Neural Network Identification of Poets Using Letter Sequences

Johan F. Hoorn, Stefan L. Frank, Wojtek Kowalczyk, and Floor van der Ham
Vrije Universiteit, Amsterdam, The Netherlands

Abstract

Stylistic differences among poets are usually sought in sound and semantics. In human analysis, the criteria for recognizing stylistic differences are manifold and intermingled. This study demonstrates that successful identification of poets based on their work is possible using one criterion: letter sequences. Poets show preferences for certain letter combinations, which are unique to their writing style. Using this criterion in machine computation demonstrates that semantics are not needed to identify poets correctly, and that, as a concession to utter parsimony, one minimal criterion of unique letter sequences is enough to fingerprint an author. A small sample of the work of three Dutch poets was used: Bloem (1887–1966), Slauerhoff (1898–1936), and Lucebert (1924–94). This sample formed the training set for the neural network program to analyse the unique letter patterns for each poet. Next, the program was fed a set of new poems, for which the author was to be identified. In choosing between two poets, the program succeeded in identifying the poet correctly for 80–90% of the new poems. When the choice was between three poets, the score was ~70% correct. Since raw ASCII files are sufficient as input, and human pre-coding is unnecessary, neural network analysis of letter sequences may turn out to be a powerful tool in categorization and identification problems, such as genre, stylistics, and plagiarism.

1 Introduction

The underlying idea of this paper is that in each individual mind, specific sounds are uniquely associated: the psychophonological fingerprint of the speaker, as it were. This should be especially true for poets, who use sounds as a creative basis for their work.

If the distinctive features in the style of a poet could be distinguished from the features that are shared with others, the identification of the distinctive features in a given poem should indicate the author with a *certain degree of probability:*

> The customary practice in literary criticism is to demonstrate such formal properties of poetry and prose by pointing to instances.

Correspondence:
Johan F Hoorn,
General and Comparative Literature,
Faculty of Arts, Vrije Universiteit,
De Boelelaan 1105, 1081 HV,
Amsterdam, The Netherlands
E-mail:
hoornj@let.vu.nl

© Oxford University Press 1999

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

> There is justification for this when we consider the effect upon the reader or listener, of whom the critic is an example. But before inferring any process in the behaviour of the writer, it is necessary to allow for the patterning of his verbal behaviour to be expected from chance. In no case, perhaps, can we say that any one instance of alliteration or other formal similarity is due to a special process, but a general pattern may be demonstrated. Alliteration, for example, may be detected by a statistical analysis of the arrangements of initial consonants in a reasonably large sample. A tendency to alliterate is shown by the extent to which the initial consonants in a given literary work are not distributed at random. (Skinner, 1957)

There is a long-standing tradition in the study of literature to analyse the stylistics of poetry in combination with semantics. One of the leading scholars posits:

> An analysis of any linguistic sign whatever can be performed only on condition that its sensible aspect be examined in the light of its intelligible aspect (. . .) and vice versa. The indissoluble dualism of any linguistic sign is the starting point of present-day linguistics in its stubborn struggle on two fronts. Sound and meaning—both these fields have to be thoroughly incorporated into the science of language: speech sounds must be consistently analyzed in regard to meaning, and meaning, in its turn, must be analyzed with reference to the sound form. (Jakobson, 1971)

However, for the proper determination of authorship, the dualism of the linguistic sign is not so indissoluble. If a machine can identify an author correctly by counting the systematic repetitions of letter combinations, then it is impossible to maintain that speech sounds—or their orthographic reflection—*must* be analysed with regard to meaning. If this may constitute an antidote to the Jakobsonian metastasis of formal–semantic equivalence theory, the same machine analysis may count as a powerful apparatus of resuscitation for those who swore by the soundness of the theory. Why? Because humans have a memory capacity problem.

When a literary scholar scrutinizes a poem, he/she may well find interesting phono-semantic connections or highly significant formal equivalencies. However, the scholar also runs the risk of overlooking certain features. In *Les aveugles* (Baudelaire, 1857/1986, p. 66),[1] he/she may find that 'tous ces aveugles?' ('all those blind men?') in the last line of the sextet not only rhymes with 'ris et beugles' ('laughs and rages') in the last line of the first terzet. He/she may also find that 'beugles' sounds like and—with a slight metathesis—shares most letters with 'Bruegel'; Pieter Bruegel the Elder (1525–69) who painted *The Parable of the Blind Men*, after which the poem supposedly is modelled (Van Buuren in Baudelaire, 1986).

Focusing on this aspect, the literary scholar might forget, for instance,

1 Les aveugles
Contemple-les, mon âme; ils
sont vraiment affreux!
Pareils aux mannequins;
vaguement ridicules;
Terribles, singuliers comme les
somnambules;
Dardant on ne sait où leurs
globes ténébreux.
Leurs yeux, d'où la divine
étincelle est partie,
Comme s'ils regardaient au
loin, restent levés
Au ciel; on ne les voit jamais
vers les pavés
Pencher rêveusement leur tête
appesantie.
Ils traversent ainsi le noir
illimité,
Ce frère du silence éternel. O
cité!
Pendant qu'autour de nous tu
chantes, ris et beugles,
Éprise du plaisir jusqu'à
l'atrocité,
Vois! Je me traîne aussi! mais,
plus qu'eux hébété,
Je dis: Que cherchent-ils au
Ciel, tous ces aveugles?
Baudelaire (1857/1986)

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

that in the same poem within one or two letters (spaces included), the *r* is often followed by an *i*. Within two letters: v*rai*ment, pa*rei*ls, te*rri*bles, pa*rti*e, noi*r i*llimité, tra*î*ne. Within one letter: *ri*dicules, te*rri*bles, *ri*s, é*pri*se. Why would this observation be trivial, if the doctrine is that *every* formal equivalence is a semantic equivalence? It obviously becomes highly significant if the combination of *ri* is connected to the sound of laughter in French ('rire' is 'to laugh'), and refers to what Baudelaire thinks is the silly gait of these blind men. The letter sequence *ave* in p*avé*s, tr*aver*sent ends in *ave*ugles, and may well stress the tribute Baudelaire is making to those blind men walking by on their way to heaven.

Equivalence implies repetition. A search for formal equivalencies should thus be performed by tracing the systematically repeated letter strings. If certain repetitions of letter combinations are found throughout Baudelaire's collected works, and turn out to be a unique feature of his writing style, then the trivial letter count loses its insignificance once and for all. Since a human brain cannot memorize every letter in every position of every combination in every poem, and thus fails to discern the bulk of formal equivalencies, an electronic 'brain' could carry out the formal analysis, while the human brain assigns semantics to it.

Previously, only a few attempts had been made at machine-aided poetry analysis. Hayward (1991, 1996) developed a connectionist model of poetic metre, which formed the basis for the successful identification of poets. However, in personally assigning values to stress in a line of poetry, his model suffers from a large degree of subjectivity. Therefore, his work cannot be considered a formal description.

Burrows (1992) and Holmes and Forsyth (1995) exploited multivariate word frequency analysis to discriminate between authors. The latter study also investigated measures of vocabulary richness. Kjell (1994) determined authorship by neural network analysis of letter-pair frequencies ('2-tuples'). However, as Holmes (1985) states, these measures 'take no account of the serial nature of language; a sentence-length distribution, or a word frequency list, does not preserve the order of words or sentences'.

With regard to the order of letters, the window analysis of the present research avoids this criticism. The aim of the research is twofold. The first question is whether a completely computational system can learn to recognize the author of a particular poem. Second, do poets consistently use certain letter sequences, and do these sequences contain enough information to 'fingerprint' the poet? In both cases, it is important that the system is not dependent on any human interpretation or pre-coding whatsoever. Such a dependency would make the system not completely computational, would make analysis of large corpora infeasible, and would raise the question of whether the information really is in the letter sequences or 'in the eye of the beholder'. Therefore, we did not use phonetic transcriptions, because pronunciation has high individual variability, making transcription subject to human interpretation.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

2 Formal Representation and Analysis of Texts

The determination of authorship is a classification problem. The classes are the poets whose poems are to be classified. A general approach to solving classification problems consists of extracting and analysing a number of features from each of the objects to be classified. Features that can be abstracted from poetry include average line length, word length, average number of rhyming letters, average number of syllables per word and per line, poetic metre, frequency of metaphor use, and many others (cf. also Hanauer, 1996).

Although all these features may be useful in classifying poetry, the present study is limited to the information contained in letter sequences, what are termed the *n*-gram frequencies of a text. Listing the relative frequencies of letter sequences for each poet and poem will be the first approach to poet identification.

