Using Ensemble Learning for Detecting Data Abnormities in Databases

Jerome Mitchell
Elizabeth City State University
Campus Box 391
1704 Weeksville Road
Elizabeth City, NC 27909

Abstract - Software engineers at the University of Kansas have developed SmartXAutofill, an intelligent data entry assistant for predicting and automating inputs for eXtensible Markup Language (XML) and other text forms based on the contents of historical documents in the same domain. SmartXAutofill utilizes an ensemble classifier, which is a collection of a number of internal classifiers[1] where each individual internal classifier predicts the optimum value for a particular data field. As the system operates, the ensemble classifier learns which individual internal classifier works better for a particular domain and adapts to the domain without the need to develop special classifiers. The ensemble classifier has proven that it performs at least as well as the best individual internal classifier. The ensemble classifier contains a voting and weighting system for inputting values into a particular data field.

Because the existing technology can predict, suggest, and automate data fields, the investigator contributed in testing whether the same technology can be used to identify incorrect data. Given existing data transmitted by sensors and other instruments, the investigator studied whether the ensemble technology can identify data abnormalities and correctness in future sensor data transmission. The solution would be applied in a project funded by the

National Science Foundation, Polar Radar for Ice Sheet Measurements (PRISM), using innovative sensors to measure the thickness and characteristics of the ice sheets in Greenland and Antarctica, with the goal of understanding how the ice sheets are being affected by global climate change.

PRISM sensors continuously send information that is collected and catalogued. The ensemble classifier will check the data for correctness by predicting which values should be there, and if the actual values are different, it will flag the data as possibly corrupted, and allow an operator to later study it and determine if it is correct or not. This technology will allow the PRISM intelligent systems to automatically determine the correctness of sensor and other data, and contributes to the PRISM project by adding a level of intelligence and prediction to the sensor suite.

1. Introduction

1.1 Motivation

XML is a markup language designed to describe data and to focus on what data is [6]. It enables people in the same discipline to exchange data and information without disconcerting about the data format and media. As technology advances, people in a variety of domains have developed their

[1] Internal classifiers are synonymous with classification algorithms.

own applications to annotate their information. These applications, however, demand for large XML forms to be manually entered into XML documents, are for simple data entry, require a faultless match between the incomplete input form and the stored templates, and in addition, there is no support for the complexity and nested structures of XML grammars.

1.2 Approach

1.2.1 What is SmartXAutofill?

SmartXAutofill is an intelligent data entry assistant that predicts, suggests, and automates inputs for XML documents. The system uses an ensemble of classification algorithms to suggest the optimum value for a particular input field. SmartXAutofill can understand the complex XML grammars and uses a machine learning technique to formulate suggestions, which is based on a comparative match between data entered in the input form and data stored in a historical document collection.

Initially, SmartXAutofill was designed to support the automation of inputs for XML documents but was modified to illustrate the "classification problem" in artificial intelligence for text-based forms.

2. Ensemble Learning

Ensemble learning is a machine learning technique that selects a collection, or ensemble, of hypotheses from the hypotheses space and combines predictions [3]. An ensemble classifier is composed of a set of independent trained classification algorithms whose predictions are combined when classifying new instances. One of the most widely used ensemble methods is called boosting. Boosting is a machine learning meta-algorithm for performing supervised learning [4].

3. Ensemble and Suggestors

An ensemble classifier is a collection of a number of classification algorithms [1]. Each internal classifier uses different approximate techniques to predict the optimum field for a particular XML or text document. There are a number of classification algorithms that can be incorporated into an ensemble classifier, but at the University of Kansas, four individual classification algorithms were used for the XML phase: a K-Nearest Neighbor classifier that uses an instance based technique, a Naïve Bayesian classifier that uses a statistical technique, a frequency classifier which selects the most frequent value for an XML field, and a recency classifier that selects the most recently entered value for a field. In the text-based phase, three classifiers were used: the Naïve Bayesian classifier and the K-Nearest Neighbor with K being equal to 1 and 3. The ensemble classifier learns which individual internal classifier works better for a particular domain, and it adapts to the domain without the need to develop special classification algorithms for the numerous domains created by the user. The ensemble classifier has proven to perform at least as well as the best individual classification algorithm.

4. Voting and Weight

Because the ensemble classifier receives predictions from many classification algorithms, voting is used for parsing these inputs into a single suggestion. The voting system forms an agreement on which value is suggested by most of the classification algorithms. All the algorithms return the same number suggestions. Each of the algorithms' suggestions is ranked; if an internal classifier returns X suggestions, the top one receives a value of X, the second X-1, with the last suggestion receiving a value of one. In addition to their rank, suggestions are modified by the weight of their classifier. Initially, all classifiers have the

same weight, but it is modified depending on which classifier works better for different node in a domain.

