An Evaluation of Text Classification Methods for Literary Study Dissertation Proposal

Bei Yu
GSLIS, UIUC

January 04, 2006

Abstract

Literary text classification differs from current text classification in other domains in the following aspects: 1) data - literary texts exibit more varieties of language uses because of their long history and creative characteristics ; 2) category labels - literary scholars assign more kinds of text category labels by topics, styles, genres, authors, eras, and many other literary concepts mixing these factors; 3) purposes - literary scholars use classification as example-based search strategy as well as for feature-category correlation analysis.

Current text classification algorithms are evaluated on topic spotting tasks using "young" benchmark corpora in the domains of news articles, scientific literature, and webpages. It is worth questioning if these evaluation results will be consistent in the literary domain.

Two major factors affect an algorithm's performance in a classification task: 1) the algorithm's inference model (including parameter tuning); 2) the data preprocessing choices. In other words, a successful application relies on the right model and the right features.

This thesis will evaluate the performance of a few popular text classification algorithms on literary text classification tasks under different data preprocessing choices. This study focuses on two data preprocessing choices, stop word removal and stemming. Two classification algorithms, multinomial naive Bayes (NB) and linear-kernel Support Vector Machines (SVM), are chosen for the comparison. Both algorithms are popular and can do both classification and feature-category correlation analysis. The full candidate feature set consists of all word tokens.

Three binary classification tasks are collected from Nora - a pioneering literary text mining project: 1) classifying Dickinson's poems as erotic or not-erotic; 2) classifying chapters in early American novels as sentimental or not-sentimental; and 3) fiction/non-fiction classification. The document size scales up from short poems, book chapters, to books; the document collection size also scales up from single author's work to multiple authors' work. Because the three tasks represent different data types and classification purposes, we treat them as three different case studies. Therefore we do not make cross-task comparison of each algorithm.

The expected results will provide guidance for literary scholars to choose suitable algorithms appropriate data preprocessing operations. The classification results also will provide feature-category correlation analysis and example-based search service to help literary scholars explore the three literary text collections.

Contents

1 Motivation and problem

1.1 Text mining and literary study

The development of information technology and digital libraries is changing the means of scholarly communication and research. With the increased abundance of digital material, making use of the digital collections becomes one vital issue in digital libraries. Future digital libraries should not only support information organization and access, but should also assist knowledge discovery from digital collections [26, 11].

Over the past decade, text mining techniques have been used for knowledge discovery from text collections in many domains, such as scientific literature, business documents and web pages. Text mining tasks include but are not limited to document classification and clustering, topic detection and tracking, trend analysis, information extraction, concept hierarchy construction, text summarization, and question answering. Researchers have also started to integrate these text mining techniques into digital libraries to provide new information services [46].

Humanities scholars are one large user group in digital libraries [28]. Also in the past ten years many humanities computing projects (e.g. Perseus, Orlando, VWWP) have digitized terabytes of full-text humanities resources. Recently some scholars who are specialized in humanities computing have started pioneering work on building easy-to-use text mining tools [1, 2] for the humanities domain. For example, the Nora project aims at developing a web-based text mining and visualization tool to explore literary text collections.

Scholars have also used various text mining techniques to solve literary research problems, such as authorship attribution [22], stylistic analysis [10], ordering transcript versions [42], genre analysis [38], and ontology-based information extraction [3].

1.2 Text classification for literary study

Text classification (also called "predictive text mining") has been one of the most widely used text mining techniques [45]. A classification task consists of two steps. The first step is the training process. A set of documents $D = \{d_1, d_2, \ldots, d_n\}$ are pre-labeled and used as the training examples $\{(d_1, l_1), (d_2, l_2), \ldots, (d_n, l_n)\}$. Each document d_i is defined as a feature vector in the feature space $F = \{f_1, f_2, \ldots, f_m\}$. The second step is the prediction process. Given a set of unlabeled documents, the classifier predicts the label of each document.

Many classification methods (not including neural network and k-nearest neighbor) can also be used for finding a compact feature subset which accounts for most of the discriminative power in the whole feature set [50, 18]. For example, linear classifiers can be represented as a linear function of the features

$C = f(f_1, f_2, \ldots, f_m)$. The weights of the features in the function correspond to their discriminative power in classification.

Is text classification technique also useful in literary domain? The evidence from literary text mining practice has demonstrated the potential of text classification in literary study.

1.2.1 Classification and correlation analysis as typical literary research activities

The results of a comparative vocabulary analysis [49] provide evidence of the potential of text classification techniques in literary study. The vocabulary use was compared among three corpora (1) "KDD" - a corpus of data mining literature; (2) "MUSE" - a corpus of critical literature; and (3) "ANC" - American National Corpus. Table 1 lists 18 highly frequent KDD keywords that represent either the major data mining techniques or the kinds of knowledge to be discovered from the raw data. The frequencies of these keywords in the three corpora are compared in the table. "Df-pcnt" means the document frequency is normalized by the total number of documents. "Tf-ppm" means the term frequency is normalized as the proportion per million words.

Table 1 shows that 11 most frequent data mining keywords (in KDD) are also common in literary essays (in MUSE) but not common in news articles (in ANC). Figure 1 visualizes their frequency differences in TF-ppm. The 11 keywords stand for models, frameworks, patterns, sequences, associations, hierarchies, classifications, relations, correlations, similarities, and spatial relations. Among the 11 keywords common in both KDD and MUSE corpora, "classification" is the only research activity as represented at the word level that is shared by the two communities. "Association", "correlation", and "similarities" are among the kinds of knowledge that both data mining researchers and the literary scholars want to discover from the raw data.

These comparative vocabulary analysis results show that classification and correlation analysis are also typical research problems in the literary domain.

1.2.2 Typical literary text classification scenarios

Many well-studied literary text mining problems can be modeled as text classification problems. Authorship attribution is a typical example of literary text classification problems [22, 43, 34]. A classification algorithm (e.g. discriminant analysis) learns a classifier from the candidate authors' other works, and then uses the classifier to predict the authorship of the disputed literary works.

Text classification can also be used for example-based search in large digital collections. Sometimes a

keyword	kdd-tf	muse-df:tf	muse-df-pcnt:tf-ppm	nytimes-df:tf	nytimes-df-pcnt:tf-ppm
cluster	52	13:17	10:22	50:56	1:24
model	40	99:438	**80:562**	335:597	**8:254**
pattern	29	49:121	**40:155**	167:207	**4:88**
network	23	30:76	24:97	325:724	8:307
classif **classifi**	35	14 19 :64	**11** **15** **:82**	16 50 :74	**0** **1** **:32**
rule	19	81:210	65:269	708:1409	17:598
associ	15	103:479	**83:614**	762:1161	**18:493**
graph graphic	15	2 15 :25	2 12 :32	7 244 :504	0 6 :214
stream	15	19:26	15:33	103:117	2:50
serial seri	10	20 80 :213	16 65 :273	32 537 :946	1 13 :403
relat **relationship**	10	117 98 :1367	**94** **79** **:1753**	386 323 :911	**9** **8** **:487**
framework	10	38:69	**31:88**	25:28	**1:12**
correl	9	20:30	**16:38**	15:21	**0:9**
similar **similarli**	9	99 66 :475	**53** **80** **:609**	484 77 :649	**12** **2** **:276**
spatial	7	19:55	**15:71**	8:8	**0:3**
decis	7	57:161	46:206	683:1110	16:471
hierarch **hierarchi**	6	20 41 :178	**16** **33** **:229**	4 4 :45	**0** **0** **:13**
sequenc **sequenti**	6	34 8 :80	**6** **27** **:103**	4 51 :71	**0** **1** **:30**

