On poem recognition

Article *in* Pattern Analysis and Applications · November 2006
DOI: 10.1007/s10044-006-0044-8 · Source: DBLP

2 authors:

Hamid R. Tizhoosh
University of Waterloo

Rozita Dara
University of Guelph

Pattern Anal Applic (2006) 9:325–338
DOI 10.1007/s10044-006-0044-8

THEORETICAL ADVANCES

On poem recognition

H. R. Tizhoosh · R. A. Dara

Received: 1 March 2005 / Accepted: 21 July 2006 / Published online: 3 October 2006

Abstract Literature is one of the most significant forms of human culture. It represents a high level of intellectual activity. Humans write, read, and enjoy poems in different cultures. The purpose of this work is to initiate research in the field of poem recognition, not to understand them by electronic devices, but to provide humans with modern tools to search for and work with lyrics. This study is concerned to illuminate the challenges of poem recognition from different perspectives. In this paper, various measures of poetry are introduced and their effectiveness are investigated in different case studies using fuzzy logic, bayesian approach, and decision trees.

1 Introduction

Despite ongoing research in the topic areas of poetry generation and natural language understanding [1–4], a system that can truly understand and identify poetry remains elusive. One of the main difficulties in achieving such a system is that natural language is highly ambiguous and, hence, can encompass a very large set of meaning permutations for a given text. Furthermore, the exact interpretations that humans assign to poems depend on their individual experiences and settings. In this study, we introduce the poem recognition as a branch of text document classification, which has been widely overlooked by the research community. This work is not concerned with developing algorithms that understand and interpret poetic texts, but algorithms that distinguish between poem and prose, or in a more general sense, between poem and non-poem. In addition, we will not make any distinction between classical poem (presence of rhyme) and free verse (absence of rhyme).

Poetry is a complex literary medium of many forms that is usually held in a different regard than "common" prose text, even though there is not always a clear distinction separating the two. Consider that poetry is defined by human interpretation, then the task of poem recognition becomes not just modeling objective boundary functions, but modeling human interpretation and understanding. In particular, poetry commonly employs literary devices such as metaphor and imagery that require interpretation and evoke thoughts and emotions in human readers that are not immediately obvious from the surface of the words [5].

The purpose of this work is to emphasize the significance of poem recognition within the text document classification. However, one may find meaningful applications for poem recognition as well. For instance, searching for poems in large databases or in the World Wide Web could help researchers to conduct their work in a more efficient way. Currently, the search through the Internet for poetic content is limited to finding keywords in HTML documents, without any specific considerations of structural or contextual properties of a poem. Consider the situation in which one is interested to find information about "T.S. Eliot love". Obtaining web pages which contain these words

H. R. Tizhoosh (✉) · R. A. Dara
Pattern Analysis and Machine Intelligence Lab,
University of Waterloo, 200 University Avenue West,
Waterloo, ON, Canada N2L 3G1
e-mail: tizhoosh@uwaterloo.ca

R. A. Dara
e-mail: rdara@uwaterloo.ca

may not be the main interest of the user. The main concern may be finding web pages that contain poems by T.S. Eliot, which are in the context of "love". Current state of the art does not provide such a search functionality. In addition to above applications, research on poem recognition may bring us new knowledge about this highly intellectual activity from a machine learning perspective. Investigation on poem recognition will illuminate human ability to understand them, and hence, develop more capable algorithms for imitating this ability.

Text categorization, the assignment of text documents to one or more predefined categories, has been used to classify news [6], find interesting information on the web [7, 8], and real-time sorting of email [9]. A number of statistical classification and machine learning techniques have been applied to text categorization, including regression models [10], *k*-nearest neighbor classifiers [11], bayesian classifiers [12, 13], decision trees [14], and neural networks [15, 16]. Most of the previous works in text document classification have used highly syntactical approaches or rule based approaches. Poem recognition can make use of some of the different syntactical aspects of poems but must rely primarily on identifying the lyrical nature of poetry and its differences from ordinary prose. This knowledge has not been used yet. We suggest that consideration of genre specific features (rhyme, meter, rhythm, and many other) would certainly increase the classification accuracy of the existing techniques and the recognition accuracy, even if we can successfully classify some poems using general text classification procedures by extracting common context-independent features.

As previously discussed, the use of machine learning methods in poetry recognition has been overlooked. This study is a stepping-stone and an attempt toward this direction, which can build to develop an effective and efficient tools for poem classification. To achieve this goal, we had to answer two important questions, (1) what are the most effective poetic features, and how can they be extracted?, and (2) what machine learning methods can be used for poem recognition?

The main contribution of this study is to introduce and extract such features, and to investigate their effectiveness to classify text data into poetry or prose using machine learning techniques. We selected three popular text classification methods including decision tree and naive bayesian. Empirical findings of several experiments suggest that the proposed poetic features boost classification accuracy of the selected machine learning algorithms.

This paper is organized as follows: in Sect. 2, the related works are briefly discussed. A general overview of poetry definitions is provided in Sect. 3. The first proposed approach to poetry recognition, using a measure of poetry, is introduced in Sect. 4. Two other approaches, namely bayesian classifier and decision tress, along with experimental results for different case studies are presented in Sects. 5 and 6, respectively. In Sect. 7, results are discussed in details. Section 8 concludes the finding of this work.

2 Related works

Manurung et al. [17] discussed the difficulties of poetry generation in comparison with natural language generation and proposed a stochastic hill-climbing model for poetry generation. Their proposed method took into account the additional semantic, syntactic and lexical constraints of poetry over prose. The generated poems were successful in satisfying the above constraints. Manurung also developed an evolutionary algorithm that formulated poetry generation as a state-space search problem to satisfy the criteria of grammar, meaningfulness and poeticness [18]. The developed system was capable of generating texts based on meter constraints, but encountered difficulties in doing so with semantic constraints, when the size of the semantic target grew larger. In an attempt by Graber, a semantic parser system was created to interpret natural language texts. The data gathered from the parser was used to directly generate poetry [5]. Despite all the effort, the system was unable to perform at the expected confidence level, in which the subjects were statistically unable to determine computer authorship.

The authors were unable to find any research directly dealing with the poem recognition. This lack of study has prompted us to initiate the present work.

3 Poetry

Regardless of the uncertainty around the formal definition of poetry, it is an undeniable fact that, because of the nature of the poem, it can be easily distinguished from common prose, even in the eyes and ears of a child. The art of poetry includes emotion, description, similes, metaphors, alliteration, assonance, sounds and comparison. Several different types of poetry has been recognized in the literature [19, 20]. Among them, ballad, limerick, haiku, free verse, and sonnet are the well-known types of poetry.

- Ballad: a story is told in a verse. It tells about a dramatic event in detail. In a ballad, action is very important.

