DOI 10.3233/JAI-2010-0001
IOS Press

Supporting Collaborative Learning and E-Discussions Using Artificial Intelligence Techniques

Bruce M. McLaren, *Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Saarbrücken, Germany and Human-Computer Interaction Institute, Carnegie Mellon Univ., Pittsburgh, PA, USA*
bmclaren@cs.cmu.edu

Oliver Scheuer, Jan Mikšátko, *Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Saarbrücken, Germany*
oliver.scheuer@dfki.de, honza.miksatko@dfki.de

Abstract. An emerging trend in classrooms is the use of networked visual argumentation tools that allow students to discuss, debate, and argue with one another in a synchronous fashion about topics presented by a teacher. These tools are aimed at teaching students how to discuss and argue, important skills not often taught in traditional classrooms. But how do teachers support students during these e-discussions, which happen at a rapid pace, with possibly many groups of students working simultaneously? Our approach is to pinpoint and summarize important aspects of the discussions (e.g., Are students staying on topic? Are students making reasoned claims and arguments that respond to the claims and arguments of their peers?) and alert the teachers who are moderating the discussions. The key research question raised in this work: Is it possible to automate the identification of salient contributions and patterns in student e-discussions? We present the systematic approach we have taken, based on artificial intelligence (AI) techniques and empirical evaluation, to grapple with this question. Our approach started with the generation of machine-learned classifiers of individual e-discussion contributions, moved to the creation of machine-learned classifiers of pairs of contributions, and, finally, led to the development of a novel AI-based graph-matching algorithm that classifies arbitrarily sized clusters of contributions. At each of these levels, we have run systematic empirical evaluations of the resultant classifiers using actual classroom data. Our evaluations have uncovered satisfactory or better results for many of the classifiers and have eliminated others. This work contributes to the fields of computer-supported collaborative learning and artificial intelligence in education by introducing sophisticated and empirically evaluated automated analysis techniques that combine structural, textual, and temporal data.

Keywords. Collaborative learning, artificial intelligence, machine learning, shallow text processing

INTRODUCTION

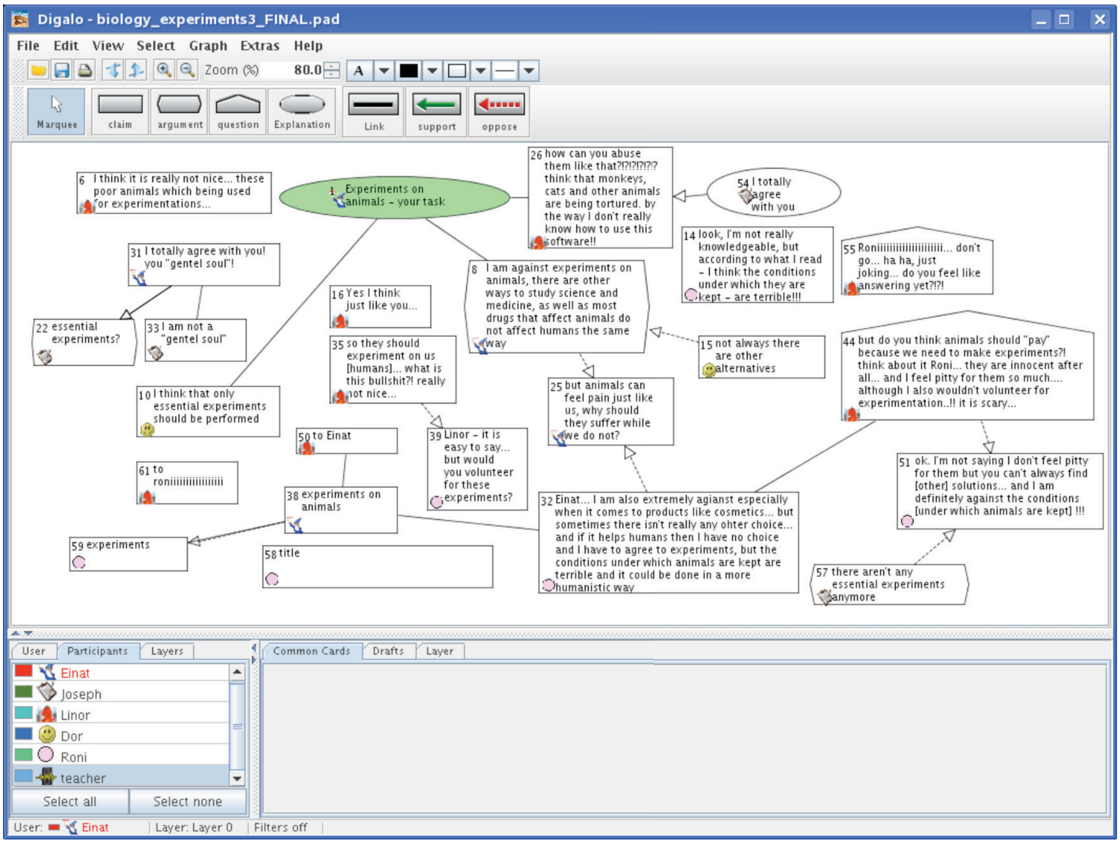
In recent years, many software tools and techniques have been developed to support and help students in learning to argue (Scheuer, Loll, Pinkwart, & McLaren, 2010). In fact, it is becoming increasingly common for students to use computer-based tools to discuss, debate, and argue with one another in a synchronous fashion about topics presented in a classroom (Lingnau, Harrer, Kuhn, & Hoppe, 2007; Schwarz & De Groot, 2007). The purpose of such tools is to help students learn to discuss and argue in a rational, well-reasoned, and considerate fashion (Andriessen & Schwarz, 2009). Such collaborative

ISSN 1560-4292/10/$27.50 © 2010 – IOS Press and the authors. All rights reserved.

software tools are designed to allow students to work on separate computers but communicate in synchronous fashion, contributing to an evolving discussion through a shared "workspace" (cf. Pinkwart, 2003).

In Israel, the Netherlands, and the U.K., for instance, we are working with and have collected data from more than 15 classrooms that are using the tools Digalo (Schwarz & Glassner, 2007) and FreeStyler (Hoppe & Gaßner, 2002; http://www.collide.info/index.php/FreeStyler/) to engage students in e-discussions.

These tools have a common model: a shared graphical environment with drag-and-drop widgets that allows students on different computers to express their ideas, claims, questions, and arguments in a visual fashion. Students make contributions by dragging and dropping shapes with different semantics (e.g., a "claim" or an "argument"), filling them with text containing their contributions to the discussion (e.g., "I don't agree with John's claim, because …"), and linking the shapes to the contribution of other students with labeled links, such as "opposes" or "supports." An example of such an e-discussion, one that was created using the Digalo collaboration software, is shown in Figure 1.



1 Experiments on animals – your task

6 I think it is really not nice… these poor animals which being used for experimentations…

26 how can you abuse them like that?!?!?!?!?!? think that monkeys, cats and other animals are being tortured. by the way I don't really know how to use this software!!

54 I totally agree with you

31 I totally agree with you! you "gentel soul"!

14 look, I'm not really knowledgeable, but according to what I read – I think the conditions under which they are kept – are terrible!!!

55 Roniiiiiiiiiiiiiiiiiiii… don't go… ha ha, just joking… do you feel like answering yet?!?!

8 I am against experiments on animals, there are other ways to study science and medicine, as well as most drugs that affect animals do not affect humans the same way

16 Yes I think just like you…

22 essential experiments?

33 I am not a "gentel soul"

35 so they should experiment on us [humans]… what is this bullshit?! really not nice…

15 not always there are other alternatives

44 but do you think animals should "pay" because we need to make experiments?! think about it Roni… they are innocent after all… and I feel pitty for them so much…. although I also wouldn't volunteer for experimentation..!! it is scary…

10 I think that only essential experiments should be performed

25 but animals can feel pain just like us, why should they suffer while we do not?

50 to Einat

39 Linor – it is easy to say… but would you volunteer for these experiments?

51 ok. I'm not saying I don't feel pitty for them but you can't always find [other] solutions… and I am definitely against the conditions [under which animals are kept] !!!

61 to roniiiiiiiiiiiiiiiiii

38 experiments on animals

32 Einat… I am also extremely agianst especially when it comes to products like cosmetics… but sometimes there isn't really any ohter choice… and if it helps humans then I have no choice and I have to agree to experiments, but the conditions under which animals are kept are terrible and it could be done in a more humanistic way

59 experiments

58 title

57 there aren't any essential experiments anymore

Fig. 1. An e-discussion in Digalo, a tool for supporting online collaboration.

However, simply providing students with collaborative computer-based tools will not necessarily lead to fruitful discussion and collaboration. Evidence from the computer-supported collaborative learning (CSCL) literature (Dillenbourg, Baker, Blaye, & O'Malley, 1995), as well as from educational psychology (Cohen, 1994; Salomon & Globerson, 1989), suggests that fruitful collaboration does not occur spontaneously. One approach to address this that has been explored by CSCL researchers is the notion of "collaboration scripts," instructions for groups of learners on what activities to execute, when they need to be executed, and by whom they need to be executed in order to foster knowledge acquisition and learning (O'Donnell, 1999; Kollar, Fischer, & Hesse, 2003; Weinberger, Ertl, Fischer, & Mandl, 2005; Diziol, Rummel, Spada, & McLaren, 2007). Another approach is to provide a software agent that can coach and/or tutor the collaborating students (Constantino-Gonzalez & Suthers, 2002; Walker, McLaren, Rummel, & Koedinger, 2007). A third approach is to include an artificial student in the collaboration whose responsibility it is to provide student-like contributions and, at the same time, peer coaching (Vizcaíno, 2005).

Yet another approach, the one we have taken on the ARGUNAUT project, is to provide feedback to a teacher so that he or she can help students stay on topic, elicit contributions from all members of the groups, suggest the use of supported claims and arguments, and generally guide the students toward fruitful discussion and collaboration. Such an approach has more generally been referred to as *e-moderation* (Coghlan, 2001), with a variety of principles proposed on how best to achieve its aims (Salmon, 2004). The burden of moderating multiple, *synchronous* e-discussions – that is, simultaneously supporting multiple groups of students collaborating in real time – is especially onerous and has been the focus of our project (Hever et al., 2007; De Groot et al., 2007). Preliminary research indicated that teachers, overwhelmed with e-moderating multiple synchronous discussions (Gil, Schwarz, & Asterhan, 2007), need technology to ameliorate this situation. In particular, our aim has been to use technology to *summarize* what is occurring in student e-discussions and *alert* teachers to critical aspects and events of those discussions.

To support teachers in the difficult task of e-moderation, the ARGUNAUT system provides online, automated feedback regarding important aspects and characteristics of each discussion, explicitly focusing attention on events or situations that may require the teacher's intervention or support. The key idea is to analyze student contributions and e-discussions using machine learning (Witten & Frank, 2005; Han & Kamber, 2006), shallow text processing (Rosé et al., 2008), and case-based graph matching (McLaren, 2003). The automated analysis is used to alert teachers to important patterns in the e-discussions as they grapple with simultaneous, synchronous e-discussions. The underlying approach of ARGUNAUT leverages the structure of the argument graphs, the textual contributions of the students, and the temporal sequence of those contributions.

The primary research question we raise and explore in this paper involves whether automated analysis techniques can be created to identify critical aspects of e-discussions.

> **Research Question 1:** *Is it possible to automate the identification of salient contributions and patterns in student e-discussions?*

This question is still open within the educational technology research community; to our knowledge there have been no successful approaches to analyzing graphical e-discussions in an automated fashion. Furthermore, the use of such automated analysis to support teacher *e-moderation* is also novel. This paper thoroughly explores the first research question, presenting our research methodology and results in attempting to answer this question.

