DERIVING DOCUMENT DESCRIPTORS FROM DATA

Richard S. Forsyth,

Bristol Stylometry Research Unit,
Department of Mathematical Sciences,
University of the West of England,
Bristol BS16 1QY, UK.

email: forsyth_rich@yahoo.co.uk

[Cite as:
Forsyth, R.S. (1997). Deriving Document Descriptors from Data. In: L. Dorfman et al. (eds.) *Emotion, Creativity and Art*. Perm, Russia.
]

Abstract
Recently there has been an upsurge of interest in the problem of text categorization, e.g. of newswire stories (Hayes & Weinstein, 1991; Apté et al., 1993). However, classifying documents is not a new problem: workers in the field of stylometry have been grappling with it for over a hundred years (Mendenhall, 1887). Typically, they have given most attention to authorship attribution, while more modern research in text categorization, conducted from within the paradigm of Artificial Intelligence, has concentrated on discrimination based on subject matter. Nevertheless both fields share similar aims, and it is the contention of the present author that they could profit from being more aware of each other. Accordingly, the present study addresses an issue common to both approaches, the problem of finding an effective set of attributes or features for text discrimination. Stylometers, in their quest to capture consistent and distinctive features of linguistic style, have proposed and used a wide variety of textual features or markers (Holmes, 1994), including measures of vocabulary richness (Yule, 1944), grammatical transition frequencies (Wickmann, 1976), rates of usage of frequent function words (Mosteller & Wallace, 1984), and preferences for words in certain semantic categories (Martindale & McKenzie, 1995). In many text-categorization tasks the choice of textual features is a crucial determinant of success, yet it is not usually treated as a major focus of attention. This is often true of AI-based text-categorization studies as well. It would be desirable if this part of the process were better understood. This paper, therefore, reports an empirical comparison of nine different methods of textual feature-finding that: (1) do not depend on subjective judgement; (2) do not need background knowledge external to the texts being analyzed, such as a lexicon or thesaurus; (3) do not presuppose that the texts being analyzed are in the English language; and (4) do not presume that **words** (or word-based measures) are the only possible textual descriptors. Results of a benchmark test on 13 representative text-classification problems suggest that one of these techniques, here designated *Monte-Carlo Feature-Finding*, has certain advantages that merit consideration by future workers seeking to characterize stylistic habits efficiently without imposing many preconceptions.

Keywords: Feature Selection, Machine Learning, Minimum-Deviance Classification, Monte-Carlo Methods, Pattern Recognition, Stylometry, Text Categorization.

1. Introduction

In their attempts to capture consistent and distinctive features of linguistic style, stylometrists have used a bewildering variety of textual indicators (see: Holmes, 1994). In the majority of stylometric studies, however, the choice of which indicators (or `markers') to use in a given problem is left to the discretion of the investigator. An advantage of this practice is that it allows the exercise of human judgement, and thus can sometimes save a time-consuming search for suitable descriptors. On the other hand, it also inevitably involves subjectivity. Very often the choice of suitable linguistic markers is crucial to the development of an effective discriminant rule; but, being subjective, it may not be replicable on another problem. A further disadvantage is that each stylometrist typically has a `tool-kit' of favoured marker types which encompasses only a small fraction of those that might be used.

The situation is similar in the related fields of multivariate pattern recognition and machine learning (Everitt & Dunn, 1991; Quinlan, 1993): most studies begin by presuming that a suitable set of attributes or features has already been found. In text analysis this presumption is more than usually questionable. It is arguable, for instance, that Mosteller and Wallace (1984 [1964]), in their classic study of *The Federalist Papers*, brought a good deal of background knowledge to the task of finding features that would distinguish Hamilton's from Madison's writings, and that once they had discovered reliable verbal markers such as `upon' and `while' the game was almost over. As part of an automated inductive system, it would be desirable for this part of the process to be less dependent on human expertise.

For these and other reasons, a number of studies have appeared recently (e.g. Burrows, 1992; Binongo, 1994; Burrows & Craig, 1994; Kjell, 1994; Ledger & Merriam, 1994; Bayer et al., 1996) in which the features used as indicators are not imposed by the prior judgement of the analyst but are -- at least to a large extent -- dictated by the texts being analyzed. The main aim of the present paper is to advance this trend, by conducting a test of several different methods of textual feature-finding (some proposed by previous researchers and some newly devised) on a mixed set of text-classification problems. Although no set of textual markers can be entirely free of preconceptions, the methods of feature-finding tested here depend only minimally on human judgement. In addition, none of them presupposes that the text being analyzed is in English.

Thus this paper describes an experiment with a straightforward plan: a single classification technique is applied to 13 test problems using eight different types of textual marker. The main response variable is the proportion of correct classifications achieved on unseen test samples; the main factor of interest is the source of markers.

Before describing these different marker sources, however, it is necessary to give a brief outline of (1) the classification algorithm used and (2) the benchmark problems.

2. Method of Classification Used

As studying the strengths and weaknesses of different classification techniques was not the main focus of this investigation, a simple nearest-centroid classifier was used for the tests reported in this paper. This algorithm, as implemented here, uses a training set of examples with known class membership to compute a centroid (multi-dimensional average) for each category. Then on a fresh or unseen case a measure of distance from (or equivalently, proximity to) each class centroid is computed and the current case is assigned to the category of the centroid which it most resembles. This method is described as "simultaneous key identification" by Sneath & Sokal (1973). It belongs to a class of similarity-based classifiers which McKenzie & Forsyth (1995), among others, have shown to be generally robust in practice.

Most such algorithms use a Euclidean or City-block distance metric, but in the present case the distance measure used is termed deviance. It is computed as follows

$$Deviance(i,c) = \sum_j \frac{(x_{ij} - m_{cj})^2}{(m_{cj} + 1.0)}$$

where i is the current case, j is a feature index, c is a class code, and m_{cj} is the mean value of class c on feature j in the training set. If the features are, for example, word frequencies then x_{ij} is simply the number of times that word j occurs in line i. (This relies on using lines, or blocks, of approximately equal length, as is done here.)

This measure is asymptotically related to the Chi-squared statistic

$$\chi^2 = \sum \frac{(o - e)^2}{e},$$

with m_{cj} being the expected value under the hypothesis that the instance belongs to category c. The +1.0 in the denominator can be seen as a slight bias, downgrading the effect of infrequently occurring features, as well as avoiding division by zero.

Overall, this is a simple, fast and intuitively appealing classification technique that appears to give good results.

3. Materials: The Bristol Benchmark Suite

In many areas of computing, benchmarking is a routine practice. For instance, when compilers are tested for compliance to a programming-language specification, it is normal to apply them to a suite of benchmark cases and record any divergence from expected behaviour.

There is insufficient room here to go into the pros and cons of benchmarking in any depth, except to acknowledge that sets of benchmarks do have drawbacks as well as advantages -- one drawback being that once a a benchmark suite becomes widely accepted as standard, an incentive exists to optimize performance on that suite, possibly at the expense of other problem types. Nevertheless benchmarking does have a role to play in setting objective standards. For example, it is arguable that in the field of forecasting, the work of Makridakis and colleagues

(e.g. Makridakis & Wheelwright, 1989), who tested a number of forecasting methods on a wide range of (mostly economic) time series, transformed the field -- leading to both methodological and practical advances.

Likewise, in machine learning, the general acceptance of the Machine-Learning Database Repository (Murphy & Aha, 1991) as an agreed standard, and its employment as the basis for extensive comparative tests (e.g. Michie et al., 1994) has thrown new light on the strengths and weaknesses of various competing algorithms.

Although billion-byte public-domain archives of text exist, e.g. Project Gutenberg and the Oxford Text Archive, stylometry currently lacks an equivalent set of accepted test problems. Therefore we at Bristol have compiled a textual benchmark suite to meet this need. The current version of this suite is known as Tbench96. Despite its deficiencies, it does present a broader variety of test problems than other workers in stylometry and allied fields have previously used.

3.1 Selection Criteria

The text-categorization problems in this suite were selected to fulfil a number of requirements.

