Learning to Estimate Slide Comprehension in Classrooms with Support Vector Machines

Nimit Pattanasri, Masayuki Mukunoki, and Michihiko Minoh, *Member*, *IEEE Computer Society*

Abstract—Comprehension assessment is an essential tool in classroom learning. However, the judgment often relies on experience of an instructor who makes observation of students' behavior during the lessons. We argue that students should report their own comprehension explicitly in a classroom. With students' comprehension made available at the slide level, we apply a machine learning technique to classify presentation slides according to comprehension levels. Our experimental result suggests that presentation-based features are as predictive as bag-of-words feature vector which is proved successful in text classification tasks. Our analysis on presentation-based features reveals possible causes of poor lecture comprehension.

Index Terms—Lecture analytics, lecture comprehension, learning skills, SVM.

1 Introduction

The success in classroom teaching is at best measured by comprehension level. Monitoring comprehension is a complicated task as it requires sensitive observation skills as well as active learning environment. On one hand, insensitive instructors may fail to monitor and interact with students effectively. On the other hand, in-class discussion seems ineffective for checking understanding level when students show no response at all, a problem that arises often in engineering education [6]. In fact, nothing is wrong with those who learn well in a passive way.

We address the problem of comprehension monitoring from a different angle. Rather than guessing about their comprehension level, an instructor asks students to report their own comprehension explicitly (in an anonymous manner). For example, through handheld devices, students give a rating based on how much they understand each presentation slide. This immediate feedback offers a general idea of what is going on in a classroom.

Tuning instructional modes to learning styles—rather than to understanding capability—has been claimed extensively as an effective strategy to enhance learning performance (see a recent review in [12]). Learning styles focus on an idea that different people learn information in different ways. Centered around a concept of experience, Kolb's model [11] characterizes learning styles based on how information is perceived (concrete experience versus abstract conceptualization) and then processed (reflective observation versus active experimentation) while Flaming's model [7] concentrates on perception modalities, resulting in four sensory styles: visual (learning by seeing), aural (learning by listening), read/write (learning by reading or writing), and kinestheic (learning by doing). However, impressive this categorization may seem, the benefit of matching learning styles with *only* compatible instructional modes, asserted by these theories, is becoming increasingly questionable (see a comprehensive criticism in [19], [4]). As a counter example, Massa and Mayer [14] have shown that students who prefer visual style of learning may perform equally well (or sometimes better) with a verbal mode of instructions, and vice versa. That is, a learning preference does not necessarily dictate actual capability.

Connecting students' comprehension with instructional materials is our departure from learning styles which relate students' performance with personal attributes. We apply a machine learning technique for analyzing students' comprehension gathered in a classroom. The capability to "learn" how students comprehend in a class would shed light on the quality of teaching. This paper offers a method to quantify what makes good lecture materials and presentation. The impact on presentation elements toward students' comprehension, being positive or negative, can be revealed in light of explicit feedback gathered in the classroom. The result of this impact analysis can be used as a guideline for instructional design in future lessons.

2 Estimating Comprehension as a Machine Learning Problem

Impact of lecture materials toward students' comprehension cannot be analyzed unless student models are available. The term "models" means ones that describe the data; in other words, ones that are capable of estimating students' comprehension. This paper assumes presentation slides to be instructional units in a classroom and, therefore, our problem is to estimate comprehension on a slide basis. Topics can be equally a good choice of comprehension units, but slides allow us to associate contents with their presentation styles, making quality analysis on lecture materials possible.

We formulate the problem of estimating comprehension of slides as a supervised machine learning problem. We first

- *N. Pattanasri is with the Department of Social Informatics, Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo, Kyoto 606-8501, Japan. E-mail: nimit@dl.kuis.kyoto-u.ac.jp.*
- *M. Mukunoki and M. Minoh are with the Academic Center for Computing and Media Studies, Kyoto University, Yoshida Nihonmatsu-cho, Sakyo, Kyoto 606-8501, Japan. E-mail: {mukunoki, minoh}@media.kyoto-u.ac.jp.*

Manuscript received 8 Nov. 2010; revised 13 Feb. 2011; accepted 1 Apr. 2011; published online 26 Apr. 2011.
For information on obtaining reprints of this article, please send e-mail to: lt@computer.org, and reference IEEECS Log Number TLT-2010-11-0122.
Digital Object Identifier no. 10.1109/TLT.2011.22.

train a classifier by a machine learner. The input of a learner is a set of m labeled slides $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})$, where $\mathbf{x}^{(i)}$ is a set of n dimensional input features $(x_1^{(i)}, \ldots, x_n^{(i)})$, $y^{(i)}$ is a class of slides (e.g., *understand* or *not understand*), and the output is a learned classifier $g : \mathbf{x}^{(i)} \rightarrow y^{(i)}$. We use this classifier to classify slides for a target student. Training a separate classifier for each student emphasizes individual differences—the only common viewpoint our approach shares with learning-styles theories [12].

Generalizability of statistical inference depends on an assumption that all feedback data of each student must be drawn from the same distribution. This assumption holds to an extent that there is no sudden change in a way that students give rating. This limitation does not mean that the rating given by a student must be uniform throughout a lecture, but the ability to understand lecture materials—especially from a viewpoint of learning skills—remains relatively unchanged over time.

Intuitively, comprehension is affected by lecture contents. To estimate comprehension level, we can use the bag-of-words model as predictive features which have proved successful in text classification [21]. The effectiveness is, however, questionable in our problem setting because new technical terms (features) are often introduced in each lecture. That is, comprehension about the current lecture, encoded in word features, is not always predictive for comprehension about the next lecture. Besides, analysis of word features offers little value for understanding the learning process. Knowing that a particular word or topic has a negative impact on comprehension only informs instructors about how difficult the topic is. In fact, instructors can extract such information directly from students' test scores (i.e., topic comprehension). However, analysis of test scores hardly reveals the source of comprehension problems.

It seems against our intuition at first to use features other than contents to classify slides according to comprehension levels. However, "bad" presentation also hinders students' comprehension. Besides, an efficient way to process information differs from student to student; a visualizer may learn well with visual aids while a verbalizer may learn well with texts [10]. This intuition leads us to investigate to what extent presentation-based features such as word count and illustrations affect comprehension for each student. We show during data analysis how to interpret features' weights which are the by-product of training a classifier. Some obstacles in lecture comprehension can be revealed in light of this analysis.