A second research question, one that is the ultimate goal of this work but that necessarily depends on an affirmative answer to Research Question 1, revolves around whether automated detectors can effectively support teachers in e-moderating classroom discussions, that is,

Research Question 2: *Do automatically identified contributions and patterns in student e-discussions help teachers in e-moderating simultaneous, synchronous e-discussions?*

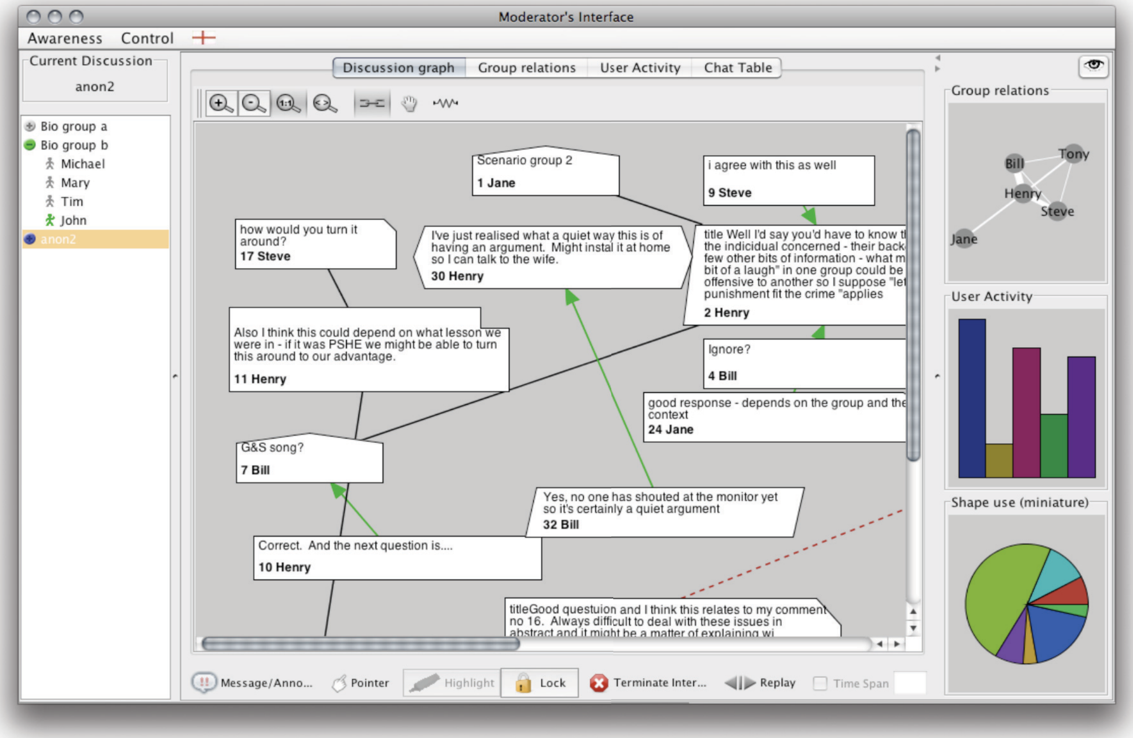
Our work in addressing the second question is in its beginning stages, due to the necessity of (a) first needing to answer Research Question 1 and (b) requiring a fully-functioning and empirically-validated analysis system to conduct studies. Nevertheless, as reported later, in the "Discussion" section, there have been two published studies (Wichmann, Giemza, Krauß, & Hoppe, 2009; Schwarz & Asterhan, in press) that have at least preliminarily shown the benefits of ARGUNAUT's e-moderation approach (i.e., use of its most basic tools, not including the automated analysis), as well as a small study, discussed in this paper, that explores a teacher's use of ARGUNAUT's automated analysis tools. Thus, the paper provides at least a glimpse of an answer to the second research question, with initial evidence indicating that teachers can gainfully use ARGUNAUT, including its automated analysis tools, to e-moderate synchronous classroom discussions.

The paper is organized as follows. First, we provide an overview of the ARGUNAUT system and its pedagogical approach. Second, we present and discuss the empirical results we have obtained in attempting to address the first research question above. Third, we discuss how our approach has succeeded, as well as come up short, in answering the first research question and also present some preliminary results in answering the second research question. Fourth, we compare and contrast our approach and research to other attempts at doing automated analysis of e-discussions and discussion threads. Finally, we conclude with a summary of our research and discuss future directions.

OVERVIEW OF THE ARGUNAUT SYSTEM

As discussed above, students using the ARGUNAUT system discuss and debate questions within a shared workspace on different networked computers in a synchronous fashion. Such a general approach to learning argumentation was pioneered in the *Belvedere* system (Suthers et al., 2001), a multi-user, graph-based diagramming tool for scientific argumentation, and has since been further developed and evaluated by a variety of other researchers (for a review of Belvedere, and other approaches to the support of argumentation with software tools, see Scheuer et al., 2010).

While students using ARGUNAUT are typically situated in a single classroom, the members of each group are distributed spatially, so that the primary communication medium between collaborators is the computer (and they are instructed to communicate that way by the teacher). Each discussion starts with a shape containing the question to be discussed, created by the teacher. The questions raised by the teacher vary, but they usually involve controversial topics such as experiments on animals ("What is your opinion about experiments on animals?") or abortion ("Should Riki and Yosi abort the fetus?"). Topics such as these allow students to take different positions and promote a lively discussion. The students are primed for their debate by, for example, reading some relevant texts or listening to a lecture on a relevant topic and are asked to provide, for instance, at least 3 reasons to support the positions they take. As explained above, students contribute by adding shapes, entering text into the shapes and connecting them by links. Shapes and links are not just simple text boxes and connectors; they have types and comprise a visual language. The on-going e-discussion can take multiple paths, and the entire discussion is captured in the shared workspace for students to review and reflect upon.



Scenario group 2
1 Jane

i agree with this as well
9 Steve

how would you turn it around?
17 Steve

I've just realised what a quiet way this is of having an argument. Might instal it at home so I can talk to the wife.
30 Henry

title Well I'd say you'd have to know t the indicidual concerned - their back few other bits of information - what m bit of a laugh" in one group could be offensive to another so I suppose "le punishment fit the crime "applies
2 Henry

Also I think this could depend on what lesson we were in - if it was PSHE we might be able to turn this around to our advantage.
11 Henry

Ignore?
4 Bill

good response - depends on the group and the context
24 Jane

G&S song?
7 Bill

Yes, no one has shouted at the monitor yet so it's certainly a quiet argument
32 Bill

Correct. And the next question is....
10 Henry

titleGood questuion and I think this relates to my comment no 16. Always difficult to deal with these issues in abstract and it might be a matter of explaining wi

Fig. 2. ARGUNAUT's Moderator's Interface and some of its shallow alerts.

A teacher can monitor multiple ongoing discussions in parallel using a tool called the "Moderator's Interface," shown in Figure 2. The teacher can toggle between the different e-discussions by selecting the different groups shown in the list on the left of Figure 2 (i.e., Bio group a, Bio group b, etc.). Within each discussion, important aspects are visualized as *awareness displays*. These displays are presumed to be helpful to a teacher as he or she tries to find pedagogically meaningful aspects of the discussion (e.g., critical thinking, dialogism). In Figure 2, on the right, three awareness displays are shown. The graph in the upper right shows the frequency with which students in the currently selected discussion have responded to one another's contributions, as well as which students are most in the center of the discussion through frequent responses to others' contributions. The middle right graph shows a comparison of the number of contributions made by each student, a rough indication of student engagement, while the graph in the lower right shows a comparison of the types of contributions made by students, a rough indication of *how* students are engaging with one another (e.g., Are the students asking one another questions? Making arguments?).

"Alerts"[1] are a special, more focused type of Awareness Display that are visualized as colored circles next to the object to which they refer (e.g., students, contributions). "Shallow alerts" are those that can be computed in a straightforward fashion (e.g., alerts that indicate inactive students or the use

[1] Note that an earlier conference paper (McLaren et al., 2007) refers to "alerts" as "awareness indicators." We use "alerts" in this paper, since this term more aptly describes the *end-user* purpose of the technology and is favored by pedagogical experts. The term "awareness indicators," on the other hand, is a more technical term indicating the core analysis technology.

of profanity in contributions). "Deep alerts," the primary focus of the current paper, are those that are generated using more sophisticated artificial intelligence-based techniques. There are three types of deep alerts in the ARGUNAUT system:

- *Shape-level alerts* reflect characteristics of individual contributions (e.g. whether a particular contribution contains a "reasoned claim");
- *Paired-shape alerts* reflect characteristics of pairs of linked contributions (e.g., whether two shapes constitute a "contribution followed by counterargument"); and
- *Cluster alerts* reflect characteristics of arbitrary sets of two or more (although typically not more than five) linked contributions (e.g., a sequence of shapes that constitute a "chain of opposition" in which two students argue back and forth).

A total of eleven deep alerts have been created, evaluated, and incorporated into the current ARGUNAUT system. These alerts will be described in further detail in the following sections; however, an example of one such alert is shown in Figure 3. In this figure, a snippet taken from the Moderator's Interface indicates the top five *Chain of Opposition* cluster alerts in a particular e-discussion. The *Chain of Opposition* cluster is three or more shapes in length and involves, typically, two students arguing back and forth, each opposing the other's argument. Since the deep alerts are based on AI heuristic techniques, their results are, by definition, approximate. Thus, the *Chain of Opposition* clusters found by ARGUNAUT are ordered from highest to lowest probability. This is indicated visually through the varying sizes of the small circles shown in the geometric center of each of the identified clusters;[2] the larger the circle the higher the probability. In Figure 3, the teacher has selected one of the *Chain of Opposition* clusters for closer inspection; additional information regarding that cluster is displayed in the "cloud" pop-up.

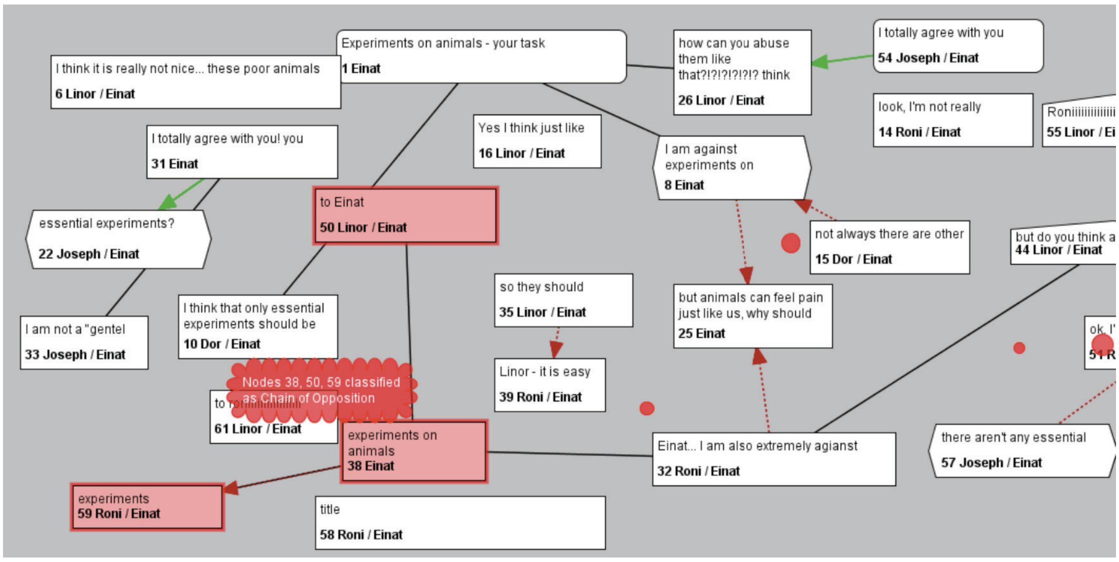


Fig. 3. A zoomed-in view of ARGUNAUT's Moderator's Interface displaying a selected e-discussion and the deep alert *Chain of Opposition* within the discussion.