1.Provenance: the true category of each text should be well attested.
2.Variety: problems other than authorship should be included.
3.Language: not all the texts should be in English.
4.Difficulty: both hard and easy problems should be included.
5.Size: the training texts should be of `modest' size, such as might be expected in practical applications.

The last point may need amplification. Although some huge text samples are available, most text-classification tasks in real life require decisions to be made on the basis of samples in the order of thousands or tens of thousands, rather than hundreds of thousands or millions, of words. An enormous training sample of undisputed text is, therefore, something of a luxury. It was felt important that the method used here should be able to perform reasonably well without reliance on this luxury.

Subject to these constraints, 13 test problems were chosen. This collection contains four authorship problems, three chronology problems, three content-based problems, and three miscellaneous problems. As is usual in machine learning, each category of text was divided into non-overlapping training and test sets. Fuller information on Tbench96 is given in Appendix A.

3.2 Pre-processing

In order to impose uniformity of layout and thus reduce the effect of factors such as line-length (not usually an authorial decision and in any case very easy to mimic) all text samples were passed through a program called PRETEXT before being analyzed. This program makes some minor formatting changes: tabs and other white-space characters are converted into blanks; runs of multiple blanks are converted into single blanks; and upper-case letters are converted into lower case. By far the most important change made by PRETEXT, however, is to break running text into segments that are then treated as units or cases to be classified.

Just what consititutes a natural unit of text is by no means obvious. Different researchers have made different decisions about the best way of segmenting long texts and thus turning a single sequence of characters into a number of cases or observations. Some have used fixed-length blocks (e.g. Elliott & Valenza, 1991); others have respected natural subdivisions in the text (e.g. Ule, 1982). Both approaches have merits as well as disadvantages.

Because linguistic materials have a hierarchical structure there is no universally correct segmentation scheme. Textual units could range from single phrases or sentences at one end of the scale to chapters or even whole books at the other. Generally speaking, smaller text units are too short to provide opportunities for stylistic habits to operate on the arrangement of internal constituents, while larger units are insufficiently frequent to provide enough examples for reliable statistical inference. The compromise adopted for the present study was to break all texts into blocks of roughly the same length. In fact, each block boundary was taken as the first new-line in the text on or after the 999th byte in the block being formed. As a result, mean block size is always between 1010 and 1030 characters. Such units will be referred to as kilobyte lines.

The number of words per kilobyte line varies according to the type of writing. A representative figure for Tbench96 as a whole is 185 words per line. This is, in fact, the median word length per line of the five files nearest to the median overall line length (of 1018 bytes). The exact figure is unimportant, though it should be noted that each kilobyte line (almost invariably less than 200 words) is quite short in comparison with the size of text units that previous stylometrists have felt worth analyzing. In other words, this is an attempt to work with text units near the lower limit of what has thus far been considered feasible. Evidence of this is provided by the two quotations below, made 20 years apart.

> "It is clear in the present study that there is considerable loss in discriminatory power when samples fall below 500 words". (Baillie, 1974)

> "We do not think it likely that authorship characteristics would be strongly apparent at levels below say 500 words, or approximately 2500 letters. Even using 500 word samples we should anticipate a great deal of unevenness, and that expectation is confirmed by these results." (Ledger & Merriam, 1994)

Although Felton (1994) has studied 100-word text blocks (in New Testament Greek) and Simonton (1990) even analyzed word usage in the final couplets of Shakespeare's 154 sonnets (averaging 17.6 words per couplet), it remains true that the block size in Tbench96 is small relative to most previous stylometric studies; and therefore that it poses a relatively challenging series of performance tests.

4. Data-Driven Feature-Finding

The main focus of this investigation is a comparative examination of a number of methods of data-driven feature-finding. Only methods that do not require the exercise of judgemental expertise on the part of an investigator were considered. Thus the feature types examined below share the following desirable properties: (1) they are easy to compute; (2) they are interlingual, i.e. they are not limited to the English language.

Perhaps the simplest possible method of obtaining a set of textual features is simply to treat each letter of the alphabet as a feature, i.e. to count the frequency of each of the 26 letters in each kilobyte line. This requires no background knowledge except that the texts concerned are encoded in the Roman alphabet, or have been transliterated into it.

At first glance this would seem not just simple, but simplistic. However, some previous studies -- most notably Ledger (1989) and Ledger & Merriam (1994) -- have reported surprisingly good results when using letter-counts as stylistic indicators. At the very least, this establishes a baseline level of performance: more sophisticated features sets need to outperform letter counting to justify their added complexity.

In fact, the technique included here as a baseline is that of Ule (1982) rather than that of Ledger & Merriam (1994). Ule counted not just the frequencies of the 26 latin letters A to Z (ignoring the distinction between upper and lower case) but also the ten arabic numerals 0 to 9. The reason for this choice was to balance two conflicting aims: on the one hand it is important to compare feature sets of equal size so that if one feature set consists of only 26 attributes then all other features sets should also consist of only 26 attributes; on the other hand it is important not to restrict other features sets, such as words, to an unrealistically small subset. The number 36 was chosen, somewhat arbitrarily, as a compromise between these two conflicting desiderata.

A natural progression from counting the 36 pre-specified alphanumeric characters in each text block is simply to count the frequencies, in each block, of the 36 commonest characters in the training text as a whole. This allows spaces and other punctuation symbols to play a role as discriminators. The danger here is that such characters (e.g. dashes, commas, quotation marks) may be less intrinsic to the original texts than letters. For instance, Shakespeare's punctuation is mostly the work of subsequent editors. Nevertheless, it was decided to investigate the effect of moving from letter-counting to character counting, to find out whether this liberalization confers an advantage, as there are many modern text-categorization problems where punctuation is known to be integral to the text, and eschewing its use amounts to ignoring possibly useful information. (In the two cases in TBench96 where punctuation is not original, JOJO and TROY, punctuation marks have been removed from the files held on disk.)

If counts of single characters are useful as features then so might counts of digrams be useful, or of higher-order n-grams. Kjell (1994) reported good results in assigning *Federalist* essays written either by Hamilton or Madison to the correct authors using a neural-network classifier to which letter-pair frequencies were given as input features. In this study, this approach was extended to characters (not just letters) and generalized to include comparison of trigrams and tetragrams as well.

Another type of textual feature has been used extensively by Burrows (1992) as well as Binongo (1994), among others, not only in authorship attribution but also to distinguish among genres. Essentially it involves finding the most frequently used words and treating the rate of usage of each such word in a given text as a quantitative attribute. The exact number of common words used varies by author and application. Burrows and colleagues (Burrows, 1992; Burrows & Craig, 1994) discuss examples using anywhere from the 50 most common to the 100 most common words. Binongo (1994) uses the commonest 36 words (after excluding pronouns). Greenwood (1995) uses the commonest 32 (in New Testament Greek). In the present study, the most frequent 36 words in the combined training samples of each problem were used -- without exclusions. Most such words are function words, and thus this approach can be said to continue

the tradition, pioneered by Mosteller & Wallace (1984 [1964]), of using frequent function words as markers.

In addition, two novel approaches to textual feature-finding were tested in the present study. These are described more fully in sections 5 and 6.

Altogether, eight different feature types were tested using the benchmark suite, namely those listed in Table 1.

Table 1 -- Types of Text Marker Investigated.

Number	Name	Brief Description
0.	ALPHANUM	26 letters of the Roman alphabet plus numerals 0 to 9
1.	UNIGRAMS	Most frequent 36 characters in the combined training data
2.	DIGRAMS	Most frequent 36 character-pairs in training text
3.	TRIGRAMS	Most frequent 36 character-triplets in training text
4.	TETRAGRAMS	Most frequent 36 character-quadruplets in training text
5.	WORDS	Most frequent 36 words in training text
6.	DOUBLETS	Most frequent 36 substrings found in training text by progressive pairwise chunking (see Section 5)
7.	TEFFSUBS	Most distinctive 36 substrings found in training text by TEFF program (see Section 6)

5. Progressive Pairwise Chunking

To broaden the scope of this comparison somewhat, two novel feature-finding techniques were also tested.