Table 1: KDD keyword frequency comparison between MUSE and ANC-NYTIMES

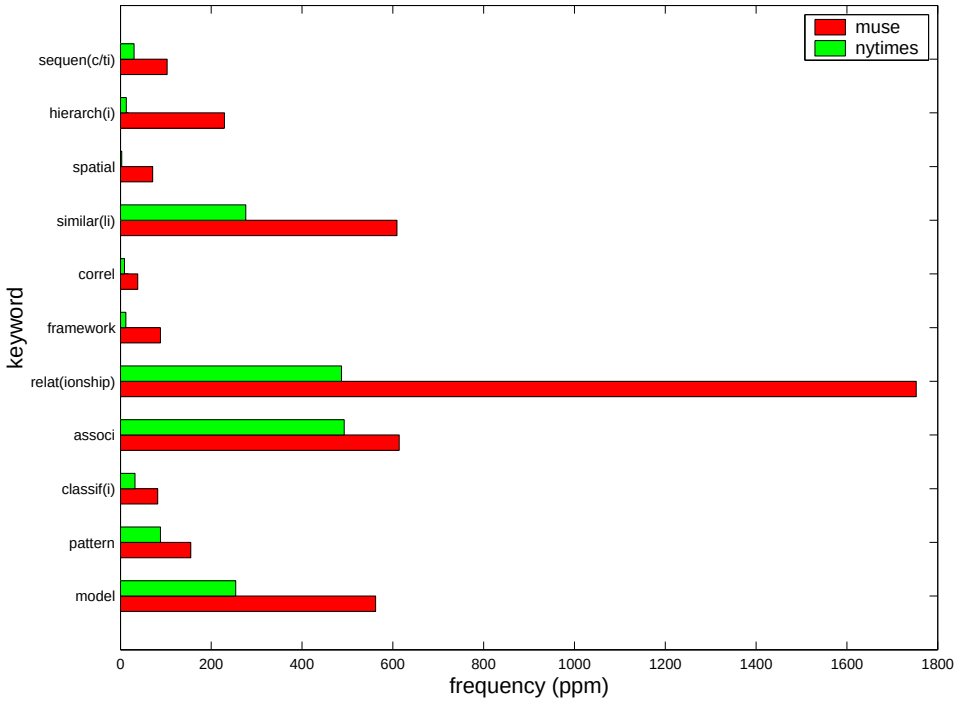


Figure 1: TF of Selected KDD keywords common in MUSE but not common in ANC-NYTIMES

scholar knows only a few examples of a document type and needs to find more similar ones. This task is similar to information retrieval, except that the queries are replaced with document examples.

This search function (or called "find me more like this") is especially useful when a user's information need is difficult to formulate as a keyword-based query, for example, finding examples of ekphrastic poetry [1] from a large poem collection. Such information needs are common in literary research. Ellis and Oldman [16] reported the dissatisfaction with digital libraries by English scholars. They complained that keyword search cannot return comprehensive results, and current digital libraries deprive English scholars of freely browsing, a traditional library activity which helps them to discover knowledge through serendipity.

The "find me more like this" task differs from the previous classification task in that the number of training examples is very small, therefore active learning techniques are needed to facilitate effective prediction of relevant documents using a small training set.

In some cases feature-category correlation analysis is the real purpose of a literary text classification problem. In such cases, the literary scholar probably knows the text collection well, and wants to use classification as way to analyze the collection, and possibly generate new hypotheses from the classification and feature analysis results. For example, decision-tree classification and concept-tree clustering methods were used to explore the correlation between the graphic properties and the genres of Shakespeare's plays [38]. Scholars know that "Othello" is a tragedy, but they discovered from the clustering results that "Othello" is more like a comedy in terms of some theatrical structure characteristics. Interestingly, there is critical literature that argues "Othello" has much in common with comedy.

Sometimes prediction and correlation analysis are both purposes of literary text classification. For example in authorship attribution, the scholars need to identify the most discriminative stylistic markers as the internal evidence to validate the authorship attribution. Therefore the above two tasks are often the two sides of a coin, just as Craig [14] concluded that "classification and description can be mutually supportive: the first confirms the validity of the second, while the second helps to establish the stylistic mechanisms underlying a successful classification."

The scholars may also apply the classifier learned from one text collection to another one as a means of comparative study. For example, scholars may want to use the classifier learned from Dickinson's erotic poems to study the erotic language use in Victorian women writers' works.

In summary, typical use scenarios of text classification for literary study include:

1. prediction (e.g. "is this an erotic poem?")

[1] Ekphrasis is the "literary representation of visual art" (Hefferman). An ekphrastic poem is written in response to all kinds of artworks, including drawings, paintings, sculpture, dance, movie, etc.

2. feature-category correlation analysis (e.g. “why is it an erotic poem?”)

3. example based search (e.g. “can you find more erotic poems?”)

4. comparative study (e.g. “what is the difference between Dickinson and other women writers in terms of erotic language use?”)

1.3 Evaluation of text classification methods for literary text classification tasks

The classification methods, from the machine learning and information retrieval theory perspective, have been well studied in the past decades. The major problem in literary text classification applications is how to choose approriate methods given the characteristics of literary text.

Empirical evaluation is an important approach to acquire knowledge of the fitness of algorithms to real-world data. To date, most text classification evaluation work is conducted experimentally [47, 15, 21]. Analytical evaluation is hard because real world text classification problems are not formalizable [41]. Actually the category assignment of a document is a subjective judgement. To obtain objective evaluation, classification methods have to assume that the categories are simply symbolic labels and are consistent concepts to the users.

1.3.1 The need for re-evaluation of classification methods in literary domain

Literary text mining is a new research area. There is no empirical comparison about the effectiveness of current text classification methods on literary text classification tasks. Most literary text classification research focuses on one single method without comparing to other methods. Discriminant analysis was used for authorship attribution [14]. Perceptron was used to classify philosophical text segments to three classes ”perception”, ”knowledge” and ”mind” [35]. Cross-methods comparison results will be more valuable toward choosing appropriate algorithms for future literary text classification applications.

To date, many text classification methods have been evaluated outside the literary domain [41, 15, 47]. But it remains open whether the evaluation results are transferrable to the literary domain because literary text classification differs from current text classification in other domains in the following ways:

1. data - literary texts exibit varieties of language because of their long history and creative characteristics;

2. category labels - literary scholars assign many kinds of text category labels by topics, styles, genres, authors, eras, and many other literary concepts that combine these categories;

3. purposes - literary scholars use classification also as example-based search tool as well as feature-category correlation analysis tool.

