditionally relied on lexical or orthographic features. This trend goes at least as far back as Mosteller and Wallace's 1963 study of the *Federalist* papers, in which the relative frequencies of function words were used to attribute the disputed papers to Madison [31]. Syntactic features have, with a few recent exceptions, largely been ignored. This is hardly surprising, given that computers powerful enough to parse large collections of text have only recently become commonplace. Nevertheless, sentence structure is intuitively as important as word choice in determining an author's style.

Even among those studies that have addressed syntactic features, dependency grammars have been under-represented. Most have looked at very shallow features, such as part-of-speech tags. Those that attempted a deeper syntactic analysis have primarily used phrase structure grammars. The use of dependency grammars for stylometric analysis is almost unknown.

Though lexical methods have been successful at the task of authorship attribution, there is always a call for new types of style markers. R. W. Bailey remarked that features "should

be salient, structural, frequent and easily quantifiable, and relatively immune from conscious control" [4]. Syntactic features would seem to meet all of these conditions.

This work demonstrates that syntactic features, and in particular grammatical dependencies, can be useful features for authorship attribution. No attempt is made to evaluate different learning methods or distance metrics, or to perform stylometric tasks other than authorship attribution. Instead, the same general experimental setup is tried with various features, both dependency-based and otherwise, to determine how syntactic features compare to other features currently in use.

Chapter 2 gives a review of the existing literature on authorship attribution. In it, I note the relative paucity of stylometric approaches that take advantage of deep syntactic features. I also look at the various types of machine learners that have been used for authorship attribution, and the corpora that have been used to train and test these learners.

Chapter 3 describes my first experiment using grammatical dependency information for authorship attribution. I report some successes with using syntactic features on a test corpus consisting of early twentieth-century detective novels. The same classifier is then tested on some of Juola's Ad-Hoc Authorship Attribution Competition tasks [20], with less impressive results.

Chapter 4 describes an attempt to improve my methods by identifying and using only those features that do the best job of distinguishing among authors. I report some improvements over the results obtained in chapter 3. I also demonstrate that the improved syntactic features do a better job than other features of clustering works according to authorship.

Finally, chapter 5 gives some suggestions for further research. I suggest ways in which the method described in this work might be improved, as well as possible applications for the technique.

Chapter 2

Literature review

2.1 Syntactic features in stylometry

In most reviews of the history of stylometry, syntactic features are conspicuous in their near-absence. Holmes, in his 1994 survey [17], identified only a few syntactic features in use, and observes that such features are usually ignored because of the relative lack of data compared to lexical features. A subsequent work by Holmes in 1998 [18] noted the growth of syntactic methods in the 1990s, and attributed this growth to increases in computing power and the availability of digital texts. By 2000, Stamatatos et al. observed that syntactic features were being used more often, but expressed doubts as to their usefulness: "Current NLP tools are not able to provide accurate calculation results for many of the previously proposed style markers" [37] . For this reason, they claimed, most researchers preferred to avoid syntactic features. [1] Juola, in 2006 [21], also identifies only a few (quasi-)syntactic features. The subsequent survey by Stamatatos in 2009 [38] was an exception to the rule,

[1] It should be noted, however, that for the task of authorship attribution, an *accurate* calculation may not be all that important: as long as a parser's output is *consistent*, the features it identifies can be used to train machine learners — even if they do not correspond exactly to the features a human expert might identify.

citing a number of studies that drew features from full or partial parsers, as well as some more shallow methods.

2.1.1 Quasi-syntactic or shallow syntactic features

Many of the syntactic features that have been used for stylometric analysis may be termed "quasi-syntactic" or "shallow syntactic" features. These are superficial, easily counted features that serve as a proxy for deeper syntactic information without requiring a full parse of the sentence. Examples of such quasi-syntactic features include sentence length [46, 30], part-of-speech distribution [1, 7, 8], and type-token ratio [25, 5]. Function words [31, 33] can also be considered quasi-syntactic features, since they tend to be prepositions, conjunctions, determiners, or other words which indicate the presence of certain syntactic structures.

Stamatos et al. used a tool known as a "sentence and chunk boundary detector" (SCBD) to provide shallow syntactic features for authorship attribution of Modern Greek texts [36, 37]. These features included counts and lengths for various types of phrases (noun phrases, verb phrases, and so forth). The syntactic features, both alone and in combination with lexical features, outperformed purely lexical features, though the authors noted that accuracy improved with text length.

The forensic linguist Carole Chaski has used some quasi-syntactic features for authorship attribution [9]. These include "syntactically classified punctuation" (in which punctuation marks are counted according to the kind of syntactic boundary they mark) and markedness. "Markedness" refers to the use of vocabulary or syntactic structures that are less common in the language than their alternatives. For example, in English, head-final constructions (such as "white house", where *house* is the head) are more common than non-head-final constructions (such as "house on the corner"). Chaski argued that markedness was a useful feature for distinguishing authors; however, her methodology has been called into question [16].

2.1.2 Deeper syntactic features

"Deep" syntactic features are those that result from a full or partial parse of a sentence. They are much harder to extract than lexical or quasi-syntactic features, but potentially carry much more information about sentence structure. Since there have been relatively few attempts at using deep syntactic features for stylometry, this subsection will look at a few of them in detail.

Perhaps the first serious use of deep syntactic features was by Baayen et al.[2]. The authors used parse data from the Nijmegen corpus to construct a list of tokens in the form of phrase structure rewrite rules, ordered these tokens from most to least frequent, and gave each an identifier based on its frequency. (For example, W0084 would represent the 84th most frequent rewrite rule.) These identifiers were then used as "pseudo-words" for stylometric analysis. Two experiments were performed. Each experiment applied the same analysis to the actual words of the texts and to the rewrite rule identifiers and compared the two results. In the first experiment, the features used for classification were measures of vocabulary richness (based on relative word frequencies, type-token ratios, and the number of hapax legomena and dislegomena). In the second experiment, the features were the relative frequencies of the fifty highest-frequency function words or rewrite rules. In both experiments, principal component analysis was used to classify samples, and both showed that rewrite rules outperformed purely lexical analyses.

Van Halteren [43] used the Amazon parser to derive rewrite rules, and used three types of constituent n-grams: left-hand side label alone, left-hand side label plus one label from the right-hand side, and left-hand side label plus two labels from the right-hand side. In addition to these syntactic features, a number of lexical features were used. Experiments showed that purely lexical methods outperformed purely syntactic ones. The best combined methods outperformed the best individual systems, but there was no clear winner between lexical-lexical combinations and lexical-syntactic combinations.

In an experiment by Kaster [23], grammatical dependency relations, partial parse trees, and syntax tree depth provided features for analysis, alongside lexical and orthographic measures. It was found that the additional features outperformed chance but did not do as well as the bag-of-words method. However, the combination of both types of features resulted in significant improvements.

Raghavan et al. [35] used probabilistic context-free grammars (PCFGs) to identify authorship. A PCFG consists of a set of production rules, with a probability associated with each rule. Given a PCFG for a language and a sentence in that language, one can calculate a probability for that sentence by multiplying the probabilities for the rules used to generate it. Raghavan et al. used the Stanford Parser[26] to parse the training texts, induced a PCFG for each author from the parsed texts, and calculated the probabilities for the texts in the test set. Each text was assigned to the author whose PCFG gave it the highest probability. The experiment showed that PCFG-based author attribution outperformed a bigram model on two data sets (one of New York Times articles on football, the other of poems from Project Gutenberg), but did not fare as well for other data sets.

Levitsky [28] extracted syntactic features such as sentence and phrase length and complexity from hand-parsed texts. No attempt was made to classify unknown documents, but the results for various authors were compared, and it was found that certain traits were specific to certain authors — a high frequency of subject clauses in Hemingway, for example, and attributive clauses for Dreiser.

Notably, almost all of these approaches used phrase structure grammars. The only use of dependencies was in [23], and these were only "functional" dependencies — the agent, action, and objects of the sentence. More detailed grammatical dependency relations do not appear to have been used for stylometric analysis.

2.2 Choosing texts for training and testing

Since most authorship attribution methods are a form of machine learning, most experiments follow a familiar pattern: the learner is trained on a set of texts, then tested on a smaller, disjunct set of texts, and its success rate in classifying the test set is evaluated. However, it is not immediately obvious what texts should be chosen for training and testing, and a poorly chosen training sample can skew the results of the experiment. For example, some measures such as type-token ratio increase as a function of sample size, and so a system that had only been trained and tested on lengthy texts might not perform as expected given shorter texts [16, 3].

There are two main strategies to selecting texts for author attribution, each with its own risks [16]. The first is to train using a narrow selection of similar texts, for example texts of similar length and topic, from authors of similar linguistic backgrounds. The problem with this approach is that a learner trained on such a sample might generalize poorly, and not perform as well when used on texts different from those it was trained on. The other approach is to train with a large, diverse collection of texts in the hope of finding generally useful features. The problem with this approach is that the learner may pick up on incidental features that would not be useful for distinguishing between more similar authors. A learner that could distinguish Shakespeare from Emily Dickinson might simply be identifying differences in the way English was written in two different centuries, and thus be unable to distinguish Shakespeare from Christopher Marlowe.

Sometimes the task for which a learner is to be used helps dictate the training corpus. Thus a learner that will primarily be used to identify contemporary works would probably not need to be trained on eighteenth-century manuscripts. It might instead be trained on a collection of recent newspaper articles [11, 37], or customer feedback data submitted through a company's web page [13].

Other times the training corpus is determined primarily by expediency, as when a readily available, computer-readable corpus already exists. Some experiments have used the British National Corpus [27] and the Project Gutenberg texts [23, 14].

If there is anything like a "standard" corpus for authorship attribution testing, it is the *Federalist* papers, eighty-five anonymously published political tracts written by James Madison, Alexander Hamilton, and John Jay. Twelve of these texts were of disputed authorship, having been attributed to either Madison or Hamilton. These papers were the subject of Mosteller and Wallace's pioneering work on authorship attribution [31], and have since been used by many others to test new stylometric methods [24, 19, 42, 32]. Unfortunately, the *Federalist* papers are probably not the best choice for testing new methods of authorship attribution. While there is a general consensus that the disputed texts were written by Madison, this fact is not known for certain. Furthermore, Madison and Hamilton are known to have collaborated on some of the other texts, and even if Madison were the primary author of the disputed texts, this does not rule out the possibility that Hamilton may have made some contributions. Finally, if we assume the disputed texts are by Madison, that means that Hamilton wrote almost twice as many papers as Madison. The uneven number of samples for the two authors might confuse some learning methods.

There is one body of texts that might serve as a good benchmark corpus for authorship attribution. As part of his 2004 Ad-Hoc Authorship Attribution Competition, Juola assembled a diverse body of texts to serve as training and testing sets [20, 22]. This collection consists mostly of English texts, though there are some Latin, Dutch, French, and Serbian texts as well. Essays, letters, novels, plays, speeches, and poems are all represented. This corpus could be especially useful for determining whether an authorship attribution method that does not perform particularly well on one type of text might prove effective on others. However, most of the problem sets feature few training texts per author, so methods that

work better with a large amount of training data would be expected to perform poorly on these tasks.

2.3 Learning methods

Once a set of features has been selected and the training and testing documents chosen, one must run the actual experiment. Typically this means either using traditional statistical methods to ascribe authorship, or training a machine learner to classify the texts. As with the other aspects of authorship attribution, a wide variety of learning methods have been attempted.