The first of these derives what are called DOUBLETS in Table 1 from text samples, by adapting a method proposed independently by Wolff (1975) and Dawkins (1976) in different contexts.

This algorithm scans byte-encoded text sequences, seeking the most common pair of symbols. At the end of each scan it replaces all occurrences of that pair by a newly allocated code. This process is repeated for the next most common pair, and so on till the requested number of

pairings have been made. The program used here assumes that ASCII codes from 128 onwards are free for reassignment (as is the case with Tbench96) so byte codes from 128 upwards are allocated sequentially. As output the program produces a list of doublets. These need not be digrams, since previously joined doublets can be linked in later passes -- thus building up quite long chains, if they occur in the data, without demanding great computational resources.

The program thus gets away from the artificiality of using fixed-length substrings such as digrams, trigrams or tetragrams. It is able to find markers that are shorter than words (e.g. an affix such as `ed ') or longer (e.g. a collocation such as `of the'). As an illustration, Table 2 lists the doublets found in the training sample of Dylan Thomas's writings (from Tbench96 problem NAMESAKE), in descending order of frequency.

Table 2 -- Doublets from Text by Dylan Thomas.

```
FREQLIST output;  date: 06/25/96 10:15:38
gramsize = 20
1  C:\BM95\DT.TRN
46042 bytes.

1 input file read.
C:\BM95\DT.TRN
Most frequent markers :
   1 `e `                    1694
   2 `th`                    1520
   3 ` th`                   1305
   4 ` the`                  1125
   5 ` the `                  985
   6 `s `                     921
   7 `d `                     906
   8 ` a`                     850
   9 `in`                     764
  10 ` s`                     726
  11 `, `                     712
  12 `an`                     545
  13 ` w`                     544
  14 ` i`                     509
  15 `t `                     475
  16 `er`                     474
  17 ` b`                     447
  18 ` h`                     404
  19 ` m`                     391
  20 `y `                     386
  21 `and`                    384
  22 `ed`                     384
  23 ` f`                     382
  24 `and `                   354
  25 `re`                     349
  26 `ou`                     347
  27 ` and `                  346
  28 `ea`                     341
  29 `of`                     341
  30 ` of`                    334
  31 `es`                     329
  32 `ed `                    328
```

```
33 `on`                    325
34 ` c`                    324
35 `le`                    304
36 `ar`                    301

47 `ing `                  240

61 ` the s`                165

64 `s, `                   150

66 `in the `               143

80 `'s `                   105

91 ` their`                 78
92 ` that `                 71
```

To save space, only the most frequent 36 doublets have been listed, together with a selection of less common doublets that illustrate the potential of the method. It will be seen that as well as pure digrams (e.g. `re'), this program tends to find common trigrams (e.g. `and'), words (e.g. ` the '), morphemes (e.g. `ing '), grammatical endings (e.g. `'s ') and collocations (e.g. `in the '). In addition, it sometimes finds substrings such as `the s' which do not fall naturally into any pre-existing linguistic category: the fact that Dylan Thomas is rather fond of following the definite article with a word starting with the letter `s' is one that more conventional approaches to feature finding might well overlook.

In the present experiment, two versions of this progressive chunking method were tested. In the first, simpler, method, the 36 most common doublets were taken from the combined training text for each of the 13 problems in Tbench96. However, this might be expected to be biased in cases where the different text categories are not represented by samples of equal size, so a second version of this method was also tested. This second method forms a separate list of doublets from each of the text types in each problem, then merges these to produce a set of 36 markers by picking the 36 (distinct) doublets with the lowest ranks, according to frequency. The rank used for a doublet is that with the lowest numerical value in **any** of the text categories under consideration. Thus the markers selected must be common in at least one of the classes of text being discriminated.

6. Monte-Carlo Feature-Finding

The last method of feature-finding tested in this study takes the idea implicit in the progressive chunking method (that it is desirable to employ a variety of marker substrings, both longer and shorter than words) to what may be thought its logical conclusion. Monte-Carlo Feature-Finding is simply a random search for substrings that exist in the training data. Here this process is implemented by a program called TEFF (Text Extending Feature Finder). This finds textual markers (short substrings) without any guidance from the user, merely by searching through a given set of training texts.

Essentially TEFF picks many substrings (3600 in the experiment reported here) from the combined training text completely at random. The length of each substring is also a random number from 1 to 8. All distinct substrings thus found are saved and then ranked according to a distinctiveness measure, i.e. according to a measure of their differential rate of occurrence in the different text categories for the problem concerned. Chi-squared is used to measure distinctiveness. For the present experiment only the most distinctive 36 substrings were kept.

6.1 Elastic Strings

However, earlier trials of Monte-Carlo Feature-Finding (Forsyth, 1995) have indicated that the basic procedure, as described above, tends to generate substrings that are fragmented at what seem linguistically inappropriate boundary points, even when they prove effective as discriminators. For instance, with samples from the *Federalist Papers*, a precursor of TEFF often found `pon' or `upo' instead of `upon '. So TEFF incorporates a procedure that extends each substring, as soon as it is generated, to the maximal length justified by the training text. This refinement does not require the program to have any knowledge of English orthography or morphology.

The idea is that if a substring is embedded in a longer string that has exactly the same occurrence profile then retaining the shorter substring is an inadvertent and probably unwarranted generalization. For example, if `adver' happens always to be part of `advertise' or `advertising' or `advertisement' in every occurrence in a particular sample of text it seems a safer assumption that `advertis' characterizes that text than `adver', which could also appear in `adverbial' or `adverse' or `animadversion' or `inadvertent' -- which, with our knowledge of English, we suspect to characterize rather different kinds of writing.

So TEFF employs a procedure that takes each proposed marker string and tacks onto it character sequences that always precede and/or follow it in the training text. The heart of this process is a routine called Textend(S) that takes a proposed substring S and extends it at both ends if possible. An outline of its operation is given as pseudocode below.

```
REPEAT
IF  S is invariably[1] preceded by the same character C
THEN  S = concatenate(C,S)
IF  S is invariably followed by the same character C
THEN  S = concatenate(S,C)
UNTIL  S reaches maximum size or S is unchanged during loop
```

In TEFF, this procedure is only used within the same category of text that the substring is found in. For example, with the Federalist data, if `upo' were found in the Hamilton sample, as it most probably would be, then a common predecessor/successor would only be sought within that sample.

---

[1] Originally `invariably preceded' (or followed) meant exactly that, but the process was rather slow, so current versions of this procedure actually stop looking after 39 consecutive occurrences of the same predecessor or successor. This does not affect substrings that occur less than 39 times, of course; and appears to make little difference to the rest.

This is a simple but effective procedure which does eliminate the most glaring examples of improper text fragmentation. As this is a rather subjective judgement, a specimen of the results of applying this procedure to a list of substrings produced by from the *Federalist* data is given below as Table 3. This shows each input substring, then a colon, then the resultant extended version of that substring -- both bounded by grave accents to show whether blanks are present before or after. Thus,

```
27  `deraci` : ` confederacies`
```

means that the 27th item was derived from the substring `deraci' which turned out always to be embedded within the longer string ` confederacies'. This listing is intended to give readers a chance to appreciate how the program works and judge its effectiveness.

