Doctoral Consortium

A Framework for the Estimation of Students' Programming Abilities

Ella Albrecht
Institute of Computer Science
University of Goettingen
Göttingen, Germany
ella.albrecht@cs.uni-goettingen.de

ABSTRACT

In times of increasing numbers of students and high usage of e-learning systems, student models are a good way to get an overview of what is currently occurring in the classroom, analyze students' behavior and estimate their learning progress. In our work, we develop a framework which estimates a student's programming knowledge by looking at his responses to open-ended programming assignments. The model we construct incorporates multiple applications of multiple skills in one exercise, multiple submissions and varying knowledge components involved in the same exercise.

1. INTRODUCTION

During the last years, the number of students has increased rapidly. Especially in introductory courses, hundreds of students are attending. This makes it infeasible for educators to take care of each student individually. On the other hand, to deal with large amounts of students, many institutes use e-learning and e-assessment systems to support their teaching. These systems allow large data collection on which data mining and learning analytics techniques can be applied to build student models. Student models are used to estimate a student's cognitive state, e.g., his/her motivation, knowledge, misconceptions or learning style and preferences [4]. A student model can be used to provide students personalized course material fitting to their current knowledge and learning habits. Furthermore student models can be used to predict student's performance and identify students which are at risk to intervene in a timely manner. Besides, we can use a student model to identify problematic course contents. This knowledge can be used as a basis for restructuring and redesigning the course.

In our research, we want to develop a framework for the estimation of student's knowledge regarding programming. Therefore, we look at students' solutions to open-ended programming exercises. For each exercise, it is defined which knowledge components (KC) are required to solve the exercise correctly. KCs describe the individual components of knowledge which are required to solve a particular task or problem. The task in an introductory programming course is to learn to write simple programs which meet the specifications given in text form, i.e., the exercise description. Therefore KCs can be, e.g., the programming language's constructs, i.e., syntax and semantics, correct usage of a compiler or IDE, error understanding and debugging ability, or the translation of specifications to program code. Then, it is checked whether the student has applied the KCs in his/her solution correctly. From theses observations a student model can be constructed which is able to estimate a student's knowledge state.

2. PROBLEM STATEMENT

Knowledge cannot be assessed directly, because there may be several reasons why a student made a mistake. For example, a missing `break` in a `switch-case`-block may be just due to sloppiness, because the student does not know the `break`-statement, or because the student does not understand how the commands in a `switch-case`-block are executed. Because of these uncertainties often probabilistic models are used for student modeling.

Bayesian Knowledge Tracing (BKT) [5] is one of the most widely spread student modeling approaches. It uses Hidden Markov Models to model students' learning. It was at first applied to programming exercises for LISP in the ACT Programming Tutor. The domain knowledge was represented by production rules of the form "to achieve goal X do Y" where Y may be a subgoal. The knowledge of a student was described as the probability that the student knows a rule. Since there was a deterministic order of which rules need to be applied to solve an exercise correctly, the student's knowledge could be estimated by looking at the student's solutions rules order. But in imperative or object-oriented languages like C, C++, or Java one can only extremely rarely define a deterministic order of statements.

Kasurinen and Nikula [7] have applied BKT on students' results to Python exercises. As domain knowledge they have defined guidelines for preferred solutions, e.g., each open file should be closed. Moreover, they have checked whether the student has used the guideline in his/her solution. However, the set of KCs was very limited.

Berges and Hubwieser [2] as well as Yudelson et al. [10] used the Rasch model from Item Response Theory (IRT) to estimate student's knowledge of object-oriented concepts in Java instead. In IRT, the relationship between responses to items, i.e., exercises, and a latent trait, i.e., an ability or KC, is described as a logistic function. Different from BKT, it also takes the difficulty of an item into account.

BKT as well as IRT have the main drawback that they are single skill models, i.e., for each KC a separate model is constructed, and it is assumed that each exercise only requires one KC. For programming assignments, this assumption is of course not sustainable. Performance Factor Analysis (PFA) [8] is able to deal with multiple skills per exercise but as BKT and IRT also does not consider dependencies between KCs. However, in the programming domain there are dependencies between KCs, e.g., one needs to know how assignments or incrementing works when using a `for`-loop, or that the knowledge of a `while`-loop can influence the knowledge of a `for`-loop. It was also shown that integrating dependencies of knowledge into a student model can improve the model [3, 6]. Another special property of programming assignments is that KCs can be required multiple times in one exercise, e.g., if multiple loops are needed to solve the exercise. We also want to investigate the influence of substeps during the solution process to a model's accuracy. To the best of our knowledge, there does not exist a modeling approach so far which fulfills all of the requirements for programming assignments we have stated above.

3. RESEARCH METHODOLOGY AND APPROACH

Before we can make use of a student model in a course, several steps have to be taken. First, we need to identify what we expect the students to learn in our course, i.e., which KCs shall be acquired. In the first iteration of our research, the KCs we want to use for our model are the concepts of the programming language, e.g., `if`, `for`, variables, arrays etc., rules for good programming practice, e.g., each declared variable shall be used, allocated memory has to be freed, etc., as well as the fulfillment of the specifications by checking whether the program produces the correct output.

In a second step, we need to know which KCs are required to solve a particular exercise as we want to build our student model from the data we gain from their solutions to programming assignments. For example, summing up the numbers from 1 to 100 requires among others the knowledge of loops or recursion. This example also shows us, that it is actually not that easy to define which concrete concepts are really mandatory to solve the exercise as we could write a correct solution without knowing loops if we know recursion and vice versa. In our work, we develop a knowledge requirements model (KRM) which models required KCs related to language concepts for a particular exercise. The general mapping of language constructs, e.g., elements of an abstract syntax tree (AST), to concrete KCs has to be done beforehand by a domain expert. The KRM for a particular exercise is learned automatically from different correct solutions to that exercise based on their ASTs and structural analysis. We divide correct solutions into blocks and determine the set of KCs used in the block. From these sets we construct a tree where each path describes an alternative solution. By comparing a student's solution to the KRM, one can get the KCs which were applied correctly, incorrectly or are missing in the student's solution.

Despite the comparison with the KRM, we also use compiler and static analysis tool messages to assess the incorrect application of a KC, e.g., static analysis tools can deliver hints on, e.g., misunderstanding of control flow. Dynamic tests like unit tests, help us to evaluate a student's general program writing ability, i.e. whether a student is able to write a program which meets the specifications, i.e., does what it is intended to do.

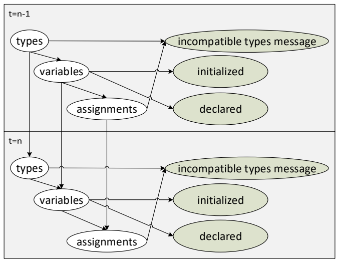


Figure 1: Example structure for a part of a DBN student model

The third step deals with the construction of the student model. We use Dynamic Bayesian Networks (DBN) for student modeling as they seem most appropriate to us. A DBN is a two-time-sliced Bayesian network where the state of a hidden variable depends on the states of the variables it depends on and the variable's state in the previous time step. Making observations in each time step updates the probability distribution of a hidden variable being in a particular state.

In our case, the hidden variables are the KCs, e.g., in Figure 1 the hidden variables (blank circles) are the concepts *types*, *variables*, and *assignments*. Observations in our student model are the results from the comparison of the student's solution with the KRM, compiler and static analysis tool messages as well as results from dynamic tests, e.g., in Figure 1 the observations (filled circles) are whether the student has declared and initialized a variable as well as whether an error message regarding incompatible types in an assignment appears. These variables can have the states *true* or *false*. With DBNs, we are able to deal with multiple KCs per exercise, their interdependencies, the uncertainty of which KC is affected by a certain observation and the uncertainty of which KCs are required to solve a particular exercise.

In our work, the structure of the DBN is defined manually by a domain expert. Though, one could also learn dependencies between KCs from data. The parameters of the DBN are learned from data using an expectation maximization algorithm with reasonable parameter constraints defined by an expert, e.g., limits for guess and slip probabilities. One problem that may occur, is that the parameter space is too large and we get computational problems when estimating the parameters of the model, if we use a very fine-grained KC definition. Therefore, we need to evaluate which granularity to choose to be able to estimate the parameters and still have an accurate model. Furthermore, we have to reason how to integrate multiple occurrences of the same KC in one exercise. Possible treatments are, e.g., majority vote or using uncertain evidences with a probability according to the ratio of correct/incorrect applications. We also want to analyze, whether multiple submissions, i.e., substeps preceding the final solution, improve the model.

In the second iteration of our research, we want to add further KCs which concentrate on more cognitive skills. The first one is the debugging ability, which we want to assess by comparing two subsequent submissions when the first one indicates an error (or a failure) and check whether the problem was fixed.

As a further KC, we want to include variable roles [9]. Variable roles describe patterns of variable usage. They are defined by the successive values the variables obtain. An example for a role would be the most-wanted holder which is a variable that holds the best value encountered so far when going through a succession of values, e.g., when searching the smallest value in an array. The proper collocation of variable roles is essential for solving a task or achieving a goal in a program. Usually, students intuitively use variable roles in their programs. The lack of knowledge of a particular role could explain why a student may have problems to solve an exercise.

We want to evaluate our model by comparing it to common student modeling approaches like BKT, IRT and PFA.

In a last step, we want to analyze the model constructed from the data of our introductory C course to find out what students which are at risk have in common, which KCs seem most difficult to the students and how many exercises are required at least (on average, to reach a particular percentage of students) to gain sufficient knowledge in a certain KC.

