Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 421642, 12 pages
http://dx.doi.org/10.1155/2015/421642

Research Article

The Effects of Feature Optimization on High-Dimensional Essay Data

Bong-Jun Yi,[1] Do-Gil Lee,[2] and Hae-Chang Rim[1]

[1]*Department of Computer and Radio Communications Engineering, Korea University, Anam-dong 5-ga, Seongbuk-gu, Seoul 136-713, Republic of Korea*
[2]*Research Institute of Korean Studies, Korea University, Anam-dong 5-ga, Seongbuk-gu, Seoul 136-713, Republic of Korea*

Correspondence should be addressed to Do-Gil Lee; motdg@korea.ac.kr

Received 24 November 2014; Accepted 26 January 2015

Academic Editor: Sanghyuk Lee

Current machine learning (ML) based automated essay scoring (AES) systems have employed various and vast numbers of features, which have been proven to be useful, in improving the performance of the AES. However, the high-dimensional feature space is not properly represented, due to the large volume of features extracted from the limited training data. As a result, this problem gives rise to poor performance and increased training time for the system. In this paper, we experiment and analyze the effects of feature optimization, including normalization, discretization, and feature selection techniques for different ML algorithms, while taking into consideration the size of the feature space and the performance of the AES. Accordingly, we show that the appropriate feature optimization techniques can reduce the dimensions of features, thus, contributing to the efficient training and performance improvement of AES.

1. Introduction

Generally, essay scoring is performed manually by skilled assessment experts. However, when essays are scored manually, there are a couple of limitations. First, it is difficult to acquire consistent results from the scoring, because of human errors and biased preconceptions. Second, it requires a considerable amount of time and effort in scoring. Third, it is impractical for humans to provide a detailed analysis or individual feedback. Consequently, there is growing interest in a computerized system that can automatically assess essays, since the system could potentially assist or even replace human assessors.

So far, most AES approaches have tried to find an appropriate ML algorithm and have focused on finding useful features for training the ML algorithm for AES [1–5]. In this paper, we attempt to use all of the features, which have been proven to be useful in training the AES system, and analyze the effects of integrating those features. However, if the vast numbers of features are used for training the system all together, the following problems may arise.

(1) The limited number of training data does not properly represent the expanded high-dimensional feature space; thus, optimized training cannot be performed.

(2) The vast number of features must be considered at the same time; thus, an increase in training time is required.

The phenomenon (1) is referred to as the "curse of dimensionality." Most features used for AES have values of integers or real numbers, and there are hundreds of features. Thus, a system using these features cannot help but fall into the "curse of dimensionality." ML based systems must train systems with high-dimensional data by spending a tremendous of time for training. Therefore, it is essential that the feature space be reduced, so that optimal performance can be obtained, and training can be made more efficient.

In this paper, we experiment with and analyze the effects of three different techniques to reduce the feature space: normalization, discretization, and feature selection. The normalization techniques can transform the different ranges of each feature value into a fixed range, thereby reducing the whole

range of feature values. The discretization techniques combine feature values represented by numbers into corresponding groups and convert feature values belonging to a specific group into one corresponding integer value, thus reducing the number of feature values. The feature selection techniques reduce feature space by selecting and using features that are relevant to answers and features that are easily distinguishable from different samples.

The normalization, discretization, or feature selection techniques unfortunately do not always have positive effects on the performance of the application. An appropriate combination of feature optimization techniques must be selected for corresponding domains with ML algorithms. Our research shows that using the appropriate feature optimization techniques can reduce the dimensions of features and thus result in the efficient training and performance improvement of AES.

The remainder of this paper is organized as follows.

(i) In Section 2, we discuss previous approaches to AES and the reduction of feature space.

(ii) Section 3 presents the ML based AES architecture, ML algorithms for AES, and diverse features for ML based AES.

(iii) In Section 4, feature optimization techniques (mainly normalization, discretization, and feature selection techniques) are described.

(iv) In Section 5, we discuss and analyze the experimental results of various feature optimization techniques.

(v) Finally, in Section 6, we conclude the paper.

2. Related Works

Various studies on AES have taken place. Often, research that is based on the ML approach focuses on exploring novel features and learning methods to improve the performance for essay scoring. Project Essay Grade [1], the first AES study, used multiple regression (MR) and achieved correlation values that were similar to those achieved by human assessors. The study used mostly number based features, including the number of words, phrases, parentheses, apostrophes, commas, periods, colons, and semicolons. Intelligent essay assessor (IEA) [2] assessed only the content of essays that were written by native English speakers, using latent semantic analysis (LSA). Words were in essays as a vector of the semantic space; the vector dimension was reduced by using LSA. The ungraded essays were compared with graded essays, by using the cosine similarity measure; the score for the most similarly graded essay was assigned to the score of the ungraded essay. BETSY [6] evaluated essays using a classifier that is based on Naive Bayes (NB), which employs specific words and phrases as features. The E-rater [7] system used new features that were extracted using natural language processing (NLP) techniques. In 2010 and 2012, a few studies have used the support vector regression (SVR) or ranking SVM ML algorithm to effectively combine various features [3–5].

The previous AES studies avoided using too many features, because various kinds of useful features are not widely known, and the increase in the number of features would increase the training time for ML methods. In order to utilize various and vast amounts of useful features, to efficiently perform AES, the appropriate feature optimization techniques, such as the normalization technique, discretization technique, and feature selection technique, are required.

Several different approaches have been developed to reduce feature space, by using normalization and discretization techniques, thereby improving the performance [8–11]. Shalabi et al. have applied three normalization techniques (min-max, z-score, and decimal point) to image data in the preprocessing step of ML [8]; Jayalakshmi and Santhakumaran have applied various normalization techniques to medical data concerning diabetes diagnosis [9]. Two researches have experimented and compared various normalization techniques and shown that performance improvement is possible, by employing an appropriate normalization technique for the task.

Chmielewski and Grzymala-Busse's study [10] and Dougherty et al.'s study [11] are representative studies for comparing discretization techniques, which can reduce the dimensions of feature values. Chmielewski and Grzymala-Busse proposed a discretization technique, based on entropy [10], and Dougherty et al. proposed a discretization technique, which can apply to the NB based supervised ML method [11]. Both studies have compared their proposed approaches with previous discretization methods, by employing various sets of experimental data, and their experimental results have shown that their proposed approaches have significantly improved the classification accuracy.