Table 3 -- Examples of `Stretched' Substrings from Federalist Text.

```
1   `upo` : ` upon `
2   `pon` : `pon`
3   ` would` : ` would `
4   `there ` : ` there `
5   ` on ` : ` on `
6   `up` : `up`
7   `na` : `na`
8   `owers` : `powers`
9   `partmen` : ` department`
10  `wers` : `wers`
11  `epa` : `epa`
12  `ould` : `ould `
13  ` on the` : ` on the`
14  ` on` : ` on`
15  ` form` : ` form`
16  `court` : ` court`
17  `wo` : `wo`
18  `powers ` : `powers `
19  `overnme` : ` government`
20  `ou` : `ou`
21  `ernment` : `ernment`
22  ` there` : ` there`
23  `presi` : `preside`
24  ` cour` : ` cour`
25  `nat` : `nat`
26  `nmen` : `nment`
27  `deraci` : ` confederacies`
28  `dicia` : ` judicia`
29  `he stat` : ` the stat`
30  `heir` : ` their `
31  `ed` : `ed`
32  `cour` : `cour`
33  `feder` : `federa`
34  `nst` : `nst`
35  `onsti` : ` constitu`
36  `ve` : `ve`
37  `e t` : `e t`
38  `, would` : `, would `
```

```
39  `pa` : `pa`
40  `ep` : `ep`
41  `dep` : `dep`
42  `d by` : `d by `
43  `ongres` : ` congress`
44  `e` : `e`
45  `the na` : ` the nat`
46  `stituti` : `stitution`
47  `xecutiv` : ` executive`
48  ` by ` : ` by `
49  ` govern` : ` govern`
50  `execu` : ` execut`
```

It is hoped that readers will agree that expansions such as `partmen' to ` department', `dicia' to ` judicia', `he stat' to ` the stat', `ongres' to ` congress' and `heir' to ` their ' represent gains in clarity.

TEFF cannot eliminate short and apparently unsuitable substrings altogether: it remains somewhat sensitive to misspellings, typographical errors and the presence of rare words or proper names. Further improvements, however, would inevitably make the program more complex and slower, so they have been left as a future development.

6.2 Measuring Distinctiveness

The measure of distinctiveness used in TEFF is Chi-squared (Pearson, 1900), with expected values computed on the basis of equal rates of occurrence in the various categories of text being examined. This has been reported as effective in characterizing textual differences by McMahon et al. (1979) and Hofland & Johansson (1982). However, Chi-squared has also been criticized, e.g. by Church & Hanks (1989) and by Kilgarriff (1996), on the grounds that overall frequency does not distinguish words or strings that occur plentifully in only one section of a text from those that occur more steadily throughout it. Ideally we want markers that **permeate** a particular kind of text. Thus:

"an ideal index should actually be a measure of permeation; that is, it should consider both frequency and distribution information." Stone & Dunphy (1966), p. 33.

In pursuit of this ideal, TEFF offers two measures of distinctiveness. In the first, Chi-squared is computed using unadjusted frequency counts; in the second, the square root of the count for each kilobyte line is accumulated rather than the frequency count itself. This latter measure effectively penalizes strings that occur very often but only in a narrow segment of the text. It thus has a better claim to be an index of permeation than the unmodified Chi-squared score. To assess this claim, both variants were compared in the present experiment.

An illustration of the kind of markers found by TEFF is given in Table 4. These markers were derived from the NEWS problem (see Appendix A), which comprises Associated Press news-wire stories from December 1979 categorized under four headings; respectively: Financial, International, Sporting and Washington news stories. The most distinctive 36 markers (using the square-root transformation of frequency counts) are shown.

Table 4 -- Distinctive Markers for Newswire Stories.

```
TEFF output;  date: 06/09/96 08:17:52

1  C:\BM95\NEWS.F1
82006 bytes.
2  C:\BM95\NEWS.I1
89357 bytes.
3  C:\BM95\NEWS.S1
198772 bytes.
4  C:\BM95\NEWS.W1
106010 bytes.
proportion in class 1 = 0.172228867
proportion in class 2 = 0.187667425
proportion in class 3 = 0.417460629
proportion in class 4 = 0.222642029

square-root transformation applied.

Rank           Marker  Chi-score
1               `game  166.254166    1.  0.  135.  5.
2              ` game  162.480333    1.  0.  133.  5.
3                `;    155.941624    50.  0.  11.  5.
4              `ball   137.041336    0.  3.  115.  3.
5           ` percent  130.11657    61.  4.  15.  24.
6             ` perce  129.036581    61.  4.  15.  25.
7               `bal   124.351617    1.  5.  123.  9.
8             `tball   122.906595    0.  0.  91.  0.
9             `illion  116.045838    57.  11.  8.  44.
10               `aye  114.173206    4.  2.  105.  3.
11             `rter   110.159787    16.  6.  2.  55.
12              `foot  106.239507    0.  0.  81.  1.
13            ` compa  96.9160971    38.  1.  4.  17.
14          ` million  96.7285928    49.  9.  8.  28.
15             `nment  90.3955129    18.  50.  3.  26.
16              ` pla  87.6624853    26.  28.  193.  30.
17           ` govern  86.1895697    17.  53.  7.  26.
18           `e price  84.907721    22.  1.  1.  1.
19       ` government  82.3910106    17.  45.  2.  25.
20             `point  80.5118908    10.  5.  98.  6.
21          ` industr  80.1861149    28.  2.  0.  10.
22             ` gain  80.0638193    24.  2.  4.  0.
23           ` milita  77.487219    1.  30.  0.  19.
24           ` coach   75.5259154    1.  0.  59.  0.
25            `minist  75.2878554    15.  44.  4.  30.
26                ` $  73.8903222    65.  11.  42.  50.
27              `cent  71.7358419    78.  28.  52.  38.
28                `tm  70.6109704    15.  8.  13.  51.
29      `e department  70.258599    5.  3.  1.  31.
30                `8   67.688812    112.  34.  144.  54.
31              ` rep  65.8161221    43.  47.  32.  81.
32          ` market   65.7894658    19.  4.  0.  0.
33          ` missile  65.1953018    0.  19.  0.  3.
34             `arter  65.1355868    23.  15.  36.  73.
35              `cen   64.7014381    80.  31.  59.  39.
36              `son   64.3484833    14.  26.  164.  46.
```

The last four columns give (adjusted) counts for the substring concerned in the four different types of text, in the order: Financial, International, Sporting, and Washington stories. Thus, for example, ` percent' is distinctive of financial stories; ` missile' is distinctive of international stories; `game' is distinctive of sports stories; and ` rep' (as in Representative and Republican) is distinctive of Washington stories. Some markers, such as ` govern' distinguish two types (international and Washington) from the other two.

7. Results

To recapitulate, textual features found by eight different methods were tested on a range of text-categorization problems. As two variants of the last two methods were tested, the factor of interest (type of text marker used) has 10 levels. Thus this experiment has a simple 2-factorial design, with 10 levels on the first factor (source of text markers) and 13 levels on the second (problem number). The latter is essentially a nuisance factor: the problems do differ significantly in difficulty, but this is of no great interest. The main response variable measured is the percentage of correct classifications made **on unseen test data**. Mean values are shown in Table 5.

Table 5 -- Mean Success Rates on Test Data.

Marker Type	Percentage Correct	Klecka's Tau
0. ALPHANUM	69.15	0.4369
1. UNIGRAMS	73.41	0.5078
2. DIGRAMS	70.11	0.4528
3. TRIGRAMS	69.65	0.4426
4. TETRAGRAMS	68.89	0.4301
5. WORDS	68.99	0.4309
6a. DOUBLETS (combined)	70.26	0.4527
6b. DOUBLETS (merged)	69.38	0.4400
7a. TEFFSUBS (with raw counts)	73.77	0.5203
7b. TEFFSUBS (with sqrt of counts)	75.00	0.5325

These figures are aggregated over all 13 problems in Tbench96. Percentage success rate is given because it is familiar and thus easy to interpret. However, as different problems vary in the number of categories to be distinguished (2, 3 or 4), it does not convey a clear impression of improvement over chance expectation.

7.1 A Measure of Relative Error Reduction

For such a purpose Klecka (1980) recommends computing tau, according to the formula below.

$$\tau = \frac{n_c - \sum_{i=1}^{g} p_i n_i}{N - \sum_{i=1}^{g} p_i n_i}$$

Here N is the number of cases in total; n_c is the number of correct classifications; n_i is the number of cases in group i; g is the number of groups; and p_i is the prior probability of a case belonging to group i. In Table 5, above, prior probabilities were taken as equal, so that $p_i = 1/g$ for every category.

Klecka's tau can be considered as measuring the proportional reduction in error compared to chance expectation (here with equal priors). Thus the classifiers make about 50% fewer errors, averaged over all 13 problems, than would be expected from pure guesswork.

