International Journal on Natural Language Computing (IJNLC) Vol.13, No.1, February 2024

Document Author Classification Using Parsed Language Structure

Todd K. Moon, Jacob H. Gunther
Electrical and Computer Engineering Department, Utah State University, Logan, Utah

Abstract

Over the years there has been ongoing interest in detecting authorship of a text based on statistical properties of the text, such as by using occurrence rates of noncontextual words. In previous work, these techniques have been used, for example, to determine authorship of all of *The Federalist Papers*. Such methods may be useful in more modern times to detect fake or AI authorship. Progress in statistical natural language parsers introduces the possibility of using grammatical structure to detect authorship. In this paper we explore a new possibility for detecting authorship using grammatical structural information extracted using a statistical natural language parser. This paper provides a proof of concept, testing author classification based on grammatical structure on a set of "proof texts," *The Federalist Papers* and *Sanditon* which have been as test cases in previous authorship detection studies. Several features extracted from the statistical natural language parser were explored: all subtrees of some depth from any level; rooted subtrees of some depth, part of speech, and part of speech by level in the parse tree. It was found to be helpful to project the features into a lower dimensional space. Statistical experiments on these documents demonstrate that information from a statistical parser can, in fact, assist in distinguishing authors.

Keywords:

author identification; natural language processing; statistical language parsing; stylometry.

1 Introduction and Background

There has been considerable effort over the years related to using statistical methods to identify authorship of texts, based on examples from candidate authors, in what is sometimes called "stylometry" or "author identification." Statistical analysis of documents goes back to Augustus de Morgan in 1851 [1, p. 282], [2, p. 166], who proposed that word length statistics might be used to determine the authorship of the Pauline epistles. Stylometry was employed as early as 1901 to explore the authorship of Shakespeare [3]. Since then, it has been employed in a variety of literary studies (see, e.g. [4, 5, 6]), including twelve of *The Federalist Papers* which were of uncertain authorship [7] — which we re-examine here — and an unfinished novel by Jane Austen —which we also re-examine here. Information theoretic techniques have also been used more recently [8]. Earlier work in stylometry has been based on "noncontextual words," words which do not convey the primary meaning of the text, but which act in the background of the text to provide structure and flow. Noncontextual words are at least plausible, since an author may address a variety of topics, so particular distinguishing words are not necessarily revealing of authorship. In noncontextual word studies, a set of most common words noncontextual is selected [2], and documents are represented by word counts, or ratios of word counts to document length. A review of the statistical methods is in [9]. As a variation, sets of ratios of counts of noncontextual word patterns to other word patterns are also employed [10]. Statistical analysis based on author vocabulary size *vs.* document length — the "vocabulary richness" — has also been explored [11]. For other related work, see [12, 13, 14, 15]

A more recent paper [16] considers the effectiveness of a wide variety of feature sets. Feature sets considered there include: vectors comprising frequencies of pronouns; function words (that is, articles, pronouns, particles, expletives); part of speech (POS); most common words; syntactic features (such as noun phrase, or verb phrase); or tense (e.g. use of present or past tense); voice (active of passive). In [16], feature vectors are formed from combinations of histograms, then reduced in dimensionality using a two-stage process of principle component analysis [17] followed by dimension reduction using linear discriminant analysis (LDA). In their LDA, the within-cluster scatter matrix is singular (due to the high dimension of the feature vectors relative the number of available training vectors), so their scatter matrix is

DOI: 10.5121/ijnlc.2024.13104

regularized. To test this, the authors consider a range of regularization parameters, selecting one which gives the best performance.

More recent work [18] mentions the survey in [15] in which commonly used features in the authorship field are word and character n-grams. As noted, there are risks the statistical methods might be biased by topic-related patterns. As [18] observe, “an authorship classifier (even a seemingly good one) might end up unintentionally performing topic identification if domain-dependent features are used. ... In order to avoid this, researchers might limit their scope to features that are clearly topic-agnostic, such as function words or syntactic features.” The work presented here falls in the latter category, making use of grammatical structures statistically extracted from the text. These appear to be difficult to spoof. Examination of other recent works [19, 20] indicate that there is ongoing interest in author identification methods, but none making use of the grammatical structures use here; there is a tendency to rely more on traditional n-grams.

In this work the feature vectors are obtained using tree information from parse trees from a natural language parsing tool [21]. These features were not among the features considered in [16]. The grammatical structures are, it seems, more subtle than simple counts of classes of words, and hence may be less subject to spoofing or topic bias, since it seems unlikely that an author intending to imitate another would be able to coherently track complicated patterns of usage, and the features do not include any words from the documents. It is found that the tree-based features perform better than the POS features on the test data considered.

The feature vectors so obtained can be of very high dimension, so dimension reduction is also performed here. However, to deal with the singularity of the within-cluster scatter matrix, a generalized SVD approach is used, which avoids the need to select a regularization parameter.

This paper provides a proof-of-concept of these tree-based features to distinguish authorship by applying them to documents which have been previously examined, *The Federalist Papers* and *Sanditon*. The ability to classify by authorship is explored for several feature vectors obtained from the parsed information.

2 Statistical Parsing and Extracted Features

Part-of-speech (POS) tagging classifies the words in a sentence according to their part of speech, such as noun,verb, or interjection. Because of the complexity of English language, there is potential for ambiguity. For example, many words (such as “seat” or “bat” or “eye”) can be either nouns or verbs. The ambiguity can be dealt with using statistical parsing, in which a large corpus of language is used to develop probabilistic models for words which are based on contextual words. These models are typically trained with the assistance of human linguistic experts. The parser used in this work uses a language model developed using the annotated corpus called the Penn Treebank, which is a corpus of over 7 million words of American English, collected from multiple sources, labeled using human and semi-automated markup [22, 23]. The parser is described in [21]. It is a probabilistic context free grammar (PCFG) parser [24], with language transition probabilities determined based on the Penn Treebank corpus. The parser software is known as the Stanford Parser [25]. Parsing results presented here are produced by version 4.2.0, released November 17, 2020.

Table 1 lists the POS labels (the POS tagset) associated with words when a sentence is parsed by this parser. It also lists the syntactic tagset, produced by the parser when doing grammatical parsing. (see [23, Table 1.1, Table 1.2], [26, Chapter 5]).

A brief introduction to statistical parsing is provide in Appendix A.

As an example of the parsing, consider the first sentence of *The Federalist Papers* 1 by Alexander Hamilton:

> After an unequivocal experience of the inefficacy of the subsisting federal government, you are called upon to deliberate on a new Constitution for the United States of America. (1)

Parsing this sentences yields the tree representation portrayed in figure 1(a). The leaf nodes correspond to the words of the sentence, each labeled with a POS. The non-leaf (interior) nodes represent syntactic (grammatical structure) information determined by the parser. The label of each node of the tree is referred to as a *token*. The parse tree can be represented using the text string shown in 1(b). This is formatted to show the various levels of the tree implied by the nesting of the parentheses in figure 1(c).

In preparing to extract feature vectors from a parse tree, some additional tidying-up is performed. The parser creates a ROOT node for each tree, which is therefore uninformative and is removed. Punctuation nodes in the tree, such as (, ,), (. .), or (. ?) are removed. Since the intent is to explore how the parsed information can be used for

Table 1: Penn Treebank POS Tagset and Syntactic Tagset.

POS tagset [23, Table 1.2]			
CC	coordinating conjunction (and,but, or)	CD	cardinal number (one, two, three)
DT	determiner (a, the)	EX	existential “there”
FW	foreign word	IN	preposition or subordinating conjunction (of, in, by)
INTJ	interjection	JJ	adjective (yellow)
JJR	adjective, comparative (bigger)	JJS	adjective, superlative (biggest)
LS	list item marker (1, 2, one)	MD	modal (can, should)
NN	noun, singular or mass (llama, snow)	NNS	noun, plural (llamas)
NNP	proper noun, singular (IBM)	NNPS	proper noun, plural (Carolinas)
PDT	predeterminer (all, both)	POS	possessive ending (’s)
PRP	personal pronoun (I, you, he)	PRP$	possessive pronoun (your, one’s)
RB	adverb (quickly, never)	RBR	adverb, comparative (faster)
RBS	adverb, superlative (fastest)	RP	particle (up, off)
SYM	symbols (+, %, &)	TO	“to”
UH	interjection (ah, oops)	VB	verb, base form (eat)
VBD	verb, preterite (past tense) (ate)	VBG	verb, gerund (eating)
VBN	verb, past participle (eaten)	VBP	verb, non-3sg pre (eat)
VBZ	verb, 3sg pres (eats)	WDT	wh-determiner (which, that)
WP	wh-pronoun (what, who)	WP$	possessive WH- (whose)
WRB	wh-adverb (how, where)		
$	dollar sign	#	pound sign
“	left quote	”	right quote
(	left parenthesis	)	right parenthesis
,	comma	.	sentence-final (. ! ?)
:	mid-sentence punc (: : … – -)	″, ‘, ’	straight double quote; left single open quote, right single close quote
Syntactic Tagset [23, Table 1.1]			
ADJP	Adjective phrase	ADVP	Adverb phrase
NP	Noun phrase	PP	Prepositional phrase
S	Simple declarative clause	SBAR	Subordinate clause
SBARQ	Direct question introduced by *wh*-element	SINV	Declarative sentence with subject-aux inversion
SQ	Yes/no questions and subconstituent SBARQ excluding *wh-element*	VP	Verb phrase
WHADVP	Wh-adverb phrase	WHNP	Wh-noun phrase
X	Constituent of unknown or uncertain category	*	“Understood” subject of infinitive or imperative
0	Zero variant of *that* in subordinate clauses	T	Trace of wh-Constituent

classification, rather than the words of the document, the words of the sentence are removed from the parse tree. With these edits, the sentence (1) has the parsed representation

$$\begin{aligned}&\text{(S(PP(IN)(NP(NP(DT)(JJ)(NN))(PP(IN)(NP(NP(DT)}\\&\text{(NN))(PP(IN)(NP(DT)(JJ)(JJ)(NN)))))))}\\&\text{(NP(PRP))(VP(VBP)(VP(VBN)(PP(IN))}\\&\text{(S(VP(TO)(VP(VB)(PP(IN)(NP(NP(DT)(JJ)(NN))}\\&\text{(PP(IN)(NP(NP(DT)(NNP)(NNP))(PP(IN)(NP(NNP))))))))))))}\end{aligned} \tag{2}$$

From this prepared data, various feature vectors were extracted, as described below. (The text manipulation and data extraction was done using the Python language, making extensive use of Python's dictionary type. The parsed string (2) can be used, for example, as a key to a Python dictionary.)

