Segmenting a document by stylistic character

Neil Graham* and Graeme Hirst†
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 3G4

February 24, 2003

Abstract

As part of a larger project to develop an aid for writers that would help to eliminate stylistic inconsistencies within a document, we experimented with neural networks to find the points in a text at which its stylistic character changes. Our best results, well above baseline, were achieved with time-delay networks that used features related to the author's syntactic preferences. Low-level and vocabulary-based features were not found to be useful.

1 Introduction

There are many different ways that authors can collaborate in the writing of a text [Posner and Baecker, 1992]. Variables include whether the granularity of the individual authors' unit of contribution is sentences, paragraphs, or sections; whether or not one of the authors acts as 'editor', revising the writing of all the others; and, if not, whether each author revises only their own work or also that of some or all of the others. Thus a collaboratively written text might be nothing more than a concatenation of segments by each of the participating authors, or it could be essentially the work of a single person 'heavily influenced' by the other participants, or it could be something between these extremes.[1] Regardless of the method by which the authors produce the text, if the result is a sequence of stylistically disparate segments, then it is deprecated and quite possibly hard to read — though the reader need not be conscious of the reason for the difficulty, merely finding the text to be 'difficult' or 'badly written'. (Baljko and Hirst [1999] showed that although stylistic judgments are subjective, readers who were asked to classify writing samples by style generally agreed with one another in their judgments of stylistic similarity and difference.)

By contrast, an ideal collaboratively written text is one in which the authors have 'harmonized' with one another to the point that the text is so stylistically homogeneous that it is no longer possible to find stylistically distinct segments; the authors 'speak with one voice'. Thus a software aid for collaborative writers that would help them achieve this unity of style would not be a 'style checker' in the popular sense, looking for 'good' or 'bad' style (though it could work in conjunction with such a tool); rather, it would look for stylistic inconsistencies in a document and suggest to the authors how they could be ameliorated. Nor need such a

*Now at IBM Canada Ltd.
†Corresponding author; *gh@cs.toronto.edu*

[1] This paper is an example of its own subject matter. Graham and Hirst have very different writing styles. Graham wrote text describing his portion of the research; Hirst then wrote the remaining material and incorporated Graham's text into it with extensive editing to try to harmonize their styles. The paper has *not* been used as data in the research that it reports.

tool be limited to collaborative writing; a single writer might inadvertently fail to ‘harmonize with herself’, especially in a document that is written in several bursts over an extended period of time.

Within this idea lie several research problems, not least of which is the difficulty of presenting linguistically naive users with advice on stylistic nuances and subtleties. However, in this paper we concentrate on the initial problem of detecting the inconsistencies — searching the text for any paragraphs or regions that are stylistically distinct from others. More generally, we can think of this as segmentation of the text by stylistic character. Even by itself, this could be a helpful diagnostic tool for collaborating writers, showing them where they had failed to unify their styles. And the segmentation of text by style (and thus, implicitly, by authorship) has other uses as well in literary studies and in forensic applications such as the detection of plagiarized sections in students’ essays.

2 Stylistic segmentation is not authorship identification

Our task, then, is to take a text that might be a sequence of stylistically distinct segments and identify the boundaries between the segments (if any) — the points where the style changes.

Clearly, our problem is related to that of authorship identification, but it is not at all the same problem. Certainly, if we could stylistically identify one segment of a text as John’s and another as Mary’s, then we would have ipso facto detected a stylistic difference or inconsistency; and conversely, if we could say that the style of an entire text, or one segment of it, is ambiguous or half-way between that of John and that of Mary, then we could say that the authors have harmonized their style. But that’s not really what we are trying to do. All we care about is whether or not the text is internally consistent, and it doesn’t matter whether that is achieved by revising John’s text to match Mary’s style or by revising both so that they ‘meet in the middle’. (Nor will we necessarily have a single-author corpus for each of the collaborating writers.)

Nonetheless, the two problems are similar insofar as they involve comparing texts for stylistic similarity or difference. In the case of authorship identification, this involves comparing the target text with an attested corpus of a candidate author; in our case, it involves comparing one region of a text with another. Research in authorship identification has concentrated on determining the textual attributes that are most likely to reflect an author’s individual style and on which statistical methods are best employed in the task. (For a review of research in authorship identification, see [Holmes, 1994].) The main difference between the two cases lies in the size of the texts being compared. In authorship identification, it is assumed that both the target text and the attested corpus are reasonably large. Because most stylistic attributes are based on the frequency of occurrence of some linguistic or textual feature, the larger the target text and the attested corpus are, the more reliable the result will be; if either is too small, statistical significance cannot be achieved. In our case, however, we could be comparing paragraphs or segments as small as 100 words or less. (Attribution of authorship when the text or the attested corpora are small, which has particular utility in forensic applications, has generally involved non-automated qualitative judgments.)

