



Poetic Features for Poem Recognition: A Comparative Study

Hamid R. Tizhoosh

*Pattern Analysis and Machine Intelligence Group,
Systems Design Engineering Department,
University of Waterloo,
Waterloo, Ontario, Canada*

tizhoosh@uwaterloo.ca

Farhang Sahba

*Systems Design Engineering Department,
University of Waterloo,
Waterloo, Ontario, Canada*

fsahba@alumni.uwaterloo.ca

Rozita Dara

*Pattern Analysis and Machine Intelligence Group,
University of Waterloo,
Waterloo, Ontario, Canada*

rdara@engmail.uwaterloo.ca

Received February 25, 2008. Received in revised form June 30, 2008. Accepted September 19, 2008.

Abstract

Poetry is a form of art that is used to express emotions and feelings. Humans can easily distinguish poetry without any sophisticated tools. This study is concerned with developing intelligent methods which can be used to distinguish poem from prose. The goal is to distinguish and extract effective poetic features with which poems/lyrics can be accurately classified from other type of texts. In this paper, we propose five different approaches to poem classification. In each approach, we extracted a different set of poetic features and evaluated their performances against each other. In addition, we empirically assessed the effectiveness of traditional text classification methods for poem recognition and compared it with the proposed poetic features. While all of these approaches performed well, some showed superior results. Findings of this study suggest that the proposed features generate highly accurate classifiers, which can be used for poem mining in large databases such as World Wide Web.

1. Introduction

Humans have a great capability of recognizing the nature of a text. In contrast, designing accurate computer algorithms for text classification is a challenging task. With the growth of information, particularly in digital form, research in the area of text categorization has been increasing in the past decade. Text classification algorithms have been developed to categorize news [21], patents [19], e-mails [2], magazine articles [18] or any other specified classification to retrieve information from the Internet [10]. In addition, search engines are constantly looking for new methods to segment one special type of data from their repository. Previous works in text classification use techniques to associate specific terms and contents with the type of documents [6]. These methods are not easily transferable to poem classification, since a poem is not limited to specific keywords. A poet may use any word he/she desires. In fact, poems are often structured differently than any other text document, and therefore, an alternative approach must be applied for poem recognition.

Computer algorithms that can recognize and classify poetry from prose may have some important applications. A direct application can be improvement of automated document retrieval from a large database, like the World Wide Web. For instance, if we search for “Eliot mystery” we expect to find the poem “Macavity - The Mystery Cat” by T.S.Eliot. However, the first four search results are completely irrelevant. More drastic examples can be easily identified where current search machines are unable to provide accurate results for poem search. In addition, an alternative application is that the design of such algorithms can provide valuable insight into how humans recognize poems, what features are important and how the features contribute to the recognition. This can lead to the development of algorithms that emulate human processing, which in turn can be applied in other pattern recognition and data mining areas.

Our main concern here is neither to understand poems the same way humans do, nor to find the best algorithm to classify poetry. The goal is to investigate the effectiveness of some of the features discussed in this paper to classify documents as a poem or prose. Features of interest are those that offer sufficient distinction between poem and non-poem. This work is the continuation of the earlier effort by Tizhoosh and Dara [22]. This time, we take a more systematic approach to the task of poem recognition by introducing and developing poetic features in three different categories of shape meter and rhyme. In addition, we studied their recognition capability collectively and individually and compared the recognition capability of the proposed poetic features with traditional text classification approaches using a carefully selected data set.

In this paper, we focus on extracting genre-specific poetic features for English language such as rhyme, meter, and shape to advance new techniques for poem classification. Poems come in different shapes and forms, suggesting a large extent of variability within the poem document domain. For example, a traditional Shakespearean sonnet will have a very different structure in comparison to a free verse poem in which formal rules of poetry (e.g. rhyme, meter) are not applied. In addition, we would like to assess the proposed poetic features with respect to text classification procedures by extracting common context-independent features. We expect that the proposed features boost the classification accuracy. Five different approaches to poem recognition are introduced in this paper. In the first approach, we implement and examine a popular text classification method for poem classification, and compare it with the performance of the poetic features. The next three approaches focus on poetic features, their individual and combined performance. In the last one, we combine the feature space generated using text classification method with a set of poetic features and analyze the results with respect to other approaches.

The primary focus here is on the features and their extraction, not on the classification technique. Therefore, we selected two popular classifiers, naive Bayesian and Multilayer Perceptron (MLP), for the classification task. The Multilayer Perceptron has been successfully used in variety of applications, including text classification [14]. In addition, the use of naive Bayesian classifiers in text classification has been studied before [6]. Most of these works use Bayes rule to analyze specific word organization, not structural features. The classification approach proposed in this paper focuses on both aspects of the text.

This paper is organized as follows. Section 2 includes a brief description of works related to poem recognition. Section 3 discusses the poem characteristics. Section 4 is a brief review on naive Bayesian classifier. Section 5 presents the proposed approaches including details about the features for each algorithm. In section 6, the experimental results are presented. Finally, discussions and conclusions are described in sections 7 and 8, respectively.