[2] The circles indicating deep alerts are colored red, so they are highly visible to the teacher, but they are of course not as visible in a black-and-white figure.

The general architecture of ARGUNAUT and its most important processes are shown in Figure 4. The online part of the process is shown on the left side of the figure, while the offline classifier development process is shown in the right side of the figure. The online, end user environment is the Digalo or FreeStyler tool used by students as they collaborate with their fellow students in small groups. The Moderator's Interface is depicted just to the right of the end user environment and encompasses the Shallow Loop, Alerts, an Annotation Tool, the Classifier Proxy and other functionality. The Protocol Processor, shown below the Moderator's Interface, logs all actions taken by the students, such as creating a new shape, link, or textual contribution. The right side of Figure 4 depicts the Deep Loop, the component of ARGUNAUT that provides the deep alerts to the Moderator's Interface and, ultimately, to the teacher.

While a teacher uses the Moderator's Interface to monitor the simultaneous, synchronous[3] e-discussions (the logged representation of these e-discussions are called "discussion maps"), two dedicated components are employed to analyze the discussion maps online and provide the teacher with alerts. The *Shallow Loop* is an integrated component of the Moderator's Interface, providing shallow alerts in a local fashion. The *Deep Loop* is a Web Service-based independent component that provides deep alerts. The Classifier Proxy provides deep alerts as a local proxy for the remote Classification Service (in the middle of Figure 4, between the online and offline processes), which offers a set of classifiers each of which is capable of computing one deep alert. Classifiers are developed by an AI engineer using the Classifier Development Environment (shown on the right side of Figure 4) in an offline process and are then deployed to the Classification Service for online, run-time use. The AI-based Classifier Development Environment unfolds as two sub-components: The Machine Learning Training System is used to "train" classifiers targeted at shapes and paired-shapes using many annotated examples, in a supervised learning fashion (Witten & Frank, 2005; Han & Kamber, 2006). The Case-Based Graph Matching System is used to develop classifiers targeted at clusters using a small set of cluster examples. It is possible for teachers (and researchers) to add to the body of examples of both components through the Annotation Service shown on the border between the online and offline systems (since it can be used in both online and offline fashion), which sends new annotations to an Annotation Database. The Annotation Service is aimed at providing new examples for already existing classifiers to improve their performance. The Classification Service and Annotation Service constitute the interface between the online and offline process and are collectively called the "Classifier Web Service."

The offline process of developing new classifiers, shown on the right side of Figure 4, takes annotated discussions, translates them to a form suitable for either machine learning or the case-based graph matching algorithm (different translators are used for the different approaches), and, with the intervention of an AI Engineer, generates new classifiers that are subsequently accessed at run time to first classify actions of the students in real time and then alert teachers (i.e., the "deep alerts").[4]

[3] There is nothing inherent in the tools to preclude their use in an asynchronous fashion. However, the pedagogical approach discussed in this paper is focused on synchronous, in-classroom discussion. See (McAlister, Ravenscroft, & Scanlon, 2004; Schwarz & Asterhan, in press) for a discussion of and relative advantages of synchronous versus asynchronous communication.

[4] There are numerous other functions of the Moderator's Interface that are not discussed in this paper. For more details see Harrer, Ziebarth, Giemza, and Hoppe (2008) and Wichmann et al. (2009).

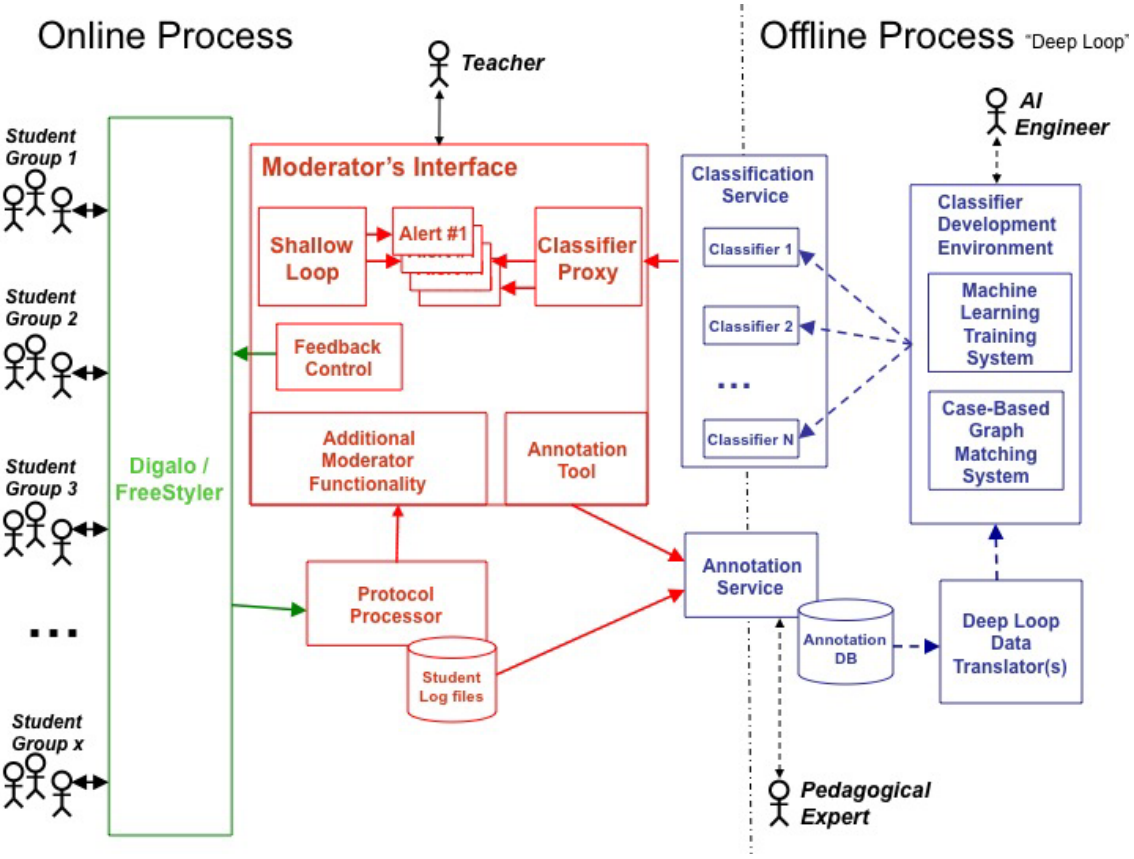


Fig. 4. ARGUNAUT's architecture for automated analysis. Solid arrows denote online processes; the dashed arrows indicate offline processes.

Pedagogical experts[5] on the ARGUNAUT project have annotated (offline) components of many past discussions as to whether, for instance, students made reasoned claims, were on topic, or were engaged in raising and answering questions. This corresponds to the process shown at the bottom right of Figure 4. These annotations were used to train (or provide examples for) the current live set of ARGUNAUT classifiers to identify meaningful types of contributions (or groups of contributions) in new e-discussions. The annotations and subsequent classifications are based on structural, textual, and temporal elements of the student contributions. The specific way that these elements are used to generate the classifiers is explained in detail in the following section.

[5] The term "pedagogical experts" is used throughout the paper to describe the members of the ARGUNAUT project team who come from education or educational psychology backgrounds and who did the corpus coding. These individuals include (but are not limited to) Rakheli Hever and Julia Gil of the Hebrew University of Jerusalem (Israel) and Maarten De Laat of Exeter University (U.K.).

CAN E-DISCUSSIONS BE AUTOMATICALLY ANALYZED?

In this section we describe the work we have done to try to answer our primary research question: Is it possible to automate the identification of salient contributions and patterns in student e-discussions? First, we describe our methodology and then we turn to a discussion of how we conducted empirical studies at three levels of e-discussions: the shape level, the paired-shape level, and the cluster level.

A key to our research approach was to methodically evaluate the different levels of the discussion maps, starting with single contributions, moving to pairs of linked contributions, and, finally, focusing on clusters (i.e., two or more connected contributions). We took this approach for several reasons. First, starting with (and solving) the simplest constructs within discussion maps, before moving on to more complex structures, seemed to increase our chances of creating at least some usable classifiers. That is, at the outset of our work we had a high degree of belief that we could create classifiers that would work at the single contribution level (due to the prior success of, for instance, Rosé et al., 2008), but were less sure at the paired level, and even less so at the cluster level. Second, we believed that the classification results of the lower levels could possibly be used as *attributes* in the solutions at the more complex levels. That is, we believed the classification of single contributions might be usable as inputs to or attributes of the classification process of the paired contributions entailing those single contributions (and subsequently to clusters, as well). Finally, we realized that the simpler structures would be easier for our pedagogical experts to annotate, meaning we would much sooner have data to evaluate and experiment with by starting with the simpler structures.

Another key to our approach was deciding on the techniques that would most likely lead to success at each of the levels of investigation. Across all levels, we knew that we needed language-processing techniques, given that a central feature of each student's contribution is the text that he or she types. Some of the issues inherent in language analysis, well known to the natural language community, include references that participants make to other's contributions, use of prior knowledge brought to bear by participants, and implications that span across different contributions. Due to the success that Carolyn Rosé and her students had with a comparable corpus of argumentation data (Rosé et al., 2008; Dönmez, Rosé, Stegmann, Weinberger, & Fischer, 2005), but with a different purpose (i.e., they were, at least initially, more interested in helping human coders (semi)-automatically annotate data, rather than in doing online automated analysis and also were focused primarily on classifying single contributions), we chose TagHelper as a means to extract meaningful, yet shallow, attributes of text. Given text strings, TagHelper performs stemming (i.e., finding the roots of words), extracts textual attributes such as unigrams and bigrams (single words and pairs of words occurring in the text), part-of-speech bigrams (paired grammatical structures such as noun-verb, adjective-noun) and punctuation, and filters out words that are not likely to contribute to a classifier's performance such as stop and rare words. The various attributes extracted by TagHelper were then used, in conjunction with structural and temporal attributes of the discussion maps, as descriptive attributes of the student contributions in our machine learning and case-based graph matching approaches.

At the shape and paired-shape level, we pursued a classic machine learning approach (McLaren et al., 2007; Scheuer & McLaren, 2008), using an off-the-shelf data mining tool called RapidMiner (formerly called *YALE*, Mierswa, Wurst, Klinkenberg, Scholz, & Euler, 2006), since (a) such an approach is often successful in finding patterns in data when annotated data with well-defined categories is available (our pedagogical experts had defined some categories, were in the process of defining others, and we knew they could annotate the data) and (b) the unit of analysis is consistent between the annotated data and the subsequently classified data. By (b) we mean that we knew we could annotate both individual and paired contributions that could then be used to train classifiers for, respectively, individual and paired contributions (at run-time). Put another way, we knew that at run-time we could precisely identify the structures (i.e., shapes and paired shapes) for which we had developed pre-trained classifiers and could apply the classifiers to the invariant structures. However, at the cluster level, in which an *arbitrary* and *varying* number of contributions (but typically five or less) might represent a particular category, we knew we could not (directly) apply a supervised learning approach. Thus, at that level, we investigated the idea of using *example clusters*, that is, sub-graphs of discussion maps, as a means for searching for similar clusters in new maps (Mikšátko & McLaren, 2008; McLaren et al., 2009). We call this a *case-based graph-matching approach*. It is inspired both by case-based reasoning (Kolodner, 1993; McLaren, 2003) and the query-by-example approach used in database languages (Zloof, 1977). In the following sections, we describe, in more detail, how we developed and applied the various AI techniques to the different levels of the discussion maps to create the classifiers that support the deep alerts of ARGUNAUT, as shown in Figure 4.