In a second approach, incomplete poems are used. A text is split into short fragments, and a neural network analyses a small number of letter sequences (what are termed 'windows'). The results are generalized over the complete text. The performance of the two approaches will be compared in this paper.

2.1 *n*-Gram representation and analysis

In an *n*-gram representation, each poem is described by the tabulation of relative frequencies with which different short strings of letters, *n*-grams, occur. The length of such a string is given by the value of *n*. Thus, if $n = 1$, the frequency table gives the relative frequencies of the occurrences of all single letters. Unlike Kjell (1994), the space (_) counts as a letter, so that silence is also considered to be a sound. The famous line by Slauerhoff

In Nederland wil ik niet leven[2]

gives as monograms ($n = 1$): i,n,_,n,e,d,e,r,l,a,n,d,_,w,i,l,_,i,k,_,n,i,e,t,_,l,e,v,e,n. The trigrams ($n = 3$) are: in_,n_n,_ne,ned,ede,der,erl,rla,lan,and,nd_,d_w,_wi, . . ., etc.

For each poem, the different *n*-grams are produced and counted. By dividing the absolute number of each *n*-gram by the total number of *n*-grams in the poem, relative frequencies of *n*-grams are obtained.

The number of possible *n*-grams grows exponentially with *n*. Since there are twenty-seven different characters (twenty-six letters and the space), this number equals 27^n. For $n > 2$, using every possible *n*-gram is not feasible. Even using the existing *n*-grams, i.e. those that actually appear in the texts, is not useful since most occur so infrequently that they have no discriminating power. Therefore, only a fraction of the *n*-grams is considered, each of which should have a minimum frequency of occurrence for at least one of the poets, in order to appear in the frequency table. The rationale is that *n*-grams that do not appear regularly in the work of any of the poets cannot be useful for identification.

2 In The Netherlands, I don't want to live.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

In most research (see, for instance, De Heer, 1982; Tauritz *et al.*, 1997), the value of n is set at 3. This kind of research is usually concerned with classifying English text by meaning. It is far from certain that the same value of n is useful for classifying Dutch poetry by author. Therefore, a short experiment was conducted to determine the value of n for which the frequency tables are most discriminating among authors (see Appendix I.1 for a description of this experiment). This value turned out to be 3 (trigrams).

A trigram frequency table can be seen as a vector with one component for each trigram frequency. Each component has a value between 0 and 1 that gives the relative occurrence of a certain trigram in the poem on which the vector is based. Three techniques for analysing these vectors are neural network classification, k-nearest neighbour classification, and naive Bayes classification. In all methods of analysis, the first step is splitting the set of poems into a test set (the set of poems that are to be classified) and a training set (the set of poems from which the classification is learned).

2.1.1 *Neural network classification*

Neural networks have been shown to be useful in classification problems. Vectors corresponding to items for which the classification is known are called training vectors. These can be fed into the network as a training set. A feedforward network using the standard backpropagation algorithm can then learn to classify the test set correctly (cf. Kjell, 1994). This network will have one input unit for each trigram. The number of output units is equal to the number of different poets. Each output unit represents a poet. The output unit that is most activated determines how an input vector or pattern is classified.

To find the optimal weighting of the network interconnections, the network needs to be trained. In the training phase, the network is shown a sample of the possible input patterns (the training set—here a sample of poems represented by trigrams) combined with the corresponding desired outputs (here the correct poets). Training is done by a process called backpropagation. First, the network is initialized at random: all weights are assigned a random value. Then, an input pattern is applied, resulting in an output. The error of this output is calculated by comparing it with the desired output, and propagated backwards into the network, updating the weights at each layer to decrease the difference between the actual and the desired output. The appropriate formula is known as the generalized δ-rule, and can be found in any textbook on neural networks (e.g. McClelland and Rumelhart, 1986). When all training patterns have been shown, they may be used for further cycles of training until the network's output error is considered acceptable.

After training, the network should not only classify the patterns in the training set correctly, but should also be able to generalize to patterns that were not in the training set (the test set—a fresh sample of poems by the same authors).

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

2.1.2 k-*Nearest neighbour classification*

In nearest neighbour classification (Cover and Hart, 1967), the training vector that is nearest to the test vector is determined. The test vector is classified as belonging to the same class as this training vector. To determine the nearest training vector, different distance measures can be used, for instance Euclidean distance or a correlation coefficient. These and other distance measures are given in Appendix II. *k*-Nearest neighbour classification is a generalization of this. Instead of considering only one nearest training vector, *k*-nearest training vectors are taken into account. The class to which the majority of these *k* vectors belong is chosen as the class of the test vector.

2.1.3 *Naive Bayes classification*

Naive Bayes classification is the method used in the study by Mosteller and Wallace (1964) to classify the *Federalist Papers*. In the naive Bayes method, the probability of a test vector belonging to a class is estimated. To do this, we first need to know the probabilities of occurrence of this test vector $\vec{a}$ and the class c_j. These probabilities are called the a priori probabilities $P(\vec{a})$ and $P(c_j)$. According to Bayes' theorem, the probability of test vector a belonging to class c_j is:

$$P(\vec{a})|c_j) \frac{P(c_j)}{P(\vec{a})}$$

in which $P(\vec{a}|c_j)$ is the probability of vector $\vec{a}$ occurring in class c_j. The class with the highest probability is chosen for the test vector. Since it is very unlikely that two different poems result in the same vector $\vec{a}$, this possibility is ignored and all $P(\vec{a})$ are assumed to be the same. Therefore, the class for which $\vec{a}$ has the highest probability can be determined by:

$$\max_j \arg\{P(\vec{a}|c_j)\, P(c_j)\}$$

which returns j for which $P(\vec{a}|c_j)\, P(c_j)$ is maximal. $P(c_j)$ can simply be estimated by dividing the number of occurrences of c_j in the training vectors by the total number of training vectors. $P(\vec{a}|c_j)$ can be estimated by:

$$\prod_{i=1}^{m} P(a_i|c_j) \qquad (1)$$

in which m is the number of different trigrams used and, $P(a_i|c_j)$, the probability of component a_i occurring in class c_j, is estimated based on the training vectors. In order to do this, all vectors should be discretized so that every component a_i can only have one of a fixed number of levels.

Note that Equation 1 is only correct when the components $a_1, \ldots, a_m$ are independent, which is obviously not the case.[3] Bayes classification is known as 'naive' because this problem is ignored and Equation 1 is still assumed to give a reasonable estimate. In a recent study, it has been shown that the naive Bayes classifier can produce good results even if

3 Definite articles such as *de* (the), which are quite common, will result in a large value for both the trigrams *_de* and *de_*. This shows that they are not independent.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

the independence condition does not hold (Domingos and Pazzani, 1997).

One problem with using Bayes classification is that it treats all values as nominal: when estimating $P(a_i|c_j)$, the training vectors with a certain value for a component are counted. Other values are not taken into account, which means that the values 1 and 20, for instance, are seen as being as different as 4 and 5 are. k-Nearest neighbour and neural network classification, on the other hand, do treat values as numerical: they can recognize that the difference between 1 and 20 is greater than the difference between 4 and 5.

A more thorough explanation of k-nearest neighbour and naive Bayes classification can be found in most books on machine learning or pattern recognition, for instance Mitchell (1997).

2.2 Window representation and analysis

Instead of complete poems, a window representation uses excerpts as training examples or test items. These excerpts are obtained by shifting over the text a 'window' through which only a short sequence of letters can be seen. The size of this window, W, determines how many letters of the poem are visible. Each window of letters constitutes a test or training pattern.

The patterns are converted into a numerical representation. Each letter in a sequence should be given its own separate representation, so that similar patterns also get similar representations. Further, different letters should remain incomparable. A representation such as A = 1, B = 2 is inconvenient, because 'B' would be twice as much as 'A', or 'B' would have twice as much of some property of 'A'. A useful representation is one in which each letter has its own place in a row of digits. If a particular letter occurs, the corresponding digit is set to 1, and all the other numbers are set to 0. For instance, the word POET is represented as:

```
0000000000000001000000000000  (P)
0000000000000010000000000000  (O)
0000100000000000000000000000  (E)
0000000000000000000010000000  (T)
```

Twenty-eight digits per character are used: one for each letter of the alphabet, one for a space, and one for the end of the line. Neural networks can be used to analyse the patterns in this window representation. When twenty-eight different characters are used, the network needs $28W$ input units. The number of output units is equal to the number of different poets.

It is not possible to predict which statistical properties of the letter sequences the neural network will use to distinguish the poets. The network could, for instance, calculate the strengths between a letter and those letters preceding or following it. Given a particular letter p, it might estimate that the third letter following a p is an e, and that the seventh is an s (if the window size is at least seven). Such predictions can be tested in a fresh sample of poems. If specific predictions can be made for

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

each author, analysing letter sequences is useful for authorship determination.