There have also been various approaches to reducing the dimensions of high-dimensional data, by selecting appropriate features from the vast number of possible features in many domains, by using good feature selection techniques [12–15]. Yang and Pedersen experimented and compared feature selection techniques, including document frequency, information gain, mutual information, chi-square, and term strength in the domain of the text categorization [12]. Kakkonen et al. succeeded in reducing the dimensions of feature space by using LSA, PLSA, and LDA techniques in the AES domain [13]. Yu and Liu improved performance in 10 domains: medical, chemical, census, insurance company, spoken letters, and so forth [14]. Kalousis et al. also achieved performance improvement in three domains: proteomics, genomics, and text mining, by reducing the dimensions of high-dimensional data [15].

So far, we have introduced studies on various domains, which have shown performance improvement by reducing dimensionality. In this paper, we introduce a new domain that can reduce the dimension of features by applying feature optimization techniques.

3. Automated Essay Scoring Based on Machine Learning

According to previous studies, there are many features that have been found to be useful for AES. In this section, we provide diverse features for AES and a brief description of useful ML algorithms for AES.

Table 1: It represents the list of features used for learning AES.

Category	Types of features
Basic	(i) Number of characters, words, vocabularies, and sentences (ii) Number of characters, words, and vocabularies without stop word (iii) Number of vocabularies with more than n characters (a) e.g., $n = 1, 2, 3, 4, 5, 10, 20, 30$ (iv) Number of vocabularies with more than n characters and below m characters (a) e.g., $(n, m) = (1, 5), (1, 10), (2, 5), (2, 10), (6, 10)$ (v) Number of vocabularies per frequency of word (vi) Frequency of the most frequent words without stop word (vii) Square of the number of vocabularies (viii) Average length of word and sentence (ix) Variance of sentence length (x) Average distance between the same words and lemmas (xi) Number of POS types (xii) Average number of POS types per sentence (xiii) Average frequency of word per POS type (xiv) Maximum frequency of word per POS type (xv) Ratio of each POS type (by word and character) (xvi) Number of words and vocabularies per POS type
Dictionary	(i) Ratio of words and vocabularies in each dictionary (a) Elementary, middle, and GRE dictionary (ii) Ratio of advanced words and vocabularies (a) Number of words in elementary and middle dictionary/number of words in GRE dictionary
n-gram	(i) Number of n-gram types (word bigram, POS trigram) (ii) Maximum frequency of n-gram (iii) Average frequency of n-gram types (iv) Average frequency, ratio of n-gram type appeared over n times (v) Ratio of n-gram type appeared over n times (a) e.g., $n = 2, 3, 4, 5, 10, 20$ (vi) Ratio of n-gram type appeared over n times and below m times (a) e.g., $(n, m) = (2, 5), (2, 10), (6, 10)$ (vii) Average, maximum, minimum, and variance of perplexity of word or POS sequence (viii) Subtraction of maximum and minimum perplexity of word or POS sequence (ix) Number of sentences below perplexity threshold
Advanced NLP	(i) Average number, maximum frequency, ratio of compound nouns, noun phrases, and named entities per sentence (ii) Frequency of discourse marker (iii) Weighted sum of discourse marker (iv) Number of mechanic grammatical errors (v) Number of pattern grammatical errors

3.1. Features of AES. In this study, we include most features that have been proven to be useful for AES; our newly proposed features, including advanced NLP techniques, are also used in conjunction.

The frequency, average, and ratio of characters, words, and sentences are used for the basic features. Under the assumption that the distribution of the part-of-speeches (POSs) is different, according to the essay grades, we also used the features relating to POS. The level of vocabulary usage is evaluated by using external resources, including elementary and middle dictionaries and the dictionary for the graduate record examination (GRE), and is also used for features. The various features relating to n-gram are also used, after being extracted from the lexical based language model and the POS based language model, which are constructed from large amount of external corpora. The naturalness of each sentence is measured according to its perplexity and is also used for features. The numbers of compound nouns, noun phrases, named entities, discourse markers, and grammatical errors are calculated by using advanced NLP techniques, which are also used for features.

The number of features used in our study exceeds 300 and is represented in groups, in Table 1. Representing an essay with such a large number of features that have a diverse range of feature values would mean that the vector, including all features, would yield high-dimensional data. Therefore, it is mandatory that a process should be developed to reduce dimensions, via feature optimization, in order to efficiently train such a large number of features.

3.2. Machine Learning Algorithms for AES. So far, there have been many attempts to utilize ML algorithms for AES. The ML algorithms for AES can be classified into two categories: regression and classification. Regression is used for predicting

or estimating the corresponding target value given the feature values of the specific instance by analyzing the relationships between the feature values and the target value. Classification is used to determine the corresponding category of the specific instance.

The big difference between the two approaches is in the characteristics of the target values. The target values for regression are ordered and continuous, while the target values for classification are unordered and discretized. For AES, the regression approaches try to predict the continuous target score based on feature values that represent the characteristics of essays; the classification approaches try to identify the categorized score under the assumption that each essay belongs to a specific category.

In this study, we have experimented and compared two different regression based ML algorithms and two different classification based ML algorithms for AES and have tried to identify the appropriate ML algorithms for AES by performing different experimentations.

In this section, we provide a brief description of the four ML algorithms, which are applied in our experimentation for AES: MR model [16], maximum entropy (ME) model [16], support vector machine (SVM) [17], and SVR [18]. We selected these ML algorithms for the following reasons.

(i) MR is the most widely used algorithm in AES research.

(ii) ME achieves good results in document classification, using many features.

(iii) SVM is the best algorithm for solving various classification problems.

(iv) SVR applies regression to SVM, and it is expected that SVR may have the benefits for both regression and classification.

3.2.1. Multiple Regression Model. The MR model [16] is the oldest [1] and the most widely used approach among ML based AES studies. This model tries to find w_i that satisfies the expression in (1), under the assumption that the relationship between the feature values and target values is linear:

$$y = \sum_{i=0}^{N} w_i \times f_i. \tag{1}$$

Each instance of the training data represents an essay, and an essay is represented by N feature values f_i (e.g., the length of essay and the number of advanced vocabularies) and the target value y (i.e., the score of the essay). The real feature values denoted by $f_1 \sim f_N$ consist of N possible different values, and f_0 always has the value, 1, reflecting the constant weight, w_0. The training process is to find the optimal weight, w, for predicting the essay scores most precisely in all of the training data. The sum-squared error technique is widely used for finding optimal weight, w.

