Bigrams of Syntactic Labels for Authorship Discrimination of Short Texts

Graeme Hirst and Ol’ga Feiguina[1]
Department of Computer Science, University of Toronto, Toronto, Ontario, Canada

Correspondence:
Graeme Hirst, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, M5S 3G4.
E-mail: gh@cs.toronto.edu

Abstract

We present a method for authorship discrimination that is based on the frequency of bigrams of syntactic labels that arise from partial parsing of the text. We show that this method, alone or combined with other classification features, achieves a high accuracy on discrimination of the work of Anne and Charlotte Brontë, which is very difficult to do by traditional methods. Moreover, high accuracies are achieved even on fragments of text little more than 200 words long.

1 Introduction

Methods for identifying or discriminating the authorship of a text typically rely on both the questioned text and the body of attested work of the putative author being relatively large. In the canonical case, a novel or play of uncertain or disputed authorship is compared against attested corpora that are several times as large or more. The smaller the text or the comparison corpus, the less certain the results are. Thus, methods that could perform authorship tests with greater reliability on smaller texts would be welcome both in literary studies and in forensic analysis. In this article, we show the potential for the use of syntactic information in achieving this goal, and, in particular, we show that the use of bigrams of labels from a partial parser provides a good compromise between part-of-speech tagging and a complete parse.

1.1 Why short texts?

We are, of course, not the first to think about attribution of authorship of small texts. For example, Burrows (2002) tried his well-known Delta method (which looks for differences between texts in the distribution of frequent words) on poems of less than 500 words. He achieved an accuracy of 27% in identifying the correct author from a set of twenty-five candidates, and concluded that while the procedure is ‘effective enough’ on texts greater than 1,500 words, for shorter texts it is only ‘a basis for selecting a likely group of candidates’ (Burrows, 2002, p. 276). Zheng *et al.* (2006) used a variety of text and vocabulary-richness features to identify authors of short ‘for sale’ messages in a newsgroup. They achieved an accuracy of 97.6% when they included such features as ‘telephone number in signature’ and the presence of domain-specific words such as *obo* and *thx*, and 90% with just function words, vocabulary-richness, and superficial text features.[2]

Glover and Hirst (1996) and Graham *et al.* (2005) looked at a different version of the problem: finding authorship boundaries in collaboratively written text (with the ultimate aim of helping the authors to harmonize their style better). They therefore needed to discriminate the authorship of texts as short as several paragraphs or just

doi:10.1093/llc/fqm023 Advance Access Published on 1 October 2007

a single paragraph. Glover and Hirst used conventional authorship discrimination methods (Holmes, 1994) to see how well they could work on text of just a few paragraphs, but found the results to be mediocre. Graham *et al.* used a variety of neural-net methods along with conventional features on text that was tagged with the part of speech of each word. They found that time-delay neural nets gave results well above baseline, albeit not good enough for practical use, when used with features such as part-of-speech, punctuation, and function-word frequencies, but not with vocabulary-related features. (Graham *et al.* also tried simple letter bigrams, but these failed for texts of less than about 500 words.)

Ultimately, the problem with small texts is that they are small. They contain less information, and hence fewer clues to authorship. It therefore becomes more important to use as much information as possible from what is given. An obvious strategy to try is to make better use of the syntactic properties of the text.

1.2 Why syntax?

Finding patterns in the way that an author uses vocabulary, both content words and function words, has been proven to be useful in authorship attribution (Holmes, 1994). Less work has explicitly considered the way that authors use syntactic structure. Implicitly, however, the importance of syntactic information is seen in the success, from Mosteller and Wallace (1964) on, of using counts of the most frequent function words, as these can be viewed as some basic indicators of an author's syntactic usage. And, as noted earlier (Section 1.1), Graham *et al.* (2005) found the use of frequencies of part-of-speech tags to be helpful. But part-of-speech tags, while syntactic in nature, do not reflect syntactic structure *per se*. Among the research that has explicitly looked at features of syntactic structure in text, perhaps the most prominent is that of Baayen *et al.* (1996), who worked with sections of two English crime novels (around 20,000 words each), and of Stamatatos *et al.* (2000; 2001), who worked with 300 Greek newspaper articles (an average of about 1,100 words each). Syntactic patterns proved to be useful in both cases.

Baayen *et al.* represented each sentence as the sequence of re-write rules implied by its syntactic structure, viewed every such rule (there were about 4,200 kinds) as a pseudo-word, and then applied standard vocabulary-based authorship-discrimination measures to the 'vocabulary' of this representation of the text. They used five measures of vocabulary richness, which we denote collectively as *KDRSW*: Yule's measure *K* and Simpson's measure *D* of the lexical repetition rate; Honoré's measure *R* and Sichel's measure *S* of hapax legomena and dislegomena, respectively; and Brunet's measure *W* based on the type/token ratio. (See Baayen *et al.*'s paper for definitions and discussions of each.) Baayen *et al.* also directly analyzed the use of the fifty most frequently used rules and of low-frequency rules. The study found that this method performed better and was more uniform across texts than the same measures applied directly to the vocabulary of the texts. Baayen *et al.* suggest that the success of their method compared to purely lexical measures is due to syntactic processes being less manipulable by authors and hence varying less across their texts. They were skeptical, however, of the use of fully automatic parsers—their work was based on the Nijmegen corpus, which was semi-automatically annotated with syntactic structure (Oostdijk, 1991).

