Unnatural Feature Engineering: Evolving Augmented Graph Grammars for Argument Diagrams

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

Collin F. Lynch
North Carolina State University
Raleigh, North Carolina, USA
cflynch@ncsu.edu

Min Chi
North Carolina State University
Raleigh, North Carolina, USA
mchi@ncsu.edu

ABSTRACT

Graph data such as argument diagrams has become increasingly common in EDM. Augmented Graph Grammars are a robust rule formalism for graphs. Prior research has shown that hand-authored graph grammars can be used to automatically grade student-produced argument diagrams. But hand-authored rules can be time consuming and expensive to produce, and they may not generalize well to novel contexts. We applied Evolutionary Computation to automatically induce empirically-valid graph grammars for argument diagrams that can be used for automatic grading or provide the basis for hints. Our results show that our approach can generate more relevant rules than experts or other state of the art algorithms, and that these evolved rules outperform the alternatives.

Keywords

Evolutionary Computation, Augmented Graph Grammars, Argument Diagramming, Feature Engineering

1. INTRODUCTION

Intelligent tutoring systems and computer-supported collaboration platforms have grown increasingly popular in recent years. As they have grown in popularity they have also been applied in increasingly complex domains such as argumentation [14], legal reasoning [22] and writing [6]. MOOCs and other online educational platforms have also grown in popularity yielding large repositories of user-system interaction logs [10], and classical tutors and educational games have grown more common in classrooms yielding large repositories of student data [13]. Much of this data can be represented as rich graph structures such as argument diagrams [17] or interaction networks [7].

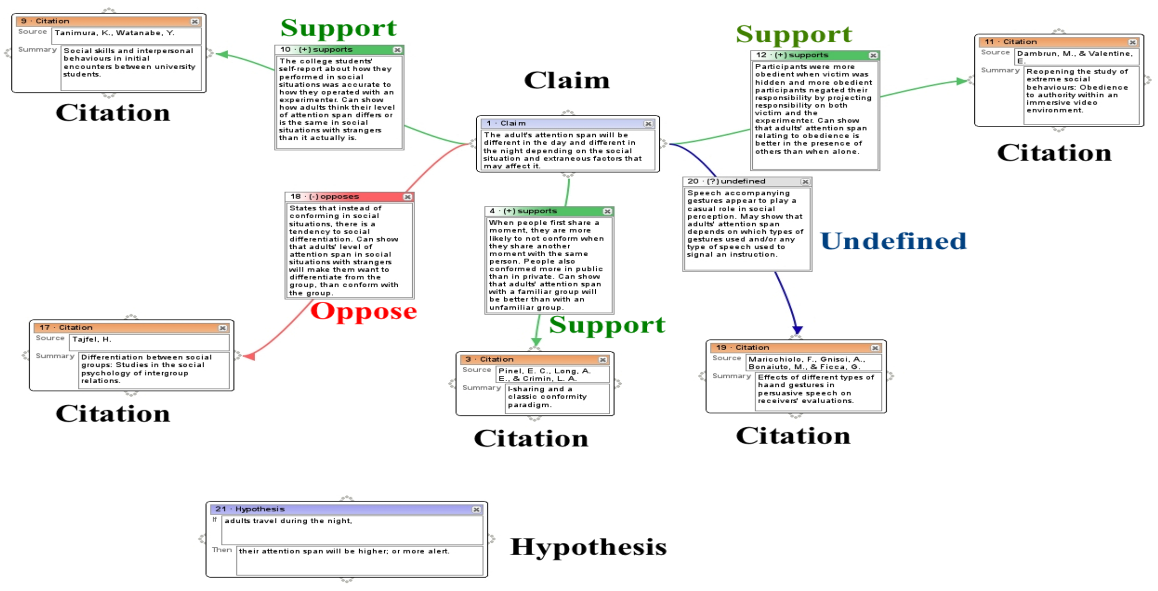
Despite the increasing prevalence of graph data, comparatively little work has been done on automatically evaluating student-produced graphs or graph logs. In prior work we demonstrated that hand-authored Graph Grammars can be used as features to automatically grade student-produced argument diagrams [16, 17]. But hand-authoring complex rules is time consuming, expensive, and does not generalize well to novel contexts. Other authors have developed analytical tools tuned to path analysis [24, 3], however these are tailored to a specific task. Other more general purpose algorithms (e.g. [30, 5]) have limitations and are unsuited to the induction of generalized rules that use negation or other complex elements. Therefore it has not yet been shown that it is possible to *automatically* induce complex, empirically-valid, rules for rich graph structures that are comparable to rules produced by domain experts.

In this paper we will describe our work on the automatic induction of Augmented Graph Grammars for student-produced argument diagrams. Our goal in this work is to explore ways to automatically induce empirically-valid graph rules that can be used as *features* for automatic grading and which can provide the basis for hints. While our previous work was focused on inducing positive rules in [33] and in [19], in this work we applied Evolutionary Computation (EC) to induce both positive and negative rules for student graphs that incorporate more complex elements such as negation and generalized types. Additionally, in our previous work we compared the induced rules with a small number of expert rules while in this work, we will compare our induced rules to a full set of complex rules authored by domain experts and rules produced by other the state of the art induction algorithms.