3.2.2. Maximum Entropy Model. The ME model [16] is also referred to as the multinomial logistic regression model. The ME model is widely used in many applications as an alternative to the NB model, because there is no independent assumption, unlike in the NB, in which there is a strong assumption that each feature is independent; NB was once applied for AES [6]. We have chosen the ME model for the AES experiment, because the model has been a good classification algorithm that showed high performance in various tasks, including text categorization, POS tagging, and Named Entity Recognition [19–22].

The fundamental formula for the ME model is represented in expression (2). This model tries to find the corresponding probability value that belongs to class c (grade of the essay) when the specific instance x (essay) is given. The specific instance x is assigned to the class c that has the highest probability value among the classes with calculated probability values. The specific instance x is represented by a combination of various features denoted by f_i, and there are always exits that correspond with the weight w_i, depending on the feature f_i. The normalization factor z is needed to convert the exponential value into a true probability:

$$p(c \mid x) = \frac{1}{z} \exp \sum_i w_i f_i. \tag{2}$$

The final formula of the ME model is represented by expression (3). The w_i in (2) is converted to w_{ci} in (3), because the feature i for each class c has a different weight. The function, f_i, is an indicator function, since only 0 and 1 are used for feature values in ME. The normalization constant z in the denominator is calculated by making a summation of computation results for every class. The feature values for the essay, including the length of the essay and the number of advanced vocabularies, are transformed into 0 or 1 by the indicator function, multiplied by weights, and then combined into the total value by the log linear model. The probability of assigning the grade c to the given essay x is calculated for each grade separately, and the final grade is produced for the essay score based on the value of the probability:

$$p(c \mid x) = \frac{\exp\left(\sum_{i=0}^{N} w_{ci} f_i(c, x)\right)}{\sum_{c' \in C} \exp\left(\sum_{i=0}^{N} w_{c'i} f_i(c', x)\right)}. \tag{3}$$

3.2.3. Support Vector Machine. The SVM [17] is one of the most representative ML algorithms for classification. The SVM algorithm represents each instance by a dot in the high-dimensional space with the same number of dimensions for the number of features and finds the most appropriate hyperplane to properly separate the dots.

In Figure 1, two classes are represented by white dots and black dots, indicating the training instances described in the two-dimensional space; there are only two features x_1 and x_2. The hyperplane x is determined by w and b and is represented by the following expression:

$$w \cdot x - b = 0. \tag{4}$$

In the training process for SVM, it tries to find a w for which the distance is maximized from the nearest instance. The class of the new instance is determined by identifying the

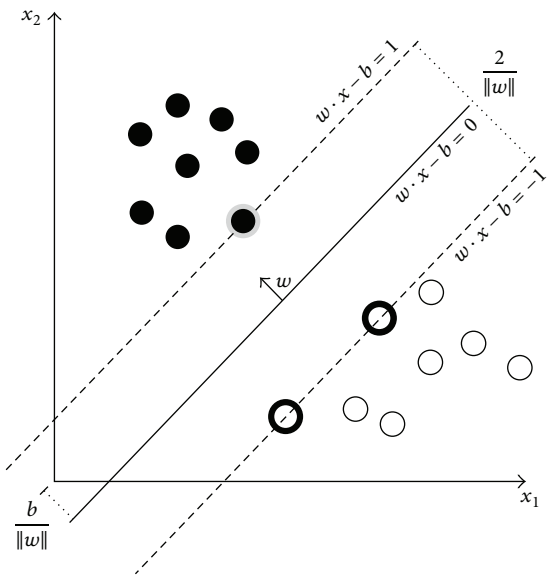

Figure 1: Concept of the support vector machine (http://en.wikipedia.org/wiki/Support_vector_machine).

appropriate space among the possible spaces, separated by the hyperplane.

The task for AES is to classify an essay into one of six grades, by utilizing more than 300 different kinds of features. In order to find many hyperplanes to separate the six different grades into more than 300 high-dimensional spaces, much time is needed for optimization. Therefore, the feature space must be reduced through feature optimization techniques.

3.2.4. Support Vector Regression. SVR [18] is the algorithm for applying SVM to regression. According to Li and Yan's study, it is known as the best ML algorithm for AES [4]. Thus, we chose SVR for our experimentation:

$$f(x) = \langle w, x \rangle + b \quad \text{with } w \in X, b \in \mathbb{R}. \tag{5}$$

$$\text{minimize} \quad \frac{1}{2}\|w\|^2$$
$$\text{subject} \quad \begin{cases} yi - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_1 \leq \varepsilon. \end{cases} \tag{6}$$

In the training process for SVR, it tries to find $f(x)$, represented in expression (5), which satisfies the conditions represented by expression (6). It tries to find the hyperplane, which makes w the smallest against the instances for which the distance from the hyperplane is less than ε.

3.3. System Architecture of AES. Figure 2 shows the architecture of the AES system, based on ML methods. There are two work processes: learning and prediction. Labeled training data is input into the NLP module in the learning process, and unlabeled data is also input into the NLP module in prediction process. Identical processing is applied from the natural language process to the feature converting process in the learning and prediction processes. First, essays are input into the NLP module, which includes the sentence breaker, tokenizer, lemmatizer, POS tagger, noun phrase extractor, and named entity extractor. The results from the NLP module are delivered to the feature extractor.

The feature extractor uses all natural language processed information to extract features that accurately represent essays characteristics. More than 300 diverse features can be employed for training the AES model. The features can be classified into six categories, as shown in Table 1. There are features relating to length (e.g., the number of characters, words, and sentences), features relating to ratio (e.g., the proportion of a specified discourse marker or POS tag), and manufactured features from the result on NLP or statistics. These features have been used in recent studies.

The feature optimizer selectively performs normalization, discretization, and feature selection for optimal performance. The labeled essays are input for training into the learner module, and the unlabeled essays are input for prediction into the predictor module. The learner module is used to create the training model, and the training model is used in the predictor module to calculate the final grade (score) of an essay.

In this paper, we focus on ML dependent feature optimization techniques for reducing dimensions of features, because the performance of AES is dependent on ML algorithms and feature optimization.

