In *Proceedings of Nineteenth International Conference on Pattern Recognition (ICPR 2008)*, December, 2008.

# Improving Digital Ink Interpretation through Expected Type Prediction and Dynamic Dispatch

Kah Seng Tay
*MIT Computer Science and Artificial Intelligence Laboratory (CSAIL)*
*kahseng@mit.edu*

Kimberle Koile
*MIT Center for Educational Computing Initiatives (CECI)*
*kkoile@mit.edu*

## Abstract

*Interpretation accuracy of current handwriting applications can be improved by providing contextual information about an ink sample's expected type. This expected type, however, has to be known or provided a priori, and poses several challenges if unknown or ambiguous. We have developed a novel approach that uses a classic machine learning technique to predict this expected type from an ink sample. With this approach, we can create a "dynamic dispatch interpreter" by biasing interpretation differently according to the predicted expected types of the ink samples. When evaluated in the domain of introductory computer science, our interpreter achieves high interpretation accuracy (87%), an improvement from Microsoft's default interpreter (62%), and comparable with other previous interpreters (87-89%), which, unlike ours, require additional expected type information for each ink sample.*

## 1. Introduction

We have been investigating digital ink interpretation for use in a wireless classroom interaction system that enables real-time feedback and assessment of students' digital ink answers to in-class exercises. ([1][2][3]) have shown that such systems hold great promise for improving student interaction and learning in classrooms. The system, Classroom Learning Partner (CLP), consists of a network of Tablet PCs running software for posing questions to students, interpreting their handwritten answers and aggregating the answers into equivalence classes. With CLP, students handwrite their answers instead of choosing from a pre-defined set. As a result, students can engage not just in recognition of material, but in higher-order tasks such as analysis, synthesis, and evaluation, which are necessary for learning [4].

The interpretation accuracy can be improved [3][7][8] by biasing an interpreter with contextual information of the answer types: e.g., the instructor specifies ahead of time what type of answer is expected for each exercise, and that information is used to dispatch to a specialized ink interpreter. CLP currently has an interpretation accuracy of 89% using interpreters biased with contextual information of answer types [2][3]. What if the interpreter itself, however, could figure out an answer's type dynamically when the answer was submitted? This capability would free the instructor from having to spend time pre-specifying answer types; enable instructors to create exercises during class; and increase the interpretation rate for students' incorrect answers that happened to be of a different type than expected. This paper describes a novel method for increasing ink interpretation rates by such dynamic dispatch: it uses machine learning techniques to extract features from ink strokes to predict the ink answer type, then dispatches to specialized interpreters based on the type.

## 2. Related work

Artificial intelligence algorithms have contributed significantly to improving handwriting recognition to date. Specific techniques used include support vector machines, hidden Markov models (HMMs) [5], neural networks, genetic algorithms, and convolutional time delay neural networks (TDNN), used by Microsoft's default handwriting recognizer [6]. Biasing ink interpretation with templates and annotation can improve the interpretation accuracy for mailing addresses [7] and forms [8]. CLP employs the use of instructor-specified expected answer types for biasing interpretation [9], choosing a different domain-specialized interpreter for each expected type.

## 3. Approach

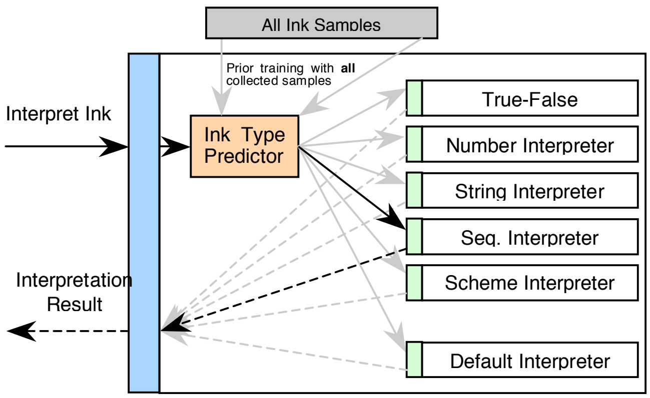


**Figure 1. Dynamic dispatch interpreter**

Our interpretation system uses machine learning techniques to predict expected ink sample type, then dynamically dispatches interpretation to a specialized interpreter (see Figure 1). For the experiments described in this paper, we use 1810 ink samples collected from students spanning 181 representative examples of answers. Eighty-eight of these examples lie within the domain of introductory computer science (including the 21 from [9]) and 93 within introductory chemistry, since these are the two domains in which CLP is being used. The examples chosen for our ink type prediction experiments include diagrams and text, and span 8 different types and 14 subtypes.