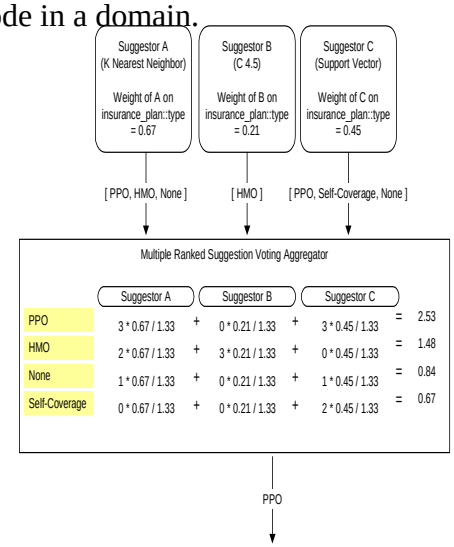

Figure 1: Classifier A proposes three suggestions; the first one receives a rank value of 3, the second one of 2, and the third one of 1; the rank values are multiplied by the weight of the internal classifier (0.67) and then normalized by the sum of the weights of all the classifiers. The same occurs for the suggestions by the other internal classifiers. The suggestion with the highest support is the one selected by the ensemble and presented to the user.

5. SmartXAutofill Design

There are two phases and three techniques on which the SmartXAutofill was created. Phase 1 predicts/suggests for all XML input fields and phase 2 predicts/suggests for the first and last input fields for text-based collections. Technique 2 portrays an ensemble of classification algorithms, and technique 3 also portrays an ensemble of classification algorithms but with a real world scenario.

5.1 How does it work?

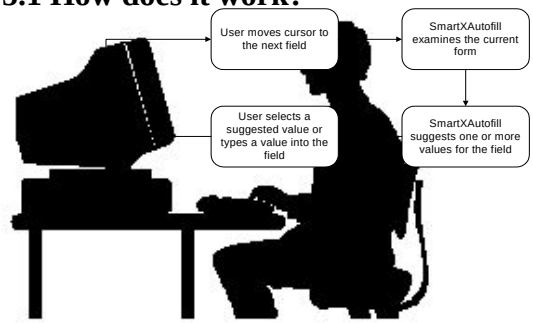

Figure 2: The steps of how the system is operated

5.1.2 Implementation

SmartXAutofill has three main subsystems - the suggestion engine, which makes recommendations for input fields, the TestingFramework, which is used to evaluate the performance of the system by emulating a user, and the human interface system, which provides a front-end Graphical User Interface (GUI) to the end user. The TestingFramework also writes a log file, which displays the details of each suggestion made by the suggestion engine. This log file is parsed and analyzed to evaluate the performance of the system [5].

Cross validation indicates the different testing and training routines, which is used by the TestingFramework. Cross validation is where data is divided into J subsets, and each time, one of the J subsets is used as the test set, and the other J-1 subset is used to form the training set.

Phases 1 and 2 and techniques 2 and 3 use three document collections: a seeded collection, a training collection, and a testing collection.

Seeded Collection - When the ensemble of classification algorithms starts to suggest a value, the seeded collection is provided to predict values. The documents in the seeded collection do not affect the confidence of the ensemble classifier; these documents are just appended to the collection.

Training Collection – The training collection is used to train the ensemble classifier. The ensemble classifier is a meta-learning system, which uses documents from the training assortment to train itself in order to learn from the suggestions. It picks out a document, suggests values for the classification node, modifies it's weight based on the results, and adds the document

to the document collection [2].

Testing Collection – The testing collection is used to perform testing on the ensemble classifier. The testing document is not appended to the document collection. The testing collection does not influence the weights of the ensemble classifier.

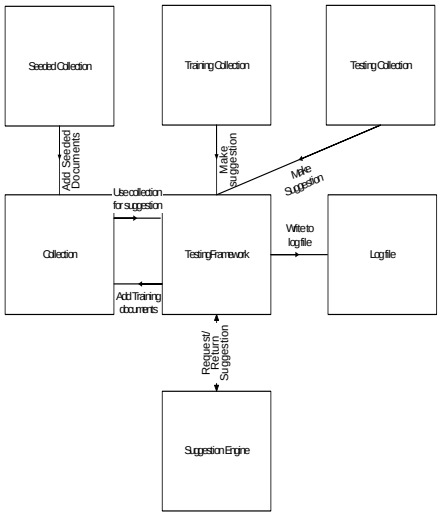

Figure 3: The ensemble starts with a seeded document collection, which is used for suggesting values. The seeded documents are not trained on and do not alter the weight of the suggestors. The weights of all the suggestors on all nodes are equal now; therefore, the suggestors have equal importance across all nodes. The suggestors are trained using the documents in the training collection. The system trains itself and changes the weights of the suggestors based on their performance. The testing process is used to perform testing on the ensemble classifier. Testing does not affect the weights of the ensemble classifier. (In technique 3, the system trains with all the documents in the training collection, while making suggestions for them.)

6. Contributions to SmartXAutofill

The investigator explored SmartXAutofill as it relates to the Polar Radar for Ice Sheet Measurements (PRISM) project. PRISM is funded through the National Science Foundation (NSF) as a tool for using sensors to measure the thickness and characteristics of the ice sheets in Greenland and Antarctica, and having a goal for understanding how the ice sheets are being affected by global climate change. The PRISM project is composed of multiple teams geared toward providing a centralized task toward measuring the ice sheets. The investigator worked with the Intelligent Systems team by enabling the NSF project to add a higher level of intelligence through the SmartXAutofill technology.