First, however, we will briefly describe the annotation process. As discussed previously, the data was collected during actual classroom sessions in Israel,[6] the Netherlands, and the U.K., the final corpus comprising 84 in-class discussions.[7] The coding process was influenced by theoretical notions of dialogism in the context of critical reasoning and argumentation (Schwarz & Glassner, 2007; Schwarz & De Groot, 2007; Wegerif, 2006). The ARGUNAUT pedagogical experts looked for dialogic and argumentation aspects, even at the individual contribution level. All but two of the coding categories were identified through a bottom-up analysis of a relatively small subset of the 84 discussion maps, with single and groups of recurring contributions identified as potentially representing interesting, important, or problematic phenomena. Two of the cluster categories, *Deepening* and *Widening*, were selected in top-down fashion; they are interactional categories that indicate creative reasoning and the emergence of new ideas (De Laat, Chamrada, & Wegerif, 2008; Wegerif, 2007; McLaren et al., 2009). A *Deepening* occurs when students provide further argumentation for an on-going perspective (Baker, Andriessen, Lund, van Amelsvoort, & Quignard, 2007). A *Widening* occurs, on the other hand, when a student (or students) attempts to diverge from the current perspective by either questioning it or presenting a new perspective (Wegerif, 2007). New perspectives enable participants in a dialogue to see things in a new way and expand their understanding, thus a widening move in a debate is also a creative move.

The pedagogical experts agreed on the shape and paired-shape categories and developed coding instructions by jointly working in groups of 2 or 3 coders on a relatively small subset of the corpus. The instructions that were created consist of detailed explanations of when each category code applies, as well as examples of each. Each code is binary; that is, either it applies to an instance ('positive instance') or it does not apply ('negative instance'). After the coders collaboratively reached agreement on and mutual understanding of the instructions by focusing on the smaller set of discussion maps, they worked individually to code all of the instances in a much larger set of

[6] Because the discussion language in Israel was Hebrew, we translated these discussions into English before coding. While we recognize this compromises the authenticity of the data, we believed this was a necessary first step in our experimental work, since focusing on a single language as an initial test of our approach was important to making the work tractable. More is said about this in the "Discussion" section.

[7] Note that while 84 e-discussions were included in our overall corpus and analysis, we did not annotate all discussions with *all* annotation categories. For example, some of the discussion maps were annotated with only shape and paired-shape categories, others were annotated with only cluster categories, while a few were annotated with both.

discussion maps in the corpus. This step of the process was evaluated for inter-rater reliability, computed by the Kappa (κ) statistic,[8] with Cohen's κ (1960) applied when two coders did the coding and Fleiss's κ (1971) applied when more than two coders did the coding. This yielded acceptable values, very near or above 0.7, for all but two of the individual categories (*Critical Evaluation of Opinion* and *Summary*) and all but one of the paired-shape categories (*Qualifier/Compromise*). The categories with unacceptable inter-rater reliability were eliminated from further analysis. Finally, the coders assigned a final code to each disputed instance (i.e., those that disagreed in the Kappa analysis) by jointly reviewing and agreeing on each instance.

At the cluster level, the coding process was similar but much more difficult; thus, we diverged from the procedure a bit. In particular, we were unable to achieve acceptable inter-rater reliability for *any* of the categories, even after multiple iterations of instruction writing, due to the greater complexity of the cluster categories, as well as the variability of the selected structures (i.e., At the shape and paired-shape level, we could programmatically generate *all* instances; at the cluster level the coders not only had to assign categories to instances but also had to *find* instances of the categories). Thus, unlike the process at the shape and paired-shape level, we did not end up with a single, agreed-upon set of coded instances; rather, the result was different sets of coded instances per coder. Nevertheless, we decided to continue our experiments with the cluster categories by focusing on how well our automated analysis approach would find the clusters identified by the individual coders. We argue for the reasonableness of such an approach in the "Discussion" section of the paper.

Shape-Level Classifiers

Shape-Level Categories

Our pedagogical experts annotated a total of 1,188 individual shapes,[9] using the process described above. The full set of original shape-level categories, along with the number of annotations done within each category, the number of positive instances (i.e., the number of shapes that matched the criteria for that category, as determined by agreement between the coders), and the proportion of positive instances, is shown in Table 1.

Originally, we were interested in all of the categories shown in Table 1, but after initial machine learning experiments, we focused both our annotation efforts and machine learning experiments on only two categories, *Topic Focus* and *Reasoned Claim*. Two categories, *Critical Evaluation of Opinions* and *Summary*, were dropped due to low inter-rater reliability. The other three categories (i.e., *Task Management*, *Request for Clarification*, *Intertextuality*) led to weaker machine learning results (i.e., Kappas well under .60) most likely because of imbalanced class distributions and too few

[8] A κ value of 1.0 signifies perfect agreement, a κ value of 0 means agreement at chance level, and κ below 0 means agreement worse than chance.

[9] The number of annotations reported here and in Table 3 corresponds to the data that was actually used for machine learning. The numbers vary between categories (e.g., 1188 for Topic Focus, 671 for Summary) based on a number of factors, such as how much data was available at the time each category was annotated, annotations missed by the coders, and technical issues that resulted in the loss of some instances. Also, the reported number of annotations in Tables 1 and 3 is less than the total number of annotations actually done by the coders. Our earlier work (Scheuer & McLaren, 2008) included and reported the coders' annotations of the initial question shapes, which are provided by the teacher not the students, and thus clearly do not belong in the corpus of annotated student contributions.

examples for one class, two problems well known for their detrimental effects on machine learning (Japkowicz & Stephen, 2002; Weiss, 2004).

Table 1
The Categories and Annotations of the Individual Shapes

Category	*Explanation / Coding*	*Examples*	*Positive Instances*
Topic Focus	A contribution that focuses on the topic or task.	"Its not nice of human beings to exploit animals for their own needs. I think animals also have rights." (Counter-example) "I'm bored."	***994 / 1188*** ***84%***
Reasoned Claim	An individual contribution that contains critical reasoning or argumentation (i.e., claim + backing). Student provides an explanation or some backing (e.g. evidence) to illustrate a position/opinion. If you can add "because" between two parts of the contribution, it is probably critical reasoning.	"I am against experiments on animals, because to my opinion it is not fair to use them against their will while they cannot reject." "Here it's not like with humans, as the father disengages from them, and he doesn't see them even in the afternoon, and he doesn't belong to the pack any more"	***500 / 1188*** ***42%***
Critical Evaluation of Opinions	Evaluation and/or judgment of another's opinion and/or one's own opinion, and/or the relationship between them. Only applies when a contribution is "on topic."	"I haven't got a definitive opinion for or against" "well... if that is your claim so I completely agree with you."	***62 / 671*** ***9%***
Summary	Summarizing previous discussion or calling for such a summary. Only applies when a contribution is "on topic."	"It seems to me you all think the ducks evolved because their environment changed."	***5 / 671*** ***0.8%***
Task Management	Comments about how to proceed with and manage the given task, such as "add titles," "write more," "answer him," etc.	"would you stop sending empty messages?!?!?!" "don't surf the net" "don't forget to add arrows"	***98 / 968*** ***10%***
Request for Clarification	A request for clarification, reason, explanation, information, etc. from another person. Only applies when a contribution is "on topic."	"What are you basing this on?" "What do you mean by that?"	***81 / 671*** ***12%***
Intertextuality	Explicit evidence of quoting or referring to external material. Only applies when a contribution is "on topic."	"It says in Wikipedia that …" "…in our discussions last week in class…"	***23 / 671*** ***3%***

As can be seen in Table 1, all but one category, *Reasoned Claim* with a proportion of positive instances of 42%, showed an overwhelming majority of one class, with proportions ranging between

84% (the positive *Topic Focus* annotations) and more than 99% (the negative *Summary* annotations). Our lack of success in automated learning of some of the discarded categories may also be attributed to the ill-definedness of those categories; more specifically, while humans were able to consistently identify members of the categories, the key attributes of the categories may be too difficult for a computational approach to identify and/or use.

Creation of the Shape-Level Classifiers

The shape and paired-shape classifiers were developed in two steps: First, the annotated data was translated into a format amenable to standard machine-learning algorithms. This is indicated in Figure 4 by the "Deep Loop Data Translator(s)" box. Second, experiments with a multitude of machine-learning techniques (and parameters) were carried out in order to derive the most effective classifiers for each category. The "Machine Learning Training System" box of Figure 4 represents this process.

In our experimentation, we encoded as much information as possible in attribute-value form, without considering the specific categories of interest (e.g., *Topic Focus*, *Reasoned Claim*), in hopes that the inference mechanism would focus on and use the most predictive attributes. Shapes were analyzed in terms of structural attributes (shape and link types, incoming and outgoing links) and textual attributes (textual content of shapes extracted by TagHelper). As part of its pre-processing, TagHelper also did stemming (e.g., reducing "trusting" to "trust") and removal of rare words (i.e., removal of terms that rarely occur in the corpus and thus are unlikely to support learning). Some attributes were dropped (e.g., # in-links of type "opposition") when initial machine learning experiments indicated they did not improve the results. The full set of attributes finally used for shape-level machine learning is shown in Table 2.

Table 2
Attributes used for machine learning at the shape level

Type of Attribute	*Specific Attributes*
Structural	• Shape type • # undirected links • # in-links • # out-links
Textual (derived using TagHelper)	• **Unigrams:** Simple term (equivalent to keyword search) • **Bigrams:** consecutive terms (paired word phrases, such as 'common denominator') • **POS bigrams:** Part-of-speech bigrams (shallow syntactical structures, e.g., Noun-Verb, Adjective-Noun) • **Punctuation:** Obviously, a question mark is a strong indicator of a question • **Text Length:** The overall text length of the contribution

A flow chart of the machine learning experimentation process we employed, as well as how the process fed into the Classifier Web Service, is depicted in Figure 5. The left side of the figure shows the experimentation process. Given the annotated map data, the annotations were first processed by a simple program that extracted and/or calculated the structural attributes of each annotation, such as the shape type and the number of links going in and out of each shape (i.e., the attributes in the first row of Table 2). Next, the data was processed by TagHelper (Rosé et al., 2008) to derive the textual attributes (i.e., the attributes in the second row of Table 2). The derived textual attributes were saved in a file (the "Text Attributes file" shown in the center of Figure 5) so they could also be accessed by the Classifier Web Service, the run-time "user" of the classifiers. Note that temporal attributes (e.g., shape1 was created before shape2, the text of shape1 was written before the text of shape2) were also extracted at the first (structural) and second (TagHelper) extraction step. While the shape level does not use such attributes, the paired-shape level does; this is discussed below. The RapidMiner toolkit, freely available software that supports interactive experimentation with a wide range of machine learning algorithms (Mierswa et al., 2006), then accepted the training data, which now consisted of the derived structural, textual, and temporal attributes and was used to experiment with different machine learning algorithms, different parameters, and so on. Once a suitable algorithm was found, a classifier was generated, one that could also be accessed by the Classifier Web Service. (The right side of Figure 5 is explained later in the paper.)