2.3 Comparison of trigram and window representation

Although both representations of poems are suitable for a neural network classification technique, we will call this input 'patterns' in the case of the window representation and 'vectors' in the case of the trigram representation. Patterns can be seen as sequences of letters, while vectors are sequences of frequencies.

In the window representation, each poem is converted into a number of patterns, proportional to the size of the poem.[4] In the trigram representation, each poem results in only one vector. Therefore, the size of the training set is much larger in the window representation than in the trigram representation, which makes the window representation slow to process.

It might seem as if the trigram representation is simply a window representation with $W = 3$. Indeed, the n-grams of a text are identical to the patterns of the same text when $W = n$. However, there are important differences. All letters of a poem are preserved in the window representation, while the poem's trigram vector consists only of counts of some combinations of letters. Furthermore, the similarity of different trigrams is not reflected in the vector. Even two trigrams that only differ in one letter constitute two different components of the vector, and are regarded as having nothing in common. Two patterns that are similar, on the other hand, have representations that are similar and are therefore expected to have a similar effect during analysis.

3 Method

The collected works of three poets were compared: Bloem (1979), Slauerhoff (1961), and Lucebert (1974). It was expected that the first two poets would be more similar to each other than to Lucebert. The first two are contemporaries from the first half of the twentieth century, whereas the latter is from the second half. Therefore, the writing styles of the first two are more akin to each other (traditional verse form, line-end rhyme, little alliteration) than either of them are to Lucebert's style (free verse, unexpected rhyme, much alliteration). If the present approach is effective, Lucebert's poems should at least be correctly distinguished from the other two authors. As a more sophisticated test, it should be possible to distinguish Bloem from Slauerhoff on the basis of their use of letter sequences.

Thirty poems were chosen at random from each of the poets (for the selected poems, see Frank, 1998, Appendix A.1). Four main experiments were conducted, three of which involved discriminating between two poets (Lucebert versus Slauerhoff, Lucebert versus Bloem, and Slauerhoff versus Bloem). In the last experiment, a classification had to be made between the three poets.

The sets of poems were split randomly four times into training sets

4 The number of patterns formed out of a poem is $L - (W - 1)$, where L is the number of letters in the poem and W is the number of letters in the window. This is because each letter of a poem is the start of a pattern, except for the last $W - 1$ ones, where there is not enough poem left to fill the window.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

and test sets, each containing fifteen poems per poet (see Frank, 1998, Appendix A.2). Each experiment was performed with all of the four different training/test splits. The percentage of correctly classified poems, averaged over these four test sets, was taken as the performance of an analysis. Some of the analyses were preceded by a brief initial experiment to determine a good setting of parameters. These experiments are described in detail in Appendix I (this paper).

In coding the poems, no distinction was made between upper and lower case letters. Letters with an accent were treated as the corresponding letter without the accent, and Dutch diphthongs such as *ij* were treated as two letters: *i* followed by *j*. The poems' titles and all punctuation marks were removed.

Regardless of how different a test poem was from any of the training set, it was always classified. Only if a tie between two classes occurred was the poem considered unclassified: this counted as a misclassification.

3.1 Trigram representation

Trigram frequency tables were obtained for each of the sixty poems in the two-poet cases, and for each of the ninety poems in the three-poet case. Only those thirty trigrams which occurred with a minimum frequency of 0.4% for at least one of the poets were used. These trigrams could differ for different combinations of poets. See Appendix III.2 for trigrams and minimum frequencies.

3.1.1 *Neural network classification*

Two independent variables were used: h (the number of hidden units, with values 0, 4, 8, and 12) and c (the number of training cycles, with values 5,000, 10,000, and 30,000). For each number of training cycles, a network was trained from the start. Thus, the performance of the same network after 5,000, 10,000, and 30,000 training cycles was *not* measured.

The feedforward network had three layers, except in the case of $h = 0$, where there were only two layers. Standard backpropagation was used during training. After training each set, the network was tested with the corresponding test set. Every experiment was performed twice.

3.1.2 k-*Nearest neighbour classification*

The experiment described in Appendix I.2 determined which values of k and which distance measure should be used. These values of k were 3, 5, 7, 9, and 11. All six distance measures described in Appendix II were used. Each had a 'vote' on the classification of each test vector. In the case of a tie, the classification was based on the nearest training vector only (this is a 1-nearest neighbour classification). If this did not resolve the tie, the correlation–coefficient distance measure was decisive.

Kjell (1994) normalized the vectors by normalizing all elements separately. In this research, the normalization procedure depended on the distance measure used (see Appendix II).

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

3.1.3 *Naive Bayes classification*

The relative frequency of occurrence of trigram i in a poem is a_i. For each vector and for each i, the interval $[0 \ldots \max(a_i)]$ was discretized into d equal size levels. The values of d ranged from 2 to 20 (which was the result of the experiment described in Appendix I.3). For each value of d, all test vectors in the four sets were classified with respect to the corresponding training sets.

3.2 Window representation

All poems were converted into the representation explained earlier. The codes were turned into patterns with $W = 4$, 8, 10, and 14, using the shifting window method. The values of W were determined in the experiment which is described in Appendix I.4.

3.2.1 *Neural network classification*

In the window representation, the number of patterns in the training and test sets is not equal to the number of poems (as in trigram representation). It is approximately proportional to the total size of all poems in the training set.[4] In the test sets, this is irrelevant, but in the training sets it is not. If the training set for poet A is significantly larger than the training set for poet B, one runs the risk of creating a biased network. Since it has analysed poet A more often, the connections to output unit 'poet A' have higher weights, and the network is more prone to guess 'A' than 'B'. On the other hand, the number of test poems for each poet is the same, so a bias for either poet would reduce the number of correctly classified poems. Therefore, the size of the training sets should be approximately the same. Whenever this was not naturally the case, adjustments were made by adding or removing some small poems from the set (see Frank, 1998, Appendix A.2).

The patterns were used as input for a feedforward network with $28W$ input units, and two or three output units (equal to the number of poets). The number of hidden units was varied: $h = 20$, 40, and 60, as determined by the experiment described in Appendix I.4. A standard backpropagation algorithm was used during the training period. After training a set, the network was tested on the corresponding test set. In testing, every pattern was classified. The unit with the highest activation determined the classification of a pattern. The classification of a whole poem was defined as the classification of the majority of its patterns. If the same number of patterns was assigned to two different poets (and there was no third poet with an even larger number of patterns assigned to it), a tie occurred and the poem was considered misclassified.

4 Results

4.1 Results of trigram representation

Results for the neural network classification for all combinations of poets are shown in Table 1. The averaged results, as well as those for k-nearest neighbour and naive Bayes classification, can be found in Table 2. For

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

Table 1 Percentages of correctly classified poems in neural network classification, averaged over the four test sets and two repeated experiments, for all combinations of poets

Poets	No. of training cycles	No. of hidden units			
		0 (%)	4 (%)	8 (%)	12 (%)
Lucebert/Slauerhoff	5,000	82.1	82.3	81.9	79.6
	10,000	82.5	81.3	80.5	80.9
	30,000	81.7	79 6	76.3	80.9
Lucebert/Bloem	5,000	83.0	74 2	79.2	78.4
	10,000	83.8	77 5	82.5	78.8
	30,000	84 2	81 3	80 4	80.8
Slauerhoff/Bloem	5,000	76 3	70.0	69 2	70 0
	10,000	76 3	71 7	71.3	72 5
	30,000	75 4	69 6	68 4	68 8
Luc/Sla/Blo	5,000	72 0	64 8	67 2	66 2
	10,000	72.3	71 4	69 5	73 9
	30,000	74.5	68.3	70.3	69 2

Table 2 Percentage of correctly classified poems with trigram representation, averaged over the four test sets, for all three types of classification

Poets	Classification type		
	k-Nearest neighbour[a] (%)	Naive Bayes[b] (%)	Neural network[c] (%)
Lucebert/Slauerhoff	80.3	71 2	82 1
Lucebert/Bloem	79 5	67 6	83 6
Slauerhoff/Bloem	68 8	67.5	76 0
Luc/Sla/Blo	61.0	56.1	72.9

[a]Averaged over $k = 3, 5, 7, 9$, and 11
[b]Averaged over $d = 2,$, 20
[c]Averaged over the results without hidden units

k-nearest neighbour and naive Bayes classification, the percentage of correctly classified poems was simply taken to be the average of the percentages for all the different values of the parameters (k and d, respectively). For the neural network classification, further statistics were applied.