Statistical methods such as discriminant analysis and principal component analysis appear to be a favorite, having been used by a number of studies [31, 36, 37, 2, 9, 43]. Support vector machines are another popular method [13, 11], as are neural networks [24, 42]. Other methods that have been attempted include ant colony optimization [32], genetic algorithms [19], and Markov models [14].

In the end, the choice of learner may be less important that it would seem. Studies [44, 45] cited in [37] have shown that with a large enough training set (more than 300 instances per category), most classification methods perform comparably; with fewer than 10 positive instances per category, linear least-squares fit and k-nearest neighbors outperform neural networks and naive Bayes.

Chapter 3

An experiment with dependency-based features

This chapter describes an experiment that was conducted in order to evaluate the viability of dependency-based features for stylometric analysis.

3.1 The corpus

In order to minimize the effects of genre and time period, a corpus was selected consisting of roughly contemporaneous works, most of which belonged to a single genre. The corpus used in this experiment was adapted from that used in [14]. It consisted of a total of forty texts from three authors of early twentieth century detective fiction, all freely available from Project Gutenberg [34]. The corpus used here differed from that used in [14], in that it included only the works of E. W. Hornung, Sax Rohmer, and J. S. Fletcher. The works of Maurice LeBlanc were omitted because that author originally wrote in French, and it would be impossible to determine whether any style markers in the English translations were due to the author or to the translators. The works of Arthur J. Rees were omitted because only four

texts by that author appeared on the Project Gutenberg website. All texts were manually processed to remove extraneous verbiage such as chapter headings, illustration captions, and tables of contents. A complete list of the texts used appears in appendix B.

3.2 Representing dependencies in text

The texts to be classified were first parsed by the Stanford Parser [29], which was configured to output typed dependencies. In order to make the output of the parser more amenable to text processing, the typed dependencies were converted to a format I devised, which I call DepWords.

The DepWords format is a concise representation of the dependency relationships among words in a sentence. It was inspired in part by [2], which rendered syntactic information as "pseudo-words" that could be analyzed using the same methods used for actual words. In the DepWords format, each word in a text is replaced with a string of characters representing the chain of dependency types from that word to the sentence root.

The current version of the Stanford Parser (2.0.1 as of this writing) uses fifty-three different dependency types, including the `root` type that indicates a word is not dependent on any other. Since the number of types is one more than twice the number of letters in the Latin alphabet, a convenient shorthand was to assign an arbitrary upper or lowercase letter to each dependency type, and to use a pound sign (`#`) to represent the root. Each DepWord would begin with this root symbol, followed by one letter for each dependency relation leading up to the current word. For example, the root itself would be represented as `#`, while a word that depends directly on the root with the `nsubj` relation would be represented as `#c`. A word dependent on this word with the `nn` relation would be represented as `#ca`, and so forth. (A complete list of DepWords characters and their corresponding dependency types can be found in appendix A.)

For example, here is the first sentence of *Dead Men Tell No Tales*, by E. W. Hornung:

> Nothing is so easy as falling in love on a long sea voyage, except falling out of love.

The first few dependencies for this sentence produced by the Stanford Parser are as follows:

```
nsubj(easy-4, Nothing-1)
cop(easy-4, is-2)
advmod(easy-4, so-3)
root(ROOT-0, easy-4)
.  .  .
```

Finally, here is the DepWords representation of this sentence:

```
#c #O #D # #o #oi #oio #oioj #oio #oiojS #oiojF #oioja #oioj #oigr #oigR
#oig #oigq #oigo #oigoj #r .
```

(The end of each sentence is marked by a lone period.)

It is easy to extract a number of useful features from a text represented in the DepWords format. The average word length gives some indication of how complex a text's sentences tend to be, with longer chains of dependencies indicating greater complexity. A word frequency histogram tells us what the most common dependency relations are in a document. Word n-gram frequencies tell us what types of dependencies are likely to be found together in sentences. Information about particular types of dependencies can be obtained by filtering for words ending in the letter corresponding to the desired type.

3.3 Features

For each run of the experiment, the features extracted from each text consisted of the relative frequencies of a given style marker. Each run used a different style marker. The dependency-based style markers included DepWord units, bigrams, and trigrams, as well as simple dependency type units, bigrams, and trigrams. (The latter were extracted from the DepWords representation by ignoring all but the last letter in each DepWord, which would correspond to the dependency relation between that word and its immediate governor.)

Non-dependency-based style markers used included word units, bigrams, and trigrams; part of speech units, bigrams, and trigrams; and function words (from the list used in [31, 33]).

3.4 The classifier

The classification task was largely based on that used in [33], a simple method which nevertheless yielded impressive results. Each text in the test set was compared to each text in the training set according to the distance metric described in section 3.5. The resulting distances were fed to a 3-nearest-neighbor algorithm (as opposed to the nearest-neighbor algorithm used in [33]). A test text was assigned to the author of at least two of the three nearest training texts. If all three were by different authors, the test text was assigned to the nearest one.

3.5 The distance metric

The distance metric used in this experiment was *rank distance*, as described in [33]. Rank distance consists of the sum of the absolute values of the differences between rankings of each style marker. For example, if two documents are being compared based on word frequencies,

and the most common word in one text is the third most common word in the other, then the rank distance between those two texts is at least $|1 - 3| = 2$. Ties are assigned the arithmetic mean of their rankings. If the second and third place entries both have the same frequency, then each gets a ranking of 2.5.

Since the number of style markers found in the corpus could be quite huge — in the hundreds of thousands in the case of word trigrams — it was usually not feasible to compare the rankings of all of them. For that reason, only the most frequently occurring markers of any given type were used. In each run of the experiment, the top n style markers for the training set as a whole were calculated, and each pair of texts were compared according to their relative rankings for these n markers.

3.6 The experiment

Each run of the experiment used a different style marker (words, parts of speech, dependency types, or DepWords) grouped as either unigrams, bigrams, or trigrams, and considered the rankings of only the top n most frequent types (where n=10, 50, 100, 500, or 1000). A final set of runs used the set of function words from [31], and was thus similar to the experiment described in [33]. Ten-fold cross-validation was used to evaluate the learner: The set of texts was arbitrarily divided into ten folds of four texts each.[1] For each run of the experiment, the classifier was run ten times, each time using a different fold for the test texts and the other nine folds as the training texts. While not every fold contained texts from all three authors, the distribution of the texts guaranteed that for each run, all three authors would be featured in the training set (if not the test set). The results are given in table 3.1. Percentages refer to the percentage correct over all ten fold groupings.

[1]The list of texts in each fold can be found in appendix B, section B.4.

Table 3.1: Experimental results

Feature	Top 10	Top 50	Top 100	Top 500	Top 1000
Word 1-gram	80.0	97.5	97.5	97.5	97.5
Word 2-gram	80.0	97.5	97.5	97.5	97.5
Word 3-gram	87.5	97.5	97.5	97.5	97.5
POS 1-gram	87.5	95.0	95.0	95.0	95.0
POS 2-gram	85.0	97.5	95.0	97.5	97.5
POS 3-gram	85.0	92.5	95.0	97.5	97.5
Dependency 1-gram	72.5	95.0	95.0	95.0	95.0
Dependency 2-gram	85.0	92.5	97.5	97.5	97.5
Dependency 3-gram	80.0	95.0	97.5	97.5	97.5
DepWord 1-gram	45.0	92.5	95.0	95.0	95.0
DepWord 2-gram	82.5	92.5	95.0	97.5	97.5
DepWord 3-gram	90.0	97.5	95.0	97.5	97.5
Function Words	90.0	97.5	97.5	97.5	97.5

Note that a number of these runs are equivalent. For example, since there are only fifty-three dependency types, the Top 100, 500, and 1000 experiments for dependency 1-grams are identical. The same is true for function words, of which there are only seventy.

3.7 Results

As would be expected, the higher the number of features used to calculate rank distances, the better the classifier performed. The top 1000 of any kind of style marker gave a success rate of 97.5 percent, with the exception of part of speech, dependency, and DepWord units, all of which had a success rate of 95.0. None of the experiments had a 100% success rate, and, interestingly, the one text that was consistently misclassified was *In the Days of Drake* by J. S. Fletcher. This work was written in 1897, whereas most of the other works by that author were written in the 1920s; furthermore, it was not a work of detective fiction, but an historical novel set in the time of Sir Francis Drake. These two factors doubtless influenced

the work's style, and it is noteworthy that even the purely syntactic features reflected this difference.

Also of note is the fact that there is considerable variation among the different features as to how quickly they improve when the number of rankings considered is increased. It is hardly surprising that looking at only the top ten dependency or DepWord units gives poor results, since the ten most frequent types of dependencies used in a text would be expected to be fairly constant for a given language and not good indicators of style. However, when trigrams rather than individual units are considered, the top ten DepWords perform as well as or better than the top ten of all other types of features, at 90%.

The big winner here, however, is still function words. They give the same results as DepWord trigrams, except in the case where the top 100 most frequent features are considered, in which the accuracy of DepWord trigrams actually decreased slightly.

The fact that increasing the number of rankings considered can sometimes lead to a decrease in accuracy is important. It indicates that the classifier can be confused by extra information. The superior performance of function words, which yield a rather small set of features, also attests to this fact. More work is needed to determine which dependency types or DepWords make the best style markers.

3.8 A further trial: The Ad-Hoc Authorship Attribution Competition

In order to judge the effectiveness of dependency-based stylometry on a broader range of corpora, I tested the method described in this chapter on some of the problem sets from Patrick Juola's Ad-Hoc Authorship Attribution Competition [20]. Each problem featured its own training and test sets, and training data was not shared between tasks (or with the detective corpus experiment). I ran three experiments: one using the top 1000 DepWord

Table 3.2: DepWord, dependency, and function word performance on the Juola tasks

Problem set	# of authors	# of test texts	# correct (DepWord)	# correct (dependency)	# correct (function words)
A	13	13	1	1	5
B	13	13	1	3	5
C	5	9	8	8	9
D	3	4*	3	3	2
E	3	4*	2	2	2
G	2	4	3	2	4
H	3	3	1	1	1

*indicates one text was by an author not featured in the training set

trigrams, one using the top 1000 dependency trigrams, and one using the Mosteller and Wallace function words. I only ran the experiment for problems A, B, C, D, E, G, and H; the other problem sets featured either archaic English or languages other than English, and could not be used with the methods being tested. The results of the experiment are shown in table 3.2.

For most of the problems, dependency-based methods and function words fared about as well. For tasks A and B, which consisted of short student essays, all methods performed rather poorly, with function words faring better than the other methods. These problems featured more authors than any of the other problems, which may have made the authorship attribution task more difficult. The brevity of the samples probably had a great effect on accuracy, since it made all style markers rarer. The fact that function words are fairly common in even short texts may explain why they outperformed the other methods for these problems. For task H, which consisted of speeches, no method did better than one out of three. The rest of the tasks featured novels and plays, and here the dependency-based methods fared rather well. It should be noted that for tasks D an E, the test sets each contained one text that was not by any of the authors in the training set. These texts were

ignored, since the methods used here were not designed for outlier detection and had to assign each text to one of the known authors.

