NEW DIRECTIONS IN TEXT CATEGORIZATION

Richard S. Forsyth

[formerly of**: Bristol Stylometry Research Unit**
Department of Mathematical Sciences
University of the West of England
Bristol BS16 1QY
UK.]

[1996: rs-forsyth@csm.uwe.ac.uk]
[2015: forsyth_rich "AT" hotmail "DOTcom"]

Please cite as:
Forsyth, R.S. New directions in text categorization. In Gammerman, A. (1999) ed. *Causal Models and Intelligent Data Management*. Berlin: Springer. (ISBN 3-540-66328-2.)

"As we near the end of the twentieth century, the printed book also appears to be drawing near the end of its five-century career." -- Philip Davies Roberts (1986).

Abstract

As more and more documents are held in machine-readable form, problems of efficient text processing and text analysis become more pressing. An important kind of text processing, which has recently attracted the attention of researchers in Artificial Intelligence (AI), is text categorization, e.g. automatically assigning news stories (Apté et al., 1993) or medical case notes (Lehnert et al., 1995) a suitable category code. However, classifying documents is not a new problem: workers in the field of stylometry have been grappling with it for more than a century (e.g. Mendenhall, 1887). Typically, stylometers have given most attention to authorship attribution and used statistical methods, while AI-based research has concentrated on discrimination by subject matter, using machine-learning techniques. The present paper reports several recent studies drawing on both these traditions. In addition, it investigates various methods of Textual Feature-Finding, i.e. methods of choosing textual features or attributes that: (1) do not depend on subjective judgement; (2) do not need knowledge sources external to the texts being analyzed, such as a computerized lexicon; (3) do not presume that the texts being studied are in English; and (4) do not assume that the **word** is the only possible textual unit.

Keywords: Bayesian Inference, Classification, Discrimination Trees, Evolutionary Computing, Feature Selection, Machine Learning, Monte-Carlo Methods, Pattern Recognition, Stylometry, Text Processing.

1. Introduction

The world is awash with text.

When Saint Alcuin (c.732-804 A.D.) was librarian at York Minster he presided over the revival of learning known as the "Northumbrian Renaissance". Before being destroyed by the Vikings in 866, Alcuin's library held a collection of about 300 manuscript volumes, estimated as almost half the books in Europe at that time (Allinson, 1994). In modern terms, allowing 60,000 words per book and using 6 bytes as a rough estimate of storing each word in a computer memory, that comes to 108 megabytes. Were he alive today, Alcuin could equip himself with a portable laptop computer that could hold four or five copies of his entire library -- without using any form of data compression.

Alcuin flourished during the so-called Dark Ages, notorious as a low point in European scholarship; but even the fabled library of Alexandria, the greatest treasury of literature assembled during classical civilization, is said to have contained no more than half a million papyrus scrolls (Sagan, 1981). As a scroll typically held about 10,000 words, that comprises around 5 billion words or 30 gigabytes of storage -- too much for you or me or Alcuin to carry around in a briefcase, but still not huge by today's standards. For comparison, Leech (1987) states that the text databank of the Mead Data Corporation contained 5 billion words available for on-line retrieval in the mid-1980s. It was already the size of the great library of Alexandria, and has undoubtedly grown since then.

There is, therefore, plenty of text in our industrialized society, much of it stored on computers. Indeed there is so much text being stored on, and transmitted between, computers (as electronic mail, faxes, word-processed documents and so on) that we suffer from text overload. Individuals, and indeed organizations, cannot really cope with such vast amounts of textual information. In other words, our ability to analyze text lags well behind our ability to save, retrieve and send it. This mismatch between ability to store and ability to analyze texts has motivated responses of several kinds -- including, for example, the development of programs that filter out `junk email'.

In these circumstances, automatic text processing and text analysis are becoming more important (Teskey, 1982; Smith, 1990). The present work concentrates on methods of text categorization, where the objective is to find ways of using patterns within texts to sort them as accurately as possible into classes or groups. Historically, the most important kind of text categorization has been authorship attribution, i.e. deciding who wrote a piece of text of unknown or disputed provenance. But other other types of classification task are also important.

- Clinical example: does this transcript come from a patient with damage to Broca's area of the left hemisphere or some other region of the brain?
- Financial example: was this newspaper article written about a company that went bankrupt within nine months of the story being published or not?

It will be seen that if an effective general-purpose text classification system could be developed it would have many useful applications.

The problem of categorizing texts on the basis of objective, preferably quantitative, evidence is not new (Mendenhall, 1887; von Arnim, 1896). It has been the province of stylometers for over a century. Recently researchers in Artificial Intelligence (AI) have started to address this problem by developing trainable text classification systems (Apté et al., 1993; Lehnert et al., 1995). Nevertheless, it appears that most AI researchers are ignorant of stylometry while stylometers are,

with few exceptions, unaware of the latest developments in AI.

In an attempt to encourage fruitful cross-fertilization between these disciplines, the present paper describes recent work in both traditions. Sections 2-4 describe three studies broadly within the AI approach. Sections 5-8 describe four studies from within the stylometric camp. The concluding sections (numbered 9-12) attempt to evaluate the methods described and address some of the outstanding issues raised.

2. Machine Learning for Text Classification

As Holmes (1994) says:

> "Authorship attribution can be regarded as a special kind of pattern recognition where the pattern that is being searched for is the specific feature of the text that is thought to distinguish one author from another."

It might therefore be expected that authorship attribution, and text categorization in general, would be a normal branch of pattern recognition -- itself now subsumed within the field of machine learning (Forsyth, 1989; Weiss & Kulikowski, 1991). However, this is not the case. As will be seen in section 3, artificial neural nets (one species of trainable pattern recognizer) have been used in authorship attribution; but, generally speaking, there is still little commerce between the fields of machine learning and stylometry.

One reason for this state of affairs is that, until very recently, machine-learning systems have only been able to deal with modest amounts of data, typically in the order of a few hundred training cases, whereas practical text classification problems may involve a vast corpus of text. This is beginning to change as methods of "data mining" in large databases are developed (Piatetsky-Shapiro & Frawley, 1991).

A second, more important, reason is that most machine-learning systems work with numeric feature vectors as their normal input representation. In many applications (e.g. classifying sonar signals) the data arrives in numeric form, so this is feature-vector format is convenient; but it is not well suited to representing textual information. It forces the user to interpose a pre-processing step between the raw data and the learning system. Very often the choice of what features to present to the system is the crucial determinant of success; and this is the work of an analyst rather than the computer. The work of Matthews and Merriam (1993), described section 3, exemplifies this: the hard work was done by the investigators in their choice of stylistic indicators. The neural net "learnt" how to weigh these indicators: it did not learn what indicators to use. Much the same applies to their follow-up study of Shakespeare versus Marlowe (Merriam & Matthews, 1994).

Thus, until the 1990s, very little machine learning was carried out on text data. However, as intimated above, this situation is starting to change. For example, Masand et al. (1992) have developed a system for classifying news stories, as has Goldberg (1995); while Ahonen et al. (1993) report a system that learned a small regular grammar capable of describing the structure of all but 82 entries in a Finnish dictionary containing 15970 entries. This shows that that the fields of machine learning and text processing no longer exist in hermetically sealed compartments. As an exemplar of this trend, a study by Apté et al. (1993) is outlined below.

The goal of this research was to create a system that would learn to classify documents, such as

news-wire stories or scientific abstracts, into subject categories. At present the assignment of subject codes is done by people. Huge amounts of text are involved, so the process is expensive. Rule-based systems have been reported that classify documents about as well as humans (e.g. Hayes & Weinstein, 1991; Jacobs, 1993) but they have to be developed anew for each topic area, so are also costly. A system that discovered for itself how to assign subject codes to documents would be very valuable. This is a non-trivial problem partly due to size: the system would have to deal with hundreds of thousands of documents, using tens of thousands of potential attributes and dozens of category codes. Furthermore, a document may well belong to more than one category.

The work of Apté et al. (1993) on this problem consisted of three stages:

(1) a pre-processing step in which attributes used to describe text were selected;
(2) an induction step in which rule sets that would distinguish between classes were found;
(3) an evaluation step in which the rule set was pruned to minimize its expected classification error.

The initial task was to find a set of attributes. Apté et al. wrote a program that scanned texts for words and word pairs that occurred at least five times. They excluded very high-frequency function words, so their indicators were contextual words and pairs of such words. The program found between 1000 and 10,000 indicators, depending on topic.