4. Feature Optimization

4.1. Normalization. In research studies based on ML, features that have a variety of types and ranges are used. The number of words and the ratio of words in an elementary dictionary that are used by the feature values in AES also have different ranges. When we use these features directly in the AES system, it is possible to perform nonoptimized learning. Therefore, we must convert all feature values into a fixed range.

Although there are a number of normalization methods, we selected the two most commonly used methods: min-max and z-score normalization. Min-max normalization is generally known for achieving the best performance, according to related works [8, 9]:

$$\text{min-max}\left(x_{if}\right) = \frac{x_{if} - \min(f)}{\max(f) - \min(f)}. \tag{7}$$

$$z\text{-score}\left(x_{if}\right) = \frac{x_{if} - \mu_f}{\sigma_f}. \tag{8}$$

In Formula (7), the min-max normalization converts all feature values into a fixed range, while keeping the origin interval. In Formula (8), the z-score normalization converts all feature values into a z-score for normal distribution, without keeping the origin interval. Symbol f is the set of all feature values for one feature and is extracted from the training data; x_{if} is one feature value to normalize.

4.2. Discretization. The discretizing feature values simplify data representation by converting continuous feature values

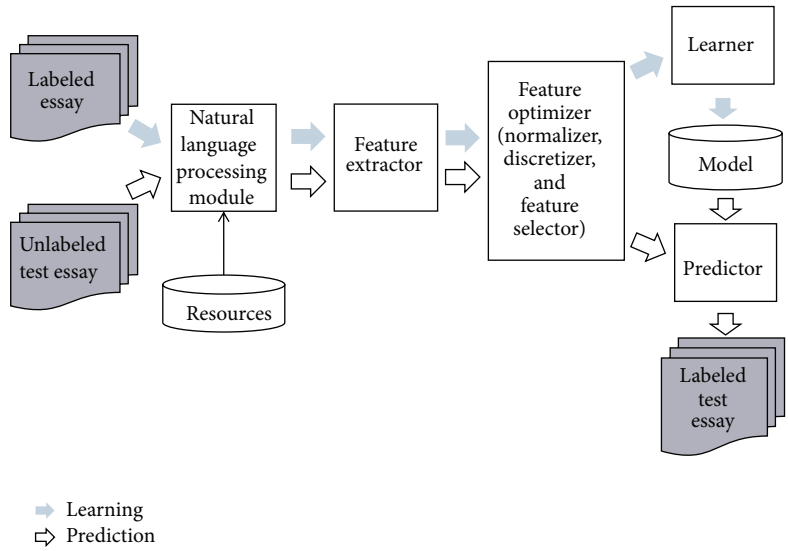

Figure 2: Automated essay scoring system architecture based on machine learning.

that belong to a specific range into a certain feature value. This process makes the feature values suitable for ML. The type of discretization method, the range of the discretized section, and the number of discretized sections all affect the performance of the ML system.

In our study, we used two simple discretization methods: discretization by instance number (DIN) and discretization by feature value (DFV). DIN assigns the same number of instances to each section, after sorting them by the feature value. DFV converts all feature values that belong to the specific range into one feature value, after setting the range of feature values for each section. DIN is advantageous when the distribution of feature values is uniform. DFV is beneficial when the distribution of feature values is normal.

4.3. Feature Selection. Feature selection filters out noisy features and discovers the optimal feature set in ML. Even though we determine a feature set with appropriate intuition and assumptions, some features in the set may produce a negative effect. Further, too many features can hinder learning or delay it. In this work, we compare three feature selection methods: correlation (COR) [14], information gain (IG) [14], and minimal-redundancy-maximal-relevance (mRMR) [23]. Feature selection is performed based on the relevance between the feature value and the golden score. We select the top n number of features by using the high relevance order between the feature value and the golden score. Then, we use Pearson's correlation coefficient and information gain to measure the relevance. These are popular measures that can calculate the relevance between the two sets.

Formula (9) is a correlation formula between the random variables X, Y. In this case, X is the golden score and Y is one feature. x_i and y_i are case of each sample and $\overline{x}$ and $\overline{y}$ are mean of random variables X, Y:

$$\mathrm{Cor}(X,Y) = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2\sum_{i=1}^{n}(y_i-\overline{y})^2}}. \quad (9)$$

To calculate the information gain, we calculate the entropy first. Entropy of random variable X and conditional entropy of X, given Y, are defined as

$$H(X) = -\sum_i P(x_i)\log_2(P(x_i)),$$
$$H(X \mid Y) = -\sum_j P(y_j)\sum_i P(x_i \mid y_j)\log_2\left(P(x_i \mid y_j)\right). \quad (10)$$

The information gain formula between random variables X, Y is defined as

$$\mathrm{IG}(X \mid Y) = H(X) - H(X \mid Y). \quad (11)$$

Used as another feature selection method, mRMR considers the dependency between the features as well as the relevance between the feature value and the golden score. The mRMR is calculated as formula (12). It is used to find the optimal feature set, S. The sum of the mutual information of feature i and the golden score c should be the maximum, and the sum of the mutual information between each feature should be the minimum:

$$\mathrm{mRMR} = \max_S\left[\frac{1}{|S|}\sum_{f_i\in S} I(f_i;c) - \frac{1}{|S|^2}\sum_{f_i,f_j\in S} I(f_i;f_j)\right]. \quad (12)$$

5. Experiments

5.1. Experimental Setup. For our experiment, we used the essay practice data that covered 13 topics. The correct answer was constructed based on scores provided by many human experts, who were hired. More than two human experts assign scores ranging from 0 to 6 to each essay. For each topic, Table 2 displays the number of essays (4677), the average number of words in an essay, and the correlation between the

Table 2: Descriptions of essay practice data.

Topic	Number of essays	Average length	Human cor.
Small town or big city	266	362.8	0.5202
Parents are the best teachers	774	345.7	0.5308
Qualities of a good neighbor	242	355.8	0.5664
Positive influence of TV or movies	227	354.1	0.5127
Reasons for attending college	241	310.5	0.5527
Dining at a restaurant versus home	385	360.1	0.4567
Why do some people go to museums	221	347.8	0.6096
Best ways to reduce stress	156	356.4	0.4420
Qualities of good parents	211	357.8	0.6260
Achieving success by working hard	370	378.5	0.4408
Rejection of the invite	528	72.8	0.7287
Report on FORBES MEDIA	528	148.8	0.6438
Hiring a family member or friend	528	207.0	0.7319
Total	4677	281.9	0.6088

two human experts. The total correlation between the human experts was found to be 0.6088. We used the average value of the two grades as a golden score for our experiment. Since the difficulty in grading varies dependently on the topic, we performed a 10-fold cross-validation for each topic. We used Pearson's correlation coefficient between the system output and the golden score as a measure of performance evaluation and used a microaveraging method to evaluate the total performance of an entire essay.