Chapter 4

Improving performance

4.1 Using box plots to find the best features

One way to improve the performance of a classifier is to reduce the dimensionality of its feature space. In the experiment described in chapter 3, dimensionality reduction was done in a rather naïve fashion: a histogram was constructed for the training set, and an arbitrary number of the most common n-grams were used to calculate rank distances. In general, accuracy increased as more n-grams were used. However, in one case, increasing the number of n-grams actually decreased accuracy: the top 100 DepWord trigrams performed slightly worse than the top 50 DepWord trigrams. This indicates that it is possible for excess features to confuse the classifier. The relatively poor performance of the classifier with only ten or fifty features might not be due to an insufficient number of features, but rather to too high a ratio of bad features to good.

A more intelligent method of feature selection would use only those features that do a good job of distinguishing among authors. Such a feature would tend to have a frequency ranking within a certain range for each author, and those ranges would differ significantly from author to author. Given the frequency ranking of one of these features in a test text,

one would then be able to determine with a fair amount of certainty which author wrote the text.

Features of the sort just described can be identified visually through the use of box plots. In a box plot, the lower and upper bounds of the box represent the 25th and 75th percentiles, respectively, and the line inside the box represents the median. For a given feature, we can produce side-by-side box plots of the rankings of that feature in the various training texts, each plot representing the results for a particular author. If there is a low degree of overlap among the various authors, the feature is a promising one.

Figure 4.1 shows a box plot for a promising feature, namely the DepWord trigram `#Z # #o`. This represents a pattern consisting of a negation term, followed by (and dependent on) the head, followed by a preposition also dependent on the head. Examples include "never heard of'," "not been in," and "never occurred to." Note that the boxes in the plots for the three authors do not overlap. E. W. Hornung's box is between 200 and 300, indicating that most of his works rank this feature fairly highly. J. S. Fletcher's box stretches from a little under 300 to a little over 400, while Sax Rohmer's begins above 400 where Fletcher's leaves off, and extends above 500. If this trigram were ranked around 220 in an unknown text, there is a strong possibility that text would have been authored by Hornung; if the trigram's rank were around 510, the text would more likely be Rohmer's. Of course, as the whiskers indicate, all of the authors have some individual texts for which this feature's rank strays within another author's range; these, however, are the exception rather than the rule.

Figure 4.2 shows a box plot for an unpromising feature. This is the DepWord trigram `# #yI #y`, representing a pattern consisting of the head followed by "to" and the infinitive form of a verb. Examples include "wanted to see," "going to stay," and "regretted to find." There is a great deal of overlap in this box plot. J. S. Fletcher's box fits completely inside E. W. Hornung's, which itself fits inside Sax Rohmer's. There is a portion of Rohmer's box from about 30 to 35 which lies outside the other two boxes; however, most of this area lies

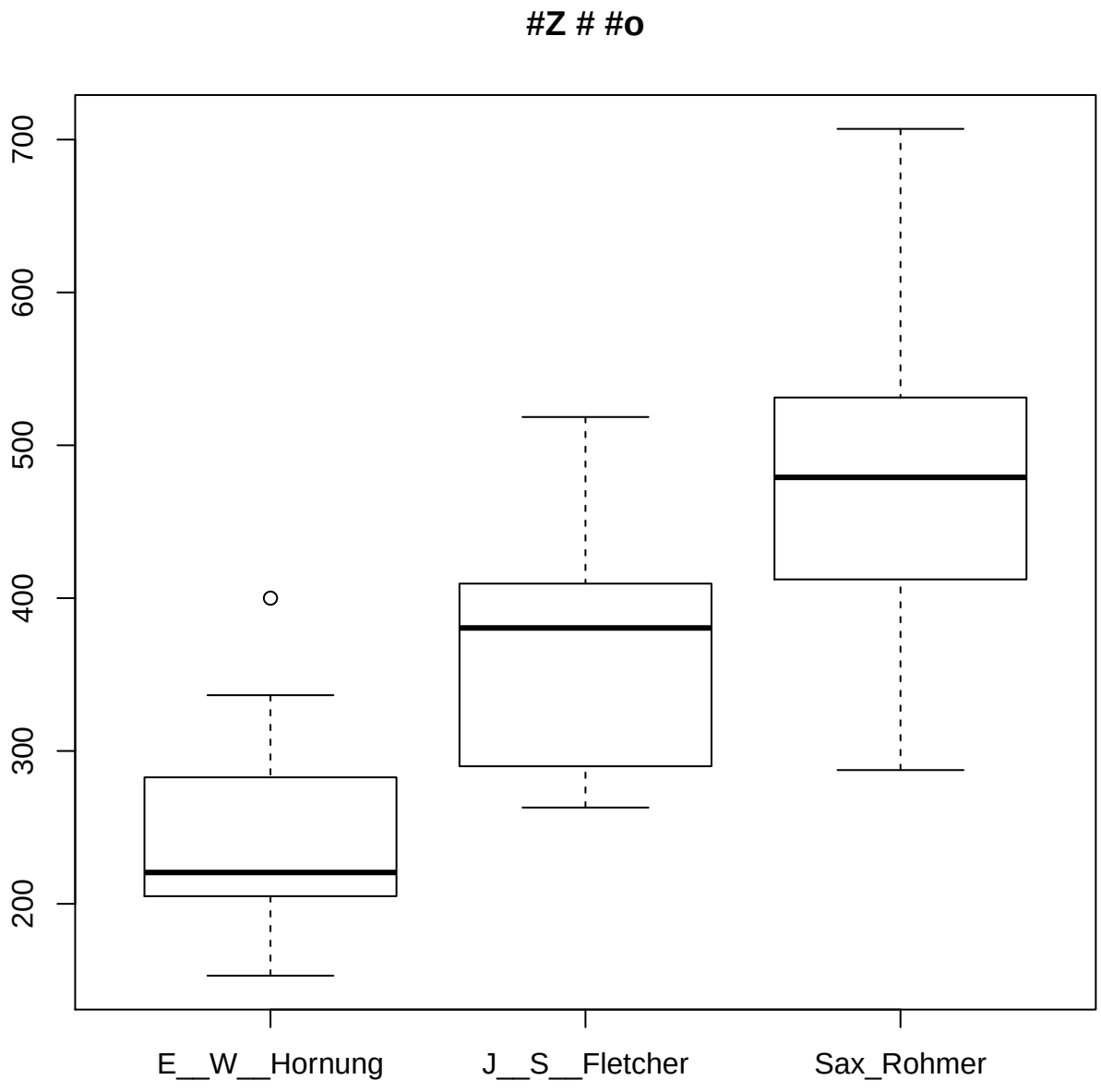

Figure 4.1: Example of a promising feature

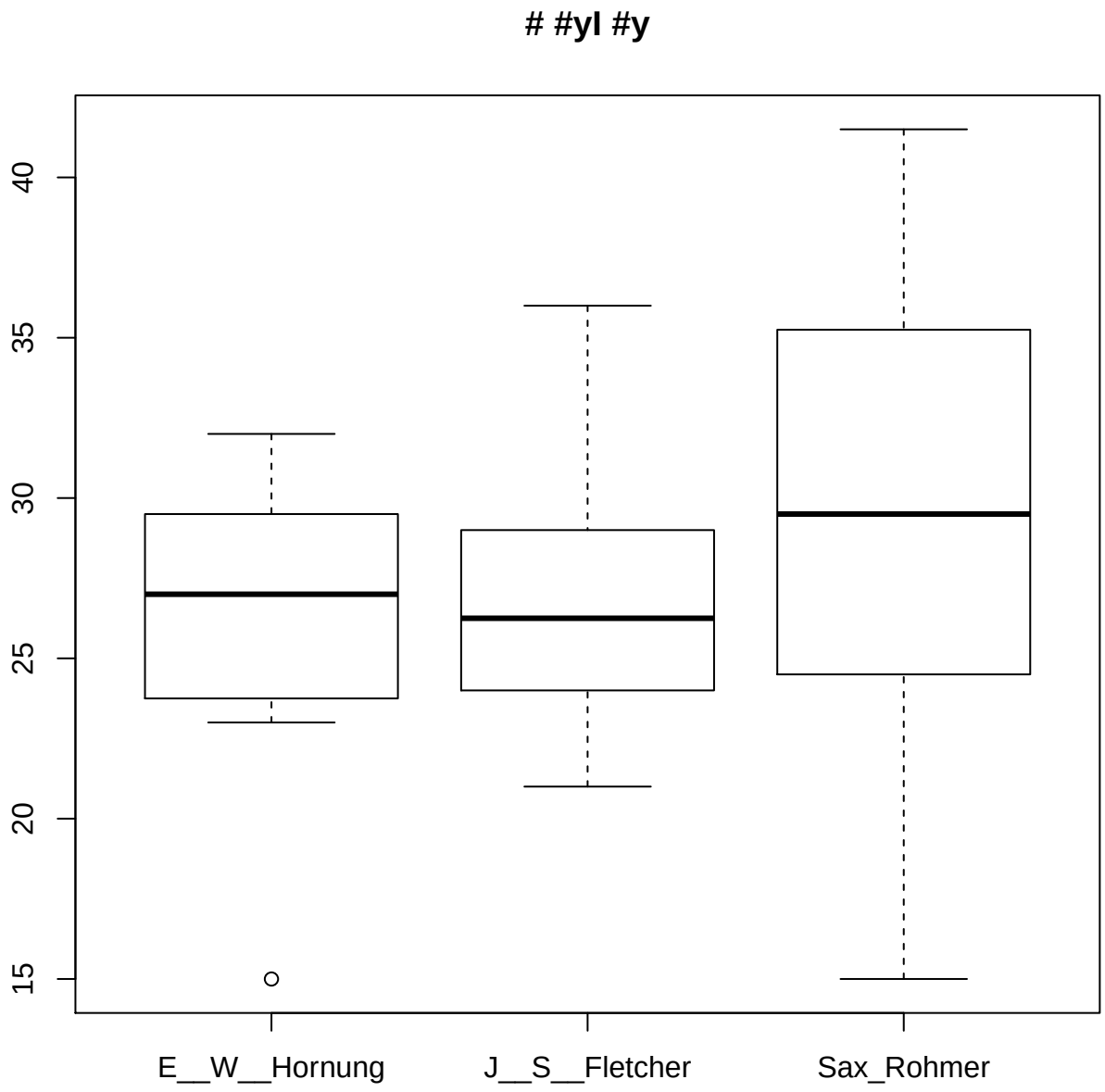

Figure 4.2: Example of an unpromising feature

inside Fletcher's whiskers. Given just the ranking for this feature in a test text, one could not draw a solid conclusion as to the authorship of the text.

Rather than attempt to determine the best features visually by looking at box plots to determine which had the least amount of overlap, I devised an algorithm for computing the *overlap ratio.* This is the ratio of the area of the boxes that overlap to the total area of the boxes. To determine this ratio, the graph is divided into a number of distinct areas along the tops and bottoms of the boxes. For example, in the graph shown in figure 4.2, the relevant areas extend from the top of Rohmer's box to the top of Hornung's, from the top of Hornung's to the top of Fletcher's, from the top of Fletcher's to the bottom of Rohmer's, from the bottom of Rohmer's to the bottom of Fletcher's, and from the bottom of Fletcher's to the bottom of Hornung's. Dividing the graph in this manner guarantees that none of the areas contain the tops or bottoms of any boxes.