We have therefore drawn on ideas from research on authorship identification, especially research on determining the textual attributes that are most likely to reflect a distinguishable style. However, we have adapted this work to our own particular needs.

3 Related work

In one of the few studies into the application of stylistic statistics to small samples, Glover and Hirst [1996] asked a number of subjects to watch an episode of a television series and write a description of it. They then randomly combined one subject’s description of the first half of the episode with another’s description of the second, thus artificial collaboratively written texts. Glover and Hirst found that stylometric statistics could distinguish the pairs written by different subjects from pairs written by a single subject with reasonable probability, even though the texts involved were almost all less than 500 words in length. Juola [1997] demonstrated a technique that, using samples as small as 500 characters, can correctly classify all the disputed *Federalist Papers*. By combining particular sentence- and chunk-boundary detectors with a particular multi-pass parser, Stamatatos et al [1999; 2000] developed a complex but effective method for ascribing the authorship of fairly small samples of modern Greek text. Despite these positive indications, there has been no study with a large corpus of small texts using a broad range of statistics; most researchers therefore remained skeptical that stylistic statistics could provide much information for very small samples.

Most of these studies used fairly conventional statistical techniques; however, the use of neural nets in conjunction with stylometric statistics is becoming increasingly popular. The pioneering work in this regard was undertaken by Matthews and Merriam [1997]. They used a very small set of function-word frequencies as input to a multilayer perceptron to examine sections of four plays that have been attributed both to Shakespeare and to Fletcher, producing results that accord reasonably well with accepted scholarship. Tweedie et al [1996a] present an extensive review of the uses of neural nets in stylometry. In their own work [Tweedie et al, 1996b] in this connection, involving the *Federalist Papers*, they trained a single network with a small hidden layer on a subset of the function words originally used by Mosteller and Wallace [1964] and reproduced Mosteller and Wallace's results. Neural nets have also found popularity in genre detection (*e.g.*, Kessler et al [1997]), where stylometric statistics have been used as indicators of genre for some time.

4 Models of 'bad' collaboratively written text

For our experiments, we need a corpus of examples of 'bad' collaborative writing in which the stylistic disparities are marked. Clearly, it is infeasible, or even impossible, to gather large amounts of naturally occurring data of this kind; so, following the idea underlying the work of Glover and Hirst (see section 3 above) we instead created synthetic data that models it by concatenating a corpus of Usenet postings and removing all textual indications of boundaries (such as headers and signatures). Our assumption was that in the resulting text, the concatenation points would be just those points at which the style of the text changes. Of course, this is an idealization: it is possible that two consecutive postings could be stylistically indistinguishable, even if by different authors; and a single posting could be stylistically disparate, for a variety of reasons including embedded quotations (see below) and multiple authors (although this is very rare in the data that we used).

The Usenet postings that we used were the articles in all issues of *Risks Digest*[2] from 5 April 1996 to 1 April 1999. This corpus serves our purposes well because it is comparatively large — slightly over 750,000 words; it is generally well-written — that is, it is generally grammatically well-formed and free of flames, advertisements, and spam — and it is fairly well-controlled for genre and topic. Thus it reflects the level of writing that is typical of business and technical documents.

We discuss elsewhere [Graham, 2000] the manner in which we processed this corpus to make it suitable for our experiments; here we just briefly touch on how we addressed some problems that the corpus presented. While it was easy to remove and record structural indications of article boundaries (standard e-mail headers), to make the data comply with our assumption that article boundaries correspond to auctorial changes, we had to find a reasonably certain method of weeding out embedded quotations from articles. This turned out to be a surprisingly challenging problem, necessitating the development of several heuristics; in the end we were able to correctly identify 95% of quotations, with only about a 20% rate of false positives. To conserve material, and to see how our experiments would fare on non-articles, we 'recycled' the larger quotations by replacing them in the corpus outside of any article, thus treating them as complete articles. Since e-mail

[2] *Forum on Risks to the Public in Computers and Related Systems* (comp.risks), ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator.

signatures could potentially bias the results of our experiments — particularly with regard to statistics such as punctuation frequencies — we also attempted to remove them wherever possible. Despite much effort, here we managed to remove 89% of signatures with about 18% false positives. The signatures that we were not able to remove were either highly idiosyncratic or were obviously difficult to distinguish from text.