Stamatatos *et al.*, on the other hand, used fully automatic (albeit imperfect) syntactic *chunking*—that is, demarcating and labeling the non-overlapping (non-recursive) phrases of the sentence without determining the complete syntactic tree structure (an example for English is shown in Fig. 1). This allowed them to compute 22 stylistic features such as the average number of words per noun phrase and the ratio of noun phrases to other chunks; they also used some highly artifactual features, such as the fraction of words that remained unanalyzed after each pass of their five-pass chunker. (In fact, all of their stylistic features were ratios and averages.) Stamatatos *et al.* found that such features were more stable with respect to reductions in text size and training-set size than either lexical features alone or a combination of both kinds of features.

```
NP[Mr. Heathcliff and I] VP[are such] NP[a suitable pair]
VP[to divide] NP[the desolation] PP[between us].
```

Fig. 1 Example of text chunked at the non-recursive phrase level, similar to that used by Stamatatos *et al*. NP = noun phrase, VP = verb phrase, PP = prepositional phrase

These two approaches have complementary strengths and weaknesses. The method of Baayen *et al*. captures a large amount of syntactic information—essentially, the complete derivation of each sentence. But the price for this is that the space of rewrite-rules is very large, and so must be reduced to a manageable number of features in order to prevent data sparsity—in this case, by the same methods as are used to reduce the space of words in an author's vocabulary to a set of features. In particular, all notion of order is lost; a text is treated as a bag of rewrite-rules, just as vocabulary-based measures treat a text as a bag of words. The method has been tried only on large texts, and presumably, by its nature, is applicable only to such texts. Moreover, as a practical matter, while automatic parsers have improved notably in the time since Baayen *et al*.'s work was published, they still achieve lower accuracy than part-of-speech taggers and chunkers. In contrast, the method of Stamatatos *et al.* can be applied to relatively short texts and takes advantage of an automatic chunker, but it is highly dependent on artifacts of the particular Greek chunker that is used; and in reducing much of the other information that it provides down to numerical quantities mostly concerning phrase length, it discards all the information about order and about the structure and derivation of the text itself.

In this article, we will present a method that aims to capture much of the strength of both these approaches while avoiding some of the weaknesses. It is based on *partial parsing*, and it condenses rather than discards information on syntactic structure and derivation by using *bigrams of syntactic labels*, as we will explain in Section 2.2. We treat these bigrams as pseudo-words, much as Baayen *et al.* did with their rewrite-rules, and consider their relative frequencies.

1.3 Support vector machines in authorship attribution

In the work to be described below, we use support vector machines (SVMs) as our classification method. SVMs have recently been shown to give very good results in various kinds of text classification tasks (Joachims, 2002). They use a supervised learning algorithm. Given a set of data samples, each labeled +1 or −1, an SVM finds a linear separation, a hyperplane, between the differently labeled datasets that maximizes the margin between the two classes. Then it can classify unseen samples by determining which side of the separating line they are on. But as datasets aren't always linearly separable, SVMs employ a kernel function to map all samples to higher-dimensional spaces, which are then checked for linear separability. A kernel function simplifies the non-linear mapping to a space of different dimensionality; it defines the dot product of the transformations of two vectors. This is more efficient than defining the transformation function, computing it for every vector, and taking the dot product of the result. The most commonly used kernels are the dot kernel, the polynomial kernels, and the radial-basis-function kernel. If it is not possible to find a linear separation for a set of data with a certain kernel, the SVM can be allowed a certain misclassification rate.

Perhaps the first researchers to apply SVMs in stylometry were Fung (2003) and Diederich *et al.* (2003). Fung used an SVM feature-selection method to correctly classify the disputed *Federalist Papers* with just the frequencies of three words. Diederich *et al.*, whose work is probably the most similar to ours, performed experiments with short to medium-length German newspaper texts (average length about 720 words, but some as short as 200 words) in which the SVM was trained to distinguish between a target author and all other authors. There were seven target authors in the experiment, with between 82 and 118 texts each, in a set totalling 2,652 texts. The features used were word frequencies in one experiment and bigrams of part-of-speech tags and function words in another, where the tags included very fine-grained morphological information. The experiments were couched in terms of a (simulated) test for plagiarism in which it is to be determined whether the target author is the genuine writer of a target text, and so the margins of the hyperplane were adjusted by a loss function that minimizes false negatives at the price of true positives: missing an instance of plagiarism was considered to be five times more 'expensive' than a false accusation of plagiarism (which was presumed to be happily resolvable by additional evidence). Indeed, the method achieved almost complete avoidance of false negatives, but a recall (i.e. recognition of a target author's work as his or her own) of only 72% when using word frequencies and 61% when using bigrams of tags and function words. Because of the loss-function adjustments and the one-against-many classification, these results cannot be directly compared with those that we will present below.

2 Partial Parsing

2.1 Overview of partial parsing

A partial parser (Abney, 1996) attempts to produce a structural analysis that is more than mere chunking but less than the parse tree of a fully recursive grammar; it aims for speed and for robustness in the face of noisy, unrestricted text. By working from 'islands of certainty', rather than top-down or bottom-up, it aims at the 'containment' of ambiguity rather than its complete resolution.