Current text classification methods are mostly evaluated on "young" benchmark corpora in the domains of news articles, scientific literature, and webpages (e.g. Reuters-21578, OHSUMED, and AP collection). But one size does not fit all. Leopold and Kindermann [27] argued that the Reuters corpus is not a "representative" corpus because most stories are shorter than 100 words with restricted use of vocabulary. Obviously literary text documents are in a much more "free" style. It is worth questioning if these evaluation results will be consistent in literary domain.

Current text classification methods are mostly evaluated on topic spotting tasks. It is worth questioning if the evaluation results will be consistent in various kinds of literary text classification tasks.

The design and evaluation of current text classification methods focuses on the prediction accuracy, which causes the problem that the algorithm designer might not even have closely looked at the data [20]. Few algorithm and result details are presented to the user besides the accuracy numbers. Consequently the off-the-shelf text classification packages (e.g. T2K and SVM-Light) usually look like a black box to the end user.

For many document classification applications, such as spam filtering, the prediction accuracy is the primary concern. As long as the accuracy rate is sufficient, users may not want to know how and why the black box works. But prediction is often not the real purpose of literary text classification. Sometimes the scholars need the internal evidence which will support their "post processing" of the classification results.

Therefore the classification methods should be evaluated by more than prediction accuracy for literary study purpose. The confidence of the predictions, and the algorithms' capability of identifying highly discriminative features should also be considered in empirical evaluations.

1.3.2 The need to re-examine the effects of text preprocessing choices in literary domain

Besides the inference model of a classification method, feature extraction and selection is another factor that affects a classifier's performance. In other words, a successful classification application relies on the right model and the right features.

The simplest and the most widely used feature set for text classification is the word tokens. This unigram "bag-of-words" document representation model does not take into account the word context. Some experiments were conducted to explore if higher-order linguistic features would improve the classification accuracy. The results were not consistent [29, 15, 13, 40], and the reported improvements were not significant.

Prior to the prevalence of SVM, feature reduction was important to reduce noise, avoid overfitting and reduce computational cost. The empirical comparison results on feature selection methods (e.g. information gain, mutual information) were also mostly obtained from topic spotting tasks. Since SVM scales up well to the large feature space, feature selection becomes less important. Aggressive feature selection might even hurt the text classification performance because few features are irrelevant to the classification [21]. But stop word removal and stemming are still often used as the default document preprocessing steps [48, 17].

Stop word removal and stemming techniques are used in information retrieval to reduce the number of indexing terms. Due to the close relation between information retrieval and text categorization, sometimes it is taken for granted that these techniques would achieve similar effects in text categorization [13].

But the usefulness of stop words in text classification is actually task-dependent. The pronoun "my" was found to be a very useful word feature to identify student homepages [32]. Prepositions were also found to be highly discriminative features in joint venture document classification [39]. Stop words are even the major stylistic markers in genre analysis, stylistic analysis and authorship attribution [4, 19, 6, 7].

The assumption behind the stemming effectiveness claim is that words with the same stems can be conflated as one feature. But there is no consistent conclusion on the effectiveness of stemming in both information retrieval [20, 12] and text categorization [40, 39, 41]. For example, Riloff [39] reported that singular and plural nouns and different verb forms contribute differently to terrorism document classification.

Therefore it is worth further studying the role of text preprocessing choices, such as stop words and stemming, in literary text classification.

1.4 Purpose of this study

The purpose of this thesis is to evaluate the performance of a few popular text classification algorithms on literary text classification tasks under different data preprocessing choices.

Two classification algorithms, naive Bayes (NB) and Support Vector Machine (SVM), are chosen for the evaluation. This study will evaluate the effectiveness of NB and SVM algorithms in literary text classification problems on both prediction and feature-category correlation analysis sub-tasks. Naive Bayes is a simple but effective classification algorithm [33]. SVM is a more advanced and thus more complicated classification algorithm [21].

We focus on two data preprocessing choices, stop word removal and stemming.

This study seeks answers to the following questions with empirical evidence:

1. How effective are the NB and SVM algorithms in literary text classification?

2. What is the role of stop words in literary text classification?

3. To what extent will stemming affect literary text classification?

1.5 Expected contributions

This study is expected to provide empirical evidence of the effectiveness of naive Bayes and SVM algorithms on literary text classification tasks. It will help us learn more about the characteristics of literary text documents and its effects on the literary text classification performance. The experiment results will provide guidance for literary scholars to choose suitable text classification methods for prediction and feature-category correlation analysis purposes, and choose appropriate data preprocessing operations.

2 Research design

This section will describe the data preparation, experiment design, and experiment environment setting. The text categorization techniques that will be employed in the experiments will be explained in the next section.

2.1 Data preparation

Three binary classification tasks have been collected from Nora - a pioneering literary text mining project: 1) classifying Dickinson's poems as erotic or not-erotic; 2) classifying chapters in early American novels as sentimental or not-sentimental; and 3) fiction/non-fiction classification. The document size scales up from short poems, book chapters, to books; the document collection size also scales up from single author's work to multiple authors' work. Because the three tasks represent different data types and classification purposes, they will be treated as three different case studies. Therefore no cross-task comparison of each algorithm will be pursued.

All the documents are xml files tagged using the TEI-LITE schema. The documents are borrowed from multiple institutions, such as the Institute for Advanced Technology in the Humanities in University of Virginia, University of Maryland, Indiana University, and University of North Carolina at Chapel Hill, the Library of Congress, Brown University, etc.

2.1.1 Dickinson erotic poem classification

To study the erotic language patterns in Dickinson's poems, the literary scholars at the University of Maryland labeled 269 of Dickinson's poems. 99 poems were labeled as erotic ("hot") and the remaining 170 as non-erotic ("not hot"). The scholars expect to study the relations between the linguistic indicators and the erotic poems through classification. In other words, the scholars seek statistical linguistic evidence to answer the question "what makes the poems erotic?"

2.1.2 Sentimental chapter classification

The purpose of this task is two fold: (1) to find sentimental text segments in early American fiction; and (2) to explore what linguistic patterns characterize the sentimentality. A team of literary scholars at the University of Virginia has assigned sentimental scores (1-10 scale as well as low/high categories) for all the 121 chapters in three sentimental novels "Uncle Tom's Cabin" (45 chapters), "Incidents in the Life of a Slave

Girl" (41 chapters), and "Charlotte: a Tale of Truth" (35 chapters).

2.1.3 Fiction/Non-fiction classification

This is a genre analysis problem. The classifiers are expected to discriminate fiction books from non-fiction books and explore the characteristics of the fiction genre. The training/test examples are not ready at this moment.

2.2 Experiment design

Three Experiment sets are designed to seek answers to the three research questions raised in the last section. All the experiments will be repeated on the three data sets.

2.2.1 Experiment set 1: NB and SVM comparison

The NB and SVM algorithms will be compared under each of the four data preprocessing conditions: 1) tokenization only; 2) tokenization and stop word removal; 3) tokenization and stemming; 4) tokenization and stop word removal and stemming.

1. comparing learning curve

   Learning curve measures the test accuracy as the function of the training set size. The purpose of this experiment is to examine how many training examples should be needed for the classification task. In this experiment, 10% examples will be reserved as test examples. The algorithms will be run 9 times with increasng number of training examples from 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, to 90% of the whole example set. The turning point where the curve becomes flat shows the minimum number of training examples needed for best performance.