7.2 Pre-planned Comparisons

A preliminary analysis involved comparing the percentage success rates achieved by three pairs of marker types that were in some sense related: ALPHANUM versus UNIGRAMS, DOUBLETS combined versus DOUBLETS merged, and TEFFSUBS with and without applying the square-root transformation. As the distribution of the differences obtained from the first of these pairings was visibly skewed, a non-parametric test, the Wilcoxon signed ranks test, was used (Siegel & Castellan, 1988).

The difference between ALPHANUM and UNIGRAMS was significant at the $p<0.05$ level ($W=13$; $p=0.045$). The difference between DOUBLETS using combined text and DOUBLETS obtained from each text type separately and then merged was not significant ($W=29$; $p=0.756$). Nor was the difference between TEFFSUBS selected by Chi-squared on raw frequency counts and that based on adding the square roots of counts in each kilobyte line significant ($W=44$; $p=0.724$). (All probabilities quoted are 2-tailed.)

Thus, while there is no strong evidence that the two variants of DOUBLETS or TEFFSUBS differ in effectiveness, there is evidence that using the most common 36 characters in a combined training text as markers is more effective that using 26 letters plus 10 numerals.

7.3 Analyses of Variance

A 2-way Analysis of Variance was also performed, both on percentage success rate and on Klecka's tau, to investigate the effect of the factors problem number (with 13 levels) and marker type (with 10 levels). Using percentage of successful classifications as a response variable, problem number was a very highly significant factor ($F_{12,108}=74.31$, $p<0.0005$) and marker type was a highly significant factor ($F_{9,108}=2.63$, $p=0.009$). Much the same result was obtained using tau as the dependent variable. Once again the effect of problem number was very highly significant ($F_{12,108}=86.62$, $p<0.0005$) and the effect of marker type was highly significant ($F_{9,108}=2.58$, $p=0.01$). Thus even after allowing for the fact that most of the variance is

attributable to the difference in difficulty of the 13 problems, the Null Hypothesis that all 10 marker types are equally effective must be rejected.

This design does not permit testing for an interaction effect. However, to investigate the factor of marker type further, two subsequent analyses were performed.

Firstly, the percentage success rates were transformed using the arcsine-square-root transformation, as recommended by Zar (1984). This is a variance-stabilizing transformation that is applied to percentages or proportions to make them fit the assumptions of the ANOVA model. A 2-way Analysis of Variance on these transformed data confirmed the results quoted above, indicating a very highly significant main effect of problem number ($F_{12,108}$=78.07, p <0.0005) and a highly significant effect of marker type ($F_{9,108}$=2.77, p<0.006).

Secondly, the residuals or differences of each transformed score from the mean for its problem, were obtained. This is a way of removing the effect of differential problem difficulty by analyzing deviations from the mean score for each problem. Then a 1-way analysis of variance was performed on these residuals, using the 10 levels of marker type as the factor of interest. The result showed a highly significant effect ($F_{9,120}$=3.08, p=0.002).

In addition, the opportunity was taken to apply Dunnett's method of multiple comparisons with a control level, and Hsu's method of multiple comparisons with the best (Minitab, 1991). In the former case, ALPHANUM, the baseline method, was chosen as the control level; in the latter case TEFFSUBS with square-root transformation was the reference level, as it happened to have the highest mean score.

Using an overall alpha value of 0.05, which translates to an individual significance level of 0.00734, Dunnett's test found only one method to be significantly different from the base level (ALPHANUM), namely TEFFSUBS using the square-root transformation.

Using the same overall significance level (0.05) Hsu's multiple-comparison method indicated that all but three methods were significantly different from the best. These three were: UNIGRAMS, TEFFSUBS using raw counts, and TEFFSUBS using the square-root transformation. In other words, only three candidates remain as realistic contenders to be the best method of the 10 tested here.

7.4 A Regression Analysis

To gain another perspective on these figures, a regression analysis was also carried out -- more in exploratory than predictive or hypothesis-testing mode.

To begin with, three indicator variables (taking values 0 or 1 only) were created to indicate the type of problem: authorship, chronology or content (with all three indicators left at zero for the miscellaneous problems). Another four indicator variables were created to deal with marker types. Specifically, one of these indicated whether ALPHANUM was the marker type being used (the only list of markers pre-selected without reference to the training texts); one indicated whether either sort of TEFFSUBS strings were being used (the only marker types selected by distinctiveness rather than frequency); one indicated whether WORDS were being used; and the fourth indicated whether the square-root transformation was being used (with TEFFSUBS). In addition, a variable called Invcats, the reciprocal of the number of categories (i.e. 1/2, 1/3 or

1/4), was also computed as an alternative way of dealing with the unequal difficulty of problems involving different numbers of text categories. This variable is effectively the proportion of correct answers expected by chance, given equal prior probabilities.

Then a stepwise regression was performed with percentage correct classifications on unseen data as the dependent variable and the eight variables described in the previous paragraph as predictors. The procedure halted after including five predictor variables, which together accounted for 71.16% of the variance in success rate, giving the following regression equation.

Percent = 9.597 + 28.9*Auth + 27.3*Chronol + 23.9*Content
+ 87.2*Invcats + 4.4*Teffsubs

While this cannot be taken seriously as a predictive formula (because the selection of problems in Tbench96 is limited and because the assumption of additive effects is questionable) its interpretation does shed some light on the relationships between the types of problem and marker used in this experiment.

The three coefficients for indicators Auth, Chronol, and Content, state, in effect, that the Authorship problems have, on average, a 28.9% higher success rate than the miscellaneous grouping, that Chronology problems have on average a 27.3% higher success rate, and that Content-based problems have on average a 23.9% higher success rate -- other factors being held constant. Thus all the well-posed problems are easier than the miscellaneous group, which includes two quasi-random problems, and there is some suggestion that the Authorship problems selected here are easier than the Content-based problems.

The multiplier for Invcats of 87.2 can be interpreted as saying that, other variables being held at zero, a 2-category problem would be expected to have a 53.197% success rate (87.2 times 1/2 plus the constant of 9.597), a 3-category problem a 38.66% success rate, and a 4-category problem a 31.397% success rate.

Finally the multiplier for TEFFSUBS gives an estimate of the increment in success rate (4.4%) expected by using either variant of the TEFF program. The fact of its selection, as a significant parameter in the equation, suggests that selecting markers for distinctiveness not just frequency does have some beneficial effect.

8. Discussion

In this study a variety of stylometric indicator types have been examined by empirical testing. This has given rise to a preliminary `pecking order' among marker types based on results obtained on a benchmark suite of text classification problems. This ranking suggests firstly that it is worthwhile to allow non-alphanumeric characters as textual features as well as alphanumeric ones, and secondly that it is worthwhile to select textual features for distinctiveness rather than simply by frequency. This latter point may seem obvious, in retrospect, but the possibility did exist that selection using Chi-squared would be too crude to be effective or that selection might lead to over-fitting and hence poorer performance on unseen test data than that achieved by markers selected purely on frequency.

It is also worth noting that one of the simplest ways of finding textual features (UNIGRAMS, which just uses the commonest 36 characters in the training texts) is also one of the best. Moreover, no advantage was evident from using longer n-grams.

The two novel types of textual marker tested in this paper performed creditably. They both merit serious consideration by future researchers in this area.

DOUBLETS obtained by progressive pairwise chunking would appear to be at least as informative as common words, which have previously held pride of place in stylometry. Strings obtained by Monte-Carlo Feature-Finding (TEFFSUBS) gave significantly better results than the other types. While this study is limited in scope, its results do suggest that exclusive concentration on the **word** as the primary type of linguistic indicator may be counter-productive.

Of course this conclusion only carries weight to the extent that the benchmark suite used here (Tbench96) is adequate, and it can only be regarded as a prototype. Nevertheless the very idea of using an agreed suite of benchmark problems to help provide an empirical perspective on various aspects of text classification is in itself a contribution to stylometry; and while Tbench96 is admittedly imperfect, it already presents a wider range of problems than customarily used, and it provides a starting point for future developments.