Breaking students' comprehension down into features of slides ignores the fact that there are relevant factors outside the teaching materials. Instructor's explanation, for instance, is an important feature that also affects comprehension level. This spoken discourse shares the underlying representation with slides' contents, that is, (spoken versus written) words. Because the spoken discourse augments the story in slides while slides themselves support the narrative, there is typically much overlap in both sets of spoken and written words. The redundancy allows us to exclude spoken words from analysis, thanks to slides' contents which can be thought of as a summary of the discourse, and which usually contain important keywords or phrases such as technical terms that are the main themes in narrative. Nevertheless, whenever spoken words are easily accessible, they can be added as extra features without significant effort.

Despite the absence of spoken discourse, our model still can incorporate how such discourse is delivered to students as its compensation. For example, slide duration indicates how long an instructor spends explaining each slide. To put this into a broader perspective, these "instructor" features are a part of presentation styles. With this point in mind, more features can be discovered from nonverbal communication which also highlights the success in classroom teaching [16]. Pointing and tracing gestures might facilitate learning when combined with a verbal mode of instruction [22]. This paper, however, focuses on "lecture" features while nonverbal features are a challenge for future study.

Predicting students' performance by machine learning techniques is, however, not new. Page [17] claimed that text complexity features such as essay length and word length can predict essay writing scores that are highly correlated with human graders. However, Larkey [13] found that content-based features alone are as much effective as those text complexity features. This is not surprising as the choice of words is important for essay writing. Because presentation slides offer richer contents than essays, text complexity features are subsumed by our presentation-based features which also include multimedia complexity and temporal features. More important, we are not aware of related work that provides feature analysis similar to our study (see Section 7).

Research questions:

1. Are presentation-based features as effective as content-based features for slide classification?
2. Analyzing students' comprehension feedback against presentation styles might allow us to measure students' learning skills, but is it possible to filter the effect of lecture contents from presentation styles?

3 Data Sets

We obtained a collection of slides from a digital image processing course taught in Japanese by the third author at Kyoto University in 2009. The course delivered a series of 12 week lectures, 1.5 hour long each, and 206 slides in total.

Students who enrolled this course were asked to sign agreements to make their feedback data accessible to researchers. Feedback data were recorded in every class manually by students who filled in questionnaires with 5-point scale corresponding to *very_low*, *low*, *average*, *high*, and *very_high*. We distributed questionnaires in the beginning of each class, and asked students to specify their own comprehension for each slide as soon as the next slide was coming. We also explained that the feedback data have no effect on test scores and the final grade of the course.

We obtained the questionnaire data from a total of 102 students, each of which constitutes a single data set. As it was a voluntary research program, the data sets we can gather from most students are scarce (Fig. 1). About half of the students provided labels for less than 50 slides. Although lecture attendance is mandatory in principle, most students were absent in some classes. Moreover, they did not necessarily complete the questionnaires for every

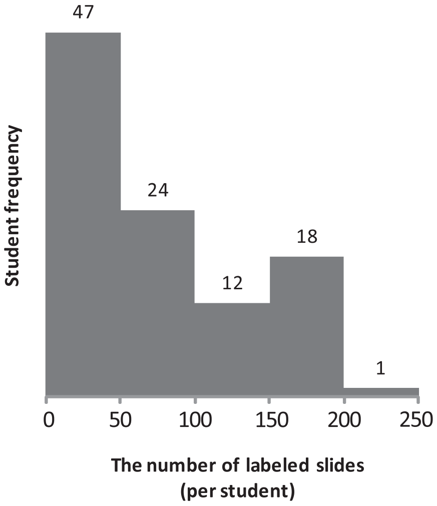


Fig. 1. Histogram of labeled slides count.

TABLE 1
The Number of Samples Distributed over Five Classes of Comprehension Level

Student ID	very low	low	average	high	very high	Dataset size
0493	4	24	74	30	9	141
1136	2	6	14	13	15	50
1412	13	34	59	46	20	172
1824	4	18	25	18	52	117
2189	3	9	7	8	14	41
2296	21	13	26	39	20	119
4419	2	15	15	17	11	60
5494	10	48	16	24	6	104
6802	18	38	52	36	26	170
6876	4	10	25	74	72	185
7541	5	7	16	11	17	56
8058	2	12	31	25	9	79
8709	3	2	7	80	74	166
8883	7	10	8	7	10	42
9081	32	6	10	20	10	78
9260	4	35	25	59	57	180
9653	18	13	2	6	61	100

slide (e.g., when they did not listen to the lecture). Only 31 students provided labels for over 100 slides, but closer inspection reveals that some of the data are subject to insensitive judgments or noise.

Accordingly, we applied the following selection and filtering criteria to all 102 data sets. First, we eliminated data sets whose class distribution does not cover all five comprehension levels. Next, we tried to manually remove the rating data that seem insensitive or spurious; the same rating was given for a whole class. The selection process resulted in 17 data sets whose statistics are shown in Table 1.

From a machine learning perspective, a problem of multiclass classification is usually more difficult than binary classification. This is simply suggested by classification accuracy of random guess; 20 percent for 5-level slide classification and 50 percent for binary classification. To increase accuracy of our prediction at the expense of coarse-grained indicator, we duplicated additional 17 data sets with replacement of two new labels as shown in Table 2. The labels of *high* and *very_high* are merged into *understand* while *low* and *very_low* are combined into *not_understand*. All examples with the label of *average* are discarded.

TABLE 2
The Number of Samples Distributed over Two Classes of Comprehension Level

Student ID	not_understand (very_low+low)	understand (high+very_high)	Dataset size
0493	28	39	67
1136	8	28	36
1412	47	66	113
1824	22	70	92
2189	12	22	34
2296	34	59	93
4419	17	28	45
5494	58	30	88
6802	56	62	118
6876	14	146	160
7541	12	28	40
8058	14	34	48
8709	5	154	159
8883	17	17	34
9081	38	30	68
9260	39	116	155
9653	31	67	98

4 Feature Preparation

In this paper, we characterize slides based on two broad categories of features:

4.1 Content-Based Features

Classifying slides according to comprehension level can be viewed as a special case of automatic text categorization, which is a task of assigning a category to a document.

Text categorization has reached a mature level and spans a wide range of applications such as sentiment analysis for movie reviews [18], emails classification based on intention [2], customer satisfaction analysis [8], spam detection in online posts [9], and stock prediction based on financial news [20]. Machine learning techniques are the key to solving this problem and often employ the bag-of-words model as classificatory features. Document representation is simplified by the assumption that the order of words is not important for the task of document classification.

It is intuitive to use word features in our problem context because a word (or a topic) that appears in slides is often a "direct" indicator of comprehension. The criteria for selection is that any technical terms or topics that have impact on understanding are manually extracted as features. From our slide collection, we obtain 227 unique topics as content-based features. Each slide has an average and a median of three unique topics. As for the feature representation, we choose a binary model; that is, whether a topic appears in a slide or not while ignoring topic frequency.