Once the graph has been divided into relevant areas, the overlap area is computed as follows: Any area that is contained in only one box (i.e., the area from the top of Rohmer's to the top of Hornung's) is not counted. For any other area, its size is multiplied by the number of boxes containing it. The space from the top of Hornung's to the top of Fletcher's is multiplied by 2, since it falls within both Hornung's and Rohmer's boxes; the space from the top of Fletcher's to the bottom of Rohmer's is multiplied by 3, since it falls within all three boxes.

The overlap area is divided by the total area of all boxes to produce the overlap ratio. An overlap ratio of one indicates that all boxes occupy exactly the same space; an overlap ratio of zero indicates no overlap at all. The closer a feature's overlap ratio is to zero, the better that feature should be at indicating authorship.

4.2 A look at some of the best DepWord trigrams

The overlap ratio algorithm was run on the list of the 1000 most frequent DepWords trigrams in the detective corpus. Forty-seven of these were found to have overlap ratios of zero, indicating no overlap. This section examines some of these trigrams in order to give a better idea of the types of phrases that might serve as style markers.

4.2.1 `#TS #T #To`

The trigram in figure 4.3 represents determiner-noun-preposition groups in which the noun is directly dependent on the head. Examples include "a glass of," "a lot of," "the path between," and "a reputation for." This trigram occurs more frequently in the works of Sax Rohmer than in those of J. S. Fletcher, and less frequently in the works of E. W. Hornung. Also of interest is the fact that Fletcher's box is much smaller than Hornung's or Rohmer's, indicating the number of tokens of this feature per text is relatively constant for Fletcher.

4.2.2 `#r #LR #Lr`

The trigram in figure 4.4 represents a word between two punctuation marks. The word and the latter punctuation mark are both dependent on the clausal complement of the head, and the word's dependency type is simply `dep` – a type used by the Stanford Parser when it is unable to determine the precise type. A closer look at the texts reveals that this trigram almost always corresponds to an exclamation that begins a quoted sentence: an open-quote, followed by an "oh," "ah," or "well," followed by a dash or comma. Sax Rohmer uses these relatively sparingly, and is fairly consistent across his works. Hornung and Fletcher are more likely to use this type of phrase, and Fletcher shows a high degree of variability in the frequency with which he uses it. Since this phrase appears only in quoted dialogue, it might be useful to investigate whether the variability correlates with the amount of dialogue in

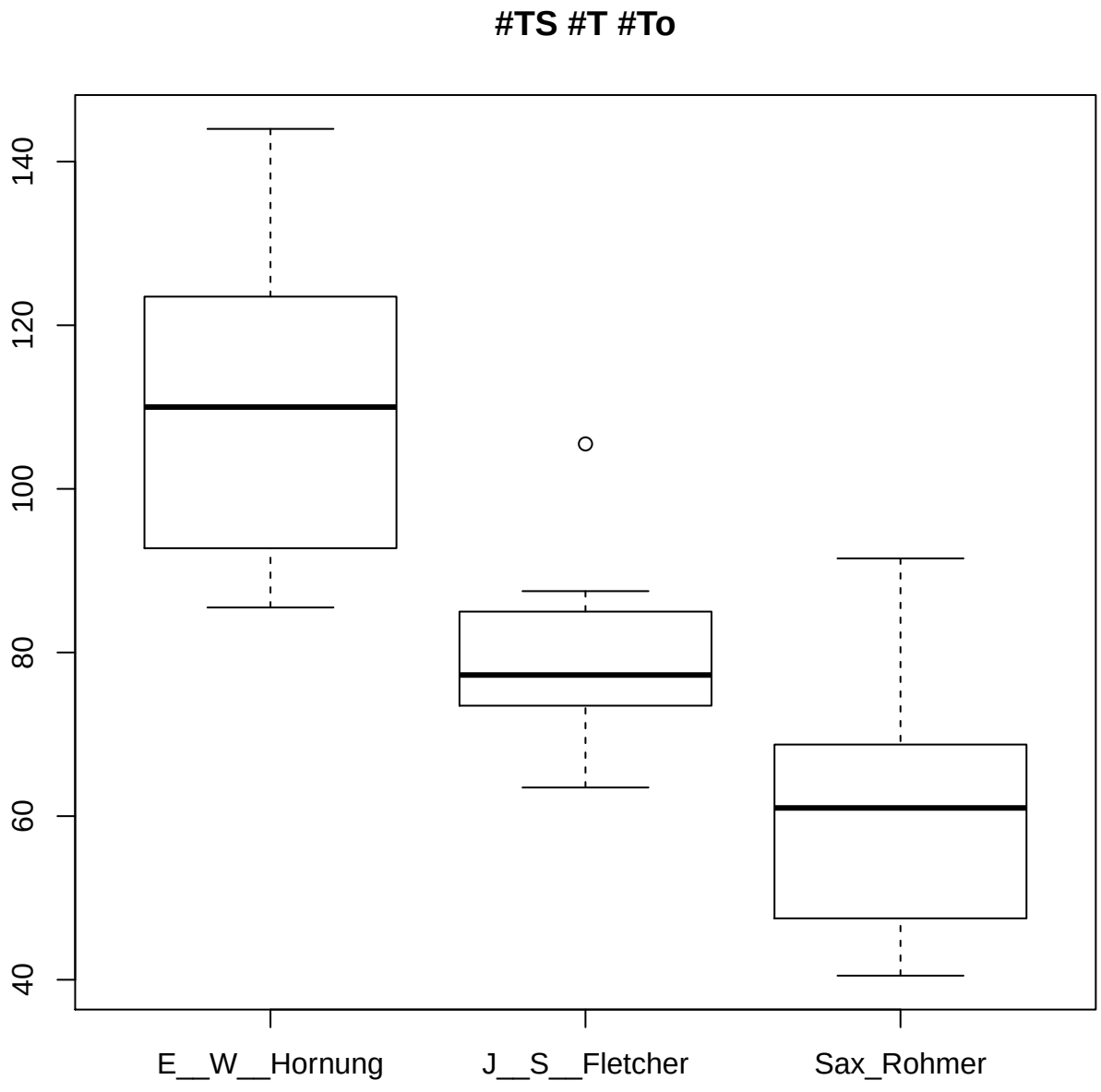

Figure 4.3: Box plots for DepWord trigram `#TS #T #To`

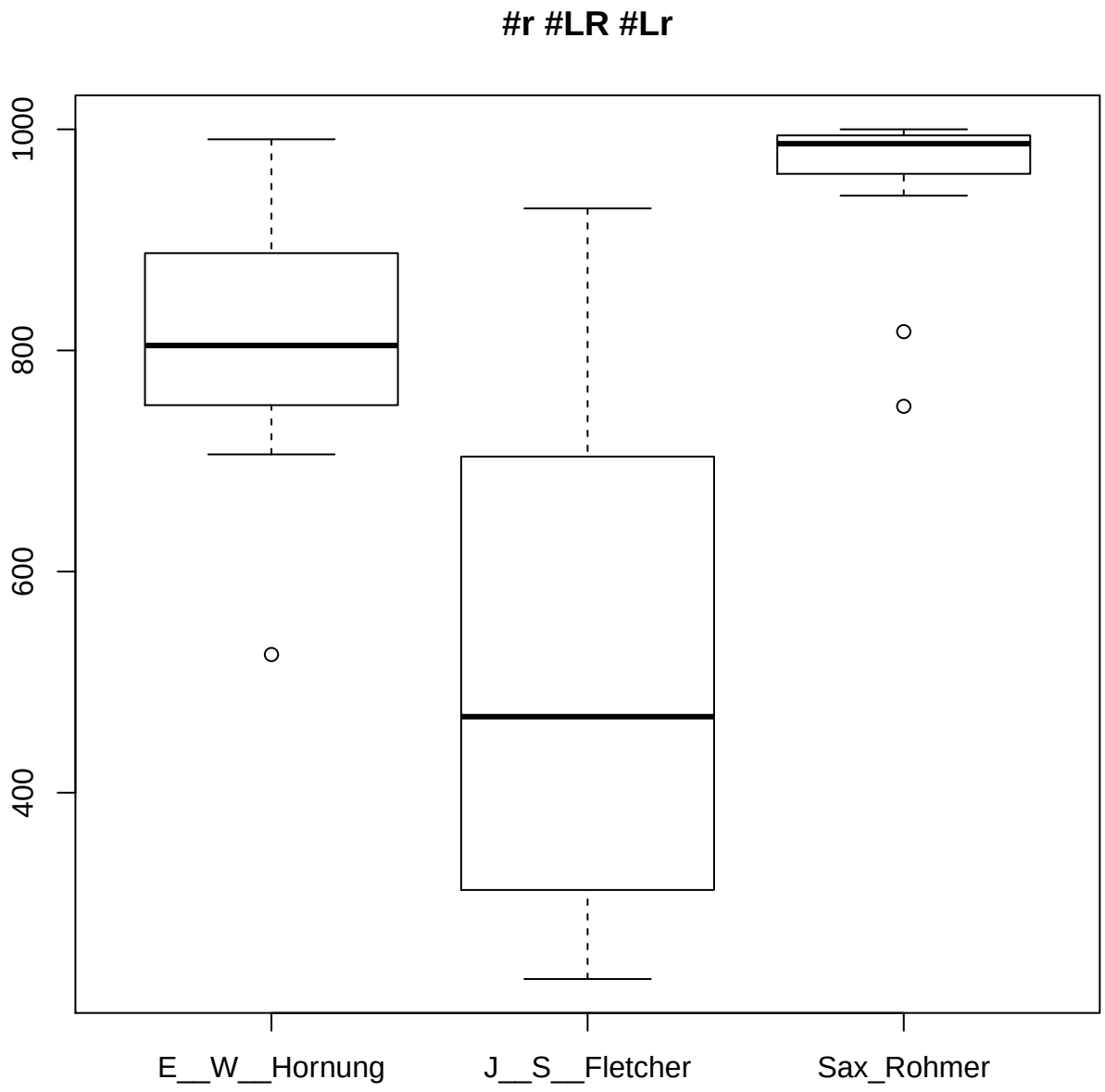

Figure 4.4: Box plots for DepWord trigram `#r #LR #Lr`

a story, since works that contain more quoted dialogue in general would be more likely to contain this particular type of phrase.

4.2.3 `#K #N #NTS`

The trigram in figure 4.5 represents a coordinating conjunction followed by a verb and a determiner. The particular syntactic structure represented is best shown by an example. Here is a representative sentence from *The Chestermarke Instinct* by J. S. Fletcher: "He produced a key, opened the door, *and motioned the* clerks to enter." For most sentences, the Stanford Parser chooses the main verb as the root. However, in this sentence there are three verbs (*produced, opened, motioned*) associated with the subject (*he*). The Stanford Parser deals with such sentences by selecting the first verb (*produced*) to be the root, and having subsequent verbs depend on the root with type `conj`. An unusually high frequency of trigrams of this type thus indicates a preference for such multi-verb sentences.

Sax Rohmer seems to be the big winner here, with most of his works ranking this trigram at 200 or better. He is also fairly consistent in his usage of the structure. E. W. Hornung seems to be less inclined to use the structure, but shows a great deal more variation across his works, in some cases even ranking this feature below 800.