Much else remains to be done. Two inviting avenues for future research are: (1) combining lexical markers such as used here with syntactic markers (as used, for instance, by Wickmann, 1976) and/or semantic markers (as used, for instance, by Martindale & McKenzie, 1995); (2) investigating interaction effects between problem type (e.g. authorship versus chronology) and marker type. A more extensive benchmark suite might allow investigation of whether certain types of marker are better with certain types of problem. For example: are common words better in authorship studies while TEFF markers work better on content-based discriminations? There was some suggestion of this in the figures obtained here, but the results are inconclusive. An extended benchmark suite would permit serious investigation of such questions.

Finally, more work is needed on feature-selection as well as feature-finding. There is no reason to believe that 36 is an optimal number of features. As explained in section 4, the number 36 was chosen as a somewhat arbitrary compromise, to ensure a `level playing field' for purposes of comparison. By most standards, 36 is rather too many variables for convenience. Certainly, just showing a user a list of 36 textual features, even if they are presented in order of distinctiveness, is unlikely to promote deep insight into the nature of the differences between the text types being studied. If, however, equivalent (or better) performance could be achieved with a reduced subset of markers then the development of an accurate classifier using them would have the beneficial side-effect of promoting deeper insight into the data -- possibly an even more valuable outcome than just having a good classification rule. Initial experiments to this end, using a simple stepwise forward-selection procedure, have proved disappointing. It remains to be seen whether it is intrinsic to the nature of text that accurate classification requires the use of many indicators or whether a more efficient variable-selection algorithm, such as a genetic algorithm or simulated annealing (Siedlecki & Sklansky, 1988; Reeves, 1995), would eliminate this problem.

Goldberg (1995) has argued that textual features have some inherent characteristics -- infrequency, skewed distribution, and high variance -- which together imply that simple yet robust classification rules using just a handful of descriptors will seldom if ever be found for

linguistic materials. This is an interesting conjecture, which is worth attempting to falsify. A variable-selection program, which could reduce the number of textual features needed for successful text classification from around 100 candidates to less than 20 would settle this question. It would also be a useful text-analytic tool. Moreover, a serious attempt to develop such a tool, even if it ended in failure, would have interesting implications, since it would tend to corroborate Goldberg's thesis. I hope to pursue this line of investigation in future.

Appendix A : Details of Benchmark Data Sets

The 13 text-classification problems that constitute TBench96 (Text Benchmark Suite, 1996 edition) form an enhanced version of the test suite used by Forsyth (1995). They also constitute a potentially valuable resource for future studies in text analysis. Collecting and editing Tbench96 has been rather an arduous chore, but enhancing and maintaining it could become a full-time job. Already some of the problems of corpus management (Aijmer & Altenberg, 1991) have presented themselves. It is hoped that support to overcome such problems may in due course be forthcoming, so that successors to Tbench96 may offer a genuine resource to the research community. Ideally they could be made publicly available, e.g. on the Internet, for comparative studies; but, as some of the original texts used are still under copyright, the best way of doing this will require legal advice.

Summary information about the selection of works from various authors and subdivision into training and test files is given below, as well as information concerning the sources and sizes of the texts used in the benchmark suite.

NOTE: A policy adhered to throughout was never to split a single work (article, essay, poem or song) between training and test sets.

A.1 Sources of Benchmark Data

Authorship / Prose

FEDS (2 classes): A selection of papers by two *Federalist* authors, Hamilton and Madison. This celebrated, and difficult, authorship problem -- subject of a ground-breaking stylometric analysis by Moseller & Wallace (1984 [1964]) -- is possibly the best candidate for an accepted benchmark in stylometry.

An electronic text of the entire Federalist papers was obtained by anonymous ftp from Project Gutenberg at
GUTNBERG@vmd.cso.uiuc.edu
For checking purposes the Dent Everyman edition was used (Hamilton et al., 1992 [1788]). Here the division into test and training sets was as follows.

Author	Training paper numbers	Test paper numbers
Hamilton	6, 7, 9, 11, 12, 17, 22, 27, 32, 36, 61, 67, 68, 69, 73, 76, 81	1, 13, 16, 21, 29, 30, 31, 34, 35, 60, 65, 75, 85
Madison	10, 14, 37-48	49-58, 62, 63

Thus, for Madison, all undisputed papers constitute the training set while the `disputed' papers constitute his test sample. This implies that we accept the view expounded by Martindale & McKenzie (1995), who state that: "Mosteller and Wallace's conclusion that Madison wrote the disputed Federalist papers is so firmly established that we may take it as given." For Hamilton, a

random selection of 17 papers was chosen as a training sample with another random selection of 13 papers as test set, giving test and training sets of roughly the same size as Madison's.

JOJO(2 classes): Writings by Joseph Smith, the founder of the Mormon religion, and Joanna Southcott, a relgious prophet contemporary with Smith -- from files kindly donated by Dr David Holmes of UWE Bristol. Southcott's work was supplied in four files: one from her diaries, two files of prophetic meditations, and one file of prophetic verse. Smith's three files were all extracts from his diaries. In his case the third, and largest diary file was taken as test data and the first two jointly as a training sample. In Southcott's case, the training data consists of her diaries and the first file of her prophecies; her test data comprises the second file of prophecies and her verses. As no punctuation was present in any of Southcott's samples, punctuation marks were also removed from Joseph Smith's files. These texts (and others) have been analyzed by Holmes (1992). Although there is a mixture of genres here, inspection of the texts shows that they all have a highly religious flavour, and that both authors modelled their style on the language of the King James Bible.

Authorship / Poetry

EZRA (3 classes): Poems by Ezra Pound, T.S. Eliot and William B. Yeats -- three contemporaries who influenced each other's writings. For example, Pound is known to have given editorial assistance to Yeats and, famously, Eliot (Kamm, 1993).

A random selection of poems by Ezra Pound written up to 1926 was taken from *Selected Poems 1908-1969* (Pound, 1977), and entered by hand. It was supplemented by random selection of 18 pre-1948 *Cantos*, obtained from the Oxford Text Archive. Poems by T.S. Eliot were from *Collected Poems 1909-1962* (Eliot, 1963), scanned then edited by hand. A random selection of 148 poems by W.B. Yeats was taken from the Oxford Text Archive. For checking purposes *Collected Poems* (Yeats, 1961) was used.

In the above case division into training and test sets was made by arranging each author's **files** in order and assigning them alternately to to test or training sets. As this data is held (with a few exceptions) in files each of which contains writings composed by one author in a single year, this mode of division meant that works composed at about the same time were usually kept together; and, even more important, that single poems were never split between test and training files. The effect of this file-based blocking is presumably to make these tests somewhat more stringent than purely random allocation of individual poems to test and training sets would have been. Similar remarks apply to the following problem.

NAMESAKE (2 classes): Poems by Bob Dylan and Dylan Thomas. Songs by Bob Dylan (born Robert A. Zimmerman) were obtained from *Lyrics 1962-1985* (Dylan, 1994). In addition, two tracks from the album *Knocked Out Loaded* (Dylan, 1988) and the whole A-side of *Oh Mercy* (Dylan, 1989) were transcribed by hand and included, to give fuller coverage. An electronic version of *Lyrics 1962-1985* is apparently available from the Oxford Text Archive, but this was not known until after this selection had been compiled. Further information about the Oxford Text Archive can be obtained by sending an electronic mail message to
ARCHIVE@vax.oxford.ac.uk

Poems of Dylan Thomas were obtained from *Collected Poems 1934-1952* (Thomas, 1952) with four more early works added from *Dylan Thomas: the Notebook Poems 1930-1934* (Maud, 1989). Most were typed in by hand.

Chronology

ED(2 classes): Poems by Emily Dickinson, early work being written up to 1863 and later work being written after 1863. Emily Dickinson had a great surge of poetic composition in 1862 and a lesser peak in 1864, after which her output tailed off gradually. The work included is all of *A Choice of Emily Dickinson's Verse* selected by Ted Hughes (Hughes, 1993) as well as a random selection of 32 other poems from the *Complete Poems* (edited by T.H. Johnson, 1970). Data was entered by hand.