We then determined sentence boundaries and tagged the corpus with parts of speech. Following van Halteren et al [1998], we used three different part-of-speech taggers [Ratnaparkhi, 1996; Schmid, 1995; Brill, 1995] in order to improve the accuracy of our tagging.

5 Experimental method

Our approach uses neural networks to decide whether or not two segments of text are stylistically distinct. The main strengths of a neural network-based approach are that neural nets are often more tolerant of noise in the data and more apt to generalize than conventional statistical classification methods. These two properties are important to us, as our methodology is fundamentally statistical and not based on heuristics, and our data is quite noisy. The existence of a large body of excellent neural network simulation software was also a great benefit: we used the SNNS (Stuttgart Neural Network Simulator) system from the Universities of Stuttgart and Tübingen [SNNS, 1990–98].

Our use of neural nets differs from their use in other studies related to authorship attribution. First, we train our neural nets not on corpora produced by a small number of known authors, but on a corpus written by a large number of unknown authors. Second, we compute a broad range of stylometric statistics on a large number of very small samples of work, including many thought not to be reliable on such samples. Third, our neural net architectures need to be different from those used by previous workers in this area, as we require our nets to tell us whether or not two samples are different, rather than to fit a single sample into one of a set of pre-defined classes. Finally, we use some stylometric statistics, such as distribution of punctuation, that have been widely ignored in the literature because researchers have felt they are prone to contamination [Holmes, 1994].

Thus we cast the problem of text segmentation by stylistic character as a binary classification problem for neural networks. The network is presented with statistical data (as described in section 6 below) from two segments of the text, where a segment, for us, is always exactly one paragraph. The network's job is to classify the pair of segments as either stylistically distinct or not: 1 or 0 respectively. In production use, the segments presented would be consecutive paragraphs of a text; classifying them as stylistically distinct would imply that there is a change of style, possibly due to an authorship boundary, between them.

6 Features used

There was little a priori indication as to which statistics would be best suited to small sample sizes, so we chose several categories of statistics that have been used previously in authorship attribution, stylistic analysis, genre detection, and information retrieval, in order to compare the efficacy of each. Here, we will briefly define each of these statistics and discuss our motivation for including it.

6.1 Surface features

For each sample we computed the average word-length and the frequency of each word-length, in terms of both characters and syllables. (Holmes [1994] reports that word character-length distributions, far from characterizing the work of any particular author, depend more on the genre in which an author is writing, but we included them anyway because of the simplicity of the computation.) It is true that syllable-length and

Table 1: Function words and punctuation marks for which we computed per-sample frequencies.

the	of	and	to	a	in	that	he	for	it
with	as	his	on	at	by	i	this	not	but
from	or	an	they	which	you	one	her	all	she
there	their	we	him	when	who	more	no	if	out
.	?	!	:	;	,	—	(	)	[
]	{	}	<	>	"	“	”	‘	/

character-length correlate very strongly, but we felt that the cognitive processes alleged to underlie the usefulness of each category were sufficiently distinct to justify our studying both. The syllable-count for each word was taken from the MRC2 electronic dictionary [Wilson, 1987]. (Character counts greater than 15 were treated as 15; syllable counts greater than six were treated as six.) We also computed the frequencies in the text of each part-of-speech category (as defined in the Penn Treebank [Marcus et al, 1993]), a measure that has been used in many studies. And we computed sentence length in terms of words, although most scholars believe this information is too genre- and topic-specific to characterize authorship. While there does seem to be a consensus that sentence-length distributions provide more information than average sentence length alone, the small size of our samples makes it extremely unlikely that sentences of any one length will occur more than once in any sample.

The use of function-word frequencies also has a long history in stylistic research, including the well-known work of Mosteller and Wallace [1964]. To decide which words to use, we simply examined all words in the Brown corpus [Kučera and Francis, 1967] with frequencies above 0.2%, and manually removed from this list all words commonly used either as content words (*i.e.*, verbs or nouns, excluding pronouns). The 40 words that emerged from this process are listed in table 1. In addition, we decided to record the frequency distribution of common punctuation marks (table 1). Some researchers have been very reticent to consider punctuation in their studies, since this is often beyond the control of the text's original author; however, since our corpus is not copyedited, we decided to include it.