The next step involved applying the SWAP-1 rule-induction algorithm (Indurkhya & Weiss, 1991) to a large training set of texts using the attributes identified in step 1, above. SWAP-1 uses a heuristic search to find the conjunctive rule (involving logical AND between attributes) that covers as many instances of one class as possible but no counter-examples. Once found this rule is added to the rule-set and the cases it covers are removed from the training data. If cases still remain, the process is repeated: another such rule is induced and the cases it covers are removed. This goes on till the example set is empty. To cope with more than two categories, the program is simply run several times, each time learning a rule-set that distinguishes between a particular category and all the other categories treated as negative instances.

The sort of attributes used and rules produced are illustrated below from a test where the system learned to distinguish between news stories about (American) football and non-football stories.

kicker	=>	football article
injure reserve	=>	football article
award & player	=>	football article

Line 1 shows a simple indicator, a content word; line 2 shows what stylometers would call a collocation; and line 3 shows a logical conjuction of indicators, which, unlike a collocation, need not appear sequentially.

The final step was to prune these rules down to a compact covering set on the basis of performance on a set of validation texts. In this respect working with large data sets (one of their trials was on a collection of over 10,000 Reuters news stories) was a blessing: it allowed them the luxury of using large enough samples in both training and validation phases to derive statistically reliable estimates of likely error rates on fresh cases. This meant that they could purge from the rule-set rules that only covered anomalies in the training data.

Results from a study of stories taken from the United Press International news-wire service are tabulated below.

Table 1 -- Recall & Precision on News-Wire Stories.

Subject	Recall	Precision
Air Transportation	57%	89%
Football	87%	95%
Hockey	84%	91%
Mergers & Acquisitions	39%	71%

These results were obtained on a sample of over a thousand unseen cases. The terms recall and precision are used as in the information-retrieval literature: recall is the percentage of cases in the database of the desired category that are selected by the rules; precision is the percentage of cases selected by the rules that do in fact belong to the category sought.

It can be seen that this system excelled at picking out sporting stories but that performance on what might be thought the most financially valuable task (finding articles about mergers and acquistitions) was poor. Apté et al. looked at some of these stories and concluded that the human coders had made many classification errors on them! (This interpretation may well be right, given the somewhat nebulous nature of some categories in the coding scheme.)

Although not above reproach, this study does indicate that the time has come when machine-learning techniques can usefully be applied to a range of practical text-classification tasks. In particular, it suggests: (1) that the laborious hunt for textual attributes or indicators could be done by machine; and (2) that what a learning system has learned can be represented in a compact and reasonably intelligible form (e.g. as a "rule base" rather than as a mysterious matrix of connection-weights).

Another recent study that tends to support this conclusion is the work by Lehnert et al. (1995) who report on an inductive system that learnt to distinguish two types of "encounter notes" concerning asthmatic patients (written by physicians and held on a computerized medical database) with some success. This system worked by growing a discrimination tree from data.

3. Radial Basis Functions and the Bard

In recent years there has been a resurgence of interest in neural computation, which typically means programming conventional Von Neumann computers to behave as if they contained networks of simple processing units (modelled loosely on biological neurons) each connected to its neighbours by links of varying strengths, known as connection weights. An essential feature of such "neural" nets is that they can be trained by using algorithms that alter the connection weights between nodes in response to feedback comparing the net's actual with its desired output. It has been found that such artificial neural nets exhibit "emergent properties" that enable some hard computational problems to be solved (Aleksander & Morton, 1990; Wasserman, 1993). A favoured area of application for such systems is pattern recognition.

Researchers who have attempted to harness this pattern-recognizing ability to text-categorization tasks include Thirkell (1992), Kjell (1994) and Tweedie et al. (1994). As an example of this type of work, a study by Matthews & Merriam (1993) concerned with the plays of Shakespeare and another Elizabethan playwright, John Fletcher, is described here.

Matthews and Merriam used a neural architecture called the Multi-Layer Perceptron (see, for instance, Beale & Jackson, 1990) with five inputs, two hidden nodes and two output lines, which they trained using the back-propagation training scheme. They wanted the network to learn to distinguish between Shakespeare and Fletcher so they gave it 50 training instances from known works by each author. Since Multi-Layer Perceptrons work with numeric inputs, these training examples were not in text form but were numeric vectors containing five stylistic indicators and an ID code. They compared two different sets of five indicators, based on previous work by Horton (1987) and Merriam (1992).

The Merriam set of five indicators were the following ratios: `did/(did+do)', `no/T10', `no/(no+not)', `to the / to' and `upon/(on+upon)' -- where T10 is the total frequency of 10 common function words suggested as diagnostic of Shakespeare by Taylor (1987). The five discriminators based on Horton's work consisted of ratios calculated by dividing the total number of words in the samples by the number of occurrences of the five function words `are', `in', `no', `of', and `the'.

They trained their nets on 100 examples, 50 from each author, where each example was a feature-vector computed from a 1000-word block of undisputed text. Then they validated both nets on other undisputed examples, using a score they called "Shakespearian Characteristic Measure" or SCM to classify each example unambiguously. SCM was computed by dividing the value on the network's Shakespeare output line (a number between zero and one) by the sum of the outputs on both the Shakespeare and Fletcher output lines. A score above 0.5 was taken as a sign of Shakespearean authorship.

On a validation set of eight plays by Shakespeare and two by Fletcher, both nets gave 10 correct answers out of 10, based on this SCM score. But when individual acts, i.e. samples based on smaller amounts of textual information, were presented to the nets, the Horton-based network performed better than the Merriam-based network, with 85% correct classifications as opposed to 65% correct.

When used on plays of mixed and/or uncertain authorship, the Horton network allocated *Double Falsehood* and *The London Prodigal* to Fletcher and gave *Henry VIII* and *Two Noble Kinsmen* to Shakespeare, although when individual acts were examined it gave results suggestive of collaboration between these two authors in *Henry VIII*. Its verdict was that Acts I, IV and V were by Shakespeare while Acts II and III were by Fletcher. This agrees with majority opinion among literary scholars, derived from other sources of evidence.

According to Wilson (1993), Shakespeare suffered from "scrivener's palsy" by the time that *Henry VIII* was written and had given up play-writing for that reason. He had to be dragged back from retirement in Stratford to help with one last play because his theatrical associates had been persuaded to accept a commission to present a royal command performance to celebrate the wedding of King James's daughter, princess Elizabeth, in February 1613, and -- with the incomparable bard no longer part of their team -- were in danger of missing their deadline. Since the hand that, according to Ben Jonson, had scarce blotted a line, could by then hardly wield a quill, the up-and-coming dramatist John Fletcher was also drafted in to help complete the play on time. In the circumstances, it would have made sense to have allotted the opening scenes and finale to the old

master of stagecraft, while assigning the young tyro to fill in the middle. Thus the historical evidence, such as it is, fits in with what the neural nets found.

Matthews and Merriam broke no new ground as far as neural computing is concerned (Multi-Layer Perceptrons trained by back-propagation being quite old-fashioned in that field) but they did show that artificial neural nets could be applied successfully to a stylometric problem. They also found that, in this case at least, a set of indicators based on simple function-word counts were better discriminators than a similar set consisting of collocations and proportional pairs. Undoubtedly this work will inspire others to follow them into "neural" stylometry.

A criticism that can be made of this work, however, is that their trained nets made mistakes on fairly large samples (entire acts of undisputed plays) even with only two candidate authors. Another criticism applies not just to their work but to all such systems. The "knowledge" gained by a neural network during training is inscrutable: it consists of a matrix of connection weights. If we simply want a black-box classifier, this may not matter; but if we want to understand what stylistic differences the system has discovered then we will be disappointed. Attempts to "de-compile" the knowledge implicit in a connection-weight matrix have been made (e.g. Sejnowski & Rosenberg, 1987) and research in this area continues, but the effort involved is typically an order of magnitude more than that of setting up and training the network in the first place. (For this reason, among others, Holmes & Forsyth (1995) have argued that artificial neural nets are not the most appropriate from of machine learning to apply to stylistic problems.)

Another weakness is that the SCM has no statistical underpinnings. Without further work it is impossible to know what evidential weight should be given to an SCM score of 0.77, for instance. It certainly isn't a probability, and is best regarded just as an index.

In addition, the back-propagation algorithm is something of a "connectionist cliché" (Forsyth, 1990) and suffers from numerous well-known drawbacks of which the most important is that it takes an inordinately long time to learn (Wasserman, 1993; Michie et al., 1994).