2. Related Works

Text classification is an expanding area of research in the field of data mining and knowledge extraction. There are many types of statistical and machine learning algorithms to classify text documents. Some examples include support vector machines [25], decision-trees [25], Neural Networks [14, 20], regression models [28, 26], k-nearest neighbor [27], and naive Bayesian [2, 16, 24]. Despite of some limitations in its assumptions, naive Bayesian classifiers have performed surprisingly well in text classification, which states that document attributes are independent of each other given the context of the classification [15].

In the area of literary texts, some work was done to attempt to identify the features used by humans to classify poems [7]. The results indicate that both changes in phonetic and graphical information affect classifications. On another level, there have been some developments in the area of poetry generation and understanding. With the means of classification techniques, poetic concepts in Portuguese such as stanzaic form, the number of syllables on the verses, and the rhyme scheme were used to classify and suggest ending word of a verse [13]. With such a system, the authors intended to help students to understand the structure and rhyme of poems and to encourage them to write their own poems. Manurung developed an evolutionary model to formulate poetry generation [15]. The process is considered more complex than natural language generation because of the different language elements used in poetry. Logan and Kositsky [12] applied a standard semantic text analysis technique to a collection of lyrics found on the Web and explored the use of this analysis to determine artist similarity. They compared their results with the existing acoustic similarity technique.

In the area of poem recognition and classification using machine intelligence and data mining techniques only one preliminary work has been reported [22]. Furthermore, according to the current research and information related to poetry, it is still not clear what the important features are. This paper investigates the poetic feature selection for poetry recognition with the long-term goal of developing customized search procedure for literary texts.

3. Poetry

Poetry constitutes a strong unity between content and form and is a particular form of literature characterized by specific use of sound and meanings of language to create ideas and feelings [3, 15]. Elements of poetry that allow humans to distinguish it from other types of text include rhythm and meter, sounds, imagery and form. Poems are written using these different elements of language at four levels: sonic, typographical, sensory and ideational level. The sonic level refers to the elements that can be heard, like rhythm and meter. Rhythm is the flow of sound produced by the poem and meter is the repeating pattern of the rhythm [23]. A meter is usually counted in syllables. The typographical level refers to the features that can be determined from simply looking at the printed page. They can include shape, rhymes, meter and meaning. The sensory level involves the use of language constructions that appeal to emotions. Finally, the ideational level deals with the words, specifically grammar and syntax. While a perfect algorithm would need to extract elements at all different levels, this study focuses on the language elements at the typographical level. The aim is to determine the performance of a classifier that only uses visual features. In the following subsections, an overview of most significant poem features rhyme, meter, shape and meaning will be provided.

3.1 Rhyme

Rhyme is widely recognized as being associated with poems. Rhyming is common in poetry and it basically means sounds agree with, and more often refers to, end rhymes. The difficulty in leveraging rhyme as a feature is to determine the degree to which two words rhyme. The Moby pronunciation dictionary, from Carnegie Mellon University, provides a map from words to phonemes, but this

does not entirely solve the problem as robustly identifying rhyme requires operating at the syllable level rather than at the phoneme level [11]. The problem is made more difficult by the presence of proper nouns and relatively uncommon words in the poem corpus which are not listed in the Moby dictionary. In the current system, Moby's phonemes are used as the basis for rhyming, with the ratio of the number of matching phonemes, starting from the end of the word, to the smallest number of phonemes in the two words being tested as the rhyme score. If words are not found then the matching is done based on letters. The algorithm looks for *AAAA*, *ABAB*, and *AABB* rhyme patterns in the text, with each line that fits the sequence contributing its rhyme score to the total for that pattern. The document's feature value is then the highest of these scores.

3.2 Meter

Ideally a poem classification system should be able to detect the metrical structure of the poem. This involves finding periodic sequences of stressed and unstressed syllables. Determining stresses within words is difficult to accomplish robustly, although algorithms do exist to estimate how stresses fall within and among words [5, 11]. A decent proxy for meter is simply the number of syllables per line; while the exact pattern of stresses is lost it retains a sense of the rhythmic regularity of poetry. Syllables per line can be determined via proper dictionaries such as the Moby hyphenation dictionary, from Carnegie Mellon University. This, combined with basic stemming, has 80% accuracy in determining the number of syllables per word in the corpus. Words which are not found in the dictionary are broken down according to the simple heuristic that every three letters contributes one syllable to the word.

3.3 Shape

Poems have distinguishable structural features (*shape*) which can be used to classify poems. These features are simple and easy to distinguish, e.g. the lines of poems are often much shorter than those of prose. In addition, recognizing line grouping can be helpful for some of the more traditional styles of poetry which will often include regular line breaks (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. Moreover, pre-line white space may be a valuable measure. One major problem with shape is that it can be eliminated by text formatting and/or re-formatting.

3.4 Meaning

Some poems have line breaks at regular intervals and there is not much variety in the line length. In addition, many poems do not have rhyme or rhythm. An undeniable quality in almost any poem or lyrics is the image provided by words choice. If quantitatively measured, this feature would provide the clearest measure of poetry available. There are several ways to extract such features. One potential way to measure the type of imagery is to calculate the number of nouns, verbs, adjectives, and adverbs in each phrase, and set it to relationship with the number of total words in that phrase. An average of such calculation can provide a measure without access to the actual meaning of the words. In addition, phrase repetition is another common characteristic of poetry (lyrics) which can be easily measured.