2. BACKGROUND

2.1 Argument Diagrams

Argument diagrams are semi-formal graphical representations that reify key features of arguments such as *hypothesis* statements, *claims*, and *citations* as nodes and the *supporting*, *opposing*, and *clarification* relationships between them as arcs. Argument diagrams directly connect the syntax of the argument representation to the underlying semantics thus making it clear and computationally tractable. Argument diagrams can serve to make the often implicit structure of an argument *salient* to students while also *constraining* them to make relevant contributions [29]. Prior researchers have shown that argument diagrams can be used to scaffold students' understanding of existing arguments [12, 8]; can frame collaborative learning [26]; and can help to support scientific reasoning [29].



9 · Citation
Source: Tanimura, K., Watanabe, Y.
Summary: Social skills and interpersonal behaviours in initial encounters between university students.

10 · (+) supports
The college students' self-report about how they performed in social situations was accurate to how they operated with an experimenter. Can show how adults think their level of attention span differs or is the same in social situations with strangers than it actually is.

1 · Claim
The adult's attention span will be different in the day and different in the night depending on the social situation and extraneous factors that may affect it.

12 · (+) supports
Participants were more obedient when victim was hidden and more obedient participants negated their responsibility by projecting responsibility on both victim and the experimenter. Can show that adults' attention span relating to obedience is better in the presence of others than when alone.

11 · Citation
Source: Dambrun, M., & Valentine, E.
Summary: Reopening the study of extreme social behaviours: Obedience to authority within an immersive video environment.

18 · (-) opposes
States that instead of conforming in social situations, there is a tendency to social differentiation. Can show that adults' level of attention span in social situations with strangers will make them want to differentiate from the group, than conform with the group.

4 · (+) supports
When people first share a moment, they are more likely to not conform when they share another moment with the same person. People also conformed more in public than in private. Can show that adults' attention span with a familiar group will be better than with an unfamiliar group.

20 · (?) undefined
Speech accompanying gestures appear to play a casual role in social perception. May show that adults' attention span depends on which types of gestures used and/or any type of speech used to signal an instruction.

17 · Citation
Source: Tajfel, H.
Summary: Differentiation between social groups: Studies in the social psychology of intergroup relations.

3 · Citation
Source: Pinel, E. C., Long, A. E., & Crimin, L. A.
Summary: I-sharing and a classic conformity paradigm.

19 · Citation
Source: Maricchiolo, F., Gnisci, A., Bonaiuto, M., & Ficca, G.
Summary: Effects of different types of haand gestures in persuasive speech on receivers' evaluations.

21 · Hypothesis
If: adults travel during the night,
Then: their attention span will be higher; or more alert.

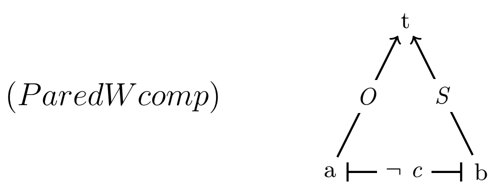
Figure 1: A student-produced Argument Diagram.

A sample student-produced diagram is shown in Figure 1. The diagram includes a central research *claim* node, which has a single text field indicating the content of the research claim. A set of citation nodes are connected to the claim node via supporting, opposing and undefined arcs colored green, red, and blue respectively. 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. At the bottom of the diagram, there is a single isolated hypothesis node that contains two text fields, one for a conditional or *IF* field, and the other for a consequence *THEN* field.

2.2 Augmented Graph Grammars

Graph Grammars are a graph-based representation for rules about graphs that are analogous to string grammars. Graph grammar rules are composed of standard graph elements such as nodes and directed or undirected arcs. As with string grammars they are defined by a finite alphabet of basic or *ground* node and arc types as well as a set of production rules for *variable* elements. A single graph *rule* defines a space or *class* of matching graphs. Graph grammars can be used to generate graphs from an initial seed via recursive rule applications where each variable element expands to a larger subgraph. They can also be used to match graphs in a layered fashion by first mapping all ground elements to individual nodes or arcs and then recursively matching the sub-elements. Graph grammars have been used for analysis and graph transformation in domains such as visual programming [9] and mechanism analysis [27].

Augmented Graph Grammars are an extension of traditional graph grammars that are allow us to match *rich graphs* with complex node and arc types that contain sub-elements, text, and other variable structures [15]. Augmented Graph Grammars also support: negated elements which select for the nonexistence of subgraphs; generalized node and arc types



$$\left\{\begin{array}{c} t.Type = \text{“}claim\text{''}or\text{“}hypothesis\text{''} \\ a.Type = \text{“}citation\text{''} \\ b.Type = \text{“}citation\text{''} \\ c.Type = \text{“}comparison\text{''} \end{array}\right\}$$

Figure 2: A simple augmented graph grammar rule that detects uncompared counterarguments.

which match multiple items; complex element constraints which allow us to compare individual elements; complex graph expressions which allow for universal and existential quantification; and the incorporation of NLP rules or other external features. As such they are an ideal rule representation for the analysis of argument diagrams, user-system interaction logs, and other educational data.

A sample rule is shown in Figure 2. This rule is designed to identify cases of uncompared counterarguments, that is: there is an opposing arc O from the citation a to the node t and also a supporting arc S from the citation b to the node t, however, there exists *no* comparison arc between the two citations a and b. This is designated by the negated arc $\neg c$. Here node t is either a claim or hypothesis. The variable elements O and S are defined by recursive production rules which are not shown. Those rules define supporting paths as chains of supporting arcs and opposing paths as chains of supporting arcs with any odd numbered (including single) chain of opposing arcs.