4.2 Presentation-Based Features

As how instructors present materials also affects comprehension, we develop a set of 11 presentation-based features: *wordCount*, *slideDuration*, *slidePosition* (the distance from the beginning slide of a lecture), *minFontSize* (the smallest letter size), *seatDistance* (seat row number), *slideSimilarity* (how similar it is to the previous slide), *hasBullet*, *hasChart* (e.g., histogram, bar chart, pie chart, function graph, scatter plot), *hasDiagram* (e.g., tree diagram, network diagram, flowchart, Venn diagram), *hasFigure*, and *hasMathFormula*.

All features except *wordCount* and *slideSimilarity* are extracted manually from slides. Unlike English, Japanese sentences have no explicit word boundary. We use ChaSen [15], a morphological analyzer, for segmenting Japanese words in slides. Only informative words including nouns, verbs, and adjectives are extracted to estimate the word count. For *slideSimilarity*, we apply a cosine measure with *tf.idf* weighting to calculate similarity score for each pair of two consecutive slides. In general, *seatDistance* can indicate the level of students' interest. We decide to include *seatDistance* as it also augments *minFontSize*. That is, slide legibility also depends on the seat location in the classroom.

It is important to emphasize that our current study is in engineering education contexts. Lecture materials present conceptual ideas with mathematical elements, which govern the choice of features such as *hasMathFormula* and *hasDiagram*. Although these features are mostly pertinent to engineering context, our approach is general and applicable to various disciplines. More features can be added if they become relevant, and irrelevant features can be removed depending on characteristics of lecture contents.

5 Support Vector Machines

A powerful statistical learner is solicited to prevent the overfitting problem as our data sets are very small but high-dimensional. Support vector machines (SVM) [5] is proved effective to handle this unsatisfactory characteristic of data (see, e.g., in [1] where SVM is used to deal with very small gene expression data sets). Here we provide a brief introduction to SVM.

For most learners, a prediction outcome is determined by a decision boundary or a separating hyperplane given as $g(\mathbf{x}^{(i)}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + b = 0$ where $\mathbf{x}^{(i)}$ is the input features vector of a data point i, $\mathbf{w}$ is the weights or coefficients vector, and b is the bias. The prediction outcome for a data point is positive if $g(\mathbf{x}^{(i)}) > 0$ and negative otherwise. The intuition behind SVM that makes it different from other learners is confidence level of this prediction. Intuitively, we can make very confident prediction of positive and negative class, respectively, when $g(\mathbf{x}^{(i)}) \gg 0$ and $g(\mathbf{x}^{(i)}) \ll 0$ because, in that case, a data point is very far from a hyperplane. In contrast, a small change to hyperplane's direction can easily cause a different prediction of data points lying near the hyperplane. Therefore, the key idea of SVM is to find an optimal hyperplane with a large margin that separates positive from negative data points. Because data points lying far from the hyperplane have no role in deciding hyperplane's direction, SVM ignores them and computes the hyperplane direction based on only those lying near the hyperplane, which are called support vectors.

The problem of finding an optimal hyperplane is reduced to solving the following optimization problem:

$$\begin{array}{lll} \underset{\mathbf{w},b}{\text{minimize}} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i & \\ \text{subject to} & y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + b) \geq 1 - \xi_i, & i = 1,\ldots,m \\ & \xi_i \geq 0, & i = 1,\ldots,m, \end{array}$$

where m is the number of data points (or training examples), C is the regularization parameter that balances between accuracy and generalizability, and ξ_i is a slack variable that allows misclassification for a data point i.

Intuitively the problem becomes maximizing the margin which is represented by the term $\frac{2}{\|\mathbf{w}\|^2}$. This objective function, however, is equivalent to minimizing $\frac{1}{2}\|\mathbf{w}\|^2$, which can be solved efficiently by off-the-shelf optimization software. The constraint term $y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + b)$ indicates whether a data point i is correctly classified. By adding a slack variable ξ_i to this term, it is possible for the hyperplane to misclassify data points (i.e., if $\xi_i > 1$, then $y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + b) < 0$, resulting in misclassification). This regularization term gives better control over outliers as well as helps to avoid the overfitting problem in most cases.

Inherently, SVM can manage only the case of binary classification. There are two common extensions for building a multiclass classifier. Let c be the number of classes. The first strategy is to build c binary classifiers to classify one-class versus the rest (also known as the one-versus-all classification). The predicted class is the one that classifies a testing data point with the greatest margin. Another extension is to use as many as $c(c-1)/2$ binary classifiers to separate each class from every other class (also known as one-versus-one classification). The prediction outcome is the one predicted by most classifiers. Although more classifiers are used in this case, there are usually far less number of training data in each classifier, resulting in faster training time.

Because we exploit the coefficients vector for analyzing learning skills (see Section 7), any machine learners that offer such vector (e.g., logistic regression) is applicable to our analysis approach. However, our study is limited to SVM as it is among the state-of-the-art learners and often performs better than others.

6 Experiments

We compare classification performance among three feature categories: *Content (C)*, *Presentation (P)*, and *Content + Presentation (C+P)*.

6.1 Experimental Setup

We use an efficient algorithm of SVM, called sequential minimal optimization (SMO), implemented in Weka [23]. The regularization parameter C is set to 1, and the option *buildLogisticModels* is set to false as probability estimates are not used in this study. One-versus-one strategy is used for multiclass classification.

TABLE 3
Comparison of 5-Class Classification Accuracy among the Baseline (B), Content-Based Features (C), Presentation-Based Features (P), Content-Based Features with Feature Selection (C + FS), Presentation-Based Features with Nonlinear Kernel (P + NK), Combined Features (C + P), Combined Features with Feature Selection (C + P + FS)

Student ID	B	C	P	C+FS	P+NK	C+P	C+P+FS
0493	0.52	0.53	0.52	0.56	0.62	0.57	0.62
1136	0.30	0.42	0.32	0.44	0.40	0.44	0.44
1412	0.34	0.42	0.45	0.47	0.41	0.62	0.59
1824	0.44	0.62	0.56	0.63	0.56	0.62	0.63
2189	0.34	0.51	0.63	0.63	0.68	0.63	0.68
2296	0.33	0.41	0.55	0.44	0.67	0.54	0.57
4419	0.28	0.40	0.38	0.47	0.43	0.45	0.45
5494	0.46	0.57	0.58	0.61	0.66	0.64	0.70
6802	0.31	0.46	0.35	0.46	0.44	0.48	0.49
6876	0.40	0.43	0.51	0.45	0.54	0.51	0.58
7541	0.30	0.57	0.63	0.59	0.63	0.64	0.64
8058	0.39	0.39	0.48	0.39	0.44	0.41	0.49
8709	0.48	0.58	0.54	0.59	0.57	0.67	0.67
8883	0.24	0.38	0.60	0.38	0.64	0.55	0.55
9081	0.41	0.56	0.38	0.56	0.47	0.50	0.56
9260	0.33	0.61	0.54	0.58	0.55	0.66	0.69
9653	0.61	0.65	0.57	0.67	0.62	0.69	0.70
Average	0.38	0.50*	0.51*	0.52*	0.55*	0.57*	0.59*