4. Classification

As discussed before, we selected naive Bayesian and Multilayer Perceptron (MLP) for the classification task. The naive Bayesian classifier has been successfully used for text classification [2, 16, 24]. It is a simplified form of the statistical Bayesian method [4, 17]. The naive Bayesian classifier determines the class as follows

$$C_{NB} = \max_j P(C = C_j) \prod_{i=1}^n P(a_i | C = C_j), \quad (1)$$

where $P(C_j)$ is the a priori probability of class C_j and $P(a_i | C_j)$ is the a posteriori probability of an instance of a_i given C_j . The a posteriori probability for different attributes (or features) is determined from the set of training data. These probabilities tend to be very small for large data sets. Thus, the logarithm version of Equation 1 can be used:

$$C_{NB} = \max_j \left[\log P(C_j) + \log \sum_{i=1}^n P(a_i | C_j) \right]. \quad (2)$$

Multilayer Perceptron (MLP) have previously been applied to the problem of categorizing text documents [14, 20]. Unlike naive Bayesian, MLP does not assume the conditional independency among the data attributes. An MLP with one hidden layer through the back-propagation learning algorithm has been used for poem classification. The number of hidden nodes were set to 9 in all the experiments. A fraction of the training examples were held out for validation (early stopping criterion). The performance of this set was monitored throughout the iterative learning procedure to avoid over-fitting. Furthermore, other parameters such as the number of hidden nodes were set using trial-and-error method.

5. Proposed Approaches

We conducted an extensive background research on different types of poems and various aspects of poem. In the previous section, we gave an overview of various poetic features. In the following sections, an overview of the proposed approaches in five different case studies is presented. Each case study is concerned with a different set of features. Maintaining the same underlying method was important to ensure that the only difference between algorithms were the features. Among these cases, approach 1 is a classical text classification method which is used as a baseline. More detailed explanation is highlighted in the related sections.

5.1 Approach 1: Traditional Text Classification Method

In this approach, we first used the most common preprocessing techniques to obtain the training and test data. During preprocessing of the data, we skipped headers, pruned words occurring in less than three documents and used a stop list. We employed two popular feature selection techniques, Information gain and Mutual Information, to further reduce the dimensionality [27].

Information gain (IG) measures the number of bits of information available for category prediction given the presence or absence of a given term. The information gain of a term, t , given classification, C , can be estimated by

$$\begin{aligned} \text{IG}(t, C) = & -P(C) \log P(C) \\ & + P(t)P(C | t) \log P(C | t) \\ & + P(\bar{t})P(C | \bar{t}) \log P(C | \bar{t}). \end{aligned} \quad (3)$$

Given a word t and a classification C , the mutual information MI between t and C can be estimated by

$$\text{MI}(t, C) = \log \frac{P(t \wedge C)}{P(t)P(C)}, \quad (4)$$

or alternatively,

$$\text{MI}(t, C) = \log P(t | C) - \log P(t) \quad (5)$$

5.2 Approach 2: Shape Features

Word distributions in modern poetry may vary widely, while features such as short lines are more universal. The majority of poems can be visually identified without actually reading the words. This structural information is required in a quantitative form for analysis. When it comes to poetry, among all structural characteristics, length of line is the most distinguishable attribute in human eyes. In this approach, we first assessed this feature separately by extracting mean and standard deviation of line length.

1. Average μ_{LL} and standard deviation σ_{LL} of line length.
2. Average μ_{PS} and standard deviation σ_{PS} of paragraph size.
3. Average μ_P and standard deviation σ_P of punctuation (periods, colons, apostrophes) .
4. Average μ_{LC} and standard deviation σ_{LC} of list characteristics (such as bullets, dashes or numbers) in each paragraph.
5. Average μ_N and standard deviation σ_N of numbers per paragraph.

For above features we define a Gaussian probability distribution for each attribute.

5.3 Approach 3: Combined Rhyme, and Shape Features

This approach classifies a random text as poem or non-poem using rhyme and shape features of the documents. Ten features were extracted and combined for each document:

1. Average number of rhymes per lines:

$$\mu_R = \frac{1}{L} \sum_{i=1}^L N_{rhyme,i}, \quad (6)$$

where $N_{rhyme,i}$ is the total number of rhymes in line i .

2. Total number of lines of text L ;
3. Total number of paragraphs (group of lines separated by a blank space) P ;
4. Total number of words N_{word} ;
5. Average number of words per paragraph;

$$\mu_W = \frac{1}{P} \sum_{i=1}^P N_{word,i} \quad (7)$$

where $N_{word,i}$ is the total number of words in paragraph i .

6. Average and standard deviation of number of letters per line;
7. Average and standard deviation of number of words per line;
8. Average number of periods per line.

$$\mu_{periods} = \frac{1}{L} \sum_{i=1}^L N_{periods,i} \quad (8)$$

where $N_{periods,i}$ is the total number of periods in line i .