4. CURRENT STATUS & NEXT STEPS

We have implemented a framework for the collection of metrics regarding students' solutions [1] which was successfully introduced in our introductory C programming course. It is mainly an e-assessment system where students can upload their solution and get some basic feedback. It collects compiler messages, results from static analysis tools, and results from dynamic tests to capture the correctness of the solution. In the first year, we got about 10,000 submissions of on average 250 students. We expect similar numbers this year.

Furthermore, we have identified the different KCs that we have in our course by going through the course material and previous programming errors of students. Based on that, we defined a hierarchical structure of KCs where the sinks are basic observations in form of rules like, e.g., the function returns a value if the return type is not void. We have also mapped compiler/static analysis tool messages to different concepts and implemented an AST parser. In a next step, we want to use the AST to filter the KCs from source code and construct our KRM.

Next, we plan to conduct a small case study with only a few KCs to evaluate the feasibility of our DBN student model.

5. EXPECTED CONTRIBUTIONS

In our work, we develop a framework for the estimation of students' knowledge regarding programming. One of our main contributions is the definition of a student model which has the following properties which are needed to construct the model based on solutions to programming assignments: multiple KCs per exercise are possible and their interdependencies are considered, uncertainty of affected KCs can be handled, individual KC requirements and usages can be treated, multiple submissions can be integrated, and a KC can be used multiple times in the same exercise.

Another contribution will be a KRM which is automatically generated from model solutions for each exercise and can be used to evaluate which KCs were applied correctly or incorrectly by the student.

Furthermore, we plan to not just look at language related KCs, but also more cognitive skills like, e.g., debugging ability. We hope that our model helps to get better insights into the learning process of students.

From the doctoral consortium we expect to get some feedback on our student model, especially hints for the evaluation w.r.t. metrics and data sets. We are also looking forward for further ideas for additional or alternative KCs which we can integrate in our model.

6. REFERENCES

[1] E. Albrecht and J. Grabowski. Towards a framework for mining students' programming assignments. In *2016 IEEE Global Engineering Education Conference (EDUCON)*, pages 1096–1100, 2016.

[2] M. Berges and P. Hubwieser. Evaluation of source code with item response theory. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pages 51–56, New York, NY, USA, 2015. ACM.

[3] A. Botelho, H. Wan, and N. Heffernan. The prediction of student first response using prerequisite skills. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, pages 39–45, New York, NY, USA, 2015. ACM.

[4] K. Chrysafiadi and M. Virvou. Review: Student modeling approaches: A literature review for the last decade. *Expert Syst. Appl.*, 40(11):4715–4729.

[5] A. T. Corbett and A. Bhatnagar. *Student Modeling in the ACT Programming Tutor: Adjusting a Procedural Learning Model With Declarative Knowledge*, pages 243–254. Springer, Vienna, 1997.

[6] Y. Huang, J. Guerra, and P. Brusilovsky. A data-driven framework of modeling skill combinations for deeper knowledge tracing. In *Proceedings of the 9th International Conference on Educational Data Mining EDM*, pages 593–594, 2016.

[7] J. Kasurinen and U. Nikula. Estimating programming knowledge with bayesian knowledge tracing. In *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 313–317, New York, NY, USA, 2009. ACM.

[8] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis –a new alternative to knowledge tracing. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*, pages 531–538, Amsterdam, The Netherlands, 2009. IOS Press.

[9] J. Sajaniemi. An empirical analysis of roles of variables in novice-level procedural programs. In *Proceedings of the IEEE 2002 Symposia on Human Centric Computing Languages and Environments (HCC'02)*. IEEE Computer Society, 2002.

[10] M. Yudelson, R. Hosseini, A. Vihavainen, and P. Brusilovsky. Investigating automated student modeling in a java MOOC. In *Proceedings of the 7th International Conference on Educational Data Mining EDM*, pages 261–264, 2014.

Student Use of Inquiry Simulations in Middle School Science

Elizabeth McBride
University of California Berkeley
Tolman Hall
Berkeley, CA, USA
bethmcbride@berkeley.edu

ABSTRACT

My research focuses on the integration of science and design through the use of interactive simulations and other scaffolding tools. I specifically look at patterns of use in interactive simulations. To conduct this research, I have developed a curriculum about solar ovens used by middle school students, during which students are guided by an online curriculum to design, build, and test physical solar ovens. This curriculum utilizes interactive simulations as a tool to help students plan the design for their solar ovens. I have evaluated scaffolding for the simulation steps, and plan to evaluate other patterns of student use, based on action log data.

Keywords

Interactive Simulations, Science Education, Inquiry, Log Data

1. RESEARCH TOPIC

My research focuses on the integration of science and design through the use of interactive simulations and other scaffolding tools. I specifically look at patterns of use in interactive simulations. I conduct this research in secondary schools, and work in collaboration with teachers. Through my dissertation work, I aim to answer the following questions:

- What types of use patterns in interactive simulations are beneficial for integrating science and design learning?
- How can we use tools to support integrated understanding in writing activities (e.g.,automated guidance)?

My work is situated in the learning sciences, using techniques from educational data mining and artificial intelligence to understand how students' activities impact their learning and how to improve the learning experience. Recently, I have used natural language processing to develop automated classifiers for multiple short response questions [6]. Using these classifiers, I plan to develop automated guidance for student writing during the curriculum, which will deploy during spring 2017. I have also studied student use of interactive simulations, using log data, feature engineering, and clustering to make sense of patterns (submitted to EDM 2017).

To conduct this research, I have developed a curriculum that is run using an online platform and offers students the opportunity to use interactive simulations while they design a physical artifact. In previous work, I have found that the simulation is beneficial, especially when students use it during the design phase of the curriculum [8]. My work has also been published in a variety of other conference venues [7, 10, 11, 9].

1.1 Curriculum

My research utilizes a curriculum about solar ovens that is run using the Web-based Inquiry Science Environment (WISE). During this curriculum, students design, build, and test a solar oven. They go through the design, build, test process two times to get an idea of how engineers iterate on their designs based on results from testing (Figure 1). This curriculum was designed using the knowledge integration framework [5]. The knowledge integration framework has proven useful for design of instruction featuring dynamic visualizations [14] and engineering design [1, 12]. The framework emphasizes linking of ideas by eliciting all the ideas students think are important and engaging them in testing and refining their ideas [5].

Students are allowed to use only a certain set of materials (e.g., tin foil, black construction paper, plastic wrap, Plexiglas, tape), in addition to a cardboard box they bring from home. Students use an interactive computer simulation to test the different materials in their oven. This simulation helps to elicit student ideas before they get to the building process, consistent with the knowledge integration framework. The testing portion of the project allows students to distinguish their ideas.

Throughout the project, students respond to short response questions about the choices they are making in their design and how their ovens work. This curriculum is unique, since it is guided by an online platform, but students also design, build, and test their solar ovens in a hands on portion of the project.

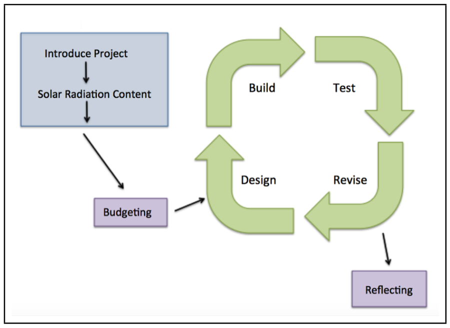


Figure 1: Outline of the solar ovens curriculum

The curriculum takes between 10-15 class periods (45 minutes per class period). Students complete this project in groups of 2 or 3 students. Students also complete a pretest the day before the project begins and a posttest the day after completing the solar ovens project. Students do the pretest and posttest individually. The pre-/posttests measure student understanding of science concepts and practices.

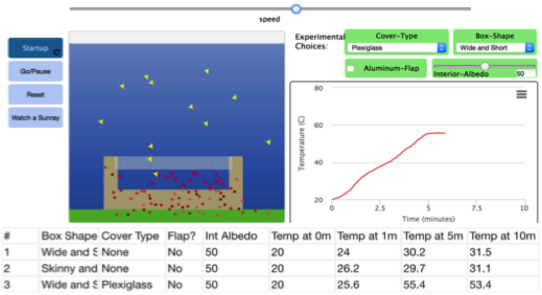
1.2 Interactive Computer Simulation

The interactive simulation (figure 2) was built using NetLogo [15]. Students can manipulate the simulation in a number of ways. They can change the cover on top of the oven, whether or not there is a reflective flap on top of the box, the shape of the box (wide and short or skinny and tall), and the albedo (reflectivity) of the inside of the box. Students may also manipulate the speed at which the simulation runs. Once a simulation runs to the end of the graph (10 simulated minutes), a new row is added to the table below the visualization with the settings and results from the trial. If the students do not allow the simulation to run until the simulated 10 minutes finish, nothing is added to the table.

The scaffolds we developed for the interactive simulation are twofold; short response questions direct students to investigate capabilities and limitations of the simulation and an automatically generated table helps students to keep track of trials they have run. The table includes information about all of the settings used in that trial, as well as the results of the trial at certain time points (e.g. 5 minutes, 10 minutes).

2. PROPOSED CONTRIBUTIONS

Making sure students use interactive simulations to aid in learning is a difficult task. To try to encourage students to take advantage of these simulations during learning, various scaffolding methods have been used. Often, these scaffolds are implicit, or built into the system with the simulation [13]. For example, guiding questions are used with inquiry simulations to direct students' attention toward certain features of simulations [4]. Students are also often encouraged in science classes to run multiple trials and control variables between trials (only change one variable between trials). A control of variables strategy can help students to determine the effect of a single variable on a more complex system, although in some cases students may benefit from more exploratory strategies [12].