2. comparing accuracy/coverage curve

   Coverage is defined as a proportion of the predictions ranked by the classifier's confidence measure. For example, after the classifier outputs the ranked prediction results, the 10% coverage will include the top 10% predictions, and the 100% coverage will include all the prediction results.

   The purpose of this experiment is to compare the algorithms' confidence in the predictions. This experiment will use 50/50 train/test split. The accuracies will be obtained at coverage 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100%. The algorithm with an accuracy/coverage curve closer to the upper-right of the graph is more confident at the same accuracy rate.

3. comparing accuracy/feature_reduction curve

   This experiment compares the algorithms by their effectiveness on feature ranking. 50/50 train/test split will still be used in this experiment. The experiment will be repeated using the features ranked as top 100%, 90%, 80%, 70%, 60%, 50%, 40%, 30%, 20%, and 10%. The turning point where the accuracy begins to decline shows the minimum number of features needed, in other words, how many features can be eliminated without reducing the prediction accuracy.

2.2.2 Experiment set 2: the role of function words

Experiment set 1 will have evaluated the effect of removing a widely used stop word set on classification performance. This experiment will focus on examining the role of different stop word categories as classification features.

Denote F as the full feature set, PRN as the pronouns, $PREP$ as the prepositions, DT as the determiners, $CONJ$ as the conjunctions, AUX as the auxiliary verbs, RP as the particles, and CD as the numerals. The classification accuracies will be compared using feature set $\{F\}$, $\{F - PRN\}$, $\{F - PREP\}$, $\{F - DT\}$, $\{F - CONJ\}$, $\{F - AUX\}$, $\{F - RP\}$, and $\{F - CD\}$.

2.2.3 Experiment set 3: the effect of plural nouns and verb forms stemming

Experiment set 1 will also have evaluated the effect of full stemming on classification performance. This experiment will focus on examining the effect of a partial stemming - verb and noun form conflation - on classification accuracy. The classification accuracies with and without partial stemming will be compared for both algorithms.

2.3 Algorithm implementation

The experiments will use T2K [2] as the text preprocesser, in which the GATE tokenizer and Porter Stemmer have been included.

T2K has implemented the multinomial NB algorithm. This implementation conforms to the algorithm design introduced in Chapter 6 of Mitchell's Machine Learning textbook [33]. The current implementation needs to be revised to support feature ranking and prediction ranking outputs.

The SVM-Light package will be used for the SVM evaluations [3]. The package provides prediction values

[2] T2K (Text-to-Knowledge) is software developed on top of D2K (Data-to-Knowledge), both developed by the Automated Learning Group at NCSA, UIUC. Nora project is then developed on top of D2K/T2K and other programs by connecting them through web services. The Nora team has access to D2K/T2K source code.

[3] The SVM-Light package is downloaded from http://svmlight.joachims.org/

for ranking, but it needs to be revised to output feature ranking results. This package has been used in many text categorization evaluations.

3 Text categorization techniques

This section introduces the details of the techniques employed in the experiments.

3.1 External and internal feature ranking approaches in classification

There are two approaches to analyze the feature-category correlation. One approach is to use external feature ranking methods to identify the most informative feature subsets [48, 17], and then evaluate the ranking effectiveness by the classification accuracies using the selected feature sets. Another approach is internal feature ranking – to use the classifier itself to rank the features, and also evaluate the ranking effectiveness by the new classification accuracy with the selected feature subset [14, 50, 18].

This study takes the second approach. The multinomial naive Bayes and linear-kernel SVM algorithms will be used for feature ranking. In the future the internal and the external feature ranking methods will also be compared. If the internal ranking methods work no worse than the external methods, we can save the implementation cost by choosing the internal ranking methods.

3.2 Multinomial naive Bayes algorithm

3.2.1 Comparison of naive Bayes variations

There are many variations of naive Bayes implementations. The multi-variate Bernoulli model (also called binary independence model) and multinomial model are the two often used models [32, 30]. Both models assume the feature conditional independence and disregard word order.

In the multi-variate Bernoulli model, given a training document set D with vocabulary $V = w_1, w_2, \ldots, w_m$, a document is represented as a "binary" word feature vector with length m: $d = (w_1, w_2, \ldots, w_m)$. Each word feature w_j is "1" if the word occurs in the document, and "0" if it does not occur. This model does not take into account the word frequencies and document length. A test document's class posterior is calculated by multiplying the probabilities of all the feature values, including the word features that do not occur in the document.

In the multinomial model, given a training document set D with vocabulary $V = w_1, w_2, \ldots, w_m$, a document is also represented as a word feature vector with length m: $d = (w_1, w_2, \ldots, w_m)$. But the value of each feature w_j is its frequency in the document. A test document's class posterior is calculated by multiplying the probabilities of all the words that occur. In this model the trained classifier should remain the same if we scramble the words in a document and concatenate all the document examples in each class

into one single example. In this sense, the size of each document does not affect the classifier.

Experiment results show that multi-variate Bernoulli model works well on data sets with small vocabulary, but multinomial model works better at large vocabulary sizes. The multinomial model is also more popular in text categorization applications [32, 30]. Therefore the multinomial naive Bayes model is chosen for this study.

3.2.2 The multinomial naive Bayes model

In the training process the algorithm estimates $P(w|c_k)$ for each word w. Denote $l(c_k)$ as the total length of all the documents in class c_k, and $|V|$ as the vocabulary size, then

$$P(w|c_k) = \frac{freq(w) + 1}{l(c_k) + |V|} \tag{1}$$

This estimation uses Laplace smoothing (also called "add-one" smoothing) to save a small amount of probability for words that will appear in test documents but not in the training data.

In the classification process, given a new document $d = (w_1, w_2, \ldots, w_j, \ldots, \ldots, w_l)$, the class posterior is

$$P(c_k|d) = P(c_k) \prod_j P(w_j|c_k)^{freq(w_j)}. \tag{2}$$

Document d is classified as c_1 if $\frac{P(c_1|d)}{P(c_2|d)} > 1$ and c_2 otherwise.

3.2.3 Feature ranking using naive Bayes

The naive Bayes algorithm can provide feature ranking with regard to two classses c_1, c_2. Assuming we are especially interested in class c_1, the word feature's class conditional probability ratio $FR_w = \frac{P(w|c_1)}{P(w|c_2)}$ determines each word's contribution to the class posterior probability because the prediction is determined by the following class posterior ratio PR

$$\begin{aligned} PR &= \frac{P(c_1|d)}{P(c_2|d)} \\ &= \frac{P(c_1)}{P(c_2)} \prod_w FR_w^{freq(w)} \end{aligned}$$

Because the value range of FR is $(0, \infty)$, the log transformation is used to map the values to $(-\infty, +\infty)$. Hence a feature with a positive $logFR$ value is more correlated to c_1, and a feature with negative $logFR$ value is more correlated to c_2. The larger the $|logFR|$ value, the more informative the feature is for the

classification. Therefore in this study $|logFR|$ will be used as the criterion for naive Bayes feature ranking.