3 Parse Tree Features

The richness of the parsed representation introduces the possibility of many different feature vectors. Of the many possible feature vectors that might be chosen, four are discussed here. Examples are provided based on the sentence above to illustrate the features.

All Subtrees One set of features is the set of all subtrees of a given depth encountered among all the parsed sentences. For example, Figure 2 shows eleven subtrees of depth 3 extracted from (2). Subtrees of a given depth may appear more than once within a sentence. For example, the subtree

(NP(NP(DT)(JJ)(NN))(PP(IN)(NP(NP)(PP))))

appears twice in (2).

Across all the sentences in the documents considered, there is a very large number of subtrees. This leads to vectors of very high dimension. This is a problem that is dealt with later.

Rooted Subtrees A rooted subtree is a subtree of a tree whose root node is the root node of the overall tree, down to some specified level. The first few rooted subtrees can be thought of summarizing the general structure of a sentence, with the amount of detail in the summary related to the number of levels of the subtree. Fig. 3 illustrates the subtrees of levels one, two, and three for the tree of Fig. 1.

Part-of-Speech A simple set of features ignores the tree structure, and simply extracts the counts of tokens in the parse tree. For (2), the counts of the POS are

S	P	IN	NP	DT	JJ	NN	PRP	VP	VBP	VBN	TO	VB	NNP
2	7	7	11	5	4	4	1	4	1	1	1	1	3

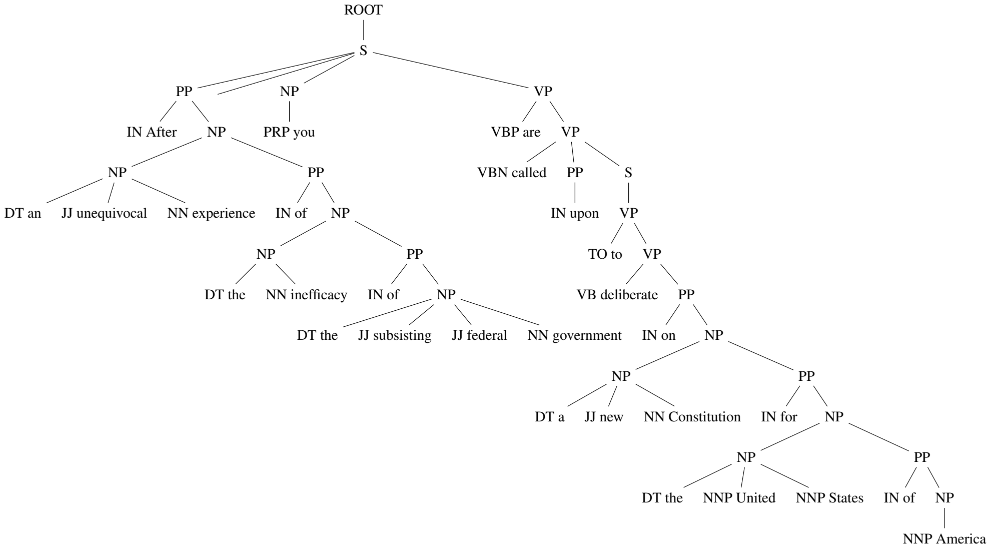
POS by Level A more complicated set of features is the histogram of tokens at each level of the tree. For the tree of (2), this is shown in Table 2.

For purposes of author classification, the idea, of course, is to see how the patterns in the feature vectors obtained from the sentences of one author compare with the patterns in the feature vectors of other authors.

4 Classifier

This section describes the basic operation of the classifier employed in the tests for this paper. In this paper, when "classes" are referred to, it refers to the different authors under consideration. Let k denote the number of classes (authors).

Let n_i denote the number of documents associated with author i, $i = 1, 2, \ldots, k$. Let $\mathbf{v}_{i,j} \in \mathbb{R}^m$ denote a feature vector associated with a document (e.g. a normalized histogram of the all subtrees counts for a document). The set of all feature vectors for author i is formed by the columns the $m \times n_i$ matrix $V_i = \begin{bmatrix}\mathbf{v}_{i,1} & \mathbf{v}_{i,2} & \ldots & \mathbf{v}_{i,n_i}\end{bmatrix}$, $i = 1, 2, \ldots, k$.



(a) Graphical representation of parse tree

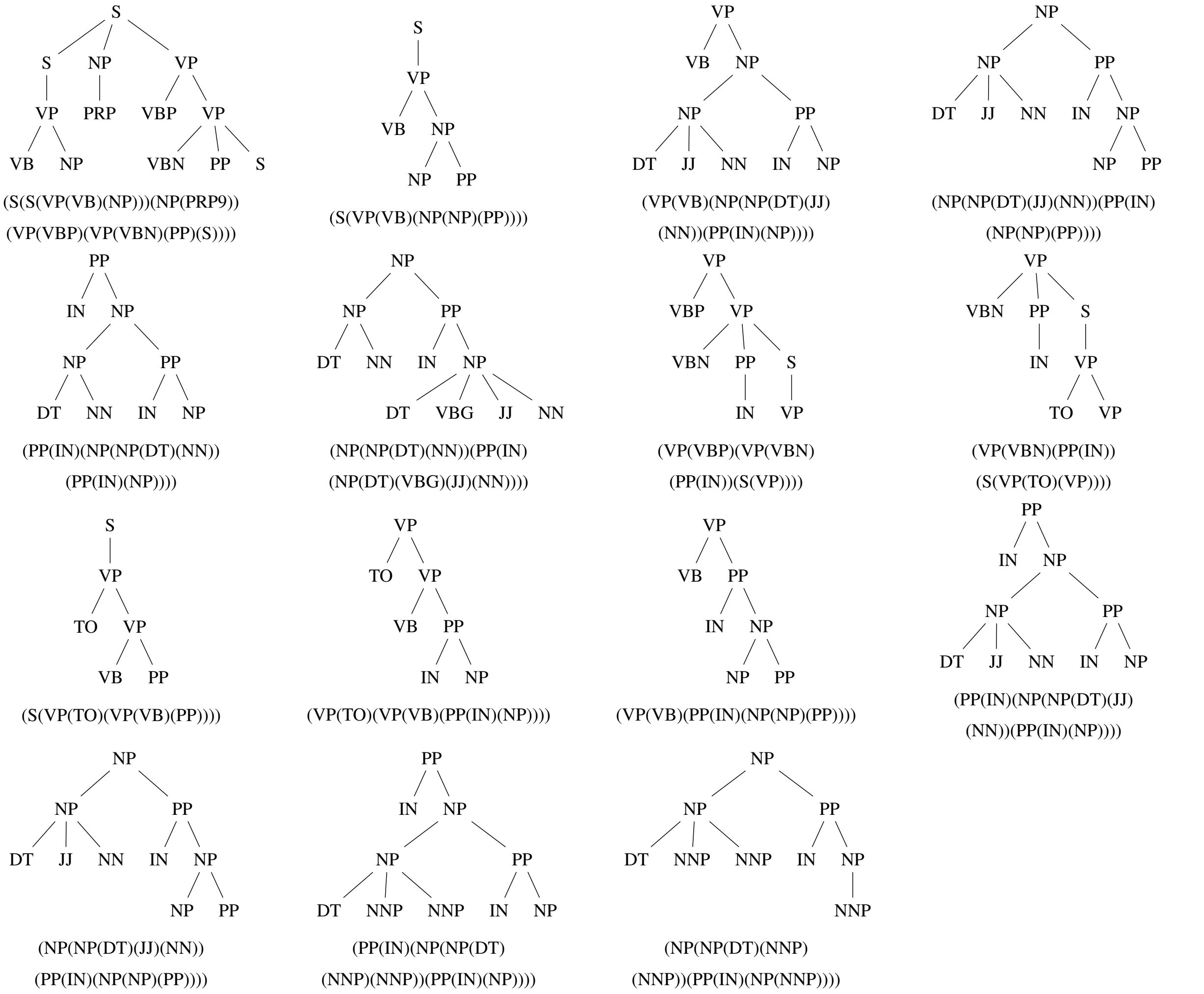
```
(ROOT(S (PP(IN
After)(NP(NP(DT an)(JJ
unequivocal)(NN
experience))(PP(IN
of)(NP(NP(DT the)(NN
inefficacy)) (PP(IN
of)(NP(DT the)(JJ
subsisting)(JJ federal)(NN
government)))))))(,
,)(NP(PRP you))(VP(VBP
are) (VB(VBN called)(PP(IN
upon))(S(VP(TO to) (VP(VB
deliberate) (PP (IN on) (NP
(NP(DT a)(JJ new)(NN
Constitution)) (PP(IN
for)(NP(NP(DT the)(NNP
United)(NNP States))(PP(IN
of)(NP(NNP
America))))))))))))))
```