- Haiku: a short poem with seventeen syllables, written in three lines with five syllables in the first line, seven in the second, and five in the third.
- Cinquain: a five-line poem with two syllables in the first line, four in the second, six in the third, eight in the fourth, and two in the fifth.
- Villanelle: a longer poem with 19-line poem, five tercets and one quatrain at the end. Two of the lines are repeated alternately at the end of the tercets.
- Limerick: a five-line poem, which usually has a funny content. Lines 1, 2, and 5 rhyme with one another, and lines 3 and 4 rhyme with one another.
- Sonnet: this form tends to be complicated and elegant. The most popular sonnet contains three quatrains and ends with a couplet.
- Free verse: this is a modern type of poetry. Most modern poetries do not rhyme and have a set meter (see Sects. 3.1 and 3.2). However, rhythm is often important in this form. Free verse is often written in short lines.

James Reeves [21] succinctly describes two essential characteristics that, either independently or together, provide a good description of poetry. Most poems posses both (1) a compression of the language; suggesting hidden power, and (2) a strongly pictorial and acoustic quality inherent in their rhythm. In addition to these two elements, a more practical approach is to consider poem's shape, either regular or irregular. All of these together provide three very useful classification insights with which a potential set of features can be derived. In short, a poem is characterized by its patterns of rhyme, rhythm, shape, and meaning. In the following sections, various features for each of the three main categories are discussed.

3.1 Rhyme

Sound patterns, hereafter called alliteration and rhyme, are two useful measures of prosody. Generally, poetry places emphasis on rhyming. Rhyme means sounds agree, and more often refers to end rhymes (words at the end of a line). It is understood that this definition of poetry is quite relative to what is considered poetry. Alliteration refers to a similar beginning consonant for a given word, while rhyme refers to the similar ending vowel and consonants. One excellent example of a rhyme pattern is the end-of-line-rhyme common to many classic poems. One simple way to calculate *rhymingness* λ is

$$\mu_{\text{rhyme}} = \frac{\sum_l n_{\text{rhyme}}}{n_p}, \tag{1}$$

where all the rhymes n_{rhyme} of all lines n_p are matched to see if they repeat [22]. The sum of all the rhymes that match in this way is divided by the number of lines to give a number between 0 and 1.

Rhyme, alliteration and within-word sound patterns should also be measured throughout the text (not just at the end of a line) to capture the poetry of non-traditional works. Prosodic patterns still exist in these poems, though there structure tends to be less rigid. One method of finding patterns in a text is to compare similarly placed sound groupings between adjacent words. Though it is in a sonnet, an excellent example is the line ''This fabulous shadow only the sea keeps'' of Hart Crane's ''At Melville's Tomb''. Several consonant and vowel patterns are evident. The matches between 'this' and 'fabulous', between 'shadow' and 'only', and between 'sea' and 'keeps', are important to capture in order to distinguish poetic prosody.

The ability to distinguish text that rhymes from text that does not rhyme is vital to poem recognition. The last few letters of the words can be simply checked in order to see if they match. If they do, then the words probably rhyme, if they do not the words may or may not rhyme. For example, by checking that the last few letters of the words match, it can be determined that the words, 'checking' and 'determining' rhyme, since they both end in 'ing'. Unfortunately, there is a wide range of words that rhyme and do not have similar endings. For example, the words 'days' and 'haze' rhyme but the last pair of letters in each word is different. To understand whether or not two words, whose last few letters do not match, rhyme, it is necessary to be aware of the sound created by each word. The simplest solution to this is to use a syllable dictionary. A syllable dictionary is a set of words whose phonetic sounds have been computed and stored. Returning to the example of 'days' and 'haze' described above, it can be noticed that both words have the 'eh' (days and haze) sound. Suppose that the dictionary represents this sound with the letter 'e' and represents the 'z' sound at the end of each word by the letter 'z'. Now assuming that the dictionary has already stored the phonetic sounds of 'day' and 'haze' (as, 'de' and 'hez'), both words can be looked up to see if they have similar ending sounds. In the case of 'days', it can be noticed that the word has an 's' at the end and is therefore plural. As a result, by adding a 'z' to the end of the sound of 'day', 'de' becomes 'dez', hence the two words rhyme.

3.2 Rhythm

A distinguishable feature of poetry is the *meter* structure. When we speak, we put stress in different parts of the words. Poems are made of number of syllables. The more consistent and/or frequent the pattern, the more poetically prosodic a text is. However, it is not trivial to determine the number of syllables per word. There are several different ways to determine the number of syllables in words as outlined by [23]. The dictionary used to store the phonetics of words can also be used to store the number of syllables per word which can later be retrieved.

Poetry in English is often made up of poetic units or feet. The most common feet are the iamb, the trochee, the anapest, and the dactyl. Each foot has one stress [24]. For example, iambic is marked by a set of strong-weak pairs of syllable stress, feet. An example of how to measure the iambic meter *IM* of a text is given in the following calculation:

$$\mathrm{IM} = \left(1 - \frac{\sigma_{\text{feet-per-line}}}{\mu_{\text{feet-per-line}}}\right) \times \frac{\sum \mathrm{Feet}_{\text{iambic}}}{\mathrm{Feet}_{\text{total}}}, \tag{2}$$

where $\mu_{\text{feet-per-line}}$, and $\sigma_{\text{feet-per-line}}$ measure the mean and standard deviation of feet in a given line, $\mathrm{Feet}_{\text{iambic}}$ is the number of iambic feet in a given line, and $\mathrm{Feet}_{\text{total}}$ is the total number of feet. This calculation is fairly ad-hoc, and other variations might prove more effective. It is also possible to measure the similarity of text meter to a given set of pattern templates common to different types of poetry. Haiku, for example, could be well characterized by such a scheme.

3.3 Shape

Structural features (Shape) can be employed to classify poems as well. Shape is a valuable and simple measure, simply because the lines of poems are often much shorter than those of prose. Shape features are also useful for recognizing many contemporary poets, who intentionally employ line breaks and spacing that is inconsistent with the normal flow of the text. In addition, recognizing line grouping can be helpful for some of the more traditional styles of poetry (such as a sonnet) which will often include line breaks at regular intervals to form stanzas (paragraphs). A feature of interest is line length patterns. By counting the number of words (or characters) per line, and then testing the entire set of lengths for inconsistencies (i.e., high standard deviation) some information about the change in shape can be extracted. Finding specific instances of long-short pairs may also be of interest. Moreover, pre-line white space may be a valuable measure. If we search for the same sorts of irregularities (or patterns of irregularity) as with line length, a pattern may emerge that can be characterized as poetic. Another measure of shape common to many poems is the existence and consistency of stanzas. The number of stanzas is measured by the number of groups of two or more lines separated by a blank line. Though it is possible for this format to exist in prose, it is far more common in poems. One major problem with shape is that it can be eliminated by text formatting and/or reformatting.