Experimental Choices:
Cover-Type: Plexiglass
Box-Shape: Wide and Short
Interior-Albedo: 50

#	Box Shape	Cover Type	Flap?	Int Albedo	Temp at 0m	Temp at 1m	Temp at 5m	Temp at 10m
1	Wide and S	None	No	50	20	24	30.2	31.5
2	Skinny and	None	No	50	20	26.2	29.7	31.1
3	Wide and S	Plexiglass	No	50	20	25.6	55.4	53.4

Figure 2: The interactive simulation used by students to test solar ovens and visualize energy transformation; below the table simulation is output from the automatically generated table

Using log files from student interactions with the curriculum and output from the automatically generated tables (simulation scaffolding), we use feature engineering to identify how students use the model and whether these uses have an impact on learning. I developed features that have to do with the control of variables strategy, such as the number of trials (rows) a student runs and the percent of those trials that are systematic. These types of techniques have also been used with more complex simulations and microworlds (e.g., [3, 2]). We use results from pre- and posttests to assess student learning in tandem with the log data from the curriculum.

The data in this work comes from 635 students across three schools and five teachers. During this study, students participated in a pretest and posttest (each lasting one class period), as well as the 2-3 week long curriculum. During the curriculum, students worked in teams of 2-3. These 635 students formed 255 teams.

3. RESULTS

I used pretest and posttest scores to understand the effect of actions with the simulation on learning. I then examined the role the number of rows of data a student generated using the table scaffolding on learning. I found that the number of rows generated in iteration 1 of the simulation is a significant predictor of individual posttest scores, when controlling for pretest scores and curriculum group (b = 0.10, t(546) = 2.68, p < 0.01). Next, I examined the impact of controlling variables on learning. I found that the number of *Control Of Variables (COV) Trials* run, however, is not quite a significant predictor of posttest score, when controlling for group and pretest score (b = 0.06, t(546) = 1.63, p = 0.10). In addition, using a dummy variable for conducting any *COV Trials* does not significantly predict posttest scores when controlling for pretest scores and group (b = 0.005, t(546) = 0.13, p = 0.90). Together, these results indicate that the control of variables strategy, while a good practice in science, is not as helpful for developing an understanding of the scientific principles at play in a simulation. More experimentation using the model is beneficial for developing a better understanding of the scientific concepts.

I then split the students up based on their actions during the simulation step (did not generate any rows in table, generated one row, generated 2 or more rows). I found that generating 2 or more rows in the table significantly predicts posttest scores, when controlling for pretest score and working group (b = 0.12, t(546) = 3.11, $p < 0.01$), though generating no rows or 1 row were not significant predictors. I also developed a variable, *Percent Systematic*, that is the percentage of the total rows a group generated that used the control of variables strategy. This variable has the ability to show more nuance in how students were employing the control of variables strategy, but was also not predictive in determining posttest scores, when controlling for pretest and group id (b = 0.05, t(508) = 1.32, $p = 0.188$).

There were also two short response scaffolding questions on the same step as the interactive simulation. I generated a variable based on the number of questions students answered (0, 1, or 2). This was predictive of posttest score, when controlling for pretest score and group id (b = 0.10, t(546) = 2.56, $p = 0.011$).

Overall, evidence suggests that students should be encouraged to experiment with the model and guided to produce at least two rows of data in the table to improve learning outcomes and use the short response questions. Perhaps changing more than one variable at a time in this type of environment indicates that students are spending more time thinking about possible outcomes. I have further examined this data using k-means clustering algorithms.

4. FURTHER QUESTIONS

I have finished the majority of data collection for my dissertation. I will conduct one more study during the spring of 2017, and there will be the potential for a follow-up study later. This is an important time for me to get feedback on my work, especially on the analysis of the action log data I have collected from over a thousand students. I will begin the writing phase of my dissertation work during the summer, and expect to complete my dissertation within the next 12 months.

During the doctoral consortium, I would like to discuss the following:

- How to assess patterns in student actions in interactive simulations (Tools and packages for doing this and assessment of what it means to be a meaningful pattern)
- Designing studies that integrate education theory and data mining
- Assessment of inquiry skills in online environments
- Use of event logs in online curriculum to assess student use of curriculum and how this can be used to assess learning in tandem with other methods

5. REFERENCES

[1] J. Chiu, P. Malcolm, D. Hecht, C. DeJaegher, E. Pan, M. Bradley, and M. Burghardt. Wisengineering: Supporting precollege engineering design and mathematical understanding. *Computers & Education*, 67:142–155, 2013.

[2] C. Conati, L. Fratamico, S. Kardan, and I. Roll. Comparing representations for learner models in interactive simulations. In *Artificial Intelligence in Education*, pages 74–83. Springer, 2015.

[3] J. D. Gobert, Y. J. Kim, M. A. Sao Pedro, M. Kennedy, and C. G. Betts. Using educational data mining to assess studentsâĂŹ skills at designing and conducting experiments within a complex systems microworld. *Thinking Skills and Creativity*, 18:81–90, 2015.

[4] C. Hmelo and R. Day. Contextualized questioning to scaffold learning from simulations. *Computers & Education*, 32(2):151–164, 1999.

[5] M. Linn and B. Eylon. *Science learning and instruction: Taking advantage of technology to promote knowledge integration*. Routledge, 2011.

[6] E. McBride, A. Dixit, and M. Linn. Submitted – using machine learning to automatically score text responses in middle school science projects. In *Proceedings of the 18th International Conference on Artificial Intelligence in Education*, 2017.

[7] E. McBride, M. Martinez-Garza, J. Vitale, and M. Linn. Middle school student use of interactive climate change simulations: Periods of observation and activity. In *Paper presented at the Annual Meeting of the American Educational Research Association, San Antonio, TX*, 2017.

[8] E. McBride, J. Vitale, L. Applebaum, and M. Linn. Use of interactive computer models to promote integration of science concepts through the engineering design process. In *Proceedings of the 12th International Conference of the Learning Sciences*, Singapore, Singapore, June 2016 2016.

[9] E. McBride, J. Vitale, L. Applebaum, and M. Linn. Examining the flow of ideas during critique activities in a design project. In *Proceedings of the 12th International Conference of Computer Supported Collaborative Learning*, June 2017 2017.

[10] E. McBride, J. Vitale, L. Applebaum, J. Madhok, and M. Linn. Using virtual models to improve science understanding in a hands-on solar ovens unit. In *Paper presented at the Annual Meeting of the American Educational Research Association, San Antonio, TX*, 2017.

[11] E. McBride, J. Vitale, H. Gogel, M. Martinez, Z. Pardos, and M. Linn. Predicting student learning using log data from interactive simulations on climate change. In *Learning @ Scale*, Edinburgh, UK, April 2016 2016.

[12] K. McElhaney and M. Linn. Investigations of a complex, realistic task: Intentional, unsystematic, and exhaustive experimenters. *Journal of Research in Science Teaching*, 48(7):745–770, 2011.

[13] N. Podolefsky, E. Moore, and K. Perkins. Implicit scaffolding in interactive simulations: Design strategies to support multiple educational goals. *Chemistry Education Research and Practice*, 14(3):257–268, 2013.

[14] K. Ryoo and M. Linn. Can dynamic visualizations improve middle school students' understanding of energy in photosynthesis? *Journal of Research in Science Teaching*, 49(2):218–243, 2012.

[15] U. Wilensky. Netlogo. 1999.

Developing Chinese Automated Essay Scoring Model to Assess College Students' Essay Quality

Ju-Lu, Yu
Graduate Institute of Educational Information and Measurement, National Taichung University of Education, Taiwan
No.140, Minsheng Rd., West Dist.,
Taichung City 40306, Taiwan (R.O.C.)
ddog5633@yahoo.com.tw

Bor-Chen Kuo
Graduate Institute of Educational Information and Measurement, National Taichung University of Education, Taiwan
No.140, Minsheng Rd., West Dist.,
Taichung City 40306, Taiwan (R.O.C.)
kbc@mail.ntcu.edu.tw

Kai-Chih Pai
Graduate Institute of Educational Information and Measurement, National Taichung University of Education, Taiwan
No.140, Minsheng Rd., West Dist.,
Taichung City 40306, Taiwan (R.O.C.)
minbai0926@gmail.com

ABSTRACT

The present study aimed at proposing a Chinese automated essay scoring model to assess college students writing quality. Thirty-one related Chinese linguistic indicators were developed based on Coh-Metrix indices and characteristics of Chinese texts. Essay collected from 277 college students were analyzed using automated Chinese text analyze tool. A stepwise regression was used to explain the variance in human scores. The number of words, number of low strokes, content words frequency, minimal edit distance (all words) and minimum frequency for content words predicted 55.8% variance in human scores. On the other hand, seven indicators: number of words, content words frequency, concreteness, Measure of Textual Lexical Diversity, minimal edit distance (part of speech), minimal edit distance (all words) and words per sentence were predictive of human essay ratings by using discriminant analysis. The present study further explored the effectiveness of the Chinese automated essay scoring model by using three different methods: stepwise linear regression, discriminant analysis, and Nonparametric Weighted Feature Extraction classification (NWFE). The preliminary results showed that NWFE classification method produced higher exact matches (51.3%) between the predicted essay scores and the human scores than stepwise regression (47.3%) and discriminant analysis (47.3%).

Keywords

Chinese automated essay scoring, writing quality, NWFE classification, Chinese linguistic indicators

1. INTRODUCTION

Essay scoring has traditionally relied on expert raters. These scoring methods need to spend more time and a large amount of human scoring. Based on these limitations, automated essay scoring becomes the important research for essay assessment. According to the results of past studies, automated essay scoring reported perfect agreement (i.e., the exact match of human and computer scores) from 30-60% and adjacent agreement (i.e., within 1 point of the human score) from 85-99% [1]. Moreover, recently the study of analyzing the scored essays using Coh-Metrix has increased noticeably [2, 4, 5, 6, 7, 8, 13, 14, 15]. Coh-Metrix is an automated text analysis tool that provides lots of different linguistic indices [10]. The tool can provide these indices by combining lexicons, a syntactic parser, and several other components that are widely used in computational linguistics.