(b) Textual representation of parse tree

```
(ROOT
  (S
    (PP
      (IN After)
      (NP
        (NP
          (DT an)(JJ unequivocal)(NN experience))
        (PP
          (IN of)
          (NP
            (NP
              (DT the)(NN inefficacy))
            (PP
              (IN of)
              (NP
                (DT the)(JJ subsisting)(JJ federal)
                (NN government)))))))
    (, ,)
    (NP
      (PRP you))
    (VP
      (VBP are)
    (VB
        (VBN called)
        (PP
          (IN upon))
        (S
          (VP
            (TO to)
          (VP
              (VB deliberate)
          (PP
              (IN on)
          (NP
              (NP
                    (DT a)(JJ new)(NN Constitution))
              (PP
                    (IN for)
              (NP
                    (NP
                        (DT the)(NNP United)(NNP States))
                    (PP
                        (IN of)
                        (NP
                          (NNP America))))))))))))))
```

(c) Formated textual representation of parse tree

Figure 1: Example parse tree



(S(S(VP(VB)(NP)))(NP(PRP9))
(VP(VBP)(VP(VBN)(PP)(S))))

(S(VP(VB)(NP(NP)(PP))))

(VP(VB)(NP(NP(DT)(JJ)
(NN))(PP(IN)(NP))))

(NP(NP(DT)(JJ)(NN))(PP(IN)
(NP(NP)(PP))))

(PP(IN)(NP(NP(DT)(NN))
(PP(IN)(NP))))

(NP(NP(DT)(NN))(PP(IN)
(NP(DT)(VBG)(JJ)(NN))))

(VP(VBP)(VP(VBN)
(PP(IN))(S(VP))))

(VP(VBN)(PP(IN))
(S(VP(TO)(VP))))

(S(VP(TO)(VP(VB)(PP))))

(VP(TO)(VP(VB)(PP(IN)(NP))))

(VP(VB)(PP(IN)(NP(NP)(PP))))

(PP(IN)(NP(NP(DT)(JJ)
(NN))(PP(IN)(NP))))

(NP(NP(DT)(JJ)(NN))
(PP(IN)(NP(NP)(PP))))

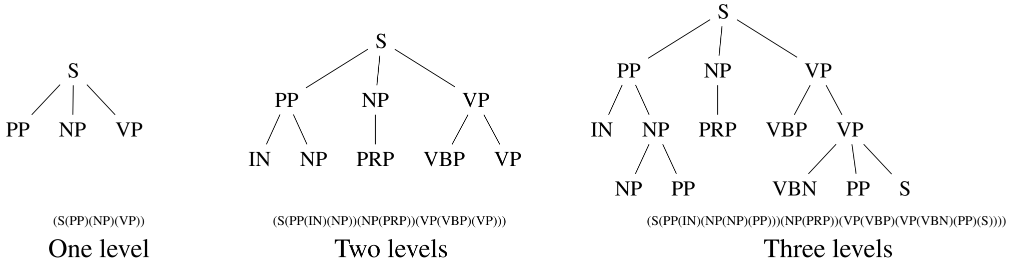
(PP(IN)(NP(NP(DT)
(NNP)(NNP))(PP(IN)(NP))))

(NP(NP(DT)(NNP)
(NNP))(PP(IN)(NP(NNP))))

Figure 2: Some subtrees of depth 3 extracted from the tree in (2)

Table 2: POS counts by level for the tree (2).

Level	Counts					
1	S: 1					
2	PP: 1	NP: 1	VP : 1			
3	IN : 1	NP: 1	PRP: 1	VBP: 1	VP: 1	
4	NP: 1	PP: 1	VBN: 1	PP: 1	S: 1	
5	DT: 1	JJ: 1	NN: 1	IN: 1	NP: 1	IN: 1	VP: 1
6	NP: 1	PP: 1	TO: 1	VP: 1			
7	DT: 1	NN: 1	IN: 1	NP: 1	VB:1	PP: 1	
8:	DT: 1	JJ: 2	NN: 1	IN: 1	NP: 1		
9:	NP: 1	PP: 1					
10:	DT:1	JJ: 1	NN: 1	IN: 1	NP: 1		
11:	NP: 1	PP: 1					
12:	DT: 1	NNP: 1	IN: 1	NP: 1			
13:	NNP: 1						


S
PP NP VP
(S(PP)(NP)(VP))
S
PP NP VP
IN NP PRP VBP VP
(S(PP(IN)(NP))(NP(PRP))(VP(VBP)(VP)))
S
PP NP VP
IN NP PRP VBP VP
NP PP VBN PP S
(S(PP(IN)(NP(NP)(PP)))(NP(PRP))(VP(VBP)(VP(VBN)(PP)(S))))

One level

Two levels

Three levels

Figure 3: Rooted Subtrees of the tree in (2) of one, two, and three levels

An overall $m \times n$ data matrix is formed as $V = \begin{bmatrix} V_1 & V_2 & \dots & V_k \end{bmatrix}$, where $\sum_{i=1}^{k} n_i = n$. Let N_i denote the set of column indices of V associated with the vectors in class i. The centroids (that is, the mean of the set of vectors) of the feature vectors for each class are computed by

$$\mathbf{c}^{(i)} = \frac{1}{n_i}\sum_{j=1}^{n_i} \mathbf{v}_{i,j} \text{ and the overall centroid is } \mathbf{c} = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n_i} \mathbf{v}_{i,j}.$$

In the tests performed for the investigation in this paper, the classifier works as follows (see figure **??**).

- For each feature vector under consideration $\mathbf{v} = \mathbf{v}_{i,j} \in V_i$ coming from class i, the vector $\mathbf{v}_{i,j}$ is removed from the pool of vectors in V_i, producing a set of vectors $\widetilde{V_i}$ and the centroid $\tilde{\mathbf{c}}_i$ of the resulting data is computed:

  $$\tilde{\mathbf{c}}^{(i)} = \frac{1}{n_i - 1}\sum_{i \in N_i \backslash j} V_i.$$

  Centroids for all the other classes are computed, but without removing the vector under consideration, so $\tilde{\mathbf{c}}^{(j)} = \mathbf{c}^{(j)}$.

- The vector $\mathbf{v}$ under consideration is compared with the class centroid for each class, and the estimated class is that class whose centroid is closest to $\mathbf{v}$, where the distance measure is simply Euclidean distance:

  $$\hat{i} = \arg\min_{j} \|\mathbf{v} - \tilde{\mathbf{c}}^{(j)}\|$$

- A count of the vectors $\mathbf{v}$ which do not classify correctly is formed, where there are $n - 1$ possible errors.

5 Dimension Reduction

As described in section 3, the number of elements m of the feature vectors can be very large. It has been found to be helpful to reduce the dimensionality of the feature vectors by projecting them into a lower dimensional space. The reduction of dimension is similar to principle component analysis (PCA) [17], but is used when the dimension of the vectors exceeds than the number of observations of vectors in the classes. This has been used in other textual analysis problems [27, 28] and facial recognition problems [29]. (In [16], dimension reduction is accomplished in a two-stage process, with PCA being following by a process similar to the one described here.) In this section we introduce the criterion used to perform the projection. In Appendix B, a few more details are provided (see [27, 29] for more detail).

While the feature vectors are in high-dimensional space, the salient concepts of dimension reduction can be illustrated in low dimensional space, such as figure 4. In that figure there are two 2-dimensional data sets, denoted with $\circ$ and $\times$, respectively. The problem is to determine for a given vector which class it belongs to. Also shown in the figure are two axes upon which the data are projected. (For "projection," think of the shadow cast by the data points by a light source high above the projection line.) The 1-dimensional data produced in Projection 1 have a cluster widths denoted by S_{W2} and S_{W2}. This is the within-cluster scatter, a measure of the variance (or width) of the densities. There is also a between-cluster scatter, a measure of how far the cluster centroids are from the overall centroid. In Projection 1, the

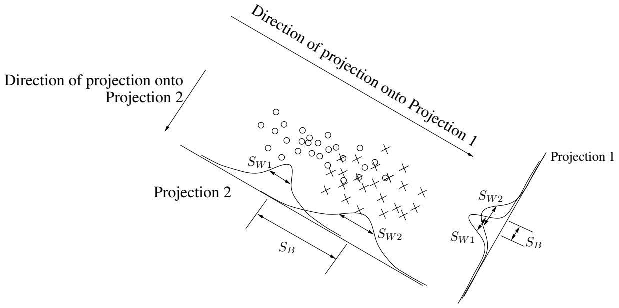

Direction of projection onto Projection 1
Direction of projection onto Projection 2
Projection 1
Projection 2
S_{W1}
S_{W2}
S_B
S_{W2}
S_{W1}
S_B

Figure 4: Illustration of within-cluster and between cluster scattering and projection.

between-cluster measure is rather small compared with the width of the cluster widths. By contrast, the 1-dimensional data produced in Projection 2 have a much larger between-cluster measure S_B. The within-cluster scatter S_{W1} and S_{W2} are also larger, but the between-cluster measure appears to have grown more than these within-cluster measures. Projection 2 produces data that would be more easily classified than Projection 1.

More generally, one can conceive of rotating the data at various angle with respect to the axis that data are projected upon. At some angles, the between cluster scatter will be larger compared to the within-cluster scatters.

In light of this discussion, the goal of the projection operation is to determine the best “angles of rotation” to project upon which maximize the between-cluster scatter while minimizing the within-cluster scatter. In general, there are k clusters of data to deal with (not just the two portrayed in figure 4). All this takes place in very high dimensional space, where we cannot visualize the data, so this is done via mathematical transformations. In higher dimensions, it is also not merely a matter of projecting onto a single direction. In m dimensions, the dimension of the projected data could be 1-dimension, 2-dimensions, etc., up to $m - 1$ dimensions. It is not known in advance what the best dimensionality to project onto is, so this is one of the parameters examined in the experiments described below.