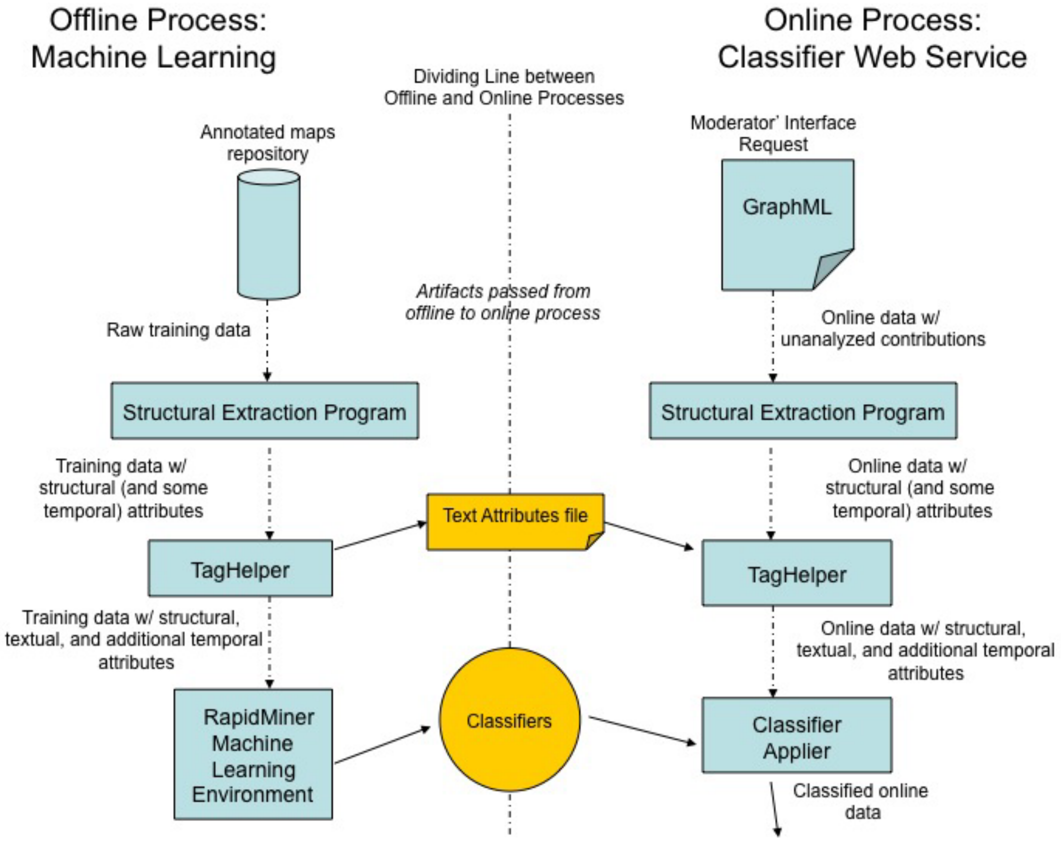


Fig. 5. The process flow of offline machine learning experimentation (shown on the left), which feeds into the "live" Classifier Web Service (shown on the right).

Shape-Level Experiments

The shape and paired-shape experiments described in this paper extend those reported in McLaren et al. (2007) and Scheuer and McLaren (2008). In McLaren et al. (2007) we compared different attribute sets and machine learning algorithms using a subset of data that was available at that time. In Scheuer and McLaren (2008) we used the complete data set, that is, the same data reported here. We tested different machine learning algorithms and attribute sets until satisfactory results were achieved. In the experiments reported here, we used the attribute sets of the best classifiers from the earlier experiments and systematically experimented with algorithms that have been shown to be effective for text categorization tasks: Support Vector Machines (SVM) (Joachims, 1998), Naïve Bayes (McCallum & Nigam, 1998) and Boosted Decision Trees (Boosted DT) (Schapire & Singer, 2000). We also tried the Decision List algorithm, since this led to the best results for some of our classifiers (Scheuer & McLaren, 2008). Unlike the earlier experiments, we also used attribute selection, specifically, chi-squared attribute selection of the top 100 attributes. Since SVMs typically cope well with high dimensional input spaces, we also tested SVMs without attribute selection. Finally, we used SVM with cost balancing activated, meaning that the relative weights (or misclassification costs) of the two classes were adapted to the class distribution during SVM training. In our earlier experiments we achieved increased performance using this option, presumably because of the skewed class distributions in our data set, as described above.

We measured classifier reliability using the Kappa statistic (Cohen, 1960). In this case, κ measured the chance-corrected agreement between a machine-learned classifier and a gold standard. Kappa is not vulnerable to unbalanced class distributions and thus is a more appropriate criterion than the widely used error and hit rate (Ben-David, 2006). The decision as to whether a classifier's performance is acceptable for real-world use depends on domain and application. An acceptability threshold of 0.8, or at least 0.7 (Rosé et al., 2008; Krippendorff, 1980), is recommended in content analysis. Given that in our approach we have teachers will be aware of the possibility of possible misclassifications by the classifiers and the need to use their own judgment, we considered a slightly more generous interpretation sufficient. Thus, we used an acceptability threshold of 0.61, which means, according to Landis and Koch (1977)[10], a "substantial" agreement between a machine-learned classifier and a gold standard.

We estimated the reliability of our classifiers by cross-validating data from one discussion (i.e., the test set) against the data from the remaining discussions (training set). Because data from one discussion was never in both the training and test sets simultaneously, we avoided intra-discussion dependencies and bias.

Figure 6 shows the results at the shape level. We achieved performances well above chance with all algorithms. Four of five Reasoned Claim classifiers surpassed our acceptance threshold of 0.61. The best result was achieved using Boosted Decision Trees combined with attribute selection (κ = 0.66). Results for Topic Focus were somewhat lower: only the SVM classifier without attribute selection yielded acceptable results (κ = 0.62).

[10] Scale: *<0* Poor; *0-0.2* Slight, *0.21-0.4* Fair, *0.41-0.6* Moderate, *0.61-0.8* Substantial, *0.81-1.0* Almost Perfect Agreement

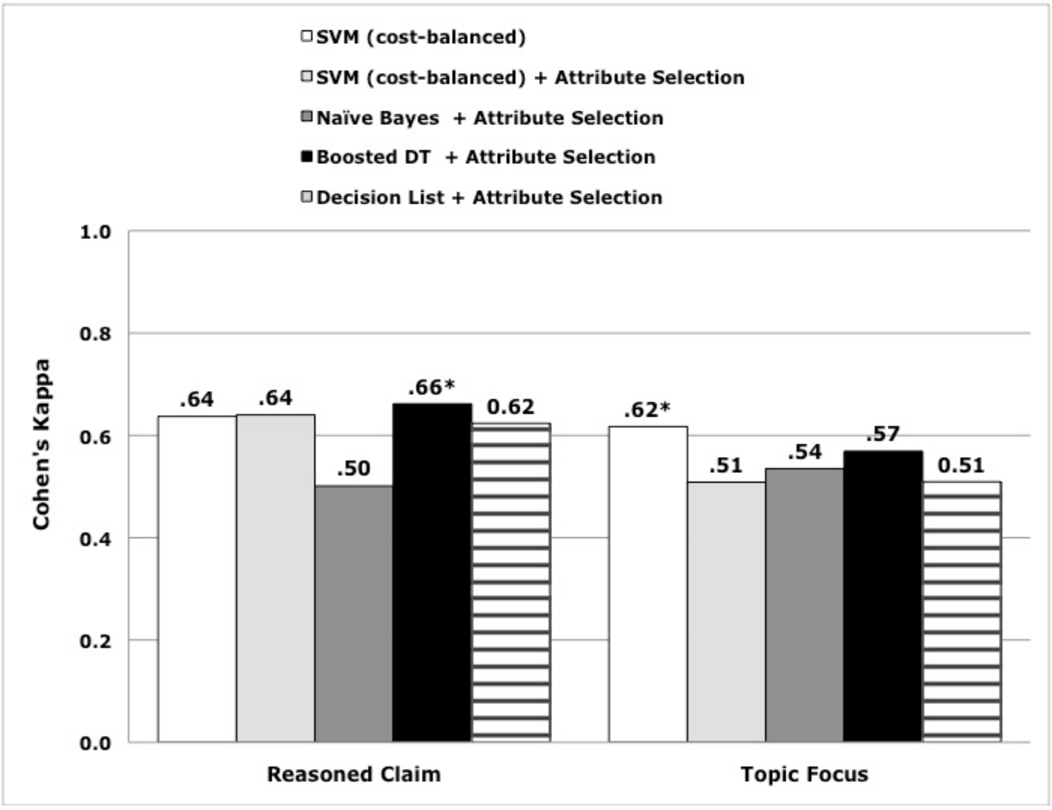


Fig. 6. Comparison of classifier accuracies (in Cohen's Kappa) for five machine-learning algorithms for the shape-level categories Reasoned Claim and Topic Focus.

Integrating the Shape-Level Classifiers with the Classifier Web Service

Since these two classifiers performed well enough, we integrated them into the Classifier Web Service, meaning we made the resulting classifiers available to the Web Service, as shown in the process at the bottom of Figure 5. The Moderator's Interface can, in turn, access these two classifiers as "deep alerts." An example of how the teacher ultimately sees the end product of the classifiers is shown in Figure 7. In this figure, which depicts the moderation of five simultaneous e-discussions, seven of the ten contributions of the displayed e-discussion involve *Reasoned Claim*s, as indicated by the dots shown on the right of seven of the shapes. In the figure, the teacher has selected one of those shapes for closer inspection.

The process that is used to compute the alerts displayed in Figure 7 is sketched out in Figure 5. The right side of Figure 5 depicts the automated online process used to analyze discussions by means of machine-learned classifiers that have been derived in the offline process on the left side of Figure 5. Starting from the top right, a discussion is provided (by the Moderator's Interface) in an XML format called "GraphML," which represents the current discussion state. The discussion is processed in two steps in order to map shapes (and paired-shapes) into the attribute space on which the classifier was previously trained. In a first step the "Structural Extraction Program" segments the input into the respective analysis units (shapes or paired-shapes) and structural attributes are extracted (e.g., on the shape level: shape type, the # of undirected links, the # of in-links, # of out-links, as shown in Table 2). In a second step the TagHelper software extracts text attributes. This process uses the Text Attributes file to ensure that the extracted text attributes are the same as those used during training. (As on the left side of Figure 5, temporal attributes can also be derived at either the first or second extraction step, at least for paired shapes). Shapes (and paired-shapes) are now represented as attribute vectors that comply with the format expected by the machine-learned classifiers (the "Classifier Applier" in Figure 5). The classifier analyzes these attribute vectors, determines the classes of the shapes (and paired-shapes) and returns the results (alerts) to the Moderator's Interface.

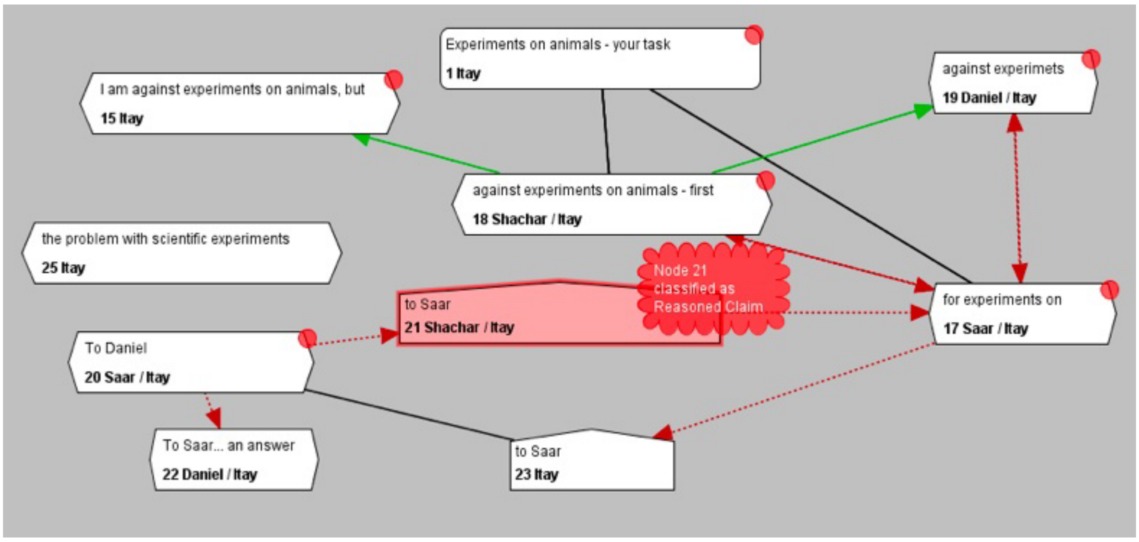


Fig. 7. A zoomed-in view of ARGUNAUT's Moderator's Interface displaying a selected e-discussion and the deep alert *Reasoned Claim*. The *Reasoned Claims* are indicated by dots on the right side of seven of the shapes. The moderator has selected one of the *Reasoned Claims* in the discussion, indicated by the 'cloud,' which shows the node number (21) and the *Reasoned Claim* classification.