Chinese language features in the characteristics of different from the English, cannot be directly applied to the Chinese essay writing. Most of the experts will consider the following sections: Number of words, structure organization, vocabulary diversification, typos, and punctuation. Based on the development of Coh-Metrix, automated text analyze tool were developed in Chinese. Totally 66 Chinese related linguistic indicators were used to analyze the characteristics of Chinese texts [12].

Writing the literacy assessment is an important standardized testing to assess college students' writing skill in Taiwan. The assessment is to detect whether students can express personal comments on specific issues. Students need to read an article, respectively, and express personal comments by writing the essay in two hundred words. These essays were scored by two experts and score from 0-5. However, we need to a lot of experts and spend more time to score. To propose a suitable automated scoring model is important and needed.

2. PROPOSED CONTRIBUTIONS

The purpose of the study is to explore the characteristics of Chinese writing and propose a suitable Chinese automated essay scoring model to assess college students writing quality. Past studies explored the variety of human scoring were predicted by different text features using regression analysis. Moreover, they proposed automated essay scoring model and examined the essay matches by linear regression and discriminant analysis. A Nonparametric Weighted Feature Extraction (NWFE) classification method was also used to examine the essay matches in the present study.

Nonparametric Weighted Feature Extraction (NWFE) is based on a nonparametric extension of scattering matrices. It could reduce parametric dimensional and increase classification accuracy [11]. The present study used linear regression analysis and discriminant analysis of the gradual selection of variables for the NWFE classification method and examine the accuracy of essay matches.

3. Method

3.1 Text Indices Selection Procedure

The present study collected Chinese essay from college students in Taiwan. All essay was analyzed by Chinese automated text analyze tool. The tool provides 62 Chinese linguistic indices, includes basic text measures (e.g., text, sentence length), words information (e.g., word frequency, concreteness), cohesion (semantic and lexical overlap, lexical diversity, along with the incidence of connectives), part of speech and phrase tags (e.g., nouns, verbs, adjectives), and syntactic complexity (e.g., Sentence syntax similarity, Minimal Edit Distance).

The first step, correlation analyses was conducted to examine the strength of relations between the selected indices and the human scores of essay quality. Text indices retained based on a significant correlation with human scores. Multicollinearity was then assessed between the indices (r >.900). The index retained based the strongly with human scores when two or more indices demonstrated multicollinearity. Finally, totally thirty-one indices were used in the study.

3.2 Essay Scoring

277 essays were collected from college students in Taiwan. Each essay in the study was scored independently by two expert raters using a 5-point rating. The rating scale was used to assess the quality of the essays and had a minimum score of 0 and a maximum score of 5. The experts evaluated the essays based on a standardized rubric used in the Chinese writing literacy assessment in Taiwan. The results of correlation between two experts are 0.788. It indicated that consistency of expert scoring.

3.3 Essay Evaluation

Three different methods were used to examine the accuracy of automated essay scoring: linear regression analysis, discriminant analysis, and NWFE classification. Text features were selected by linear regression and discriminant analysis. The leave-one-out method was used to experiment with training essay set and testing the essay set. The present compared the exact matches of the essay by using the three methods.

4. Preliminary Results

4.1 Linear Regression Analysis: Text Features

A stepwise regression analysis was conducted to examine which text indicators were predictive of human essay ratings. 40 Chinese text features were used in the study. The results presented in Table 1. Five indicators were a significant predictor in the regression model: Number of words, the number of low strokes, content word's frequency, minimal edit distance (all words) and the minimum frequency of content words, F = 12.074, p <.001, r =.747, r^2 =.558. The results from the linear regression demonstrate that the five variables account for 55.8% of the variance in the human scoring of writing quality.

Table 1. Stepwise regression results for text features

Indicators	*B*	*SE*	Β
number of words	.011	.001	.529
number of low strokes	.000	.000	-.131
content words frequency	.824	.402	.086
minimal edit distance (all words)	2.334	.618	.238
minimum frequency for content words	-.148	.042	-.154

4.2 Discriminant Analysis: Text Features

The purpose of the discriminant analysis was to examine whether features are predictive of human scoring. The results of the discriminant analysis showed that seven text features could predict human scorning, includes the number of words, content word frequency, concreteness, Measure of Textual Lexical Diversity, minimal edit distance (part of speech), minimal edit distance (all words) and words per sentence.

4.3 Exact and Adjacent Matches

Table 2 and Table 3 presented the results of exact and adjacent matches. The linear regression analysis (stepwise) selected features: The number of words, number of low strokes, content words frequency, minimal edit distance (all words) and minimum frequency for content words. The exact matches (leave-one-out) between the predicted essay scores (rounded to 0-5) and the human scores is 47.3% exact accuracy and 95.3% adjacent accuracy.

The discriminant analysis (stepwise) selected features had the number of words, word frequency of content words, minimal edit distance (local), MTLD, the number of terms, concreteness, and minimal edit distance (part of speech). The exact matches (leave-one-out) between the predicted essay scores and the human scores is 47.3% exact accuracy and 93.9% adjacent accuracy.

The present study conducted NWFE classification method to examine the effectiveness of automated essay scoring. The results showed that 48.7% exact matches between predicted scores and human scoring, which text features selected by linear regression. Moreover, 51.3% exact matches between predicted scores and human scoring, which text features selected by discriminant analysis.

Table 2. Comparison of Exact

Classification method	Text features selected by linear regression	Text features selected by Discriminant
Linear regression	47.3%	46.6%
Discriminant	45.5%	47.3%
NWFE	48.7%	51.3%

Table 3. Comparison of Adjacent

Classification method	Text features selected by linear regression	Text features selected by Discriminant
Linear regression	95.3%	93.9%
Discriminant	94.2%	93.9%
NWFE	89.9%	90.3%

5. Conclusion

Past studies have found that the number of words was an important indicator of human score [4, 15]. The results of the study also presented that the number of words has a high significant correlation with human scores. The number of words, the minimal edit distance (local), and the number of low strokes three indicators belong to Descriptive and Syntactic Complexity categories in Coh-Metrix. MTLD belongs to Lexical Diversity. These indicators are related the scoring guide of writing for college students in Taiwan.

Comparing exact matches between linear regression analysis (stepwise) and discriminant analysis (stepwise). The results of leave-one-out of exact matches linear regression and discriminant analysis showed consistency. Moreover, regardless of method linear regression analysis (stepwise) or discriminant analysis (step-wise) selection indicators, the accuracy of exactly matched of NWFE method is higher than the other two classification methods.

6. Future Works

Past studies have investigated the potential for component scores that are calculated using the linguistic features by Coh-Metrix in assessing text readability [9, 12]. Moreover, one study has explored correlations between human ratings of essay quality and component scores based on similar natural language processing indices and weighted through a principal component analysis [2]. However, this approach has not been extended to computational assessments of essay quality In Chinese. The present study will adapt a similar approach to passing studies [9, 12]. We will conduct a principle component analysis (PCA) or factor analysis to reduce the number of indices selected from Chinese automated text analyze tool into a smaller number of components comprised of related features. The present study will further explore the correlation between component scores and human scoring. A Chinese automated essay scoring model based on text component scores will be developed and explored.

7. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of this study by the Ministry of Education.

8. REFERENCES

[1] Attali, Y., & Burstein, J. 2006. Automated Essay Scoringwith E-rater V.2. *Journal of Technology, Learning and Assessmen*t, 43.

[2] Crossley, S. A., & McNamara, D. S. 2014. Does writing development equal writing quality? A computational investigation of syntactic complexity in L2 learners. *Journal of Second Language Writing,* 26, 66-79.

[3] Crossley, S. A., & McNamara, D. S. 2014. Developing component scores from natural language processing tools to assess human ratings of essay quality. In W. Eberle & C. Boonthum-Denecke (Eds.), *Proceedings of the 27th International Florida Artificial Intelligence Research Society (FLAIRS) Conference* (pp. 381-386). Palo Alto, CA: AAAI Press.

[4] Crossley, S. A., Dempsey, K., & McNamara, D. S. 2011. Classifying paragraph types using linguistic features: Is paragraph positioning important? *Journal of Writing Research*, 3, 119-143.

[5] Crossley, S. A., Roscoe, R. D., & McNamara, D. S. 2013. Using automatic scoring models to detect changes in student writing in an intelligent tutoring system. In C. Boonthum-Denecke & G. M. Youngblood (Eds.), *Proceedings of the 26th International Flordia Artificial Intelligence Research Society (FLAIRS) Conference*, 208-213. Menlo Park, CA: The AAAI Press.

[6] Crossley, S.A. & McNamara, D.S. 2010. Cohesion, coherence, and expert evaluations of writing proficiency. In S. Ohlsson & R. Catrambone (Eds.), *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, 984-989. Austin, TX: Cognitive Science Society.

[7] Crossley, S.A., & McNamara, D.S. 2011. Text coherence and judgments of essay quality: Models of quality and coherence. In L. Carlson, C. Hoelscher, & T.F. Shipley (Eds.), *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 1236-1231. Austin, TX: Cognitive Science Society.

[8] Guo, L., Crossley, S. A., & McNamara, D. S. 2013. Predicting human judgments of essay quality in both integrated and independent second language writing samples: A comparison study. *Assessing Writing*, 18, 218-238.

[9] Graesser, A.C., McNamara, D.S., and Kulikowich, J. 2012. Coh-Metrix: Providing multilevel analyses of text characteristics. *Educational Researcher*, 40(5), 223-234.