3.2.4 Prediction ranking using naive Bayes

The class posterior ratio PR defined in the above formula can also be used as a confidence measure for the naive Bayes predictions. After the log transformation (same as that in feature ranking), a positive $logPR$ value represents the confidence of assigning the example to c_1, and a negative PR represents the confidence of assigning the example to c_2. The larger the $|logPR|$ value, the more confident the classifier is for the prediction. Therefore we use $|logPR|$ as the confidence criterion to rank the naive Bayes prediction results.

3.3 Support Vector Machines with linear kernel

SVM is a supervised learning method that tries to maximize the generalization and thus overcome the overfitting problem [44, 9]. Given the training examples $(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)$, SVM tries to maximize the margins of the decision boundary by finding the maximum of the functional

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i}^{l} \sum_{j}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{3}$$

subject to the constraints

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \alpha_i \geq 0, i = 1, 2, \ldots, l. \tag{4}$$

The examples on the margins have non-zero α_i values; they are called the Support Vectors (SV). The other examples have zero α_i values, and considered not contributative to the classifier.

In the above formula $K(x_i, x_j)$ is the kernel function. Although SVM can handle non-linear boundaries with the kernel tricks, studies show that linear kernel suits the text categorization problem well, and that polynomial kernel and RBF kernel do not improve the performance significantly [27]. Therefore we stick to the simple linear kernel $K(x_i, x_j) = x_i \cdot x_j$ in this study.

Given a test example x, the linear decision function is

$$f(x) = w \cdot x + b$$

where

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i$$

and

$$b = y_i - w \cdot x_i$$

The classification decision is $D = sign(f(x))$.

3.3.1 Feature ranking using SVM

In SVM, each feature f_i has a weight w_i which can be understood as the feature's contribution to the discriminative power of the decision function [18, 50]. Positive w_i represents the feature's contribution to c_1 and negative w_i represents the feature's contribution to c_2. In this study $|w_i|$ will be used as the feature ranking criterion for SVM.

3.3.2 Prediction ranking using SVM

The value of the decision function, SVM's output for each prediction, can be considered as a kind of confidence criterion of the prediction. The larger the absolute value, the farther the point is from the decision boundary, and therefore the classifier is more "sure" of the prediction.

This is a rough mapping between the SVM output and the prediction confidence. Actually a sigmoid model

$$P(y = 1|f) = \frac{1}{1 + exp(Af + B)} \tag{5}$$

can be used to transform the SVM output to class posterior [36, 15].

In this study the prediction ranking will be used only for drawing the accuracy/coverage curve. Direct comparison of the class posteriors between algorithms will not be pursued. Therefore the plain SVM output $|f(x)|$ will be used as the prediction ranking criterion.

3.4 Construction of the stop word list

The concepts "stop words", "common words", and "function words" are usually considered as synonyms in information retrieval and text categorization. But "common words" and "function words" are overlapping but not equivalent concepts.

Korfhage defined stop words generally as the ones that will be ignored in information processing [25]. Sometimes "stopwords" are defined as highly frequent words in a collection, which is equivalent to the concept of "common words" [5, 23]. In this definition, the stop word selection criterion is the word frequency. Because the cutting threshold is an arbitrary decision, the size of a stop word list could range from a few words to a few hundred. This selection criterion is also domain dependent. For example "data" is not a common word in every day English, but it is a highly frequent word in data mining literature. A stop word list generated by frequency count may not be transferrable to other domains.

"Stop words" are sometimes defined as the function words. Function words have an important role in grammar but carry little meaning, and therefore do not contribute much to query-document matching and topic categorization [31]. Most function words are common words and also closed-class words. Hence it is possible to generate a more objective stop word list based on this definition.

This study focuses on seven parts-of-speech categories of functions words: prepositions, determiners, pronouns, conjunctions, auxiliary verbs, particles and numerals. These closed-class function words are enumerated to build a set of "stop words". The word list for each category is obtained from [23]. A few words appeared in more than one list. In this study they will be arbitrarily assigned to their first part of speech in the dictionary. For example "before" could be used as preposition, conjunction and adverb, but it will be considered as only a preposition in the stop word experiments.

This study tries to avoid the part-of-speech tagging process because the current taggers are trained on modern English in news and other domains, and it is hard to evaluate the PoS-tagging accuracy for literary text documents, especially for poetry.

3.5 Stemmer

This study will use the Porter Stemmer, for which the source code is available online [37]. The Porter Stemmer cannot stem irregular nouns and verbs. Since this study will specifically explore the effects of plural nouns stemming and inflected verbs stemming, the external look-up tables for irregular nouns and verbs are needed as complemetary to Porter's rule-based stemming [4].

The Porter stemmer can recognize past tense and present tense verbs by applying the "-ed/-ing" stemming rules. But the stemmer cannot separate plural nouns from third-person singular verbs because it uses the same rule "-s" on both of them. For example, the stemmer chops "sets" to "set" no matter if the context

[4] An irregular verb list can be obtained from webpage http://www.learnenglish.de/Level1/IRREGULARVERBS.htm. I will look for a fairly complete irregular noun list from English grammar books. To date my best candidate online resource is http://www.esldesk.com/esl-quizzes/irregular-nouns/irregular-nouns.htm

is a verb phrase "sets up" or a noun phrase "feature sets". Consequently the stemming rules by themselves can not separate nouns and verbs. In order to further evaluate the effects of plural nouns stemming and inflected verb stemming separately, this study has to either employ the Brill's transformation-based tagger [5] [8] and manually evaluate the results, or just manually separate the words stemmed using the same rule, depending on the size of the candidate word set.

[5] it has been implemented in T2K

4 A pilot study of naive Bayes for Dickinson erotic poem classification

The experiments as planned in Section 2 have not been conducted. This section describes a pilot study of using naive Bayes for Dickinson poem classification, which was finished prior to the experiment design. The results and findings in this case study stimulated the research plan laid out in this proposal.

Dickinson erotic poem classification is an example of using both classification and correlation analysis in literary study problems. Multi-nomial naive Bayes algorithm was used to study the erotic language used in Dickinson's poems. 269 Dickinson's poems were labeled as either "hot" or "not hot". Each poem was represented as a vector of word token features.

4.1 Classification

Table 2 shows the average precision, recall and F1-measure $\frac{2PR}{P+R}$ for 5 runs of classification using 85% randomly drawn poems as training examples and the rest 15% as testing examples. The average F value for "hot" class is 0.53, and 0.69 for "not hot" class. It is not surprising because the original data set includes only 99 "hot" poems and 270 "not hot" poems.

	Precision	sd	Recall	sd	F1	sd
"hot"	0.51	0.09	0.55	0.10	0.53	0.13
"not hot"	0.67	0.19	0.72	0.22	0.69	0.05

Table 2: naive Bayes classification on the total set of 269 poems

In the second experiment the algorithm is run on a balanced data set with 99 "hot" poems and 99 "not hot" poems. Table 3 shows that the average F value increased to 0.75 for "hot" class and 0.76 for "not" class.

	Precision	sd	Recall	sd	F1	sd
"hot"	0.75	0.07	0.75	0.07	0.75	0.05
"not hot"	0.75	0.07	0.77	0.07	0.76	0.04

Table 3: naive Bayes classification on balanced set

The classification accuracy is expectedly not very high. If it were, we would have to conclude that the erotic language use Dickinson uses is cliché. But the accuracy is much higher than random guess, therefore the concept of "poem eroticism" can be approximately learned from the labeled poems. Also the following feature analysis is informative for capturing the erotic language characteristics.