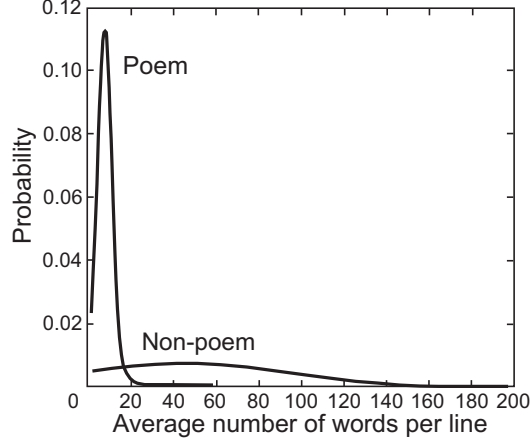


Fig. 1: Probability function estimation for the number of words per line.

These features are used to capture the shape and rhythm of the documents. The total number of rhymes per lines is used because rhymes are a common element in poems to create sound. In addition, the average number of letters per line was used as a measure of meter. A posteriori probability for each feature was estimated using a Gaussian distribution. The mean and standard deviation of the distributions are calculated using the feature values from the training data. Figure 1 depicts the Gaussian distribution for the average number of words per line.

5.4 Approach 4: Rhyme, Meter, and Shape

In this approach, we combined various structural features to capture all aspects of poetry. Poetic features used in this approach are number of lines, meter, rhyme, and the presence of numerical characters. As of shape features, we extracted average and standard deviation of the number or words per line, letters per paragraph, and paragraph (stanza) per document.

1. $\mu_{word.line}$ and standard deviation $\sigma_{word.line}$.
2. $\mu_{letters}$ and standard deviation $\sigma_{letters}$.
3. $\mu_{paragraph}$ and standard deviation $\sigma_{paragraph}$.
4. *Numbers*: The final feature used in the current system tracks the percent of words which contain numbers. This is based on the simple heuristic that poems tend to avoid numerical representation of information.

$$\mu_{numeric} = \frac{N_{numeric}}{N_{word}} \quad (9)$$

where $N_{numeric}$ is the total number of numerical characters used in the text.

5. *Metrical structure*: Similar to the other structural features both average and variance are measured for syllables per line.

$$\mu_{syllables} = \frac{1}{L} \sum_{i=1}^L N_{syllables,i} \quad (10)$$

$$SD_{syllables} = STD(N_{syllables,i}) \quad i = 1, \dots, L \quad (11)$$

6. *Rhyme*: Average $\mu_{rhyme.line}$ and standard deviation $SD_{rhyme.line}$.

Finally, for each document, a feature vector is built and used to train the classifier.

5.5 Approach 5: Combining Poetic Features (Approach 3) with Word Frequency

The last classification algorithm combines poetic features with word frequency to categorize a document. In this approach, however, vocabulary extraction and indexing process are slightly different from approach 1. After stop-word removal and stemming pre-processing steps, the initial vocabulary is analyzed to retain words that contain useful information. We first calculated relative frequency RF for each word as

$$RF_{ij} = \frac{n_{ij}}{\sum_{i=1}^n n_{ij}}, \quad (12)$$

where n_{ij} is the occurrence of word i in class C_j ($j = 1$ or 2). Then, we apply the following selection criteria to skip over less informative words and to reduce the dimensionality:

1. $RF > 0.002$ in poems and in non-poems;
2. Difference in RF_s between poem and non-poems < 0.30 .

The first criterion selects the words that are less frequent in poems and non-poems. The last criterion retains words that have a significant difference in their occurrence in both classes. The threshold values for both criteria were obtained based on trial-and-error by analyzing the relative frequencies of different sample vocabularies and the corresponding words. Subsequent to the selection of the vocabulary, the probability of observing a particular word in the feature vector, given that the document is from class C_j , is calculated by

$$P(w_k | C_j) = \frac{n_{ij} + 1}{N_j + |V|}, \quad (13)$$

where N_j is the total number of words in the training documents from class C_j and $|V|$ represents the size of the vocabulary.

To build the final vector space, we combined poetic features obtained in Approach 3 with the set of attributes obtained by the word frequencies to categorize the documents. This combination consists of an equally weighted linear combination of logarithm of the conditional probabilities for all the attributes. Once the feature vectors are built, they are used to train the classifier.

6. Experiments and Results

6.1 Data Set

We carefully collected 850 text documents in order to include challenging cases. The data set includes 500 poems and 350 non-poems. The poems vary in style, length, shape and content. They include classical poems, as well as free verse poems that do not contain rhymes. The size of poems vary between 3 and roughly 100 lines. The non-poems include a wide variety of documents, such as news articles, interviews, excerpts from books, short stories, advertising, etc. They also vary in style, structure and content. Many selected non-poems are quite short since the classification of long prose (several long paragraphs) as such seems to be trivial. These documents were collected through various sources including online magazines and websites. The manual selection appeared to be more reasonable because the required diversity and the degree of difficulty could be controlled based on subjective perception. In contrast, an automated sample collection, even though capable

of providing much larger number of samples, would not be able to exclude trivial cases or include really challenging ones.