[10] Graesser, A. C., McNamara, D. S., Louwerse, M. M., & Cai, Z. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, and Computers*, 36, 193-202.

[11] Kuo B.-C., and Landgrebe, D. A. 2004. Nonparametric weighted feature extraction for classification, *IEEE Transactions on Geoscience and Remote Sensing*, 42(5), 1096-1105.

[12] Kuo B.-C., and Liao C.-H. 2014. The Automated text analysis for Chinese text. *2014 Workshop on the Analysis of Linguistic Features (WoALF 2014)*, Taipei, Taiwan.

[13] McNamara, D. S., Crossley, S. A., Roscoe, R. D., Allen, L. K., & Dai, J. 2015. A Hierarchical classification approach to automated essay scoring. *Assessing Writing*, 23, 35-59.

[14] McNamara, D.S., Graesser, A.C., McCarthy, P., & Cai, Z. *Automated evaluation of text and discourse with Coh-Metrix.Cambridge*: Cambridge University Press, 2014.

[15] Roscoe, R.D., Crossley, S.A., Weston, J.L., & McNamara, D.S. 2011. Automated assessment of paragraph quality: Introductions, body, and conclusion paragraphs. In R. C. Murray & P. M. McCarthy (Eds.), *Proceedings of the 24th International Florida Artificial Intelligence Research Society (FLAIRS) Conference*, 281-286. Menlo Park, CA: AAAI Press.

Teaching Informal Logical Fallacy Identification with a Cognitive Tutor

Nicholas Diana
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
ndiana@cmu.edu

John Stamper
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
john@stamper.org

Kenneth R. Koedinger
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
koedinger@cmu.edu

ABSTRACT

In this age of fake news and alternative facts, the need for a citizenry capable of critical thinking has never been greater. While teaching critical thinking skills in the classroom remains an enduring challenge, research on an ill-defined domain like critical thinking in the educational technology space is even more scarce. We propose a difficulty factors assessment (DFA) to explore two factors that may make learning to identify fallacies more difficult: type of instruction and belief bias. This study will allow us to make two key contributions. First, we will better understand the relationship between sense-making and induction when learning to identify informal fallacies. Second, we will contribute to the limited work examining the impact of belief bias on informal (rather than formal) reasoning. We discuss how the results of this DFA will also be used to improve the next iteration of our fallacy tutor, how this tutor may ultimately contribute to a computational model of informal fallacies, and some potential applications of such a model.

Keywords

Cognitive Tutors, Informal Logical Fallacies, Informal Reasoning, Cognitive Task Analysis, Difficulty Factors Assessment

1. INTRODUCTION

Despite the recognized importance of critical thinking in traditional education, critical thinking is largely absent from the educational technology space (e.g., online courses/MOOCs, cognitive tutoring systems, etc.). Some of the recent work on critical thinking in educational technology has focused on comparing critical thinking in face-to-face and computer-mediated interactions. Researchers often use content-analysis to identify instances of critical thinking in online and face-to-face discussions [3, 10]. In this work, critical thinking is not the primary focus of the course, but rather an epiphenomenon.

Other work, particularly in the domains of philosophy, writing and law, has addressed critical thinking more directly. For example, some recent work has demonstrated that argument diagramming using a graphical interface improved argumentative writing skills [6] as well as critical thinking skills more generally [5]. However, similar gains are seen using paper-and-pencil argument diagramming as well, suggesting the software may be more of a convenience than a necessary factor [4].

Despite the challenges of working in an ill-defined domain [8], another intersection of critical thinking and e-learning has been in intelligent tutoring systems (ITS). For example, Ashley and Aleven [1] built an ITS to teach law students to argue with cases more effectively. The study we propose extends this work on critical thinking in the ITS space to a more general population. We will build a cognitive tutor that teaches users to identify several common informal logical fallacies. We chose informal fallacies because they offer a degree of structure to the otherwise ill-defined domain of informal reasoning, making the content more amenable for use in a cognitive tutor. Using this tutor, we will conduct a difficulty factors assessment (a type of a cognitive task analysis) [7] to evaluate the impact of two factors on the user's ability to identify logical fallacies.

The first factor explored will be *type of instruction*. The Knowledge-Learning-Instruction (KLI) framework lists three types of learning processes, and suggests that the best instruction for teaching a specific skill depends on the type of process used to learn that skill. The purpose of the *type of instruction* manipulation is to better understand the learning processes that underpin the identification of logical fallacies. Specifically, we are interested in whether this skill is more efficiently learned using induction (e.g., showing many examples of the fallacy) or sense-making (e.g., providing detailed descriptions of the fallacy's mechanics). Textbooks used to teach logical fallacies often take both approaches, giving readers an explanation of a fallacy followed by some small number of examples. As this skill may consist of multiple, more fundamental skills (or knowledge components), the mixed approach used by textbooks may prove to be the most efficient. Nevertheless, the proportion of time to devote to each learning process remains an open question that this experiment may help answer.

The second factor that may negatively impact a student's ability to identify logical fallacies is *belief bias*, the tendency

Table 1: Breakdown of the problems used in the tutor. Note that *(F)*, *(A)*, *(C)*, and *(L)* correspond to *for*, *against*, *conservative* and *liberal*, respectively. For example, in the first cell of the table, we see an *apolitical* prompt, which *fallacy 1* is used to argue *for*.

	Apolitical	Political	Apolitical	Political	Apolitical	Political
Fallacy 1	(F)	(C)	(A)	(L)	(F)	(C)
Fallacy 2	(A)	(L)	(F)	(C)	(A)	(L)
Fallacy 3	(F)	(C)	(A)	(L)	(F)	(C)
Fallacy 4	(A)	(L)	(F)	(C)	(A)	(L)
Fallacy 5	(F)	(C)	(A)	(L)	(F)	(C)
Fallacy 6	(A)	(L)	(F)	(C)	(A)	(L)

to judge arguments more favorably if we agree with the conclusion. Early work on belief bias explored its effect on formal reasoning using syllogisms [9, 2], but there is some evidence that suggests that belief bias may operate differently in informal reasoning [11]. The proposed study builds on and contributes to this research by empirically testing the effect of belief bias on learning to identify informal fallacies.

2. FUTURE RESEARCH PLANS

2.1 Difficulty Factors Assessment

We will use a Difficulty Factors Assessment (DFA) to identify the factors (if any) that make it more or less difficult for students to learn how to identify logical fallacies. The proposed experiment will explore the impact of two primary factors as well as several secondary factors.

2.1.1 Type of Instruction

The proposed experiment will explore the impact of *type of instruction* by randomly assigning each participant to one of three conditions. In each condition, when the participant is given a problem and asked to identify the logical fallacy, they will be given a set of possible answers and the option to view more information about each of the answers. In the first condition, when participants ask for more information they will be shown a brief, but detailed description of the mechanics of each fallacy (sense-making). In the second condition, participants will be shown two examples of each fallacy (induction). In the the third condition, participants will be shown a description and one example for each fallacy (mixed).

In addition to comparing the effect of increased examples between groups, we will be able to compare this effect within groups by treating completed problems as viewed examples. This analysis will help us pinpoint the average number of examples needed to be able to identify the fallacies used in the experiment, and compare that number to the average numbers seen in common textbooks.

2.1.2 Belief Bias

The proposed experiment will explore the impact of *belief bias* on a student's ability to identify logical fallacies by altering the political orientation of problem content and comparing performance on those problems with the participant's personal political orientation. Of the 36 problems presented, half will be apolitical (i.e., politically neutral) and half will be political. Of the political problems, half will have a conservative orientation, half a liberal orientation. The apolitical problems are also split into two categories (for an issue or against an issue) for balance. Problems can be broken down into three subcomponents: the prompt (either political or apolitical), the fallacy, and the conclusion (either for/against or conservative/liberal). Table 1 shows the breakdown of each problem.

2.1.3 Secondary Factors Explored

In addition to the main effects of *type of instruction* and *belief bias*, our design also allows us to explore several secondary factors. We can test whether *type of instruction* has a differential effect on specific fallacies. For example, sense-making may be more important for learning to identify a circular argument, while examples may be sufficient for learning to identify a Post Hoc fallacy. We can also test whether participants are more likely to identify a fallacy given the nature of the prompt (political vs. apolitical) or the valence of the conclusion (for/against or conservative/liberal).

2.2 Towards a Computational Model of Logical Fallacies

The ultimate goal of this work is to develop a computational model of logical fallacies. Achieving this goal requires overcoming several large challenges.

2.2.1 Lack of Labeled Examples

First, to train a model to detect such a nuanced use of language will most likely require a large number of labeled examples. Furthermore, these examples will most likely have to be varied and authentic (perhaps unlike many of the purposefully illustrative examples used in textbooks). To solve this shortage of labeled examples, we propose using our cognitive tutor to train crowd workers to identify fallacies in real-world media sources. The quality of those labels can be evaluated using traditional crowdsourcing methods (e.g., consensus of the crowd). High quality labels can then be automatically integrated into the tutor training system, increasing the number of potential examples crowd workers can use to achieve mastery. This increase in the number of examples may be especially important if our DFA reveals that learning to identify informal fallacies is a primarily inductive skill. Figure 1 shows the feedback loop relationship between crowd workers and the cognitive tutor.

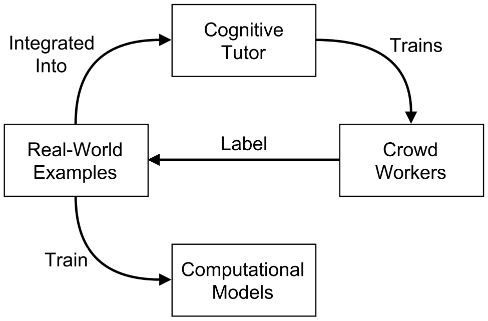


Figure 1: Feedback loop relationship between the cognitive tutor and crowd workers. The real-world examples labeled by crowd workers can be used to both improve the cognitive tutor and train computational models.