4.2 Feature ranking

The classifier also ranked the word features by the ratios of their class conditional probabilities in the training set.

Table 4 shows parts of the feature ranking list for Dickinson poems. The top five "hot" features are "must", "Thee", "Bud", "Woman" and "joy". The bottom five "not hot" features are "Some", "Whose", "Was", "+" and "Sky".

Log_prob_Hot (p1)	freq_Hot	Log_prob_Not (p2)	freq_Not	p1/p2 ratio	Word
-7.30	6	-9.57	0	2.26	must
-7.30	6	-9.57	0	2.26	Thee
-7.45	5	-9.57	0	2.11	Bud
-7.45	5	-9.57	0	2.11	Woman
-7.45	5	-9.57	0	2.11	joy
...	...	...	...	...	...
-8.55	1	-6.68	17	-1.87	Some
-9.25	0	-7.26	9	-1.98	Whose
-9.25	0	-7.17	10	-2.07	Was
-9.25	0	-7.17	10	-2.07	+
-9.25	0	-7.08	11	-2.16	Sky

Table 4: naive Bayes feature ranking

The list of features with large positive ratio values was compared against a manually generated feature list which includes key "indicators" such as body parts, senses, emotions, seeds, music, winged animals, etc., based on the literary scholars' previous research and intuition. Table 5 shows the difference between the two lists.

"hot" words in both lists	tasted, faces/face, touching/touches/touch, Lords/lord, Berries, feel, Nights, hand/Hands, Nut, Butterfly, seal, Queen
"hot" words in scholar list only	Music, tune, warm, cold, Bee/Bees/bee, night/Night, Lightning/lightning, blood, Love/loves, sun/Sun, nuts, berry, Arms, cut, Itself/itself
"hot" words in TextNB list only	mine, must, Bud, Woman, Vinnie, joy, Thee, write, Eden, luxury, remember, always

Table 5: Comparison between the scholars' indicator list and the ranked feature list

Table 5 shows that some indicators in the scholar's list are not considered "hot" by the algorithm. The feature ranking list also presents some new "hot" words to the scholars. After looking at the feature ranking list, one literary scholar's first impression was "interesting!", second "I knew it!", and third "surprising!". Most "new" indicators are actually not new. Scholars have known most of them but it is hard for them to

recall all the indicators at the word level when they prepared the list.

But there are still some new findings for them. For example, the scholars found "mine" as a new indicator of Dickinson's erotic language [24].

> *The word "mine" as a new indicator identified by D2K is exemplary in this regard. Besides possessiveness, "mine" connotes delving deep, plumbing, penetrating–all things we associate with the erotic at one point or another. And Emily Dickinson was, by her own accounting metaphor, a diver who relished going for the pearls. So "mine" should have already been identified as a "likely hot" word, but has not been, oddly enough, in the extensive critical literature on Dickinson's desires. (Martha Nell Smith, English Department, University of Maryland)*

4.3 Prediction ranking

Table 6 shows an example of the classification results sorted by the log class posterior probability ratio $log(\frac{P(\text{“}hot\text{”}|d)}{P(\text{“}not\_hot|d)})$. The ranking list helps the scholars focus on the wrong predictions the classifier has made with strong confidence, such as "DEAmsEDCSHDh248.1" and "DEAmsEDCSHDh370.1". The scholars re-examined the poems with controversial labels, but stuck to their previous assignments. It demonstrates that the scholars do not accept the classifier's conclusion blindly.

4.4 Visualizing feature-category correlation

The relation between a poem's prediction and the "hotness" level of the word features in the poem can be visualized by color coding the poems by the feature and document ranking measures.

Figure 2 shows three examples of color highlighted poems. The left and right poems are training examples. The middle one is a testing example because some new words are not color highlighted.

The color coding schema in figure 2 assigns a color to each word feature by their log conditional probability ratio R of two classes. Red ($FF0000$) represents the "hottest" words with the highest R value R_{max}, yellow ($FFFF00$) represents the ideal neutral words with $R = 0$, and green $00FF00$ stands for the most "not hot" words with the lowest R value R_{min}.

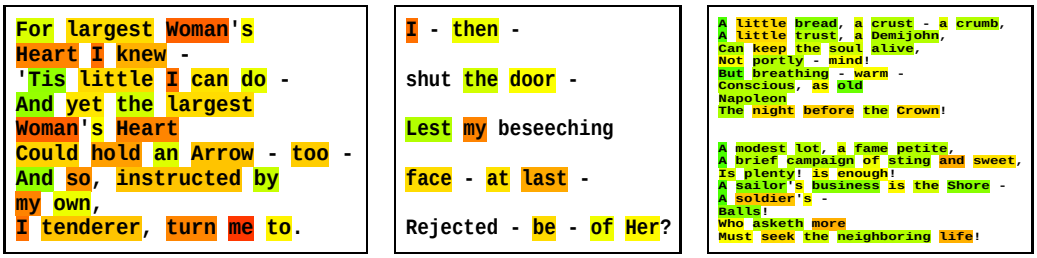
For a word w with positive R value, its color RGB value scales from red ($FF0000$) to yellow ($FFFF00$). The decimal value of the amount of green color is computed using the following formula before being converted back to the hex number.

$$g(w_{hot}) = round(\frac{R_{max} - R}{R_{max} - 0} * 255) \tag{6}$$

For a word w with negative R value, its color RGB value scales from yellow ($FFFF00$) to green($00FF00$).

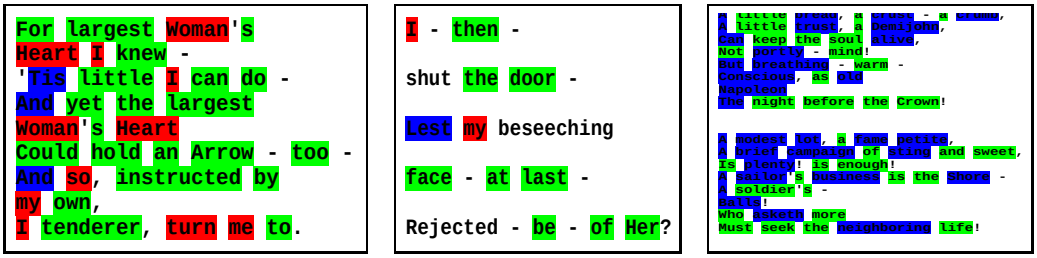
log_prob_ratio	poem	class	prediction
9.96	DEAmsEDCSHDhb63.1	Hot	Hot
7.46	DEAmsEDCSHDhb90.1	Hot	Hot
5.91	DEAmsEDCSHDh271.1	Hot	Hot
5.81	DEAmsEDCSHDhb7.1	Hot	Hot
4.78	DEAmsEDCSHDhb71.1	Hot	Hot
4.64	DEAmsEDCSHDhb188.1	Hot	Hot
3.80	DEAmsEDCSHDh315.1	Hot	Hot
3.46	**DEAmsEDCSHDh248.1**	**Not**	**Hot**
1.87	DEAmsEDCSHDh314.1	Not	Hot
1.87	DEAmsEDCSHDhb43.1	Hot	Hot
1.72	DEAmsEDCSHDh242.1	Not	Hot
1.65	DEAmsEDCSHDh288.1	Not	Hot
1.43	DEAmsEDCSHDh247.1	Not	Hot
1.02	DEAmsEDCSHDh362.1	Not	Hot
1.85	DEAmsEDCSHDhb12.1	Hot	Hot
0.22	DEAmsEDCSHDh308.1	Not	Hot
-0.04	DEAmsEDCSHDa80-7.1	Hot	Not
-0.69	DEAmsEDCSHDh268.1	Not	Not
-0.77	DEAmsEDCSHDh353.1	Not	Not
-0.97	DEAmsEDCSHDhb180.1	Hot	Not
-1.03	DEAmsEDCSHDh343.1	Not	Not
-1.10	DEAmsEDCSHDh324.1	Not	Not
-1.75	DEAmsEDCSHDhb53.1	Hot	Not
-2.46	DEAmsEDCSHDa690.1	Not	Not
-2.47	**DEAmsEDCSHDh370.1**	**Hot**	**Not**
-3.07	DEAmsEDCSHDh235.1	Not	Not
-3.80	DEAmsEDCSHDa655.1	Not	Not
-3.92	DEAmsEDCSHDh346.1	Not	Not
-4.85	DEAmsEDCSHDh359.1	Not	Not