JP(3 classes): Poems by John Pudney, divided into three classes. The first category came from *Selected Poems* (Pudney, 1946) and *For Johnny: Poems of World War II* (Pudney, 1976); the second from *Spill Out* (Pudney, 1967) and the third from *Spandrels* (Pudney, 1969). Every distinct poem in these four books was used.

John Pudney (1909-1977) described his career as follows: "My poetic life has been a football match. The war poems were the first half. Then an interval of ten years. Then another go of poetry from 1967 to the present time" (Pudney, 1976). Here the task is to distinguish his war poems (published before 1948) from poems in two other volumes, published in 1967 and 1969 -- i.e. there are three categories.

WY(2 classes): Early and late poems of W.B. Yeats. Early work taken as written up to 1914, the start of the First World War, and later work being written in or after 1916, the date of the Irish Easter Rising, which had a profound effect on Yeats's beliefs about what poetry should aim to achieve. Same source as in EZRA, above.

For these problems the classification objective was to discriminate between early and late works by the same poet. The division into test and training sets for Emily Dickinson and Yeats was once again file-based: in these cases the files of each poet were ordered chronologically and assigned alternately (i.e. from odd then even positions in the sequence) to two sets. The larger of the two resulting files was designated as training and the smaller as test file. With John Pudney, each book was divided into test and training sets by random allocation of individual poems.

Subject-Matter

MAGS (2 classes): This used articles from two academic journals *Literary and Linguistic Computing* and *Machine Learning*. The task was to classify texts according to which journal they came from. In fact, each `article' consisted of the Abstract and first paragraph of a single paper. These were selected by taking a haphazard subset of the volumes actually present on the shelves in UWE's Bolland Library on two separate days, then scanning in (photocopies of) the relevant portions and editing them. The results were as follows.

Literary & Linguistic Computing			Machine Learning		
Year	Articles	Words	Year	Articles	Words
			1987	1	229
			1988	1	200
			1989	1	235
1990	24	6629	1990	15	4069
1991	17	4715	1991	12	3598
1992	15	4489	1992	27	7788
1993	12	3331	1993	6	2041
1994	5	1152	1994	4	1369
1995	2	547	1995	2	530

For both journals the articles from even years were used as the training sample while those from odd years formed the test sample.

NEWS (4 classes): This data-set consists of News stories extracted from the Associated Press wire service during December 1979. A total of about 250,000 words was obtained from the Oxford Text Archive, where it was deposited by Dr G. Akers in 1980. Stories in this archive are classified into at least six mutually exclusive categories. For Tbench96, four of these story types were extracted: F -- Financial stories; I -- International stories; S -- Sports stories; and W -- Washington stories. The Washington category covers US domestic politics, including quite a large number concerning President Carter's re-election campaign plans. This is quite a hard task as, although the Financial and Sports stories have several tell-tale signs, International and Washington stories are often about the same sort of topics. For training data, stories up to 15th December were used. For test data stories after that date were used. (The original files are ordered chronologically.)

TROY (2 classes): Electronic versions of the complete texts of Homer's *Iliad* and *Odyssey*, both transliterated into the Roman alphabet in the same manner, were kindly supplied by Professor Colin Martindale of the University of Maine at Orono. Traditionally each book is divided into 24 sections or `books'. For both works the training sample comprised the odd-numbered books and the test sample consisted of the even-numbered books. The task was to tell which work each kilobyte line came from. (It is possible that this task is an authorship discrimination as well (Griffin, 1980).)

Miscellaneous:

GENDERS (2 classes): short stories written by first-year undergraduate students at the University of Maine on the subject: boy meets girl (or vice versa). These texts were kindly supplied by Professor Colin Martindale of the Psychology Department of the University of Maine at Orono.

These stories arrived in an arbitrary order. Even-numbered stories were used as training data, odd numbered stories as test data. The objective was to distinguish tales written by male authors from those written by female authors. As the authors were in general of similar age, educational attainment and cultural background, this is by no means a trivial task.

AUGUSTAN (2 classes): The Augustan Prose Sample donated by Louis T. Milic to the Oxford Text Archive. For details of the rationale behind this corpus and its later development, see Milic (1990). This data consists of extracts by many English authors during the period 1678 to 1725. It is held as a sequence of records each of which contains a single sentence. Sentence boundaries as identified by Milic were respected.

To obtain test and training sets a program was written to allocate sentences at random to four files. The larger two of these were then treated as training sets, the smaller two as test data.

RASSELAS (2 classes): The complete text of *Rasselas* by Samuel Johnson, written in 1759. This was obtained in electronic form from the Oxford Text Archive. For checking purposes, the Clarendon Press edition was used (Johnson, 1927 [1759]). This novel consists of 49 chapters. These were allocated alternately to four different files. Files 2 and 4 became the training data; files 1 and 3 were used as test sets.

The inclusion of random or quasi-random data may need a few words in justification. The chief objective of doing so here was to provide an opportunity for what statisticians call overfitting to manifest itself. If any of the approaches tested is prone to systematic overfitting -- in the sense of exploiting random peculiarities in the training data -- then this last pair of problems will tend to reveal it. A success rate significantly **below** chance expectation on either of these data sets would be evidence of overfitting. The present author's view is that, as a general rule, some `null' cases should form part of any benchmark suite: as well as finding what patterns do exist, a good classifier should avoid finding patterns that don't exist.

A.2 Sizes of Benchmark Problems

Problem	Categories	Kilobytes (training, test)
FEDS	Alexander Hamilton James Madison	218, 148 227, 140
JOJO	Joseph Smith Joanna Southcott	57, 32 77, 72
EZRA	Ezra Pound T.S. Eliot W.B. Yeats	80, 79 49, 40 109, 86
NAMESAKE	Bob Dylan Dylan Thomas	67, 65 45, 41
ED	Emily Dickinson to 1863 after 1863	28, 23 23, 17
JP	John Pudney: War Poems Poems from Spill Out Poems from Spandrels	16, 10 17, 15 22, 20
WY	W.B. Yeats to 1914 after 1915	61, 34 52, 47
MAGS	Literary & Linguistic Computing Machine Learning	78, 54 88, 44
NEWS	Financial Stories International Stories Sports Stories Washington Stories	80, 67 87, 62 194, 82 103, 71
TROY	Iliad Odyssey	346, 308 251, 255
GENDERS	Stories written by Male students Stories written by Female students	56, 53 99, 103
AUGUSTAN	Random sentences Random sentences	121, 104 108, 108
RASSELAS	Even-numbered chapters Odd-numbered chapters	60, 45 52, 48

Acknowledgements

I thank Dr David Holmes and Professor Colin Martindale for providing some of the text files used in this benchmarking exercise, as well as for helpful comments. In addition, the following

institutions -- the Oxford Text Archive, Project Gutenberg, and UWE's Bolland Library -- have also provided resources without which this study could not have been completed.

References

Aijmer, K. & Altenberg, B. (1991) eds. *English Corpus Linguistics*. Longman, London.

Apté, C., Damerau, F. & Weiss, S.M. (1993). Knowledge Discovery for Document Classification. In: *Proc. AAAI-93 Workshop on Knowledge Discovery in Databases*. AAAI Press, Menlo Park.

Baillie, W.M. (1974). Authorship Attribution in Jacobean Dramatic Texts. In: J.L. Mitchell, ed., *Computers in the Humanities*, Edinburgh Univ. Press.

Bayer, T., Renz, I., Stein, K. & Kressel, U. (1996). Domain and Language Independent Feature Extraction for Statistical Text Categorization. In: L.J. Evett & T.G. Rose, eds., *Language Engineering for Document Analysis and Recognition*. Nottingham Trent University, Nottingham.

Binongo, J.N.G. (1994). Joaquin's Joaquinesquerie, Joaquinesquerie's Joaquin: A Statistical Expression of a Filipino Writer's Style. *Literary & Linguistic Computing*, 9(4), 267-279.

Burrows, J.F. (1992). Not unless you Ask Nicely: the Interpretive Nexus between Analysis and Information. *Literary & Linguistic Computing*, 7(2), 91-109.

Burrows, J.F. & Craig, D.H. (1994). Lyrical Drama and the "Turbid Montebanks": Styles of Dialogue in Romantic and Renaissance Tragedy. *Computers & the Humanities*, 28, 63-86.