We have conducted a preliminary experiment considering all three feature optimization methods, in order to roughly determine all base parameter values prior to conducting the main experiments for investigating the effect of each feature optimization method. If we apply each individual optimization method separately without combining other optimization methods, the performance of AES deteriorates, and the effects of the feature optimization methods disappear. For example, we do not obtain the effect of the normalization method when we do not apply the discretization method and the feature selection method altogether. For this reason, we have obtained base robust parameter values, which indicated satisfactory performance for most cases, by conducting preliminary experiments (i.e., for normalization: min-max; for discretization: DFV with 10 sections; for feature selection: 80 feature selections with correlation). In the following experiments, the intended feature optimization method is tested and modified with these base parameter values.

We used the following three ML packages for experiment:

(1) MR: GNU Scientific Library (http://www.gnu.org/software/gsl/),

(2) ME: Maximum Entropy Modeling Toolkit for Python and C++ (http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html),

(3) SVM and SVR: LIBSVM (http://www.csie.ntu.edu.tw/~cjlin/libsvm/).

Table 3: Comparison of normalization methods (correlation).

Normalization	MR	ME	SVM	SVR
None	0.1702	0.3166	0.2839	0.2980
min-max	0.7383	0.7160	0.7675	0.7756
z-score	0.7315	0.7289	0.7701	0.7747

5.2. Experiment for Normalization. For each of the four ML algorithms (MR, ME, SVM, and SVR), we compared the following three normalization methods: none, min-max, and z-score. The other feature optimization techniques and parameters were applied equally (DFV with 10 sections, 80 features selected with correlation).

Although the difference between the normalization methods is insignificant, the difference in performance between nonnormalization and normalization was noteworthy (Table 3). As a result, the normalization process was determined to be useful for AES using ML algorithms.

5.3. Experiment for Discretization. We compared the following three discretization methods: none, DIN, and DFV. For each discretization method, we performed experiments on different numbers of sections (2–16, in increments of 2) to determine whether the number of discretized sections affected the performance. We applied other optimization techniques and parameters equally. Min-max normalization was performed, and 80 features were selected, using correlation.

We have performed discretization experiments using four different ML algorithms. As shown in Figure 3, the experimental results show that ML algorithms vary considerably in performance. In case of MR, we could obtain a better performance without performing the discretization method. This is because MR treats the real numbers for feature values. If we convert the feature values into specific integers, by applying a discretization method, this would yield bad effects on training for the machine learning. In the case of ME, the performance is rarely different between when a discretization method is applied and when it is not. Any method of discretization or any number of sections for discretization hardly made a difference. A discretization method did not provide any effects on the performance of ME, because ME uses an indicator function, which internally converts its feature values to 0 or 1. The cases that used SVM improved in performance after discretization. Since SVM is an algorithm used to find a support vector that properly separates each instance, there seems to be an improvement in performance when instances that have similar feature values were shifted to one side. The cases that used SVR also improved in performance

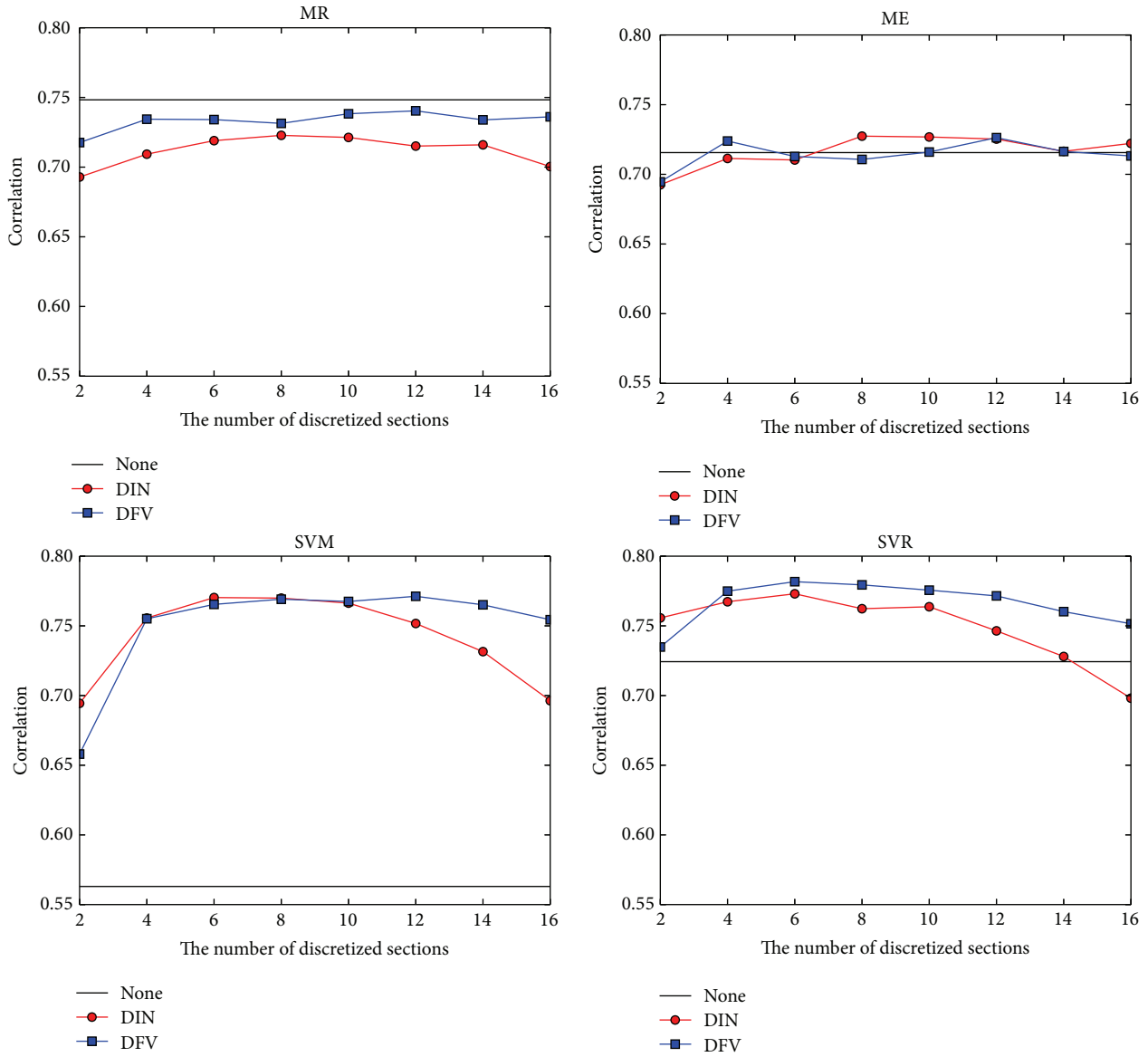

Figure 3: Comparison of discretization methods.

after discretization, although the improvement is not as good as SVM cases.