3.4 Meaning

A poem has line breaks at somewhat regular intervals, but the variety in line length is minimal, and it does not have any abnormal white space. More importantly, there are no noticeable prosodic patterns of either rhyme or rhythm. A nearly intangible quality inherent in almost any poem is the image provided by words choice and phrasing. This quality is very difficult to capture, and yet, would provide the clearest measure of poetry available. There are various ways to extract features which represents this quality. One potential way to measure intense imagery may be to calculate the number of content words per phrase. This content intensity is calculated by counting the nouns, verbs, adjectives, and adverbs in each phrase and dividing by the number of total words in that phrase. An average of such calculations might provide the only measure of intensity available without access to the actual meaning of the words. Phrase repetition, though also employed in some prose, is well used in many poetic texts.

We conducted an extensive background research on different types of poems and various aspects of poem which can be beneficial in distinguishing between poem and prose. In this section (Sect. 3), we gave an overview of various poetic features. In the following sections, we focus on the methods that can be used to extract these features, and applied machine learning algorithms for poem recognition. Since considering all the aforementioned poetic features were beyond the scope of this paper, we left some of these features for future study. In the following, we present one preliminary experiment, and four different case studies. Each case study is concerned with a different set of features or/and different machine learning technique. More detailed explanation of these features and classifiers is highlighted in the related section.

4 A measure of poetry: preliminary experiment

Our goal, in this preliminary experiment, was to introduce and propose a set of simple shape features. We were also interested to evaluate their efficiency in distinguishing prose from poem. We define a measure of poetry μ_{poetry}, with which we can distinguish poetry from prose. Since there are many ambiguous cases, where a clear decision cannot be made, it would be more beneficial to define the measure of poetry as a fuzzy measure $\mu_{\text{poetry}} \in [0, 1]$. Following features were considered to measure the total poetic value of a text document:

- n_p: Number of lines will indicate the text length. Generally, poems should have a smaller number of lines compared to prose.
- $\overline{n}$: Average number of words per line in a poem is usually small as well.
- n_{rhyme}: Number of words that rhyme is a strong feature for recognition of classic poems and some free verses.
- $w_i \in [0, 1]$: Poetic weight or word value is a number representing the poetic significance of every word. This would be a crucial feature in recognizing poems in absence of rhymes.
- n^*: Number of words with $w_i \neq 0$ will indicate how intensively poetic words have been employed.

In order to define a measure of poetry, we first need to establish a thorough understanding about these features. For instance, if the ratio of the number of lines n_p, and the average number of words/line$\overline{n}$ is low, then we can cautiously conclude that the document is *not text*. Therefore, we define the measure $\mu_{\text{not-text}}$ as

$$\mu_{\text{not-text}} = \frac{n_p}{\overline{n}}. \tag{3}$$

If we solely consider the number of rhymes n_{rhyme} with respect to the average number of lines n_p, then a second measure μ_{rhyme} can be defined

$$\mu_{\text{rhyme}} = \frac{n_{\text{rhyme}}}{n_p}. \tag{4}$$

Lastly, one could determine the frequency or intensity of words with high poetic values in the document by

$$\mu_{\text{poetic-words}} = \frac{\sum_{n^*} w_i}{n^*}. \tag{5}$$

Obviously, these simple measures can classify some simple cases. However, in order to introduce a reliable measure, a fuzzy rule can be employed to aggregate these measures and estimate a more general measure of poetry, μ_{poetry}. A simple rule could be formulated as:

IF the document is *not text* **AND** contains *rhymes* **AND** has *poetic words*, **THEN** its measure of poetry is *high*.

Using product operator, the measure can be calculated as follows:

$$\mu_{\text{poetry}} = \frac{n_p}{\overline{n}} \times \frac{2n_{\text{rhyme}}}{n_p} \times \frac{\sum_{n^*} w_i}{n^*} = \frac{2n_{\text{rhyme}} \sum_{n^*} w_i}{\overline{n}n^*}. \tag{6}$$

To represent the logical AND, we use the product operator instead of minimum operator mostly because it is more pessimistic. For the sake of normalization, the measure can be given as

$$\mu_{\text{poetry}} = \min\left(1, \frac{2n_{\text{rhyme}} \sum_{n^*} w_i}{\overline{n}n^*}\right). \tag{7}$$

The poetic word value, as shown in Table 1, can be determined in different ways. One can find the relative frequency of all words in a sufficiently large set of poems, and assign words with high frequency a high poetic value. Post-processing of these values by an expert is not necessary, but would be beneficial. An alternative method is to assign weights using subjective knowledge of several experts, as we have done in this table.

In order to test the reliability of the measure of poetry, three text documents (prose) and seven different poems were used. Table 2 summarizes the classification results. As it can be seen, with exception of the last case, the classification results by the measure match the true document type, which shows that the proposed poetry measure is a promising method. The last poem, which has a measure of poetry of 0, however, is misclassified for two reasons. First the poem consisted of only three lines, and second, it did not contain any rhymes or words with poetic values as listed in Table 1.

Table 1 Examples for subjective poetic weights, averaged over several experts

Word	Star	Twilight	Night	Light	Love	Feel	House	Blood	War
w_i	0.92	0.89	0.85	0.84	0.74	0.73	0.07	0.01	0.01

Table 2 Poem recognition using the measure of poetry (FV = free verse)

Document	Description	$\bar{n}$	n^*	n_{rhyme}	$\sum_{n^*} w_i$	Poetic (%)
News	CBC headline news	27	3	2	1.7	8
Report	Scientific subject	36	12	3	3.4	4
Health	Guidelines for AIDS	22	9	3	5.3	16
Spleen	Poem (T.S. Eliot)	5	8	11	5.9	100
Snow flakes	Poem (E. Dickinson)	6	3	4	2.1	93
Rubaee	Poem by Khayyam	7	5	4	4.2	96
Clenched soul	FV (Pablo Neruda)	7	14	6	8.8	100
On night	FV (A. Shamloo)	7	12	6	9.1	100
Gypsy melody	FV (L. Hughes)	3	4	2	2.6	86
Green memory	FV (L. Hughes)	3	0	0	0	0

In order to cope with similar cases, which are mainly common in modern poetry, we can extend the measure of poetry as follows:

$$\mu_{\text{poetry}} = \min\left(1, \frac{n_p}{\bar{n}} \times \max\left(\frac{2n_{\text{rhyme}}}{n_p}, \frac{\sum_{n^*} w_i}{n^*}\right)\right). \tag{8}$$

The new measure is the numerical evaluation of the following rule: **IF** the document is *not text* **AND** (has *rhymes* **OR** contains *poetic words*), **THEN** its measure of poetry is *high*.