** marks results that differ significantly from the baseline at $p < 0.01$ (two-tailed paired t-test).*

TABLE 4
Comparison of Binary Classification Accuracy among the Baseline (B), Content-Based Features (C), Presentation-Based Features (P), Content-Based Features with Feature Selection (C + FS), Presentation-Based Features with Nonlinear Kernel (P + NK), Combined Features (C + P), Combined Features with Feature Selection (C + P + FS)

Student ID	B	C	P	C+FS	P+NK	C+P	C+P+FS
0493	0.58	0.84	0.79	0.85	0.90	0.88	0.91
1136	0.78	0.72	0.86	0.75	0.83	0.78	0.86
1412	0.58	0.78	0.70	0.81	0.74	0.88	0.89
1824	0.76	0.84	0.85	0.85	0.89	0.83	0.87
2189	0.65	0.88	0.94	0.88	0.94	0.94	0.94
2296	0.63	0.70	0.84	0.73	0.86	0.84	0.87
4419	0.62	0.84	0.82	0.87	0.80	0.84	0.87
5494	0.66	0.82	0.82	0.83	0.88	0.85	0.85
6802	0.53	0.65	0.75	0.67	0.76	0.75	0.82
6876	0.91	0.95	0.91	0.95	0.91	0.93	0.94
7541	0.70	0.75	0.75	0.75	0.75	0.75	0.78
8058	0.71	0.75	0.58	0.75	0.65	0.77	0.77
8709	0.97	0.97	0.97	0.98	0.97	0.97	0.97
8883	0.50	0.71	0.82	0.71	0.91	0.76	0.76
9081	0.56	0.72	0.56	0.75	0.78	0.65	0.75
9260	0.75	0.85	0.83	0.85	0.86	0.91	0.92
9653	0.68	0.70	0.63	0.74	0.74	0.72	0.78
Average	0.68	0.79*	0.79*	0.81*	0.83*	0.83*	0.86*

** marks results that differ significantly from the baseline at $p < 0.01$ (two-tailed paired t-test).*

As our data sets are too small to perform training-test split, *leave-one-out cross validation (LOOCV)* is used for approximating generalization errors. For each data set, an example is held out as the test set and the rest of the data set forms the training set. The process repeats until every example becomes a test example exactly once. Classification accuracy is then averaged among all tests. The baseline approach in our study is a classifier that always predicts the majority class (ZeroR in Weka). This baseline is often used in comparison as its classification accuracy is usually higher than random guess.

The number of features usually sets a tradeoff between bias and variance of classifier performance. Therefore, we vary the number of features by applying a feature selection technique which reduces the feature number, and a nonlinear kernel which transforms the input feature space to a higher-dimensional feature space, thereby increasing the number of features.

Assuming the classifier performance is estimated by LOOCV, feature selection is performed during the training stage where, in a training set, one example is held out for the evaluation stage and the rest examples become available to a feature selection algorithm. Features are selected based on information gain, $I(Y;X) = H(Y) - H(Y|X)$ where $H(Y)$ is the entropy of class attribute Y, $H(Y|X)$ is the conditional entropy of Y given that we know the value of a feature X. $I(Y;X)$ measures the benefit of knowing X for reducing uncertainty about Y. Information gain is computed for each feature, and features are ranked according to information gain scores. In our experiments, the final number of features is varied from 10 to 200 (in step of 10). A feature set that produces the highest accuracy on the training set is considered optimal.

A linear classifier treats input features independently; as a result, the number of dimensions is as many as the number of input features. To increase the number of dimensions, a nonlinear kernel models feature conjunctions. The idea is that comprehension level may not be estimated by a single feature alone, but by a combination of features like *minFontSize* AND *seatDistance* or *wordCount* AND *slideDuration*. In such a case, a quadratic kernel is needed. In our experiments, we use a polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$ where the order of polynomial d is varied from 2 to 6 and only the best results are reported.

6.2 Experimental Results

Table 3 shows accuracy of multiclass classification. At a glance, every method is statistically better than the baseline ($p < 0.001$). With an average of 89 features remaining, *Content+Presentation+FeatureSelection* is the best method and is statistically better than others except *Presentation+NonlinearKernel* ($p = 0.025$). *Content+Presentation*, with 86 features remaining on average, performs better than *Content* with $p < 0.001$ and also better than *Presentation* with $p = 0.003$. *Presentation* is slightly better than *Content*, but the result is statistically indistinguishable.

Table 4 shows performance comparison for binary classification. Again, every method is significantly better than the baseline ($p < 0.003$). With an average of 59 features remaining, *Content+Presentation+FeatureSelection* is the best method and is statistically better than others except *Presentation+NonlinearKernel* ($p = 0.182$). *Content+ Presentation*, with 49 features remaining on average, performs better than *Content* and *Presentation* but the results are not statistically significant (with $p = 0.014$ and $p = 0.056$, respectively). *Presentation* is as good as *Content*.

Based on these empirical results, the first question we posed earlier has been answered: *Presentation* is as effective as *Content* for slide classification according to comprehension levels.

6.3 Discussion

For multiclass classification, we observe that the gap between training error and test error (LOOCV) is large in several data sets for both *Content* and *Content+Presentation*. This situation suggests an overfitting problem (high variance). Since adding more training examples is not an option, we instead try to apply a smaller set of features. For *Presentation*, the gap between training and test error is small; however, the training error is still very high. This suggests incapability of features for the classification task (high bias). As the number of presentation-based features is currently fixed, a nonlinear classifier (polynomial kernel) is applied to map the input feature space to a higher-dimensional feature space.

Learning to estimate multilevel comprehension is inherently difficult. We speculate that our data gathering in the classroom was not efficient. First, the data gathering process puts a lot of cognitive load on students as they had to report their own comprehension at the same time they tried to understand the lecture contents. Second, students were asked to perform, in a sense, multiclass classification manually. That is, they need to distinguish between closely related classes such as *high* from *very_high* and *low* from *very_low*. Without precise justification from students, the classes get mixed and are not clearly separable. This reason might explain, in most cases, why the gap between training and test errors that we observed is much smaller when these classes are merged (binary classification).