> A common problem with traditional parsers is that correct low-level phrases are often rejected because they do not fit into a global parse, due to the unavoidable incompleteness of the grammar. This type of fragility is avoided when low-level phrases are judged on their own merits. (Abney, 1996, p. 338)

Abney's partial parser Cass (Abney, 1997), which we use here, uses a cascade of finite-state automata in place of a conventional grammar. It is thus deterministic and nonrecursive.

The input to Cass is a text in which each word has already been tagged with its part of speech. This can be accomplished automatically with high accuracy with a tagger such as that of Brill (1995). The output, the partial parse, may be represented as a tree or as the corresponding set of rewrite-rules for each phrase. For example, consider the following sentence (from chapter 42 of *Villette* by Charlotte Brontë); observe that punctuation marks are treated as separate words.

(1) Let it be theirs to conceive the delight of joy born again fresh out of great terror , the rapture of rescue from peril , the wondrous reprieve from dread , the fruition of return .

Figure 2 shows this text after part-of-speech tagging by the Brill tagger; the tags should be self-explanatory, e.g. VB = base-form verb, PRP = personal pronoun, NN = singular noun, VBN = past-participle verb, RB = adverb, JJ = adjective, IN = preposition. The result of subsequent partial parsing by Cass is shown in tree format in Fig. 3 and in phrase format in Fig. 4. Again, the tags should be largely self-explanatory; most part-of-speech tags pass through without change (except for conversion to lower-case); tags for structures include, e.g. nx = noun chunk, np = noun phrase, ng = noun group, and analogously for verbal and adjectival structures. In the tree format, structure is shown by levels of bracketing, with indentation and line breaks as a visual aid.

```
Let/VB it/PRP be/VB theirs/PRP to/TO conceive/VB the/DT delight/NN
of/IN joy/NN born/VBN again/RB fresh/JJ out/IN of/IN great/JJ
terror/NN ,/, the/DT rapture/NN of/IN rescue/NN from/IN peril/NN
,/, the/DT wondrous/JJ reprieve/NN from/IN dread/NN ,/, the/DT
fruition/NN of/IN return/NN ./.
```

Fig. 2 Output of the Brill part-of-speech tagger for example (1)

```
<sent_break>
  [vp
    [vx
      [vb Let]]]
  [c
    [c0
      [nx
        [prp it]]
      [vx
        [be be]]]
    [nx
      [prp theirs]]]
  [infp
    [inf
      [to to]
      [vb conceive]]
    [ng
      [nx
        [dt the]
        [nn delight]]
      [of of]
      [nx
        [nn joy]]]]
  [vnp
    [vnx
      [vbn born]]
    [ax
      [rb again]
      [jj fresh]]
    [in out]
    [pp
      [of of]
      [nx
        [jj great]
        [nn terror]]]]
  [cma ,]
  [ng
    [nx
      [dt the]
      [nn rapture]]
    [of of]
    [nx
      [nn rescue]]]
  [pp
    [in from]
    [nx
      [nn peril]]]
  [cma ,]
  [nx
    [dt the]
    [jj wondrous]
    [nn reprieve]]
  [pp
    [in from]
    [nx
      [nn dread]]]
  [cma ,]
  [ng
    [nx
      [dt the]
      [nn fruition]]
    [of of]
    [nx
      [nn return]]]
  [per .]
<sent_break>
```

Fig. 3 Output of Cass in tree format for example (1)

In the phrase-rule format, each line shows the application of one rule, listing its category, starting position in text, and children.

```
vx      0     vb
nx      1     prp
vx      2     be
nx      3     prp
inf     4     to vb
nx      6     dt nn
nx      9     nn
vnx     10    vbn
ax      11    rb jj
nx      15    jj nn
nx      18    dt nn
nx      21    nn
nx      23    nn
nx      25    dt jj nn
nx      29    nn
nx      31    dt nn
nx      34    nn
ng      6     nx: of nx:
ng      18    nx: of nx:
ng      31    nx: of nx:
pp      14    of nx:
pp      22    in nx:
pp      28    in nx:
infp    4     inf: ng:
vnp     10    vnx: ax: in pp:
c0      1     nx: vx:
vp      0     vx:
c       1     c0: nx:
```

Fig. 4 Output of Cass in phrase-rule format for example (1). The columns show the category of each phrase, its starting point in the text (where 0 is the point before the first word), and the categories of its children. In the right-hand column, colons indicate nonterminals

2.2 Partial parsing for authorship discrimination

Partial parsing offers a compromise between complete parsing and chunking, and thus a potential solution to the limitations of the methods of both Baayen *et al.* and Stamatatos *et al.* While it does not have the accuracy of a complete, hand-assisted parser, it is possibly nonetheless accurate enough for the task; and it has the additional benefits of being fast and fully automatic. It potentially allows the use of the kinds of features used in each of these studies, and it allows the use of a new kind of feature, syntactic-label bigrams, which we will describe below, that captures some information about order as well.

First, following Baayen *et al.*, we can use the *KDRSW* vocabulary-richness measures on the rules that the partial parser uses, and we can use counts of the most- and least-frequently used rules as features for classification. There are, however, far fewer rules in the default grammar for Cass than in the grammar used in Baayen *et al.*'s

```
vp vx vb c c0 nx prp vx be nx prp infp inf to vb ng nx dt
nn of nx nn vnp vnx vb n ax rb jj in pp of nx jj nn cma ng
nx dt nn of nx nn pp in nx nn cma nx dt jj nn pp in nx nn
cma ng nx dt nn of nx nn per
```

Fig. 5 The stream of labels from the tree-format output of example (1) shown in Fig. 3

corpus; in our experiments to be described below, we observed that 2,360 rules were used.