Table 6: Poem ranking by class posterior probability ratio



(a) "hot" poem (b) "not hot" poem (test example) (c) "not hot" poem

Figure 2: Illustration: color coding poems (RYG scaling)



(a) "hot" poem (b) "not hot" poem (test example) (c) "not hot" poem

Figure 3: Illustration: color coding poems (RGB)

The decimal value of the amount of red color is computed as

$$r(w_{not\_hot}) = round(\frac{R - R_{min}}{0 - R_{min}} * 255) \tag{7}$$

The literary scholars appreciate this visualization technique in that it visualizes the relations between the features and the predictions in the algorithm. But the scholars soon demand more freedom in the color coding schema. One scholar suggests using fewer colors to avoid the overwelming information carried in a large palette. The revised red-green-blue (RGB) color coding effects are shown in figure 3 on the same poems as in figure 2. The visualization metaphor is to use warm colors to represent "hot" words, and cool colors for "cool" words. Therefore red means the "hot", green means "neutral", and blue means "cool" words. In the ranked feature list, the first 1/3 features are colored as red, the second 1/3 as green, and the last 1/3 as blue.

4.5 Issues in text preprocessing options

4.5.1 Removing uppercases?

The feature ranking results for the training examples show that many capitalized words rank high. It is surprising that most of them are nouns and not at the beginning of lines. A literary scholar explains this phenomenon as "good old-fashioned emphasis" in that "*use of the upper case in Dickinson and other 19th c. poetry has long puzzled many scholars. The earliest conjectures asserted a connection to German, which of course capitalizes many nouns.*"

A new hypothesis emerges naturally - are these emphasized nouns more important than other words as indicators of the erotic language? Table 7 shows the classification results with only the 1798 capitalized words as features on the same balanced data set. The results show that the F values for both classes are very close to the ones acquired from classification with 4060 word features. For "hot" class the F value (0.77) is even slightly higher. The classification result confirms the hypothesis that these capitalized nouns are more important.

	Precision	sd	Recall	sd	F1	sd
"hot"	0.79	0.08	0.76	0.09	0.77	0.08
"not hot"	0.76	0.09	0.70	0.13	0.72	0.07

Table 7: naive Bayes classification using emphasis word features in balanced set

4.5.2 Stemming?

Table 8 also shows that "Woman" is a "hot" word but "Women" is not, while "Bees", "Bee" and "bee" are all neutral words. A literary scholar agrees that "Woman" depicts a strong emotional figure, but "Women" is more neutral. The result shows that stemming might not be suitable for poetic analysis.

4.5.3 Removing stop words?

Table 9 shows that many pronouns are highly discriminative features in Dickinson erotic poem classification. The first person pronouns ("I", "mine" and "me") and second person pronouns ("you" and "your") are "hot" indicators, while the male third person pronouns ("him", "his", "himself" and "Himself") are "cool" indicators.

Log_prob_Hot (p1)	freq_Hot	Log_prob_Not (p2)	freq_Not	p1/p2 ratio	Word
-7.45	5	-9.57	0	2.11	Woman
-7.86	3	-8.18	3	0.32	Women
-8.15	2	-8.47	2	0.32	Bees
-7.86	3	-7.96	4	0.09	Bee
-9.25	0	-8.87	1	-0.37	bee
-6.94	9	-6.57	19	-0.37	him
-7.05	8	-6.31	25	-0.74	his
-9.25	0	-8.47	2	-0.77	himself
-9.25	0	-7.96	4	-1.28	Himself
-7.30	6	-8.47	2	1.16	mine
-5.31	50	-6.57	19	1.25	me
-4.16	160	-4.88	107	0.71	I
-6.61	13	-7.37	8	0.76	your
-4.68	95	-6.20	28	1.51	you

Table 8: singular and plural nouns as different features

Log_prob_Hot (p1)	freq_Hot	Log_prob_Not (p2)	freq_Not	p1/p2 ratio	Word
-6.94	9	-6.57	19	-0.37	him
-7.05	8	-6.31	25	-0.74	his
-9.25	0	-8.47	2	-0.77	himself
-9.25	0	-7.96	4	-1.28	Himself
-7.30	6	-8.47	2	1.16	mine
-5.31	50	-6.57	19	1.25	me
-6.61	13	-7.37	8	0.76	your
-4.68	95	-6.20	28	1.51	you

Table 9: pronouns as informative features

4.6 Other interesting problems about literary text mining

4.6.1 Evaluation of usefulness of the text mining methods

The scholars found "mine" as an interesting indicator, while they paid no attention to other words which rank exactly the same as "mine", for example, "much" also appears 6 times in "hot" poems and twice in "not hot" poems. They look equally important for the algorithm, but absolutely different for the scholars. An explanation is that the scholars must have been aware what they are looking for. The numbers provide a hint, and the scholars figure out the logic behind it with knowledge external to the data. Another reason is that the scholars look for "new" knowledge. Evidence of confirming prior knowledge is interesting but not surprising, and therefore not valuable for the scholars.

These observations suggest that objective evaluation of text mining tools might not be enough. User evaluation is subjective, but could possibly elicit more ideas of customizing text mining methods for literary study purpose. For example, it could be a research question if there are other feature ranking criteria that facilitate more effective literary knowledge discovery?

4.6.2 What if the algorithm went wrong?

After the literary scholars drew some findings out of the feature ranking list, an unexpected thing happened - a bug was caught in the algorithm implementation code. The ranking was then changed. While some findings still hold, there are some findings affected by this change. For example, the following conclusion is drawn from the old ranking list where "Vinnie" is a strong "hot" indicator. But in the new list "Vinnie" is a fairly neutral word, appearing 5 times in "hot" poems, and 3 times in "not hot" poems.

Actually the bug was caught when the literary scholars ask for adding word frequencies to the output. If they did not demand such details, the bug might never have been caught.