Some of these problems have already been solved in a paper by Lowe & Matthews (1995), who used a Radial-Basis-Function network architecture on data extracted from the works of Shakespeare and Fletcher. Such networks are much faster to train than Multi-Layer Perceptrons and their outputs can be made to approximate, with enough training data, posterior conditional probabilities of category membership given data. On the Shakespeare-versus-Fletcher problem, Lowe & Matthews obtained slightly greater accuracy with a Radial-Basis-Function network than with the Multi-Layer Perceptron. However, their choice of indicators was purely judgemental: they still used Horton's five function words as descriptors.

4. An Evolutionary Algorithm for Text Classification

It is becoming recognized that computer-assisted authorship attribution can be seen as a pattern-recognition problem, in which the object of the exercise is to train some sort of classification program to distinguish between positive and negative examples of a particular author's work. It follows from this that techniques developed in the field of Machine Learning can usefully be applied to authorship attribution; and indeed studies within this paradigm have started to appear (see above) -- albeit only recently.

Most studies of this type have employed a `connectionist' or `neural' approach to machine learning.

For example, Merriam & Matthews (1994) trained a Multi-Layer Perceptron Network to distinguish between well-attested samples of writing by the Elizabethan dramatists William Shakespeare and Christopher Marlowe, and then applied the trained network to anonymous works and plays of dubious or mixed authorship, such as Henry VI Part 3, with illuminating results.

A less conventional learning technique was used by Holmes & Forsyth (1995). They employed a rule-finder package based on a `genetic algorithm' to seek relational expressions characterizing the authorial styles of Hamilton and Madison, the main contributors to the *Federalist Papers* (Hamilton et al., 1992 [1788]). This book consists of 85 essays, written in 1787 and 1788, urging New Yorkers to support ratification of the then newly proposed federal constitution of the United Staes of America. The essays were published pseudonymously, over the name Publius (a reference to Publius Valerius Publicola, who led the revolt against the Tarquin kings of ancient Rome). Three people, Alexander Hamilton, John Jay and James Madison, were known to have been involved in the composition of the *Federalist Papers*, but the authorship of specific papers remained in doubt for almost 200 years.

Hamilton was killed in a duel in 1804, but a slip of paper identifying the authors of individual papers was found in a book belonging to a friend, Egbert Benson. This list has never been seen since 1818 but has been accepted as authentic by some later editors. Madison, however, after retiring from the presidency, made a different list in a copy of the book sent to him for correction in 1818. Twelve papers (numbers 49-58, 62 and 63) were claimed by both men. Thus arose a celebrated authorship dispute which became the subject of a landmark stylometric study by Mosteller & Wallace (1984 [1964]). Mosteller & Wallace used both Bayesian and classical statistical techniques to model usage of frequent function words like `and' and `both'. After a series of careful analyses they concluded that the odds overwhelmingly favoured the proposition that all 12 `disputed' papers were written by Madison -- a result now generally accepted. It is assumed that emotional stress, just prior to a fatal duel, impaired Hamilton's memory; or else that Benson made a mistranscription. Possibly both sources of error were responsible for the confusion: Hamilton forgetting about papers 62 and 63 in his haste, and Benson writing 48 instead of 58 (or XLVIII instead of LVIII).

This famous authorship problem is acknowledged as a difficult one, since both Hamilton and Madison wrote about related subjects in common form using very similar styles. It was tackled with the BEAGLE system (Forsyth, 1981), which can be seen as an early precursor of what would nowadays be called genetic programming (Koza, 1992). This software has been fully described elsewhere (Forsyth & Rada, 1986; Forsyth, 1989) so the following paragraphs only summarize the main features of its operation.

The PC/BEAGLE package consists of six main modules, of which only three are of interest here: HERB (Heuristic Evolutionary Rule Breeder) which generates classification rules by a process modelled on the Darwinian idea of natural selection; STEM (Signature Table Evaluation Module) which forms a decision table from the rules produced by HERB; and LEAF (Logical Evaluator And Forecaster) which applies this decision table to classify cases that may not have been seen before.

HERB is in essence a variant of the genetic algorithm (see: Goldberg, 1989; Reeves, 1995) that works with structures that are Boolean relational expressions. Its learning algorithm is outlined below.

1. Evaluate each rule on every training example and compute a fitness score based on the non-parametric correlation between the rule's truth value and the condition being predicted (with a

penalty according to rule length, to encourage brevity).

2. Rank the rules in descending order of merit and remove the bottom half.

3. Replace `dead' rules by crossing a pair of randomly slected survivors, thus assorting and recombining information from better rules.

4. Mutate a small number of rules picked at random (apart from the best rule) and apply a tidying procedure to remove any redundancy thus introduced.

Each pass round this program loop is called a `generation' by analogy with the biological model.

An example of a HERB-generated rule -- taken from a preliminary run on the *Federalist* data where the system was given the task of finding Madison-indicating rules -- is shown below.

((KIND < 0.00002) & (TO < 35.205))

The rule above will be true (indicating Madisonian authorship) when the value of the variable KIND is less than 0.00002 and the value of the variable TO is less than 35.205; otherwise it will be false (indicating non-Madisonian authorship). In this case the variables refer to the frequency, per 1000 words, of the words `kind' and `to'.

HERB requires data in standard `flat-file' format, with a row representing a case or training instance and columns representing variables, as is usual for such packages. In order to present the program with data in a form it could use, an electronic version of the complete text of the *Federalist Papers* was obtained (from Project Gutenberg, at the University of Illinois) and a Snobol program was written to compute frequencies of a user-supplied set of words in each paper. This pre-processing step transformed 1.2 megabytes of text into 85 rows of numbers, where each line represented an individual essay and each column represented the usage rate (scaled per 1000 words) of a particular word.

In all the runs reported, HERB was run for 256 generations and set to produce three rules which were reduced by STEM to two by discarding the worst, i.e. that rule whose absence least degraded the classification performance on the training data. STEM combines rules into a structure called a signature table (a concept introduced by Samuel, 1967) which is used by LEAF as the basis for probabilistic classification of seen or, more interestingly, unseen cases.

The package was trained on 69 papers of known authorship and then tested on the `disputed' papers (which are nowadays generally accepted as being by Madison), using as features the 30 function words identified by Mosteller & Wallace (1984 [1964]) as useful discriminators. It produced quite succinct rules such as

((ON - THERE) < 2.832)
and
((UPON - BOTH) < WHILST)

the first of which is indicative of Hamiltonian authorhsip when true, the second of which is indicative of Madisonian authorship. To interpret the latter: it says that we subtract the rate (per thousand words) of usage of `both' from that of `upon', then if the result is less than the rate of

`whilst' Madison is the likely author; otherwise Hamilton is.

When tested on the unseen 12 papers, odds factors derived from the PC/BEAGLE rules indicated that all 12 were more likely to be by Madison than by Hamilton. On current thinking about the Federalist problem, this corresponds to attributing all 12 correctly -- using a compact collection of quite comprehensible rules.

Thus an evolutionary approach proved itself competitive with more well-established methods of text categorization on a severe test problem. However, once again the features used were preselelected rather than being found by the classifier itself.

5. Text Classification by Vocabulary Richness

A stylometric study by Holmes (1992) represents a refinement of the techniques used by Kjetsaa (1979) (who evaluated an accusation of plagiarism against Mikhail Sholokov, Nobel-prize-winning author of *And Quite Flows the Don*) and thus a step away from reliance on individual word frequencies. This was an investigation of the Mormon scriptures in terms of vocabulary richness (lexical variety) -- an approach originally pioneered by Yule (1944).

The Mormon church claims that its holy book *The Book of Mormon* was translated by Joseph Smith in 1827 from golden plates which he discovered in New York State, engraved in a language called Reformed Egyptian. The book claims to recount the history of a family of Jews who escaped from Jerusalem just before its destruction in 597 BC and sailed to North America. Ever since its publication, doubts have been expressed about its authenticity. Holmes decided to subject the matter to stylometric study, using a method based on five measures of vocabulary richness.