But second, in compensation for the loss of fine-grainedness in the ‘vocabulary’ of rewrite-rules, we observe that the sequence of labels of bracketed substructures of the partial parse—in canonical display format, the first label of each line or the ‘left edge’ of the display—also contains a considerable amount of information about the syntactic structure of the sentence, and moreover does so very concisely. We can take the stream of labels as a partial representation of the syntactic structure of the sentence (Fig. 5). Instead of just looking at frequencies of these labels, as we might with part-of-speech tags, we can gain additional information about structure by regarding them as an ordered stream and taking the bigrams in it as our basic tokens whose frequencies we use. For example, the stream in Fig. 5 contains the bigrams vp-vx, vx-vb, vb-c, ...

The Cass default grammar contains 126 possible syntactic labels. These comprise twenty-eight non-terminal labels, thirty-six part-of-speech tags for words, and ten tags for punctuation marks that are passed through from the Brill part-of-speech tagger, and fifty-two new part-of-speech tags for words. (The fifty-two new tags, and several of the twenty-eight non-terminal labels, are mostly for use with very specific, easy-to-recognize situations, such as dates, measure phrases (*72 miles*), U.S. city–state pairs (*Peoria, Ill.*), and verbal auxiliaries, that help to create ‘islands of certainty’.) Of course, not all of the $126^2 = 15{,}876$ possible types of bigram are licensed by the grammar; in our experiments, we observed 2,999 types, using 115 different syntactic labels.

3 Experiments

3.1 Texts

In a pilot study, we experimented with our label-bigram frequency method, as described in Section 2.2, on discriminating Charles Dickens’s *David Copperfield* from Jane Austen’s *Sense and Sensibility* and *Emma*. We chose these for the pilot study as these two authors are known to be easily distinguishable; in fact, it can be done by a method as simple as distributions of letter bigrams (Graham *et al*., 2005) (which we verified for these particular novels). Reassuringly, our method performed well in this easy case, but, as will be noted below, it also alerted us to some classification features that were not worth further consideration.

For a serious test of the method, we turned to the harder task of distinguishing the Brontë sisters, Charlotte and Anne. The reason we looked at this pair of authors is that they are, of course, of the same era, same social and economic background, and same gender; they had similar educations; they strongly influenced one another in the development of their writing; and their novels are similar in genre. Any differences can be attributed only to elements of individual style. In their authorship-verification study of twenty-one novels by ten Victorian authors, Koppel *et al*. (2004) found the Brontë sisters to be the hardest to discriminate. We used Charlotte’s *Villette* 1853 and Anne’s *Agnes Gray* 1847 and *The Tenant of Wildfell Hall* 1848. These novels, we determined, can *not* be discriminated just by the distributions of their letter bigrams.

3.2 Preparation of the texts

We used approximately 250,000 words from each author, downloaded from the Project Gutenberg website. Our text pre-processing involved removing chapter titles, finding sentence boundaries using Perl’s Lingua module, and formatting the texts as required for input to the part-of-speech tagger. We used the Brill (1995) part-of-speech tagger, followed by the Cass partial parser (Abney, 1997).

Table 1 The Brontë datasets (approximately 250,000 words from each author)

Block size	Number of blocks
1,000	480
500	942
200	2,232

As a result, each text was represented in two ways: as a stream of syntactic labels, extracted from the tree-format output of Cass, and as a list of the phrase-rules used in the process of partial parsing.

From copies of the processed texts, three datasets were created, which varied by the size of the blocks into which they were broken: roughly 1,000, 500, or 200 words of text. Because the break was always made at the sentence boundary following the required number of words, the average size of each block was actually somewhat larger—1024.8, 524.8, and 223.4 words, respectively—but for simplicity, we will refer to the block sizes as 1,000, 500, and 200. Table 1 shows, for each block size, the number of blocks in the datasets.

3.3 Syntactic features

Our primary features of interest were the frequencies of bigrams of syntactic labels, the frequencies of rewrite-rules, and the application of the *KDRSW* vocabulary-richness measures to rewrite-rules, as described in Section 2.2.

In looking at the frequencies of rewrite-rules, whereas Baayen *et al.* considered only the fifty most-frequently used rules, we experimented with values as high as 150. While experimenting with our pilot dataset (Dickens and Austen), we noticed, however, that the performance of the frequently used rules improves if the *very* frequent rules are not included; this is perhaps analogous to the removal of high-frequency words in some other kinds of text-classification tasks.

Replicating Baayen *et al.*'s use of rewrite-rule frequencies at the lowest-frequency end gave surprisingly bad results. This likely has to do with the difference in grain-size between the partial parser and the annotation of the Nijmegen corpus. Having got nowhere with this feature even with our easiest dataset (Dickens and Austen, 1,000-word blocks), we stopped experimenting with it.