In all the following tests, the level of significance $\alpha = 0.05$. ANOVA did not show significant interaction between the number of training cycles and the number of hidden units (Luc/Sla, $P > 0.5$; Luc/Blo, $0.4 < P < 0.5$; Sla/Blo, $P > 0.5$; Luc/Sla/Blo, $P > 0.5$). In general, the highest percentages of correctly classified poems were obtained without hidden units. This was only significant for Lucebert/Bloem ($F_{3,12} = 6.72$; $0.001 < P < 0.01$) and Slauerhoff/Bloem ($F_{3,12} = 8.88$; $0.001 < P < 0.01$). Since the networks not only gave better results, but were also faster without a hidden unit layer, it was decided henceforth to ignore the networks with a hidden unit layer.

Another ANOVA showed that there was no significant effect of the number of training cycles for any of the combinations of poets (Luc/Sla, $0.1 < P < 0.2$; Luc/Blo, $P > 0.5$; Sla/Blo, $0.3 < P < 0.4$; Luc/Sla/Blo, $P > 0.5$). This means that the results for the different

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

numbers of training cycles can be seen as results for three repeated experiments, and that all the values can be averaged to give the mean percentage of correctly classified poems.

Obtaining confidence intervals for these percentages is not as easy as it may appear. *k*-Nearest neighbour and naive Bayes classification are purely deterministic, and in no way do they depend on chance (as opposed to neural network classification, in which the network is initialized randomly). This means that repeating the experiment produces identical results instead of an estimate for the variance. Using different values of parameters *k* and *d*, and averaging the results, as done here, is not in fact a repeated experiment.

There are two ways to estimate the confidence interval. The first is conservative: for each value of the parameters, there are 2 or 3 (poets) × 30 (poems) = 60 or 90 different cases. Each experiment is regarded as binomial with $n = 60$ (in the two-poet cases) and $n = 90$ (in the three-poet case). The other method is liberal. It takes the total number of cases tested, which is 2 or 3 (poets) × 30 (poems) × the number of values of the parameters (for *k* this is 5, for *d* it is 19) × 2 (each poem appears twice in the test sets).

For neural network classification, obtaining the confidence interval is easier because there are six repeated experiments (two for each of the three different training cycle cases), and the intervals can be calculated easily using a *t*-test. Table 3 shows the resulting 95% confidence intervals.

4.2 Results of window representation

The results for the three combinations of two poets are shown in Table 4. Tukey tests were conducted to test for interaction between window size and number of hidden units. No significant interaction was found for Lucebert/Slauerhoff ($0.2 < P < 0.3$) and Slauerhoff/Bloem ($0.2 < P < 0.3$). The interaction was significant for Lucebert/Bloem ($F_{1,5} = 6.61$; $0.025 < P < 0.05$), which means that a one case per treatment ANOVA could not be performed for that combination. This test showed effects of window size for both Lucebert/Slauerhoff ($F_{3,6} = 9.87$; $0.001 < P < 0.01$) and Slauerhoff/Bloem ($F_{3,6} = 7.82$; $0.01 < P < 0.025$). The effect of the number of hidden units was almost significant for Lucebert/Slauerhoff

Table 3 Ninety-five per cent confidence intervals for percentages of correctly classified poems with trigram representation for all three types of classification

Poets	Interval type	Classification type		
		k-Nearest neighbour (%)	Naive Bayes (%)	Neural network (%)
Luc/Sla	Conservative	70.2–90 4	59 7–82.7	81 6–82.6
	Liberal	77 1–83.5	69 3–73.1	
Luc/Blo	Conservative	69.3–89.7	55.8–79 4	82.5–84.7
	Liberal	76 3–82.7	65.7–69.5	
Sla/Blo	Conservative	57 1–80 5	55 6–79.4	75 3–76.7
	Liberal	65.1–72.5	65 6–69.4	
Luc/Sla/Blo	Conservative	50 9–71.1	45.8–66 4	71.5–74.3
	Liberal	57 9–64.1	54.4–57.8	

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

Table 4 Percentage of correctly classified poets, averaged over the four test sets, for all two-poet combinations

Poets	Window size	No. of hidden units		
		20 (%)	40 (%)	60 (%)
Lucebert/Slauerhoff	4	71.2	69.2	74.2
	8	75.8	85.8	89 2
	10	76.7	90.0	85.8
	14	83.8	88.3	87.5
Lucebert/Bloem	4	63.3	72 5	86.7
	8	85.0	87.5	90.0
	10	82 5	92.5	86 7
	14	81 7	85 0	85.8
Slauerhoff/Bloem	4	67.5	56.7	63 3
	8	72 5	76 7	85.0
	10	71 7	77 5	80 0
	14	71.7	75.0	81 7

Table 5 Percentage of correctly classified poets, three-poet case, averaged over the four test sets

Luc/Sla/Blo	No. of hidden units	
Window size	40(%)	60(%)
8	72.2	76.7
10	63.9	65 5
14	69.4	72 8

($F_{2,6} = 4.69$; $0.05 < P < 0.01$) and not significant for Slauerhoff/Bloem ($0.1 < P < 0.2$).

Poorly performing networks were considered uninteresting. With $W = 4$ and $h = 20$, the percentages were low, and these networks were ignored. Consequently, there were no interactions or W effects left (interaction: Luc/Sla, $0.1 < P < 0.2$; Luc/Blo, $P > 0.5$; Sla/Blo, $P > 0.5$; window size: Luc/Sla, $P > 0.5$; Luc/Blo, $0.4 < P < 0.5$; Sla/Blo, $P > 0.5$). The effect of h was almost significant for Slauerhoff/Bloem ($F_{1,2} = 11.4$; $0.05 < P < 0.01$), and not significant for the other combinations (Luc/Sla, $P > 0.5$; Luc/Blo, $P > 0.5$).

As a point estimate of the performance of window representation, the results for $W = 8$, 10, 14 and $h = 40$, 60 can be averaged. If a 95% confidence interval is required, however, the same problem arises as in k-nearest neighbour and naive Bayes classification for trigram representation. In this case, the problem is not that repeated experiments would give the same results, but that there are no repeated experiments at all. Again, only a confidence interval can be obtained that is either too conservative or too liberal; both are given in Table 7.

Tests with $W = 4$ and $h = 20$ were not performed for the three-poet case. Results for the other values of W and h are shown in Table 5. The corresponding decision matrix is shown in Table 6. Point estimates and confidence intervals can be found in Table 7.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

4.3 Overview of results

Figure 1 gives an overview of all point estimates of the percentage of poems classified correctly. Results are shown for both representations, all types of classification, and all combinations of poets.

Table 6 Decision matrix for window representation: Percentages of poems by each poet and percentage of all poems, classified as . . ., total of all six cases shown in Table 5

Poet	Classified as			
	Lucebert (%)	Slauerhoff (%)	Bloem (%)	Unclassified (%)
Lucebert	78.9	8.9	11 7	0 6
Slauerhoff	15.8	47 8	35.6	0 8
Bloem	8.1	8 1	83.6	0.3
All	34.3	21 6	43.6	0.6

Table 7 Point estimates and confidence intervals for all poet combinations

Poets	Point estimate[a] (%)	Interval type	95% confidence interval[b] (%)
Luc/Sla	87.8	Conservative	79.5–96 1
		Liberal	85 5–90 3
Luc/Blo	87 9	Conservative	79 6–96.2
		Liberal	85 5–90 3
Sla/Blo	79.3	Conservative	69 1–89.6
		Liberal	76.3–82 3
Luc/Sla/Blo	70 1	Conservative	60.6–79.6
		Liberal	67 4–72.8

[a]Averaged over $W = 8, 10, 14$ and $h = 40, 60$

[b]In the two-poet cases, conservative confidence intervals are based on binomial distribution with $n = 60$, for the liberal interval $n = 2 \times 6 \times 60 = 720$ In the three-poet case, $n = 90$ for the conservative interval and $n = 1080$ for the liberal interval

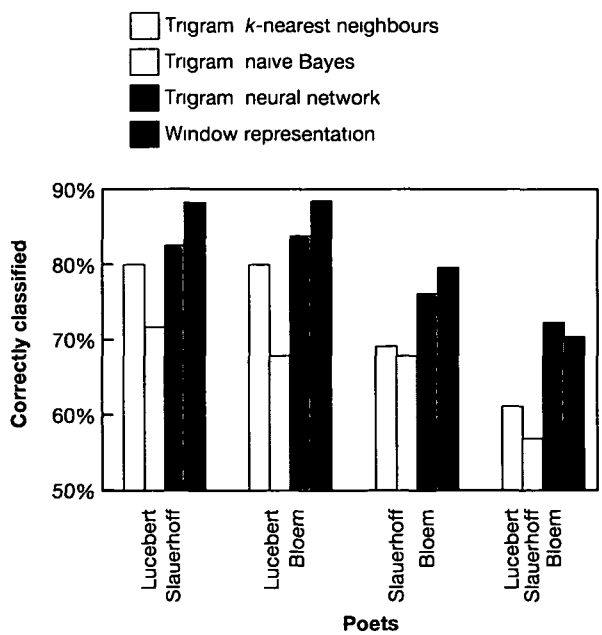

Fig. 1 Overview of results. For all four combinations of poets, the percentage of correctly classified poems is shown. *k*-Nearest neighbour classification, white; naive Bayes classification, hatched; neural network classification, blocked; window representation, black.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

5 Discussion

From the conservative confidence intervals, it may be concluded that each method gives results that are significantly above chance levels (which are 50% for the two-poet cases and 33% for the three-poet case). Bloem, Slauerhoff, and Lucebert each have their own typical use of letter sequences. This use is consistent enough, and the differences between the poets are great enough to identify their poetry reliably based on this single criterion.