With this discussion in place, we now describe the mathematics of how the projection is accomplished. For class i, with within-cluster scatter matrix —- that is, a measure of how the data in the in the class vary around the centroid of the class — is

$$S_{Wi} = \sum_{j \in N_i} (\mathbf{v}_j - \tilde{\mathbf{c}}^{(i)})(\mathbf{v}_j - \tilde{\mathbf{c}}^{(i)})^T$$

The total within-cluster scatter matrix is the sum of the within-class scatter matrices,

$$S_w = \sum_{i=1}^{k} \sum_{j \in N_i} (\mathbf{v}_j - \tilde{\mathbf{c}}^{(i)})(\mathbf{v}_j - \tilde{\mathbf{c}}^{(i)})^T.$$

For the data considered here, which has high dimensions and not a lot of training data, S_w is singular, that is, not invertible.

The between-cluster scatter matrix is the scatter of the individual class centroids compared with the overall centroid,

$$S_b = \sum_{i=1}^{k} \sum_{j \in N_i} (\tilde{\mathbf{c}}^{(i)} - \mathbf{c})(\tilde{\mathbf{c}}^{(i)} - \mathbf{c})^T = \sum_{i=1}^{k} n_i (\tilde{\mathbf{c}}^{(i)} - \mathbf{c})(\tilde{\mathbf{c}}^{(i)} - \mathbf{c})^T.$$

The idea behind dimension reduction is to find a $\ell \times m$ matrix G^T with $\ell < m$ then use G^T to transform the data according to $\mathbf{v}_i^\ell = G^T \mathbf{v}_i$. One may think of the matrix G^T as providing rotation of the vectors and selection of the dimensions which are retained after rotation.

The operation $\mathbf{v}_i^\ell = G^T \mathbf{v}_i$ may be thought of (naively) as feature selection: elements of $\mathbf{v}_i$ are retained in $\mathbf{v}_i^\ell$ which improve the clustering. Actually, beyond mere feature selection, the transformation G^T also produces linear combinations of feature vectors which improve the clustering of the data (and hence may improve the classifier capability).

Based on the discussion above, the matrix G^T is selected to minimize the within-cluster scatter S_w of the transformed data while also making the between-cluster scatter S_b as large as possible.

It may be surprising that projecting into lower dimensional spaces can improve the performance — it seems to be throwing away information that may be useful in the classification. What happens, however, is that the information

Table 3: Number of subtrees in union and intersection of sets of all subtrees

Author	# Docs	Total # Sentences	Total # Words	# Subtrees of depth 2	# Subtrees of depth 3	# Subtrees of depth 4
Hamilton	51	3126	206586	9660	19785	25834
Madison	14	1034	65492	4653	8182	9419
Jay	5	159	11732	1590	2150	2044
UncertainHorM	12	688	40607	3200	5363	6077
HandM	3	154	7928	1007	1373	1293
Totals	85	5161	332345	20110	36853	44667

Table 4: Summary statistics of *Federalist* data

Depth	# in union	# in intersection
2	14377	283
3	30121	195
4	39607	78

discarded is in directions that are noisy, or may be confusing to the classifier. As the results below indicate, projecting onto lower dimensions can significantly improve the classifier.

6 The Federalist Papers

With this background, we now turn attention to applying features derived from statistical parsing to two different sets of documents, the first being *The Federalist Papers*. *The Federalist Papers* consists of a series of 85 papers written around 1787 by Alexander Hamilton, James Madison, and John Jay in support of the U.S. Federal Constitution [30]. Of these papers, 51 are attributed unambiguously to Hamilton, 14 to Madison, 5 to Jay, and 3 to both Hamilton and Madison. The remaining twelve papers are of uncertain attribution, but are known to be by either Madison or Hamilton. In [7, 8], statistical techniques were used to determine that all twelve ambiguous papers were due to Madison.

The authors used in this study are $\{$Hamilton, Madison, Jay, UncertainHorM, HandM$\}$. A machine-readable copy of *The Federalist Papers* was obtained from the Gutenberg project [31]. Each of the 85 papers is considered a separate document, each of which is parsed into a separate file. Below we consider the performance of classifiers based on the feature vectors described above.

All Subtrees Subtree extraction is done on all trees for each of the *Federalist* papers. Table 3 shows the total number of sentences and words for each author considered. The table also shows the number of different subtrees at any level for the depths considered. The number of subtrees grows rapidly with the depth, leading to large feature vectors. Table 4 shows the number of subtrees in the union of the subtrees of all the authors and in the intersection of the subtrees of all the authors.

To form feature vectors using the subtree information, the top N subtrees (by count) for each author are selected (where we examined $N = 5, 10, 20,$ and 30), then the union across authors was formed of these top subtrees. In the tables below, the number of subtrees in the union of the top N is denoted as “length(union)”. The fact that this number exceeds N indicates that not all authors have the same top N subtrees. This length(union) is m, the dimension of the feature vector used for classification before projection into a lower dimensional space.

The subtrees in the union of the top N form the row labels in a term-by-document matrix, where the terms (rows) are the subtrees and there is one column for each paper. This term-by-document is filled with counts for each subtree, then the term-by-document matrix is normalized so that each column sum is equal to 1. Classification was done by nearest distance to the class centroid, as described in section 4.

Classification results are shown in Table 5. The results (most of them) are also shown in figure 5. The test conditions are the number N (for the top N), the depth of the tree, and the dimension of the reduced dimension space ℓ. There is an error count in the column “# Err”, which is the number of errors (out of 85) using the original high-dimensional feature vectors. There are also error counts # Err$_\ell$, for $\ell = 1, 2, 3, 4, 5$, which are the number of errors for data projected in the ℓ-dimensional space as described in section 5. The # Err column never achieves a value less than 16, illustrating that the raw subtree features do not provide good classification. However, the reduced-dimension data can achieve good classification. For example, with top $N = 5$, subtrees of depth 4 achieve an error count of 1 for $\ell = 2$ and $\ell = 3$ dimensional projection. In fact, the error count is actually better than the table shows. For all error counts of 1, the one error that occurs is a classification of the author HandM as the author Madison. Since it is understood that the HandM papers were actually written by Madison, this is a correct classification.

There is clearly a broad “sweet spot” for these features. Taking the top N at least 10, a subtree depth of 3 or 4, and projected dimension of $\ell = 2, 3$ or 4 provides the best performance. Interestingly, in all cases, moving to $\ell = 5$ actually slightly increases the number of errors to 2.

Table 5: Classification of *Federalist* papers based on "all subtree" feature vectors

top N	subtree depth	length(union)	# Err	# Err$_2$	# Err$_3$	# Err$_4$	# Err$_5$
5	2	16	25	29	10	13	15
	3	45	22	12	5	2	3
	4	103	25	1	1	2	2
10	2	36	21	27	5	4	6
	3	80	20	1	1	1	2
	4	194	22	1	1	1	2
20	2	82	16	1	1	1	2
	3	182	19	1	1	1	2
	4	402	20	1	1	1	2
30	2	146	15	1	1	1	2
	3	286	19	1	1	1	2
	4	700	23	1	1	1	2

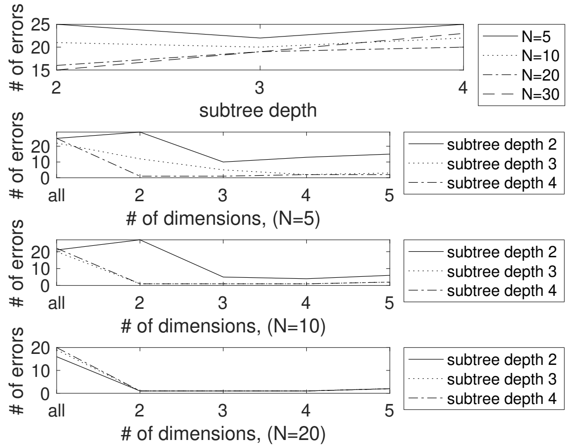


Figure 5: Error counts for the "all subtree" features

These results indicate that the subtree feature *does* provide information which can distinguish authorship, with appropriate weighting of features and selection of the dimension.

Rooted Subtrees We next considered using rooted subtrees as feature vectors. A few examples of these trees created from a Hamilton paper in the *The Federalist Papers* are shown in figure 6. There is quite a variety of possibilities (more than initially anticipated).

Some summary information about the number of different trees by author and level is shown in Table 6. Table 7 shows the number of rooted trees in the union of the trees across the authors, and the number of trees common to all authors (the intersection). There is so much variety in these trees that it is only at level 1 that there are any trees common to all authors, which is the tree S(NP)(VP), that is, a sentence with a noun phrase and a verb phrase.

The feature vectors were formed as follows. For each level of rooted subtree, the top N trees for each author were selected, and the union of these trees across the documents formed the terms in a term-by-document matrix. The number of trees obtained is shown in the column # Trees of Table 8.

Incidence counts were formed for each document of each user. Classification was done by nearest distance to the class centroid, as described in section 4. This gave rather high error counts for each of the different levels. Then the data was projected into ℓ-dimensional space, and the error counts # Err$_\ell$ are computed. The results are shown in Table 8. The columns # Err, #Err$_2$, …#Err$_5$ show the number of errors for full dimension then 2-, …, 5-dimensional projections. Figure 7 graphically portrays the tabulated results.

As in the "all subtree" feature case, these low-dimensional projections do very well. In fact, as before, the error that occurs in the case that there is one error is when the HandM author was classified as the Madison author, which is in fact a correct classification.