To do so he took 14 extracts from *The Book of Mormon* of approximately 10,000 words each, taking care to edit out reported speech so that the words of each extract were only those of the supposed author of the book concerned (*The Book of Mormon*, like the Bible, being divided into books by a number of different prophets). In addition, he obtained three text samples from private diaries and correspondence written or dictated by Joseph Smith as well as three samples from the biblical book of Isaiah (King James Version). Another Mormon scripture, *The Book of Abraham*, which Smith claimed to have translated from a papyrus written by the Hebrew patriarch Abraham that came into his possession in 1835, was also included. Finally, three segments of about 10,000 words each from *The Doctrine and Covenants* were also included in this analysis. This is a standard work of Mormonism, a collection divine revelations given to Joseph Smith concerning the establishment and regulation of the Church of Latter-Day Saints, published at intervals from 1833 onwards. The main point about this work is that nobody claims it to have been written by anyone other than Smith.

For all these texts, five different measures of vocabulary richness were computed. A hierarchical single-linkage cluster analysis was then performed on this set of 5-dimensional vectors, using a Euclidean distance metric. This divided the texts into three main clusters: Joseph Smith's personal writings formed one group; the three extracts from Isaiah formed another, and all the rest of these samples fell into a third grouping. If the various extracts from this latter group had really been composed by at least seven different people, then this would be very surprising. Moreover, it contradicts a group of Mormon scholars (Larsen et al., 1980) who reported finding linguistic evidence of multiple styles within the *Book of Mormon*.

Next, a principal-component analysis was carried out on the five-dimensional data and the texts

plotted as points in the space defined by the first two principal components. On this basis the texts appeared to fall into four main groups: Joseph Smith's personal writings; the chapters from Isaiah; two Mormon books (Nephi 1 and Mormon 1) as an outlying cluster; and all the rest together. The presence of *The Book of Abraham*, supposedly written almost 1500 years earlier than *The Book of Mormon*, near the middle of this large cluster cast considerable doubt on the proposition that the Mormon scriptures represent the words of several different authors, as does the fact that different extracts nominally from the same prophet (e.g. Alma, Mormon, Moroni and Nephi) showed no tendency to cluster together more tightly among than between prophets.

Holmes concluded that *The Book of Mormon* sprang from the "prophetic voice" of Joseph Smith himself, as did his revelations in *The Book of Abraham*. He also noted that the results of the principal component analysis could be replicated with just two of the five variables used, Honoré's R and the proportion of twice-used words (*dislegomena*) in the vocabulary. Since the former measure (Honoré, 1979) depends on the proportion of once-used words, this suggests that the proportions of words used exactly once or twice in a text can be used as sensitive indices of authorial style. Both measures have been shown to be stable for a given author in texts of more than 1000 words by Sichel (1986).

A problem with this study, from the point of view of text classification, is that the allocation of texts to groups given a plot on the two dimensions of vocabulary richness is done on a rather informal basis. To turn Holmes's method into an automatic classification technique, we would need more information about the distribution of Honoré's R and of *dislegomena* within and between a wide range of authors.

It might also be argued by a devout Mormon that the clustering of texts allegedly by many different prophets into a single group, labelled as Smith's "prophetic voice" by Holmes, is a translation effect, i.e. that Smith imposed his linguistic habits as translator on originally diverse sources. However, the inclusion of the *Doctrine and Covenants* robs this objection of much of its force. This work is not claimed to be a translation (except perhaps from the word of God?) yet it clusters with texts that are so claimed. Hard evidence concerning the relative contribution of translator versus originator to the stylistic structure of translated texts is at present very sparse; but it would surely be a very heavy-handed translator who could wipe out all trace of the original author.

Overall, therefore, this this is an ingenious study that extends the range of proven stylistic indicators and underlines the importance of multivariate methods.

The same approach, using vocabulary richness or lexical diversity to assign text samples to appropriate categories has since been applied to the classic problem of the *Federalist Papers*, with some success (Holmes & Forsyth, 1995). However, Baayen et al. (1996) have recently cast doubt on the robustness of this method. In a comparative study, they found that vocabulary richness was less effective at discriminating between two authors writing in the same genre than syntactic diversity. (See also section 8.)

6. Text Classification with Frequent Function Words

The work of J.F. Burrows represents an even more thoroughly multivariate approach to authorship attribution than that of Holmes, described above. Burrows's methods have several variants and he has used them for other purposes than authorship attribution, but we will only consider here, briefly, his demonstration (Burrows, 1992) that they were capable of distinguishing quite easily between the

three Bronte sisters -- Anne, Charlotte and Emily.

First he found the 75 commonest words in a file of about 56,000 words of text, containing roughly equal amounts written by each of the trio. These texts (retrospective first-person fictional narratives in each case) were then divided into 13 blocks of approximately 4000 words each, five by Anne and four each by Charlotte and Emily. Then the relative frequencies of the 75 commonest words were calculated, giving 13 separate 75-dimensional vectors of numbers. From this data, treated as a 13-by-75 matrix, eigenvectors were computed. The next step was to plot the 13 samples in the 2D space of the first two eigenvectors (equivalent, in practice, to the two main components of a principal-components analysis). The three sets of extracts quite clearly fell into three distinct groups, corresponding to the three different writers. Except for one of Charlotte's extracts, which was nearer to one of Emily's than to any other of her own, every point on this graph had as its nearest neighbour another point from the same author.

Such a clear discrimination among three authors of the same gender, related both by heredity and by upbringing and writing at about the same time in a single narrative form, serves as a very convincing demonstration of the power of multivariate methods in stylometry -- which Burrows has extended to other sets of authors and to differentiating other aspects of style, such as linguistic change over time. In contrast with the laborious search for special verbal indicators of much previous work (e.g. Morton, 1978; Mosteller & Wallace, 1984) this method has the great merit of letting the texts "speak for themselves". It is particularly interesting in this connection, as Burrows points out, that "colourless" largely functional words have so much to say about patterns of relationship among authors.

Recently Binongo (1994) has confirmed the usefulness of this approach by applying essentially the same method, using in his case the 36 commonest words, to distinguish between three Filipino authors who write mainly in English -- with considerable success. Also Greenwood (1995) has applied much the same method to New Testament Greek, using the 32 commonest words to examine works attributed to Saint Luke. Thus using common words as indicators is becoming something of a standard procedure in stylometry. It has proved a surprisingly effective approach, given its basic simplicity.

Nevertheless, this sort of work does have shortcomings as far as automatic text classification is concerned: (1) the dimensions defined by the two most important eigenvectors have no obvious interpretation; and (2) the method of assigning author to extract is usually only implied. Would we assign an unknown point to the class of its nearest neighbour or to the class of the nearest centroid? Do we compute Euclidean or some other distance metric in this space? Should we treat both dimensions equally or weight them by importance in some fashion?

7. Do Authors Have Semantic Signatures?

None of the studies reviewed thus far have tried to find semantic indicators of authorship. Indeed the concentration on "content-free" function words, by Mosteller & Wallace (1984) and Burrows (1992) as well as others, is predicated on an assumption that the meaning of a passage of text gets in the way as far as determining authorship is concerned. A recent study by Martindale & McKenzie (1995), however, challenges this assumption.

Martindale and McKenzie (1995) returned to the *Federalist Papers* as a test case. They took it for granted that Madison wrote all 12 `disputed' papers and looked for alternative ways of establishing

the same conclusion. They tried several methods, of which the most interesting was based on reviving the idea of content analysis, which was popular among social psychologists in the 1960s but which gave way to more formal methods of natural language processing as the field of computational linguistics grew to prominence. As they point out:

"Although content analysis should be useful in author attribution, it has seldom been used. It is not that it has previously been tried with no success. It has simply not often been tried." (Martindale & McKenzie, 1995)

In one of their analyses they used a program which is essentially an updated reimplementation of the General Inquirer of Stone et al. (1966). This puts words into one of 55 mutually exclusive semantic categories. Ignoring function words, it usually categorizes about 80 to 90 percent of the words in a text. It uses a kind of thesaurus called the Harvard Psychosocial Dictionary (Mark III) which includes categories such as those in Table 2.

Table 2 -- Examples of some Psychosocial Categories.

Semantic Category	Sample Words in that Category
Male Role	boy, brother
Thought Form	basic, contrast
Urge	eager, incentive
Ought	duty, ought, proper
Move	pull, run

Analysis of the relative frequencies produced by this program suggested that Hamilton used the categories Large Group, Bad, Urge and Ought significantly more than Madison, while Madison used words falling under the headings Quantity and Thought Form more frequently than Hamilton.