This example rule was designed by a domain expert in argumentation. It is designed to identify cases where a student has presented conflicting background information but has made no attempt at resolution. This is a critical structural flaw that is commonly found in student-produced arguments. Students at all levels frequently absorb the lesson that they must show conflicting citations but routinely fail to explain those citations or to resolve the differences in a way that clarifies their own argument. As we have shown previously such expert-designed rules can be empirically-valid and predictive of student performance [16]. However manually designing rules can be both costly and inefficient.

Thus our goal is to automatically induce meaningful rules, rules that highlight structural flaws or argumentation errors; rules that generalize beyond basic types; and rules that include negated elements(detecting non-existing cases).

2.3 Graph Grammar Induction

Current grammar induction algorithms fall into one of two broad categories: frequent subgraph matching, or graph compression. Frequent subgraph algorithms include Yan and Han's gSpan algorithm [32], Inkokuchi's AGM [1], and the FSG algorithm [20]. These algorithms carry out controlled graph walks to identify common structures. They are quite effective, particularly in grounded domains such as cheminformatics where the graphs, in this case molecular models, have low degree and exact matches are required. However the algorithms do not support disjoint subgraphs, negation, or generalized elements. While we can, in theory, insert explicit negation arcs that would expand the size of the graphs exponentially and thus make any search process intractable. Similarly, while we could replace individual elements with generalized forms that would simply force the system to use a smaller range of types and would not allow for context-sensitive generalization of elements. These algorithms are also ill-suited for identifying errors as the search process is strictly unsupervised and finds frequently-occurring structures without reference to external weights.

Graph compression algorithms such as Subdue take a different approach to the problem. Subdue is a recursive beam-search algorithm that generates a hierarchical grammar by recursive collapse based upon the MDL principle [5]. Subdue operates by iteratively identifying the most frequently occurring arc in the graph and then reducing it to a new variable node. Unlike gSpan the resulting grammar is hierarchical and the beam search process can be used for supervised learning given a suitable set of positive and negative examples [11]. The candidate graphs are ranked according to a normalized error metric:

$$\frac{(PosGraphsNotCovered + NegGraphsCovered)}{TotalExamples}$$

While Subdue is more flexible than the frequentist approaches it too does not support generalized elements, negation, or disjoint subgraphs.

2.4 Related Work

We have previously shown that domain experts can hand author augmented graph grammars that are empirically-valid and which can be used as features in a regression model to automatically grade student-produced diagrams [16, 17]. In more recent experiments we have also shown that it was possible to apply EC to induce graph grammars that are positively correlated with argument grades and that we can apply χ^2-filtering to select unique rules from the large space of candidates [19]. We were also able to show that the induced rules outperformed rules generated by both Subdue and gSpan and outperformed similar expert rules that fit into the limited rule space. The rules produced in that study, however, were limited in scope. While they supported disjoint graphs, they did not identify errors, and did not support generalized elements or negation. In this work we will build upon these results to include generalization and negation, and we will compare the resulting rules to a full set of 77 hand-coded expert rules.

3. METHODS

We conducted two experiments on the induction of Augmented Graph Grammars using EC. First we applied EC to induce graph rules composed of static node and arc types that were both positively and negatively correlated with the overall argument quality. That is, we sought to identify ground rules that either highlighted good features of arguments **(positive)** or matched structural flaws**(negative)**. We then compared them to expert-produced rules and to rules induced by the Subdue and gSpan algorithms. In our second experiment we applied EC to induce rules that also incorporated generalized nodes as well as negated arcs **(detecting non-existing cases)**. We describe them below.

Evolutionary Computation is a general beam-search algorithm based upon Natural Selection. The EC algorithm begins with a population of candidate solutions in a shared solution representation. This population may be randomly generated or supplied by the user. The individual solutions are then ranked by means of a fitness function which may be an absolute performance metric or a form of tournament selection. The next generation of the population is then formed by a combination of fitness proportional selection, crossover or recombination of candidate solutions, random mutation of solutions, and elitist cloning. EC algorithms proceed iteratively until a given fitness threshold is reached or a fixed number of generations has passed. EC has been used in a number of applications such as tuning Neural Networks [21], and evolving computer code [2].

EC has a number of advantages over other special-purpose induction algorithms. Firstly, it is very flexible, the behavior of the system is determined by the user-specified solution representation and the genetic operators. This makes it easy to tune the behavior of the system to include new types of elements or to test out alternative inductive biases. Secondly, EC is very robust, the basic algorithm can be applied in a wide range of domains and it can be used in areas where the contours of the search space is unknown. There are a number of widely-available EC systems. For the purposes of this research we used *pyEC* an open-source EC engine [18] coupled with *AGG* an engine for graph matching using Augmented Graph Grammars [15].

The rules induced in Experiment I consisted entirely of ground nodes and arcs while the rules induced in Experiment II included generalized node types and negated comparisons as shown in Figure 2. For both experiments we assessed the

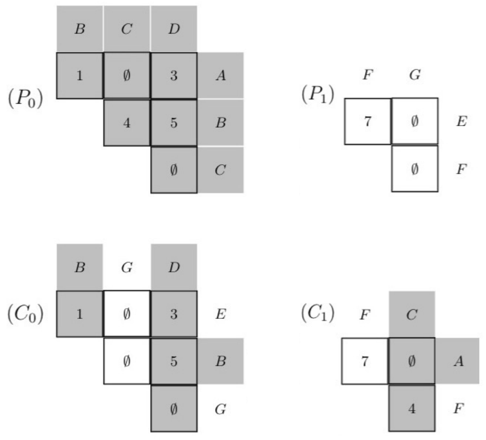


Figure 3: Canonical matrices for crossover.

fitness of the rules using the same nonparametric frequency correlation that we discussed in Subsection 2.4 with the target values being maximized or minimized depending upon the experimental goals.