In cases of SVM and SVR, which show effects of the discretization methods ideally, there are large performance differences for DIN and DFV discretization methods, between a number of sections. We have found that the performance of DFV was better than the performance of DIN. This is because the original distribution of feature values is maintained for the DFV method, while, for the DIN method, the same number of feature values per section was assigned compulsively.

We have also found that the performance decreases as the number of sections increases. This is because too many sections cause the decreased number of feature values per section, the sparseness problem in some cases, and the diminished effects of discretization.

The experimental results show that MR and ME did not yield a better performance by performing a discretization method, but SVM can get dramatic improvements in performance. SVR also improved in performance, although the improvement is not as good as SVM cases.

5.4. Experiment for Feature Selection. We compared the following four feature selection methods: none, COR, IG, and mRMR. For each feature selection method, we performed experiments on different numbers of features (20–160, in increments of 20). We applied other optimization techniques and parameters equally (min-max normalization, DFV with 10 sections).

Figure 4 shows the different characteristics of each ML algorithm. For MR, when the number of features is excessive, its correlation is presented as 0; in other words, the training was unsuccessful and the model for prediction was not created. When we used more selected features, the performance decreased in ME; we could observe a similar pattern in

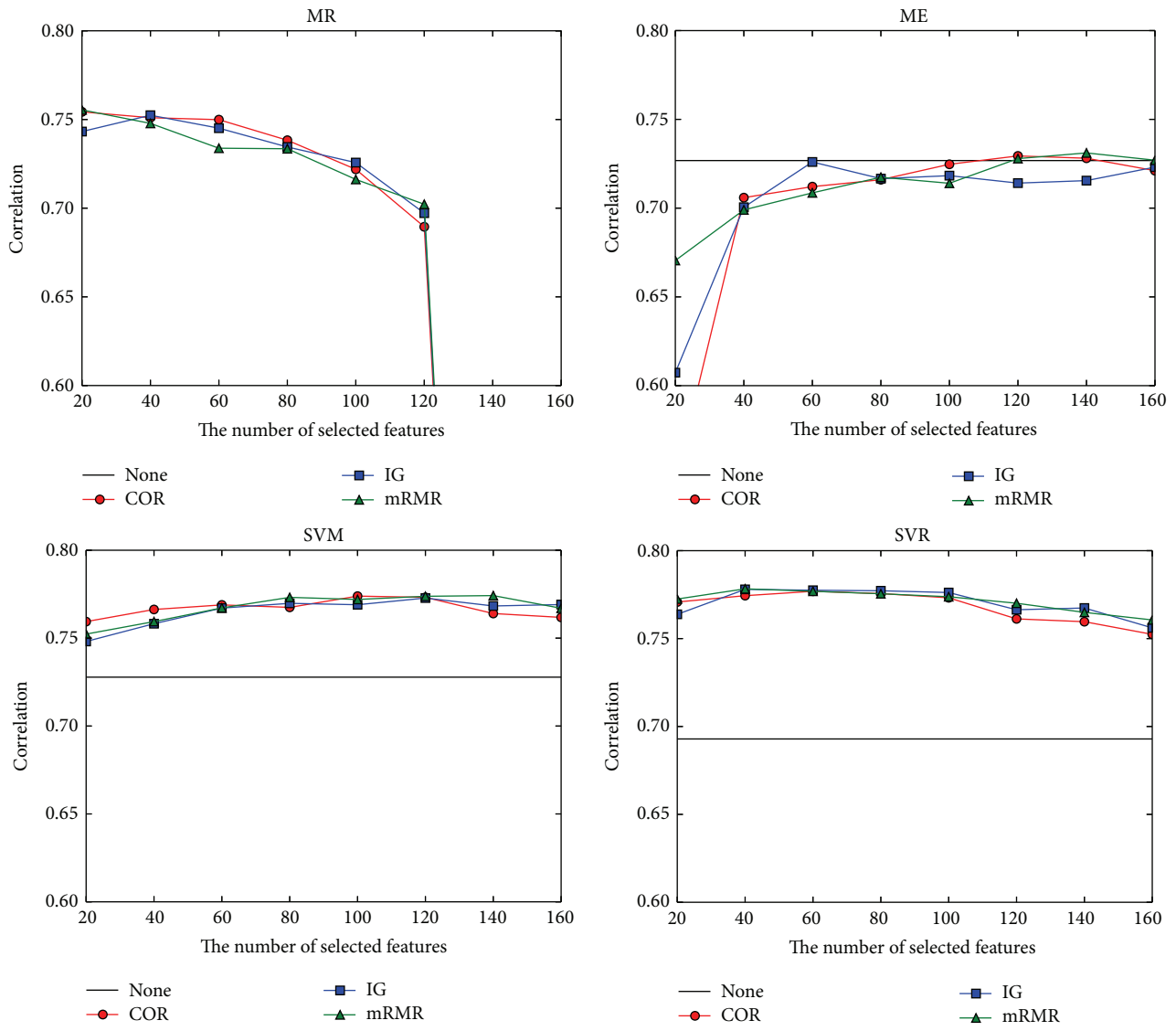

Figure 4: Comparison of feature selection methods.

SVM and SVR. The more features we selected, the higher the chance of selecting noisy features. In the majority of cases, we can achieve an optimal performance when features are selected using only correlations, and the performance of mRMR is at the lowest.

Using features that consider only the relevance of the golden score is more advantageous than using selected features that consider both the relevance of the golden scores and dependencies between features. We assume that the dependencies between features would yield bad side effects, because of the different characteristics of features used in AES.