We ran 20 different experiments on our collected ink samples to observe how accurately classifiers could predict expected types and subtypes (see Figure 2 for some examples). Each experiment used a subset of the types we want to be able to predict. Based on our previous work, we hypothesized that we would be able to accurately predict the correct type at least 80% of the time, and that greater accuracy is obtained with fewer types in the experimental subset.

### 3.1. Classification

Ink type prediction is a classic class prediction problem in machine learning: using extracted features, for a particular sample we predict the class (type or subtype, in our case). These subtypes can be used to further specialize our type prediction: e.g., if our machine learning component predicts that a sample is a sequence, and a comma subtype is predicted, for example, the sample type can be specialized to a comma-delineated sequence, as opposed to just a sequence, with elements that could be delimited by anything. This contextual information is used by the sequence interpreter in its chunking algorithms, which employ heuristics to separate ink samples into smaller parts to simplify and improve interpretation.

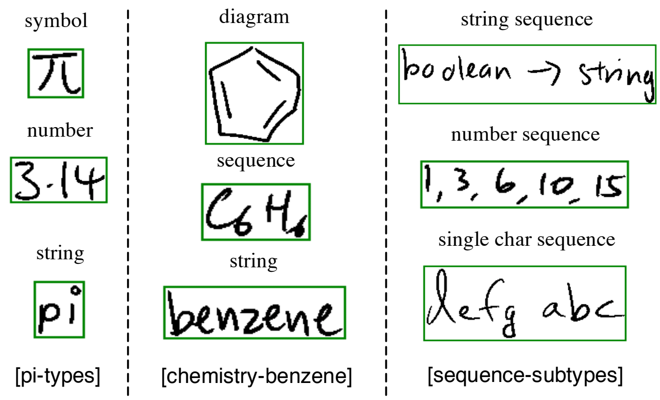


**Figure 2. Type/subtype prediction experiments**

### 3.2. Feature selection

The dynamic nature of digital ink strokes allows many possible temporal and spatial features to be extracted for machine learning. We extract information about individual strokes as well as the vector of all strokes in each ink sample. With only basic knowledge of our domain of expected answer types, we choose some distinct features to differentiate classes; others are generic features that we feel might be useful to the problem space of short written text or diagrams.

Some the features that we consider are:

1. Total number of strokes
2. Total number of positive stroke adjacent spacings
3. Sample height span
4. Sample width span
5. Sample width-height ratio
6. Stroke area density of points
7. Stroke horizontal density of points
8. Stroke heights
9. Stroke widths
10. Stroke lengths
11. Stroke points count
12. Stroke adjacent spacing
13. Stroke adjacent spacing differentials
14. Number of stroke intersections
15. Stroke angles
16. Stroke speeds
17. How close each stroke looks to a number

For each feature that applies to individual strokes (6-17), we extract information about the smallest and largest three values, as well as the 25th, 50th and 75th percentiles. We also consider the entire ink sample as a vector of strokes (for each of these features 6-17) and use this vector as an additional collective feature. For these feature vectors, we calculate their means and variances as additional scalar features.

Dimensionality reduction was then performed, the details of which have been omitted due to paper length.

## 4. Evaluation

We evaluate classification accuracy with several classification algorithms, each with distinctive learning methods, using the Waikato Environment for Knowledge Analysis (WEKA) [10]: an SVM trained with sequential minimal optimization (SMO) [11], a C4.5 decision tree [12], and a probabilistic Naïve Bayes classifier. We compute the accuracy of our class predictions using stratified cross validation (CV) randomized across each of the training and test sets.

The goal of the evaluation we describe in this paper is to highlight the variation in accuracy for some classifiers, rather than to find the perfect classifier for our ink type prediction.

### 4.1. Ink type prediction accuracy

We evaluated ink type prediction with two models: *K*-fold and leave-one-out CV, allowing us to obtain unbiased accuracy results by preventing testing on the same samples that were used during training. Figure 3 displays, for some *K*-fold CV experiments, the accuracy rates of predicting the correct type according to the number of features selected after dimension reduction. We see that there was no single best classifier, although SMO tended to perform better than the other two learners (see Table 1). Each experiment also requires a different optimum number of attributes to obtain peak accuracy in type prediction.