2.2.2 Modeling the Semantic Nature of Fallacies

Informal Logical Fallacies is an umbrella term that encompasses a diverse array of fallacies. Some of these fallacies may be easier for a machine learning model to detect. For example, the *Slippery Slope* fallacy often has the generic structure: "First X, pretty soon there'll be Y too!" These kinds of syntactic features will likely be easier to detect than the semantic features necessary to identify a fallacy like *Circular Reasoning*. Finding the right method for approaching these more difficult cases will be one of the key challenges of this work.

2.2.3 Potential Applications

If we meet these challenges and are able to detect logical fallacies in real-world text, there are potential applications in media (both traditional and social), politics, and education. One could imagine a plugin for your favorite word processor that underlines an *Appeal to Ignorance* just as it would a misspelled word. Similarly, one could imagine how broadcasts of presidential debates in the future might be accompanied by a subtle notification anytime a candidate uses *Moral Equivalence*.

In conclusion, we propose a plan to develop a computational model of informal logical fallacies. The first, and most concrete, step of this process is developing a better understanding of the factors that promote and hinder how we learn to identify informal fallacies. We propose a difficulty factors assessment to explore the impact of sense-making versus induction support, as well the impact of belief bias. Discovering how these factors regulate learning will not only allow us to build a better tutor, but will improve our understanding of how we learn informal logical fallacies in general.

3. REFERENCES

[1] K. D. Ashley and V. Aleven. Toward an intelligent tutoring system for teaching law students to argue with cases. In *Proceedings of the 3rd international conference on Artificial intelligence and law*, pages 42–52. ACM, 1991.

[2] J. S. Evans, J. L. Barston, and P. Pollard. On the conflict between logic and belief in syllogistic reasoning. *Memory & cognition*, 11(3):295–306, 1983.

[3] J. Guiller, A. Durndell, and A. Ross. Peer interaction and critical thinking: Face-to-face or online discussion? *Learning and Instruction*, 18:187–200, 2008.

[4] M. Harrell. No computer program required: Even pencil-and-paper argument mapping improves critical-thinking skills. *Teaching Philosophy*, 31(4):351–374, 2008.

[5] M. Harrell. Assessing the efficacy of argument diagramming to teach critical thinking skills in introduction to philosophy. *Inquiry: Critical Thinking Across the Disciplines*, 27(2):31–39, 2012.

[6] M. Harrell and D. Wetzel. Improving first-year writing using argument diagramming. *Proc. of the 35th Annual Conf. of the Cognitive Science Society*, (1987):2488–2493, 2013.

[7] K. R. Koedinger and A. Terao. A cognitive task analysis of using pictures to support pre-algebraic reasoning. In *Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society*, pages 542–547. Citeseer, 2002.

[8] C. Lynch, K. Ashley, V. Aleven, and N. Pinkwart. Defining ill-defined domains; a literature survey. In *Proceedings of the workshop on intelligent tutoring systems for ill-defined domains at the 8th international conference on intelligent tutoring systems*, pages 1–10, 2006.

[9] J. J. B. Morgan and J. T. Morton. The distortion of syllogistic reasoning produced by personal convictions. *Journal of Social Psychology*, 20(1):39–59, 1944.

[10] D. R. Newman, B. Webb, and C. Cochrane. A content analysis method to measure critical thinking in face-to-face and computer supported group learning. *Interpersonal Computing and Technology*, 3(September 1993):56–77, 1995.

[11] V. Thompson and J. S. B. T. Evans. Belief bias in informal reasoning. *Thinking & Reasoning*, 18(3):278–310, 2012.

Automated Extraction of Results from Full Text Journal Articles

R. Wes Crues
University of Illinois
Dept. of Educational Psychology
1310 South Sixth Street
Champaign, Illinois
crues2@illinois.edu

ABSTRACT

Recent mandates by federal funding agencies and universities to create open access repositories of published research allow researchers a wealth of texts to analyze. Furthermore, some publishers of academic texts have begun creating policies to permit non-commercial text mining of journal articles. This project follows the approach of [7], which automatically extracts result sentences from full-text biomedical journal articles by using support vector machines and naive Bäyes classifiers. I also experiment with using the least absolute shrinkage and selection operator (LASSO) [6, 18] as a method to select features for the classifiers. I compare this new approach with other feature selection strategies used in previous studies.

Keywords

Information extraction, text classification, feature selection

1. INTRODUCTION

Information overload is hardly a new concept, with even the Ancient Roman scholar Seneca the Elder claiming in 1 AD, "the abundance of books is distraction" [8]. Similarly, the automatic summarization of text has been researched since at least the 1950's, with Luhn's work on creating abstracts automatically [11]. In concert, United States (US) federal funding agencies, such as the National Institutes of Health (NIH) [13], the National Science Foundation (NSF) [14], and the Institute for Educational Sciences (IES) [9], and university systems such as the University of California (UC) [1] have adopted open access policies for funded and published research. Publishers of academic journals, such as Elsevier [4] and Springer [15], have adopted policies for non-commercial research of texts. Finally, some national governments (e.g., the United Kingdom (UK) [10]) have adopted changes to copyright law allowing for non-commercial research of copyright protected works.

Given these open-access and legal policy changes, a wide swath of researchers now have access to a wealth of texts to automatically analyze. Specifically, the shifts in policies and laws allows for text mining to extract result sentences from full-text journal articles. Further, publishers have created APIs which allow for access to texts. It is unlikely that future researchers will be able to carefully read and analyze all of the texts in order to extract pertinent results. However, open-access policies in the US by the NIH have enabled automated extraction since the late 2000s in some fields.

My research seeks to first expand the work done in the biomedical sciences, particularly in [7] to the educational sciences, but also to explore an additional feature selection technique. This experiment is to complement the work in [20] by using the LASSO as a feature selection technique.

2. BACKGROUND

Text mining has been recognized as a tool to reduce the time required to complete a systematic literature review [17]. There are several tasks text mining can simplify when creating a systematic review. Current text mining approaches allow relevant studies to be identified, by identifying relevant search terms, and describing the characteristics of prior investigations can be accomplished by automatic summarization [17]. This proposal is inspired by the systematic search of literature using targeted queries by the information scientist, Don Swanson, who revealed a link between magnesium and migraines in the late 1980s [16]. This finding is novel because it linked medical literature with chemistry literature. Thus, I want to uncover previously unrealized links, contradictions, and confirmations in the current literature on on how students utilize computers to enhance or hinder their educational experience.

Supervised learning using text has been heavily researched in the biomedical sciences. For example, [12] proposed to use a modified naïve Bayes classifier which can determine whether an abstract is relevant for a given topic, based on the words in previously seen abstracts. They also propose a unique weighting scheme which allows for high recall and reasonable precision. In their work, they show their proposed process can significantly reduce the time required to conduct a systematic literature review. Given the amount of publications available following from the aforementioned changes, these results could help educational researchers significantly reduce time to determine which previously published work is most relevant.

More broadly, this work addresses the need to have a "living systematic literature review" where the most up-to-date published findings can be included for practitioners and researchers to implement and be informed of these findings [3]. One study found the average time between a published finding and inclusion in a systematic literature review to average between 2.5 and 6.5 years [3]. This relates directly to an initiative by the US's Institute of Educational Sciences to use evidence based practices [19]; that is, connecting the knowledge from research to practicing the knowledge.

3. APPROACH

This project will extract sentences containing results from full-text journal articles in peer-reviewed journals. Given that journals have dozens of volumes and issues, it is likely not feasible to read and find all relevant articles needed to understand prior research. This process will create a systematic review of literature from educational journals in a targeted area: student interaction and behavior in computing environments. The systematic review will inform researchers on previous findings and update practitioners on the most current research.

3.1 Extracting Results

To extract result sentences, I will parse full-text journal articles into sentences, using a tokenizer, for example, Python's NLTK [2]. Next, I label the sentences as either containing a result or not, as well as indicate the section of the article where the sentence lies, and whether the sentence is the first or last in the respective paragraph, following from [7]. In [7], result sentences were distributed throughout the journal articles and were most common in the first or last sentence of the paragraph. Then, I will experiment with various classifiers, such as support vector machines, naïve Bayes classifiers, decision trees, and various ensemble models. The output of the classifiers will be the sentences containing results, which can then be used to form a thorough systematic review.

To train these models, I will select features using traditional metrics, such as information gain, mutual information, and the χ^2 statistic [20], which are the ones used by [7]. Interestingly, using these three feature selection strategies, not one term was selected by all three methods; however, there was overlap with terms for the χ^2 statistic and information gain, and information gain and mutual information. Because of this finding, I propose to use a different feature selection technique to select words or surface level knowledge (e.g., sentence position, section of paper) to train these classifiers.

3.2 Feature Selection

Another experiment I plan to conduct to extract words from the corpus of sentences from the journal articles is to utilize the LASSO to select words to use to train classifiers to discern sentences containing results from those that do not. Given that the LASSO is used for high dimensional data sets as a variable selection technique, in fields such as gene-expression analysis [5], this approach seems reasonable given the high dimensionality and sparseness of text data. I will experiment with various parameters of the LASSO to ensure reasonable feature selection; that is, a feature set which is not prohibitively small to provide high recall and reasonable precision, but one which is not too big to prohibit generalizablity.