Table 2 Lexical features from Graham *et al.* (2005)

1. Average word length, frequency of i-letter words, $1 \leq i \leq 15$.
2. Average syllables/word, frequency of i-syllable words, $1 \leq i \leq 6$.
3. Average words/sentence.
4. Relative frequencies of 40 function words and 20 punctuation marks (see Graham *et al.* (2005) for lists).
5. Lexical entropy H, Juola's character-level entropy $\hat{L}$.
6. Normalized type/token ratio, Simpson's index D, modified Yule's characteristic K, modified Honoré's measure R.
7. Ratio of hapax legomena and hapax dislegomena to vocabulary size (the latter is S).
8. First five terms of corrected Waring–Herdan distribution.

3.4 Lexical features

We also experimented with a variety of additional features that have been suggested by previous researchers, including vocabulary richness (of words, not rewrite-rules), and average word and sentence length. Specifically, we used the same set of lexical features that Graham *et al.* (2005) chose for their study of paragraph-level authorship discrimination, both those that Graham *et al.* found useful and those whose performance they found poor. A complete list is given in Table 2; we refer to this set below as the Graham feature set. In addition, we used frequency of part-of-speech tags, a feature that straddles (or blurs) the line between the lexical and the syntactic; nonetheless, for convenience we will refer to the Graham feature set and part-of-speech tags collectively as lexical features to distinguish them from our other, purely syntactic, features.

3.5 Features that we did not use

Because Stamatatos *et al.* worked with relatively small texts, and because they used a chunker, we had hoped to use many of their features. In practice, however, a significant portion of their more-successful features could not be used because they were specific to the chunker that they employed. Cass doesn't produce similar analysis-level information; and even if it did so, we judged it preferable not to use features that are highly software-specific, as no general principles follow. Nor did we use the average length of each type of phrase, as Stamatatos *et al*. themselves found this to have little discriminatory power. We did try using the relative frequencies of the most common phrase labels, nx, vx, pp, rx, which closely mirror the relative frequencies of NP, VP, PP, and ADVP chunks. Stamatatos *et al*. found NP and PP chunk frequencies to have good discriminatory power; but we could not replicate this. On the contrary, even on easy cases such as the Dickens and Austen 1,000-word blocks, these features had little power, and so they were dropped from further experiments. Two additional features that we considered but discarded early on in the project were average phrase length (across all types of phrase) and average number of phrases per sentence; these too did not perform well even on the Dickens and Austen datasets.

3.6 Classification with support vector machines

Representing each text as a vector of features, we used support vector machines (see Section 1.3) as a classification method.[3] We used the publicly available software *mySVM* (Rüping, 2000), and found that the dot kernel resulted in better performance than polynomial kernels. In each experimental run, we used 10-fold cross-validation: The datasets were randomly split into ten pieces, and each piece in turn was held out as test data for training on the other nine pieces; the accuracy over all ten runs was then averaged. The data were scaled before training.

4 Results

4.1 Variation within texts

Before the main experiments with our data, we wanted to ensure that variations within a text wouldn't be taken for a difference in authorship. Reassuringly, on all the datasets (Brontë sisters and Dickens–Austen), our method gives baseline-level accuracy—within five percentage points of 50%—when trying to differentiate between two datasets each containing random blocks from the same novel or containing a mixture of texts from the two authors. When we didn't randomize the selection of blocks of text and gave the method two halves of a novel, the accuracy was a little further from random—as far as fifteen percentage points from 50%. This tells us that our method is sensitive to the internal stylistic variations within novels. Nonetheless, the accuracy was low enough to show that within-text variation is not a confound.

Table 3 Average accuracy (in percent) in 10-fold cross-validation on Brontë data, by block size and features used. Boldface indicates best results for each block size

Features	**Block size**		
	1000	**500**	**200**
Syntactic features			
Label bigram freqs	99.0	93.4	84.9
Rule freqs	93.2	93.4	83.8
KDRSW on rules	76.6	76.7	70.3
Bigram and rule freqs	98.4	95.8	87.4
All syntactic features	**99.5**	94.2	87.5
Lexical features			
PoS freqs	93.8	93.4	82.7
Graham features	97.5	90.5	85.6
All lexical features	98.9	95.0	89.5
All features	99.2	**96.8**	**92.4**

4.2 Main experiments

Table 3 presents the results of our main experiments. The misclassification rates of the two authors were never very different, so we report test-set accuracy overall rather than per author, averaged over 10-fold cross-validation; boldface denotes the best result for each dataset. Confidence intervals for these best values are given in Table 4.

Table 3 shows that label bigrams can achieve a very high level of accuracy: high 90s with 1000-word blocks and mid-90s with 500-word blocks, but only mid-80s with 200-word blocks. Table 5 lists the label bigrams that were most discriminating (had high weights) across different block sizes and different trials of the 10-fold cross-validation. For all block sizes, the best accuracies were achieved using either all syntactic features or all syntactic and lexical features. This makes it very clear that features based on fully automatic syntactic analysis are at least as good as lexical features in performance. In fact, for 1,000-word blocks we see something of a ceiling

Table 4 Confidence intervals (CI) for best accuracies for each block size in Table 3

Block size	Best accuracy	Feature set	.95 CI	.99 CI
1000	99.5	All syntactic features	[98.46, 100]	[98.14, 100]
500	96.8	All features	[95.05, 98.59]	[94.49, 99.15]
200	92.4	All features	[90.63, 94.11]	[90.08, 94.66]

Table 5 Label bigrams that were most discriminating across different block sizes and different trials of the 10-fold cross-validation

Bigram	Description
cc c	Coordinating conjunction followed by clause
cma c	Comma followed by clause
prp cma	Personal pronoun followed by comma
vb nx	Verb followed by noun chunk
name nnp	Name starting with proper noun[a]
uh c	Interjection followed by clause
nx nn	Noun chunk starting with common noun
dtp nn	Determiner[b] followed by noun
cc vp	Coordinating conjunction followed by verb phrase

[a] A 'name' in Cass's grammar is a sequence of proper nouns and initials that is not specifically recognized as a 'person' by virtue of starting with a title such as *Mr.* or a first name that appears in Cass's lexicon. Thus *Agnes Gray* is a name (because *Agnes* is not in the lexicon of first names), whereas *Richard Wilson* and *Mrs. Markham* are persons.