5.5. Effective Features. The AES system proposed in this paper automatically constructs the optimized set of features by selecting features from the training data. It is difficult for us to say that a specific feature is always effective, because the set of selected features is different according to the experimental settings, subjects, or folds for cross-validation. In this section, we try to identify the effective features for AES by examining the generally selected features in most cases. In order to do this, we have experimented with 130 different training procedures and tests with base parameters. The list of features shown in Table 4 was mostly selected in the 130 training procedures.

5.6. Experiment for Efficiency Improvement. In this experiment, we used a server with two AMD Opteron 4180 (6 core) processors; thus, 12 cpu cores can be employed. A 32-GigaByte memory and 64-bit Debian operating system was also employed for our experiment. Because the AES system is a complicated system including various NLP processing modules implemented by many programming languages, we tried to use serviceable resources, such as process cores and memories, as much as possible to maintain system efficiency.

Table 4: Effective feature list.

Number of times selected	Feature name	Meaning of feature
130	posNumINVoca	Number of vocabularies with IN POS tag
130	posNumIN	Number of words with IN POS tag
130	lmPosTrigramVoca	The number of different POS trigrams
130	lmPosTrigramOccMore3	The ratio of POS trigrams occurred more than 3
130	lmPosTrigramOccMore2Less5	The ratio of POS trigrams occurred more than 2 but fewer than 5
130	lmPosTrigramOccMore2Less10	The ratio of POS trigrams occurred more than 2 but fewer than 10
130	lmPosTrigramOccMore2	The ratio of POS trigrams occurred more than 2
130	lmNumVoca4Root	Biquadrate of the number of vocabularies
130	lmNumVoca	The number of vocabularies
130	lmLexWordOccMore5	The number of different words occurred more than 5
130	lmLexWordOccMore4	The number of different words occurred more than 4
130	lmLexWordOccMore3	The number of different words occurred more than 3
130	lmLexWordOccMore2Less5	The number of different words occurred more than 2 but fewer than 5
130	lmLexWordOccMore2Less10	The number of different words occurred more than 2 but fewer than 10
130	lmLexWordOccMore2	The number of different words occurred more than 2
130	lmLexWordOccMore1Less5	The number of different words occurred more than 5
130	lmLexWordOccMore1Less10	The number of different words occurred more than 10
130	lmLexWordOccMore1	The number of different words occurred more than 1
130	lmLexBigramVoca	The number of different lexical bigrams
130	lmLexBigramOccMore2Less5	The ratio of lexical bigrams occurred more than 2 but fewer than 5
130	lmAvgLexWordDistance	The average distance of same words
130	lmAvgLemmaWordDistance	The average distance of same lemmas
130	cNumWordLen8	The number of words whose length is more than 8 characters
130	cNumWordLen7	The number of words whose length is more than 7 characters
130	cNumWordLen6	The number of words whose length is more than 6 characters
130	cNumWordLen5	The number of words whose length is more than 5 characters
130	cNumWord	The number of all words
130	cNumNotStopWord	The number of all words except stop words
130	cNumNotStopVoca	The number of all vocabularies except stop words
130	cNumMidd	The number of words in the intermediate dictionary
130	cNumElem	The number of words in the elementary dictionary
130	cNumChar	The number of all characters
130	cCharNotStopWord	The number of all characters except stop words
129	posNumNN	The number of words with NN POS tag
129	lmLexBigramOccMore2Less10	The ratio of lexical bigrams occurred more than 2 but fewer than 10
129	lmLexBigramOccMore2	The ratio of lexical bigrams occurred more than 2
128	posNumJJVoca	The number of vocabularies with NN POS tag
128	posNumJJ	The number of words with NN POS tag
126	cNumWordLen10	The number of words whose length is more than 10 characters
125	posNumNNSVoca	The number of vocabularies with NNS POS tag

For extracting various features, we utilized the maximum threads by using openMP; for training and testing, we utilized all 12 cores by employing the multiprocessing module from the python standard library. In order to compare the efficiency of the AES system with different numbers of features, we used the same feature optimization techniques (the min-max normalization method, DFV with 10 sections, and the feature selection method based on correlation) and measured the time required for training and testing the AES system, with a different number of features.

We performed all four different ML algorithms introduced in Section 3.2. The experimental setup for efficiency comparison was the same as the experimental setup for feature selection. We have compared the execution times for

when the numbers of features are 20, 40, 60, 80, 120, 140, 160, and all features (i.e., 316), respectively. We did not experiment for the case when the numbers of selected features are between 160 and 312, because it turned out to be clear that the execution time was linearly increased as the same ratio increased from 20 features to 160 features.

As shown in Figure 5, the execution time for all ML algorithms, with the exception of MR, is linearly increased, according to the number of features. If we can reduce the number of features by applying a feature optimization technique without decreasing the classification accuracy, we can acquire a large gain in efficiency. For MR, there was no difference in time. This may be interpreted to be because the execution time did not increase, even with the increased number of features; however, this is untrue. If the number of features is increased, the training with the increased number of features failed, and the required time for training is almost 0; thus, there was no increased time.

In addition, we will discuss the tradeoff between efficiency and effectiveness of the AES system using feature optimization methods. According to previous experimental results, we have shown that the feature optimization methods improve the effectiveness of AES system. However, the feature optimization methods would decrease the efficiency of AES system, because of the increased processing time for feature optimization, even though the training time and testing time were reduced due to the reduced number of features. Although most of the feature optimization methods do not require an excessive amount of time, some specific feature optimization methods, such as mRMR, require processing times, to some degree. Accordingly, we have considered the tradeoff between the reduced time of training and testing and the increased processing time of feature optimization; thus, we have to select a proper method of feature optimization for practical purposes.

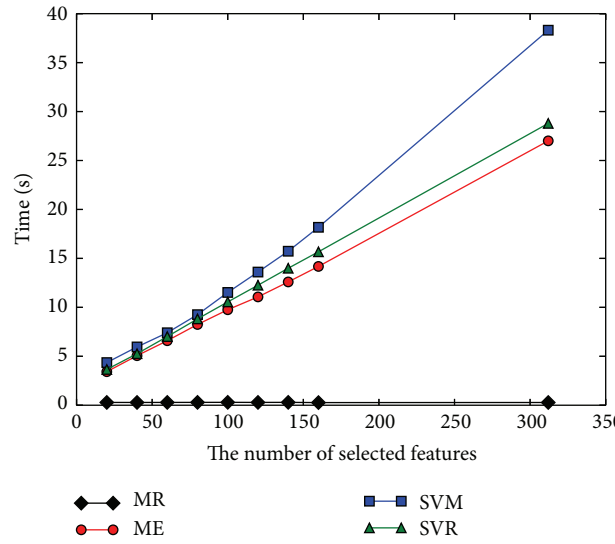

Figure 5: Efficiency comparing.