The new measure is less sensitive. However, if there are no rhymes (classical poems) and no poetic words (free verse mainly), then the new measure will fail as well. This problem can generally occur if the poem is extremely short. A more comprehensive fuzzy inference systems may reduce the text-length and the rhyme-absence sensitivity. However, the problem will remain challenging for extreme cases.

5 Bayesian approach to poem recognition

Bayesian classifiers [25, 26] are popular techniques for text document classification. These classifiers use conditional probabilities to determine the most probable classification of data given a set of probabilities gathered from training data. The most popular form of Bayesian techniques, which has been extensively applied to text classification, is naive Bayesian [13, 27]. This classifier is based on the assumption that a set of conditionally independent attributes will occur with the same conditional probabilities in the training set as they will in other data sets. Therefore given a set of conditionally independent attributes, a_1, a_2, a_3, ..., a_n, the probability of their occurrence in a target instance is given as:

$$P(a_1, a_2, \ldots, a_n | c_j) = \prod_{i=1}^{n} P(a_i | c_j). \tag{9}$$

The naive Bayes classifier assigns the test instance to class c as:

$$c = \mathbf{argmax} \quad P(c_j) \prod_{i=1}^{n} P(a_i | c_j), \tag{10}$$

where $P(c_j)$ and $P(a_i \mid c_j)$ are the priori probability of class c_j and a posteriori probability of an instance a_i given class c_j, respectively.

In the following subsections two case studies for poem recognition with naive Bayesian approach are described.

5.1 Case study #1

In this study, several poetic measures have been employed to distinguish poetry from ordinary prose. Several individual and combined Naive Bayesian classifiers were applied for classification. Total of 403 instances were obtained from different sources. Some of the poems were obtained from Project Gutenberg [28]. In addition, several ordinary prose were selected from online magazines and Web sites. Overall, 100 poetic and 100 non-poetic sample documents were used for training. In addition, 106 samples of non-poetic and 97 samples of poetic data were used to validate the classification accuracy.

Three different types of features were analyzed to determine which would be the best differentiator between poetry and ordinary prose. The first feature we used was rhyme (see Sect. 3.1).

$$\mu_{\text{rhyme}} = \frac{n_{\text{rhyme}}}{n_p}, \tag{11}$$

where n_{rhyme} represents the number of words that rhyme and n_p is the number of lines. It should be noted that text can also rhyme accidentally; therefore, the amount of rhyming in text is important.

The second feature used in this case study was meter (Rhythm) (see Sect. 3.2). This feature was best represented by looking at the number of words used in a line, the number of syllables used in a line, the number of words used throughout the poem, and the number of syllables used throughout the poem.

$$\mu_{\text{rhythm}} = \frac{\sum_L n_{\text{word}} \times \sum_L n_{\text{syll.line}}}{n_{\text{w}} \times n_{\text{s}}}, \tag{12}$$

where $n_{\text{word.line}}$ is the number or words in a line, $n_{\text{syll.line}}$ is the number of syllables in a line, n_{w} is the number of word in poem or prose, and n_{s} is the number of syllables in a poem or prose.

The last set of features that were used in this case study falls under Meaning category (see Sect. 3.4). We considered the difference in the frequency of adjectives in poetry as compared to their frequency in ordinary prose.

$$\mu_{\text{adjective}} = \frac{n_{\text{adjective}}}{n_{\text{w}}}, \tag{13}$$

where $n_{\text{adjective}}$ is the number of adjectives in a poem or prose. This feature was created in an attempt to capture the visual nature of the poems and help identify free verse poetry which would escape the rhyme and meter structure classifiers.

The above mentioned features were extracted from the training data. Then, using these features, we adopted two different approaches to poem recognition. In one approach, we combined two or more features to a single training data and trained an individual classifier using this training data. In another approach, three distinct naive bayesian classifiers were built, namely Rhyme (C_{rhyme}), Rhythm (C_{rhythm}), and Adjective ($C_{\text{adjective}}$) classifiers. Our objective for building these distinct classifiers was because of the following observation. As it can be seen in Eqs. 9 and 10, the significance of individual attributes can be overwhelmed when a classifier contains a large number of attributes. This was a problem given the feature set that we have chosen because important attributes, such as rhyme scheme, could have been weighed down by having numerous other attributes within the same classifier. Suppose that a naive bayesian method tried to use the rhyming scheme of a poem and the words of a poem. The large number of word attributes, within the classifier, would obscure the effects of rhyming. Thus, ordinary text may be identified as poetry simply because it caters to the common word probabilities of poetry. For this reason, a set of separated classifiers, C_{rhyme}, C_{rhythm}, $C_{\text{adjective}}$, was used for each of the different feature sets. The final classifier was built via fusion of these individual classifiers using linear combination method:

$$C_{\text{final}} = \alpha C_{\text{rhyme}} + \beta C_{\text{rhythm}} + \zeta C_{\text{adjective}}. \tag{14}$$

In this experiment, the weights, α, β, and ζ, had a value of either 0 or 1, depending on the objective. This is in fact a simple attempt to evaluate the effectiveness of each feature individually, and at the same time, avoid heavy learning computations of multiple classifiers when combining the features. This method is similar to the one used by Kelemen et al. [29].

We first evaluated extracted poetic features collectively, by combining two or more features building a single training data. Table 1 summarizes the list of combined features and the performance of resultant classifiers. We report the performance in terms of classification accuracy and F-measure. Accuracy is the number of correctly classified documents:

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N} \times 100, \tag{15}$$

where N is the total number of documents to be classified (in this study poem and prose), and N_{correct} is the number of correctly classified documents. F-measure is a popular metric that is used in determining how well a classifier performs on a given set of data [30]. This measure corresponds to the size of correctly classified document and the size of document that has been classified, normalized by the cumulated size of the whole data being considered. Fmeasure is calculated as follows. For each class of prose or poem (i = 1 or 2), let D_i be the number of documents whose real label is i, and B_i the number of documents whose label is assigned to be i, and C_i the number of correctly predicted examples in this class. The precision and recall of the class i are denoted as $P_i = C_i/B_i$ and $R_i = C_i/D_i$.