This implied that a discrimination rule could be based on such semantic information. Accordingly, the relative frequencies of these 55 categories were used to give numeric feature-vectors for all papers by Hamilton and Madison as well as the disputed papers. Then, having reduced the dimensionality of the data from 55 dimensions to five by Multidimensional Scaling, a linear discriminant function was formed on the papers of known authorship. This function, when applied to the disputed papers, classified 9 as by Madison, misclassifying 3 (numbers 50, 52 and 54) as Hamilton's. In addition, a type of neural-network classifier called the Probabilistic Neural network, or PNN (Specht & Shapiro, 1991), was trained on the same data. When this was used with the disputed papers it gave 10 to Madison and 2 (numbers 52 and 54) to Hamilton.

Thus the neural classifier appeared marginally superior to the linear discriminant -- provided we accept Mosteller and Wallace's verdict on the disputed *Federalist* papers. In any case, it would seem from this study that the practice of ignoring semantic information almost entirely in authorship investigations may be misguided. After all, this must rank as a hard problem for content-based discrimination, since both Hamilton and Madison were writing on essentially the same subject. As Martindale and McKenzie note, the semantic discrimination rules had no trouble with paper 55 (a problem case for Mosteller & Wallace's methods); thus content-based measures may have a valuable

role to play alongside measures derived from function-word frequencies, since these two types of indicator tend to make mistakes under different conditions.

Moreover, this approach is likely to prove well suited to more commercially promising text-classification tasks, such as categorizing newswire stories, since these need to be distinguished by content rather than author.

8. Syntax with Style

Most stylometers have avoided using syntactic or grammatical descriptors, preferring to rely predominantly on measures derived from word counts. There are good reasons for this preference: (1) parsing of unrestricted natural-language texts has, until very recently, been an arduous and error-prone process; (2) lexical methods work in any language whereas grammatical features vary between languages.

However, there have been exceptions to this rule. For example, Milic (1967) analyzed the style of Jonathan Swift, compared with some contemporaries. This involved hand-coding a large sample of text, written by Swift and some other Augustan authors, according to a version of Fries's grammatical scheme (Fries, 1952). Wickmann (1976) used a matrix giving frequencies of transitions between syntactic categories to settle (more or less) the authorship question surrounding an anonymous German Romantic novel, published in 1804, called *Nachtwachen*. Again, this involved laborious hand-coding, which is presumably why syntactic studies have been comparatively rare in stylometry.

Recently, this line of investigation has been taken up again in a study by Baayen et al. (1996), who compared texts in the same genre by two different authors which were part of the Nijmegen corpus (Keulen, 1986). These texts, detective fiction by Allingham or by Innes, had been syntactically annotated using the TOSCA coding scheme (Oostdijk, 1991).

In a carefully conducted study that included a `blind' trial, Baayen et al. demonstrated that -- for this pair of authors at least -- using syntactic information gave more accurate classification than using measures of vocabulary richness (cf. section 5) or function-word frequencies (cf. section 6).

> "We interpret this result as confirming our initial intuition that the use of function words for classificatory purposes is an economical way of tapping into the use of syntax, but that direct examination of the frequencies of syntactic constructions leads to a higher discriminatory resolution." (Baayen et al., 1996)

As with the earlier studies by Milic and Wickmann, this result was only achieved after laborious hand-coding; and Baayen et al. are pessimistic about the practicality of automatic parsing in the near future. However, recent development of robust, automatic taggers, which allocate grammatical codes to words even when a full parse tree cannot be generated, such as ENGCG (Voutilainen et al., 1992) and AUTASYS (Fang & Nelson, 1994), suggests that this (syntactic) approach to text categorization may well be ripe for a revival -- although only for texts in languages such as English where such software is available.

9. Intermezzo

A major premiss of this paper is that text categorization is usefully regarded as a type of pattern

recognition and therefore that techniques from the field of machine learning can be applied to authorship attribution and other text-classification tasks. From this point of view, the fact that there are some recent signs of convergence (outlined above) between the previously separate areas of stylometry and machine learning is a welcome development.

This view helps to provide a framework for organizing the diversity of studies described above which (although much important work has been left out in the interests of brevity) still present a wide range of techniques applied in a variety of domains. From this standpoint, the goal of text classification is not so much to assign a small number of texts to their correct category but to develop a rule or procedure for doing so, i.e. to produce a classifier. This will typically involve several stages. Thus a generic text-classification task falls naturally into five main phases:

(1) Data collection, possibly including pre-processing;
(2) Feature selection;
(3) Induction of a rule or rules based on selected features;
(4) Validation of the rule or rule-set;
(5) Application of the rule.

(Apté et al. (1993) describe only the middle three of these stages, as noted in section 2, perhaps considering the initial and final stages too obvious to mention.)

If further progress is to be made in automatic text categorization, improvements will need to be made in all of these five phases. In the rest of this article we will concentrate on work by the present author, indicative of some potential lines of advance in only two of them:

Feature Selection:
- Automating the hunt for markers;
- Using markers other than words, both below the word level (e.g. suffixes) and above (e.g. collocations).

Rule Induction:
- Establishing which learning algorithms work best with which sorts of textual data.

10. Some Methods of Textual Feature-Finding

Most classification algorithms need to be given a vector of feature values, describing the objects to be classified. Where do such features come from? An empirical study by Forsyth & Holmes (1996) sheds some light on this question.

In their attempts to capture distinctive features of linguistic style, stylometers have used a bewildering variety of textual indicators (see: Holmes, 1994). In the majority of studies, however, the choice of which indicators (or `markers') to use in a given problem is left to the discretion of the investigator (e.g. Dixon & Mannion, 1993; Matthews & Merriam, 1993; Merriam & Matthews, 1994; Holmes & Forsyth, 1995). 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 researcher typically has a `tool-kit' of favourite 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 clearly 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) 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. Data-derived textual feature types that have been proposed include:

1. Letters (Ledger & Meriam, 1994);
2. Most Common Words (Burrows, 1992);
3. Digrams (Kjell, 1994).

The first and simplest method 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 block of text. At first glance this would seem not just simple, but simplistic. However, several previous studies -- most notably Ledger & Merriam (1994), but also Ule (1982) and Ledger (1989) -- have reported surprisingly good results when using letter-counts as stylistic indicators.

The second type of textual feature has been used 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 feature. 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). 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.

The third method uses digram counts or letter-pairs as features. 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.

Note that these three methods of feature-finding share four desirable properties: (1) they are easy to compute; (2) they are easy to explain; (3) they are interlingual, i.e. they are not limited to English; (4) they require no exercise of skill by a user but can be found quite automatically. These four properties also apply to the next two methods, which were devised by the present author.

10.1 Progressive Pairwise Chunking

The first of these is here called progressive pairwise chunking. It attempts to avoid the artificiality of always using fixed-length markers (such as digrams or trigrams) while also allowing marker substrings that are shorter than words (e.g. an affix such as `ed ') or which cross word boundaries (e.g. a collocation such as `in the'). This is done by adapting a method first described (in different contexts) by Wolff (1975) and Dawkins (1976).

Essentially the algorithm scans a byte-encoded text sequence, looking for the most common pair of symbols. At the end of each scan it replaces all occurrences of that pair by a newly allocated digram 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 character codes from ASCII 128 upwards are free for reassignment (as is the case with the tests described below), so byte codes from 128 onwards are allocated sequentially. Its output is a list of doublets. These are not always digrams, since previously concatenated doublets can be linked in later passes. Thus the program can build up quite long chains, if they occur in the data -- identifying sequential dependencies of quite a high order (in a Markovian sense) without demanding excessive computational resources. In particular, it does not need the huge but sparsely filled multi-dimensional matrices that would be required by a simple-minded approach to analyzing transition probabilities spanning more than a few items.

An extract from this program's output when applied to a file containing poems by Bob Dylan and Dylan Thomas is shown as Table 3. For the sake of brevity, only the most common 16 substrings are shown, plus a selection of less common strings that illustrate the potential of this method.

Table 3 -- Example of Substrings Formed by Progressive Pairwise Chunking.

```
FREQLIST output;  date: 01/04/96 12:52:30
1  C:\BM95\BD.TRN    68016 bytes.
2  C:\BM95\DT.TRN    45952 bytes.
113968 bytes:  2 input files read.

Most frequent markers :
   1 `e `                    4030
   2 ` t`                    3501
   3 `th`                    2994
   4 ` th`                   2482
   5 `s `                    2122
   6 ` a`                    2075
   7 ` the`                  2042
   8 `d `                    1963
   9 `in`                    1814
  10 ` s`                    1772
  11 `t `                    1767
  12 ` the `                 1668
  13 `, `                    1637
  14 ` i`                    1500
  15 ` w`                    1486
  16 `an`                    1412

  29 `and `                   742

  32 ` and `                  700
```

41	`ing`	637
42	`you`	623
43	` you`	618
46	`ed `	590
60	` to `	417
62	`wh`	389
76	`'s `	293
77	`e, `	293
80	`s, `	273
81	` the s`	269
90	`in the `	210
96	`ver`	159

It will be seen that, as well as pure digrams, this method tends to find common trigrams (e.g. `ver'), words (e.g. ` the '), morphemes (e.g. `ing', `'s ') and collocations (e.g. `in the'). Some of the other substrings, such as `the s', do not fall naturally into any pre-existing linguistic grouping. As it turns out, Dylan Thomas is rather fond of following the definite article with the letter `s' -- a fact that more conventional methods of feature finding would fail to exploit.