4.2.4 `#ojo #ojojS #ojojF`

The trigram depicted in figure 4.6 does a particularly good job of illustrating how the Dep-Words format captures information that is missed by simple dependency types or parts of speech. It represents a particularly long series of dependencies from the root to the words in the trigram. Specifically, this phrase consists of a preposition followed by a determiner and an adjective, all dependent on the object of another prepositional phrase, itself dependent on the root. An example from *Fire Tongue* by Sax Rohmer: "He was recalled to his senses

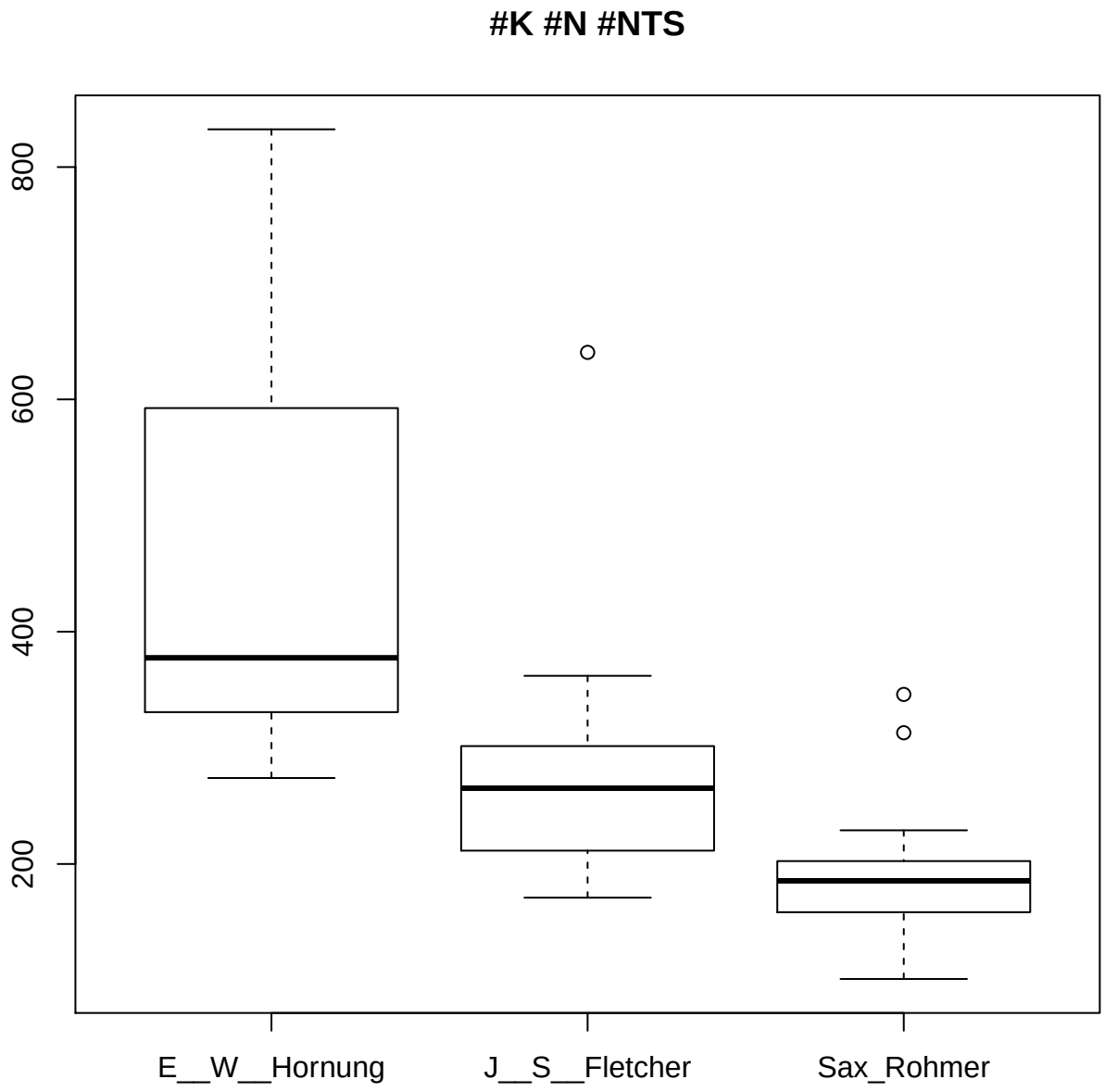

Figure 4.5: Box plots for DepWord trigram `#K #N #NTS`

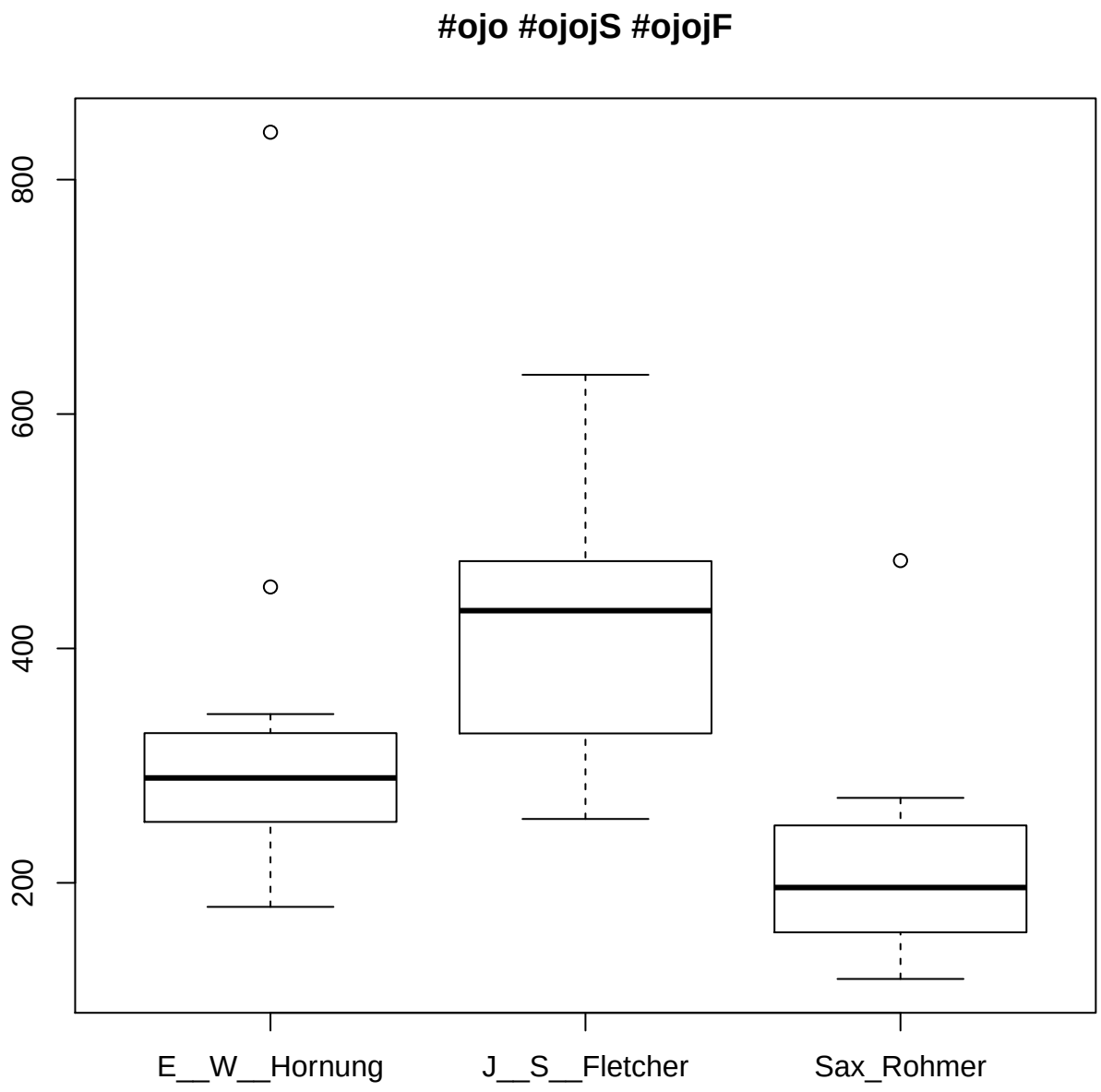

Figure 4.6: Box plots for DepWord trigram `#ojo #ojojS #ojojF`

by the voice *of the impudent* page." Sax Rohmer appears to favor this type of phrase more than the other two authors, with his box plot centered around 200. E. W. Hornung uses the phrase somewhat less frequently, and J. S. Fletcher least frequently of all. The three box plots do not overlap.

A much different distribution is found if we look at a very similar phrase. The box plot in figure 4.7 represents the trigram `#Nojo #NojojS #NojojF`. This phrase is identical to the former phrase, except that it is dependent on a conjunction instead of directly on the root. An example from *The Amateur Cracksman* by E. W. Hornung: "I had left my flat in town, and taken inexpensive quarters at Thames Ditton, on the plea *of a disinterested* passion for the river." J. S. Fletcher is much less likely to use this version of the phrase than the other two authors, and the box plots of Hornung and Rohmer mostly overlap. This phrase therefore does not do as good a job at distinguishing between Hornung and Rohmer. However, this phrase and the last phrase both have the same sequence of final dependencies, and the same sequence of parts of speech.

4.3 Performance

4.3.1 Experiments revisited

Subsequent experiments showed that the DepWord trigrams identified as most promising by the overlap method tended to outperform DepWord trigrams selected on the basis of frequency alone. In each of the experiments described in this section, an experiment from chapter 3 was repeated for DepWord trigrams, but instead of an arbitrary number of the most frequent trigrams, only the fifty most promising trigrams identified by the overlap method were used.

In the case of ten-fold cross-validation of the detective corpus, the improved DepWord trigrams were the only method to achieve a 100% success rate. This performance is less