Fmeasure is used to combine precision and recall into a unified measure. It calculates the total precision and recall on all classes $P = \sum C_i / \sum B_i$ and $R = \sum C_i / \sum D_i$. The Fmeasure is estimated as:

$$\text{Fmeasure} = \frac{2P \times R}{P + R}. \tag{16}$$

From Table 3, one observation is that the classifier with the highest accuracy is the rhythm classifier which measures the meter structure of the text. The rhyme classifier also performed well, however, the adjective classifier illustrated that this measure is clearly not a good one for poetry. From Table 3, we notice that the use of a adjective classifier is ineffective at distinguishing poetry from ordinary text. The idea behind the adjective classifier is that it would provide insight into the descriptive nature of poetry. It failed to do this principally, since there is no differentiating adjective in the descriptions. In other words, descriptive text does

Table 3 Results of individual classifiers

Classifiers	Accuracy (%)	Fmeasure (%)
Rhyme classifier	71.92	76.12
Rhythm classifier	79.31	74.87
Adjective classifier	46.79	38.87
Rhyme and rhythm classifier	79.80	75.20

not contain more of one type of word than other abstract or non-descriptive text. Another observation is that we can see that the rhyme classifier did substantially better. In fact, the rhyme classifier in this system correctly identified every test poem that had a rhyme scheme associated with it. Those that did not have a rhyming scheme were classified as ordinary prose. The ordinary texts that were identified as poetry contained a coincidentally high level of rhyming often a combination of all three of the different rhyming schemes identified by the classifier: aabb, abab and abba. In contrast to the rhyme classifier, the rhythm classifier, which identifies the meter structure of text by counting the syllables and words per line, correctly classified 95% of poems and incorrectly classified several non-poetic texts as poetry. The instances of ordinary text that were identified as poetry were classified as such for one of two reasons. The first is that their meter structure accidentally resembled the meter structure of poetry. An example of such text is the comments from an internet chat site. The second is that some of their meter structure had not been part of the training examples; because of this, the classifier had no probabilistic determination for these instances and classified them according to the instances which it could identify.

Table 4 shows the results for combining various classifiers using Eq. 14. In each case, the constant multiplier of each classifier was 1 or 0, depending on the type of classifier we were interested in.

As it can be seen, the best results were obtained by combining the rhyme classifier and the rhythm classifier using Eq. 14. The most interesting comparison is between the classifier which combines rhyme and rhythm (meter structure) and the use of two separate rhyme and rhythm classifiers. For the combined classifier, the accuracy was close to the accuracy of the rhythm classifier (92.11%). This was because, as mentioned earlier, the rhythm classifier generates more tokens than the rhyme classifier, specifically, when measuring ordinary prose in which there was no predefined meter structure. These tokens overshadowed the rhyme tokens in the classifier and caused it to behave much like a rhythm-only classifier; especially when attempting to classify non-poetic text, where rhythm classifier was particularly underperforming. The comparison between individual rhyme and rhythm classifiers and their weighted combination is drastic. Combining the classifiers using Eq. 14 resulted in a 12% increase in accuracy. On the other hand, combination of classifiers using the adjective classifier was not effective, since the adjective classifier was not able to classify poetic or non-poetic text accurately.

In addition to the discussed experiments, the adjective and the word classifier were introduced into the system to help identify non rhyming free verse poetry, which the rhyme and rhythm classifiers fail to correctly identify. However, both failed at the task. The only way to truly distinguish free verse poetry from ordinary prose is to have some understanding of the meanings of the words, or use a measure of poetry as introduced in Sect. 4.

For this case study, it can be concluded that the best results were obtained using the rhyme and rhythm classifiers. By definition poetry distinguishes itself from ordinary prose through these two techniques, with the exception of free verse poetry. Developed features were poor at recognizing free verse poetry that have no rhyming or use long meter units.

Table 4 Results for classifier combination

Classifiers	Accuracy (%)	Fmeasure (%)
Rhyme and rhythm	92.11	89.52
Rhyme and adjective	58.62	55.11
Rhythm and adjective	78.81	75.75
Rhyme and rhythm and adjective	80.29	81.58

5.2 Case study #2

In this case study, a naive bayesian classifier was used to predict whether randomly selected input texts were poetry or prose. In one set of experiments, the classifier was trained over eight classes; seven representing different types of poems selected for this study, and one representing all prose in general. Then, recognition capability of each feature was examined for different classes. In another set of experiment, all different type of poems considered as one class and the classifier was trained with two classes, poetry and prose.

Nine features were considered in total. These features were:

- total number of lines of text n_p,
- total number of words w_{total},
- average number of words per line $\mu_{\text{word.line}}$,
- average number of syllables per line μ_{syll},
- total number of paragraphs/stanzas $\text{stan}_{\text{total}}$,
- average number of lines per paragraph/stanza $\mu_{\text{line.stan}}$,
- average number of words per paragraph/stanza $\mu_{\text{word.stan}}$,
- average number of alliterations per line $\mu_{\text{all.lines}}$,
- average number of rhymes per line $\mu_{\text{rhyme.lines}}$.

The syllable counting component was written based on general rules for how English phonemes combine to produce syllables, as well as empirical observations made during the development of this component. The KTEXT morphological parser [31] was invoked to evaluate the number of syllables in words. This document parser takes ASCII input text files with corresponding class labels, and outputs a formatted text file containing the extracted feature values of all inputs. The number of syllables containing the ‘ea’ vowel was also calculated due to the high degree of variability in the number of syllables produced by this combination. The number of alliterations was measured for a particular line by incrementing the count for each word that started with the same letter of a word that has already been seen on that line. In the case of rhyme type feature, only full rhymes, half rhymes, eye rhymes, identical rhymes, and scarce rhymes were counted for the last word of each line by the rhyme detector [31].

The collected dataset consisted of 60 examples of prose taken from various Internet sources having different lengths, content, and paragraph sizes, and 210 examples of poems, 30 samples for each of 7 types of poems represented epics, haikus, limericks, sonnets, sestinas, villanelles, and free-verse. The dataset was first divided into training and test sets. Furthermore, training data was divided into smaller sets to investigate the effect of training sample size.

To examine the effect of extracted features on the classifier, the normal probability density function (PDF), used for estimating posterior probability of each class, was estimated and examined against the features. As an example, the PDF of all classes versus the average number of syllables per line is depicted in Fig. 1.