Mutation in the EC algorithm is a general-purpose operation that is designed to promote exploration by introducing heterogeneity into the population. For this set of experiments we applied basic point mutation that added, deleted, or modified individual graph elements (see [33, 19]). Here mutation occurred with a small constant frequency when individuals were added to each population.

For these experiments we employed stable matrix crossover based upon the work of Stone, Pillmore, & Cyre [28] illustrated in in Figure 3. In this form of crossover we select a pair of parent graphs using fitness-proportional selection and represent them as adjacency matrices (P_0). The nodes are represented by letters on the rows and columns, while the arcs are represented by the numbered cells within the table. Empty cells indicate the absence of an arc. The order of elements in the matrices is canonical and is determined by the order in which the nodes were added to the rule.

On crossover we align the nodes and arcs in the parent matrices and then randomly shuffle the nodes and arcs between them based upon a series of coin tosses to produce the two children (C_0). Any constraints that are attached to an individual element are copied with it. Matrix crossover always produces two children that match the size of their parents with all excess elements being copied directly to the larger of the two offspring. Table 1 shows this crossover process at the graph level. By design crossover is an adaptive process that is designed to promote homogeneity and to preserve good building blocks or partial solutions called *introns* [2].

4. DATA

Our experimental analysis was based upon two previously-collected datasets. The first is a set of student-produced argument diagrams for empirical research reports. The second is a repository of hand-authored rules defined by domain experts. Both datasets were collected as part of our prior work on the diagnosticity of argument diagrams [16, 17].

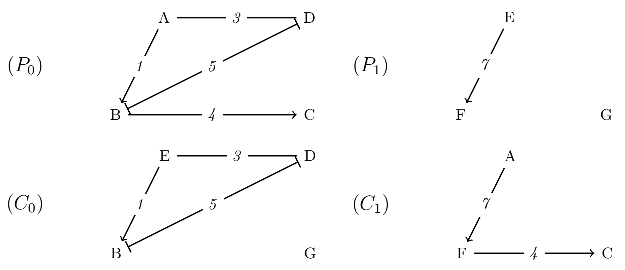


Table 1: Graphical representation for crossover.

4.1 Argument Data

Our repository of argument diagrams was collected at the University of Pittsburgh in a course on Psychological Research Methods. Students in the course learn about designing, conducting, and reporting on empirical research. The course has a significant writing component. Students complete two research projects over the course of the semester both of which result in a written report modeled on a conference publication. They are allowed to work on the projects individually or as a team of two. For the purposes of our study, the students were required to plan their written arguments graphically before writing them. The diagrams were authored using LASAD, an online tool for argument diagramming and collaboration [14]. The diagramming ontology contained four types of nodes: *citation*, *claim*, *current study* and *hypothesis*; and four types of arcs: *supporting*, *opposing*, *comparison*, and *undefined*. Currstudy nodes are used to represent factual information about the study such as the target population. Undefined arcs represent cases where nodes provide clarification or concept definitions.

After removing dropouts and one diagram containing a single node, we collected a set of 104 paired diagrams and essays from the course. These diagrams and essays were independently graded by an experienced TA according to a parallel rubric with 14 questions that were focused on the argument's quality, coherence, use of citations, and other criteria. In this work we will focus on the *gestalt* grades for overall graph and essay quality. The gestalt grades were assigned on an 11 point scale from -5 (worst quality) to +5 (complete, coherent, and persuasive) at $\frac{1}{2}$ point intervals. This same dataset was used in our prior work [19].

4.2 Expert rules

In parallel with data collection, we also collaborated with a group of domain experts to define a set of 77 *a-priori* argument rules. These rules were designed to identify individual features of argument diagrams or sub-graphs that were consistent with high quality argumentation or which represented structural flaws. Thirty-four of these rules focused on basic features such as the size or order of the diagram, the average number of parents and children, or the presence of empty elements. The remainder were complex rules that described the relationship between elements or matched larger graph structures such as the uncompared counterarguments shown in Figure 2. These rules included features that dealt with the text inside the elements, appropriate grounding of hypotheses or claims in citations, connectedness of the diagram, and the appropriate use of individual elements.

In prior work we evaluated whether or not these rules were *empirically-valid*. That is whether or not they correlated with the independently-assigned diagram grades and whether or not they could be used to predict the paired essay grades [16, 17]. In that work we assessed the validity of each individual rule by testing the correlation between the observed rule frequency on each diagram and the final graph or essay grade. The strength of this correlation was assessed using Spearman's ρ a nonparametric correlation measure [31]. We found that most, but not all of the rules were strongly correlated with the grades. We also found that some of the correlations ran counter to the experts' a-priori expectations.

5. EXPERIMENTS

In this work we induced sets of baseline rules using the Subdue and gSpan algorithms. We also conducted two sets of evolutionary experiments designated *EC-Base* and *EC-General*. The rules from each of these experiments were compared to assess their overall performance.