5.1 Comparing methods

In comparing the different methods with a conservative confidence interval, differences were only found between the neural network method and the naive Bayes method. To investigate the differences between methods further, a comparison with a liberal confidence interval was performed.

Naive Bayes classification turns out to give the lowest performance for the trigram representation. This could be due to the problem indicated earlier: it treats the trigram frequencies as nominal rather than numerical values. The results of the neural network and *k*-nearest neighbour classification are comparable, except in the three-poet case where neural networks outperform both of the others, even significantly according to the conservative interval. In general, neural networks give the best results for the trigram representation. The window representation performs even better in the two-poet cases. In the three-poet case, it does not give significantly lower results than the neural network classification for the trigram representation.

5.2 Comparing poets

Except for the Bayes classification, all the methods result in lower percentages for the Slauerhoff versus Bloem case than for any of the other two comparisons. Table 6 shows that poems by Slauerhoff are more prone to be classified as Bloem than as Lucebert, but this might not be caused by a difficulty in distinguishing Bloem and Slauerhoff. After all, the opposite is not the case: poems by Bloem are not classified more often as Slauerhoff than as Lucebert. The imbalance in the way poems by Slauerhoff are misclassified is probably the result of the network's bias. Table 6 also shows that the network prefers to classify a poem as Bloem: although 33.3% were by his hand, 43.6% of the poems were attributed to him. Since there was no surplus of Bloem's poetry in the training set, the cause of this bias is unclear. It could simply be a result of the way in which the sets of poetry were split in a training set and a test set. The training and test set of Bloem's poems might be more similar than those of Slauerhoff, purely by coincidence.

Making a distinction between Slauerhoff and Bloem seems harder than between the other poet combinations. The two are contemporaries, both of whom write in a more traditional fashion, using line-end rhymes.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

Yet, the largest window size was fourteen letters, which is not enough for line-end rhyme to occur. Thus, what made them more similar?

5.3 Questions to be answered

Due to its pioneering nature, the current research leaves many questions unanswered, some of which are reflected on in this section.

5.3.1 *How can performance be improved?*

There is no reason to assume that the obtained percentages of correctly classified poems are at a maximum. Improving performance might be possible by enlarging the training set or using other methods.

Should there be a larger training set? The effect of the size of the training set for *k*-nearest neighbour and naive Bayes classification can be estimated by comparing the results in Table 2 with the results of the leave-one-out method (Appendices I.2 and I.3). In the leave-one-out method, each single vector is tested with all other fifty-nine vectors as the training set. The results in Table 2 are based on a training/test set split with only thirty vectors in the training set. The leave-one-out method shows a consistently higher percentage of correctly classified poems than Table 2. Therefore, almost doubling the training set improves the performance for trigram representation. The leave-one-out method was not applied to neural network classification, but there seems no reason to assume that the effect would be any different. However, the effect of a larger training set was only marginal and, in practice, it might be impossible to enlarge the training set substantially, simply because there are insufficient poems available.

Are there other computational methods? Results might improve when more sophisticated versions of the methods are used, or when completely different approaches to analysis are applied. For instance, Taurıtz *et al.* (1997) also used trigram frequency tables, but applied genetic algorithms in order to classify texts by topic.

If the aim is purely to arrive at the best classification system possible, without any linguistic interest in the information contained in letter sequences, more can be included in the analysis. Not only letter sequences may be considered, then, but also punctuation marks, typographic information, and explicit representations for word length, line length, line-end rhyme, and alliteration.

5.3.2 *Why does it work?*

The answer seems obvious: since different poets consistently use different sequences of letters. But why do they? What is the classification actually based on? A clue might be found by looking into the information gains of the different trigrams (as defined in Appendix I.2), which are shown in Appendix III.2. A large information gain means that the corresponding trigram is highly discriminating between two poets: it is used by one of the poets more often than by the other. Three possible causes are: a difference in topic, a difference in spelling and grammar, and a difference

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

in personal preference. The information gains can indicate which of these aspects are important in distinguishing between two poets.

It was found that the window representation performs better for larger window sizes (Table 4). This suggests that in a window representation, criteria which are of importance are different from those in a trigram representation. Because of the huge number of different patterns, it is extremely difficult to gain an idea of which patterns are typical for a certain poet. This restricts the account of the possible answers to the question 'Why does it work?' to trigram representation.

Topic is one answer. Different poets presumably have different topics they prefer to write about (e.g. Slauerhoff, The Sea; Bloem, Death). In using different topic words, different letter sequences systematically appear on the paper. Thus, one would expect the Dutch three-letter string *zee* (sea) to be more frequent for Slauerhoff than for the other two poets. Trigrams derived from *dood* (death) should be most typical for Bloem, and perhaps *oog* (eye) and *oor* (ear) for Lucebert. Yet, the present research shows that these trigrams were not necessary to distinguish between the poets. Thus, is the conclusion justified that the poems were classified purely orthographically, and not semantically?

An objection to the orthographic explanation could be that Slauerhoff is known for writing about ships (*schip* in Dutch) more than the other two poets. Highly frequent occurrences of *schip* could make the trigram *sch* become distinctive between Slauerhoff and the other two poets. However, the same is true for *chi* and *hip* but, unlike *sch*, these did not occur often enough to make it into the trigram frequency table.

The spelling convention of the time is a possible answer. Although the supposed frequent use of *schip* is not the answer, Figs A4 and A6 (Appendix III.2) show a fairly high information gain of the trigram *sch* in the Lucebert versus Slauerhoff and Slauerhoff versus Bloem cases, whereas it is indeed absent in the Lucebert versus Bloem comparison. However, this finding coincides with a major spelling change in Dutch in the 1930s (spelling Marchant), in which *sch* for words such as *visch* (fish) and *mensch* (human being) was replaced by *s*. In the same vein, the definite article in the accusative case changed from *den* to *de*. Thus, Slauerhoff used *sch* more as a result of a general spelling reform, rather than a preference for this particular trigram.

It could be countered, however, that the same should be found for Bloem, who also wrote in the old-style spelling, and yet did not show the highly frequent use of *sch*. Unfortunately, the edition used in the present analysis (Bloem, 1979) was updated to the modern spelling, replacing *sch* by *s*. Only the use of *den* was preserved. It is exactly the conservation of this *den* that may save the viewpoint of uncontaminated orthographic analysis.

Figures A4–A6 (Appendix III.2) show that the information gain for *den* is higher when Lucebert is compared with Slauerhoff or Bloem than when the latter two are compared with each other (notice the differences in scale of the information gain axes). In other words, both Bloem and Slauerhoff could be affected by the same old-style spelling.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

However, strong performers that distinguished Bloem from Slauerhoff were *_de*, *_en*, *de_*, *en_*, *n_d*, which cannot be explained away by a spelling difference between *de* and *den*. Thus, one of the poets showed a strong preference for these particular trigrams.

Are personal preferences the answer? As indicated above, it may be observed in Fig. A6 (Appendix III.2: Slauerhoff versus Bloem) that both the trigrams *_en* and *en_* have a relatively high information gain. This obviously means that the word *en* (and) is used by one poet more frequently than by the other. Indeed, this word occurs about 50% more often in the poems by Bloem than in those by Slauerhoff. Assuming that it has no connection to any topic preferred by one of the poets (it does seem hard for a function word to have this kind of connection) and that the poems constitute a representative sample, it seems that Bloem has a strong preference for conjunction and enumeration as his stylistic devices. This does not mean, however, that the highly frequent use of *sch* for Slauerhoff is not also a result of the old-style spelling.