We also saw that leave-one-out CV still performed relatively well, with peak accuracies lower by only 6-10% than that obtained from *K*-fold CV. We discuss this observation later.

**Table 1. Prediction accuracy results for SMO**

| Experiment | # types | K-fold | Leave-one-out |
|---|---|---|---|
| number-string | 2 | 93.31 | 92.77 |
| number-sequence | 2 | 100.00 | 100.00 |
| sequence-commas | 2 | 100.00 | 100.00 |
| sequence-subtypes | 3 | 100.00 | 95.95 |
| chemistry-all | 3 | 99.52 | 99.05 |
| string-seq-scheme | 3 | 97.00 | 94.23 |
| compsci-no-number | 4 | 96.27 | 91.23 |
| compsci-no-boolean | 4 | 91.55 | 87.75 |
| compsci-all | 5 | 89.34 | 83.37 |

### 4.2. Overall interpretation accuracy

We also evaluated our dynamic dispatch interpretation system using ink type prediction on the basis of final interpretation accuracy for the domain of introductory computer science (the "compsci-all" experiment). Accuracy was measured with the edit distance between what was interpreted and the original example string used for input. We chose this domain, that consisted of five types (numbers, strings, sequences, true-false, Scheme expressions), because all of the student answers in this domain are in the form of text (as opposed to drawings), allowing us to make comparisons easily with other text interpreters, such as Microsoft's default interpreter (using TDNN), and the already deployed CLP interpreters, needing *a priori* contextual information.

Our approach described in this paper obtained close to 87% accuracy, comparable with the other interpreters developed for CLP (see Table 2). The main difference was that our dynamic dispatch interpreter required no contextual information to be provided *a priori* for each ink sample, and relied instead on machine learning to predict the expected type just from information extracted from the digital ink. Therefore, with the same basic ink input, our interpreter outperformed Microsoft's default interpreter by 24%, while nearly attaining the same level of accuracy as CLPv3 (see Figure 4).

**Table 2. Interpretation accuracy results**

| Base Type | CLPv3 | CLPv1 | Ours | Msft |
|---|---|---|---|---|
| 1. Number | 98.27 | 98.27 | 93.51 | 30.74 |
| 2. Scheme Exp. | 84.72 | 84.72 | 84.72 | 80.91 |
| 3. Sequence | 87.03 | 76.22 | 83.35 | 71.17 |
| 4. String | 78.06 | 78.06 | 74.08 | 54.95 |
| 5. True-False | 97.64 | 97.64 | 97.64 | 74.53 |
| **Total (%)** | **89.14** | **86.98** | **86.66** | **62.46** |