We also examine how well a classifier can generalize across students. To this end, we build a binary classifier that uses as the training set data from all students except a target student. The accuracy is measured with the test set collected from the target student. The results in Table 5 show that a classifier trained with one's own data outperforms a classifier trained with others' data ($p < 0.001$) even if more training examples are available to the latter. As expected, predicting performance of a previously unseen student based on observation of other students' feedback is less reliable due to individual differences.

TABLE 5
Comparison of Binary Classification Accuracy between a Classifier That Is Trained with One's Own Data and One That Is Trained with Data Obtained from All Other Students

Student ID	Trained with one's own data	Trained with others' data
0493	0.88	0.76
1136	0.78	0.83
1412	0.88	0.77
1824	0.83	0.75
2189	0.94	0.91
2296	0.84	0.73
4419	0.84	0.82
5494	0.85	0.57
6802	0.75	0.58
6876	0.93	0.78
7541	0.75	0.80
8058	0.77	0.79
8709	0.97	0.69
8883	0.76	0.71
9081	0.65	0.57
9260	0.91	0.74
9653	0.72	0.59
Average	0.83	0.73

Both binary classifiers use combined features ($C + P$).

7 Measuring Learning Skills

Any prediction without explanation is hard to be justified. More important, the ultimate goal of this paper is not about "prediction" in itself. Rather, we shall seek explanation for poor comprehension from the prediction model.

Content+FeatureSelection and *Content* are not a viable option because presentation styles are not available for analysis. With *Presentation+NonlinerKernel*, we cannot interpret presentation styles independently due to feature conjunctions. Despite being the most effective method, *Content+Presentation+FeatureSelection* filters out different presentation features across students such that there is little overlap, making it difficult to compare results with learning preferences. Compared to *Content+Presentation+FeatureSelection*, *Content+Presentation* trades classification accuracy with a more robust way to interpret results. We prefer *Content+Presentation* to *Presentation* not only because it performs better, but it also offers the analysis that filters the effect of lecture contents from presentation styles.

We shall analyze linear classifier models under *Content+Presentation*. We choose the binary model rather than multiclass simply because it is easier to interpret the result.

7.1 Separating Effect of Contents from Presentation Styles

The need of separating contents from their presentation arises from a fact that comprehension level depends not only on teaching styles but also on difficulty level of topics themselves. The appearance of equations, for instance, seems to lead to poor comprehension, but their complexity varies greatly from topic to topic. The impact of presentation elements is unclear unless subtracted with topic impact.

Ranking features according to the size of coefficients is a common approach to impact analysis. In our application context, the impact analysis can be done at two levels: *individual* and *collective*. Even though we emphasize individual differences by building separate classifiers, this does not prevent us from building a classifier that combines feedback from all students. At the collective level, SVM tries to optimize classification accuracy by taking into account feedback data provided by all students.

Using *Presentation* alone makes an independent assumption that features other than presentation styles are irrelevant to comprehension level. However, analyzing the impact without taking contents into account has a strong bias toward presentation styles, which can be misleading at times. This problem can be seen clearly when Tables 6 and 7 are compared side by side. In an extreme case of *ID9653*, all presentation-based features disappear from the top ranks and are replaced completely by content-based features. That is, the quality of presentation styles is likely to have little impact on this student, which would otherwise remain unobserved with the absence of content-based features. In contrast, for *ID4419*, a few presentation-based features

TABLE 6
Presentation-Based Features (P) Ranked by the Absolute Value of SVM Coefficients

All	4419	9653
(−)**slidePosition**	(+)**hasBullet**	(−)slideSimilarity
(+)**minFontSize**	(−)slidePosition	(−)hasDiagram
(+)**hasBullet**	(−)slideSimilarity	(+)wordCount
(−)slideDuration	(−)**hasChart**	(−)hasBullet
(−)seatDistance	(−)seatDistance	(+)seatDistance
(−)hasMathFormula	(−)wordCount	(+)minFontSize
(+)wordCount	(−)hasMathFormula	(−)hasMathFormula
(+)hasFigure	(+)minFontSize	(−)hasFigure
(−)slideSimilarity	(+)hasDiagram	(−)slidePosition
(+)hasChart	(+)hasFigure	(−)slideDuration
(−)hasDiagram	(+)**slideDuration**	(−)hasChart

TABLE 7
Top-10 Combined Features (C+P) Ranked by the Absolute Value of SVM Coefficients

All	4419	9653
(−)topic129	(+)**hasBullet**	(−)topic28
(−)topic174	(−)**hasChart**	(−)topic25
(−)**slidePosition**	(−)topic51	(−)topic141
(−)topic83	(−)topic74	(−)topic175
(+)**hasBullet**	(−)topic83	(−)topic94
(+)topic148	(−)topic5	(−)topic133
(−)topic5	(−)topic165	(−)topic119
(+)**minFontSize**	(−)topic224	(+)topic41
(+)topic18	(−)**slideDuration**	(+)topic80
(+)topic54	(−)topic129	(+)topic207

remain in the top ranks, implying that he or she still relies heavily on those presentation elements. When the feature impact is analyzed collectively across all students, we see that *slidePosition*, *minFontSize*, and *hasBullet* have the great impact even with the presence of content-based features.

When the impact analysis is done collectively, one should keep in mind that those dominant features may differ from class to class, due to individual differences, and from subject to subject, due to unique characteristics of each discipline. Therefore, the results must be interpreted with care. In other words, our approach is general but applies in a class-specific and subject-specific manner.

Although content-based features are not useful for revealing comprehension problems, incorporating them into a classifier can "systematically" filter out the impact of contents from presentation styles. We believe that the use of combined features together with this empirical result should address the second question we posed in Section 2.

7.2 Making Inference from Presentation Styles

As a classification result depends on the sign of the distance from the separating hyperplane, coefficients in the hyperplane equation play the essential role in deciding the prediction outcome. Intuitively, positive coefficients contribute to the prediction of *understand* while negative coefficients to *not_understand*. This interpretation results in positive and negative impact of how students respond to presentation styles (attributes) as shown in Table 8.

Take a student with *ID0493* as an example. This student learns well with visual aids such as figures, diagrams, and charts (*hasFigure*, *hasDiagram*, and *hasChart* being positive). In contrast, many words in a single slide may hurt his or her comprehension (*wordCount* being negative). *minFontSize* being negative is probably attributed by the fact that most figure captions are of small size. In general, *slideSimilarity* indicates how smooth is the topic transition, or to what extent one can tolerate nonsmooth transition. The negative impact of *slideSimilarity* toward some other students surprised us at first. The more content overlapping the consecutive slides share, the less understanding they will gain. In this case, we speculate that the topic difficulty arises when a topic develops more deeply. Since this student did not change the seat row throughout the course, the impact analysis of *seatDistance* is not available.