The specific binomial logistic LASSO model I will use to select terms is

$$\log \frac{P(result = 1|\mathbf{x})}{P(result = 0|\mathbf{x})} = \beta_0 + \mathbf{x}^T \beta, \tag{1}$$

where $result$ equals one if the sentence x_i contains a result, and zero otherwise. Note that $\mathbf{x}$ is a matrix, where each row is a sentence, one column is $result$, and the other columns are words and surface-level features about the sentence. In the estimation phase, the model's likelihood function is penalized by a shrinkage parameter λ. This shrinkage parameter shrinks unimportant βs towards zero, thus leaving only the most important terms with nonzero βs. These terms will then be used to train the classifiers to extract result sentences to be used in systematic literature reviews. Further, the magnitude of each β can be beneficial in determining relative importance of a term.

For this portion of the project, I will experiment with various λs to determine which give the best performance when training the models to extract result sentences. A comparison of the feature selection strategies in [7, 20] will be conducted to determine any relationship between these feature selection strategies and the LASSO.

4. CURRENT STATUS

My current tasks are to complete a literature review of text classification. In this literature review, I address traditional classifiers from multivariate statistics and machine learning, but also accompany background on generating systematic literature reviews. The literature review also includes a discussion of evidence based practices and speculates on how a living systematic literature review might impact education research.

A concurrent stage is procuring and processing texts for analysis. In [7], seventeen full-text articles were analyzed, with around 2550 total sentences being considered. Thus, once all texts have been selected, I will begin labeling the sentences as containing a result or not containing a result. Efforts are underway to procure a small research fund to pay a research assistant to also label sentences as a measure of inter-rater reliability.

5. PROPOSED CONTRIBUTIONS

This work provides contributions to the fields of information science and educational data mining. One contribution is an alternative feature selection strategy which could improve performance of supervised learning methods. Because feature selection is arguably the most important analysis phase in text classification, using the LASSO in addition to strategies already used might help better performance in text classification.

Another contribution of the work is introducing the concept of a living systematic literature review to educational research. Due to the explosion of the amount of published research in education, and the interest in evidence based practice to be utilized in education, this work can address those desires.

6. ADVICE SOUGHT

I would like advice on any or all of these concerns:

1. Are there other approaches, besides classifiers such as support vector machines, naïve Bayes, discriminant analysis, neural networks, and decision tree classifiers that would be useful for this approach?
2. What suggestions do you have for analyzing the result sentences once they have been discovered by the classification algorithms?
3. Do you have any suggestions for experiments with the shrinkage parameter, λ, for selecting terms when using the LASSO?
4. Are there any specific metrics you would suggest to use for analyzing the results of either result extraction or selecting terms?

7. REFERENCES

[1] Academic Senate of the University of California. UC systemwide academic senate open access policy, 2013.

[2] S. Bird. NLTK: The natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

[3] J. H. Elliott, T. Turner, O. Clavisi, J. Thomas, J. P. Higgins, C. Mavergames, and R. L. Gruen. Living systematic reviews: an emerging opportunity to narrow the evidence-practice gap. *PLoS Med*, 11(2):e1001603, 2014.

[4] Elsevier, Inc. Text and data mining policy, 2014.

[5] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer, 2001.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[7] H. A. Gabb, A. Lucic, and C. Blake. A method to automatically identify the results from journal articles. *iConference 2015 Proceedings*, 2015.

[8] Hewlett Packard. Dizzying volumes of data is nothing new.

[9] Institute of Educational Sciences. IES policy regarding public access to research, 2016.

[10] Intellectual Property Office. Exceptions to copyright: Research, 2014.

[11] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[12] S. Matwin, A. Kouznetsov, D. Inkpen, O. Frunza, and P. O'Blenis. A new algorithm for reducing the workload of experts in performing systematic reviews. *Journal of the American Medical Informatics Association*, 17(4):446–453, 2010.

[13] National Institutes of Health. Revised policy on enhancing public access to archived publications resulting from NIH-funded research, 2008.

[14] National Science Foundation. NSF's public access plan: Today's data, tomorrow's discoveries (NSF 15-22), 2015.

[15] Springer. Springer's text- and data-mining policy, 2016.

[16] D. R. Swanson. Migraine and magnesium: Eleven neglected connections. *Perspectives in Biology and Medicine*, 31(4):526–557, 1988.

[17] J. Thomas, J. McNaught, and S. Ananiadou. Applications of text mining within systematic reviews. *Research Synthesis Methods*, 2(1):1–14, 2011.

[18] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[19] US Department of Education: Institute of Educational Sciences. Identifying and implementing educational practices supported by rigrous evidence: A user friendly guide, 2003.

[20] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.

Intelligent Argument Grading System for Student-produced Argument Diagrams

Linting Xue
North Carolina State
University
Raleigh, North Carolina, USA
lxue3@ncsu.edu

ABSTRACT

Current automated essay grading systems are typically focused on the semantic and syntax analysis of written arguments via Natural Language Processing techniques. Few systems focus on the automatic assessment of argument *structure*. In this work, we propose to build an Intelligent Argument Grading System to automatically assess and provide feedback on the structure of arguments of student-produced argument diagrams, which are graphical representations for real-word argumentation. The proposed system contains two stages. In the first, it automatically induces empirically-valid graph rules for expert-graded argument diagrams. An assessment model is trained from the dataset of manually-graded argument diagrams with the feature of induced graph rules. In the second stage, the assessment model automatically grades and provides feedback by identifying both good features and structural flaws in students' work. The significance of this work will be that the proposed system can save high cost of labor by automatically inducing empirically-valid rules, grading, and providing feedback on the structure of arguments for students. We anticipate that the automatic feedback can help students revise their structural plans accordingly before they start to write essays, which will in turn lead them to produce more high-quality arguments.

Keywords

Argument Diagrams, Structure of Arguments, Automated Grading System, Automatic Feedback

1. INTRODUCTION

Argumentation is an essential skill in scientific domains including physics, engineering, and computer science, where students must articulate and justify testable hypotheses through argumentative reasoning. As a consequence, automated essay grading systems have become particularly useful tools for argument assessment (e.g. [1, 3, 9]). Prior research has shown that automated assessment systems can be used to assess student-produced arguments correctly and cost-effectively. Current automated grading systems rely on either surface-level analysis of linguistic features within a bock of text (as in [3]) or deeper Natural Language Processing (NLP) that utilizes machine learning techniques (as in [9, 1]). These systems are typically designed to evaluate on the basis of readability (e.g. the number of prepositions and relative pronouns or the complexity of the sentence structure), shallow semantic analysis (e.g. lexical semantics or the relationships analysis among named entities), and syntax analysis (e.g. grammatical analysis). Ultimately, these systems return the scores or feedback on the content and the qualities of the students' writing based on a predictive model that is trained by the dataset stored in the system.

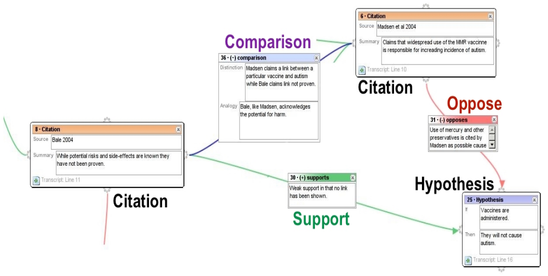
However, very few active systems are focused on automatic analysis of the rhetorical structure of arguments to address structural flaws. Argument structure refers to the organization of the key components of argumentation (e.g. hypotheses, citations, or claims), which can reveal how the students justify their research hypotheses by using relevant evidence to support or oppose conclusory statements. In real-life teaching, the students are encouraged to structure their argumentative essays before they start writing by formulating a research hypothesis based on the research question, listing relevant evidence and factual information, and identifying the logical relationships between them. Evaluating the draft structure of these arguments and identifying flaws can help students to revise their plans and to produce high-quality arguments in the future. It is possible for human experts to grade draft arguments. However that process is costly and time-consuming.

In this work, we propose to build an Intelligent Argument Grading System that can automatically grade and provide feedback on the structure of students' arguments. The system will be based upon LASAD [4], an online tool for argument diagramming and collaboration. The input to the system will be a valid argument diagram, the output is the grade and feedback pointing out the outstanding substructures and structural flaws in the student's work.

2. BACKGROUND

2.1 Argument Diagrams

Argument diagrams are visual representations of real-world argumentation that reify the essential components of arguments such as *hypotheses* statements, *claims*, and *citations* as nodes and the *supporting*, *opposing*, and *clarification* relationships as arcs [6]. These complex nodes and arcs can include text fields describing the node and arc types or free-text assertions, links to external resources and other data. Argument diagrams have been used in a variety of domains, including science [10], law[8] and philosophy [2] to help students learn written argumentation. Prior researchers have shown that argument diagrams can be used to scaffold students' understanding of existing arguments [2] and can help to support scientific reasoning [10].



6 · Citation
Source: Madsen et al 2004
Summary: Claims that widespread use of the MMR vaccinne is responsible for increading incidence of autism.
Transcript: Line 10

36 · (~) comparison
Distinction: Madsen claims a link between a particular vaccine and autism while Bale claims link not proven.
Analogy: Bale, like Madsen, acknowledges the potential for harm.

31 · (-) opposes
Use of mercury and other preservatives is cited by Madsen as possible cause

8 · Citation
Source: Bale 2004
Summary: While potential risks and side-effects are known they have not been proven.
Transcript: Line 11

30 · (+) supports
Weak support in that no link has been shown.

25 · Hypothesis
If: Vaccines are administered.
Then: They will not cause autism.
Transcript: Line 16

Figure 1: A student-produced Argument Diagram.