10.2 Monte-Carlo Feature-Finding

Another method of feature-finding tested by Forsyth & Holmes (1996) 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 CHISUBS. This finds textual markers (short substrings) without any guidance from the user, merely by searching through a given set of training texts.

The operation of CHISUBS may be described very simply. Firstly the program repeatedly extracts substrings of length S (where S is a random number between 1 and L) from randomly chosen locations in the training text until N distinct strings have been found. Next the best C of these substrings are retained and printed, where `best' means having the highest Chi-squared score. Chi-squared is used as an index of distinctiveness here because McMahon et al. (1978) used it succesfully for a similar purpose -- expected values being calculated on the basis of equal rates of usage.

In the experiments reported here, the values for the parameters mentioned above were as follows: L=7, N=3600, C=96. Thus the program sought 3600 different substrings, of from 1 to 7 characters long, and then retained only the most discriminating 96 of them.

Table 4 shows the result of running CHISUBS on the *Federalist* training data. It illustrates the sort of markers found by this process. Only the best 26 markers have been listed, to save space.

Table 4 -- Marker Substrings Derived from Federalist Data.

```
CHISUBS output;  date: 01/04/96 13:35:23
gramsize = 7
1  C:\BM95\HAMILTON.TRN
224555 bytes.
2  C:\BM95\MADISON.TRN
232931 bytes.
proportion in class 1 = 0.490845085
proportion in class 2 = 0.509153822

Grams kept = 96

Rank         Substring  Chi-score    Frequencies
1                `pon`  90.7905528    141.  22.
2             ` would`  69.4649456    334.  158.
3             `there `  68.7778962    137.  32.
4               ` wou`  67.7237888    334.  160.
5               ` on `  67.0970655    119.  293.
6            ` would `  64.6422832    322.  155.
7                `up`  61.7127024    292.  137.
8                `na`  58.418292    693.  455.
9              `owers`  56.8864289    30.  128.
10           `partmen`  56.2652439    12.  90.
11              `wers`  51.6241599    34.  129.
12               `epa`  51.3068526    31.  123.
13              `ould`  49.8865689    461.  282.
14               `oul`  49.3130926    461.  283.
15           ` on the`  47.318841    52.  155.
16            `ould `  46.4552952    446.  276.
17              ` on`  44.845061    288.  489.
18            ` form`  44.3970607    45.  139.
19            `court`  42.6803592    72.  13.
20               `wo`  42.6386397    399.  245.
21           `powers `  40.9096955    22.  93.
22           `governm`  40.6851794    149.  291.
23               `ou`  37.9349158    1285.  1031.
24           `ernment`  37.2283842    154.  291.
25            ` there`  36.5487883    160.  72.
26             `presi`  36.2364995    67.  14.

[The grave accent (`) is used here as a string delimiter, since single and
      double quotation marks may occur in these text markers.]
```

To interpret this listing, it should be noted that, in the last two columns, the frequency of usage in Hamilton's training sample comes before the frequency in Madison's. As both authors are represented by almost the same amount of text, this can be read as saying that ` would', for example, is a Hamilton marker (334 : 158) while ` on the ' is a Madison marker (52 : 155). Thus even this simple printout provides some interesting information -- although many of these items appear to be what Mosteller & Wallace (1984) would call "dangerously contextual".

Whether such reliance on contextual or content-bearing linguistic items is a weakness can be answered by empirical testing; but CHISUBS also suffers from two structural faults which, unless corrected, would detract from its appeal even if such markers prove effective in practice. Firstly, as

can easily be seen, there is plenty of redundancy (e.g. ` would' as well as ` would '). Secondly, these text fragments are often segmented at what seem to be inappropriate boundary points (e.g. `governm', which surely ought to be ` government'). The most notable example of improper fragmentation in Table 4 is `pon', which anyone acquainted with the *Federalist* problem will immediately realize is an imperfect surrogate for ` upon'.

10.3 How Long is a Piece of Substring?

CHISUBS has no background knowledge: it knows nothing about words, morphemes, punctuation, parts of speech or anything specific to English or other languages. It treats text simply as a sequence of bytes. This lack of preconceptions is an advantage in that it could deal with other natural languages such as Latin, artificial languages such as C++, or indeed non-linguistic material such as coded protein sequences, without amendment. But it has the disadvantage that it often produces substring markers which, to a user, appear to be truncated at inappropriate places. Examples of this problem are `rpus' instead of `corpus' and, most irritatingly, `pon' instead of ` upon' from the *Federalist* samples.

So the CHISUBS program has been modified to alleviate this problem -- without the need to introduce any background knowledge such as a lexicon or morphological rules (specific to English) that would reduce the generality of the method. The revised program, TEFF (Textual Extended Feature Finder), picks short substrings at random by the same method as described, but each substring is `stretched' as much as compatible with the data as soon as it is generated and before being saved for evaluation.

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. This simple but effective procedure does seem to 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 CHISUBS from a *Federalist* text sample is given below as Table 5. 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'. From this listing it hoped that readers will be able to appreciate how the program works and judge its effectiveness.

Table 5 -- 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 `
```

[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.