6. Conclusions and Future Work

This paper presented feature optimization techniques consisting of appropriate normalization, discretization, and feature selection methods that can be applied to the ML based AES system. We have shown that both the effectiveness and efficiency of the system can be improved. These feature optimization techniques reduce the high-dimensional feature space. As a result, the performance is improved and the training time also decreased. By experimenting and analyzing the relationship between the ML algorithms and the feature optimization techniques in the domain of the AES with a large number of English essay data, we have obtained many useful findings.

We can summarize the results of the experiments with the following four main discoveries.

(i) The different combinations of feature optimization techniques give rise to large variation in performance.

(ii) A normalization technique is essential for every ML method. There is a 2.3-fold performance difference in the minimum and 4.3-fold performance difference in the maximum between a ML method employing a normalizing technique and a ML method that is not employing a normalizing technique.

(iii) The discretization technique is not useful for the MR or ME model. On the contrary, a discretization technique is mandatory for SVM, and the performance of SVM is improved when a discretization technique and an appropriate number of sections are used.

(iv) Because the reduced number of features is useful for the MR model, an appropriate feature selection technique is required for MR. On the contrary, ME model can utilize the large number of features; thus, the feature selection technique is relatively less important for ME. For the SVM and SVR model, the number of features causes large variations in performance; therefore, the proper number of features must be determined when a feature selection technique is used.

Experimental results of all combinations of parameter values showed that the best performance was acquired when we used the SVR ML algorithm, z-score normalization method, DFV discretization method with six sections, and the 100 features selected from the correlation feature selection method. As a result, the final performance of AES reached the 0.7852 correlation value, which was much better than the 0.6088 correlation value obtained from two human experts.

For future works, we plan to apply these feature optimization techniques to another domain and intend to show that the feature optimization techniques are also useful for improving both the effectiveness and efficiency of the system.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2012M3C4A7033344).

References

[1] E. B. Page, "Statistical and linguistic strategies in the computer grading of essays," in *Proceedings of the Conference on Computational Linguistics (COLING '67)*, pp. 1–13, August 1967.

[2] P. W. Foltz, D. Laham, and T. K. Landauer, "The intelligent essay assessor: applications to educational technology," *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, vol. 1, no. 2, 1999.

[3] X. Peng, D. Ke, Z. Chen, and B. Xu, "Automated chinese essay scoring using vector space models," in *Proceedings of the 4th International Universal Communication Symposium (IUCS '10)*, pp. 149–153, IEEE, October 2010.

[4] Y. Li and Y. Yan, "An effective automated essay scoring system using support vector regression," in *Proceedings of the 5th International Conference on Intelligent Computation Technology and Automation (ICICTA '12)*, pp. 65–68, IEEE, January 2012.

[5] H. Chen, B. He, T. Luo, and B. Li, "A ranked-based learning approach to automated essay scoring," in *Proceedings of the 2nd International Conference on Cloud and Green Computing (CGC '12)*, pp. 448–455, IEEE, November 2012.

[6] L. M. Rudner and T. Liang, "Automated essay scoring using Bayes' theorem," *The Journal of Technology, Learning, and Assessment*, vol. 1, no. 2, pp. 1–22, 2002.

[7] J. Burstein, "The e-rater scoring engine: automated essay scoring with natural language processing," in *Automated Essay Scoring: A Cross-Disciplinary Perspective*, M. D. Shermis and J. Burstein, Eds., Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 2003.

[8] L. A. Shalabi, Z. Shaaban, and B. Kasasbeh, "Data mining: a preprocessing engine," *Journal of Computer Science*, vol. 2, no. 9, pp. 735–739, 2006.

[9] T. Jayalakshmi and A. Santhakumaran, "Statistical normalization and back propagation for classification," *International Journal of Computer Theory and Engineering*, vol. 3, no. 1, pp. 1793–8201, 2011.

[10] M. R. Chmielewski and J. W. Grzymala-Busse, "Global discretization of continuous attributes as preprocessing for machine learning," in *Proceedings of the 3rd Internation Workshop on Rogh Sets and Soft Computing*, pp. 294–301, 1994.

[11] J. Dougherty, R. Kohavi, M. Sahami et al., "Supervised and unsupervised discretization of continuous features," in *Proceedings of the 12th International Conference on Machine Learning (ICML '95)*, pp. 194–202, 1995.

[12] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the 14th International Conference on Machine Learning (ICML '97)*, vol. 97, pp. 412–420, 1997.

[13] T. Kakkonen, N. Myller, E. Sutinen, and J. Timonen, "Comparison of dimension reduction methods for automated essay grading," *Educational Technology & Society*, vol. 11, no. 3, pp. 275–288, 2008.

[14] L. Yu and H. Liu, "Feature selection for high-dimensional data: a fast correlation-based filter solution," in *Proceedings of the 20th International Conference on Machine Learning (ICML '03)*, vol. 3, pp. 856–863, 2003.

[15] A. Kalousis, J. Prados, and M. Hilario, "Stability of feature selection algorithms: a study on high-dimensional spaces," *Knowledge and Information Systems*, vol. 12, no. 1, pp. 95–116, 2007.

[16] D. Jurafsky and J. H. Martin, *Speech & Language Processing*, Pearson Education India, Gurgaon, India, 2000.

[17] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 2000.

[18] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems*, pp. 155–161, MIT Press, 1997.

[19] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *IJCAI-99 Workshop on Machine Learning for Information Filtering*, vol. 1, pp. 61–67, 1999.

[20] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, vol. 1, pp. 133–142, Philadelphia, Pa, USA, 1996.

[21] H. L. Chieu and H. T. Ng, "Named entity recognition: a maximum entropy approach using global information," in *Proceedings of the 19th international conference on Computational linguistics*, vol. 1, pp. 1–7, Association for Computational Linguistics, 2002.

[22] A. L. Berger, V. J. della Pietra, and S. A. della Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[23] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.