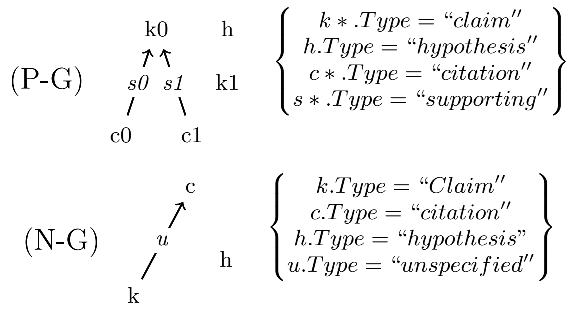
A sample student-produced diagram is shown in Figure 1. The diagram includes a *hypothesis* node at the bottom right, which contains two text fields, one for a conditional or *if* field, and the other for a consequent or *then* field. Two *citations* are connected to the *hypothesis* node via *supporting* and *opposing* arcs colored green and red, respectively. They are also connected via a *comparing* arc. Each citation contains two fields: one for the citation information and the other for a summary of the work; each arc has a single text field explaining what purpose the relationship serves.

3. PRELIMINARY RESULTS

In Lynch's study of diagnosticity of argument diagrams [5], a set of 104 paired diagrams and essays were collected at the University of Pittsburgh in a course on Psychological Research Methods. The diagrams and essays were independently graded by an experienced TA according to a parallel grading rubric. They showed that hand-authored graph rules were *empirically-valid* and were correlated with the diagram and essay grades; and thus that they could be used as the basis of predictive models for automatic grading.

Our prior work has also shown that Evolutionary Computation (EC) can be used to automatically induce empirically-valid graph rules for student-produced argument diagrams, and that the induced graph rules can be used as features for automatic grading [11, 12]. It is possible to harvest a set of diverse rules that were filtered via post-hoc Chi-Squared analysis [7]. This includes both good rules that are positively correlated with the diagram and essay grades and bad rules which are negatively correlated with the former representing positive structural features and the latter indicating flaws in the argument.

Figure 2 shows an example of a positive graph rule (P-G) and a negative graph rule (N-G) induced in our prior work. P-G shows a graph structure where the students identified at least two related citations ($c0$ & $c1$) that can be synthesized to support a single claim ($k0$) and where they included both a separate hypothesis (h) and an additional claim ($k1$).



$$\left\{\begin{array}{c} k*.Type = \text{``}claim'' \\ h.Type = \text{``}hypothesis'' \\ c*.Type = \text{``}citation'' \\ s*.Type = \text{``}supporting'' \end{array}\right\}$$

$$\left\{\begin{array}{c} k.Type = \text{``}Claim'' \\ c.Type = \text{``}citation'' \\ h.Type = \text{``}hypothesis\text{''} \\ u.Type = \text{``}unspecified'' \end{array}\right\}$$

Figure 2: Examples of positive and negative graph rule.

It shows one of the structures that students have been encouraged to incorporate into their arguments as it shows an ability to synthesize citations to form a complex claim.

N-G is a negative rule that contains a single claim node (k) which is connected to a citation node (c) via an undefined arc (u), and a separate hypothesis node (h) which may or may not be connected to the rest structure. This rule is a clear violation of the semantic guidance that students were given. In our experiment, the students were instructed to use unspecified arcs for definitions or clarifications. Some students instead used them only when they were unsure about the strength of their evidence or did not understand the citation.

4. PROPOSED SYSTEM

In this work, we propose to build an Intelligent Argument Grading System (iARG) for student-produced argument diagrams. Our goal is to automatically grade the structure of arguments for students and provide feedback that reflects the good features and structural flaws in students' work. The proposed system includes two stages, which are shown in Figure 3.

The top part of Figure 3 illustrates the first stage, **Automatic Rule Induction**, in which the system automatically induces empirically-valid graph rules for expert-graded argument diagrams. The system will contain a database of argument diagrams and expert-assigned grades, along with a database of graph rules induced by the EC algorithm with a χ-Squared filter as described in [11, 7]. After the system produces a set of individual rules, the induced rules are evaluated by domain experts to determine whether or not they are semantically valid. Only valid rules will be incorporated into the database. Note that the induced rules contain both positive and negative examples. At the end of the process, we will use supervised learning methods to train an assessment model based upon the feature of induced rules and other graph feature (e.g. the degree of diagram nodes, the complexity of diagrams, and the attribute of the hub nodes in diagrams).

In the second stage of **Automatic Grading and Feedback**, the trained model will automatically grade and provide feedback on students' submissions by identifying both good features and structural flaws of the arguments. After

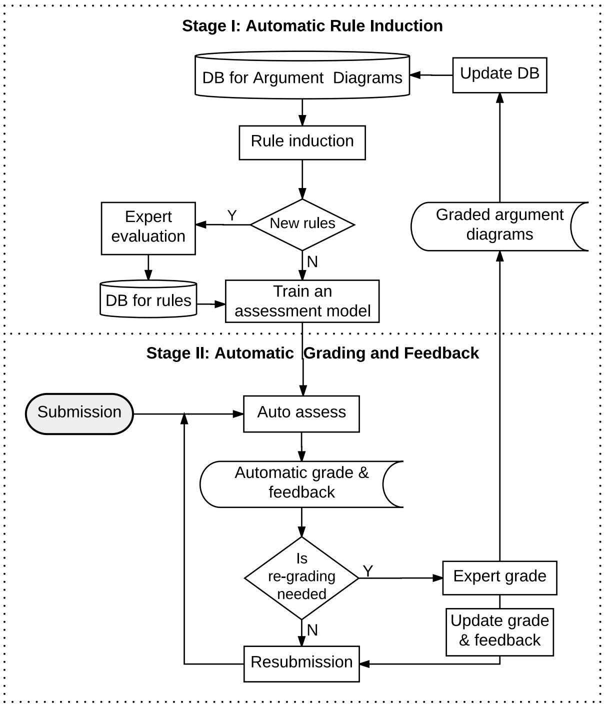


Figure 3: Flowchart for the proposed iARG

this, we will have experts re-evaluate the automatic grades and give feedback periodically, and if necessary, to re-grade the submission. We include this step because the students' submissions may include novel structures that are not included in the current rule database. In this case, the assessment model may treat these novel structures as outliers and provide uncorrected feedback. If the submissions are re-graded by experts, they will be updated to the database for argument diagrams. The rule database and assessment model will also be updated for future use.

5. FUTURE WORK & OPEN QUESTIONS

In the future work, we plan to achieve the following:

1. In Fall 2017, we plan to work with domain experts to determine whether the induced graph rules are semantically valid; whether they can be used for automatic grading; and whether they include all of the good features and structural flaws in students' work. This gives rise to our first research question: *how can we improve the performance of the graph rule induction algorithm by inducing more empirically-valid graph rules?*

2. In Spring 2018, we will leverage different supervised learning methods to train an assessment model from our current dataset of expert-graded argument diagrams with the feature of valid graph rules and other graph features. We will evaluate the assessment model on a new set of student-produced argument diagrams. Our second research question is that *what other graph features can we use to build the assessment model?*

3. In Fall 2018, we plan to implement the proposed system based upon LASAD by building databases for the argument diagrams and for the graph rules, and integrating the assessment model into the system.

4. In 2019, we will test the performance of our system in an augmentative writing class at NCSU. We will focus on accessing the automatic grades and feedback from the student's perspective and determine whether they find the automatic feedback to be useful. Thus we will not have experts to examine the automatic feedback in the second stage. Based upon the students' feedback, we will consider whether to have experts to regrade the new submission and to update the database and assessment model.

6. REFERENCES

[1] J. Burstein, C. Leacock, and R. Swartz. Automated evaluation of essays and short answers. 2001.

[2] M. Harrell and D. Wetzel. Improving first-year writing using argument diagramming. In *The 35th CogSci*, pages 2488–2493, 2013.

[3] M. A. Hearst. The debate on automated essay grading. *IEEE Intelligent Systems and their Applications*, 15(5):22–37, 2000.

[4] F. Loll and N. Pinkwart. Lasad: Flexible representations for computer-based collaborative argumentation. *International Journal of Human-Computer Studies*, 71:91–109, Januart 2013.

[5] C. F. Lynch and K. D. Ashley. Empirically valid rules for ill-defined domains. In J. Stamper and Z. Pardos, editors, *Proceedings of The* 7^{th} *International Conference on EDM*. IEDMS, 2014.

[6] C. F. Lynch, K. D. Ashley, and M. Chi. Can diagrams predict essay grades? In S. Trausan-Matu, K. E. Boyer, M. E. Crosby, and K. Panourgia, editors, *ITS*, Lecture Notes, pages 260–265. Springer, 2014.

[7] C. F. Lynch, L. Xue, and M. Chi. Evolving augmented graph grammars for argument analysis. GECCO, 2016.

[8] N. Pinkwart, K. D. Ashley, C. F. Lynch, and V. Aleven. Evaluating an intelligent tutoring system for making legal arguments with hypotheticals. *IJAIED*, 19(4):401 – 424, 2009.

[9] L. M. Rudner and T. Liang. Automated essay scoring using bayes' theorem. *The Journal of Technology, Learning and Assessment*, 1(2), 2002.

[10] D. D. Suthers. Empirical studies of the value of conceptually explicit notations in collaborative learning. In A. Okada, S. Buckingham Shum, and T. Sherborne, editors, *Knowledge Cartography*, pages 1–23. Springer Verlag, 2008.

[11] L. Xue, C. Lynch, and M. Chi. Unnatural feature engineering: Evolving augmented graph grammars for argument diagrams. In *Internatinal Educational Data Mining*, pages 255–262. IEDMS, 2016.

[12] L. Xue, C. F. Lynch, and M. Chi. Mining innovative augmented graph grammars for argument diagrams through novelty selection. EDM, 2017.