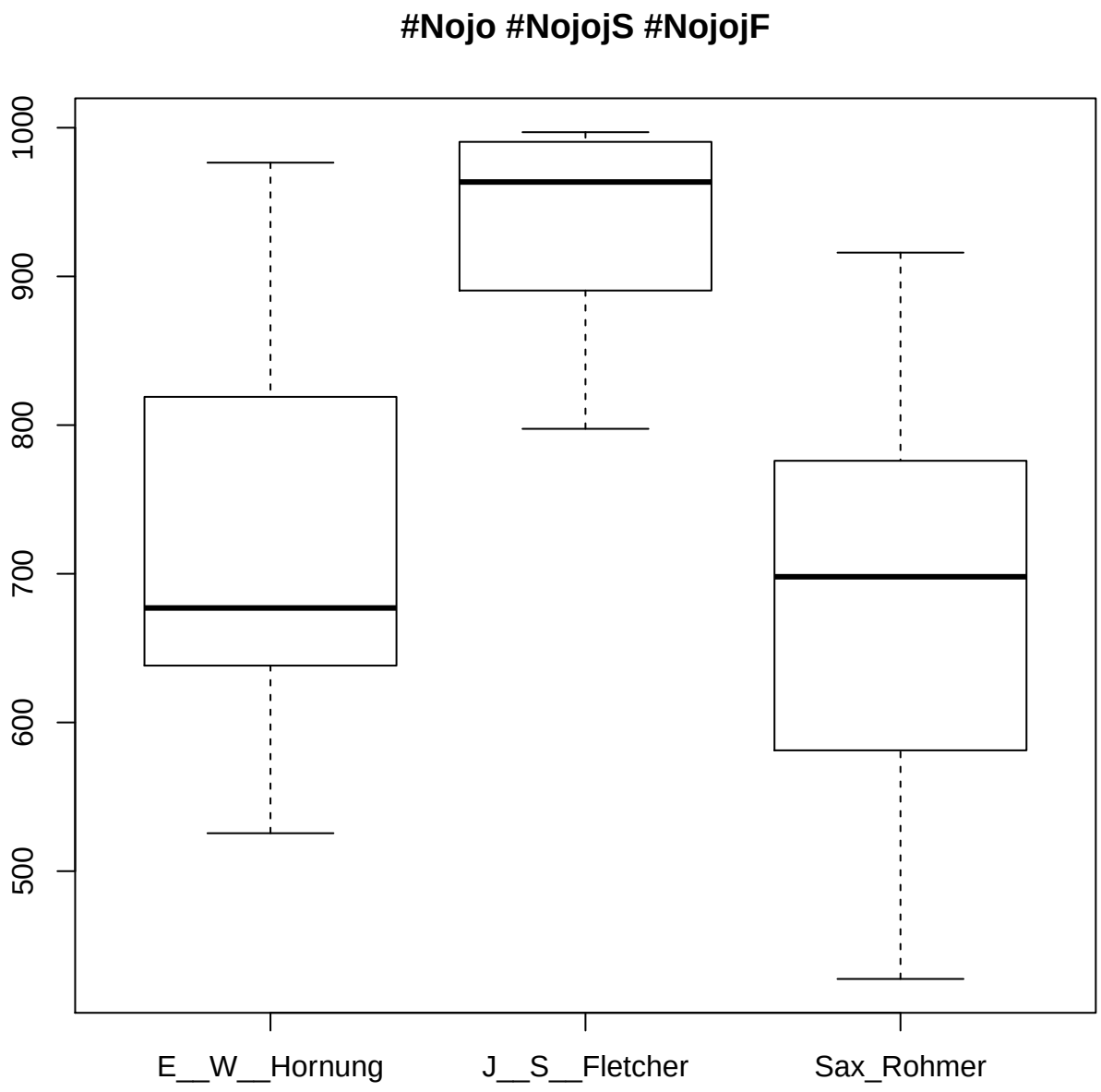

Figure 4.7: Box plots for DepWord trigram `#Nojo #NojojS #NojojF`

Table 4.1: Improved DepWord trigram performance on the Juola tasks

Problem set	# of authors	# of test texts	# correct (1000 most frequent)	# correct (50 best)
A	13	13	1	2
B	13	13	1	1
C	5	9	8	9
D	3	4*	3	2
E	3	4*	2	1
G	2	4	3	2
H	3	3	1	1

*indicates one text was by an author not featured in the training set

impressive than it might sound, because the best methods described in chapter 3 achieved 97.5% success. In those cases, only one text was misclassified, and in each case this text was J. S. Fletcher's *In the Days of Drake*. All this experiment shows, therefore, is that the improved DepWord trigrams did a better job of classifying one particularly difficult text. This achievement becomes even less impressive when one considers that the classification task featured the same set of works used to identify the promising trigrams in the first place, so there was a definite risk of overfitting.

When the classification tasks from Juola's authorship attribution challenge were re-run with improved DepWord trigrams, the results were inconclusive (table 4.1). The 50 best trigrams fared slightly better than the 1000 most frequent trigrams in problems A and C, did as well in problems B and H, and did slightly worse in problems D, E, and G. Since these experiments featured works, authors, and genres not found in the detective corpus, there was less possibility of overfitting, and the fact that performance was as good or better for most of the problems indicates that the improved DepWord trigrams are able to distinguish among authors beyond the detective corpus.

4.3.2 Distribution of texts in feature space

If we could see how the various texts are arranged in feature space, we could get an idea of how well the different types of style markers group the texts by author. One way to do this is to use multi-dimensional scaling to map the texts onto a two-dimensional space. Multi-dimensional scaling takes as its input a matrix of distances between points, and attempts to fit those points into an n-dimensional space while preserving distances as much as possible. A two-dimensional MDS plot provides a conveniet visual representation of distances between texts, since works that are close together on the plot are close together in feature space.

Figure 4.8 shows four two-dimensional MDS plots generated from distance data based on the thousand most frequent POS, word, dependency, and DepWord trigrams. The red circles represent works by J. S. Fletcher, the blue crosses works by Sax Rohmer, and the green triangles works by E. W. Hornung. We can see immediately that POS trigrams do a very poor job of distinguishing among authors in the detective corpus. Points for all three authors are dispersed seemingly randomly over the graph, with most of Fletcher's works appearing very close to Hornung's. Words do a better job, with three distinct and widely separated clusters, though a few works by Rohmer appear in the clusters for the other authors. The two dependency-based methods do the best job. For final dependencies, the only out-of-place text is a work by Fletcher that appears closer to Hornung's cluster. For DepWords, two of Fletcher's works inhabit a "no-man's land" apart from any of the clusters.

Figure 4.9 shows two-dimensional MDS plots for the same four features, using only the fifty best trigrams of each type as discovered by the overlap method. All four show definite improvement over the thousand most frequent trigrams, though some types show more improvement than others. POS does very well, with all texts clustered by author save a single Fletcher work that appears near the midpoint of the three groups. Words did especially poorly: even after the overlap method was applied, a number of texts by Rohmer wound up in the other authors' clusters, or in the middle. This is especially interesting given that

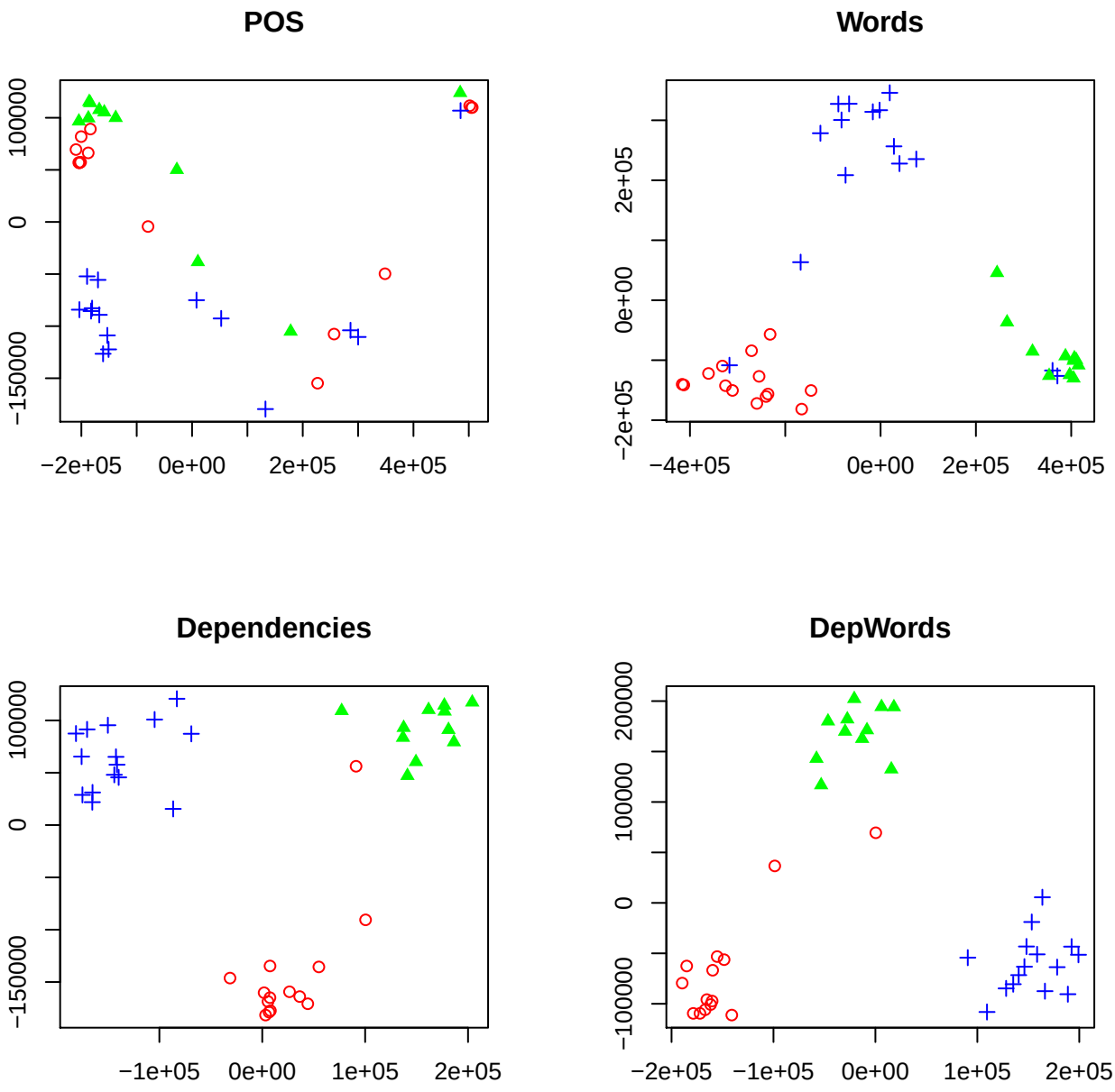

Figure 4.8: Two-dimensional metric MDS plots based on 1000 most frequent trigrams. Circle = Fletcher, Cross = Rohmer, Triangle = Hornung