In each run, the data set was divided into training and testing sets, where the training set constituted 70% of the total data set. The same percentage was applied to poems and non-poems. To obtain reliable results, each classification algorithm was tested 20 times. During each test, all the aforementioned approaches used identical training and testing sets. The documents within the training and testing data sets were selected randomly at the beginning of each trial. The MLP and naive Bayesian algorithms classified a document as poem or non-poem. We used 10% of the training data for validation of MLP performance. The performances of the proposed approaches, for the training and testing data, are summarized in the following sections.

6.2 Approach 1: Traditional Text Classification Method

As it was mentioned, we applied the most common text classification method as a baseline. We first preprocessed the data. Given that the size of vocabulary was up to 20,000 distinct words, we reduced the dimensionality by methods described in Section 5.5. Dimensionality of the data was gradually reduced to 1000. At each step of the reduction, we tested the performance of each feature selection method by training a naive Bayesian classifier. The average accuracy and standard deviation of the training results is illustrated in Table 1, for each feature selection method and corresponding vocabulary size. Among the feature selection techniques, the corresponding accuracy rate and standard deviation decreased as the vocabulary size decreased. Information gain IG generated the best overall average accuracy of 93.60% for 5000 words. Mutual information MI exhibited poor performance for smaller vocabulary sizes and also had higher standard deviation values in comparison to IG. However, for a vocabulary size of 15000 words, MI achieved the same accuracy rates as IG.

Table 1: Experimental Results for Approach 1

Feature Selection	Number of Words	Training Accuracy	Testing Accuracy
Information Gain	1000	92.15±2.17	90.33±1.29
	5,000	93.74±2.35	93.60±0.26
	10,000	92.56±2.15	92.26±0.26
	15,000	93.29±2.01	92.41±1.55
Mutual Information	1000	52.16±6.47	48.96±3.64
	5,000	34.60±4.89	36.01±0.26
	10,000	33.49±4.13	35.27±0.0
	15,000	93.29±2.01	92.91±1.55

Despite the type of feature selection method, the highest test accuracy was close to 93% for naive Bayesian classifier. This accuracy is close to the ones obtained in the previous approaches. This interesting finding suggests that even the most popular text classification method does not outperform the simplest shape features introduced in this study.

In the section 6.6, where we combine text classification method with structural features, we analyze the results in more detail to examine which algorithm classifies ordinary text more accurately and which classifies poems with high accuracy.

6.3 Approach 2: Shape Features

The accuracy of the classifier for training and test data are summarized (in percentage) in Table 2. This table also reports standard deviations over 20 trials. The first observation is that such simple features have shown to be quite effective. Both classifiers were able to distinguish poem from prose correctly in more than 92% cases, using only shape features.

Comparing MLP and the naive Bayesian, their generalization accuracies were similar. The most noticeable difference was in the standard deviation. The MLP was significantly more variable probably due to the randomness in the selection of the initial network parameters (e.g. weights).

Table 2: Accuracy of Training and Test data for Approach 2

Classifier	Training Accuracy	Testing Accuracy
naive Bayesian	92.4 ± 0.69	92.2 ± 1.4
Multilayer Perceptron	93.5 ± 7.1	92.4 ± 6.8

Another interesting observation was that, apparently, these features were mainly effective for poems. When we broke down the number of misclassified documents in each class, we realized that 16 – 20% of the non-poem documents were misclassified; while less than 2 – 3% of the poems were misclassified (see Table 3). This is most likely due to the fact that the extracted features were more representative of the poems, which could not provide an accurate representation for the non-poems. MLP slightly outperformed naive Bayesian on non-poems documents.

Table 3: Disparity between poem and non-poem accuracy

Classifier		Training Accuracy	Testing Accuracy
naive Bayesian	<i>Poem</i>	98.4	98.3
	<i>Non – Poem</i>	80.2	80.0
Multilayer Perceptron	<i>Poem</i>	97.5	96.8
	<i>Non – Poem</i>	84.7	83.4

6.4 Approach 3: Rhyme, and Shape Features

We also examined another aspect of poetry. This time, average and variance of the number of the words that rhyme were added to the shape features. The experimental results for the training and testing data are given in Table 4. Once again, both classifiers were able to achieve a very high accuracy. However, rhyme feature did not boost the classification accuracy. This can be explained for two reasons. First, rhyming is sometimes hard to capture and/or to quantitatively measure. Furthermore, with shape features we were able to achieve accuracy higher than 98% for poem. Therefore, it was expected that rhyme feature would not able to improve the remaining 2%. After detailed investigation of the misclassified data patterns, we observed a similar pattern as the

Table 4: Experimental Results for approach 3

Classifier	Training Accuracy	Testing Accuracy
naive Bayesian	92.01 ± 0.76	91.87 ± 1.36
Multilayer Perceptron	94.01 ± 3.81	92.57 ± 5.11