The rooted subtrees features have a very broad sweet spot where good classification occurs. For dimensions $\ell = 2, 3$ or 4, and at least 2 levels does very well. As for the all subtrees features, in this case also: for all error counts of 1, the one error that occurs is a classification of the author HandM as the author Madison. Since it is understood that the HandM papers were actually written by Madison, this is a correct classification, so all documents were correctly classified.

POS POS is a seemingly natural way to classify documents, but, contrary to expectations, it does not perform as well as the tree-based features. Feature vectors in this case are formed by taking the top N most common POS for each author, then forming the union of these POS. Feature vectors are formed by POS counts by author and document, normalized. Results are shown in Table 9. The raw error count for different values of N are all greater than or equal to 23. Moderate improvements can be obtained by projecting the feature vectors to lower dimensional space, with the errors for the ℓ-dimensional projection denoted by # Err$_\ell$, for $\ell = 2, 3, 4, 5$. Even in the best of circumstances, the error counts is equal to 4.

We conclude that while the POS provides a measure of distinguishability between authors, it does not provide the same degree of distinguishability as that provided by the structural information obtained from the parse trees.

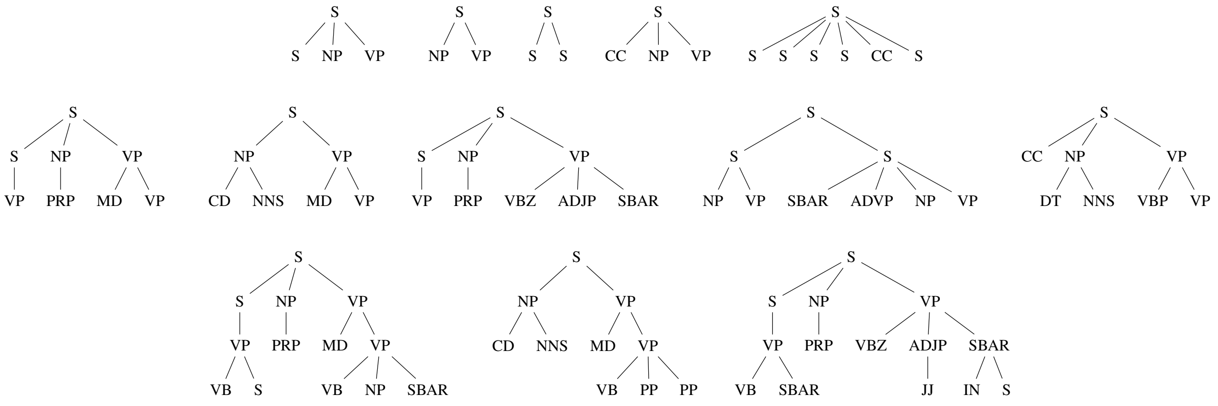


Figure 6: Example rooted trees of 1 level, 2 levels, 3 levels

Table 6: Summary statistics of number of “rooted subtrees” of different levels

Author	# Rooted sub trees of 1 level	# Rooted subtrees of 2 levels	# Rooted subtrees of 3 levels	# Rooted subtrees of 4 levels
Hamilton	285	1688	2732	2955
Madison	156	703	959	995
Jay	51	139	159	157
UncertainHorM	111	490	633	647
HandM	28	137	152	148

Table 7: Summary statistics of number of rooted trees unioned and intersected over authors

Level	# in union	# in intersection
1	398	1
2	2625	0
3	4425	0
4	4808	0

Table 8: Classification of *Federalist* papers based on “rooted subtree” feature vectors

top N	level	# Trees	# Err	# Err_2	# Err_3	# Err_4	# Err_5
5	1	37	36	15	13	11	2
	2	168	34	1	1	1	2
	3	371	35	1	1	1	2
	4	394	30	1	1	1	2
10	1	140	33	1	1	1	2
	2	413	35	1	1	1	2
	3	734	30	1	1	1	2
	4	774	30	1	1	1	2
20	1	323	33	1	1	1	2
	2	945	29	4	1	1	2
	3	1462	31	1	1	1	2
	4	1538	30	1	1	1	2
30	1	381	34	1	1	1	2
	2	1416	29	1	1	1	2
	3	2145	30	1	1	1	2
	4	2283	30	1	1	1	2

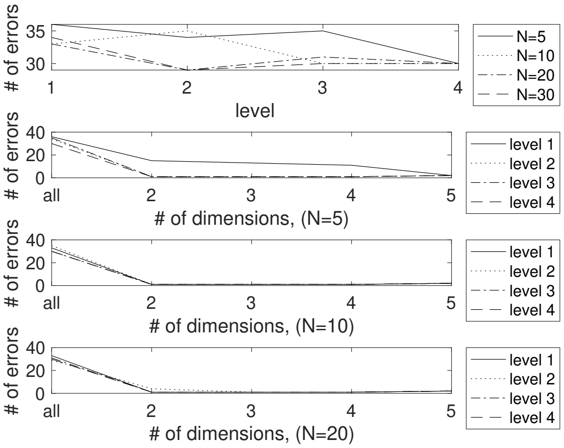


Figure 7: Error counts for the “rooted subtree” features

Table 9: Classification of *Federalist* papers based on POS vectors

top N	# POS in union	# Err	# Err_2	# Err_3	# Err_4	# Err_5
5	7	30	32	27	27	30
10	17	25	41	19	14	16
20	28	23	31	11	6	7
30	46	23	12	6	4	5

Table 10: Classification of *Federalist* papers based on “POS by level feature” vectors

top N	level	# Trees	# Err	# Err$_2$	# Err$_3$	# Err$_4$	# Err$_5$
5	1	9	42	11	35	39	40
	2	14	35	31	30	24	28
	3	12	46	38	31	32	35
	4	11	49	33	37	39	39
10	1	27	41	34	27	23	21
	2	20	36	35	28	21	24
	3	21	39	29	20	21	27
	4	19	40	23	19	20	22
20	1	33	41	36	18	18	18
	2	44	33	16	5	5	6
	3	36	38	23	13	9	9
	4	40	39	34	12	10	11

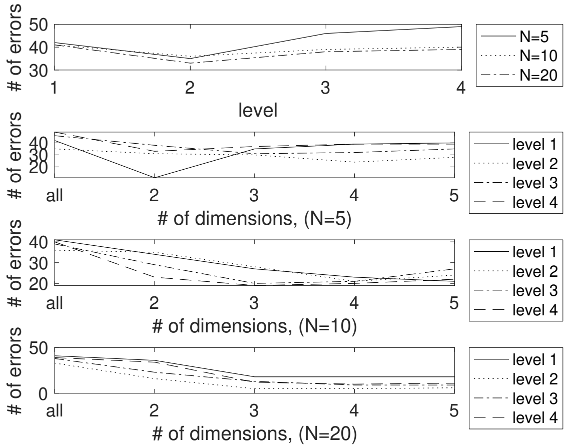


Figure 8: Error counts for the “POS by level” feature vectors

Table 11: Summary statistics of *Sanditon* data

Author	# Docs	Total # Sentences	Total # Words	# Subtrees of depth 2	# Subtrees of depth 3	# Subtrees of depth 4
Austen	2	1176	26342	4921	7378	7460
Other	4	2559	55453	8194	13795	15266
Totals	6	5161	332345	20110	36853	44667

Table 12: Number of subtrees in union and intersection of sets of subtrees for *Sanditon*.

Depth	# in union	# in intersection
2	11189	1926
3	19277	1896
4	21560	1166

POS by Level Table 10 shows the classification results for feature vectors obtained using the POS by level, using feature vectors formed in a manner similar to the other features. Figure 8 provides a graphical representation of this data. This feature vector provides some discrimination between authors, but fares substantially worse than the purely tree-based features.

6.1 Sanditon

Up until shortly before her death in 1817, Jane Austen was working on a novel posthumously titled *Sanditon* [32, p. 20]. Before her death she completed a draft of twelve chapters (about 24,000 words). The novel was posthumously “completed” by various writers with varying success. The version best known was published in 1975 [33], coathored by “Another Lady,” whose identity remains unknown. Whoever she was, she was a fan of Austen’s and attempted to mimic her style. Of this version, it was said, it “received, as compared with [its] predecessors, a warm reception from the English critics.” [34, p. 76]. Notwithstanding its literary appeal and the attempts at imitating the conscious habits of Austen, she failed in capturing the unconscious habits of detail: stylometric analysis has been able to distinguish between the different authors [2, Chapter 16].

We obtained a computer-readable document from the Electronic Text Center at the University of Virginia Library [35]. The document was evidently obtained optical character recognition (OCR) from scanned documents, so it was necessary to carefully spell-check the document, but contemporary spellings were retained. Two documents were produced, the first for Austen (with 1176 sentences) and the second for Other (with 2559 sentences). These were split into segments (for purposes of testing the classification capability). The Austen document had two segments of length 588 sentences. The Other document had four segments of lengths 640, 640, 640, 639. Subtrees of various depths were extracted from the segments, and these were classified the same way as the *Federalist* papers. Summary statistics about the documents are provided in Table 11.

Despite the attempt to duplicate Austen’s style, the segments for the different authors readily classify according to author, as shown below.