6.2 Entropy features

Some investigators of style have attempted to measure the amount of structure in an author's writing by using lexical entropy. The lexical entropy H of a passage of text is defined as follows:

$$H \stackrel{\text{def}}{=} -\sum_i \left(\frac{iV_i}{N}\right)\log\left(\frac{iV_i}{N}\right).$$

where N is the total number of tokens in the sample, V is the number of types, and V_i is the number of types that appear exactly i times. Although we calculated lexical entropy for each of our samples, a seemingly more promising measure, at the character level rather than the lexical level, is given by Juola [1997]. For a sample of text C characters long, Juola selects a 'window-size' parameter c such that $0 < c < C$. A 'window' contains c consecutive characters. Suppose such a window begins at index i (*i.e.*, at the i-th character), $1 \le i \le C - c + 1$. Juola then calculates the length of the longest sequence of characters beginning after the window (*i.e.*, at index $c + i + 1$) that is identical to some sequence of characters within the window. Letting this quantity equal L_i, Juola's statistic is then defined as:

$$\hat{L} \stackrel{\text{def}}{=} \frac{\sum_{i=1}^{C-c} L_i}{C - c}.$$

The method can be shown to converge to the information-theoretic entropy as $c \to \infty$. Juola's work on authorship attribution used windows of as little as 500 characters, and so is of clear relevance here. However, our very short samples required even smaller windows: we used 250 characters if the sample was more than 1000 characters long and a quarter of its length if it was between 200 and 1000 characters. For samples of fewer than 200 characters, we simply used $L = 1.75$ (which is Brown et al's [1992] estimate of the entropy of English), and so extremely short samples are indistinguishable with respect to this feature.

6.3 Vocabulary features

Many statistics that have appeared in the literature attempt to measure the richness of an author's vocabulary. We computed the standard type/token ratio, as well as Simpson's index, Yule's characteristic, and Honoré's measure; all these are defined and discussed extensively by Holmes [1994]. In other work [Graham, 2000], we present methods to normalize the output of the latter two statistics so that they do not bias neural networks. We also present a complex method for normalizing the type/token ratio so that it is less sensitive to sample size. For samples too small even for this method we used the average of the type/token ratios calculated across our entire corpus.

As some researchers have postulated that an author's style is reflected in hapax legomena and hapax dislegomena, we computed both measures for all of our samples, *i.e.*, V_1 and V_2. Most researchers seem to believe that even in large samples, these measures are untrustworthy; thus much work has been spent in attempting to develop statistical models — relations between i and V_i — to characterize authors' vocabularies. According to Holmes [1994], the best results have been achieved by the Waring–Herdan distribution. So, following Holmes, we used a corrected version of this distribution in our study, and computed only the first five terms of the distribution for each sample.

6.4 Summary

Table 2 lists a summary of the statistics that we computed for each of our samples. The table also shows how the statistics were grouped into ten categories for our experiments below.

7 Experiments and results

Except in a few instances, all of our networks were randomly initialized with weights between -1 and 1. Largely to conserve our data and time, we used the simplest possible strategy for testing our networks; we trained them on 90% of our data and tested them on the other 10%. We did not perform any cross-validation, but we are confident that the results that we have obtained are not aberrations because, though our best networks exhibited the usual pattern of overfitting to the training data upon intensive training, many of our networks actually found the test data easier than the training data, even after many thousands of learning steps. Thus, we are confident that no trivial patterns existed in our training or test sets that might invalidate our results.

Because all of our networks use logistic sigmoid activation functions (as most of our data points have magnitudes between 0 and 1), their outputs are always in the range 0 to 1. Thus, we have used the mean squared error (MSE) of our networks, taken over the entire training or test set, to obtain a general idea of the network's performance. For our best-performing network, described below, we also computed performance in terms of recall, precision, and accuracy.

Table 2: The statistics that we computed and their grouping into ten categories for our experiments.