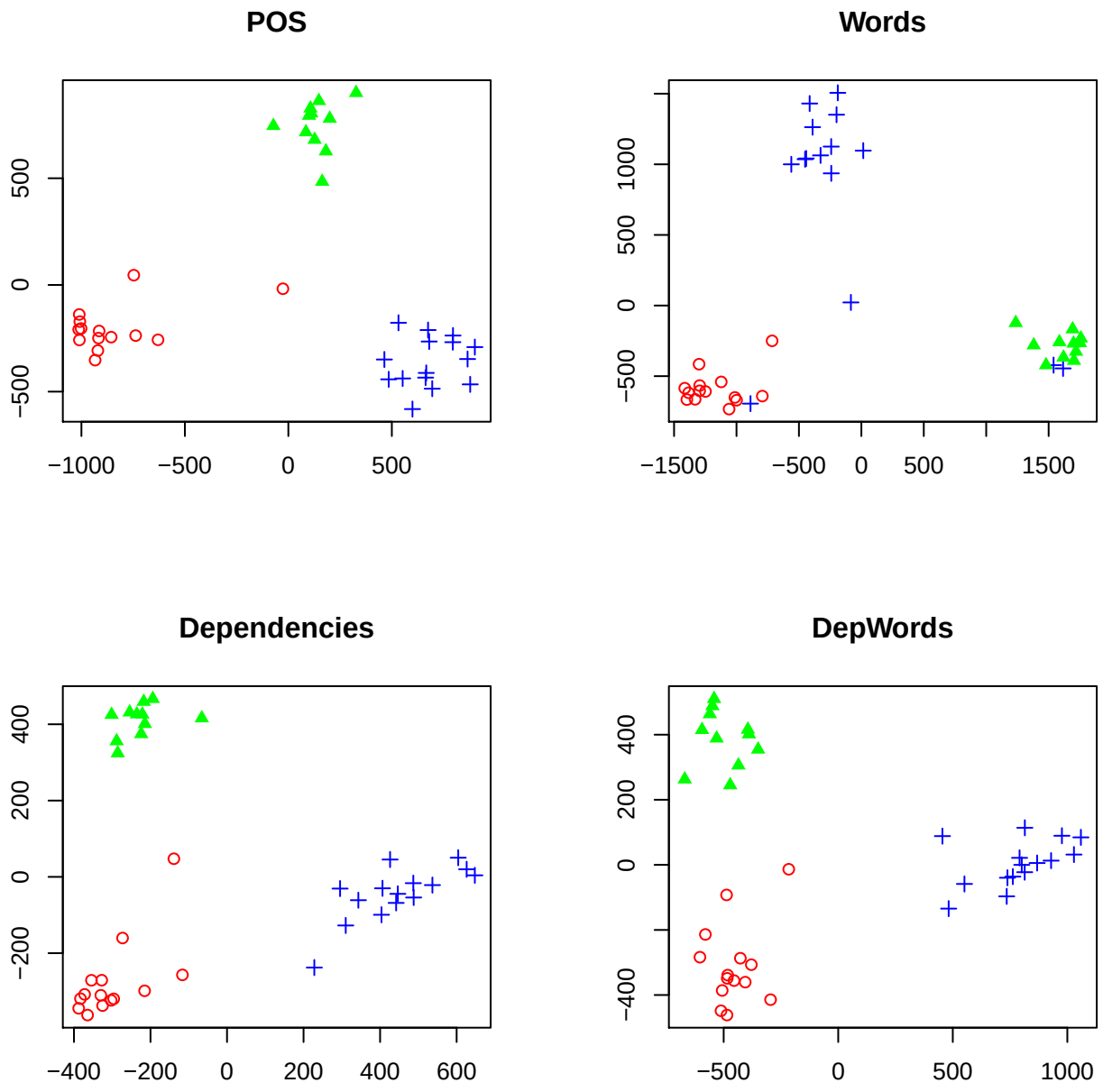

Figure 4.9: Two-dimensional metric MDS plots based on 50 best trigrams. Circle = Fletcher, Cross = Rohmer, Triangle = Hornung

there is obvious overfitting: one of the fifty best word trigrams is "said Raffles [comma]" — Raffles being the name of a recurring character in the works of E. W. Hornung. The two dependency-based features fared best, with all texts appearing fairly close to the centers of the clusters for their respective authors.

The MDS plots suggest that there may be an advantage to using syntactic information for author classification. Both final dependencies and DepWords did a better job of clustering by author even without intelligent feature selection, and the use of only the most promising trigrams was still not enough to make word trigrams as effective as dependency-based features. The fact that a character's name shows up among the best word trigrams also illustrates that purely lexical methods are apt to capture "style" markers that are really indicators of a work's subject matter or genre.

Chapter 5

Suggestions for further research

5.1 The "Long Tail" problem

One way of improving the method described in this work would be to pay more attention to rare DepWords and DepWord n-grams. The experiments described in the last two chapters concentrated on the thousand most frequent types of the style markers under investigation. One reason for this concentration was the fact that machine learning techniques do not generally do well with rare data. However, if DepWords follow the same Zipfian distribution as other linguistic phenomena, a very large percentage of the total features occur very infrequently. It is likely that many important style markers are lost when the features in the "long tail" are ignored.

Rather than concentrating on the most frequent n-grams, a DepWords-based classifier might concentrate on the rarest. There may be many distinctive phrases or patterns that do not occur in every work by a given author, but, when they do occur in a text, it is almost always a text by that author. A classifier that focused on rare patterns might work with a list of a thousand or so DepWords or phrases that are especially associated with particular authors. Instead of computing rank distances, the classifier would simply count how many

distinctive patterns occur at least once in a given text, and assign the text to the author for whom the greatest number of distinctive patterns were found.

This method has at least two advantages: First, since it is not frequency-based, it might do better with short texts for which frequencies are likely to be skewed. Second, it does not require us to pick any of the authors in the training set. If the number of distinctive patterns found is zero (or below a certain threshold), the classifier would classify the text as "unknown" or "none of the above." Such behavior would be particularly useful in cases where it is not known for certain that a text was written by one of a definite set of authors.

5.2 Longer *n*-grams

Better results might also be achieved by increasing the length of the phrases being investigated. The experiments in this work focused only on 1-, 2-, and 3-grams. My reasons for this restriction were simplicity and time; there is no reason why longer n-grams should not be considered. Indeed, Golcher [15] achieved respectable results using unrestricted substring data (i.e., all character n-grams of any length found in the texts). It could be that using DepWords n-grams of various lengths would yield better results.

5.3 The problem of overfitting

The overlap method that was used to find the most promising trigrams has a serious drawback: it produces the trigrams that do the best job of distinguishing among the authors in the training set, but does not necessarily generalize to larger sets of authors. As an illustration of the seriousness of this problem, if one runs the overlap algorithm on the 1000 most common word units in the detective corpus, one finds that one of the best discriminators is the word "Chinese." This word is more highly ranked in the works of Sax Rohmer, many

of whose novels feature a Chinese antagonist, Fu Manchu. Obviously, the frequent use of the word "Chinese" is not so much a marker of Rohmer's style as it is a reflection of the subject matter of his books. Intuitively, DepWord trigrams, which encode syntactic rather than semantic or lexical information, would be less likely to produce such coincidental correspondences. However, there is still a very real risk that purported style markers discovered using the overlap method are only effective for the particular training set used.

There are two different approaches to dealing with the overfitting problem, and the approach taken depends on the particular authorship attribution task being attempted. One approach is to train on a much larger, more diverse set of texts, with many more authors, subjects, and genres represented. This would increase the likelihood that the promising trigrams discovered are generally good at distinguishing authors' styles. Such an approach would be preferable when the texts to be classified might have been written by any of a large number of authors.

The other approach is, in effect, to embrace overfitting, and train on a small set of similar texts with a limited number of authors. This approach would be better suited for a task like classifying the disputed *Federalist* papers, in which there are only two authors who are considered likely candidates, and all the works are in the same genre and topic. For such a specific task, a narrower fit to the training data would be beneficial.

5.4 Speeding up the method

One area where there is much room for improvement is speed. The slowest and most computationally expensive part of the DepWords-based method is the dependency parsing necessary to generate the DepWords. Parsing the works in the detective corpus with the Stanford Parser took a few days on a reasonably fast machine.

Fortunately, there is a potential solution to the speed problem, at least for some tasks. The overlap method demonstrated that only certain key n-grams are particularly useful for distinguishing among authors. The rest of the n-grams are of little interest to us. This suggests that a *chunk parser,* which identifies only certain key phrases rather than producing a full parse, might be a better tool for the task.

If we opt for the first option described in section 5.3 — that of using a large, diverse corpus to identify some generally promising DepWord n-grams — we could then train a chunk parser to recognize these n-grams in texts. Since chunk parsers do not permit chunks to overlap, we would have to make sure none of the promising n-grams selected by the overlap method start with the same DepWord or DepWords that end another n-gram. There would be a large one-time expense, as the training set would have to be fully parsed, but any subsequent test texts could be parsed very quickly with the chunk parser, with perhaps a small reduction in accuracy.

5.5 Absolute versus relative rankings

A peculiarity about the way rankings are calculated in the present method may have affected the classifier's performance. Both the overlap method and function words involve the use of specially selected subsets of words or tags. At present, these subsets are ranked as though they are complete sets. For example, if one wanted to count a set of "function" words consisting only of "the," "of," and "elephant," those words would be given ranks of 1, 2, and 3 in each text (assuming there are no ties), even though "elephant" would likely rank far lower than the other two words in the complete list of words. Furthermore, "elephant" would probably rank third in every text, whether the text in question were a lengthy treatise on elephants or did not mention them at all. The word "elephant" is a sort of island, whose

position relative to the other words in the subset makes it effectively worthless for calculating rankings.

An alternate approach would be to calculate the rankings of all words and only use the rankings of the words that interest you for calculating rank distances. Such an approach would preserve distinctions among rankings that the current method overlooks. However, it would take longer, requiring the rankings of all features, rather than just a subset, to be counted. Furthermore, speed-up methods such as the chunk parsing described in section 5.4 could not be used because they only collect information about the desired subset of features.

5.6 Other grammar formalisms

While the DepWords format described in this work encodes syntactic information based on dependency grammar, there is no reason why other grammar formalisms should not be used. The choice of dependency grammars was motivated mainly by the simplicity of dependency parses, and the availability of relatively fast and reliable parsers. However, it would be a simple matter to replace the DepWords representation with one based on a different formalism.

One promising alternative to dependency grammars is *combinatory categorial grammars* (CCGs), developed by Mark Steedman [39, 40, 41]. This formalism, inspired by combinatory logic, uses a very simple set of combining rules with a rich collection of types. Each morpheme is assigned either a simple type (such as S or NP) or a complex type describing how the word may be combined with neighboring words. For example, a transitive verb might take type *(S \NP) / NP*, indicating that it combines with a noun phrase of type *NP* on its right to yield type *S \NP*, which in turn needs a noun phrase on its left to yield a complete sentence of type *S*. In CCGs, the type system does a lot of the work traditionally done by syntax rules in

other formalisms, and a word's type may contain information about that word's long-range dependencies.

CCG would be an attractive alternative to the DepWords format because, like DepWords, it attaches information about syntactic relations to the words themselves. The process by which complex categories are assigned to words is a form of "supertagging": using complex, linguistically motivated descriptions of words to simplify subsequent processing [6]. A good bit of work has already been done on automating the process of CCG supertagging [10, 12]. A classifier could use n-grams from the supertagger's output instead of DepWords for calculating rank distances and assigning authorship.

5.7 Other tasks

While the present work has focused solely on the task of authorship attribution, that is only one of the possible applications of syntactic stylometry. It might be worth investigating whether syntactic features could be used to classify texts based on other criteria, such as genre, topic, or reading difficulty.

Appendix A

The DepWords format

DepWords character	Stanford dependency name	Meaning
#	root	root of sentence
A	abbrev	abbreviation
B	acomp	adjectival complement
C	advcl	adverbial clause modifier
D	advmod	adverbial modifier
E	agent	agent
F	amod	adjectival modifier
G	appos	appositional modifier
H	attr	attributive
I	aux	auxiliary
J	auxpass	passive auxiliary
K	cc	coordination
L	ccomp	clausal complement
M	complm	complementizer
N	conj	conjunct

O	cop	copula
P	csubj	clausal subject
Q	csubjpass	clausal passive subject
R	dep	unknown dependent
S	det	determiner
T	dobj	direct object
U	expl	expletive (existential “there”)
V	infmod	infinitival modifier
W	iobj	indirect object
X	mark	marker
Y	mwe	multi-word expression
Z	neg	negation modifier
a	nn	noun compound modifier
b	npadvmod	noun phrase as adverbial modifier
c	nsubj	nominal subject
d	nsubjpass	passive nominal subject
e	num	numeric modifier
f	number	element of compound number
g	parataxis	parataxis
h	partmod	participial modifier
i	pcomp	prepositional complement
j	pobj	object of a preposition
k	poss	possession modifier
l	possessive	possessive modifier
m	preconj	preconjunct
n	predet	predeterminer

o	prep	prepositional modifier
p	prepc	prepositional clausal modifier
q	prt	phrasal verb particle
r	punct	punctuation
s	purpcl	purpose clause modifier
t	quantmod	quantifier phrase modifier
u	rcmod	relative clause modifier
v	ref	referent
w	rel	relative
x	tmod	temporal modifier
y	xcomp	open clausal complement
z	xsubj	controlling subject

Appendix B

The detective corpus

B.1 Works by J. S. Fletcher

In the Days of Drake, 1897

The Middle Temple Murder, 1919

Dead Men's Money, 1920

The Talleyrand Maxim, 1920

The Borough Treasurer, 1921

The Chestermarke Instinct, 1921

The Herapath Property, 1921

The Orange-Yellow Diamond, 1921

The Paradise Mystery, 1921

In the Mayor's Parlour, 1922

Ravensdene Court, 1922

The Middle of Things, 1922

The Rayner Slade Amalgamation, 1922

Scarhaven Keep, 1922

B.2 Works by E. W. Hornung

Dead Men Tell No Tales, 1897

The Amateur Cracksman, 1899

The Shadow of a Man, 1900

Raffles: Further Adventures of the Amateur Cracksman, 1901

The Shadow of the Rope, 1902

No Hero, 1903

Stingaree, 1905

A Thief in the Night, 1905

Mr. Justice Raffles, 1909

The Camera Fiend, 1911

Witching Hill, 1913

B.3 Works by Sax Rohmer

The Insidious Dr. Fu Manchu, 1913

The Sins of Severac Bablon, 1914

The Yellow Claw, 1915

The Devil Doctor, 1916

The Return of Dr. Fu Manchu,[1] 1916

The Hand of Fu Manchu, 1917

Brood of the Witch Queen, 1918

[1]This is the U. S. edition of *The Devil Doctor,* and differs only slightly from that work.