Table 13: Classification of *Sanditon* based on "all subtrees" feature vectors

top N	subtree depth	length(union)	# Err	# Err$_2$	# Err$_3$	# Err$_4$	# Err$_5$
5	2	9	0	0	2	4	5
	3	11	0	0	2	4	5
	4	15	2	0	2	5	5
10	2	12	0	0	2	4	5
	3	17	0	0	2	4	5
	4	29	2	0	3	4	5
20	2	30	0	0	2	4	5
	3	35	0	0	1	3	5
	4	58	1	0	2	4	5
30	2	46	0	0	2	4	5
	3	51	0	0	1	3	5
	4	88	2	0	3	4	5

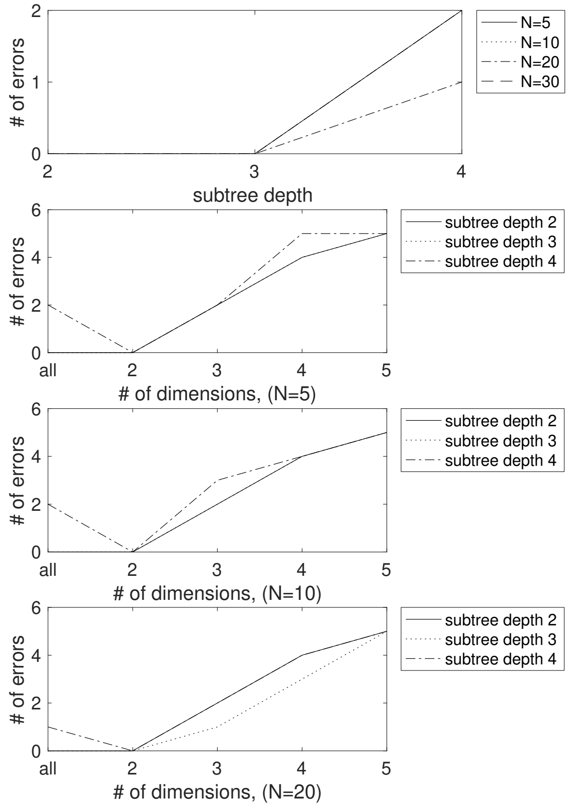


Figure 9: Classification of Sanditon based on "all subtrees" feature vectors

Table 15: Classification of *Sanditon* based on POS vectors

top N	# POS in union	# Err	# Err$_2$	# Err$_3$	# Err$_4$
5	5	0	3	4	5
10	11	0	0	1	3
20	21	0	0	2	4
30	36	0	0	2	4

All Subtrees For each six of the documents (two Austen, four Other), counts of all subtrees were extracted. As for the *Federalist* papers, the top N counts were extracted for $N = 5, 10, 20, 30$, and the union of these features was formed. This was done for subtrees of depth 2, 3, and 4. The number of trees in the union and intersection of these sets is shown in Table 12.

Classifier results for the all subtrees feature are shown in Table 13, and also portrayed in figure 9. As is shown, even with the full dimensionality (without projecting into a lower dimensional space), separation can be done completely accurately. On the other hand, the projected feature vectors do not generally perform as well as the full-dimensional data. This differs from how the lower dimensional projections worked for the *Federalist* documents.

Rooted Subtrees We next considered using rooted subtrees as feature vectors. Feature vectors were formed in the same way as for the *The Federalist Papers*. Results are shown in Table 14 and portrayed in figure 10. While not as effective at distinguishing as the subtrees features, this feature still shows the ability to distinguish between authors.

POS POS feature vectors were extracted in the same manner as for the *The Federalist Papers*. Data up to Err$_4$ were produced. The POS data was able to effectively distinguish between authors, more effectively than for the *The Federalist Papers*. Reducing the dimensionality did not improve the classifier (and beyond $\ell = 2$ made it worse).

Table 14: Classification of *Sanditon* based on “rooted subtree” feature vectors

top N	level	# Trees	# Err	# Err$_2$	# Err$_3$	# Err$_4$	# Err$_5$
5	1	8	0	5	1	3	5
	2	11	2	0	1	3	6
	3	27	4	0	3	4	6
	4	27	2	0	2	3	6
10	1	15	0	0	1	3	5
	2	28	4	0	3	3	5
	3	56	3	0	2	4	5
	4	56	2	0	2	4	5
20	1	31	0	0	2	3	5
	2	60	2	0	2	4	5
	3	115	2	0	1	5	5
	4	116	2	0	2	3	5
30	1	59	0	0	2	3	5
	2	106	2	0	2	3	5
	3	174	2	0	2	5	5
	4	176	2	0	2	3	5

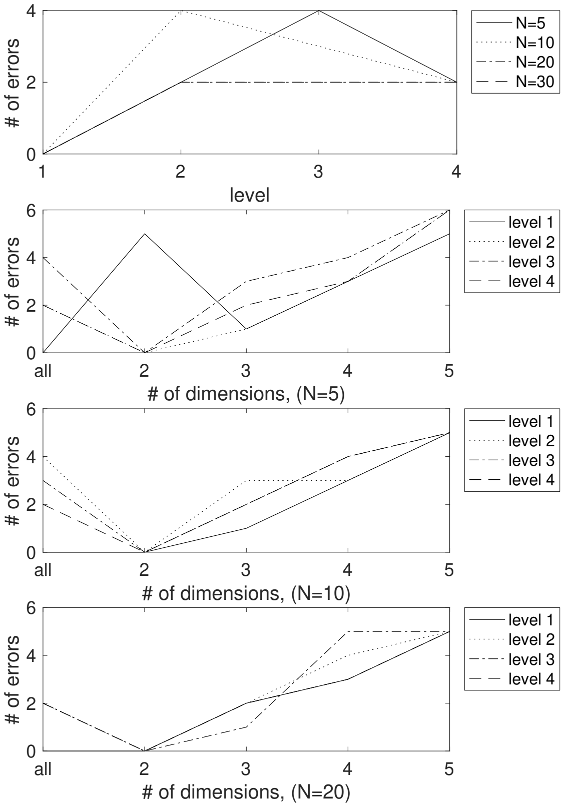


Figure 10: Classification of Sanditon based on “rooted subtree” feature vectors

Table 16: Classification of *Sanditon* based on "POS by level" feature vectors

top N	level	# Trees	# Err	# Err$_2$	# Err$_3$	# Err$_4$
5	1	5	0	2	4	4
	2	5	0	1	2	5
	3	7	1	0	1	3
	4	7	0	0	2	3
10	1	11	0	0	2	4
	2	11	0	0	2	4
	3	13	1	0	2	3
	4	10	0	0	2	3
20	1	26	0	0	3	4
	2	21	0	0	2	4
	3	21	0	0	1	3
	4	21	0	0	2	4
30	1	42	0	0	3	4
	2	36	0	0	2	4
	3	35	0	0	1	3
	4	33	0	0	2	4

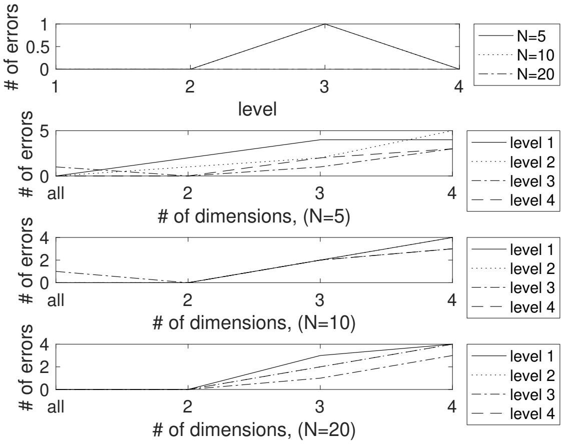


Figure 11: Classification of Sanditon based on "POS by level" feature vectors

POS by Level POS by Level feature vectors were extracted in the same manner as for the *The Federalist Papers*. Data up to Err$_4$ were produced. The classification results are shown in Table 16 and portrayed in figure 11.

The POS by Level data was able to effectively distinguish between authors, more effectively than for the *The Federalist Papers*. Reducing the dimensionality did not improve the classifier (and beyond $\ell = 2$ made it worse).

7 Conclusions, Discussion, and Future Work

As this paper has demonstrated, information drawn from statistical parsing of a text can be used to distinguish between between authors. Different sets of features have been considered (all subtrees, rooted subtrees, POS, and POS by Level), with different degrees of performance among them. Other than the POS these features have not been previously considered (to the knowledge of the authors), including in the large set of features examined in [16]. This suggests that these tree-based features, especially the features based on all subtrees, may be beneficially included among other features.

It appears that the *Sanditon* texts are easier to classify than the *The Federalist Papers*. Even without the generally performance-enhancing step of dimension reduction, *Sanditon* classifies well, even using the POS feature vectors which are not as strong when applied to the *The Federalist Papers*. This is amusing, since the completer of *Sanditon* attempted to write in an imitative style, suggesting that these structural features are not easily faked.

The methods examined here does not preclude the excellent work on author identification that has previously been done, which is usually done using more obvious features in the document (such as word counts, with words selected from some appropriate set). This makes previous methods easier to compute. But at the same time, it may make it easy to spoof the author identification. The grammatical parsing provides more subtle features which will be more difficult to spoof.

Another tradeoff is the amount of data needed to extract a statistically meaningful feature vector. The number of trees — the number of feature elements — quickly becomes very large. In order to be statistically significant a feature element should have multiple counts. (Recall that for the chi-squared test in classical statistics a rule of thumb is that at least five counts are needed.) This need to count a lot of features indicates that the method is best applied to large documents.

In light of these considerations, the method described here may be considered suplemental to more traditional author identification methods.

The method is naturally agnostic to the particular content of a document — it does not require selecting some subset of words to use for comparisons — and so should be applicable to documents across different styles and genres. The analysis could be applied to any document amenable to statistical parsing. (It does seem that documents with a lot of specialized notation, such as mathematical or chemical notation would require adaptation to the parser.)

This paper introduces many possibilities for future work. Of course there is the question of how this will apply to other work in author identification. It is curious that the dimension reduction behaves so differently for the *Federalist*

Table 17: Example rules for a PCFG (see Figure 14.1 of [26]). S=start symbol (or sentence); NP=noun phrase; VP = verb phrase; PP=prepositional phrase.)