> *"Vinnie" (Dickinson's sister Lavinia) was also labeled by the data mining classifier as one of the top five "hot" words. At first, this word appeared to be a mistake, a choice based on proximity to words that are actually erotic. Many of Dickinson's effusive expressions to Susan were penned in her early years (written when a twenty-something) when her letters were long, clearly prose, and full of the daily details of life in the Dickinson household. While extensive writing has been done on the blending of the erotic with the domestic, of the familial with the erotic, and so forth, the determination that "Vinnie" in and of itself was just as erotic as words like "mine" or "write" was illuminating. The result was a reminder of how or why some words are considered erotic: by their relationship to other words. While a scholar may un-self-consciously divide epistolary subjects within the same letter, sometimes within a sentence or two of one another, into completely separate categories, the data mining classifier will not.*

This story raises another problem about literary text mining. The story shows that the text mining

process and the scholars' post process are not independent from each other. literary scholars sometimes need to know the algorithm details, such as the inference model and the intermediate results, to support their argument. But the text mining algorithm usually looks like a black box, and these details are not presented to the end user. It is worth asking what they need and when they need them. In other words, how do we open the algorithm black box for literary text mining tasks?

4.7 Summary

The naive Bayes classification results and the visualized feature-category correlation analysis have brought the scholars some interesting and surprising findings which contribute to their literary research. The literary research findings have manifested the value of literary text mining.

This case study identified a lot of interesting problems in literary text mining. The problem of choosing algorithms and text preprocessing procedures will be further investigated in this thesis. The other problems, albeit very interesting too, are left for future work.

References

[1] Text analysis portal for research. http://tapor.humanities.mcmaster.ca/home.html.

[2] Web-based text mining and visualization for humanities digital libraries. http://www.noraproject.org.

[3] D. Archer, J. Culpeper, and P. Rayson. Love - 'a familiar or a devio'? an exploration of key domains in Shakespeare's comedies and tragedies. In *Keyword Extraction in Information Retrieval Workshop, ACH/ALLC 2005*, University of Victoria, Canada, 2005.

[4] S. Argamon, M. Saric, and S.S. Stein. Learning algorithms and features for multiple authorship discrimination. In *Proceedings of IJCAI'03 Workshop on Computational Approaches to Style Anslysis and Synthesis*, 2003.

[5] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[6] D. Biber. *Variations Across Speech and Writing*. Cambridge University Press, 1988.

[7] D. Biber. *Dimensions of Register Variation*. Cambridge Univeristy Press, 1995.

[8] E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–566, 1995.

[9] C. J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[10] F. Can and J.M. Patton. Change of writing style with time. *Computers and the Humaninities*, 38(1):61–82, 2004.

[11] H. Chen. Towards building digital library as an institution of knowledge. In *NSF Post Digital Library Futures Workshop*, 2003.

[12] K.W. Church. One term or two? In *Proceedings of SIGIR'95*, 1995.

[13] W.W. Cohen and Y. SINGER. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17(2):141–173, 1999.

[14] H. Craig. Authorial attribution and computational stylistics: if you can tell authors apart, have you learned anything about them? *Literary and Linguistic Computing*, 14(1):103–113, 1999.

[15] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM'98*, 1998.

[16] D. Ellis and H. Oldman. The English literature researcher in the age of the Internet. *Journal of Information Science*, 31(1):29–36, 2005.

[17] G. Forman. An extensive empirical study of feature selection metrics for text categorization. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[18] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machiness. *Machine Learning*, 46:389–422, 2002.

[19] D.I. Holmes. Authorship attribution. *Computers and the Humanities*, 28:87–106, 1994.

[20] D. Hull. Stemming algorithms - a case study for detailed evaluation. *Journal of the American Society for Information Science*, 47(1):70–84, 1996.

[21] T. Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. In *European Conference on Machine Learning*, 1998.

[22] P. Juola and H. Baayen. A controlled-corpus experiment in authorship identification by cross-entropy. *Literary and Linguistic Computing*. to appear.

[23] D. Jurafsky and J.H. Martin. *Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2000.

[24] M. Kirschenbaum, C. Plaisant, M.N. Smith, L. Auvil, J. Rose, B. Yu, and T. Clement. "Undiscovered public knowledge": Mining for patterns of erotic language in Emily Dickinson's correspondence with Susan Huntington (Gilbert) Dickinson. submitted to Digital Humanities 2006.

[25] R. R. Korfhage. *Information Storage and Retrieval*. John Wiley and Sons, 1997.

[26] C. Lagoze. NSF DL position paper. In *NSF Post Digital Library Futures Workshop*, 2003.

[27] E. Leopold and J. Kindermann. Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, 46:423–444, 2002.

[28] M. Lesk. The future of digital libraries. In *NSF Post Digital Library Futures Workshop*, 2003.

[29] D. D. Lewis. Feature selection and feature extraction for text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 212–217. Morgan Kaufmann, 1992.

[30] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Proceedings of ECML'98*, 1998.

[31] C.D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[32] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI 98 Workshop on Learning for Text Categorization*, 1998.

[33] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[34] F. Mosteller and D. Wallace. *Inference and Disputed Authorship: The Federalist.* Addison-Wesley, Reading, Massachusetts, 1964.

[35] J.D. Pasquale and J. G. Meunier. Categorisation techniques in computer-assisted reading and analysis of texts (CARAT) in the humanities. *Computers and the Humanities*, 37:111–118, 2003.

[36] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In P. J. Bartlett, B. Schlkopf, D. Schuurmans, and A. J. Smola, editors, *in Large-Margin Classifiers*. the MIT Press, 2000.

[37] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[38] S. Ramsay. In praise of pattern. In *"The Face of Text" - 3rd Conference of the Canadian Symposium on Text Analysis (CaSTA)*, 2004.

[39] E. Riloff. Littlewords can make a big difference for text classification. In *SIGIR'95*, pages 130–136, 1995.

[40] S. Scott and S. Matwin. Feature engineering for text classification. In *ICML'99*, 1999.

[41] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[42] M. Spencer, B. Bordalejo, L. Wang, A. C. Barbrook, L.R. Mooney, P. Robinson, T. Warnow, and C.J. Howe. Analyzing the order of items in manuscripts of the Canterbury tales. *Computers and the Humanities*, 37(1):97–109, 2003.

[43] F. Tweedie, S. Singh, and D. Holmes. Neural network applications in stylometry: The federalist papers. *Computers and the Humanities*, 30(1):1–10, 1996.

[44] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1999.

[45] S. Weiss, N. Indurkhya, T. Zhang, and F. Damerau. *Text mining: Predictive Methods for Analyzing Unstructured Information*. SpringerVerlag, 2004.

[46] I. H. Witten, K. J. Don, M. Dewsnip, and V. Tablan. Text mining in a digital library. *International Journal on Digital Libraries*, 4(1):56–59, 2004.

[47] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90, 1999.

[48] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of ICML'97*, 1997.

[49] B. Yu and J. Unsworth. Toward discovering potential data mining applications in literary criticism. submitted to Digital Humanities 2006.

[50] H. Yu. *Data Mining Via Support Vector Machines: Scalability, Applicability, and Interpretability*. PhD thesis, Computer Science Department, University of Illinois at Urbana-Champaign, 2004.