Comparing the PDFs of seven poetry classes to that of prose, PDF of prose is relatively flat, indicating that the average number of syllables per line for prose text is highly variable. Similar verification procedures were carried out on the other nine features. The rest of the results are not highlighted in this paper. However,

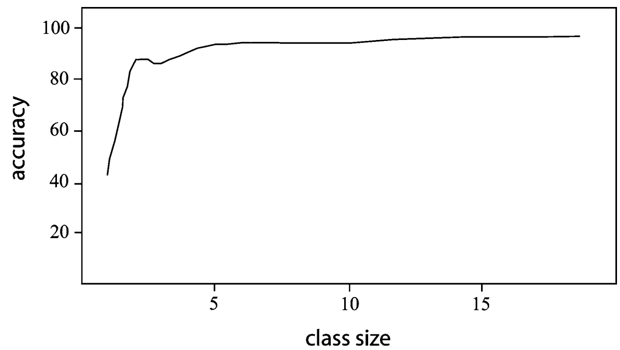


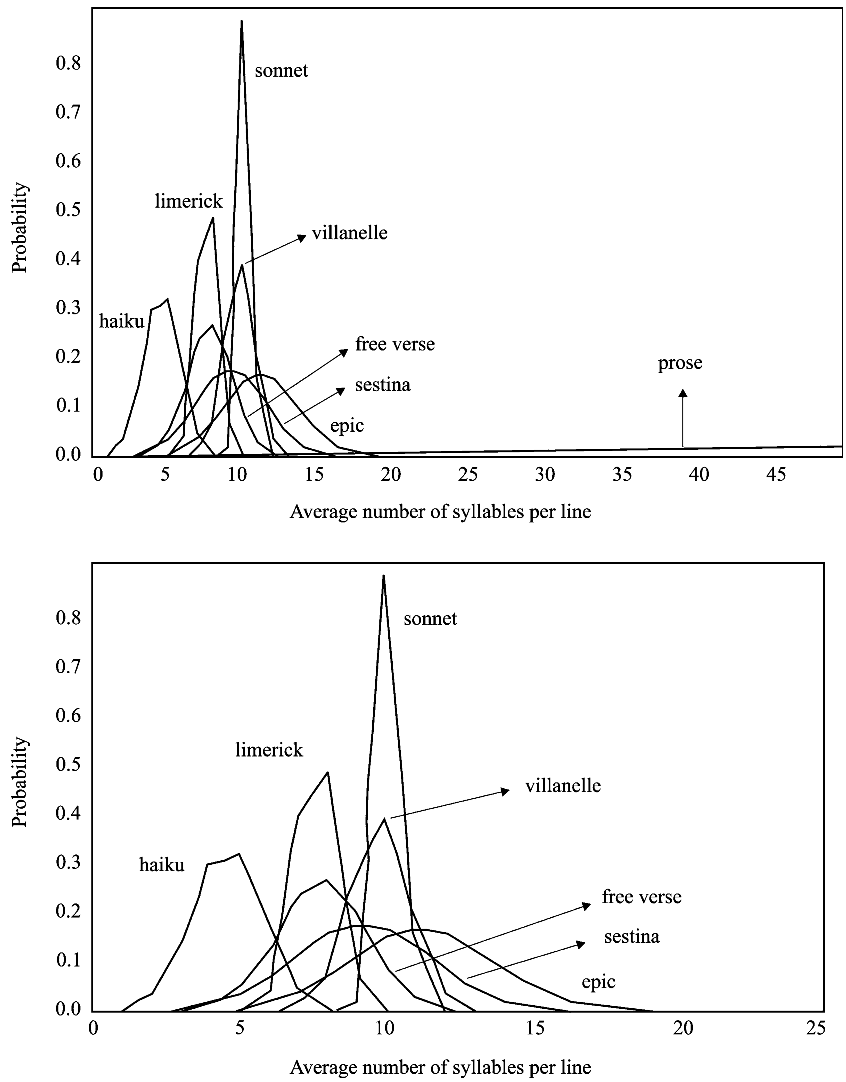
Fig. 1 *Top figure* PDF of all eight classes using average number of syllables, *bottom figure* identical to the top figure, except that we have zoomed in on PDF of different types of poems

through similar investigation, it was observed that the most distinguishing features of among the nine were the average number of syllables per line ($\mu_{\text{syll.line}}$) and the average number of rhymes per line ($\mu_{\text{rhyme.line}}$). The least distinguishing feature was the average number of alliterations per line ($\mu_{\text{all.line}}$).

We were also interested to investigate the benefit of using extracted features to distinguish between poetry and prose. Despite the low posterior probabilities that were estimated for prose, in top Fig. 1 notice that outside the main bell-shaped regions of the poetry PDF curves, the probability drop to near zero. Therefore, it is expected that the probability for different poetry types does not always dominate the naive bayes classifier with this formulation. Furthermore, since the probability estimated for prose was low with respect to the probability for the different types of poetry, a high classification accuracy for the naive bayesian classifier was expected. Based on this observation, all the features were combined and used for prediction of the unseen text data. The results are presented in Fig. 2. This figure illustrates the accuracy of the system with respect to training data size. It is important to note that x axis represents the number of classes in each training data. For each of the training set sizes, the accuracy rate was averaged over number of trails to minimize fluctuation that had occurred as the result of random selection of the training data.

Using nine extracted features, the naive bayes classifier was able to perform very well on the test data. With the increase of the training patterns, accuracy of the classifier improved as well. Using 20 patterns for each class, 160 instances for training and 80 for testing, the system was able to achieve an accuracy of 96.50% on the test data. Considering the training data as small as 160 documents, this high accuracy suggests that selected poetic features were very effective.

Fig. 2 Accuracy versus class size in the training data



6 Poem recognition with decision trees

Decision tree classifiers [14, 25] have been broadly used in various applications of pattern recognition and artificial intelligence. Decision tree can be used for classification either in deterministic or incomplete domains. The basic idea underlying decision tree systems is the iterative splitting of the feature space into regions, until a region satisfies a certain stopping criterion. Then, this region is assigned to one decision class and, subsequently, any new unlabeled observation that belongs to the same region is classified to the region's decision class.

The major key to the success of a decision tree algorithm depends on the criterion used for selecting the attribute of splitting criterion. If the attribute is a strong indicator of the class membership of a sample, it should appear as early in the tree as possible. Most decision tree learning algorithms use a heuristic for estimating the best attribute.

6.1 Case study #3

In this case study, the Festival text-to-speech system is employed to extract basic attributes [32]. Several measures of poetry in the categories of shape, rhythm, and meaning are combined to distinguish poem from prose, using decision trees. A data set containing 69 non-poem and 128 poems was obtained through the Internet and other sources. This data set has a diverse selection of different topics and poems including short stories, novels, recipeprunings, quotations, essays, verses, and free verses.

In order to extract the basic features, a freeware development and research tool, the Festival Text-to-Speech Synthesis System, was used [32]. This system provides a large number of useful functions and customizable algorithms for extracting various text features. Though chiefly designed for text-to-speech research, Festival is able to extract, in its default configuration, five distinct text structures: tokens (graphemic unit), words (linguistic unit), syllables and segments (phonetic units), and phrases (prosodic unit). When Festival is called, it passes a text file which is first cut, or ‘‘chunked‘‘ into utterances. These utterances are then passed through a series of functions which systematically transform the bare/plain text into linguistically tagged data. The steps of the procedure are as follows: first, the utterance is ‘‘tokenized’’; all the white spaces are distinguished from groups of characters. Then, those tokens are distinguished as words, and assigned part-of-speech tags. Next, syllables and segments (phonemes) are created for each word using a dictionary lookup and letter-to-sound (LTS) rules. Finally, phrase breaks are identified and labeled.