Subdue: For these experiments we used Subdue V5 [4] in supervised learning mode to induce rules that were positively and negatively correlated with the overall graph and essay grades. In order to induce positively correlated rules we partitioned the graphs into positive and negative examples based upon their graph or paired essay score. All graphs with a grade of 0 or more were treated as positive examples, and all graphs with a negative grade were treated as negative examples. We then ran the system to extract the 12 best rules. In order to induce negatively-correlated rules we reversed the assignment with rules that were graded less than or equal to 0 being treated as positive examples and all others being treated as negative. We experimented with more restrictive thresholds > 0 and < 0 and found the performance did not improve.

gSpan: In this experiment we used gSpan v6 [34]. The software runs in strictly unsupervised mode where it returns all subgraphs whose frequency exceeds a user-specified threshold. In this case we ran the software over our dataset and collected all rules that exceeded a 1% threshold and then ranked the candidate rules based upon their ρ value to identify the most positive and negative examples.

EC-Base: In this experiment, we conducted a series of six evolutionary runs that were tuned to induce negatively-correlated rules. Three of those runs used the graph grade as a target and three used the essay grade. In each case we used a fixed population size of 100 individuals and ran the algorithm for 1,000 generations. In each generation, we cloned the top 10 individuals directly into the next generation under elitism. We selected 10 individuals for point mutation and the remaining 80 individuals for crossover, then we copied the results over to the next generation. Fitness values were assigned using a fixed measure of $-\rho$ for each individual rule. The initial populations were composed of randomly-generated individuals containing 3 - 10 elements each. The nodes and arcs were all ground elements and were selected from a predefined ontology of basic types that matched the types used in the argument diagrams.

Unlike standard EC we did not rely solely on the final population of rules for our results. EC populations grow increasingly homogeneous over time making the final population virtual clones. In this case our goal was to induce a range of potential rules. We therefore collected candidate rules from each generation of the run by selecting every rule with a $\rho \leq -0.1$. The full set was used in our analysis.

EC-General: Here we conducted a series of twelve evolutionary runs. Six of the experiments were tailored to induce positively correlated rules while the rest were tailored to induce negatively-correlated ones. As with EC-Base the population size was 100, the algorithm ran for 1,000 generations, and we used $\pm\rho$ as the basic fitness metric and the mutation and crossover rate were the same as before. Unlike the EC-Base study these rules also included negated comparison arcs as well as two generalized node types: nodes that are citations or claims (*CitOrClaim*) and nodes that are hypotheses or claims (*HypOrClaim*). These elements were chosen for addition because they were used by the domain experts when crafting their rules. As before we collected candidate rules from the positive and negative runs with thresholds of ($\rho \geq 0.18$) and ($\rho \leq -0.1$) respectively. These thresholds were chosen based upon a series of exploratory runs in which we found that the ρ values became statistically significant after exceeding ± 0.18.

6. RESULTS & ANALYSIS

Table 2 shows the number of positively and negatively correlated rules for the Graph grades (columns 3 and 4) and the Essay grades (columns 5 and 6) that were collected during our experiments. *Total* designates the total number of rules produced by each method or in the expert set, while *Threshold* indicates the number for which $\rho \geq 0.18$ or $\rho \leq -0.18$ in the positive and negative cases respectively.

As Table 2 shows the EC approaches generated the largest number of candidate rules in both the positive and negative cases. Of the expert rules, most of them were positively correlated with performance but less than half of them exceeded the cutoff thresholds. Indeed only two of the expert rules did so for the essay grades. Both Subdue and gSpan identified positively and negatively-correlated rules but only a few of the positive rules exceeded the threshold. None of the negative rules did so.

Next, we will describe the rules induced during our EC-Base

Table 2: Number of Positive and Negative Rules

Methods		Graph		Essay	
		Pos	**Neg**	**Pos**	**Neg**
Subdue	Total	12	2	8	10
	Threshold	11	0	3	0
gSpan	Total	34	5	27	12
	Threshold	12	0	6	0
Expert	Total	56	21	46	32
	Threshold	25	6	0	2
EC-B	Total	82	256	172	160
	Threshold	82	51	172	22
EC-G	Total	394	392	652	518
	Threshold	394	193	652	30

⋆ Threshold: number of rules with $\rho \geq 0.18$ or $\rho \leq -0.18$

Table 3: Spearman correlation values for the best 3 rules in each experiment.

	Positive-correlated						**Negative-correlated**					
	Graph			**Essay**			**Graph**			**Essay**		
	1st	**2nd**	**3rd**	**1st**	**2nd**	**3rd**	**1st**	**2nd**	**3rd**	**1st**	**2nd**	**3rd**
Subdue	.276	.270	.253	.281	.215	.181	-.050	-.022	NA	-.173	-.167	-.164
gSpan	.352	.314	.272	.300	.281	.261	-.137	-.063	-.05	-.123	-.102	-.075
Expert	**.427***	.338	.329	.180	.138	.137	-.238	-.236	-.202	**-.256**	-.218	-.148
EC-B	.371	**.369**	**.362**	**.334**	**.334**	**.319**	**-.272**	**-.272**	**-.271***	-.233	**-.233**	**-.233**
EC-G	.396	**.391***	**.385***	**.357***	**.357***	**.356***	**-.273***	**-.272***	-.270	**-.269***	**-.269***	**-.269***

⋆ The best of results for Experiment I is in bold;
⋆ '*' is for best of results across both Experiment I and II.