To tackle the spelling problem from another angle, a new window analysis was performed on three contemporary poets, who belong to the same artistic group (the group of poets called The Fifthiers Group, who were allied with CoBrA): Lucebert (1974), Elburg (1975), and Andreus (1984). Methods were as described above. Table A1 (Appendix IV) shows that on average, 70.8% of the poems in the comparison Lucebert versus Elburg were classified correctly. With a conservative (95%) confidence interval, 59.3–82.3%; with a liberal one, 67.5–74.1%. For the comparison Lucebert/Andreus/Elburg, the mean of correctly classified poems was 56.8% (33.3% is the level of chance). Conservative estimates were 46.6–67.0%, liberal were 53.8–59.8%. Thus, again, poets of the same period were harder to distinguish, yet were quite well distinguishable. Since the poems of Lucebert, Elburg, and Andreus do not suffer from differences in spelling period and were yet correctly classified, it is unlikely—however, not impossible—that the difference between Slauerhoff, Bloem, and Lucebert is explained by the spelling change of the 1930s.

In summary, the classification could be based on at least three different aspects: topic, spelling period, and personal preference. There might be poets who are similar in all these aspects, making it hard to tell them apart. What would happen with pastiches (poems in which the style of another poet is imitated)? Favourite topics and period-dependent language are easily copied. An author's preferred combinations of letters, however, may include too many elements to memorize and may be too inaccessible to be imitated successfully.

5.3.3 *Will it work for other texts and a larger number of poets?*

For an authorship determination system to be useful in practice, it needs to know and reliably recognize more than three poets. There is no theoretical limit to the number of poets that can be included, but the reliability of the classification will certainly decrease when the number of poets increases. The desired reliability determines how many poets can be used in one analysis.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

Will it work for other combinations of poets? The fact that Lucebert, Slauerhoff, and Bloem can be distinguished at a letter-based level does not mean that the same is true for any combination of poets. However, Fig. 1 shows that for all poet pairs, window representation correctly classified the highest number of poems, followed by the three trigram representation analyses in the same order. In addition, many parameters had the same optimal value for different poet pairs. For instance, the neural network classification of trigram representations produced the best results without hidden units for all combinations of poets. This suggests that the results may be similar for other poets.

Will it work for other types of text? Is there something unique about poetry that makes letter-based determination of authors possible? Is the systematic letter-cluster repetition a specific feature of the poetic genre? This question can be answered experimentally by trying to carry out the present analysis on literary prose and journal articles (cf. Kjell, 1994). At least for high-frequency words, Mealand (1997) found different clusters for different genres in the Gospel of Mark. The 'topic' and 'period' aspects, which were important for computational analysis of poetry, are probably also present in other types of text. For 'personal preference' this need not be the case. Although authors of prose presumably also have their personal preferences, it might be the repetition of sounds and letters typical for poetry which makes these preferences clear enough to be used in determining authorship.

Will it work for historical texts? The window analysis is capable of reliably determining an author, based on a single text line as the test set. Would it recognize the author of a historical text of which the authorship is in question? With a certain degree of probability, the present approach could indicate which is the most plausible author, given sufficient historical text for which the author is known being used as the training set.

5.3.4 *Could genres, poet groups, and characters in novels be identified?*

If a poet shows unique patterns in the letter sequences, he or she also shares specific sequences with others. It might thus be that certain poets stand out against one group more than the other, indicating their looser or tighter relatedness with such a group. Certain genres might coincide with the use of specific letter patterns. The excessive style of impressionist writing may show characteristic letter sequences not present in the frugal expressionist style. In line with McKenna and Antonia (1996), certain characters in novels may have their preferred way of combining letters and letter strings (words are nothing but the latter).

5.3.5 *How does it compare with human performance?*

The present research does not claim any resemblance between poem classification by the described techniques and the way in which humans perform such a task. The present approach is not capable of recognizing 'mannequins' in *Les aveugles* (Baudelaire, 1857/1986, p. 66) as the etymological corruption of 'mannekens', meaning 'small men' in the Dutch

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

language of Bruegel's time. It would not make a connection between these 'small men' and the silly-walking blind men who Baudelaire is describing. The metathesis in *aveug*les and *vague*ment, or in *mon âme* and *somnam*bules is not recognized either. Since the present approach does not skip white spaces, the formal equivalence between *silen*ce and ain*si le n*oir cannot be detected, let alone that it may construe that this formal equivalence exhibits darkness as the brother of silence ('Ce frère du silence éternel'). However, people probably cannot judge poetry on the same letter-based level—irrespective of semantics—and as radically as studied here. Therefore, comparing human performance with machine performance is interesting for two reasons.

First, a classification machine that is useful in practice should perform significantly better than humans do. This need not be the case for all poet pairs, since it is likely that some poets are easier to distinguish at a level that is not accessible to computational analysis. For instance, Lucebert may use words that were not known during Slauerhoff's lifetime. Even people with minimal knowledge of these poets can easily spot this, whereas the present approach cannot. On the other hand, poets that seem similar to human readers might be highly distinctive to a computational method. It is to enable distinctions to be made between these poets that such a method may be a useful tool.

Secondly, comparing human judgement with automated classification teaches us about the relationship between letter sequences and poetic styles, genres, or periods. We have seen that Slauerhoff and Bloem, whose styles are akin, are also more related on a letter-based level than the other poet pairs. If, in general, two poets who human readers consider related are harder to distinguish by computational means, this would imply a strong connection between the superficial letter-based level and the deeper level of style. Since the current research is too limited to draw such a conclusion, further research is indispensable.

To summarize:

> . . . content is mediated by the configuration in which it appears. . . . The trajectory of mediation can be retraced in the structure of art works, especially in artistic technique. The more we know about technique, the more we grasp the objectivity of art works, for objectivity is rooted in the consistency of configuration. This objectivity, then, is nothing more than the truth content of art. Aesthetics has the job of surveying the topography of these constituent parts. (Adorno, 1984)

Which is exactly what we have done.

References

Adorno, T. W. (1984). *Aesthetic Theory*. London: Routledge and Kegan Paul, p. 398.

Andreus, H. (1984). *Verzamelde Gedichten*. Amsterdam: Uitgeverij Bert Bakker.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

Baudelaire, C. (1986). *Les Fleurs du Mal Een Bloemlezing.* Ambo, Atheneum-Polak and Van Gennep, Amsterdam: pp. 66, 92.

Bloem, J. C. (1979). *Verzamelde Gedichten.* Athenaeum-Polak and Van Gennep, Amsterdam.

Burrows, J. F. (1992). Not unless you ask nicely: the interpretative nexus between analysis and information. *Literary and Linguistic Computing*, 7: 91–109.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification *IEEE Transactions on Information Theory*, 13· 21–7.

De Heer, T. (1982). The application of the concept of homeosemy to natural language information retrieval. *Information Processing & Management*, 18: 229–36.

Domingos, P. and Pazzani, M. (1997). On the optimality of the naive Bayesian classifier under zero-one loss. *Machine Learning*, 29: 103–30.

Elburg, J. (1975). *Gedichten 1950–1975* Amsterdam: De Bezige Bij.

Frank, S. L (1998*)*. Computational Analysis of Poetry. Using Neural Networks to Determine Authorship. M.Sc. thesis, Vrije Universiteit, Amsterdam.

Hanauer, D. (1996). Integration of phonetic and graphic features in poetic text categorization judgements. *Poetics*, 23: 363–80.

Hayward, M. (1991). A connectionist model of poetic meter. *Poetics*, 20: 303–17.

Hayward, M. (1996). Analysis of a corpus of poetry by a connectionist model of poetic meter. *Poetics*, 24: 1–11.

Holmes, D. I. (1985). The analysis of literary style—a review. *Journal of the Royal Statistical Society Series A*, 148: 328–41.

Holmes, D. I. and Forsyth, R. S. (1995). The *Federalist* revisited: new directions in authorship attribution. *Literary and Linguistic Computing*, 10: 111–27.

Jakobson, R. (1971). *Selected Writings, II, Word and Language.* The Hague: Mouton, p. 104.

Kjell, B. (1994). Authorship determination using letter pair frequency features with neural network classifiers. *Literary and Linguistic Computing*, 9: 119–24.

Lucebert (1974). *Verzamelde Gedichten.* Amsterdam: De Bezige Bij.

McClelland, J. and Rumelhart, D. (1986). *Parallel Distributed Processing,* Volumes 1 and 2. Cambridge, MA: MIT Press.

McKenna, W. and Antonia, A. (1996). 'A few simple words' of interior monologue in Ulysses: reconfiguring the evidence. *Literary and Linguistic Computing*, 11: 55–66.