[b] From the list *that, this, these, those, few, several, much, many, last, next*.

effect: the result from label bigrams alone, like the result from all lexical features, comes within a fraction of a percentage point, and well within the 0.95 confidence interval, of the result achieved by using all features combined. Moreover, the only truly lexical set of features, the Graham set, does less well except when combined with PoS frequency counts, which, as we noted earlier, is a feature that straddles the lexical and syntactic categories.

Not surprisingly, accuracy declines with block size. Label bigrams show the steepest decline with block size, a drop of more than fourteen percentage points from 1,000-word blocks to 200-word blocks. Nonetheless, the combination of all features maintains an accuracy well above 90% even for 200-word blocks, even though no single feature or smaller combination achieves 90%.

4.3 Reducing the size of the feature space

Having established the best level of accuracy of our method and of various feature sets that include it, we next wanted to see if all of these features are really necessary. While having a large number of features isn't an issue for the SVM algorithm, in addition to automatic authorship attribution, we want to learn about style and what differentiates authors. In this respect, it is useful to determine precisely what features are required to tell them apart.

Looking at Table 3, we see that a number of combinations of feature sets achieve similar accuracies. For example, for 500-word blocks, we get 95–97% accuracy using syntactic-label bigrams and rule frequencies combined, using both sets of lexical features combined, and using all the features together. On the other hand, the *KDRSW* set of features does poorly on its own for all block sizes, and it brings little improvement (the converse, in fact, for 500-word blocks) when combined with the other two syntactic features (compare the 'Bigrams and rules' row with the 'All syntax' row). Our next task, therefore, was to determine which features deserve closer examination, and whether any of them can be done without.

We first decided that, from the observations above, the *KDRSW* feature set is not worth further attention. It's worth noting, however, that this set of features is most helpful on the 1,000-word block size, which points to the possibility of its importance in the analysis of larger texts.

Next, we turned our attention to the other syntactic features. The accuracy achieved by rule frequency counts alone is consistently less than or, at best, equal to that achieved by the label bigrams feature set. Comparing the accuracy achieved by label bigrams and rule frequencies together to that achieved by label bigrams alone, we see minor variations (down 0.6 point, up 2.4–2.5 points). Although this feature set seems to have potential for modest increases in accuracy, especially with smaller

Table 6 Accuracy (in percent) on Brontë sisters with reduced feature sets

Features	Block size		
	1,000	500	200
Label bigrams plus lexical features	99.0	96.3	90.7
Label bigrams (top seventy-five)	96.9	94.2	85.7
Label bigrams (top seventy-five) plus lexical features	98.4	96.6	91.5

text samples, we concluded that syntactic-label bigrams is the most valuable syntactic feature set.

Turning to the lexical features, we see that the combination of PoS frequency counts with the Graham feature set consistently performs better than either set independently. Therefore, we retain both as valuable.

The accuracy achieved by using all features together was very good for all datasets, but having now dismissed some features as not apparently useful, we wanted to find out if using just those that we identified as most valuable, label bigrams and the lexical features, will suffice. Using a combination of only these features resulted in the performance summarized in the first row of Table 6. These results are within a fraction of a percentage point of those achieved with the full set of features for the two larger block sizes, but 1.7 points less for 200-word blocks. However, all are within the 0.95 confidence intervals of the best results with all features (Table 4), confirming that it is safe to discard the *KDRSW* measures and the rule-frequency feature sets.

Our next question was whether we need frequency counts for all 150 most-frequent label bigrams or whether a smaller number would suffice without significant loss of accuracy. (Since our focus is on label bigrams, we didn't experiment with omitting any lexical features.) Our experiments determined that discarding the last seventy-five label bigram counts caused accuracy to drop 2.1 percentage points on 1,000-word blocks while actually increasing slightly on the smaller blocks (see the second row of Table 6 and compare the first row of Table 3). Discarding the last 100 counts, however, caused greater accuracy losses (about 4–5 percentage points), as did our attempt to discard the first twenty-five counts (decrease of about 3–4 percentage points)—in contrast to our earlier results on discarding the twenty-five most-frequent rewrite-rules. We then looked at the performance of the first seventy-five label bigrams combined with the lexical features (see the third row of Table 6). A comparison with the best results in Table 3 shows that the reduction in accuracy compared to much larger feature sets is quite small—1.1 percentage points or less for all block sizes.