1. Average word length, frequency of i-letter words $1 \le i \le 15$ (words with length > 15 counted as 15-letter words)
2. Average syllables/word, frequency of i-syllable words $1 \le i \le 6$ (words with > 6 syllables counted as 6-syllable words)
3. Average words/sentence
4. Relative frequencies of various parts of speech
5. Relative frequencies of function words (see table 1)
6. Relative frequencies of punctuation (see table 1)
7. Lexical entropy H, Juola's measure L
8. Normalized type/token ratio R, Simpson's index D, modified Yule's characteristic $(10^{-4} \times K)$, modified Honoré's measure $(\frac{\hat{R}}{100})$
9. Ratio of hapax legomena $(\frac{V_1}{V})$ and hapax dislegomena $(\frac{V_2}{V})$
10. First five terms of corrected Waring–Herdan distribution

7.1 Experiments with multilayer perceptrons

Table 3 shows the best results that we obtained with standard multilayer perceptron networks using various categories of our data as inputs. These results are selected from tests run with many different learning algorithms and many different hidden-layer sizes. The networks that are marked with asterisks in the table were generated as follows. The best networks that we had developed for the component categories were used as the bottom layer of the network, and a 'gating' network (a network with as many inputs as categories of statistics, containing a hidden layer of size noted in the table) was then used to join these component networks. In the network marked with two asterisks, all the weights in the resultant large network were then allowed to change during learning; in the networks with only one asterisk, only those in the gating network were allowed to change. While this model is based on mixtures of experts as described by, for instance, Bishop [1995], it is noteworthy that the performance of our network with two asterisks is better than that of the corresponding network with one asterisk.

The data presented in table 3 suggest many interesting conclusions. Most significantly, it is categories 4, 5, and 6 of statistics — those that characterize an author's syntactic preferences (part-of-speech and punctuation frequencies as well as function-word frequencies) — that are best at identifying authorship boundaries, both individually and together. Features at a lower level (categories 1 to 3: syllable- or character-length distributions and sentence lengths) do not do nearly as well, nor do statistics intended to measure vocabulary richness or structure (categories 7 to 10: entropy statistics, Waring–Herdan frequencies, and so on).

It is also clear that combining too many unrelated statistical categories in the same network yields poor results. If the network is organized into a mixture of experts, the results are somewhat better, but do not

Table 3: The lowest test MSE's and corresponding training MSE's that were obtained with multilayer perceptrons trained on various categories of statistics (as defined in table 2). Boldface indicates the best result. *Backprop WD* = backpropagation with weight decay; *Backprop M* = backpropagation with momentum. Asterisks denote networks built along the lines of mixtures of experts; two asterisks imply all weights were changed during learning, one implies only the weights of the gating network were changed.

Categories used	Size of hidden layer	Size of input layer	Learning algorithm	Best MSE during training	Best MSE during test
All	25	266	Backprop WD	0.1814	0.1953
1	12	32	Backprop M	0.1836	0.1770
2	8	16	Backprop M	0.1831	0.1763
3	4	2	Backprop M	0.1892	0.1814
4	30	72	Backprop M	0.1612	0.1638
5	25	80	Backprop M	0.1720	0.1725
6	16	40	Backprop M	0.1659	0.1664
7	4	8	Backprop M	0.1883	0.1779
8	20	8	Simulated Annealing	0.1929	0.1803
9	7	4	Backprop M	0.1878	0.1781
10	16	10	Simulated Annealing	0.1912	0.1798
∗ All	40	266	Backprop M	0.1861	0.1781
∗ 4,5,6	50	192	Backprop M	0.1715	0.1680
∗∗ 4,5,6	**50**	**192**	**Backprop M**	**0.1380**	**0.1584**

reach the levels of the better component networks unless the categories are related. The fact that each of the three categories comprising the final network in table 3 were individually fairly good and measured a related aspect of writing goes some way to explaining the surprising result that this network was so much better than the others. It is also worth noting that, though we tried various learning algorithms, sometimes one succeeding better than another, the organization of the network and the statistical category being tested had far more effect on the test MSE than any choice of algorithm. In the next section we expand on this point by demonstrating how an entirely different architecture further improved our results.

7.2 Results with time-delay neural networks

Time-delay neural networks have traditionally been used in situations where data are generated over time, and where each datum in a set is produced by the same process in all time intervals. By grouping corresponding data items from consecutive sets into 'features', with each item in a feature joined by coupled weights to a unit of the input layer, this homogeneity and dependence is incorporated into the architecture of the network. Other than having coupled weights, these networks may have arbitrary topologies, and may even have coupled units in their hidden layers that act in much the same way as in the input layer.