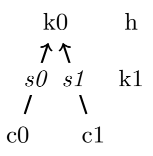
experiment and we will discuss how they compare to the expert rules and the rules induced by Subdue and gSpan. We will then discuss the EC-General rules and compare them to our earlier results.

6.1 Experiment I: EC-Base

Rows 1-4 in Table 3 list ρ values for the three best rules from the four methods. The bold values indicate the best performing rule among the sets. As the table illustrates EC-B outperformed both Subdue and gSpan across the board. And it outperformed the expert rules in most cases. The lone exception being the best positive case for the graph grades and the best negative case for the essay grades.

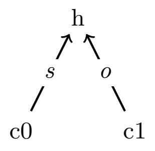
The best positively-correlated expert rule for the graph grades matched arcs with empty text fields. The best negatively-correlated expert rule with the essay grade matched graphs with no hypothesis nodes. Both of these rules relied on complex grammar features, textual rules and expressions, that were outside the scope of our current experiments.

(B-G-P)



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

(B-E-P)

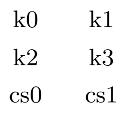


$$\left\{\begin{array}{c} h.Type = \text{“}hypothesis''\\ c*.Type = \text{“}citation''\\ s.Type = \text{“}supporting''\\ o.Type = \text{“}opposeing''\end{array}\right\}$$

Figure 4: EC-Base: Strongest Positively-correlated Rules Induced by EC.

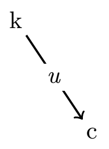
Figures 4 and 5 illustrate the best positive and negative rules induced by the EC-Base runs. In Figure 4 graph rule B-G-P represents a rule that has 5-nodes, two of which are citations ($c0$ & $c1$) that support a shared claim node ($k0$). The remaining nodes are a single claim ($k1$) and a hypothesis (h) which may or may not be connected to the rest of the structure. This reflects a graph where the authors identified at least two related citations that can be synthesized to support a single claim and where they included both a hypothesis and another claim. This is one of the structures that students have been encouraged to make in their arguments as it shows an ability to synthesize citations to form a complex claim.

(B-G-N)



$$\left\{\begin{array}{c} k*.Type = \text{“}claim\text{”}\\ cs*.Type = \text{“}currstudy\text{”}\end{array}\right\}$$

(B-E-N)



$$\left\{\begin{array}{c} k.Type = \text{“}claim''\\ c.Type = \text{“}citation''\\ u.Type = \text{“}unspecified''\end{array}\right\}$$

Figure 5: EC-Base: Stronges Negatively-correlated Rules Induced by EC.

Interestingly, the best positive essay rule (B-E-P) is very closely related to the expert rule shown in Figure 2. Here it selects for the presence of a hypothesis node (h) that is directly connected to two citations ($c0$ & $c1$). Here $c0$ directly supports h while $c1$ directly opposes it. Given that the algorithm could not induce variable arcs it is not surprising that it does not include paths. The absence of a comparison arc, however, is interesting. As we noted above the students were instructed to include one. The fact that this rule performs so well despite lacking one suggests that the students did not regularly do so.

Figure 5 shows the best negative rules. As stated above, we expect that these rules will flag errors or persistent structural flaws. B-G-N consists of 4 claim nodes ($k0 - k3$) and two currstudy nodes ($cs0$ & $cs1$) all of which may or may not be connected to one-another. While this rule has a high correlation with the grade, its semantic meaning is unclear. It is possible that it is detecting is overly large graphs that lack sufficient focus. In future work we will evaluate the matching graphs with domain experts to assess this.

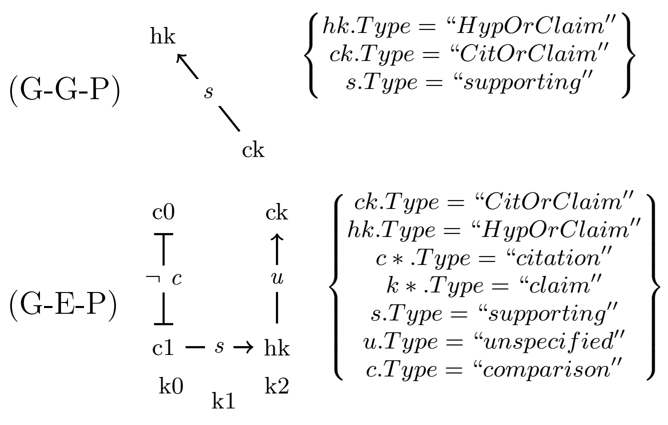
B-E-N is easier to interpret. In this case the rule contains a single claim node (k) which is connected to a citation node (c) via an undefined arc (u). This is a clear violation of the semantic guidance that students were given. The students in the experiment were instructed to use unspecified arcs for definitions or clarifications only. Some students instead used them when they were unsure about the strength of their evidence or did not understand the citation. The students were also instructed to use citations to add information to their claims, not the other way around. For a student to use an unspecified arc in this way suggests that they were unsure about the structure or content of the argument.