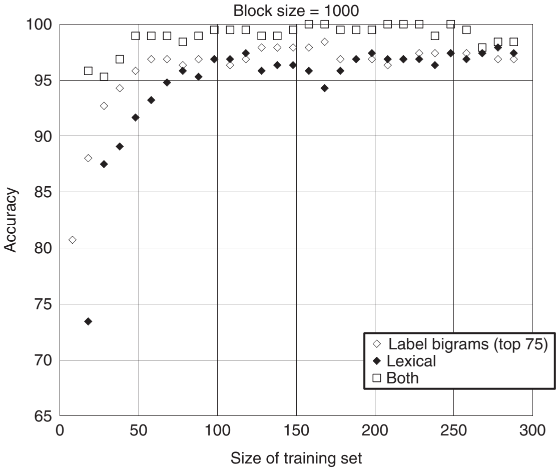

Fig. 6 Accuracy (in percent) of classification by reduced feature sets with respect to size of training set on Brontë sisters, 1000-word blocks

Finally, we examined the performance of this feature set with respect to the size of the training set. Lack of training data is a burning issue in many applications of stylometry, especially in the field of collaborative writing. We therefore prefer methods that can do well with as little training data as possible. For each dataset in this experiment, we varied the size of the training set to see how much it would affect the accuracy. Figs 6–8 show the relationship between the size of the training data (the x axis) and the accuracy of each feature set on the test data (the y axis).[4] The accuracies of label bigrams alone are denoted by open diamonds; those of the lexical features by closed diamonds; and those of both combined by open squares. It's clear from the figures that label

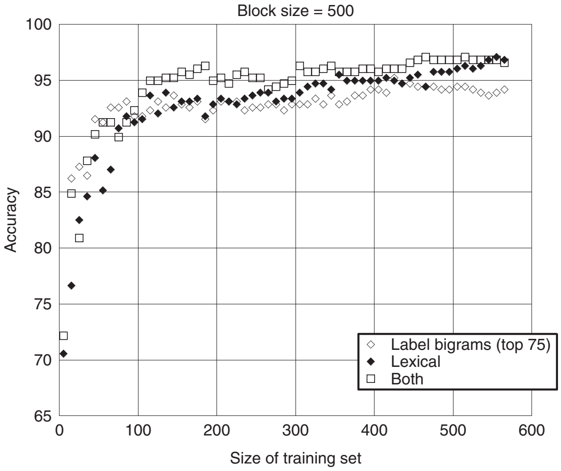

Fig. 7 Accuracy (in percent) of classification by reduced feature sets with respect to size of training set on Brontë sisters, 500-word blocks

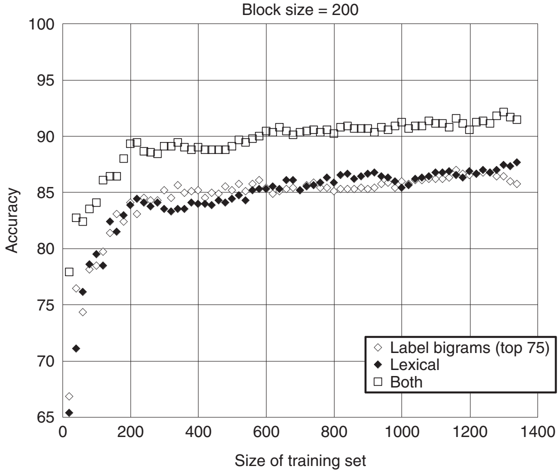

Fig. 8 Accuracy (in percent) of classification by reduced feature sets with respect to size of training set on Brontë sisters, 200-word blocks

bigrams and the lexical features perform about equally well over all; the former dominates for all sizes of the training set with large block sizes (Fig. 6), the latter with medium block sizes (Fig. 7), and neither with small block sizes (Fig. 8). But the combination of both feature sets gives a notably superior performance to either set alone in almost all cases; it is only with larger amounts of training data and the larger block sizes that the individual sets begin to rival it. It's also clear from Fig. 6 that the lexical features are more sensitive to variations in the training set size, especially for smaller sizes. Both the label bigrams alone and the combination of features are more stable with respect to size variations. This affirms the finding of Stamatatos *et al.* that syntactic features are more stable in this respect. For smaller blocks (Figs 7 and 8), the difference in sensitivity is smaller, but still apparent.

5 Conclusion

We have presented bigrams of syntactic labels from a partial parser as a new classification feature for authorship discrimination, and have showed that it can achieve high accuracy in discriminating the work of Anne and Charlotte Brontë, which previous methods have found to be particularly difficult to distinguish. Moreover, high accuracies are achieved even with relatively small fragments of text (little more than 200 words), though the smaller the fragment, the greater the accuracy is boosted by the use of additional lexical features, including (unigram) part-of-speech frequencies.

Of course, the method requires testing on many other authors and genres of text before its generality can be assured. It should also be tested for discrimination of multiple authors (using multi-class SVMs), and for authorship identification. In addition, future research should attempt further development of the features. For example, perhaps unigrams of syntactic labels (i.e. PoS tags plus the additional chunk labels) would be more effective than PoS tags alone. In addition, the set of Graham features should be investigated to see which of its elements are really necessary.

Because of its ability to work with relatively small segments of text, we anticipate that the method will also be applicable in the related task of determining authorship of the individual fragments of a collaboratively written work, including the problem of finding the boundaries of each author's contributions (Graham *et al.*, 2005).