Intuitively, it is easy to cast our experiment to fit this model; the later paragraphs of a contribution must depend on earlier paragraphs. Since our data sets are consistently ordered, we automatically have that each datum from one set measures the same underlying feature as the corresponding datum from any preceding set.

Given that time-delay networks take longer to train than multilayer perceptrons of similar size, we con-

Table 4: Some of the results obtained with various time-delay neural networks. Boldface indicates the best result.

Categories used	Number of hidden units	Training MSE	Test MSE
4,5,6	48	0.1412	0.1497
4,5,6	32	0.1420	0.1500
4,5,6	**20**	**0.1380**	**0.1495**
4,5,6	12	0.1427	0.1501
4,5,6	8	0.1394	0.1499
4,5,6	4	0.1415	0.1501
4,5,6	2	0.1473	0.1504
4,5,6	0	0.1600	0.1619
All	30	0.1874	0.1804
8	4	0.1888	0.1806
2	7	0.1832	0.1739
4	8	0.1544	0.1592

structed only networks that used two consecutive data sets. Further, at most one layer of hidden units was used, and no coupled weights were used there.

Table 4 gives some of the results we obtained with time-delay networks. The most obvious feature of this table is how well the combination of categories 4, 5, and 6 again appears to work with this network architecture. These error levels are more than 5% better than we previously observed, and, as we discuss below, lead to recalls and precisions far above chance levels. Selected single categories that we tested also led to better results when time-delay architectures were employed, whereas all the categories together produced results somewhat inferior to the mixture-of-experts model used earlier. So, while time-delay networks are usually more effective in contexts such as this, a mixture-of-experts model may sometimes be able to exploit patterns in various individual categories that are obscured when only one hidden layer is available.

7.3 Evaluation of results

We now evaluate these results by comparing them to a baseline. Three baselines suggested themselves. The first, and simplest, is always to guess that there is no authorship boundary; this is the most probable case in our test data, as the ratio of authorship boundaries to non-boundaries is nearly 1 to 3. This method results in a mean squared error of 0.2552 for our entire corpus, 0.2359 on the test set alone. A second baseline is an algorithm that randomly guesses 1 and 0 with appropriate probabilities. This algorithm generally produces MSE's of 0.316 on the entire corpus. A third baseline is to use the ratio of breaks to non-breaks to guess a single value between 0 and 1 that minimizes the MSE. This produces an MSE of 0.1901 for the entire corpus and 0.1801 for the test corpus. The fact that many of our poorer networks achieved minimum MSE's poorer than these values shows that such results are statistically insignificant.

We now consider the results of our best network in terms of recall, precision, and accuracy. Recall is the fraction of pairs representing authorship boundaries that were correctly identified as such; precision is the fraction of pairs identified as authorship boundaries for which the decision was correct; accuracy is the fraction of all decisions that were correct. However, these measures cannot be meaningfully computed for the first and third baselines. For the first baseline (always guess that there is no authorship boundary), precision and recall are zero, and accuracy is simply equal to the fraction of non-boundaries in the data (which is

Table 5: Precision, recall, and accuracy of our best time-delay network on the test suite. The threshold is the value below which the network outputs are considered to be 0 and above which they are taken to be 1.

Threshold	Precision	Recall	Accuracy
0.0500	0.2867	0.9191	0.4413
0.1500	0.3670	0.7402	0.6374
0.2500	0.4598	0.6029	0.7392
0.3000	0.5124	0.5564	0.7704
0.3200	**0.5314**	**0.5392**	**0.7791**
0.3400	0.5394	0.5196	0.7820
0.3600	0.5565	0.5074	0.7883
0.4000	0.5780	0.4632	0.7935
0.4500	0.5850	0.4216	0.7929
0.4700	0.6036	0.4069	**0.7970**
0.5000	0.6183	0.3652	**0.7970**
0.5500	0.6000	0.2941	0.7872
0.6500	0.6693	0.2083	0.7889
0.7500	0.7258	0.1103	0.7802

76.4% in the test data). The same will be true of the third baseline (guess a value that minimizes MSE) if the value guessed is below the threshold for output deemed to be 1 and its inverse otherwise. The second baseline (random guessing with appropriate probabilities) achieves a precision of just above 25% with a recall of similar magnitude and an overall accuracy of 62.6%.