The Orchard of Tears, 1918

Dope, 1919

The Golden Scorpion, 1919

The Quest of the Sacred Slipper, 1919

The Green Eyes of Bast, 1920

Bat-Wing, 1921

Fire Tongue, 1921

Tales of Chinatown, 1922

B.4 Folds used for cross-validation

Fold 1:

Rohmer, *The Golden Scorpion*

Rohmer, *Tales of Chinatown*

Rohmer, *Fire Tongue*

Rohmer, *The Sins of Severac Bablon*

Fold 2:

Rohmer, *The Hand of Fu Manchu*

Rohmer, *Brood of the Witch Queen*

Rohmer, *The Orchard of Tears*

Rohmer, *The Yellow Claw*

Fold 3:

Rohmer, *The Quest of the Sacred Slipper*

Rohmer, *Dope*

Rohmer, *The Devil Doctor*

Rohmer, *The Insidious Dr. Fu Manchu*

Fold 4:

Rohmer, *The Green Eyes of Bast*

Rohmer, *The Return of Dr. Fu Manchu*

Rohmer, *Bat-Wing*

Hornung, *Mr. Justice Raffles*

Fold 5:

Hornung, *Witching Hill*

Hornung, *A Thief in the Night*

Hornung, *Dead Men Tell No Tales*

Hornung, *The Camera Fiend*

Fold 6:

Hornung, *The Amateur Cracksman*

Hornung, *The Shadow of a Man*

Hornung, *Raffles: Further Adventures of the Amateur Cracksman*

Hornung, *The Shadow of the Rope*

Fold 7:

Hornung, *No Hero*

Hornung, *Stingaree*

Fletcher, *In the Days of Drake*

Fletcher, *The Middle Temple Murder*

Fold 8:

Fletcher, *The Herapath Property*

Fletcher, *The Talleyrand Maxim*

Fletcher, *Scarhaven Keep*

Fletcher, *The Chestermarke Instinct*

Fold 9:

Fletcher, *In the Mayor's Parlour*

Fletcher, *Dead Men's Money*

Fletcher, *The Orange-Yellow Diamond*

Fletcher, *The Paradise Mystery*

Fold 10:

Fletcher, *Ravensdene Court*

Fletcher, *The Borough Treasurer*

Fletcher, *The Middle of Things*

Fletcher, *The Rayner Slade Amalgamation*

Bibliography

[1] Antosch, F.: The diagnosis of literary style with the verb-adjective ratio. In: *Statistics and Style,* L. Dolezel and R. W. Bailey (Eds.) New York: American Elsevier. (1969)

[2] Baayen, R., van Halteren, H., Tweedie, F.: Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. In: *Literary and Linguistic Computing,* vol. 11, no. 3, 121–131. (1996)

[3] Baayen, H.: *Word Frequency Distributions.* Kluwer Academic. (2001)

[4] Bailey, R. W. Authorship attribution in a forensic setting. In: *Advances in Computer-aided Literary and Linguistic Research,* D. E. Ager, F. E. Knowles, and J. Smith (Eds.) AMLC, Birmingham. (1979)

[5] Baker, J. C. P. A test of authorship based on the rate at which new words enter an author's text. In: *Journal of the Association for Literary and Linguistic Computing,* 3:36–39. (1988)

[6] Bangalore, S., Joshi, A. K. *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing.* MIT Press. (2010)

[7] Brainerd, B.: On the distinction between a novel and a romance: a discriminant analysis. In: *Computers and the Humanities,* 7:259–270. (1973)

[8] Brainerd, B.: *Weighing Evidence in Language and Literature: A Statistical Approach.* University of Toronto Press. (1974)

[9] Chaski, C.: Who's at the keyboard? Authorship attribution in digital evidence investigations. In: *International Journal of Digital Evidence,* vol. 4, no. 1. (2005)

[10] Clark, S., Curran, J. R. The importance of supertagging for wide-coverage CCG parsing. In: *COLING '04: Proceedings of the 20th International Conference on Computational Linguistics,* 282–288. (2004)

[11] Diederich, J., Kindermann, J., Leopold, E., Paass, G.: Authorship attribution with support vector machines. In: *Applied Intelligence,* vol. 19, 109–123. (2003)

[12] Espinoza, D., White, M., Mehay, D. Hypertagging: Supertagging for surface realization with CCG. In: *Proceedings of ACL-08: HLT*

[13] Gamon, M.: Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic data. In: *COLING '04: Proceedings of the 20th International Conference on Computational Linguistics.* (2004)

[14] Goldman, E., Allison, A.: Using grammatical Markov models for stylometric analysis. Class project, CS224N, Stanford University. Retrieved from: `http://nlp.stanford.edu/courses/cs224n/2008/reports/17.pdf` (2008)

[15] Golcher, F.: A new text statistical measure and its application to stylometry. In: *Proceedings of Corpus Linguistics.* (2007)

[16] Grant, T. D., Baker, K. L.: Identifying reliable, valid markers of authorship: A response to Chaski. In: *Forensic Linguistics,* vol. 8, no. 1, 66–79. (2001)

[17] Holmes, D. I.: Authorship attribution. In: *Computers and the Humanities,* vol. 28, no. 2, 87–106. (1994)

[18] Holmes, D. I.: The evolution of stylometry in humanities scholarship. In: *Literary and Linguistic Computing,* vol. 13, no. 3, 111–117. (1998)

[19] Holmes, D. I., Forsyth, R. S.: The *Federalist* revisited: New directions in authorship attribution. In: *Literary and Linguistic Computing,* vol. 10, no. 2, 111–127. (1995)

[20] Juola, P.: Ad-hoc authorship attribution competition. Retrieved from: `http://www.mathcs.duq.edu/~juola/authorship_contest.html` (2004)

[21] Juola, P.: Authorship Attribution. In: *Foundations and Trends in Information Retrieval,* vol. 1, no. 3, 233–334. (2006)

[22] Juola, P.: Authorship attribution for electronic documents. In: *Advances in Digital Forensics,* M. Olivier and S. Shenoi (Eds.), Springer, Boston, 119–130. (2006)

[23] Kaster, A., Siersdorfer, S., Weikum, G.: Combining text and linguistic document representations for authorship attribution. In: *SIGIR Workshop: Stylistic Analysis of Text for Information Access (STYLE).* MPI, Saarbrcken, 27–35. (2005)

[24] Kjell, B.: Authorship determination using letter pair frequency features with neural network classifiers. In: *Literary and Linguistic Computing,* vol. 9, no. 2, 119–124. (1994)

[25] Kjetsaa, G. *And Quiet Flows the Don* through the computer. In: *Association for Literary and Linguistic Computing Bulletin,* 7:248–256. (1979)

[26] Klein, D., Manning, C. D.: Accurate unlexicalized parsing. In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics,* 423–430. (2003)

[27] Koppel, M., Argamon, S., Shimoni, A. R.: Automatically categorizing written texts by author gender. In: *Literary and Linguistic Computing,* vol. 17, no. 4, 401–412. (2002)

[28] Levitsky, V., Melnyk, Y. P.: Sentence length and sentence structure in English prose. In: *Glottometrics* vol. 21, 14–24. (2011)

[29] Marneffe, M., MacCartney, B., and Manning, C. D.: Generating typed dependency parses from phrase structure parses. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation,* 449–454. (2006)

[30] Morton, A. Q.: The authorship of Greek prose. In: *Journal of the Royal Statistical Society,* 128:169–233. (1965)

[31] Mosteller, F., Wallace, D. L.: Inference and disputed authorship: *The Federalist.* Addison-Wesley, Massachusetts. (1964)

[32] Oakes, M. P.: Ant colony optimisation for stylometry: The *Federalist Papers*. In: *International Conference on Recent Advances in Soft Computing,* 86–91. (2004)

[33] Popescu, M., Dinu, L. P.: Rank distance as a stylistic similarity. In: *Coling 2008: Companion Volume – Posters And Demonstrations,* 91–94. (2008)

[34] Project Gutenberg. URL: `http://http://www.gutenberg.org`

[35] Raghavan, S., Kovashka, A., Mooney, R.: Authorship attribution using probabilistic context-free grammars. In: *Proceedings of the ACL 2010 Conference Short Papers,* 38–42. (2010)

[36] Stamatatos, E., Kokkinakis, G., Fakotakis, N.: Automatic text categorization in terms of genre and author. In: *Computational Linguistics,* vol. 26, no. 4, 471–495. (2000)

[37] Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Computer-based authorship attribution without lexical measures. In: *Computers and the Humanities,* vol. 35, no. 2, 193–214. (2001)

[38] Stamatatos, E.: A survey of modern authorship attribution methods. In: *Journal of the American Society for Information Science and Technology,* vol. 60, no. 3, 538–556. (2009)

[39] Steedman, M.: Combinatory grammars and parasitic gaps. In: *Natural Language and Linguistic Theory,* 403–439. (1987)

[40] Steedman, M.: *The Syntactic process.* MIT Press. (2000)

[41] Steedman, M.: *Taking Scope: The Natural Semantics of Quantifiers.* MIT Press. (2012)

[42] Tweedie, F. J., Singh, S., Holmes, D. I.: Neural network applications in stylometry: The *Federalist Papers*. In: *Computers and the Humanities,* vol. 30, 1–10. (1996)

[43] van Halteren, H.: Author verification by linguistic profiling: An exploration o the parameter space. In: *ACM Transactions on Speech and Language Processing,* vol. 4, no.1, 1–17. (2007)

[44] Yang, Y.: An evaluation of statistical approches to text categorization. In: *Information Retrieval Journal,* vol. 1, no. 1, 69–90. (1999)

[45] Yang, Y., Liu, X.: A re-examination of text categorization methods. In: *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99),* 42–49. (1999)

[46] Yule, G. U.: *The Statistical Study of Literary Vocabulary.* Cambridge University Press. (1932)