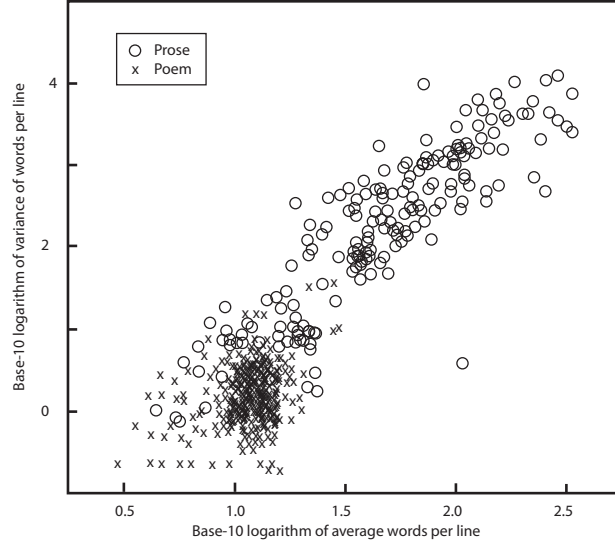


Fig. 2: Class distribution in feature space.

previous approach. The difference between the ratio of misclassified poems and non-poems was close to 18%. This observation again reinforces the fact that the generated feature space did not represent both classes unequivocally.

6.5 Approach 4: Rhyme, Meter, and Shape Features

Here, we added another structural feature to shape and rhyme features, the average and variance of the number of syllables in the text. The degree to which such extracted features are successful in separating two classes has been highlighted in Figure 2. It is important to note that a logarithmic scale is used on both axes for the sake of a better visualization. As it can be seen, extracted features produce a single highly dense cluster for poems, while ordinary text has a rather widely scattered distribution. In this approach, the average test accuracy over 20 trials was 92.1% using naive Bayesian classifier. Apparently, adding the size of syllables to the previous set of features did not improve the accuracy either. In order to gain insight into the effectiveness of each feature, we evaluated each feature individually. We calculated the difference between classifier performances with and without each feature. The average performance improvement (percentage) for each feature is summarized in Table 5. All features resulted in improved performance. Some of the features have major effect such as the number of words, while others have minor effect (e.g. number of lines). The effect of the number of words per line can also be validated by comparing its probability for poems and non-poems highlighted in Figure 1.

Table 5: Experimental Results for Approach 4 (Δ : Average percentage change in accuracy from feature inclusion)

Feature	naive Bayesian Δ	Multilayer Perceptron Δ
Words/Line	+3.77%	+3.58%
Lines/Stanza	+2.13%	+1.42%
Syllables/Line	+4.08%	+2.18%
Rhyme Score	+2.91%	+0.86%
Numbers/Words	+7.51%	+0.83%

Findings of Approaches 2, 3, and 4 suggest that some structural features are highly correlated. Existence of some features compensated for the absence of other features. In each approach, we substituted some of the features with new ones, and obtained similar accuracies every time. Since capturing rhyme and meter features is a complex and time-consuming task, these observations generally recommend that poems can be accurately classified using less expensive and less complex features such as shape.

6.6 Approach 5: Combining Poetic Features with Word Frequency

We combined poetic features obtained in Approach 3 with the set of attributes obtained using word frequency method to categorize the documents. Once the feature vectors were built, they were used to train the classifier. The experimental results for the training and testing data are given in Table 6 and 7. We examined these sets of features individually (Algorithms 1 and 2) and combined (Algorithm 3) to evaluate their performances. Both classifiers built on word frequency and poetic features have resulted in high accuracy rates. The accuracy of the classifier built on word frequency was slightly higher than the one trained with poetic features. This is likely due to the fact that there are such words, namely *poetic words*, that are strongly associated with poems. These words fall under “meaning” attribute category (Section 3.4), in which the emphasis is placed on the message that is being delivered than on the shape or appearance of the poem. The poetic attributes that were used in this approach do not include *meaning* features. On the other hand, by using word frequency method, we were able to capture this aspect to some extent. However, similar to approach 1, word frequency classifier only showed marginal improvement over the other approach. This behavior can be justified as follows: The selection of words in poetry has no limits and can be very diverse. As a result, the set of vocabulary generated for poem classification (feature space) is sparse. Subsequently, generalization capability of the classifier is not optimal. Overall, this suggests that the conventional text classification approaches can be effective and efficient means for poem recognition.

Table 6: Experimental Results for Training Data, Approach 5

Algorithm 1: Word Frequency	Algorithm 2: Poetic Features	Algorithm 3: Frequency and Poetic Features
96.80 ± 0.94	91.01 ± 0.76	98.93 ± 0.29

Table 7: Experimental Results for Testing Data, Approach 5

Algorithm 1: Word Frequency	Algorithm 2: Poetic Features	Algorithm 3: Combined Frequency and Poetic Features
92.87 ± 1.58	90.87 ± 1.36	97.29 ± 0.80

To overcome the shortcomings of the classifiers built on poetic and frequency features, we combined these features and trained a naive Bayesian classifier. The interesting observation is that the combined features provided even more accurate performance (Tables 6 and 7). To assess this behavior, we analyzed the results and recorded the classification errors for the test data. This aimed at determining whether the classifiers were misclassifying the same text. We randomly collected the data from 4 of the 20 test runs for the testing data and included the documents that had at least one misclassification. We realized that, most of the times, the documents that were misclassified by

Algorithm 1 were correctly classified by Algorithm 2 and vice versa. This suggests that these two algorithms are complementary, since they make errors on different types of text. This complementary nature of the algorithms confirms that poems have a very varied nature and require algorithms that target more than one aspect. Thus, to make these algorithms more accurate, it would be crucial to assess which features are more important for certain texts and perhaps weight them more heavily. For example, in the case of free verse poetry, putting higher weights on the meaning of the words will improve the recognition capability.