With a reliable method for extracting base features in place, the next, and most important step is extracting a useful set of features which capture the form, semantics, and prosody as discussed above. Some of the basic features are line length, phrase length, white space, word value, word part-of-speech, syllable rhyme, and syllable stress. The following features were extracted for each of the three main categories:

- n_p,
- number of phrases p,
- stanzas,
- mean of the number of characters per line $\mu_{\text{ch.line}}$,
- standard deviation of the number of characters per line $\sigma_{\text{ch.line}}$,
- $\sigma_{\text{ch.line}}$ divided $\mu_{\text{ch.line}}$,
- mean of number of lines per stanza $\mu_{\text{stan.line}}$,
- standard deviation of number of lines per stanza $\sigma_{\text{stan.line}}$,
- $\sigma_{\text{stan.line}}$ divided by $\mu_{\text{stan.line}}$,
- a sum of percentages of the number of similar sounds in groups of 2, 3, 4, 5, and 6 words, $\sum_{\text{words}}$ word.rhyme,
- a sum of percentages of the number of lines with similar final rhymes, $\sum_{\text{lines}}$ line.rhyme,
- sample mean of the similarity between a given line and an ideal iambic line, $\mu_{\text{iambic-line}}$,
- a sum of percentages of the number of lines that have the same number of feet, $\sum_{\text{lines}}$Feet,
- mean of the number of feet in each line, $\mu_{\text{feet-per-line}}$,
- standard deviation of the number of feet per line, $\sigma_{\text{feet-per-line}}$,
- number of phrases per line p_{line},
- mean of the number of content words (nouns, verbs, adjectives, adverbs) in a given phrase, μ_{word}.

In this case study, the extracted features were used to train a decision tree classifier. The data set was first divided into training and validation sets. Then, the training data was used to train the decision tree classifier using different pruning techniques. We used Fisher, Information Gain, and Purity pruning methods [33]. The error rates for Test and training data are summarized in Tables 5 and 6, respectively. The error rates presented in these tables are averaged over five trails.

By looking at the results in Tables 5 and 6, the first observation is that pruning is generally beneficial. We can easily observe the effect of overfitting in the results. For each pruning technique, there is an optimal point where the effect of overfitting or underfitting is eliminated and classification accuracy is at its best. Overfitting is defined as follows: the risk that an algorithm learns a model that fits the training data well but captures the underlying distribution of the problem poorly. The best classification has been achieved with the pruning sizes around 3–6. Another observation is

Table 5 Accuracy using test data with respect to the degree of pruning

Criterion	Degree of pruning											
	Sparse	0	1	2	3	4	5	6	7	8	9	10
Fisher	89.1	91.2	90.5	88.8	87.8	90.5	88.8	87.8	87.4	87.9	88.1	84.9
Info gain	88.6	89.0	87.9	88.3	90.3	86.8	87.9	88.6	89.1	87.2	89.1	86.1
Purity	91.9	93.4	92.2	93.6	94.0	93.3	93.4	93.0	92.2	91.4	92.3	93.1

Table 6 Accuracy using training data with respect to the degree of pruning

Criterion	Degree of pruning											
	Sparse	0	1	2	3	4	5	6	7	8	9	10
Fisher	97.0	100.0	99.6	98.5	97.2	94.5	93.4	91.2	91.8	90.9	89.3	87.4
Info gain	98.6	100.0	99.7	99.1	97.6	96.2	93.9	93.8	92.6	91.8	92.2	90.0
Purity	98.3	100.0	100.0	99.8	99.3	98.6	99.1	98.8	98.1	96.8	97.4	97.3

that the Purity pruning method resulted in better performance. Best results on test data has been obtained using Purity method (94%) and the pruning degree of 3, and the worst has been obtained Fisher method (84.9%). Furthermore, with these results, the effectiveness of the proposed poetic features is validated, since we have used only 90 documents for training from such a diverse collection of topics.

7 Overview of results

The four case studies conducted in this work are summarized in Table 7. All the approaches based on measure of poetry (MP), naive bayseian case studies (NB#1 and NB#2), and a decision tree (DT#1) are listed with the total number of samples used (N_T), and the achieved accuracies.

Table 7 does not intend to compare the techniques but just provides an overview of achieved accuracies. The primary concern of this paper was to demonstrate that poem recognition, as a sub-class of document classification, is possible. Table 7 illustrates this by showing high accuracies for different case studies. Which techniques will be the ultimate choice for the poem recognition remains subject to future investigations. Additional experiments, larger and unique data sets, additional poetic features as well as machine learning techniques are required to reach a reliable conclusions. However, the following observations can be made for each of the case studies:

- Measure of poetry: Despite the small size of data used in this study, the collection of data selected were extremely diverse for both poetry and prose. The results suggest that the new measures of poetry have the potential of providing a highly accurate recognition capability. The advantage of this approach over the others is that it is independent from rhyme and shape features, which may not be always available.
- Case study #1: The largest number of data was used in this case study. Frequency of the words and syllables has shown to be the most effective feature among the other two. The second best was the rhyme feature. By combining these two features, we were able to improve the recognition capability of the system up to 12%. These findings suggest that use of word frequency by itself, which is a common text classification approach, is not sufficient for poem recognition. The use of more sophisticated features, poetic features, is required to boost the recognition capability of the classifiers. Similar observations in case studies #2 have been made.
- Case study #2: Probability density function of seven types of poems and prose were examined against each other. An interesting observation was that PDF of prose was relatively flat comparing to all types of poetry, using the average number of syllable and rhyme. The usefulness of the poetic features was again confirmed through these results. This case study suggests that with the increase of the size of training data, higher accuracy can be achieved.
- Case study #3: A large number of poetic features were extracted. The effectiveness of decision tree with different splitting and pruning parameters have been examined. Decision tree classifier has shown to be sensitive to the size of pruning (effect of underfitting and overfitting). Similar to previous case studies, the extracted features have shown to be highly effective and resulted in a high accuracy using a small size of training data.

Table 7 Overall results

Approach	N_T	Accuracy (%)
MP	10	90
NB#1	403	92.12
NB#2	240	96.50
DT#1	187	94

8 Conclusions

In this work, we laid out the stepping stone for poem recognition and its applications. We highlighted possible challenges and presented three case studies. Each of these studies were independently designed and conducted, illuminating a certain aspect of poem recognition. The motivation for this work was to investigate the feasibility of poem recognition. We were not aiming on making concrete conclusions for this work such as "which algorithm is the best" statement. Numerous additional detailed experiments are still required before we can make such statements. More importantly, a single training and validation data set which contains representative samples should be provided to all techniques to be able to make reliable comparisons.