```
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  `feder` : `federa`
33  `ongres` : ` congress`
```

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; but it does, despite its simplicity, offer an improvement in intelligibility over the basic Monte-Carlo feature-finder (CHISUBS).

10.4 Comparative Testing

To recapitulate, textual features found by five different methods were tested by Forsyth & Holmes (1996) on a range of 10 text-categorization problems. These five methods are summarized in Table 6. Note that only the last type of marker is selected according to distinctiveness: the rest are chosen solely by frequency.

Table 6 -- Types of Textual marker Tested.

Name	Brief Description
LETTERS	26 letters of the Roman alpahbet
WORDS	Most frequent 96 words
DIGRAMS	Most frequent 96 digrams
DOUBLETS	Most frequent 96 substrings found by progressive pairwise chunking
STRINGS	Most distinctive 96 substrings found by TEFF program

The main response variable measured was the percentage of correct classifications made **on unseen test data**. Mean values, averaged over 10 different text categorization problems, are shown in Table 7.

Table 7 -- Mean Success Rates on Test Data.

Source of Textual Marker	Mean Percentage Success Rate
LETTERS	69.03
WORDS	72.96
DIGRAMS	74.18
DOUBLETS	74.87
STRINGS	79.39

These results appear in increasing order of accuracy, averaged over the 10 test problems. It would seem that LETTERS are less effective than the middle group of WORDS, DIGRAMS and DOUBLETS, while STRINGS are more effective. To test this interpretation, a 2-way Analysis of Variance on these percentage scores was performed.

As expected, there was a very highly significant main effect of problem ($F_{9,36} = 42.13$, $p < 0.0005$). Clearly some problems are harder than others. More interestingly, there was also a highly significant main effect of marker type ($F_{4,36} = 5.40$, $p = 0.002$). In other words, even after allowing for differences between problems, the Null Hypothesis that all five marker types give equal success rates must be rejected. (This design does not permit testing for an interaction effect.)

To investigate the factor of marker type further, the effect of differential problem difficulty was removed by performing a 1-way Analysis of Variance not on the raw percentage success rates but on the deviations of each score from the mean for that problem, i.e. on the residuals. Once again this revealed a highly significant effect of marker type ($F_{4,45} = 6.75$, $p < 0.0005$). In addition, Dunnett's method of multiple comparisons with a standard was performed (Minitab, 1991). For this purpose, the success rate of DIGRAMS (the marker type giving the median mean score) was taken as a norm. Using a `family error-rate' of 0.05 (i.e. with a 5% significance level overall) gave an adjusted error rate of 0.0149. At this level, scores obtained by LETTERS were significantly different from those obtained by DIGRAMS (lower), as were scores obtained using STRINGS (higher). The other two marker types (WORDS and DOUBLETS) did not differ significantly from DIGRAMS in effectiveness.

Thus the appearance of a middle group consisting of WORDS, DIGRAMS and DOUBLETS than which LETTERS gave significantly worse results and STRINGS significantly better was confirmed. This is perhaps unsurprising -- at least with benefit of hindsight -- given the fact that LETTERS implies using fewer attributes than the rest (26 versus 96) and that STRINGS are preselected for distinctiveness whereas the other types of marker are selected only according to frequency. None the less, the fact that this preselection did not seem to give rise to overfitting was by no means a foregone conclusion.

Overall, the novel methods of feature-finding (progressive pairwise chunking and Monte-Carlo feature-finding) performed creditably in this empirical test. They warrant serious consideration by future workers in the area of text categorization.

11. Which Methods Work Best? -- A Benchmarking Study

Very little work has been done to compare the efficacy of different learning algorithms when used with textual data. However results of an empirical study by the present author suggest that methods that work well with what may be called `numeric' data sets are not necessarily the best for dealing with text-classification problems. Forsyth (1995) tested the following systems on a benchmark suite consisting of 13 text-discrimination problems. These systems are listed below in (approximate) order of increasing complexity.

Table 8 -- Classification Algorithms Tested.

System Name	Brief Description
BBTC	Basic Bayesian Text Classifier
1-NNC	Single Nearest-Neighbour Classifier
IOGA	Instance-Oriented Genetic Algorithm
MAWS	Mosteller-And-Wallace System (Robust Bayesian Text Categorizer)
TEXTREE	Textual Discrimination Tree Induction (& Pruning) Software
GLADRAGS	Genetic Learning Algorithm Discovering Relational Algebraic Grouping Signatures

Two of these implement well-established techniques -- 1-NNC and TEXTREE. The former is an instance-based classification technique widely used by statisticians and in machine learning (e.g. Aha et al., 1991; Dasarathy, 1991). TEXTREE is a program for deriving discrimination trees from data, based on the CART system of Breiman et al. (1984). Discrimination-tree induction is one of the most popular inductive techniques in existence (Quinlan, 1993). The other techniques are novel -- except that MAWS constitutes an automation and extension of a method (a `robust Bayesian analysis') applied by Mosteller & Wallace (1984).

Michie et al. (1994) have established that nearest-neighbour classification and discrimination-tree induction are by no means `straw men'. So the results in Table 9, which lists these methods in order of mean percentage accuracy achieved over the 13 test problems **on unseen data**, are somewhat surprising.

Table 9 -- `Pecking Order' of Text Classification Methods Tested.

Technique	Success Rate (%)
BBTC with digrams	77.31
MAWS	74.10
GLADRAGS	70.61
BBTC with digraphs	68.83

IOGA	68.37
1-NNC	66.83
TEXTREE	66.76
`Default'	47.45

[Here 'Default' is the percentage success rate given by assigning each case in the test sample to whichever category is most frequent in the training data.]

This table summarizes a long and multi-faceted study and should not therefore be over-emphasized. Nevertheless, the fact that the simplest method (which was only really intended to establish a baseline performance level) performed best, taken together with the fact that the best-established machine-learning method performed worst, may be significant.

It is worth asking why both Bayesian methods (BBTC and MAWS) seem to work better on this sort of data than tree-based induction. It must be presumed that the weaknesses of probabilistic Bayesian classification are less brutally exposed by these tasks than the weaknesses of tree-structured classification. Whereas Bayesian methods, of the type used here, are vulnerable to interdependencies among features, tree-structured classifiers are particularly vulnerable to noise or random variability. It is thus reasonable to conclude that interaction effects among the textual attributes used here are relatively minor, whilst the problem of random variation is quite severe.

However, although the accuracy of the tree-based classifier was disappointing, such systems do have an advantage in terms of descriptive adequacy. This is illustrated by an example tree, which distinguishes articles in the journal *Literary & Linguistic Computing* from those in *Machine Learning*, shown in Table 10 after post-pruning with a method described by Forsyth et al. (1994). This tree uses only five features and has only seven leaf nodes: some of the deeper nodes of the full tree (which contained 12 leaf nodes) have been amalgamated.

Table 10 -- Pruned Tree from MAGS Data.

```
`earn`
  `ngu`
    `lin`
       8   8  0
    ~`lin`
       7   2  5
  ~`ngu`
     107   1  106
~`earn`
  `ngu`
     64   63  1
  ~`ngu`
    `text`
       21   21  0
    ~`text`
      `prob`
```