When analyzed collectively with a classifier that combines all data, students tend not to understand as a lecture proceeds (the coefficient sign of *slidePosition* being negative), prefer well-organized contents into topic-subtopic (*hasBullet* being positive), and have a problem to understand math formula (*hasMathFormula* being negative). Note that the collective impact, in most cases, follow the majority votes across students analyzed individually.

Apart from the collective analysis, the aggregate results obtained from separate classifiers might reveal interesting knowledge. For example, *hasDiagram* being positive outnumbers *hasChart* being positive; diagrams are easier to understand than charts. Some are good at processing multichannel information at the same time (both *wordCount* and *hasFigure* being positive). In general, we may use *association rule mining* to facilitate the discovery of such relationships. For example, the rule "$seatDistance = negative \Rightarrow slideDuration = positive$" can be derived with the support of 0.35 (i.e., this pattern occurs $\frac{6}{17} = 35\%$ of the time) and the confidence of 0.86 (i.e., when *seatDistance* is negative we can find $\frac{6}{7} = 86\%$ of the time that *slideDuration* is positive). This rule might be interpreted as follows: for those who have poor comprehension because of sitting far away from the front of a class, spending more time in explanation might help them understand more.

7.3 Threshold for Understanding

After all, the sign of coefficients tells us about how students comprehend a lecture. However, this guideline is sometimes vague. For example, while decision can be made precisely on whether to use bullet points in slides or not, it becomes unclear on how many words in a slide considered harmful to students' comprehension.

Table 9 presents thresholds for understanding derived from SVM using only single attributes. The thresholds are obtained by solving the separating hyperplane equation, with single attributes, $w_i \cdot x_i + b > 0$ where w_i and b are SVM's parameters, and x_i is the normalized attribute value $\in [0, 1]$ and i ranges over all numeric attributes. The actual attribute values are recovered and shown in these thresholds. It is worth noting that the precision of each threshold is not measured by the current training examples but approximated by previously unseen examples using

TABLE 8
Impact of Presentation Styles Derived from SVM's Coefficients (i.e., The Sign Being Positive or Negative)

Student ID	slide Position	has Bullet	min FontSize	has Diagram	has Chart	hasMath Formula	word Count	has Figure	slide Similarity	slide Duration	seat Distance
0493	−	+	−	+	+	−	−	+	+	+	n/a
1136	−	+	−	−	−	−	+	−	+	−	+
1412	−	+	+	−	−	−	+	+	−	−	+
1824	−	+	+	+	−	−	−	+	−	−	+
2189	−	+	−	−	n/a	+	+	+	−	+	−
2296	−	+	−	+	−	−	+	+	+	−	+
4419	−	+	+	+	−	−	−	+	−	−	+
5494	−	+	+	+	−	−	+	+	−	+	−
6802	−	+	+	−	−	−	+	−	+	+	−
6876	−	+	+	+	+	−	+	+	+	−	n/a
7541	−	+	+	n/a	−	−	n/a	+	+	+	−
8058	−	+	+	+	−	−	−	+	−	+	−
8709	+	+	+	+	+	−	−	+	−	−	+
8883	−	+	−	+	−	−	−	+	+	+	−
9081	−	+	+	−	+	+	−	+	+	+	+
9260	−	+	−	+	−	−	−	−	+	−	−
9653	−	−	+	−	−	−	+	−	−	+	+
All	−	+	+	−	−	−	−	+	+	+	n/a

"n/a" is interpreted as no impact or unknown.

LOOCV. Omitted here is a measure of recall as how much percentage of *understand* accounted for by the thresholds.

In several cases, a single attribute is not powerful enough to beat the majority-class classifier; as a result, a threshold cannot be decided based on that attribute alone, which is marked by "n/a." Nevertheless, we found that increasing the number of minority examples in the training data can address this problem. In our experiments, we use *synthetic minority oversampling technique* (SMOTEn) [3] to balance a data set where n is the percentage of "artificial" minority examples being generated. The key idea is to create synthetic examples such that they are surrounded, in the feature space, by some genuine examples. Specifically, each synthetic example is generated in the direction that falls in between two genuine examples; one is given and another is one of its k-nearest neighbors randomly chosen. This process repeats until the example number reaches the percentage specified. The nearest neighbors parameter, k, is set to 5 in our study. It is important to emphasize that synthetic examples are generated only during the training stage and not used as the test examples. The results of applying SMOTE are reasonable; as the precision increases, the range of threshold shrinks (Table 9). When a data set is highly imbalanced, SVM often has a bias toward a trivial classifier that always predict the majority class. We speculate that what accounts for the increased precision is the lower chance to obtain the majority-class classifier as a data set becomes more balanced.

Thresholds in Table 9 explain to what extent numeric attributes have positive or negative impact. For example, the threshold "$> 12.5(0.71)$" of *wordCount* in *ID0493* can be interpreted as follows: a target student tends to understand a slide, with 71 percent of the time, if it contains more than 12.5 words. The reader might notice, in this case, a contradiction between the threshold and the negative coefficient sign (Table 8), which is derived when all attributes are available to the machine learner. That is, the student is also likely to understand a slide with less than 12.5 words, but more conditions on other attributes are necessary. Therefore, the thresholds should be interpreted with care, and nothing can be inferred if the condition of the threshold is not satisfied. The knowledge obtained from data mining eventually becomes more actionable. For example, by observing the threshold of *wordCount* analyzed collectively for all students, a slide should not contain less than 49 words. (This seems to be a large number of words, but this number is better put in the context of our data set statistics which have $\text{mean} = 31$ words per slide, $\text{min} = 1$, $\text{max} = 108$, and $\text{std.} = 23$).

Note that we can easily obtain thresholds for *not_understand* by reversing the condition in all the thresholds in Table 8. However, the precision of those reversed thresholds would be much lower than the case of *understand*. The reason is that detecting rare events (*not_understand*) is more difficult especially when only single attributes are available to the machine learner.

8 Conclusion

This study shows a promising result toward estimating students' comprehension by a machine learning technique. We are aware that a limitation of our study lies in the small size of data sets. In retrospect, we think we could make our data collecting more attractive to students. For example, to facilitate slides labeling, handheld devices like iPhone should be used instead of questionnaire sheets. As we gained valuable experience in this pilot study and are in a position to convince students of the benefit of providing precise feedback that will let them understand their own learning skills, we expect students to be more engaged in real use of better system implementation.