Church, K. & Hanks, P. (1989). Word Association Norms, Mutual Information and Lexicography. *Proc. 27th Annual ACL Meeting*, Vancouver, 76-83.

Dawkins, R. (1976). Hierarchical Organisation: a Candidate Principle for Ethology. In: P.P.G. Bateson & R.A. Hinde, eds., *Growing Points in Ethology*. Cambridge University Press.

Dylan, B. (1988). *Knocked Out Loaded*. Sony Music Entertainment Inc.

Dylan, B. (1989). *Oh Mercy*. CBS Records Inc.

Dylan, B. (1994). *Lyrics 1962-1985*. Harper Collins Publishers, London. [Original U.S. edition published 1985.]

Eliot, T.S. (1963). *Collected Poems 1909-1962*. Faber & Faber Limited, London.

Elliott, W.E.Y. & Valenza, R.J. (1991). A Touchstone for the Bard. *Computers & the Humanities*, 25, 199-209.

Everitt, B.S. & Dunn, G. (1991). *Applied Multivariate Data Analysis*. Edward Arnold, London.

Felton, R. (1994). Stylometry -- an Example. *Proc. New Zealand Statistical Assoc. Annual Conf.*, 350-353.

Forsyth, R.S. (1995). *Stylistic Structures: a Computational Approach to Text Classification*. Unpublished Doctoral Thesis, Faculty of Science, University of Nottingham.

Goldberg, J.L. (1995). CDM: an Approach to Learning in Text Categorization. *Proc. 7th IEEE International Conf. on Tools with Artificial Intelligence*.

Greenwood, H.H. (1995). Common Word Frequencies and Authorship in Luke's Gospel and Acts. *Literary & Linguistic Computing*, 10(3), 183-187.

Griffin, J. (1980). *Homer*. Oxford University Press, Oxford.

Hamilton, A., Madison, J. & Jay, J. (1992). *The Federalist Papers*. Everyman edition, edited by W.R. Brock: Dent, London. [First edition, 1788.]

Holmes, D.I. (1992). A Stylometric Analysis of Mormon Scripture and Related Texts. *J. Royal Statistical Society (A)*, 155(1), 91-120.

Holmes, D.I. (1994). Authorship Attribution. *Computers & the Humanities*, 28, 1-20.

Hughes, E.J. (1993). *A Choice of Emily Dickinson's Verse*. Faber & Faber Limited, London.

Johnson, S. (1927). *The History of Rasselas, Prince of Abyssinia*. Clarendon Press, Oxford. [First edition 1759.]

Johnson, T.H. (1970) ed. *Emily Dickinson: Collected Poems*. Faber & Faber Limited, London.

Kamm, A. (1993). *Biographical Dictionary of English Literature*. HarperCollins, Glasgow.

Kilgarriff, A. (1996). Which Words are Particularly Characteristic of a Text? In: L.J. Evett & T.G. Rose, eds., *Language Engineering for Document Analysis and Recognition*. Nottingham Trent University, Nottingham.

Kjell, B. (1994). Authorship Determination Using Letter Pair Frequency Features with Neural Net Classifiers. *Literary & Linguistic Computing*, 9(2), 119-124.

Klecka, W.R. (1980). *Discriminant Analysis*. Sage Publications, Newbury Park.

Ledger, G.R. (1989). *Re-Counting Plato*. Oxford University Press, Oxford.

Ledger, G.R. & Merriam, T.V.N. (1994). Shakespeare, Fletcher, and the Two Noble Kinsmen. *Literary & Linguistic Computing*, 9(3), 235-248.

Makridakis, S. & Wheelwright, S.C. (1989). *Forecasting Methods for Managers*, fifth edition. John Wiley & Sons, New York.

Martindale, C. & McKenzie, D.P. (1995). On the Utility of Content Analysis in Authorship Attribution: the Federalist. *Computers & the Humanities*, 29, in press.

Maud, R. (1989) ed. *Dylan Thomas: the Notebook Poems 1930-1934*. J.M. Dent & Sons Limited, London.

McKenzie, D.P. & Forsyth, R.S. (1995). Classification by Similarity: An Overview of Statistical Methods of Case-Based Reasoning. *Computers in Human Behavior*, 11(2), 273-288.

McMahon, L.E., Cherry, L.L. & Morris, R. (1978). Statistical Text Processing. *Bell System Technical Journal*, 57(6), 2137-2154.

Mendenhall, T.C. (1887). The Characteristic Curves of Composition. *Science*, 11, 237-249, March supplement.

Michie, D., Spiegelhalter, D.J. & Taylor, C.C. (1994) eds. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, Chichester.

Milic, L.T. (1990). The Century of Prose Corpus. *Literary & Linguistic Computing*, 5(3), 203-208.

Minitab Inc. (1991). *Minitab Reference Manual, Release 8*. Minitab Inc., Philadelphia.

Mosteller, F. & Wallace, D.L. (1984). *Applied Bayesian and Classical Inference: the Case of the Federalist Papers*. Springer-Verlag, New York. [Extended edition of: Mosteller & Wallace (1964). *Inference and Disputed Authorship: the Federalist*. Addison-Wesley, Reading, Massachusetts.]

Murphy, P.M. & Aha, D.W. (1991). *UCI Repository of Machine Learning Databases*. Dept. Information & Computer Sceince, University of California at Irvine, CA. [Machine-readable depository: http://www.ics.uci.edu/~mlearn/MLRepository/html.]

Pearson, K. (1900). On the Criterion that a Given System of Deviations from the Probable in the Case of a Correlated System of Variables is Such that It Can Reasonably Be Supposed to Have Arisen from Random Sampling. *Philosophical Magazine*, 6(2), 157-176.

Pound, E.L. (1977). *Selected Poems*. Faber & Faber Limited, London.

Pudney, J.S. (1946). *Selected Poems*. John Lane The Bodley Head Ltd., London.

Pudney, J.S. (1967). *Spill Out*. J.M. Dent & Sons Ltd., London.

Pudney, J.S. (1969). *Spandrels*. J.M. Dent & Sons Ltd., London.

Pudney, J.S. (1976). *For Johnny: Poems of World War II*. Shepheard-Walwyn, London.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California.

Reeves, C.R. (1995). Modern Heuristic Techniques for Combinatorial Problems. McGraw-Hill International, London.

Siedlecki, W. & Sklansky, J. (1989). A Note on Genetic Algorithms for Large-Scale Feature Selection. *Pattern Recognition Letters*, 10, 335-347.

Siegel, S. & Castellan, N.J. (1988). *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, New York.

Simonton, D.K. (1990). Lexical Choices and Aesthetic Success: a Computer Content Analysis of 154 Shakespeare Sonnets. *Computers & the Humanities*, 24, 251-264.

Sneath, P.H.A. & Sokal, R.R. (1973). *Numerical Taxonomy*. W.H. Freeman & Co., San Francisco.

Stone, P.J., Dunphy, D.C., Smith, M.S. & Ogilvie, D.M. (1966). *The General Inquirer: a Computer Approach to Content Analysis in the Behavioral Sciences*. MIT Press, Cambridge, Mass.

Thomas, D.M. (1952). *Collected Poems 1934-1952*. J.M. Dent & Sons Ltd., London.

Ule, L. (1982). Recent Progress in Computer Methods of Authorship Determination. *ALLC Bulletin*, 10(3), 73-89.

Wickmann, D. (1976). On Disputed Authorship, Statistically. *ALLC Bulletin*, 4(1), 32-41.

Wolff, J.G. (1975). An Algorithm for the Segmentation of an Artificial Language Analogue. *Brit. J. Psychology*, 66(1), 79-90.

Yeats, W.B. (1961). *The Collected Poems of W.B. Yeats*. Macmillan & Co. Limited., London.

Yule, G.U. (1944). *The Statistical Study of Literary Vocabulary*. Cambridge University Press, Cambridge.

Zar, J.H. (1984). *Biostatistical Analysis*, second edition. Prentice-Hall, Englewood Cliffs, N.J.