6.2 Experiment II: EC-General

The last row of Table 3 shows the performance of the EC-General rules. These rules were compared against all of the rules in Experiment 1. The best performing rules across both experiments are in bold and marked *. As Table 3 shows EC-General produced better performing rules than EC-Base. All but one of the ρ values on the final row exceeds the corresponding value on the fourth, and the one that does not do so falls behind by only 0.001. EC-General outperformed the best negative expert rule for the essay grades (-0.269 vs. -0.256), despite the fact that the expert rule relied on complex expressions. The best expert rule for the graph grade still outperforms EC-General. Thus, our results for EC are better than all other methods save for one expert rule that relies on novel textual features.

Figure 6 shows the best positively-correlated rules for the graph and essay grades. G-G-P matches cases where a supporting arc has been drawn from a citation or claim to a claim or hypothesis. In short, it matches correct uses of supporting arcs. This is a good feature that indicates well-supported arguments. G-E-P, by contrast, is complex and selects for a graph with three claim nodes ($k0 - k2$) and two uncompared citations ($c0$ & $c1$), where $c1$ directly supports a hypothesis or claim (hk) which in turn has an unspecified arc to a citation or claim node (ck). The semantic meaning of this rule is unclear and will require deeper analysis.

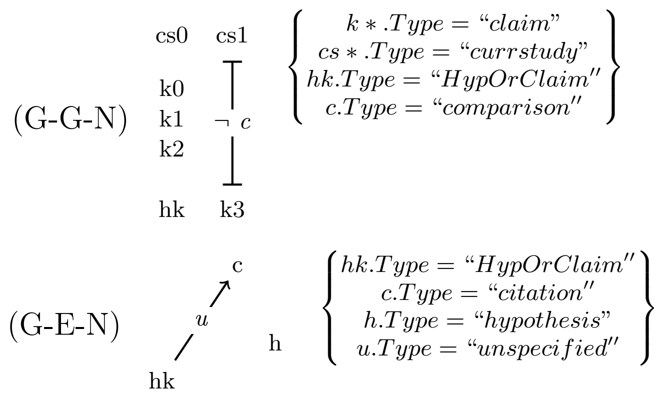
Figure 7 shows the strongest negatively-correlated rules. As with G-E-P, G-G-N, is somewhat hard to interpret. It selects for a number of disjoint nodes, and for the presence of a currstudy node ($cs0$) as well as a claim ($k3$) which are not connected via a comparison arc. Further analysis is required to determine why this rule holds. G-E-N, by contrast represents a clear variation on B-E-N. Here we select for a hypothesis or claim node (hk) that has an undefined arc to a citation along with a separate hypothesis node that may or may not be connected. This rule is interesting because in part it will select a superset of the graphs matched by B-E-N but the presence of the extra hypothesis node will restrict that somewhat. This suggests that this rule may be relatively specific to our dataset. We plan to examine the matching graphs to assess its generality.



$$\left\{\begin{array}{c} hk.Type = \text{“}HypOrClaim\text{”} \\ ck.Type = \text{“}CitOrClaim\text{”} \\ s.Type = \text{“}supporting\text{”} \end{array}\right\}$$

$$\left\{\begin{array}{c} ck.Type = \text{“}CitOrClaim\text{”} \\ hk.Type = \text{“}HypOrClaim\text{”} \\ c*.Type = \text{“}citation\text{”} \\ k*.Type = \text{“}claim\text{”} \\ s.Type = \text{“}supporting\text{”} \\ u.Type = \text{“}unspecified\text{”} \\ c.Type = \text{“}comparison\text{”} \end{array}\right\}$$

Figure 6: EC-General: Strongest Positively-correlated Rules Induced by EC.



$$\left\{\begin{array}{c} k*.Type = \text{“}claim\text{”} \\ cs*.Type = \text{“}currstudy\text{”} \\ hk.Type = \text{“}HypOrClaim\text{”} \\ c.Type = \text{“}comparison\text{”} \end{array}\right\}$$

$$\left\{\begin{array}{c} hk.Type = \text{“}HypOrClaim\text{”} \\ c.Type = \text{“}citation\text{”} \\ h.Type = \text{“}hypothesis\text{”} \\ u.Type = \text{“}unspecified\text{”} \end{array}\right\}$$

Figure 7: EC-General: Strongest Negatively-correlated Rules Induced by EC.

7. CONCLUSIONS

In this paper, we reported our work on the automatic induction of Augmented Graph Grammars for student-produced argument diagrams through EC. In prior work we demonstrated that hand-authored expert rules can be empirically-valid and that those valid rules can be used for automatic grading. We have now shown that it is possible to automatically induce complex rules for argument diagrams that match both positive and negative examples and which can therefore be used as features for automatic grading. We have also shown that the induced rules outperform all but one of the expert rules and the rules induced by other general-purpose grammar induction algorithms. The strongest expert rule was outside the scope of this experiment.

In future work we plan to work with domain experts to evaluate these rules. Our goal will be to determine whether the rules are semantically valid, and whether or not they can serve as the basis for automatic hints. We will also assess whether or not the rules can be used for data-driven grading by using them as features in a regression model. And finally we will expand the scope of our EC induction to include the automatic induction of hierarchical rules with expressions and complex element constraints.

8. REFERENCES

[1] I. Akihiro, W. Takashi, and M. Hiroshi. An apriori-based algorithm for mining frequent substructures from graph data. In *Principles of Data Mining and Knowledge Discovery*, pages 13–23. Springer, 2000.

[2] W. Banzhaf. *Genetic programming: an introduction on the automatic evolution of computer programs and its applications.* ACM, 1998.