Grammar	**Probability**	**Lexicon**
S → NP VP	[0.80]	*Det* → *that* [0.10] | *a* [.30] | *the* [.60]
S → *Aux* NP NP	[0.15]	*Noun* → *book* [.10] | *flight* [.30] |
S → VP	[0.05]	*meal* [.15] | *money* [.05]
NP → *pronoun*	[0.35]	*flights* [.40] | *dinner* [.10]
NP → *Proper-Noun*	[0.30]	*Verb* → *book* [0.30] | *include* [0.30] |
NP → *Det Nominal*	[0.20]	*prefer* [0.40]
NP → *Nominal*	[0.15]	*Prounoun* → *I* [0.40] | *you* [0.40] |
Nominal → *Noun*	[0.75]	*me* [0.15] | *you* [0.40]
Nominal → *Nominal Noun*	[0.20]	*Proper-noun* → *Houston* [0.60] |
Nominal → *Nominal* PP	[0.05]	*NWA* [0.40]
VP → *Verb*	[0.35]	*Aux* → *does* [0.60] | *can* [0.40]
VP → *Verb* NP	[0.20]	*Preposition* → *from* [0.30] | *to* [0.30] |
VP → *Verb* NP PP	[0.10]	*on* [0.20] | *near* [0.15] |
VP → *Verb* PP	[0.15]	*through* [0.15]
VP → *Verb* NP PP	[0.05]	
VP → VP PP	[0.15]	
PP → *Preposition* NP	[1.0]	

and *Sanditon* — *Federalist* best in smaller dimensions, but *Sanditon* works better in larger dimensions. Given recent furor over machine learning, it would be interesting to see if the features extracted by the grammatical parser correspond in any way to features that would be extracted by a ML tool. (My suspicion is that training on current ML tools does not extract grammatical information applicable to the author identification problem.)

A A Brief Introduction to Statistical Parsing

At the suggestion of an anonymous reviewer, this appendix was written to provide a brief a discussion of the statistical parsing, drawing very closely from [26, Chapter 14, Statistical Parsing]. More detailed discussions are provided in [21, 24]. The probabilistic grammar employed is a probabilistic context-free grammar (PCFG). In this grammar are rules for transforming nonterminal symbols to a string of symbols which could be nonterminal symbols or terminal symbols. In a PCFG each rule is accompanied by a probability. As an example, Table 17 shows a grammar for a toy language (used for airline reservations). Each rule in the table of the form $A \to B \quad [p]$ means that p is the probability that the non-terminal A will be expanded to the sequence B. This can be alternatively represented as $P(A \to B)$ or as $P(A \to B|A)$ or as $P(\text{LHS}|\text{RHS})$, where LHS and RHS mean "left hand side" and "right hand side," respectively.

In Table 17, S denotes an start symbol (for a sentence). The grammar's first rule says that a sentence may consist of a NP (noun phrase) followed by a VP (verb phrase), and that such a rule occurs with probability 0.8. The second rule says that a sentence may consist of an auxiliary (such as *does* or *can*) followed by NP then a VP, with probability 0.15. The next rule says that a sentence can be a verb phrase, VP. The tokens obtained by application of a rule can be recursively expanded using their own rules, shown in the table. Thus, a NP may consist of a pronoun, or a proper-noun, etc. The probabilities are estimated from a large corpus of data parsed by a linguist.

There is also a lexicon (or dictionary) of terms, each term of which has a probability within its lexicon type. The right side of Table 17 illustrates a lexicon for this application. For example, a determiner *Det* can be *that* (with probability 0.1) or *a* (with probability 0.3) or *the* (with probability 0.6). A noun (in this application context) can be book, flight, meal, money, flights, or dinner, with their respective probabilities. Probabilities are (again) estimated from a corpus.

A PCFG can be used to estimate the probability of a parse tree, which can be used to disambiguate multiple parsings, or it can be used to determine the probability of a sentence in a language modeling setting.

"The probability of a particular parse tree T is defined as the product of the probabilities of all the rules used to expand each of the n non-terminal nodes in the parse tree T, where each rule i can be expressed as $\text{LHS}_i \to \text{RHS}_i$:

$$P(T,S) = \prod_{i=1}^{n} P(\texttt{RHS}_i|\texttt{LHS}_i)$$

The resulting probability $P(T,S)$ is ... the joint probability of the parse and the sentence and the probability of the parse $P(T)$." [26, p. 462] . In computing the probability on a tree, there is a factor for every rule, which corresponds to every edge on the tree.

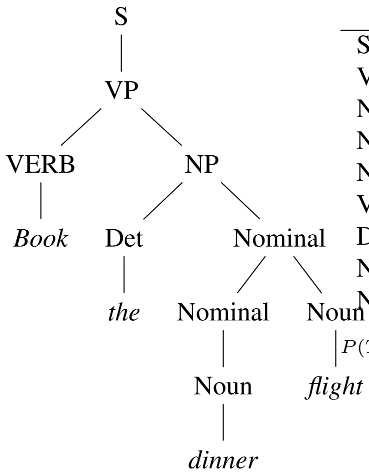
As an example, consider two different ways of parsing the sentence: "Book the dinner flight." This can be parsed (understood) either as

① Book a flight that serves dinner

or as

② Book a flight for [on behalf of] the dinner.

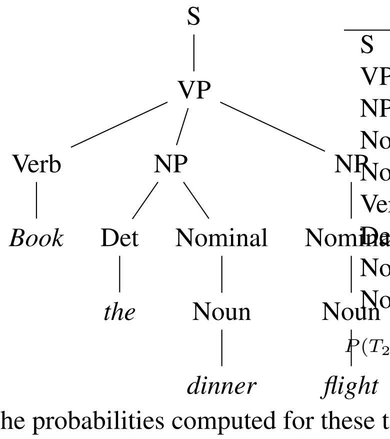
The parse tree, rules, and corresponding probability for parsing ① is shown here:



Rules			Prob
S	→	VP	.05
VP	→	Verb NP	.20
NP	→	Det Nominal	.20
Nominal	→	Nominal Noun	.20
Nominal	→	Noun	.75
Verb	→	*book*	.30
Det	→	*the*	.60
Noun	→	*dinner*	.10
Noun	→	*flights*	.40

$P(T_1) = (.05)(.2)(.2)(.2)(.75)(.3)(.6)(.1)(.4) = 2.2 \times 10^{-6}$

The parse tree, rules, and corresponding probability for parsing ② is shown here:



Rules			Prob
S	→	VP	.05
VP	→	Verb NP NP	.10
NP	→	Det Nominal	.20
Nominal	→	Noun	.75
Nominal	→	Noun	.75
Verb	→	*book*	.30
Det	→	*the*	.60
Noun	→	*dinner*	.10
Noun	→	*flights*	.40

$P(T_2) = (.05)(.1)(.2)(.15)(.75)(.75)(.3)(.6)(.1)(.4) = 6.1 \times 10^{-7}$

The probabilities computed for these two parse structures are

$$P(T_1) = 2.2 \times 10^{-6} \qquad P(T_2) = 6.1 \times 10^{-7}.$$

The parsing ① has much higher probability than parsing ② (which accords with a common understanding of the sense of the sentence).

The parser works through the text being parsed, probabilistically associating the word with its grammatical element, in the context of the tree that is being built. When competing trees are constructed, the tree with highest probability is accepted.

B Dimension Reduction: Some Mathematical Details

This material is drawn from [27]. The trace of S_w provides a measure of the clustering of the feature for each class around their respective centroids,

$$\operatorname{tr}(S_w) = \sum_{i=1}^{k} \sum_{j \in N_i} (\mathbf{v}_j - \mathbf{c}^{(i)})^T (\mathbf{v}_j - \mathbf{c}^{(i)}) = \sum_{i=1}^{k} \sum_{j \in N_i} \|\mathbf{v}_j - \mathbf{c}^{(i)}\|^2.$$

Note that S_w, being the sum of the outer product of n terms, generically has rank $\min(n, m)$. In the work here, the dimension of the feature vectors m is very large, so that $\operatorname{rank}(S_w) = n$; S_w is singular.

Similarly, $\operatorname{tr}(S_b)$ measures the total distance between cluster centroids and the overall centroid,

$$\operatorname{tr}(S_b) = \sum_{i=1}^{k} \sum_{j \in N_i} (\mathbf{c}^{(i)} - \mathbf{c})^T (\mathbf{c}^{(i)} - \mathbf{c}) = \sum_{i=1}^{k} \sum_{j \in N_i} \|\mathbf{c}^{(i)} - \mathbf{c})\|^2.$$

A measure of cluster quality which measures the degree to which $\operatorname{tr}(S_w)$ is small and $\operatorname{tr}(S_b)$ is large is

$$J_1 = \operatorname{tr}(S_w^{-1} S_b)$$

As noted above, S_w is singular, so this a conceptual expression (not actually computed). In [16], the problem of the singularity of S_w is dealt with by working with a regularized scatter matrix $S_w + \lambda I$, for some regularization parameter λ, which is found there by searching over a range of λs which provide best performance. The method described here using the SVD avoids the need to perform this search (and the possibility that some performance may have been sacrificed by not finding an optimum value of λ).

When the vectors are transformed by the transformation G^T, the scatter matrices are

$$S_{w,G} = \sum_{i=1}^{k} \sum_{j \in N_i} (G^T \mathbf{v}_j - G^T \mathbf{c}^{(i)})(G^T \mathbf{v}_j - G^T \mathbf{c}^{(i)})^T = G^T S_w G,$$

and (similarly) $S_{b,G} = G^T S_b G$ and $S_{m,G} = G^T S_m G$. The goal now is to choose G^T to make $\operatorname{tr}(S_{w,G})$ small while making $\operatorname{tr}(S_{b,G})$ large. More precisely, the matrix G is sought that maximizes

$$J_1(G) = \operatorname{tr}((G^T S_w G)^{-1} (G^T S_b G)).$$

In this case, the matrix $G^T S_w G$ may not be singular.