6.1 Procedures

SmartXAutofill, as it relates to the PRISM project, is responsible for detecting data abnormities and correctness in data transmitted by sensors and other instruments. In order to accomplish the large scale task, the SmartXAutofill was divided into several subtasks with text collections being used for this research. Subtask 1 employs predictions for all classification nodes. Subtask 2 provides the percentage for null suggestions, and subtask 3 detects false-positives, which enables detection of any incorrect values proposed from a user that would otherwise be correct. The investigator was responsible for expanding phase 2 into phase 3 (subtask1), which predicts for all classification text-based nodes. Before the ensemble classifier can train and test on text documents, the data was formatted in a comma delimited scheme, and afterwards, reading and writing source code in Java Builder was essential for implementing phase 3. The phase 3 model will better implement the correction in determining which values from data transmitted sensor instruments should be there.

6.1.1 Results

The ensemble of classification algorithms were trained and tested on all classification nodes for text collections from commercial data that is related to the results from sensory data by the PRISM project.

Key: 0 - Cross Validation; ! - # of Suggestions; * – Naïve Bayes; % – KNN-1; @ – KNN-3; ^ - **Ensemble**

Tables 1-4: Representing Accuracy of Suggestors & Ensemble for various Rewards

CB1

0	!	*	%	@	^
S1	3	.7241	.9063	.9003	**.9063**
S2	3	.7217	.8953	.8920	**.8981**
S3	3	.7224	.9052	.8990	**.9061**
S4	3	.7231	.9070	.9024	**.9072**
S5	3	.7228	.9058	.9001	**.9064**
Average		.7228	.9039	.8987	**.9048**

Table 1 displays the results from the Report Generator after the ensemble of classification algorithms was tested and trained. The ensemble classifier performed as well as the best individual internal classifier with 90% accuracy

CB2

0	!	*	%	@	^
S1	3	.5345	.7296	.7167	**.7235**
S2	3	.5459	.7304	.7228	**.7278**
S3	3	.5335	.7322	.7219	**.7257**
S4	3	.5387	.7344	.7272	**.7296**
S5	3	.5393	.7337	.7281	**.7037**
Average		.5383	.7320	.7233	**.72066**

Table 2 displays the results from the Report Generator after the ensemble of classification algorithms was tested and trained. The ensemble classifier performed as well as the best individual internal classifier with 70% accuracy.

CB3

0	!	*	%	@	^
S1	3	.5128	.6266	.5933	**.6109**
S2	3	.5179	.6341	.6030	**.6220**
S3	3	.5076	.6384	.6020	**.6210**
S4	3	.5139	.6399	.6104	**.6265**
S5	3	.5056	.6369	.5904	**.6177**
Average		.5115	.6351	.5998	**.6196**

Table 3 displays the results from the Report Generator after the ensemble of classification algorithms was tested and trained. The ensemble classifier performed as well as the best individual internal classifier with 60% accuracy.

CB4

0	!	*	%	@	^
S1	3	.3183	.3504	.3579	**.3485**
S2	3	.3179	.3503	.3584	**.3462**
S3	3	.3168	.3509	.3584	**.3465**
S4	3	.3132	.3495	.3576	**.3467**
S5	3	.3093	.3426	.3501	**.3376**
Average		.3151	.3487	.3564	**.3451**

Table 4 displays the results from the Report Generator after the ensemble of classification algorithms was tested and trained. The ensemble classifier did not perform as well as the best individual internal classifier It produced a 30% accuracy.

7. Conclusion

The SmartXAutofill predicts the optimum value for a particular data field in XML and text-based forms. SmartXAutofill, for the PRSIM project, was extended to phase 3 (stage 1) to predict all text classification nodes. The ensemble classifier performed as well as the best classification algorithm in 3 out of 4 domains. Table 4 created a lower accuracy for predicting and suggesting all classification nodes; this may result from unique nodes being suggested upon.

7.1 Future Work

Subtasks 2 and 3 from the SmartXAutofill technology can be implemented to display the effects for determining incorrect data in databases.

8. Acknowledgments

This research was conducted at the Information and Telecommunications Center (ITTC) at The University of Kansas. Funding was provided by the National Science Foundation (NSF) and Dr. L. Hayden. The research was mentored by Dr. P. Gogineni, Dr. C. Tsatsoulis, and Miss. D. Lee.

9. References

[1] Tsatsoulis, C. and Lee, D. “Intelligent Data Entry Assistant for XML Using Ensemble Learning”.
[2] Perry, S. and Subramanyan, B. Testing Technique – SmartXAutofill.
[3] Russell, S. and Norvig, P. Artificial Intelligence: A Modern Approach, Second Edition. Prentice Hall, 664-668.
[4] “Wikipedia,” URL http://en.wikipedia.org/wiki/Boosting/, July 14, 2005.
[5] Vaidyanathan, S. Design and Implementation Technique for Text Collections – SmartXAutofill
[6] “XML Usage,” URL http://www.w3school.com/xml/xml_usedfor.asp, July 25, 2005