Paired-Shape Classifiers

Paired-Shape Categories

Our pedagogical experts annotated just over 770 paired-shape instances across all of the paired-shape categories.[11] As can be seen in Table 3, all of the paired-shape categories showed a substantial majority of one class, with proportions ranging between 71% (negative *Contribution-CounterArgument* annotations) and 94% (negative *Qualifier / Compromise* annotations). Unlike the shape level, four of the five paired-shape categories (i.e., *Question-Answer*, *Contribution-CounterArgument*, *Contribution-SupportingArgument*, *Contribution-FollowedByQuestion*) led to very promising early machine learning results and thus we extensively experimented with these categories.

[11] The numbers vary between categories (e.g., 775 for Question Answer, 768 for Contribution-CounterArgument) based on a number of factors, such as how much data was available at the time each category was annotated, annotations missed by the coders, and technical issues that resulted in the loss of some instances.

(As previously mentioned, the *Qualifier / Compromise* category was dropped when we did not achieve sufficient inter-rater reliability.)

Table 3
The Categories and Annotations of the Paired Shapes

Category	***Explanation / Coding***	***Example***	***Positive Instances***
Question-Answer	A contribution in which the 1st shape is a question and the 2nd shape is an answer to that question by a different student. Typically (but not necessarily) the type of link between the shapes would be "other."	1st shape text: "In the wild does the father separate from the cubs or does he continue to live with them?" 2nd shape, from a different person than the 1st shape: "They all live in a pack" (*The link type between shapes is "other."*)	***117 / 775*** ***15%***
Contribution-Counter Argument	A contribution in which the 2nd shape opposes the claim/argument proposed in the 1st shape and provides reasons or other type of backing for the opposing claim. Typically (but not necessarily) the type of link between the shapes would be "opposition."	1st shape text: "Do not separate, the male should be a partner in what happens even after the birth. The offspring is also his and he should take responsibility." 2nd shape text, from a different person than the 1st shape: "But in a situation like this the mother can get pregnant again and so might neglect a group of cubs." (*The link type between shapes is "opposition"*)	***224 / 768*** ***29%***
Contribution-Supporting Argument	The 2nd shape supports the claim/argument raised in the previous one, and provides reasons or other type of backing for that claim. Typically the type of link between the shapes would be "support."	1st shape text: "We are against. Is it better for the male to get the female pregnant again so she'd abandon the babies?"[12] 2nd shape text, from a different person than the 1st shape: "Separate. We think you should separate because you shouldn't hurt the mom, who will be come a 'pregnancy machine' and move from one pregnancy to the next." (*The link type between the shapes is "support."*)	***217 / 769*** ***28%***
Contribution-FollowedBy Question	The 2nd shape is a question related to the 1st shape. The links vary, based on the role of the question. If it's a rhetorical question, it may be an "opposition" link. If it's a genuine request for information etc., it will likely be "other."	1st shape text: "She's also tormented because she's already exhausted as a result of all the pregnancies and also later on there's the risk that she'll neglect the cubs." 2nd shape text, from a different person than the 1st shape: "Are you for us or against us? Please answer our question." (*The link type between the shapes is "other."*)	***100 / 776*** ***13%***
Qualifier /	A relationship between two	1st shape text: "Do not separate, the male	

[12] The 1st shape is against a shape in which the student says the father should stay with the mother.

Compromise	shapes in which the 2nd shape partially supports and partially opposes the 1st shape, and/or offers some kind of compromise between the claim in the 1st shape and the counterclaim for it, and/or offers some sort of qualifier determining the circumstances in which each of the claims is more valid. Presumably the two shapes will be linked with an arrow of the "other" type, but it's conceivable that either "opposition" or "support" arrows will be used instead.	should be a partner in what happens even after the birth. The offspring is also his and he should take responsibility." 2nd shape text, from a different person than the 1st shape: "You should separate, until the offspring are bigger the female should stay with them alone until they are stronger and then the male can join."[13] (*The link type between the shapes is "other."*)	***39 / 621*** ***6%***

Creation of the Paired-Shape Classifiers

As with the shape-level classifiers, the process shown on the left side of Figure 5 was used to create and test the paired-shape classifiers. More specifically, all pairs of linked shapes in the target maps were first mined and their structural attributes extracted from the discussion maps. Then, the textual attributes were extracted using TagHelper. Temporal attributes were extracted on each step. Next, as with the shape level, the Rapid Miner tool was used to run experiments with the paired shapes, using the extracted structural and textual attributes as the representation for each paired shape (as shown in the bottom left of Figure 5).

The specific attributes used for the paired-shape experimentation are shown in Table 4. Notice two key differences between this set of attributes and those of the shape level from Table 2. First, there are simply more attributes at the paired-shape level. This is due to the greater structure involved at this level (i.e., the additional shape, at least one link between the shapes, and the links associated with the additional shape), as well as the additional text (i.e., two textual contributions instead of just one). Second, the paired-shape level introduces the notion of *temporal sequence*. One shape must have been created before the other, which is represented in our attribute data by the earlier shape being designated "Shape 1," the later shape as "Shape 2." The temporal sequence often also implies interaction between students; whether different students created the two shapes is also captured as an attribute. Note, however, that *any* participant in a discussion can create the link between two shapes, meaning therefore that connected shapes don't necessarily imply interaction between students. In practice, however, the second student almost always creates the link to the first student's shape. The notion of temporal sequence introduced at this level carries over to the cluster level, as well, since multiple shapes, and the links between those shapes, is also a proxy for the temporal interaction between students.

[13] Here there is a compromise offered between those supporting and those opposing separation, which can also be viewed as a qualifier, namely, when is it good/beneficial for the male to be separated, and when it isn't.

Table 4
Attributes used for machine learning at the paired-shape level

Type of Attribute	*Specific Attribute*
Structural	• Link type • Both shapes created by the same user?
Structural + Temporal Sequence	• link and Shape 1 [Shape 2] from same user? • shape type of Shape 1 [Shape 2] • link direction (undirected, from Shape 1 to Shape 2, from Shape 2 to Shape 1)
Textual	• Combined text length of Shape 1 and Shape 2 • Difference in text length between Shape 1 and Shape 2
Textual (derived using TagHelper) + Temporal Sequence	*Same as for the shape level, as shown in Table 2 (i.e., unigrams, bigrams, POS bigrams, punctuation, text length), except applied both to Shape1 and Shape 2 individually. Note that temporal sequence is implicitly introduced, since Shape1, as well as all of its attributes, was created before Shape2, and this ordering is then instantiated via the attribute names (e.g., Shape1_Textlength, Shape2_Textlength..*

Paired-Shape Experiments

For the paired-shape experiments, we used the same experimental setup as for shape-level classifiers, again experimenting with algorithms that have performed well in past text classification tasks (plus Decision List, as explained above). As can be seen in Figure 8, most of the results surpassed our acceptance threshold of 0.61. Boosted Decision Trees proved to be the most effective machine-learning algorithm for two of the four paired-shape categories (Contribution-CounterArgument and Contribution-SupportingArgument) with κ values of 0.71 and 0.66, respectively. In the third category (Contribution-FollowedByQuestion), SVM with attribute selection performed best, reaching κ = 0.75. In the fourth and final category (Question-Answer), Boosted Decision Trees and Decision Lists achieved a (virtually) identical best performance of κ = 0.78. Because these four categories each had at least one classifier exceed the acceptance level, we integrated the best classifier from each category into the Classifier Web Service, as was done with the two best performing shape-level classifiers (as shown on the right side of Figure 5 and discussed above).

What, if any, general conclusions can we come to by looking across the shape and paired-shape results? First of all, the boosted decision tree algorithm clearly worked best with our data, resulting in the best classifier for three of the four paired-shape categories and one of the two shape-level categories. Even in the two cases in which it did not yield the best performance – for the shape-level category Topic Focus and for the paired-shape category Contribution-FollowedByQuestion – it produced the second best approach. Thus, of the prior work in text categorization earlier cited, our results were most in line with those of Schapire and Singer (2000). Also, notice how consistently poorly the SVM algorithm without attribute selection did across the paired-shape categories when it was the best performing algorithm for one of the shape-level categories (Topic-Focus) and close to the best (and above the acceptance level) for the other shape-level category (Reasoned-Claim). This can be explained by the fact that we used far more attributes for paired shapes compared to single shapes. For single shapes we extracted text attributes only from one shape while for pairs we extracted text attributes from two shapes. Even if SVMs are more robust than other machine learning algorithms with respect to high-dimensional input spaces, SVM induction might nevertheless be impaired when the number of attributes exceeds certain limits.

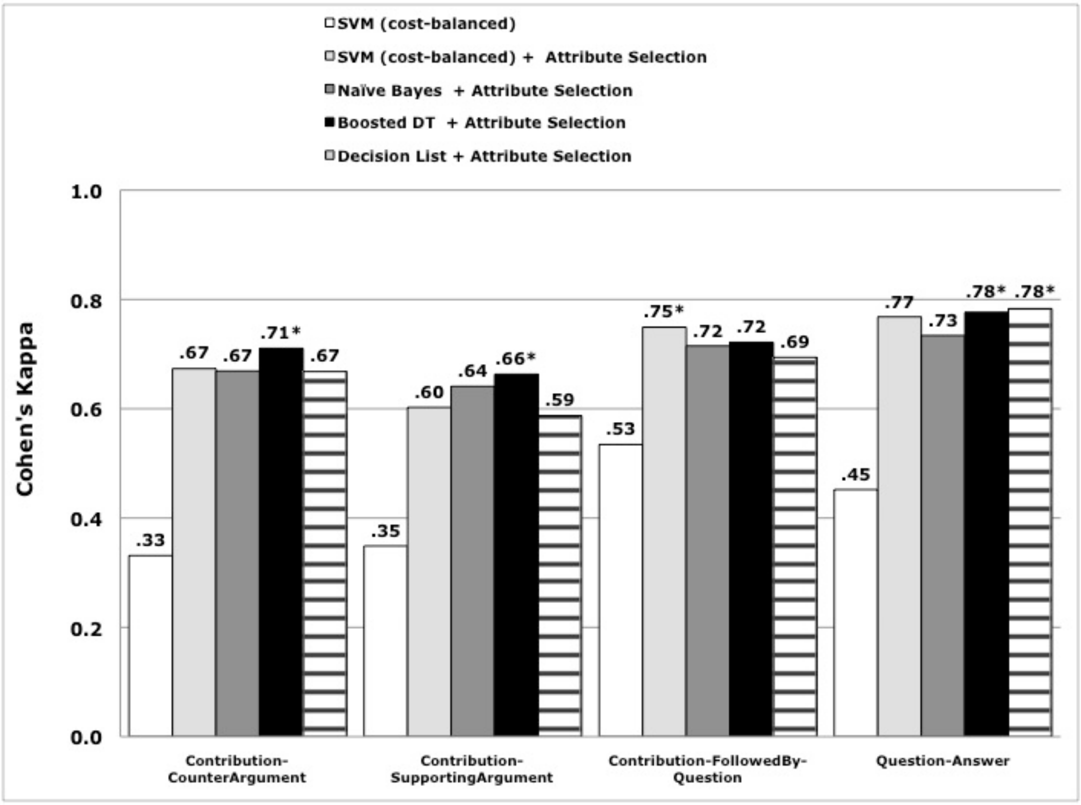


Fig. 8: Comparison of classifier accuracies (in Cohen's Kappa) for five machine learning algorithms for the paired-shape categories Contribution-CounterArgument, Contribution-SupportingArgument, Contribution-FollowedByQuestion, and Question-Answer

Cluster Classifiers

Cluster Categories

Unlike shapes and paired shapes, clusters, arbitrarily-sized collections of connected contributions, had to be *found* in the discussion maps before being classified. In other words, cluster coding involved searching discussion maps for the clusters, rather than having the instances automatically extracted and then coded by humans, as with shapes and paired shapes. Moreover, cluster instances vary in structure and pattern, including, in some (rare) cases, shapes that are disconnected from one another (i.e., no links between). Thus, cluster coding was significantly more difficult than shape or paired-shape coding. As mentioned above, this led to insufficient inter-rater reliability between our coders across all of the cluster categories. This type of group interaction coding, in which links between messages are part of the coding process, is recognized as a very challenging problem, one in which interrater reliability is difficult to achieve (Jeong, 2005, p. 373).