To express the algorithm, the following matrices are defined. The scatter matrices S_w, S_b and S_m can be expressed in terms of the matrices

$$H_w = \begin{bmatrix} V_1 - \mathbf{c}^{(1)} \mathbf{e}_{n_1} & V_2 - \mathbf{c}^{(2)} \mathbf{e}_{n_2} & \cdots & V_k - \mathbf{c}^{(k)} \mathbf{e}_{n_k} \end{bmatrix} \in \mathbb{R}^{m \times n},$$

$$H_b = \begin{bmatrix} \sqrt{n_1}(\mathbf{c}^{(1)} - \mathbf{c}) & \sqrt{n_2}(\mathbf{c}^{(2)} - \mathbf{c}) & \cdots & \sqrt{n_k}(\mathbf{c}^{(k)} - \mathbf{c}) \end{bmatrix} \in \mathbb{R}^{m \times k}$$

and

$$H_m = \begin{bmatrix} \mathbf{v}_1 - \mathbf{c} & \mathbf{v}_2 - \mathbf{c} & \cdots & \mathbf{v}_n - \mathbf{c} \end{bmatrix} \in \mathbb{R}^{m \times n}$$

as

$$S_w = H_w H_w^T \qquad S_b = H_b H_b^T \qquad S_m = H_m H_m^T.$$

That is, H_m, H_b and H_m form factors of the respective scatter matrices.

The algorithm for computing G is shown below. (adapted from Algorithm 1 of [27]).

Algorithm 1 Finding a structure-preserving, dimension-reducing matrix G:

Given matrices $H_b \in \mathbb{R}^{m \times k}$ and $H_w \in \mathbb{R}^{m \times n}$ (the factors of S_b and S_w), determines the matrix $G \in \mathbb{R}^{m \times \ell}$ which preserves the cluster structure in the ℓ dimensional space.

Input: H_b, H_w, ℓ. **Output:** G

1. Form

   $$K = \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix} \in \mathbb{R}^{(k+n)\times m}$$

   and compute its SVD

   $$K = P \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} Q^T.$$

   Determine the rank of K:

   $$t = \text{rank}(R)$$

2. Compute the SVD of a submatrix of P:

   $$P(1:m, 1:t) = U\Sigma W^T$$

3. Form an empty matrix $G \in \mathbb{R}^{m\times \ell}$.

4. (Compute G as the first ℓ columns of $Q \begin{bmatrix} R^{-1}W & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix}$. This can be done as follows:)

```
(Overwrite the first $t$ columns of $Q$ as $QR^{-1}$:)
for $j = 1 : t$
  $Q(:, j) = Q(:, j)/R(j, j)$
end
if($\ell \leq t$)
  $G(:, 1 : \ell) = Q(:, 1 : t)W(:, 1 : \ell)$
else
    Print: "Number of columns of $G$ requested exceeds number of nontrivial
      singular values pairs of $H_b^T$ and $H_w^T$"
    $G(:, 1 : t) = Q(:, 1 : t)W(:, 1 : t)$
  if($\ell > n$)
    Print: "And it exceeds the number of columns of $G$"
  else (Set the remaining columns of $G$ equal to $Q_2$)
    $G(:, t + 1 : \ell) = Q(:, t + 1, \ell)$
  end
end
```

References

[1] R. Lord, “de Morgan and the Statistical Study of Literary Style,” *Biometrica*, vol. 3, p. 282, 1958.

[2] A. Morton, *Literary Detection*. New York: Charles Scribner’s Sons, 1978.

[3] T. Mendenhall, “A Mechanical Solution of a Literary Problem,” *Popular Science Monthly*, 1901.

[4] C. D. Chretien, “A Statistical Method for Determining Authorship: The Junius Letters,” *Languages*, vol. 40, pp. 95–90, 1964.

[5] D. Wishart and S. V. Leach, “A Multivariate Analysis of Platonic Prose Rhythm,” *Computer Studies in the Humanities and Verbal Behavior*, vol. 3, no. 2, pp. 109–125, 1972.

[6] C. S. Brinegar, “Mark Twain and the Quintis Curtis Snodgrass Letters: A Statistical Test of Authorship,” *Journal of the Americal Statistical Association*, vol. 53, p. 85, 1963.

[7] F. Mosteller and D. Wallace, *Inference and Disputed Authorship: The Federalist*. Reading, MA: Addison Wesley, 1964.

[8] P. Hanus and J. Hagenauer, “Information Theory Helps Historians,” *IEEE Information Theory Society Newsletter*, vol. 55, p. 8, Sept. 2005.

[9] D. Holmes, “The analysis of literary style — a review,” *J. Royal Statistical Society, Series A*, vol. 148, no. 4, pp. 328–341, 1985.

[10] J. L. Hilton, “On Verifying Wordprint Studies: Book of Mormon Authorship,” *Brigham Young University Studies*, 1990.

[11] D. Holmes, “A Stylometric Analysis of Mormon Scriptures and Related Texts,” *Journal of the Royal Statistical Society, A*, vol. 155, pp. 91–120, 1992.

[12] K. Luyckx and W. Daelemans, “Shallow text analysis and machine learning for authorship attribution,” in *Proceedings of the Fifteenth Meeting of Computational Linguistics in the Netherlands*, pp. 149–160, 2005.

[13] J. Grieve, “Quantitative authorship attribution: an evaluation of techniques,” *Liter. Linguist. Comput.*, vol. 22, no. 3, pp. 251–270, 2007.

[14] F. Iqbal, H. Binsalleeh, B. Fung, and M. Debbabi, “A unified data mining solution for authorship analysis in anonymous textual communication,” *Inform. Sci*, vol. 231, pp. 98–112, 2007.

[15] E. Stamatos, “A survey of modern authorship attribution methods,” *J. Am. Soc. Inform. Sci. Technol.*, vol. 60, no. 3, pp. 538–556, 2009.

[16] C. Zhang, X. Wu, Z. Niu, and W. Ding, “Authorship identification from unstructured texts,” *Knowledge-Based Systems*, vol. 66, pp. 99–111, 2014.

[17] I. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[18] S. Corbara, B. Chulvi, and A. Moreo, *Experimental IR meets multilingality, multimodality, and interaction*, vol. 13390 of *Lecture Notes in Computer Science*, ch. Rythmic and Psycholinguistic Features for Authorship Tasks in the Spanish Parliament: Evaluation and Analysis. Springer, 2022.

[19] S. Corbara, C. C. Ferriols, P. Rosso, and A. Moreo, *Natural Language Processing and Information Systems*, vol. 13286 of *Lecture Notes in Computer Science*, ch. Investigating Topic-Agnostic Features for Authorship Tasks in Spanish Political Speeches. Springer, 2022.

[20] M. Sanchez-Perez, I. Markov, H. Gómez-Adorno, and G. Sidorov, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, vol. 10456 of *Lecture Notes in Computer Science*, ch. Comparison of Character n-grams and Lexical Features on Author, Gender, and Language Variety Identification on the Same Spanish News Corpus. Springer, 2017.

[21] D. Klein and C. D. Manning, “Accurate unlexicalized parsing,” in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, vol. 1, pp. 423–430, 2003. https://doi.org/10.3115/1075096.1075150.

[22] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, “Building a large annotated corpus of english: The penn treebank,” *Computational Linguisistics*, vol. 19, no. 2, pp. 313–330, 1993.

[23] A. Taylor, M. Marcus, and B. Santorini, “The Penn Treebank: An Overview.” `https://www.researchgate.net/publication/2873803_The_Penn_Treebank_An_overview`, 2003.

[24] D. Klein and C. D. Mannning, “Accurate unlexicalized parsing,” in *Proceedings of the 41st Meeting of the Association for Computational Linguisistics*, pp. 423–430, 2003.

[25] T. S. N. L. Group, “Software: Stanford parser.” `https://nlp.stanford.edu/software/lex-parser.html`, 2020.

[26] D. Juraksky and J. H. Martin, *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, NJ: Prentice-Hall, 2009.

[27] P. Howland, M. Jeon, and H. Park, “Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition,” *SIAM J. Matrix Anal. Appl.*, vol. 25, no. 1, pp. 165–179, 2003.

[28] T. K. Moon, P. Howland, and J. H. Gunther, “Document author classification using generalized discriminant analysis,” in *SIAM Conference on Text Mining*, (Baltimore, MD), May 23–25 2006.

[29] P. Howland, J. Wang, and H. Park, “Solving the small sample size problem in face recognition using generalized discriminant analysis,” *Pattern Recognition*, vol. 39, pp. 277–287, 2006.

[30] A. Hamilton, J. Madison, and J. Jay, “The federalist,” in *American State Papers* (R. M. Hutchins, ed.), vol. 43 of *Great Books of the Western World*, pp. 29–266, Encyclopedia Britannica, Chicago ed., 1952.

[31] A. Hamilton, J. Madison, and J. Jay, “*The Federalist* (machine readable).” `http://www.gutenberg.org/etext/18`.

[32] P. Poplawski, *A Jane Austen Encyclopedia*. London: Aldwych Press, 1998.

[33] J. Austen and A. Lady, *Sanditon*. London: Peter Davies, 1975.

[34] D. Hopkinson, “Completions,” in *The Jane Austen Companion* (J. D. Grey, ed.), Macmillan, 1986.

[35] “*Sanditon* (machine readable).” `http://etext.lib.virginia.edu/toc/modeng/public/AusSndt.html`.