Mealand, D. (1997). Measuring genre differences in Mark with correspondence analysis. *Literary and Linguistic Computing*, 12: 227–45.

Mitchell, T. M. (1997). *Machine Learning.* New York: The McGraw-Hill Companies, pp. 177–200.

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

Quinlan, R. (1986). Induction of decision trees. *Machine Learning*, 1: 81–106.

Skinner, B. F. (1957). *Verbal Behaviour.* London: Methuen, p. 247.

Slauerhoff, J. J. (1961). *Verzamelde Gedichten.* The Hague: Nijgh and Van Ditmar.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

Tauritz, D. R., Sprinkhuizen-Kuyper, I. G., and Kok, J. N. (1997). Evolutionary computation applied to adaptive information filtering. In van Marcke, K. and Daelemans, W. (eds), *Proceedings of the Ninth Dutch Conference on Artificial Intelligence*. Antwerp, Belgium, pp. 17–26.

Appendix I: Determining Optimal Parameter Values

I.1 *n*-Gram representation: determining *n*

It is obvious that a larger value of n gives larger differences among the n-gram frequency tables of different texts. It should be known which value of n is ideal for making a classification by author.

Classification is easiest if the test vectors resemble the training vectors. Therefore, the vectors describing different poems by the same poet should be as similar as possible, which is the case when n is small. On the other hand, to facilitate discrimination between the poets, the vectors that describe poems with different authors should be as different as possible. This is the case when n is large. This shows there is a trade-off between training and test set similarity on the one hand (the within-poet distance) and poet distinctiveness on the other (the between-poets distance). The within-poet distances should be small (small n), and the between-poets distances should be large (large n). There ought to be some optimal, intermediate value of n.

Assume two poets p and p'. The vector describing poem i by poet p is p_i, the vectors describing the complete set of poetry by each poet are p_{tot} and p'_{tot}. This is visualized in Fig. A1. Then, the between-poets and within-poet distances are given by:

$$\text{between}(p,p') = \text{dist}(p_{tot},p'_{tot})$$

$$\text{within}(p) = \tfrac{1}{m}\Sigma\text{dist}(p_i,p_{tot})$$

where m is the number of poems by poet p, and dist can be any distance measure. In order to compare the distances for different values of n, the distance measure has to be one which includes normalization, because

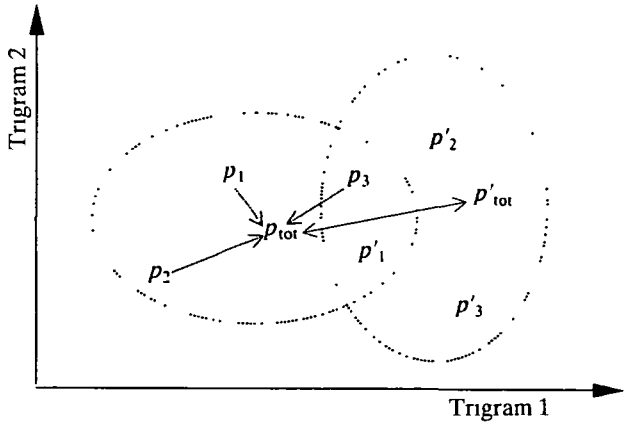

Fig. A1 Two-dimensional vector space containing a sample of vectors for p and p'.[5] The ovals mark the areas in which vectors for the same poet occur. The dark arrow has length between(p,p'). The average length of the dashed arrows equals within(p).

5 In practice, this vector space will have more dimensions: one for each trigram.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

for larger n the frequencies at which the n-grams occur will be lower. This makes the comparison of different n values impossible if the vectors are not normalized first.

n-Gram frequency tables were computed for all poems individually, as well as for the total set of poems by each author. Only Lucebert and Slauerhoff were analysed. It is assumed that the results are comparable for other combinations. The frequency tables are shown in Appendix III.1.

Next, between-poets distances and within-poet distances for $n = 1, \ldots, 5$ were calculated. The distance measure used was normalized Euclidean distance. What we were actually interested in was the difference between the within-poet distances and the between-poets distances. The within-poet distance should be small, while the between-poets distance should be large. This means that the optimal value of n is the value for which within-poet distance minus between-poets distance is minimal.

Results

As expected, larger n resulted in larger distances (Fig. A2). The difference between within- and between-poets distances is shown in Fig. A3. The difference between both within-poet and between-poets distances is smallest for $n = 3$. Therefore, the optimal n-grams are trigrams.

I.2 k-Nearest neighbour: determining k and the distance measure

To avoid the effect of training/test sets on the experiment, it is common practice to use the leave-one-out method: each of sixty vectors (thirty per poet) is tested with the fifty-nine other vectors as the training set. The score of a test is the number of test vectors that are classified correctly. The values of k that were tested were $k = 1, 3, 5, 7, 9$, and 11. The six distance measures were utilized as described in Appendix II.

There was a main effect for k but, after removing $k = 1$ (which gave the worst results), this effect disappeared (one case per treatment

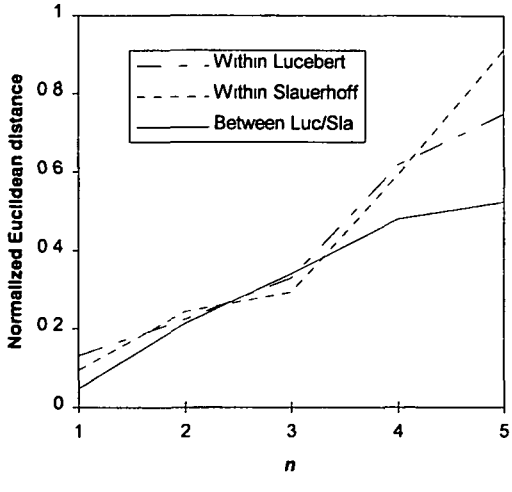

Fig. A2 Within-poet and between-poets distances as a function of n.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

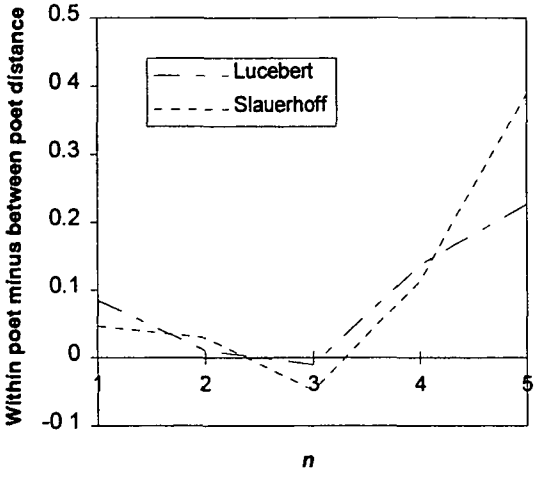

Fig. Å3 Difference between within-poet and between-poets distances as a function of *n*.

ANOVA). Although a significant distance–measure effect was found, there was no single measure with an overall best performance. Therefore, democracy was introduced, and all six distance measures 'voted' over the class of a test vector. The one with most votes won.

In addition, the effect of using the *information gain* (Quinlan, 1986) was investigated. The information gain is a measure of the importance of the different vector components. It is likely that some trigrams are more informative than others. Trigrams which discriminate more efficiently between poets should perhaps be given more weight than others. To determine the information gain, the vectors should first be discretized. When a classification is made between two poets, and the number of vectors is the same for both poets, the information gain of a component a is given by:

$$\text{gain}(a) = 1 - \sum_{i=1}^{d} \frac{p_i + p'_i}{2n} \text{I}(p_i, p'_i) \qquad \text{(A1)}$$

where n is the number of vectors from each class (the number of poems by each poet), and p_i and p'_i are the numbers of vectors from classes p and p' respectively, for which component a has value i. I(p,p') is given by:

$$\text{I}(p,p') = -\left[\frac{p}{p+p'} \log_2\left(\frac{p}{p+p'}\right)\right] - \left[\frac{p'}{p+p'} \log_2\left(\frac{p'}{p+p'}\right)\right]$$

The value of d was set to 10. The information gain was used in three different ways:

- each vector component was multiplied by its information gain;
- only the ten most informative trigrams were used;
- only the twenty most informative trigrams were used.

For the three combinations of two poets, leave-one-out tests with democratic decision making and different uses of information gain were performed. Tukey tests showed no significant interaction between k and

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

the way in which information gain was used. Moreover, there was no main effect of k ($k = 1$ was not tested for the reasons explained above).

Only for Lucebert/Slauerhoff was the effect of information gain clearly significant. In this case, exclusion of information gain gave the highest percentage of correctly classified poems (85.0% versus an average of 82.1% for the other three). In the cases of Lucebert/Bloem and Slauerhoff/Bloem, this was 80.3% versus 81.4% averaged over three and 71.0% versus 69.2% averaged over three, respectively. Since exclusion of information gain produced better results and is easier to implement, it was decided not to use information gain for the rest of the experiments.