In contrast to this, table 5 presents precision, recall, and accuracy on our test data for our very best network — the 20-hidden unit network cited in table 4. The table shows the effect of setting different thresholds for output deemed to be 1. These results show that we are able to achieve better than 53% precision and recall simultaneously if we choose the threshold value correctly. Such a result for a random process, even one having knowledge of the relative frequency of contribution boundaries, is extremely improbable. The accuracy of our most accurate network is 79.7%; it is 77.9% where the values of precision and recall are closest in magnitude. This is an improvement not only over the randomized baseline algorithm, but also slightly overtakes the accuracy of 76.4% of the first baseline on the test set.

8 Conclusion

The experiments that we have reported in this paper are intended to support the long-term goal of building a writer's assistant for diagnosing stylistic inconsistency within a document.

The experiments demonstrate that it is possible to design a system that, with significant probability, can infer the presence of authorship contribution boundaries by means of stylistic statistics, despite the very small sizes of the samples of text available. We have also shown that this is best done with statistics that capture high-level elements of style such as preferences in grammatical constructions (part-of-speech, function-word, and punctuation frequencies). Conversely, we have shown that several categories of stylistic statistics perform poorly on short texts. Especially disappointing in this regard was our failure to extend the excellent results of Juola [1997].

Our results add force to the contention that neural networks are a useful tool in stylometry. While it certainly does not seem that simple multilayer perceptrons trained on amorphous sets of stylistic data are useful, we have shown that networks trained on individual statistical categories can be joined together to produce a result better than that achievable by any of the component networks alone. Perhaps our most significant contribution is the application of time-delay networks to this field, since we are not aware of any prior literature in stylometry where this network architecture has been used. The superior performance that these networks usually exhibited compared to more traditional architectures trained on the same data suggests that this learning paradigm deserves considerably more attention than it has been accorded.

Now that we are able to locate stylistic inconsistencies with reasonable probability, an attempt to develop a system to advise users on the creation of stylistically homogeneous documents is potentially feasible. Many hurdles need to be overcome, of course, not least exactly how statistical information about stylistic inconsistencies could be shaped into a human-comprehensible form. Our use of neural networks as the discriminating mechanism compounds this challenge.

Manual examination of the internals of our most successful networks will be a first step in solving this problem. Conducting further experiments with the three most useful categories of data we have presented here, both to determine what subsets of these categories provide most information and to gain insight into interactions within and amongst the categories, will also be extremely informative and could lead to less computationally intensive methods. It would also be worthwhile to find out whether windows larger than two paragraphs might prove effective, particularly with time-delay networks.

Our success with high-level statistics also raises questions: if part-of-speech tags perform so well, would even higher-level statistics, such as frequencies of noun phrases of various lengths, proportion of prepositional phrases, and so on, be even more useful? Both Stamatatos et al [1999; 2000] and Hatzivassiloglou et al [1999] have successfully demonstrated applications for these sorts of statistics, and a study linking these ideas with powerful neural net architectures should prove very interesting. Nonetheless, we are continuing to look at low-level features, including letter bigrams [Kjell and Frieder, 1992; Kjell et al, 1994], which have also produced some very good results [Graham et al, in preparation, 2003].

Acknowledgements

This research was supported financially by the Natural Sciences and Engineering Research Council of Canada and the University of Toronto. For discussions, comments, and advice, we are grateful to Ian Lancashire, Melanie Baljko, Alex Budanitsky, and Eric Joanis. For providing on-line copies of their papers for conversion to Braille for the first author, we are grateful to David Holmes, Sameer Singh, Fiona Tweedie, Patrick Juola, Robert Matthews, David Mealand, Geoffrey Hinton, Ian Lancashire, Judith Klavans, Dan Jurafsky, and Bill Teahan. We are especially grateful to the creators of the Stuttgart Neural Network Simulator; it is to this powerful, versatile, and singularly reliable system that we owe the broad range of learning algorithms and network architectures that we were able to use in our experiments.

References

Baljko, Melanie and Hirst, Graeme. 1999. Subjectivity in stylistic assessment. *Text Technology*, 9(1), 5–17.

Bishop, Christopher M. 1995. *Neural Networks for Pattern Recognition* Oxford University Press.

Brill, Eric. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4), December 1995, 543–565.

Brown, Peter F.; Della Pietra, Stephen A.; Della Pietra, Vincent J.; Lai, Jennifer C.; and Mercer, Robert L. 1992. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1), 31–40.