Table 8 summarizes the number of misclassifications over all the 20 trials for poems and non-poems. First, we report how many documents of type poems and non-poems were misclassified by each algorithm (1, 2, and 3). Then, we counted the number of documents that were misclassified in both or all algorithms. The results again support the complimentary nature of the Algorithm 1 and 2. While Algorithm 1 and Algorithm 2 misclassified a total of 254 and 154 poems respectively, only 22 of the exact poems were misclassified by both algorithms. Similarly, only 31 non-poems were misclassified by the these classifiers over 20 runs. In addition, we can see that poems tend to be misclassified by Algorithm 1. Non-poems are misclassified by Algorithm 2 much more often than by Algorithm 1. This is an indication of which algorithm is suited for each class of documents. The word frequency works better at classifying prose, whereas poetic features do boost the recognition capability for the poems.

Table 8: Number of documents misclassified for each classifier (over 20 trials, 3300 poems and 1700 non-poems were classified)

Algorithm	Poems	Non-Poems
Algo. 1	254	67
Algo. 2	154	257
Algo. 3	89	33
Algos. 1 & 2	22	31
Algos. 2 & 3	50	18
Algos. 1 & 3	53	18
Algos. 1 & 2 & 3	14	3

7. Discussions

Findings of proposed approaches can be summarized as follows:

- The results suggest that proposed poetic features are beneficial for effective poem recognition. These features were selected based on their potential to correctly distinguish poetry from prose. We validated the efficiency of these features in the approaches 2,3 and 4 (approach 1 was also used as a baseline).
- Shape features such as line length were found to be perfectly suited for the problem of classifying text as poetry versus prose. In some cases, however, non-poem texts were wrongly classified as poem. A detailed investigation illustrated that when the text document is structured like a poem such as a list or a discussion in a chat room.
- Interestingly, when we extracted and combined features of poetic nature such as rhyme and meter, no significant improvement was observed. This could be due to the type of data extracted. When dealing with emails, text from chat rooms or other short text documents, addi-

tional information may be required to correctly classify data. The use of rhyme or syllable is recommended in these cases.

- The most interesting observation resulted from comparing the findings of the first four approaches to the last one. When the word frequency was applied, the accuracy of the naive Bayesian classifier was close to the ones resulted from using poetic features. However, when the features were combined, we observed 5% improvement in the accuracy. A detailed investigation on the type of misclassified documents illustrated that classifiers built on word frequency and poetic features were complementary since they make errors on different texts.
- Moreover, it was observed that the poems are better classified by the proposed features, while non-poems are better classified using the word frequency approach. This is likely due to the fact that proposed features were too poetic. On the other hand, since the selection of words used in poetry is very diverse, the set of vocabulary generated for poem recognition using text classification methods was not representative for all type of poems. Combining both type of features resulted in higher accuracy.

8. Conclusions

Poetry recognition is a challenging and interesting problem which has been overlooked previously. This study proposes novel poetic features and compares the effectiveness of these features in the task of poem classification. It is important to note that our intention is neither to classify different types of poems nor to classify modern poetry from prose. In addition, our objective is not to show which poetic feature is the best. Our primary goal is to distinguish between prose and all types of poems. Developing efficient algorithms to identify poems can improve the quality of online document retrieval and can also provide an insight into the nature of the features used by humans to recognize poems. The proposed features have the potential to be used in search engines, looking for actual poems as opposed to pages that contain the word poem.

In this paper, we introduced different poetic features and categorized them into four groups of rhyme, meter, shape and meaning. Then, we implemented five different approaches including a common text classification method as a baseline. In each approach, we extracted a different set of poetic features and evaluated their performances against each other. In addition, we empirically assessed the effectiveness of traditional text classification methods for poem recognition and compared it with the proposed poetic features. In another attempt, the feature space extracted using word frequency method was combined with shape features.

A collection of 850 documents was carefully gathered from different sources. This collection contained both prose and poems. The poem set included a varied set of poems, different in style, form, shape, and content, in order to make the poetic features comparison more comprehensive. In all five approaches, naive Bayesian and Multilayer Perceptron classifiers were able to achieve accuracies more than 90% on the test data. This high accuracy confirms the effectiveness of all proposed poetic features for the task of poem recognition. Among all different approaches used in this paper, the highest accuracy was achieved when poetic features were combined with word frequency.

Based on the presented results, it is not possible to state which poetic feature provides more accurate recognition ability for a specific type of poem. This study only focused on the recognition of poem versus prose, and considering all other aspects was beyond the scope of this paper. Classification of different types of poetry from each other or a more challenging task, free verse poetry from prose, will remain to be investigated in the future. It will be interesting to develop algorithms that can distinguish different types of poems. This requires to examine each type of poem and to dis-

tinguish its unique characteristics. Then, a set of poetic features can be associated to each specific type of poem. Other future objectives include considering “meaning” attribute of the poems, and using more sophisticated methods to extract poetic features. As per discussed, we did not capture the “meaning” feature provided by words choice. However, our findings suggest that this feature seems to be the most effective method for free verse poem recognition and may provide the highest recognition ability.