Consequently, a much smaller number of clusters were annotated, ranging between 18 and 36 annotations per category. Table 5 summarizes the cluster categories that were defined, annotated, and evaluated in the experiments described later in this section. Other cluster categories were originally defined, but annotations and analysis of those categories were abandoned when it became apparent that the annotations were far too variable, extremely difficult to find in new discussion maps, or led to very poor results in the initial tests with our cluster classifier algorithm, discussed in the next section. The number of maps showing in the column labeled "# of Ann." indicates the number of maps the coders reviewed and annotated. For the first three categories, a total of 24 maps were reviewed and annotated; for the final two categories, a total of 14 maps were reviewed and annotated.

Table 5
The Categories and Annotations of the Clusters

Category	***Explanation / Coding***	***Example***	***# of Ann.***
Argument **+** ***Evaluation***	Relatively strong and explicit evidence of a discussant evaluating the argument of another discussant. These are relatively difficult clusters to identify; many "borderline" cases were found. Typically these are smaller clusters, usually just a pair, which may be part of larger clusters of different types (e.g., a *Chain of Opposition* cluster can contain a sub-chain in which one of the discussant evaluates the argument of another).	1st shape text (Argument): "It should be prohibited to experiment on animals because it has no special benefit. For example: one puts an unpleasant material in its eyes just to see whether it cause him damage. experiments should be prohibited also because the effect [on us] is not the same. for example: monkeys calm down if you ampute part of their brain, but humans will get handicapped as a result of this. Animals also have families and they are teared apart from their families, or that they born in the lab instead in nature." 2nd shape text (Claim), from a different person than the 1st shape: "your opinion is partly correct. indeed, there are some experiments that hurt animals and are very cruel, and are not necessary, but other experiments are usefull and important, like what I wrote in my argument. your opinions are too extreme." (*Link between shapes is "opposition"*)	***36 (24 maps)***
Chain of Opposition	These are linear sequences of contributions with (typically) two people arguing back and forth, each time raising counter-arguments to one another's claim or argument. The minimal pattern is of three shapes from two different users. Chains of opposition are often quite long (4-6 shapes) and some expression of disagreement, either	1st shape text (Argument), Student 1: "I agree with Riki in saying that the fetus - sick or healthy - is a human being and you can't kill a human being because he is different or disabled...." 2nd shape text (Argument), Student 2: "I think they shouldn't bring him to the world and he is not a human being with a soul in my opinion until 4 months"	***20 (24 maps)***

	with opposition links or textual content, is present. To be annotated as this type of cluster, each reply in the chain will contain some backing (e.g., a reason, an example, some evidence, etc., but possibly of poor quality) and not merely a statement of opposition to the previous contribution.	3rd shape text (Argument), Student 1: "They can have fun with the child despite all the hard work. the pain they will go through when the child dies is the same as the pain from an abortion. so if they are going to suffer anyway - why not at least try?" 4th shape text (Claim), Student 2: "Its not the same thing if he is out or if he is a fetus because when he is out living he is real and as a fetus he is not visable yet - eccept maybe in the x-ray" (*Links between all shapes are "opposition"*)	
Clarification of Opinion Following Feedback	These are typically small clusters, usually simple tri-shape chains where two of the shapes are by the same user. The repeated pattern for this type of cluster is a discussion contribution (shape) by person A; followed by a contribution by person B (linked to the first contribution from person A) and then a third shape with a reply from person A (either linked to the first contribution or to person B's contribution). This reply by A provides clarification of person A's opinion, what he or she tried to express in the previous contribution. The examples found for this type of cluster may differ in the type of links between shapes, but overall are quite similar in terms of structure. Since this is a repeating pattern with fairly fixed structural aspects, the definition and search for this cluster type relied heavily on structural cues.	1st shape text (Claim), Student 1: "I am also extremely against especially when it comes to products like cosmetics... but sometimes there isn't really any other choice... and if it helps humans then I have no choice and I have to agree to experiments, but the conditions under which animals are kept are terrible and it could be done in a more humanistic way" 2nd shape text (Question), Student 2: "But do you think animals should "pay" because we need to make experiments?! Think about it... they are innocent after all... and I feel pity for them so much...." (*Link between 1st and 2nd shape is "other"*) 3rd shape text (Claim), Student 1: "OK. I'm not saying I don't feel pity for them but you can't always find [other] solutions... And be honest for a moment, what do you prefer, to have animals or humans saved... And don't misunderstand me, I really disagree [with experiments on animals] but to be honest I prefer for humans to be saved... and I am definitely against the conditions [under which animals are kept]!!!" (*Link between 2nd and 3rd shapes is "support"*)	***18 (24 maps)***
Widening	Attempts by a student (or students) to diverge from the current perspective by either questioning it or presenting a new perspective. (Note: The new perspective in the example is expressed in the 1st shape, as the students discuss the issue "Will the Internet bring the world together or deepen its divisions") Typically, at least one new branch of contributions is introduced with a "widening." In	1st shape text (Claim), Student 1: "But by becoming more aware of the different cultures, ethics, and religions, surely we are helping to create a divide as we are now being made more aware of the different styles of living and options and opportunities throughout the world that may differ from your experience." 2nd shape text (Question), Student 2: "Could this influence our own way of life?" (*Link between 1st and 2nd shape is "other"*)	***30 (14 maps)***

	the example there are two branches, 1st to 2nd contribution and 1st to 3rd contribution.	3rd shape text (Claim), Student 1: "Also, if people are able to learn about other cultures, etc., then they may not feel that they need to actually experience the culture for themselves, e.g., by visiting other countries, possibly creating a greater divide, because the global community would interact less in person, only through technology." (*Link between 1st and 3rd shape is "other"*)	
Deepening	Providing further argumentation for a perspective that is part of the current discussion.	1st shape text (Claim), Student 1: "Yes, but how are we to overcome that problem? As technology is advancing all the time, and computers are becoming more and more common place in learning resources,..... how are we to deal with the problem of certain individual not having a computer to use at home...?" 2nd shape text (Idea), Student 2: "Its hard to say... An idea would be to only use ICT learning when done in a school setting where everyone has acess to the same materials" (*Link between 1st and 2nd shape is "other")* 3rd shape text (Idea). Student 3: "places such as schools and colleges, etc ... would probly argue that that as a solution to this problem they do provide students with ict facilitys, however i still do not think this works as often they do not have enought facilities to accomadate the number of students." (*Link between 1st and 3rd shape is "other")*	***30 (14 maps)***

Creation of the Cluster Classifiers

The task of creating cluster classifiers was more difficult than developing classifiers at the shape and paired-shape level, due to cluster variability. We did not use a supervised learning approach, as with the shape and paired-shape level, since clusters are of arbitrary size, thus making it quite difficult to a priori identify the instances to be classified. Clusters presented an additional challenge because:

- The data is more complex (i.e., a larger combination of structure and text),
- The data is noisier (i.e., more student mistyping and mistakes),
- Annotated data is much scarcer, due to the difficulty of coding clusters.

We explored several approaches, including, unsupervised learning (Jain, Murty, & Flynn, 1999), pattern mining (Srikant & Agrawal, 1996), pre-defined pattern rules (Harrer, Hever, & Ziebarth, 2007) and supervised clustering (Finley & Joachims, 2005), but ultimately designed and developed an approach that best fit the problem characteristics: ***DOCE*** (***D***etection ***o***f ***C***lusters by ***E***xample) (Mikšátko & McLaren, 2008). DOCE, a *case-based graph-matching algorithm*, is based on the idea of using cluster examples to find similar clusters in other discussions and is inspired by case-based reasoning (Kolodner, 1993; McLaren, 2003), analogical mapping (Forbus, Gentner, & Law, 1994), and database query-by-example (Zloof, 1977). An example cluster, also called a "model graph," is selected and used as a search query for similar clusters across other discussion maps, called "input graphs." The output of the algorithm is a list of matching clusters in the input graph(s), sorted according to a similarity rating.

The DOCE algorithm is summarized in Figure 9. First, the example cluster (model graph) and the discussion map (input graph) are parsed from the XML file format used by the Moderator's Interface for representing a snapshot of the discussion. Both graphs are preprocessed as follows: (1) an adjacency matrix representing the structure of the graph is constructed and (2) each contribution and link in the discussion graph is characterized by an *attribute vector*, extracted from attributes such as shape/link type, text length, link direction and whether the same user created two linked shapes. As with the shape and paired-shape classifiers, TagHelper (Rosé et al., 2008) further enriches the attribute vectors with additional information from the text analysis of contributions (e.g., unigrams, bigrams, part-of-speech bigrams). As per Figure 5, a Text Attributes file is generated by TagHelper and used by the DOCE algorithm. (Note that this file is the same one used by the shape-level and paired-shape classifiers). Additionally, DOCE extends the attribute vectors of shapes (links) with shape and paired-shape classifications.[14] In the next step, DOCE compares the attribute vectors of vertices/edges in the model and input graphs by calculating their distance in a manner similar to unsupervised learning algorithms. The proximity is pre-computed for each pair of model/input objects and stored in the similarity matrices. Finally, an inexact graph matching method based on a customized version of the edit distance algorithm (Wong, You, & Chan, 1990; Tsai & Fu, 1979) is employed to find clusters with the highest structural and content similarity to the model graph. Similar algorithms have been successfully used for various purposes, such as computer vision (Gregory & Kittler, 2002) and retrieving relevant principles from ethics cases (McLaren, 2003).

The matching works as follows. An $A*$ search algorithm explores all possible vertex-to-vertex mappings between the model and input graph. In each step, a partial mapping of vertices is extended by adding a new vertex-to-vertex assignment that has the maximum content similarity (pre-computed in the similarity matrices) and the minimum structural difference, as measured by edit distance. The edit distance between partially matched graphs is calculated as a minimal sequence of primitive graph operations (such as "add an edge," "delete an edge," and "delete a vertex") that are required in order to make the graphs isomorphic. The final matching cost is the sum of all vertex/edge similarities and penalties for the edit operations. The first *n* complete mappings (i.e., mappings that cover all model vertices) are returned as resulting clusters and sorted in ascending matching-cost order.

Thus, the algorithm matches *similar* clusters on *generic* graph structures in an inexact manner (e.g., some of our cluster examples and shapes matched unconnected subgraphs in the input graphs). The matching is driven by both the *graph structure* (e.g., the users involved in the cluster and shape types) and *content* of contributions (i.e., a characterization of the text of the contribution). Note that the detection of all matching subgraphs is an NP-Complete problem but only in theoretical, not practical, terms. Applying heuristics similar to Tsai and Fu (1979) and Gregory and Kittler (2002) significantly reduces practical complexity. For instance, one can apply a heuristic that estimates future node and edge mapping costs including possible edit distance penalty; this approach is employed in our algorithm. Using heuristics such as these, the method performs reasonably well on graphs of moderate size (dozens of vertices). The graphs in our particular domain are typically within this range. (But see below the practical analysis we did of the algorithm with some of the annotated data described in Table 5.) The DOCE algorithm is described in further detail in (Mikšátko, 2007).