The method bears comparison to contemporaneous work by Gamon and by Chaski. Like us, Gamon (2004) also chose the Brontë sisters and SVMs in his investigation of syntax and semantics for authorship classification in small texts. Gamon's segments were twenty sentences long, which would be an average of roughly 500–600 words per segment. The syntactic features used were much deeper than in the present work: the productions from a complete parse of the text whose frequencies were above an experimentally varied threshold; and at a more superficial syntactic level, part-of-speech trigrams were also used. In addition, Gamon used semantic features such as tense, aspect, and verb subcategorization, and semantic modification relations, such as 'Noun Locn Noun' (a nominal node with a nominal modifier indicating location) whose frequency was above a threshold. He achieved accuracies up to 97.6%.

Like the present work, that of Chaski (2005) also makes use of novel syntactic features to discriminate the authorship of short texts. Her primary features are the markedness/unmarkedness property of each type of syntactic phrase, along with a count of clause-, phrase-, and morpheme-delimiting punctuation in the text. Her method, intended for use in forensic investigations, was tested on sixty-nine texts, averaging 290 words each, written by ten different authors who were asked to write short texts on a variety of topics. Using linear discriminant function analysis as a classifier and leave-one-out cross-validation testing, she achieved an overall accuracy of 95% in discriminating pairs of authors—a result similar to our own. It will be interesting to determine the relative performance of each method when tested on the same data.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada. We are grateful to Neil Graham for his assistance and for the use of his code for the lexical features that were used in his work.

References

Abney, S. (1996). Partial parsing via finite-state cascades. *Natural Language Engineering*, **2**(4): 337–44.

Abney, S. (1997). The SCOL manual, version 0.1b http://www.vinartus.net/spa/.

Baayen, R. H., van Halteren, H., and Tweedie, F. J. (1996). Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, **11**(3): 121–31.

Binongo, J. N. G. and Smith, M. W. A. (1999). The application of principal component analysis to stylometry, *Literary and Linguistic Computing*, **14**(4): 445–466.

Brill, E. (1995). Transformation-based-error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, **21**(4): 543–65.

Burrows, J. (2002). 'Delta': A measure of stylistic difference and likely authorship. *Literary and Linguistic Computing*, **17**(3): 267–87.

Chaski, C. E. (2005). Who's at the keyboard? Authorship attribution in digital evidence investigations. *International Journal of Digital Evidence*, **4**(1).

Diederich, J., Kindermann, J., Leopold, E., and Paass, G. (2003). Authorship attribution with support vector machines. *Applied Intelligence*, **19**(1–2): 109–23.

Fung, G. (2003). The disputed Federalist Papers: SVM feature selection via concave minimization. *Proceedings of the Richard Tapia Celebration of Diversity in Computing*. Atlanta, pp. 42–46.

Gamon, M. (2004). Linguistic correlates of style: Authorship classification with deep linguistic analysis features. *Proceedings, 20th International Conference on Computational Linguistics*. Geneva, pp. 611–617.

Glover, A. and Hirst, G. (1996). Detecting stylistic inconsistencies in collaborative writing. In Sharples, M. and van der Geest, T. (eds), *The New Writing Environment: Writers at work in a world of technology*. London: Springer-Verlag, pp. 147–68.

Graham, N., Hirst, G., and Marthi, B. (2005). Segmenting documents by stylistic character. *Natural Language Engineering*, **11**(4): 397–415.

Holmes, David I. (1994). Authorship attribution. *Computers and the Humanities*, **28**(2): 87–106.

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines: Methods, Theory* *and Algorithms*. Dordrecht: Kluwer Academic Publishers.

Koppel, M., Schler, J., and Mughaz, D. (2004). Text categorization for authorship verification. *Eighth International Symposium on Artificial Intelligence and Mathematics*. Fort Lauderdale, Florida, http://rutcor.rutgers.edu/~amai/aimath04/SpecialSessions/Koppel-aimath04.pdf.

Mosteller, F. and Wallace, D. L. (1964). *Inference and Disputed Authorship: The Federalist*. Reading, MA: Addison-Wesley.

Oostdijk, N. (1991). *Corpus Linguistics and the Automatic Analysis of English*. Amsterdam: Rodopi.

Rüping, S. (2000). *mySVM — Manual*. University of Dortmund, Lehrstuhl Informatik 8. http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/.

Stamatatos, E., Fakotakis, N., and Kokkinakis, G. (2000). Automatic text categorization in terms of genre and author. *Computational Linguistics*, **26**(4): 471–95.

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

Zheng, R., Li, J., Chen, H., and Huang, Z. (2006). A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, **57**(3): 378–93.

Notes

[1] This work was carried out while the author was with the Department of Computer Science, University of Toronto.

[2] Although Zheng *et al*. describe some of their features as 'syntactic', these are actually just frequencies of function words and punctuation. We refer to such features in this article as 'lexical'. Similarly, Zheng *et al*.'s 'structural' features are features of the layout and formatting of the message, not its syntactic or semantic structure.

[3] We also followed the example of many stylometric studies, including that of Baayen *et al*., in using principal component analysis (Binongo and Smith, 1999), and we tried using SVMs after dimensionality reduction by PCA. However, our results using SVMs directly were superior to those involving PCA—PCA didn't help uncover any underlying patterns in the data that couldn't be learned by the SVM directly—so we report only the former.

[4] Because these experiments used different random splits of training and test data from the earlier experiments, the results in these graphs for the complete datasets differ slightly from those in the Tables above.