TABLE 9
Thresholds for Understanding Derived from Single-Attribute Learners (or Decision Stumps)

Student ID	slidePosition	seatDistance	minFontSize	slideSimilarity	slideDuration	wordCount
0493	< 12.5 (0.82)	n/a	< 33.1 (0.76)	n/a	n/a	> 12.5 (0.71)
0493+SMOTE50	< 9.7 (0.87)	n/a	< 32.0 (0.77)	n/a	> 5.4 (0.64)	> 30.5 (0.76)
1136	n/a	n/a	n/a	n/a	n/a	n/a
1136+SMOTE100	< 17.1 (0.77)	n/a	< 43.2 (0.78)	n/a	n/a	n/a
1136+SMOTE200	< 11.5 (0.89)	n/a	< 31.0 (0.76)	n/a	n/a	> 35.0 (0.67)
1136+SMOTE300	< 8.4 (0.88)	> 5.0 (0.67)	< 25.0 (0.64)	> 0.17 (1.00)	n/a	> 41.3 (0.83)
1412	< 11 (0.73)	n/a	n/a	< 0.36 (0.60)	n/a	n/a
1412+SMOTE50	< 9.0 (0.78)	n/a	> 36.2 (0.40)	< 0.26 (0.59)	n/a	n/a
1824	n/a	n/a	n/a	n/a	n/a	n/a
1824+SMOTE100	< 12.4 (0.80)	n/a	n/a	n/a	n/a	n/a
1824+SMOTE200	< 9.9 (0.91)	> 4.9 (0.88)	> 27.9 (0.62)	n/a	< 9.6 (0.78)	< 49.5 (0.70)
1824+SMOTE300	< 8.2 (0.94)	> 5.0 (0.90)	> 31.0 (0.92)	n/a	n/a	n/a
2189	< 12.4 (0.91)	n/a	n/a	< 0.32 (0.61)	n/a	n/a
2189+SMOTE50	< 11.0 (0.95)	< 3.0 (0.81)	< 37.6 (0.64)	< 0.20 (0.67)	> 4.1 (0.86)	n/a
2189+SMOTE100	< 10.4 (0.95)	< 3.0 (0.81)	< 18.3 (0.22)	< 0.18 (0.72)	> 4.6 (0.91)	n/a
2296	< 15.5 (0.70)	n/a	< 34.0 (0.81)	n/a	n/a	n/a
2296+SMOTE50	< 12.0 (0.79)	< 5.0 (0.56)	< 34.0 (0.81)	n/a	n/a	> 27.6 (0.80)
2296+SMOTE100	< 10.0 (0.85)	< 5.0 (0.71)	< 32.0 (0.80)	> 0.69 (0.71)	n/a	> 31.0 (0.79)
4419	< 11.0 (0.76)	n/a	n/a	< 0.18 (0.72)	n/a	n/a
4419+SMOTE50	< 9.5 (0.85)	< 2.0 (0.74)	n/a	< 0.14 (0.77)	n/a	< 87.9 (0.59)
4419+SMOTE100	< 8.1 (0.85)	< 2.0 (0.70)	n/a	< 0.11 (0.81)	n/a	n/a
5494	< 7.0 (0.77)	n/a	n/a	n/a	n/a	n/a
6802	< 11.4 (0.72)	n/a	< 32.0 (0.57)	n/a	n/a	> 33.5 (0.62)
6876	n/a	n/a	n/a	n/a	n/a	n/a
6876+SMOTE500	n/a	n/a	n/a	n/a	< 19.4 (0.91)	n/a
6876+SMOTE1000	> 21.7 (0.20)	n/a	> 31.1 (0.97)	> 0.5 (0.53)	< 4.7 (0.93)	n/a
7541	< 24.4 (0.70)	n/a	n/a	n/a	n/a	n/a
7541+SMOTE50	< 15.3 (0.86)	n/a	n/a	n/a	n/a	n/a
7541+SMOTE100	< 14.5 (0.85)	n/a	n/a	> 0.5 (0.37)	n/a	n/a
8058	n/a	n/a	n/a	n/a	n/a	n/a
8058+SMOTE50	n/a	n/a	n/a	< 0.40 (0.70)	n/a	n/a
8058+SMOTE100	< 11.4 (0.73)	n/a	> 28.5 (0.48)	< 0.28 (0.70)	n/a	n/a
8709	n/a	n/a	n/a	n/a	n/a	n/a
8709+SMOTE1000	n/a	n/a	n/a	n/a	< 20.6 (0.97)	n/a
8709+SMOTE2000	n/a	n/a	n/a	n/a	< 7.0 (0.97)	n/a
8709+SMOTE3000	< 8.9 (0.93)	> 3.5 (0.64)	> 31.0 (0.87)	> 0.34 (0.76)	< 5.9 (0.98)	> 38.4 (0.92)
8883	< 9.5 (0.06)	< 4.0 (0.93)	> 25.0 (0.06)	n/a	n/a	n/a
9081	n/a	n/a	n/a	n/a	n/a	n/a
9260	n/a	n/a	n/a	n/a	n/a	n/a
9260+SMOTE100	< 18.4 (0.71)	n/a	< 34.0 (0.86)	n/a	n/a	n/a
9260+SMOTE200	< 11.5 (0.87)	> 4.5 (0.76)	< 32.0 (0.86)	> 0.00 (0.26)	< 2.1 (0.20)	> 31.6 (0.83)
9653	n/a	n/a	n/a	< 0.81 (0.68)	n/a	n/a
9653+SMOTE50	n/a	n/a	n/a	< 0.50 (0.70)	n/a	n/a
9653+SMOTE100	n/a	> 1.5 (0.78)	> 28.0 (0.51)	< 0.50 (0.68)	n/a	n/a
All	n/a	n/a	n/a	n/a	n/a	n/a
All+SMOTE50	< 13 (0.76)	n/a	n/a	n/a	n/a	n/a
All+SMOTE100	< 11 (0.81)	n/a	< 44 (0.70)	n/a	n/a	n/a
All+SMOTE150	< 10 (0.83)	n/a	n/a	n/a	n/a	> 49 (0.72)

In parenthesis is the (approximated) precision of detecting the class of understand *given by each threshold.*

Restricting a learning environment to computer-based instruction would also eliminate several problems in this research. Teacher's performance, for instance, is hard to be controlled in this study. Although our third author is an experienced teacher as well as an expert in the subject being taught, a human teacher might be unreliable at times. Similarly, *seatDistance* which is also unpredictable or difficult to be controlled is no longer relevant once it comes to computer-based lessons (i.e., everyone sits in front of a computer screen). More important, learning intervention becomes easier as we can focus on individual students rather than the majority of them. That is, different instructional modes can be tailored to individuals' learning skills.