I.3 Naive Bayes: determining *d*

Again, the leave-one-out method was employed. The only parameter tested was *d*, which is the number of levels into which the components of the vectors are discretized before applying naive Bayes classification. Values of *d* ranged from 2 to 20.

After the leave-one-out method for the three combinations of two poets, for all values of *d*, regression analysis was done to see if there was any significant effect of *d*. None of the estimated regression lines had a slope significantly different from zero. This means there is no effect of *d*. The averaged percentages of correctly classified poems were 71.3 for Lucebert/Slauerhoff, 73.2 for Lucebert/Bloem, and 71.6 for Slauerhoff/Bloem.

I.4 Window representation: determining *W* and *h*

Since the window representation is extremely time-consuming, it is important first to have some idea of a proper range of values for the independent variables. These variables were *W*, the number of letters in the window, and *h*, the number of hidden units. Only one of the training/test set splits was tested, and only for the Lucebert/Slauerhoff case.

A small number of hidden units or $W = 3$ resulted in low percentages of correctly classified poems (50–63%), whereas more hidden units and a larger *W* seemed to improve them (53–97%). Because of these data, it was decided to use $W = 4$, 8, 10, and 14; and $h = 20$, 40, and 60 for the rest of the experiments.

Appendix II: Distance Measures

Six distance measures were used in *k*-nearest neighbour analysis:

- Correlation coefficient:[6]

$$\text{dist}(X,Y) = 1 - \frac{\text{cov}(X,Y)}{\sigma(X)\sigma(Y)} = 1 - \frac{\Sigma(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\Sigma(X_i - \bar{X})^2 \Sigma(Y_i - \bar{Y})^2}}$$

- Euclidean:

$$\text{dist}(X,Y) = \sqrt{\Sigma(X - Y)^2}$$

6 Correlation coefficient (*cc*) and normalized inproduct (*ni*) increase when the difference between two vectors decreases. Therefore, they are actually proximity measures instead of distance measures. To compensate for this, $1 - cc$ and $1 - ni$ are used here.

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

- Normalized Euclidean: same as Euclidean, but instead of X and Y, the normalized vectors X' and Y' are used:

$$X' = \frac{X}{\sqrt{\Sigma X^2}}$$

- Hamming:

$$\text{dist}(X,Y) = \Sigma|X - Y|$$

- Normalized Hamming: same as Hamming, but instead of X and Y, the normalized vectors X' and Y' are used:

$$X' = \frac{X}{\Sigma|X|}$$

- Normalized inproduct:[6]

$$\text{dist}(X,Y) = 1 - \frac{X}{\sqrt{X \cdot X}} \cdot \frac{Y}{\sqrt{Y \cdot Y}}$$

where $\cdot$ is the inproduct operator.

Appendix III: *n*-Grams

III.1 *n*-Grams and their occurrence

Figure A4 shows the trigrams used in Appendix I.1 as well as their frequencies of occurrence.

III.2 Trigrams and their information gain

Figures A5–A7 show the thirty trigrams analysed and their information gains for all two-poet combinations. All vectors were discretized into ten equal length intervals. Information gains were calculated using Equation A1 in Appendix I.2.

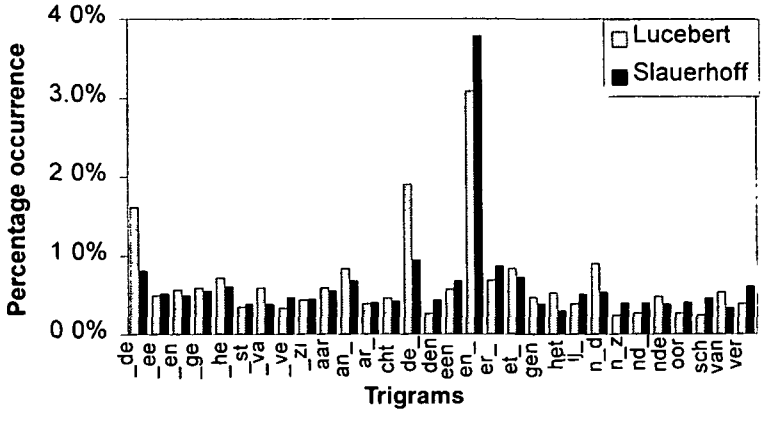

Fig. A4 Trigrams and their frequencies of occurrence

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

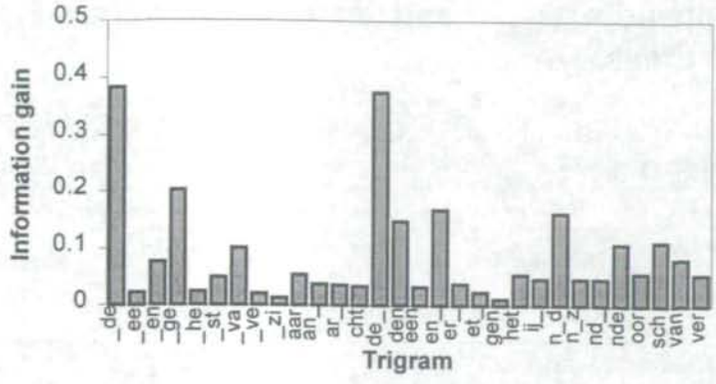

Fig. A5 Trigrams and their information gains for Lucebert/Slauerhoff classification.

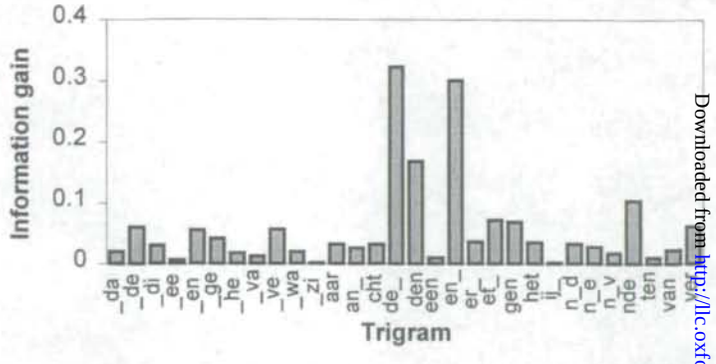

Fig. A6 Trigrams and their information gains for Lucebert/Bloem classification.

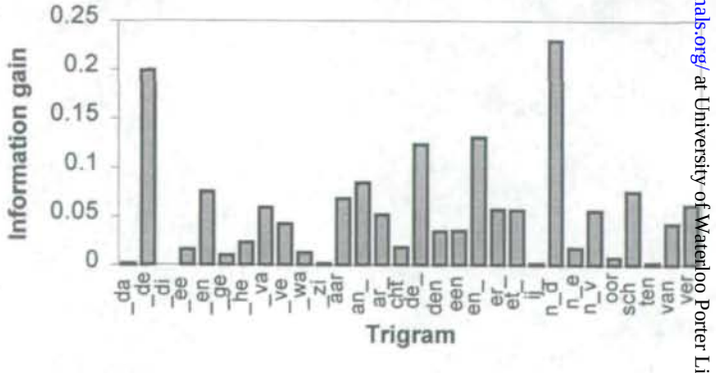

Fig. A7 Trigrams and their information gains for Slauerhoff/Bloem classification.

The following thirty trigrams were used for the three-poet classification:

_da	_he	an_	er_	n_v
_de	_va	cht	et_	nde
_di	_ve	de_	gen	sch
_ee	_wa	den	het	ten
_en	_zi	een	ij_	van
_ge	aar	en_	n_d	ver

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013

Appendix IV: Lucebert versus Elburg versus Andreus

Table A1 Point estimates and confidence intervals for window representation of Lucebert versus Elburg, and for Lucebert versus Elburg versus Andreus

Poets	Point estimate[a] (%)	Interval type	95% confidence interval[b]
Luc/Elb	70 8	Conservative	59.3–82.3
		Liberal	67 5–74.1
Luc/Elb/And	56 8	Conservative	46.6–67 0
		Liberal	53.8–59.8

[a]Averaged over $W = 8, 10, 14$ and $h = 40, 60$

[b]In the Luc/Elb case, conservative confidence intervals are based on binomial distribution with $n = 60$, for the liberal interval $n = 2 \times 6 \times 60 = 720$ In the three-poet case, $n = 90$ for the conservative interval and $n = 1080$ for the liberal interval

Downloaded from http://llc.oxfordjournals.org/ at University of Waterloo Porter Library on June 10, 2013