[3] N. Belacel, G. Durand, and F. LaPlante. A binary integer programming model for global optimization of learning path discovery. In Santos and Santos [25].

[4] D. Cook, L. Holder, J. Coble, S. Djoko, B. Eberle, G. Gelal, J. Gonzalez, I. Jonyer, and N. Ketkar. Subdue version5, 2012.

[5] D. J. Cook, L. B. Holder, and N. Ketkar. Unsupervised and supervised pattern learning in graph data. In *Mining Graph Data*, chapter 7, pages 159–180. John Wiley & Sons, Inc, Hoboken, New Jersey, 2007.

[6] J. Dai., R. B. Raine., R. Roscoe., Z. Cai., and D. S. McNamara. The writing-pal tutoring system: Development and design. *Journal of Engineering and Computer Innovations*, 2:1–11, 2010.

[7] M. Eagle, A. Hicks, B. W. P. III, and T. Barnes. Exploring networks of problem-solving interactions. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge, LAK*, pages 21–30. ACM, 2015.

[8] M. W. Easterday, J. S. Kanarek, and M. Harrell. Design requirements of argument mapping software for teaching deliberation. In *Online Deliberation: Design,Research, and Practice. Stanford*, pages 317–323. CA: CSLI Publications/University of Chicago Press, 2009.

[9] K. A. et al. Graph grammar induction on structural data for visual programming. In *Applications of Graph Transformations with Industrial Relevance*, volume 7233, pages 121–136, 2012.

[10] R. S. B. et al. A MOOC on Educational Data Mining. In S. ElAtia, O. R. Zaiane, and D. Ipperciel, editors, *Handbook of Data Mining and Learning Analytics.* Hoboken, NJ: Wiley, 2016. (in press).

[11] J. A. Gonzalez, L. B. Holder, and D. J. Cook, editors. *Graph-Based Concept Learning*, Florida,USA, 2001. AAAI.

[12] M. Harrell and D. Wetzel. Improving first-year writing using argument diagramming. In *Proceedings of the 35th Annual Conference of the Cognitive Science Society*, pages 2488–2493, 2013.

[13] K. R. Koedinger, J. C. Stamper, B. Leber, and A. Skogsholm. LearnLab's DataShop: A data repository and analytics tool set for cognitive science. *Topics in Cognitive Science*, 2013.

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

[15] C. F. Lynch. Agg: Augmented graph grammars for complex heterogeneous data. In Santos and Santos [25].

[16] 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 2014*. IEDMS, 2014.

[17] 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, *Intelligent Tutoring Systems*, Lecture Notes in Computer Science, pages 260–265. Springer, 2014.

[18] C. F. Lynch, K. D. Ashley, N. Pinkwart, and V. Aleven. Argument graph classification with genetic programming and c4.5. pages 137–146.

[19] C. F. Lynch, L. Xue, and M. Chi. Evolving augmented graph grammars for argument analysis. Genetic and Evolutionary Computation Conference, 2016.

[20] K. Michihiro and K. George. Frequent subgraph discovery. In *Proceedings IEEE International Conference on Data Mining. (ICDM 2001)*, pages 313–320. IEEE, 2001.

[21] M. Mitchell. *An Introduction to Genetic Algorithms.* MIT Press: Cambridge Massachusetts, 1999.

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

[23] K. Porayska-Pomsta and K. Verbert, editors. *Workshops Proc.* 8^{th} *International Conference on Educational Data Mining, EDM 2015*, volume 1446 of *CEUR Workshop Proceedings.* CEUR-WS.org, 2015.

[24] A. Poulovassilis, S. G. Santos, and M. Mavrikis. Graph-based modelling of students' interaction data from exploratory learning environments. In Porayska-Pomsta and Verbert [23].

[25] S. G. Santos and O. C. Santos, editors. *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*, volume 1183 of *CEUR Workshop Proceedings.* CEUR-WS.org, 2014.

[26] O. Scheuer, B. McLaren, F. Loll, and N. Pinkwart. Automated analysis and feedback techniques to support argumentation: A survey. In N. Pinkwart and B. M. McLaren, editors, *Educational Technologies for Teaching Argumentation Skills.* Bentham Science Publishers, 2012. (in press).

[27] L. C. Schmidt, H. Shetty, , and S. C. Chas. A graph grammar approach for structure synthesis of mechanisms. *Journal of Mechanical Design*, 122:371–376, 1999.

[28] S. Stone, B. Pillmore, and W. Cyre. Crossover and mutation in genetic algorithms using graph-encoded chromosomes. *Unpublished*, March 2011.

[29] 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.

[30] G. Wang, Y. Han, Z. Zhang, and S. Zhang. A dataflow-pattern-based recommendation framework for data service mashup. In *Services Computing, IEEE Transactions*, volume 8, pages 889 –902. IEEE, 2015.

[31] Wikipedia. Spearman's rank correlation coefficient — wikipedia, the free encyclopedia, 2013.

[32] Y. Xifeng and H. Jiawei. gspan: Graph-based substructure pattern mining. In *Proceedings of the IEEE International Conference on Data Mining (ICDM 2002)*, pages 721–724. IEEE, 2002.

[33] L. Xue, C. F. Lynch, and M. Chi. Graph grammar induction via evolutionary computation. In Porayska-Pomsta and Verbert [23].

[34] X. Yan. gspan: Frequent graph mining package, 2009.