Visualizing students' comprehension has huge impact on education. On one hand, our method shows how successfully instructors present materials, thereby allowing them to improve pedagogical strategies. On the other hand, gathering comprehension data in classrooms turn passive students into active one. We provide instructors with an analysis tool to measure learning skills systematically, based on presentation styles used in lecture materials. Understanding limited capability of students would support decision-making for adjusting presentation styles. Nevertheless, we must remember that the most important factor to students' success lies not only in effective presentation styles but also in a more genuine feature like sympathetic observation and intense dedication to students.

Acknowledgments

The authors would like to thank Prof. Hsiu-Ping Yueh and Dr. Wei-Jane Lin at National Taiwan University for fruitful discussions about current limitations and future directions of this research. They would also like to thank the anonymous reviewers for valuable comments that lead them to new insights and improved both the quality and presentation of this paper. Last but not least, they are grateful to all students who participated in this study. This research would not have been possible without their valuable input and continuous support.

References

[1] A. Anand and P.N. Suganthan, "Multiclass Cancer Classification by Support Vector Machines with Class-Wise Optimized Genes and Probability Estimates," *J. Theoretical Biology,* vol. 259, no. 3, pp. 533-540, 2009.

[2] V.R. Carvalho and W.W. Cohen, "On the Collective Classification of Email "Speech Acts," *Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 345-352, 2005.

[3] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique," *J. Artificial Intelligence Research,* vol. 16, no. 1, pp. 321-357, 2002.

[4] F. Coffield, D. Moseley, E. Hall, and K. Ecclestone, "Learning Styles and Pedagogy in Post-16 Learning: A Systematic and Critical Review," technical report, Learning and Skills Research Centre, 2004.

[5] C. Cortes and V. Vapnik, "Support-Vector Networks," *J. Machine Learning,* vol. 20, no. 3, pp. 273-297, Sept. 1995.

[6] R.M. Felder and L.K. Silverman, "Learning and Teaching Styles in Engineering Education," *J. Eng. Education,* vol. 78, no. 7, pp. 674-681, 1988.

[7] N. Fleming, *VARK: A Guideline to Learning Styles,* http://vark-learn.com, 2004.

[8] S. Godbole and S. Roy, "Text Classification, Business Intelligence, and Interactivity: Automating C-Sat Analysis for Services Industry," *Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 911-919, 2008.

[9] N. Jindal and B. Liu, "Opinion Spam and Analysis," *Proc. Int'l Conf. Web Search and Web Data Mining,* pp. 219-230, 2008.

[10] J.R. Kirby, P.J. Moore, and N.J. Schofield, "Verbal and Visual Learning Styles," *Contemporary Educational Psychology,* vol. 13, no. 2, pp. 169-184, 1988.

[11] D.A. Kolb, *Experiential Learning: Experience as a Source of Learning and Development.* Prentice Hall, 1984.

[12] M. Kozhevnikov, "Cognitive Styles in the Context of Modern Psychology: Toward an Integrated Framework of Cognitive Style," *Psychological Bull.,* vol. 133, no. 3, pp. 464-481, 2007.

[13] L.S. Larkey, "Automatic Essay Grading Using Text Categorization Techniques," *Proc. 21st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 90-95, 1998.

[14] L.J. Massa and R.E. Mayer, "Testing the ATI Hypothesis: Should Multimedia Instruction Accommodate Verbalizer-Visualizer Cognitive Style?" *Learning and Individual Differences,* vol. 16, no. 4, pp. 321-335, 2006.

[15] Y. Matsumoto et al., *Morphological Analysis System ChaSen version 2.2.4 Manual,* http://chasen.aist-nara.ac.jp/stable/doc/chasen-2.2.4.pdf, 2001.

[16] S. Neill and C. Caswell, *Body Language for Competent Teachers.* Routledge, 2004.

[17] E.B. Page, "Computer Grading of Student Prose, Using Modern Concepts and Software," *J. Experimental Education,* vol. 62, pp. 127-142, 1994.

[18] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs Up?: Sentiment Classification Using Machine Learning Techniques," *Proc. Conf. Empirical Methods in Natural Language Processing,* pp. 79-86, 2002.

[19] H. Pashler, M. McDaniel, D. Rohrer, and R. Bjork, "Learning Styles: Concepts and Evidence," *Psychological Science in the Public Interest,* vol. 9, no. 3, pp. 105-119, 2008.

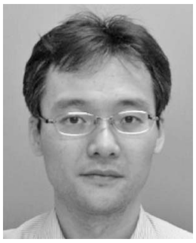
[20] R.P. Schumaker and H. Chen, "Textual Analysis of Stock Market Prediction Using Breaking Financial News: The AZFin Text System," *ACM Trans. Information Systems,* vol. 27, no. 2, pp. 1-19, 2009.

[21] F. Sebastiani, "Machine Learning in Automated Text Categorization," *J. ACM Computing Surveys,* vol. 34, no. 1, pp. 1-47, 2002.

[22] L. Valenzeno, M.W. Alibali, and R. Klatzky, "Teachers' Gestures Facilitate Students' Learning: A Lesson in Symmetry," *Contemporary Educational Psychology,* vol. 28, pp. 187-204, 2003.

[23] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques,* second ed. Morgan Kaufmann, 2005.

Nimit Pattanasri received the PhD degree in social informatics from Kyoto University in 2007. He is interested in information retrieval and machine learning to solve the problem of poor information comprehension. He is a member of SIGIR.



Masayuki Mukunoki received the bachelor, master, and doctoral degrees in information engineering from Kyoto University. He is now an associate professor in the Academic Center for Computing and Media Studies and a faculty member in the Graduate School of Informatics, Kyoto University. His research interests include computer vision, video media processing, lecture video analysis, and human activity sensing with camera.

Michihiko Minoh received the BEng, MEng, and DEng degrees in information science from Kyoto University in 1978, 1980, and 1983, respectively. He is a professor at the Academic Center for Computing and Media Studies (ACCMS), Kyoto University, Japan. He served as the director of ACCMS from April 2006 to March 2010 and concurrently served as the vice director in the Kyoto University Presidents Office from October 2008 to September 2010. Since October 2010, he has been vice-president and chief information officer at Kyoto University and the director-general at the Institute for Information Management and Communication, Kyoto University. His research interests include image processing, artificial intelligence, and multimedia applications, particularly model-centered frameworks for computer systems to help visual communication among humans and information media structures for human communication. He is a member of the Information Processing Society of Japan, the Institute of Electronics, Information, and Communication Engineers of Japan, the IEEE Computer Society, the IEEE Communications Society, and the ACM.