Glover, Angela and Hirst, Graeme. 1996. Detecting stylistic inconsistencies in collaborative writing. In: Mike Sharples and Thea van der Geest (eds.), *The New Writing Environment: Writers at work in a world of technology*. London: Springer-Verlag, 147–168.

Graham, Neil. 2000. *Automatic Detection of Authorship Changes within Single Documents.* MSc thesis, Department of Computer Science, University of Toronto, January 2000. Published as technical report CSRG-406. *www.cs.toronto.edu/compling/Publications/ Abstracts/Theses/Graham-thabs.html*

Graham, Neil; Hirst, Graeme; and Marthi, Bhaskara. 2003. Automatic detection of authorship changes within a single document. In preparation.

van Halteren, Hans; Zavrel, Jakub; and Daelemans, Walter. 1998. Improving data driven wordclass tagging by system combination, *Proceedings of the 36th Annual Meeting of the Association For Computational Linguistics and the 17th Annual Conference on Computational Linguistics, COLING/ACL*, Montreal, 491–497.

Hatzivassiloglou, Vasileios; Klavans, Judith L.; and Eskin, Eleazar. 1999. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 203–212.

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

Juola, Patrick. 1997. What can we do with small corpora? Document categorization via cross-entropy. *Proceedings of an Interdisciplinary Workshop on Similarity and Categorization*, Department of Artificial Intelligence, University of Edinburgh.

Kessler, Brett; Nunberg, Geoffrey; and Schütze, Hinrich. 1997 Automatic detection of text genre. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, 32–38.

Kjell, Bradley and Frieder, Ophir. 1992. Visualization of literary style. *Proceedings, IEEE International Conference on Systems, Man and Cybernetics*, Chicago, 656–661.

Kjell, Bradley; Woods, W. Addison; and Frieder, Ophir. 1994. Discrimination of authorship using visualization. *Information Processing and Management*, 30(1): 141–150.

Kučera, Henry and Francis, W. Nelson. 1967. *Computational Analysis of Present-Day American English*. Brown University Press, Providence, RI.

Marcus, Mitchell P.; Santorini, Beatrice; and Marcinkiewicz, Mary Ann. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2), 313–330.

Matthews, Robert A. J. and Merriam, Thomas V. N. 1997. Distinguishing literary styles using neural networks. In: Fiesler, Emile and Beale, Russell, eds., *Handbook of Neural Computation*. CD-ROM Edition, Oxford University Press, Release 97/1, Section G8.1.

Mosteller, Frederick and Wallace, David L. 1964. *Inference and Disputed Authorship: The Federalist*. Addison-Wesley, Reading, MA.

Posner, Ilona R., and Baecker, Ronald M. 1992. How people write together. *Proceedings, The Twenty-fifth Annual Hawaii International Conference on System Sciences*, volume IV, 127–138. Reprinted in: Ronald M. Baecker (ed.) *Readings in Groupware and Computer-Supported Cooperative Work*, Morgan Kaufmann, 239–250.

Ratnaparkhi, Adwait. 1996. A maximum entropy part-of-speech tagger. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, University of Pennsylvania.

SNNS. Stuttgart Neural Network Simulator, University of Stuttgart, Institute for Parallel and Distributed High Performance Systems (IPVR), Stuttgart, 1990–95, and University of Tübingen, Wilhelm-Schickard Institute for Computer Science, Tübingen, 1996–98.

Schmid, Helmut. 1995. TreeTagger — a language independent part-of-speech tagger. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Stamatatos, Efstathios; Fakotakis, Nikos; and Kokkinakis, George. 1999. Automatic authorship attribution. *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, Bergen, June 1999, 158–164.

Stamatatos, Efstathios; Fakotakis, Nikos; and Kokkinakis, George. 2000. Automatic text categorization in terms of genre and author. *Computational Linguistics*, 26(4), 471–495.

Tweedie, Fiona J.; Singh, Sameer; and Holmes, David I. 1996a. An introduction to neural networks in stylometry. In *Research in Humanities Computing 5*, Oxford University Press, 249–263.

Tweedie, Fiona J.; Singh, Sameer; and Holmes, David I. 1996b. Neural network applications in stylometry: The *Federalist Papers. Computers and the Humanities*, 39(1), 1–10.

Wilson, Michael. 1987. *MRC Psycholinguistic Database: Machine Usable Dictionary, Version 2.00* Informatics Division, Science and Engineering Research Council, Rutherford Appleton Laboratory.