```
    12   2  10
~`prob`
    25   19  6
```

In this tree the four largest terminal subsets (containing 107, 64, 21 and 25 instances) cover over 88% of the training sample. Expressing them as conjunctive clauses gives a good summary of what the program has found:

`earn' AND NOT `ngu'
==> 106 to 1 in favour of class 2;
NOT `earn' AND `ngu'
==> 63 to 1 in favour of class 1;
NOT `earn' AND NOT `ngu' AND `text'
==> 21 to 0 in favour of class 1;
NOT `earn' AND NOT `ngu' AND NOT `text' AND NOT `prob'
==> 19 to 6 in favour of class 1.

In terms of badges, it is not hard to deduce from this that `ngu' and `text' are badges of class 1, *Literary & Linguistic Computing*, while `earn' and `prob' (found in `problem' as well as `probable' and its derivatives) are badges of class 2, *Machine Learning*.

Thus the tree formalism can give a compact and comprehensible desciption of what a learning system has learned. The trouble is that TEXTREE proved less accurate than systems using less intelligible representation schemes. Further work is undoubtedly needed on reconciling accuracy with descriptive adequacy.

12. Discussion

This article has reviewed a diverse selection of modern approaches to text classification. It is clear that the conflux of machine learning with stylometry has led to a ferment of new ideas. What is less clear is which of these will prove most fruitful in the long term. The work reported in section 10 and 11 is an attempt to cast some light on this issue.

As far as feature-finding is concerned, the results provide evidence that Monte-Carlo Feature-Finding is a promising method of discovering textual descriptors. As substrings found by CHISUBS or TEFF can include a wider range of textual patterns than just words or word-pairs, there is some reason for recommending them to future workers in this area.

However the possibility of using syntactic information to classify texts, as done by Baayen et al. (1996), or semantic information, as done by Martindale & McKenzie (1995), should not be discounted. The best way of combining superficial lexical measures, such as word or substring frequencies, with syntactic and/or semantic markers, remains an open research question.

12.1 In Praise of Semi-Crude Bayesianism

The results described in section 11 also provide support for the use of a Bayesian inferential framework. Both BBTC and MAWS performed creditably. BBTC uses Bayes's Rule (Bayes, 1970 [1763]) in a very simple-minded manner, calculating posterior probabilities on the basis of digram or trigram frequencies, assuming independence between features. MAWS is rather more sophisticated, though not quite so successful as BBTC on the benchmark problems. It did, however, out-perform

more conventional classification algorithms. The fact that MAWS is an automated, albeit slightly extended, version of a technique introduced by Mosteller & Wallace in 1964 suggests that machine-learning researchers may be able to learn from stylometers, as well as *vice versa*.

12.2 What's So Special About Linguistic Data?

A problem brought to light by the experiment summarized in Table 9 is that the more accurate trainable classifiers tend also to be the most inscrutable, while the less accurate ones are those using the most transparent representation schemes. Associated with this is a positive correlation between number of features used by a classifier and its success rate. Another way of putting this is that textual classification problems seem to be relatively `feature-hungry'. Thus the tension between simplicity and accuracy is more acute in this field than in many other domains where machine learning methods have been tried.

A question that arises from this finding is whether incompatibility between simplicity and accuracy is inevitable. Goldberg (1995) has argued that textual variables 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, certainly not for short passages[2]. This is an interesting conjecture, which is worth attempting to falsify. A program that could reduce the number of textual features needed for successful text classification from several hundred (as in BBTC) 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 theoretical implications, since it would tend to corroborate Goldberg's thesis.

References

Aha, D.W., Kibler, D. & Albert, M.K. (1991). Instance-Based Learning Algorithms. *Machine Learning*, 6, 37-66.

Ahonen, J., Mannila, H. & Nikunen, E. (1993). Forming Grammars for Structured Documents. In: *Proc AAAI-93 Workshop on Knowledge Discovery in Databases*. AAAI Press, Menlo Park, CA.

Aleksander, I. & Morton, H. (1990). *An Introduction to Neural Computing*. Chapman & Hall, London.

Allinson, N.M. (1994). Personal Communication. [from: Image Engineering Laboratory, University of York.]

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, CA.

Baayen, H., Tweedie, F. & Van Halteren, H. (1996). Outside the Cave of Shadows: Using Syntactic Annotation to Enhance Authorship Attribution. Literary & Linguistic Computing, 11(3), 121-131.

[2] The trials described in sections 10 and 11 were conducted on texts divided into blocks of between 100 and 200 words, shorter than in most previous stylometric studies.

Bayes, T. (1763). An Essay towards Solving a Problem in the Doctrine of Chances. *Phil. Trans. Royal Society*, 53, 370-418. [In: E.S. Pearson & M.G. Kendall (1970) eds. *Studies in the History of Statistics and Probability*. Griffin, London.]

Beale, R. & Jackson, T. (1990). *Neural Computing: an Introduction*. Adam Hilger, Bristol.

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.

Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. (1984). *Classification and Regression Trees*. Wadsworth, Monterey, CA.

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.

Dasarathy, B.V. (1991) ed. *Nearest Neighbour (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, California.

Davis, P.D. (1986). *How Poetry Works*. Pelican Books, Middlesex.

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.

Dixon, P. & Mannion, D. (1993). Goldsmith's Periodical Essays: a Statistical Analysis of Eleven Doubtful Cases. *Literary & Linguistic Computing*, 8(1), 1-19.

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

Fang, A.C. & Nelson, G. (1994). Tagging the Survey Corpus: a LOB to ICE Experiment using AUTASYS. *Literary & Linguistic Computing*, 9(3), 189-194.

Forsyth, R.S. (1981). BEAGLE -- a Darwinian Approach to Pattern Recognition. *Kybernetes*, 10, 159-166.

Forsyth, R.S. (1989) ed. *Machine Learning: Principles and Techniques*. Chapman & Hall, London.

Forsyth, R.S. (1990). Neural Learning Algorithms: Some Empirical Trials. *Proc. 3rd International Conf. on Neural Networks & their Applications, Neuro-Nimes-90*. EC2, Nanterre.

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

Forsyth, R.S., Clarke, D.D. & Wright, R.L. (1994). Overfitting Revisited: an Information Theoretic Approach to Simplifying Discrimination Trees. *J. Experimental & Theoretical Artificial Intelligence*, 6, 289-302.

Forsyth, R.S. & Holmes, D.I. (1996). Feature Finding for Text Classification. *Literary & Linguistic Computing*, 11(4), 163-174.

Forsyth, R.S. & Rada, R. (1986). *Machine Learning: Applications in Expert Systems and Information Retrieval*. Ellis Horwood, Chichester.

Fries, C.C. (1952). *The Structure of English*. Harcout-Brace, New York.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, Mass.

Goldberg, J.L. (1995). CDM: an Approach to Learning in Text Categorization. *International J. of Artificial Intelligence Tools*, in press (March 1996?).

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

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

Hayes, P. & Weinstein, S. (1991). Adding Value to Financial News by Computer. In: *Proc. First Internat. Conf. on Artificial Intelligence on Wall Street*, 2-8.

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.

Holmes, D.I. & Forsyth, R.S. (1995). The `Federalist' Revisited: New Directions in Authorship Attribution. *Literary & Linguistic Computing*, 10(2), 111-127.

Honoré, A. (1979). Some Simple Measures of Richness of Vocabulary. *ALLC Bulletin*, 7(2), 172-177.

Horton, T.B. (1987). *The Effectiveness of the Stylometry of Function words in Discriminating between Shakespeare and Fletcher*. Doctoral Thesis, University of Edinburgh.

Indurkhya, N. & Weiss, S.M. (1991). Iterative Rule Induction Methods. *J. Applied Intelligence*, 1, 43-54.

Jacobs, P.S. (1993). Uisng Statistical Methods to Improve Knowledge-Based News Categorization. *IEEE Expert*, April 1993, 13-23.

Keulen, F. (1986). The Dutch Computer Corpus Pilot Project. In: J. Aarts & W. Meijs, eds., *Corpus Linguistics II*. Rodopi, Amsterdam.

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

Kjetsaa, G. (1979). "And Quiet Flows the Don" through the Computer. *ALLC Bulletin*, 7, 248-256.

Koza, J. (1992). *Genetic Programming*. MIT Press, Cambridge, Mass.

Larsen, W., Rencher, A. & Layton, T. (1980). Who Wrote the Book of Mormon?: an Analysis of Wordprints. *Brigham Young University Studies*, 20, 225-251.

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.

Leech, G.N. (1987). General Introduction. In: R. Garside, G. Leech & G. Sampson, eds., *The Computational Analysis of English*. Longman, Harlow, Essex.

Lehnert, W., Soderland, S., Aronow, D., Feng, F. & Shmueli, A. (1995). Inductive Text Classification for Medical Applications. *J. Experimental & Theoretical Artificial Intelligence*, 7(1), 49-80.

Lowe, D. & Matthews, R. (1995). *Shakespeare vs. Fletcher: a Stylometric Analysis by Radial Basis Functions*. Computers & the Humanities, 29, 449-461.

Martindale, C. & McKenzie, D.P. (1995). On the Utility of Content Analysis in Authorship Attribution: the Federalist. *Computers & the Humanities*, 29, 259-270.

Masand, B., Linoff, G. & Waltz, D. (1992). Classifying News Stories using Memory-Based Reasoning. In: *Proc. 15th Annual Internat. ACM SIGIR Conf. on R & D in Information Retrieval*, June 1992, 59-65.

Matthews, R.A.J. & Merriam, T.V.N. (1993). Neural Computation in Stylometry I: an Application to the Works of Shakespeare and Fletcher. *Literary & Linguistic Computing*, 8(4), 203-209.

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

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

Merriam, T.V.N. (1992). *Modelling a Canon: Principles and Examples in Applied Statistics*. Doctoral Thesis, Kings College, University of London.

Merriam, T.V.N. & Matthews, R.A.J. (1994). Neural Computation in Stylometry II: an Application to the Works of Shakespeare and Marlowe. *Literary & Linguistic Computing*, 9(1), 1-6.

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

Milic, L.T. (1967). *A Quantitative Approach to the Style of Jonathan Swift*. Mouton & Co., The Hague.

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

Morton, A.Q. (1978). *Literary Detection: How to Prove Authorship and Fraud in Literature and Documents*. Bowker Publishing Co.

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.]

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

Piatetsky-Shapiro, G. & Frawley, W.J. (1991) eds. *Knowledge Discovery in Databases*. MIT Press, Cambridge, Mass.

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.

Sagan, C. (1981). *Cosmos*. Macdonald Futura, London.

Samuel, A.L. (1967). Some Studies in Machine Learning using the Game of Checkers, Part II. *IBM J. Research & Development*, 11.

Sejnowski, T.J. & Rosenberg, C.R. (1987). Parallel Networks that Learn to Pronounce English Text. *Complex Systems*, 1, 145-168.

Sichel, H.S. (1986). Word frequency distributions and type-token characteristics. Mathematical Scientist, 11, 45-72.

Smith, P.D. (1990). *An Introduction to Text Processing*. MIT Press, Cambridge, Mass.

Specht, D.F. (1990). Probabilistic neural networks. *Neural Networks*, 3, 109-118.

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.

Teskey, F.N. (1982). *Principles of Text Processing*. Ellis Horwood, Chichester.

Thirkell, L.A. (1992). A Connectionist Approach to Authorship Determination. In: *Proc. 19th International Conf. of ALLC*, Christ Church College, Oxford.

Tweedie, F., Singh, S. & Holmes, D.I. (1994). Neural Network Applications in Stylometry: the 'Federalist' Papers. *Computers and the Humanities*, 30, 1-10.

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

Von Arnim, H. (1896). *De Platonis Dialogis. Quaestiones Chronologicae*. Rostock.

Voutilainen, A., Heikkila, J. & Anttila, A. (1992). *Constraint Grammar of English: a Performance-Oriented Introduction*. Publication No. 21, Dept. General Linguistics, University of Helsinki, Finland.

Wasserman, P.D. (1993). Advanced Methods in Neural Computing. Van Nostrand Reinhold, New York.

Weiss, S.M. & Kulikowski, C.A. (1991). *Computer Systems that Learn*. Morgan Kaufmann, San Mateo, CA.

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

Wilson, I. (1993). *Shakespeare: the Evidence*. Headline Book Publishing, London.

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

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