The main purpose of this work was to initiate the research in the field of poem recognition by providing preliminary thoughts (poetic attributes and machine learning approaches), experiments and results. A measure of poetry, as introduced in Sect. 4, seems to be a significant component in a reliable solution for poem recognition. The suggested methods make the classification widely independent from rhyme presence and shape features (which cannot always be found). Further development of the notion for the measure of poetry will certainly be extremely beneficial.

9 Originality and contribution

The idea of recognizing poems, as highly intellectual texts, was first proposed by Dr. Tizhoosh. To our best knowledge, this is the first time that poem recognition has been attempted. Undergraduate students (see Acknowledgments) have contacted experiments guided by Dr. Tizhoosh, who provided the original idea along with consultations to choose algorithm and run the experiments. Mrs. Dara, PhD candidate, has participated in design of the algorithms by supervising the students and has written the large parts of this paper based on her experience in text data recognition.

Acknowledgments The authors thank the undergraduate students A. Paleshi, R. Goorah, J. Chau and N. Gilhooly for conducting the experiments as a part of requirements for the course *SYDE422 Machine Intelligence*.

References

1. Brooks C, Warren RP (1976) Understanding poetry. Rhinehart and Winston, Holt
2. Ciardi J, Williams M (1975) How does a poem mean? Houghton Mifflin Company, Boston
3. Wormser B, Cappella D (2000) Teaching the art of poetry, the moves. Lawrence Erlbaum Associates, Mahwah
4. Frye N, Baker G, Perkins G, Perkins B (1997) The Harper handbook to literature. Addison-Wesley, New York
5. Boyd-Graber J (1999) Semantic poetry creation using lexicographic and natural language texts. MASc Thesis, Arkansas School for Math and Science
6. Selamat A, Omatu S (2004) Web page feature selection and classification using neural networks. Inf Sci 158(1):69–88
7. Sebastiani F (2002) Machine learning in automated text categorization. Assoc Comput Mach Surv 34(1):1–47
8. Lai WK, Hoe KM, Tai TS, Seah MC (2002) Classifying english web pages with ‘‘smart’’ ant-like agents. In: Proceedings of the 5th conference on World Automation Congress, vol 13, pp 411–416
9. Poulos M, Papavlasopoulos S, Chrissicopoulos V (2004) A text categorization technique based on a numerical conversion of a symbolic expression and an onion layers algorithm. Digital Inf 6(1):10–12
10. Zhang T, Oles F (2001) Text categorization based on regularized linear classifiers. Inf Retrieval 4:5–31
11. Yang Y (1999) An evaluation of statistical approaches to text categorization. Inf Retrieval 1(1–2):67–88
12. Souafi-Bensafi S, Parizeau M, Lebourgeois F, Emptoz H (2002) Bayesian networks classifiers applied to documents. In: Proceedings of 16th international conference on pattern recognition, vol 1, pp 483–486
13. Scheffer T (2004) Email answering assistance by semi-supervised text classification. Intell Data Anal J 8(5):481–493
14. Quinlan JR (1986) Induction of decision trees. Mach Learn 1:81–106
15. Lee PY, Hui SC, Fong AC (2002) Neural networks for web content filtering. IEEE Intell Syst 17(5):48–57
16. Lodhi H, Karakoulas G, Shawe-Taylor J (2002) Boosting strategy for classification. Intell Data Anal J 6(2):149–174
17. Manurung HM, Ritchie G, Thompson H (2000) Towards a computational model of poetry generation. In: Proceedings of the symposium on creative and cultural aspects and applications of AI and cognitive sciences, pp 79–86
18. Manurung HM (2003) An evolutionary algorithm approach to poetry generation. PhD Thesis, University of Edinburgh
19. Pinker S (1999) Words and rules: the ingredients of language. Basic Books, New York
20. Shadow Poetry. http://www.shadowpoetry.com/
21. Reeves J (1967) Understanding poetry. Heinemann Educational Books Ltd, London
22. Scholes R (1999) Element of poetry. Oxford University Press, Toronto
23. Fudge E (1999) Words and feet. J Linguist 35(2):273–296
24. Steele T (1999) All the fun’s in how you say a thing: an explanation of meter and versification. Ohio University Press, Athens
25. Duda R, Hart P (2001) Pattern classification, 2nd edn. Wiley, New York
26. Mitchell TM (1997) Machine learning. McGraw-Hill, New York
27. McCallum A, Nigam K (1998) A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization
28. Project Gutenberg. http://www.gutenberg.net
29. Kelemen A, Zhou H, Lawhead P, Liang Y (2003) Naive Bayesian classifier for microarray data. In: Proceedings of the international joint conference on neural networks, pp 1769–1773
30. Aas K, Eikvil L (1999) Text categorisation: a survey technical report no. 941. Norwegian Computing Center. http://citeseer.nj.nec.com/
31. KTEXT software, Summer Institute of Linguistics International. http://www.sil.org/
32. Black AW, Taylor P, Caley R (2002) The festival speech synthesis system: system documentation. The Centre for Speech Technology Research, University of Edinburgh, 1.4.3 edition
33. Esposito F, Malerba D, Semeraro G (1997) A comparative analysis of methods for pruning decision trees. IEEE Trans Pattern Mach Intell 19(5):476–491

Author Biographies

Hamid R. Tizhoosh received the MS degree in electrical engineering from University of Technology, Aachen, Germany, in 1995. From 1993 to 1996, he worked at Management of Intelligent Technologies Ltd (M.I.T. GmbH), Aachen, Germany, in the area of industrial image processing. Dr. Tizhoosh received his PhD degree from University of Magdeburg, Germany, in 2000 in the field of computer vision. He was active as the scientist in the engineering department of IPS (Image Processing Systems Inc.), Markham, Canada, until 2001. For 6 months, he visited the

Knowledge/Intelligence Systems Laboratory, University of Toronto, Canada. Since September 2001, he is a faculty member at the department of Systems Design Engineering, University of Waterloo, Canada. His research encompasses machine intelligence, computer vision and soft computing.

Rozita Dara received the BSc degree in ''Applied Mathematics in Computer Science'' from Shahid Beheshti University, Tehran/Iran, in 1997. She received her MSc degree in Computer Science from the Department of Computing and Information Sciences, University of Guelph, Canada, in 2001. She is currently in Pattern Analysis and Machine Intelligence Laboratory, Waterloo, Canada, pursuing her PhD. Her research focus areas are Machine Learning, Pattern recognition, and Cooperative Learning Systems. In addition, she is currently working in Software R&D Department, Research in Motion Ltd., Canada, where, she is involved in Analytical Customer Relationship Management, and Software Measurement and Analysis research areas.