References

- [1] E. Alpaydin, *Introduction to Machine Learning*, The MIT Press, Cambridge, MA, 2004.
- [2] I. Androutsopoulos, J. Koutsias, K.V. Chandrinou, C.D. Spyropoulos, “An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-mail Messages”, *Proc. of the 23rd Annual Int. ACM SIGIR conference on Research and development in information retrieval*, July 24-28, Athens, Greece, p.160-167, 2000.
- [3] P. B. Diehl, “Poetry, *The World Book Encyclopaedia*”, 85th edition, 15. Chicago: World Book, 2005, pp. 591–596.
- [4] R. Duda and P. Hart, “*Pattern Classification*”, 2nd Edition, Wiley, New York, NY, 2001.
- [5] E. Fudge, “Words and Feet”, *Journal of Linguistics*, 35(2), pp. 273–296, 1999.
- [6] T. Joachims, “A Statistical Learning Model of Text Classification for Support Vector Machines,” *SGIR*, New Orleans, Louisiana, USA, 2001.
- [7] D. Hanauer, “Integration of Phonetic and Graphic Features in Poetic Text Categorization Judgements”, *Poetics*, 23(5), pp. 363-380, 1996.
- [8] S. Kim, H. Seo and H. Rim, “Poisson Naive Bayes for Text Classification with Feature Weighting,” *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages*, pp. 33–40, 2003.
- [9] R. Kohavi, B. Becker, and D. Sommerfield, *Improving Simple Bayes Proc. Ninth European Conf. Machine Learning*, 1997.
- [10] W.K. Lai, K.M. Hoe, T.S. Tai, and M.C. Seah, “Classifying English Web Pages with Smart Ant-Like Agents”, *Proceedings of the 5th Biannual World Automation Congress*, 13, pp. 411–416, 2002.
- [11] M. Liberman, A. Prince, “On Stress and Linguistic Rhythm”, *Linguistic Inquiry* 10(3), pp. 249–336, 1997.
- [12] B. Logan, A. Kositsky, P. Moreno, “Semantic analysis of song lyrics,” *In the Proc. of IEEE Int. Conf. on Multimedia and Expo*, 2, pp. 827–830, 2004.
- [13] N. Mamede, I. Trancoso, P. Arajo, and C. Viana, “An Electronic Assistant for Poetry Writing,” *In the Proc. of the 9th Ibero-American Conf. on Artificial Intelligence*, LNCS 3315, pp. 286-294, 2004.
- [14] L. Manevitz and M. Yousef, “One-class document classification via Neural Networks,” *Neurocomputing*, 70(7-9), pp. 1466–1481, 2007.
- [15] H. M. Manurung, “An Evolutionary Algorithm Approach to Poetry Generation”, Ph.D. Thesis, University of Edinburgh, 2003.
- [16] A. McCallum and K. Nigam. “A comparison of Event Models for Naive Bayes Text Classification”, *AAAI-98, Workshop on Learning for Text Categorization*, 1998.
- [17] T. M. Mitchell, “*Machine Learning*”, McGraw-Hill, Publisher, 1997.
- [18] M. F. Moens and J. Dumortier, “Text Categorization: the Assignment of Subject Descriptors to Magazine Articles,” *Information Processing and Management*, 36(6), pp. 841–861, November, 2000. .
- [19] G. Richter and A. MacFarlane, “The Impact of Metadata on the Accuracy of Automated Patent Classification,” *World Patent Information*, 37(3), pp. 13–26, March 2005.
- [20] M.E. Ruiz and P. Srinivasan, “Automatic Text Categorization Using Neural Networks,” *In the Proc. of the 8th Workshop on Advances in Classification Research*, 4, pp. 59–72, 1998.
- [21] L. K. Shih and D.R. Karger, “Learning Classifiers: Using URLs and Table Layout for Web Classification Tasks”, *Proceedings of the 13th International Conference on World Wide Web*, New York, NY, pp. 193–202, 2004.
- [22] H.R.Tizhoosh, R.Dara, “On Poem Recognition,” *Pattern Analysis and Applications*, 9(4), pp. 325–338, 2006.
- [23] L. Turco, “*The New Book of Forms: A Handbook of Poetics*”, Hanover and London: University Press, 1986.

- [24] B. Wang, S. Zhou and Y. Hu. "Naive Bayes-Based Garual Chinese Documents Categorization", Proceedings of World Multiconference on Systems, Cybernetics and Informatics, 2, July, Orlando, 2001, pp. 516–521.
- [25] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization", Information Retrieval, 1(1-2), pp. 67–88, 1999.
- [26] Y. Yang and C. G. Chute, "An Example-Based Mapping Method for Text Categorization and Retrieval", ACM Transaction on Information Systems, pp. 253–277, 1994.
- [27] Y. Yang and J.O. Pedersen. "A Comparative Study on Feature Selection in Text Categorization", Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97), pp. 412–420, 1997.
- [28] T. Zhang and F. J. Oles, "Text Categorization Based on Regularized Linear Classification Methods", Information Retrieval, 4(1), pp. 5–31, April 2001.