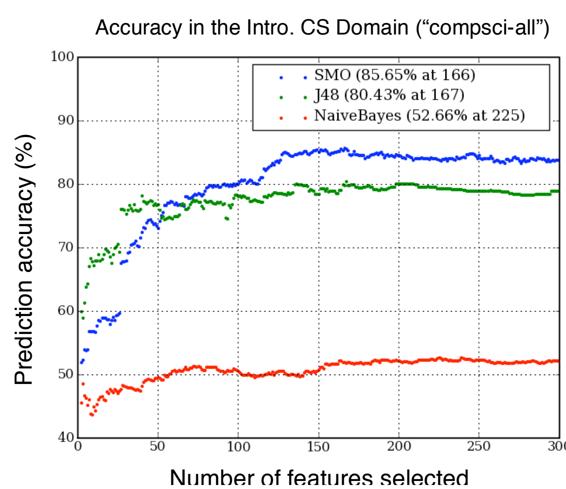


**Figure 3. Comparison of ML algorithms**

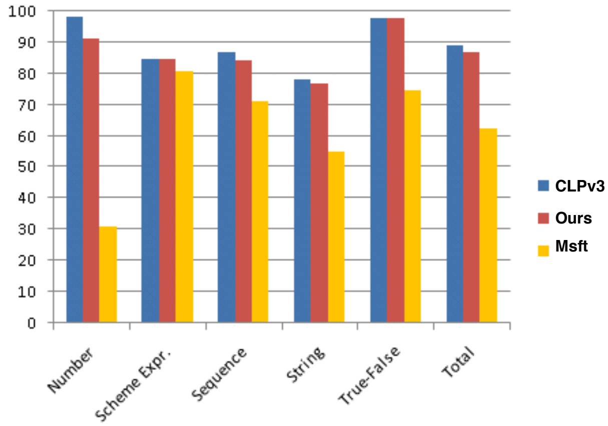


**Figure 4. Interpretation accuracy comparison**

### 4.3. Discussion

We observe that the accuracy of predicting the correct type in the number-string experiment was low, because it is inherently hard to tell whether a simple vertical stroke is a '1' (one), 'I' (capital-i) or 'l' (lowercase-L). If that stroke were to be slightly tilted, '/' or '\' would also be possibilities. This challenge illustrates the usefulness of biasing interpretation with contextual information. These ink samples, however, lack sufficient contextual information to predict their type correctly, thus lowering our accuracy rates in the number vs. string experiments.

Leave-one-out CV showed poorer prediction accuracy results than *K*-fold CV, mainly because the classifiers were not trained with all the tested representative samples in the former. The accuracy obtained, however, was still relatively high at more than 80%, showing it is possible to accurately predict correct expected types or flags of representative samples that have not been observed at all previously. The system would undoubtedly deteriorate in prediction performance the more examples we tested from outside our training subset. We thus would need to ensure that retraining is occasionally performed after deployment, which can be as simple as post-deployment supervised labeling of real data collected.

## 5. Contributions and future work

In this paper, we presented a novel application of AI methods to ink interpretation: improving accuracy by using machine learning to predict expected ink types for student digital ink answers in a classroom. Our machine learning approach extracts many features from the dynamic ink strokes and uses dimensionality reduction to generically improve prediction accuracy over the baseline for many experiment classes. We have deployed ink type prediction as a module to be used in actual CLP interpreters; these dynamic dispatch interpreters achieve far more accurate interpretations (87% accuracy) than the default Microsoft interpreter (62%), and close to that of the deployed CLP interpreters (87-89%) which require type information to be provided *a priori*.

We have shown that we can interpret ink reasonably well without the provision of *a priori* contextual information. We believe, however, that we can achieve an even higher level of accuracy if we now combine the benefits of our type prediction and dynamic dispatch with minimal amounts of contextual information. This combined approach is the focus of our current work.

## References

[1] K. Koile and D.A. Singer, Improving Learning in CS1 via Tablet-PC-Based In-Class Assessment. In *Proceedings of ICER 2006,* University of Kent, Canterbury, UK.

[2] K. Koile, K. Chevalier, C. Low, S. Pal, A. Rogal, D. Singer, J. Sorensen, K.S. Tay & K. Wu, Supporting Pen-Based Classroom Interaction: New Findings and Functionality for Classroom Learning Partner. *Proceedings of the First International Workshop on Pen-Based Learning Technologies 2007.*

[3] K. Koile, K. Chevalier, M. Rbeiz., A. Rogal, D. Singer, J. Sorensen, A. Smith, K.S. Tay & K. Wu, Supporting Feedback and Assessment of Digital Ink Answers to In-Class Exercises. *Proceedings* of *IAAI 2007*.

[4] B.S. Bloom, *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. New York: David McKay Co. Inc., New York, 1956.

[5] J. Hu, S.G. Lim & M. Brown, Writer independent on-line handwriting recognition using an HMM approach. In *Pattern Recognition Volume 33*, 133-147, January 1999.

[6] J. A. Pittman, Handwriting Recognition: Tablet PC Text Input. In *IEEE 2007*.

[7] R. Plamondon and S. Srihari, On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. In *IEEE Trans. Pattern Anal. Machine Intelligence.*, 22, Jan. 2000, 63–85.

[8] C. O' Boyle, B. Smyth & F. Geiselbrechtinger, An Automatic Configuration System for Handwriting Recognition Problems. *13th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2000.*

[9] M. Rbeiz, Semantic Representation of Digital Ink in the Classroom Learning Partner. MIT EECS Master of Engineering thesis. May, 2006.

[10] I.H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques, 2nd edition,* Morgan Kaufmann, San Francisco, 2005.

[11] J. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization. *Advances in Kernel Methods - Support Vector Learning,* Schoelkopf, B., Burges, C., and Smola, A., eds., MIT Press, 1998.

[12] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.