[14] Although we experimented with the machine-learned classifications of shapes and paired shapes, we used the human-annotated classifications in our cluster experiments. See the "Discussions" section for more discussion of this topic.

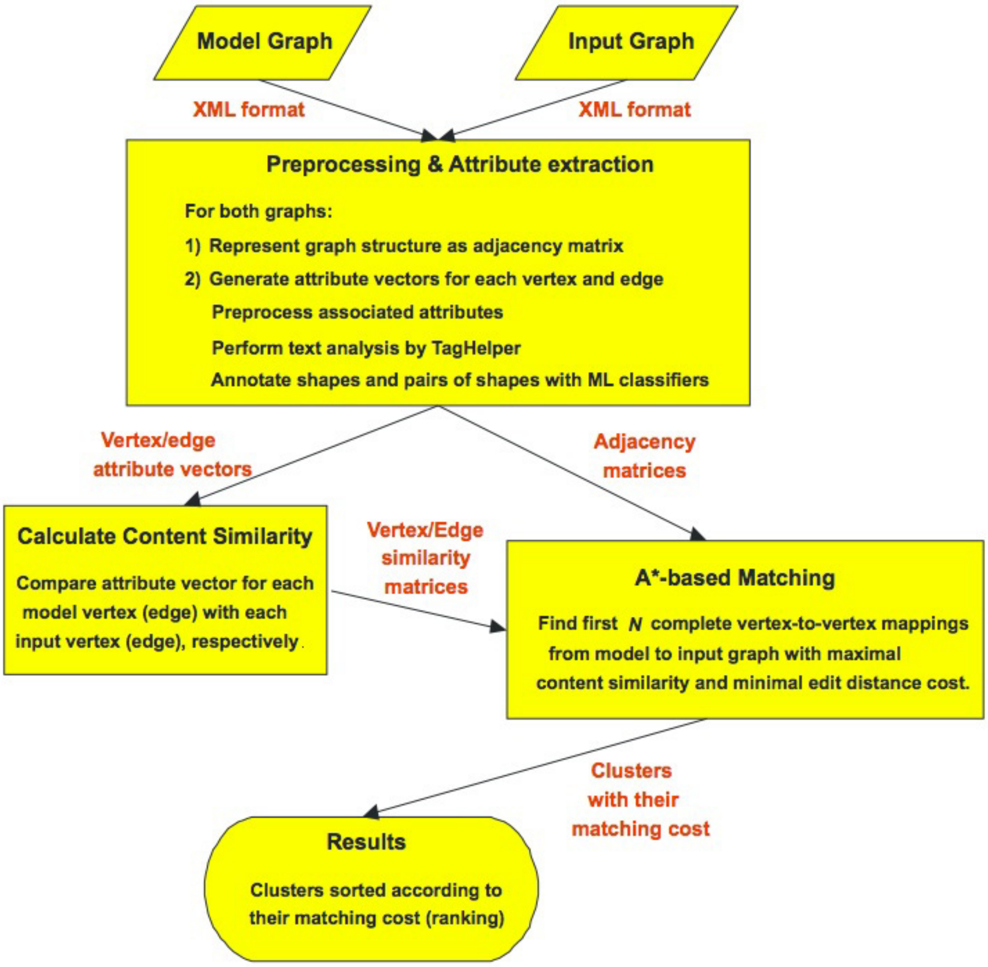

Model Graph
Input Graph
XML format
XML format
Preprocessing & Attribute extraction
For both graphs:
1) Represent graph structure as adjacency matrix
2) Generate attribute vectors for each vertex and edge
Preprocess associated attributes
Perform text analysis by TagHelper
Annotate shapes and pairs of shapes with ML classifiers
Vertex/edge attribute vectors
Adjacency matrices
Calculate Content Similarity
Compare attribute vector for each model vertex (edge) with each input vertex (edge), respectively
Vertex/Edge similarity matrices
A*-based Matching
Find first N complete vertex-to-vertex mappings from model to input graph with maximal content similarity and minimal edit distance cost.
Clusters with their matching cost
Results
Clusters sorted according to their matching cost (ranking)

Fig. 9. The DOCE algorithm.

Note that the DOCE algorithm does not follow the approach depicted in Figure 5, which is used to apply *machine-learned* classifiers to discussion maps to analyze shapes and paired-shapes. The cluster classifier, which the DOCE algorithm implements, processes discussion maps in a different manner. The input / output interfaces of DOCE and the machine-learned classifiers are identical, but in contrast to the ML classifiers, which work on a "flat" representation, DOCE's *A** search operates directly on a graph representation enhanced with pre-processed attributes for comparing subgraphs (e.g., contributions are enhanced with TagHelper attributes extracted from their texts). Additionally, DOCE does not require training because it uses pre-stored models as the basis for searching for new clusters.

Cluster Experiments

We evaluated DOCE and its ability to find clusters, at least those annotated by specific coders, in two separate experiments using the data summarized in Table 5. First, in Cluster Experiment #1, we conducted an experiment with the 24 discussion maps from an Israeli high school and an Israeli college, annotated with the cluster types *Argument + Evaluation*, *Chain of Opposition*, and *Clarification of Opinion Following Feedback* (Mikšátko & McLaren, 2008). Next, in Cluster Experiment #2, we conducted an experiment with the 17 discussion maps that were created in a U.K. university and annotated with the cluster types *Widening* and *Deepening* (McLaren et al., 2009).[15] We focused on different cluster types in each experiment because of the different interests of the pedagogical experts in Israel and the U.K.

Each of our experiments was conducted as follows. Given the separate, annotated corpus from the actual classrooms in Israel and the U.K., we took the 24 (and 17) discussion maps, and did the following:

- For each annotated example in each corpus (call them Corpus-Israel and Corpus-UK), we ran DOCE with that annotation as the model graph (as per Figure 9) against all of the other discussion maps from the same corpus containing at least two annotations of that type. (Note: We excluded maps that did not contain at least two annotations of the particular type being searched, since such discussion maps would be too easy to achieve recall of 1.0, by simply getting a relevant match to the single occurrence.)
- We considered a *relevant match* to be 70% overlap, e.g., the following annotated example and found cluster would constitute a relevant match, since there is a 75% node overlap (bold-faced nodes overlap):
  - Model graph (i.e., annotated example): (Node1, **Node3**, **Node4**, **Node5**)
  - Cluster Match (i.e., from Input Graph): (**Node3, Node4, Node5**, Node6)
- We varied parameters, such as the number (N) of clusters that were returned by DOCE and the relative impact of structural and textual properties on the similarity score of cluster pairs (e.g., Is it more important that texts or shape types are similar?).
- We evaluated information retrieval metrics recall, precision, and recall+precision on each run of DOCE. These metrics were calculated as follows:
  - *Recall* represents the number of annotations of type x covered by DOCE within its Top N, divided by the count of annotations of type x in the searched map (value between 0 and 1.0).
  - *Precision* is the number of relevant matches of type x found by DOCE in the Top N divided by N (value between 0 and 1.0).

We considered recall to be the most critical metric in our experiments, as it was most important to find *all* of the interesting clusters in a given discussion. The number of relevant matches (i.e., precision) has somewhat lower importance, in our estimation, since we as researchers, and humans in general, are typically clever enough to filter out irrelevant matches.

Unfortunately, there is no "gold standard" for performing the type of retrieval task done by DOCE. Thus, there was no other computational model to compare to DOCE in the experiments. However, as a baseline test (and as reported in Mikšátko & McLaren, 2008), in Cluster Experiment #1

[15] Eleven discussion maps were coded for both Widening and Deepening. Three discussion maps were exclusively annotated for Widening, three other maps were exclusively coded for Deepening.

we compared DOCE to a simple program that returns random clusters.[16] While a random algorithm is, admittedly, a low bar to exceed, doing *significantly* better than random demonstrates that DOCE is clearly finding (at least some) clusters of interest.

Additionally, we also tested whether *combining* results of multiple runs of DOCE with different model clusters of the same type might further improve the results. That is, we wanted to answer the question: Can multiple example clusters of the same type, provided as input to DOCE, lead to even better results in retrieving relevant clusters than a single example cluster? Such a data fusion technique, combining the preferences and results of several "experts," has been successfully used in information retrieval (IR) (Aslam & Montague, 2001) and machine learning (boosting, bagging) (Han & Kamber, 2006). For example, in IR, the same query is often submitted to several search engines and the results combined. Prior work has shown improvements in recall and precision using such a technique (Aslam & Montague, 2001).

Generally, the combination algorithms can be split into two categories, depending on whether they use ranking (i.e., positions in the Top N list) or relevance scores, and further split into another two categories: whether training is used (or not) for obtaining weights for each model. After preliminary experimentation with one algorithm from each of the four categories, we determined that an algorithm that calculates the average of relevance scores, and then orders the results according to those scores, was (usually) best for our data. The "best" models for each cluster type were chosen according to the highest recall and precision results achieved when using that model graph individually as input to DOCE, with preference given to recall, since we considered it the most important metric. Although the specific number of "best" models to combine is difficult to determine in general, and likely varies by cluster type, our preliminary results indicated that three "best" models usually led to the best overall results and, hence, we used this number of model graphs in the experiments.

Cluster Experiment #1 Results

Our Cluster Experiment #1 results were first reported in Mikšátko and McLaren (2008). The results, averaged across all cluster types, input models, and discussion maps of Corpus-Israel, are shown in Figure 10. In this figure, we compare the results of the random matcher to (a) the average of *all* human-annotated models provided as input to DOCE, (b) the average using only the *best* input model per cluster type (i.e., the human-annotated cluster of each type that led to the best recall and precision values), and (c) the average per cluster type using the best 3 input models, using the combination algorithm discussed above. As can be seen, the DOCE algorithm led to much better results than the random algorithm in all cases. Significance was confirmed by t-tests ($p < 0.0001$ in all cases).

The best performing cluster type was *Clarification of Opinion Following Feedback* (over 0.80 average recall across all cluster examples), followed by *Chain of Opposition* (approximately 0.80 average recall) and *Argument + Evaluation* (over 0.70 average recall). Additional conclusions drawn from the data of Figure 10 include:

[16] The dataset used in the experiment described here and in the paper (Mikšátko & McLaren, 2008) is the same, although the number of maps differs. In the earlier paper, we reported 27 maps, a number that also included 3 maps that were deleted from the dataset and the experiment due to the very low quality of the discussions, as determined by the pedagogical experts on our team. Here we more accurately report 24 maps, that is, a count of the maps in the dataset that were fully annotated and used in the experiment.

- Using the best model of each cluster type, DOCE was able, on average, to detect 90% of the cluster examples annotated by the pedagogical experts (third bar from left).
- The precision result can be interpreted as meaning that between 1/3 and 1/2 of the retrieved clusters were relevant. While this value is generally low for precision in information retrieval terms, it is worth noting that the input discussion maps contain approximately 3.3 annotations on average. Since we set DOCE to always return the top 5 to 10 clusters, it will always produce relatively low precision values.

Besides the results shown in Figure 10, we were able to determine that the *stability* of the DOCE algorithm with respect to different models is relatively high. More specifically, an average of approximately six out of ten clusters (for the best configuration) were in common when comparing two results sets produced by pairs of models against the same map, despite the fact that the models are often from discussions with different topics. In addition, a more fine-grained analysis of the results uncovered that, on average, more than 60% of relevant clusters are *exact* matches, in comparison to only 11% for the random matcher.

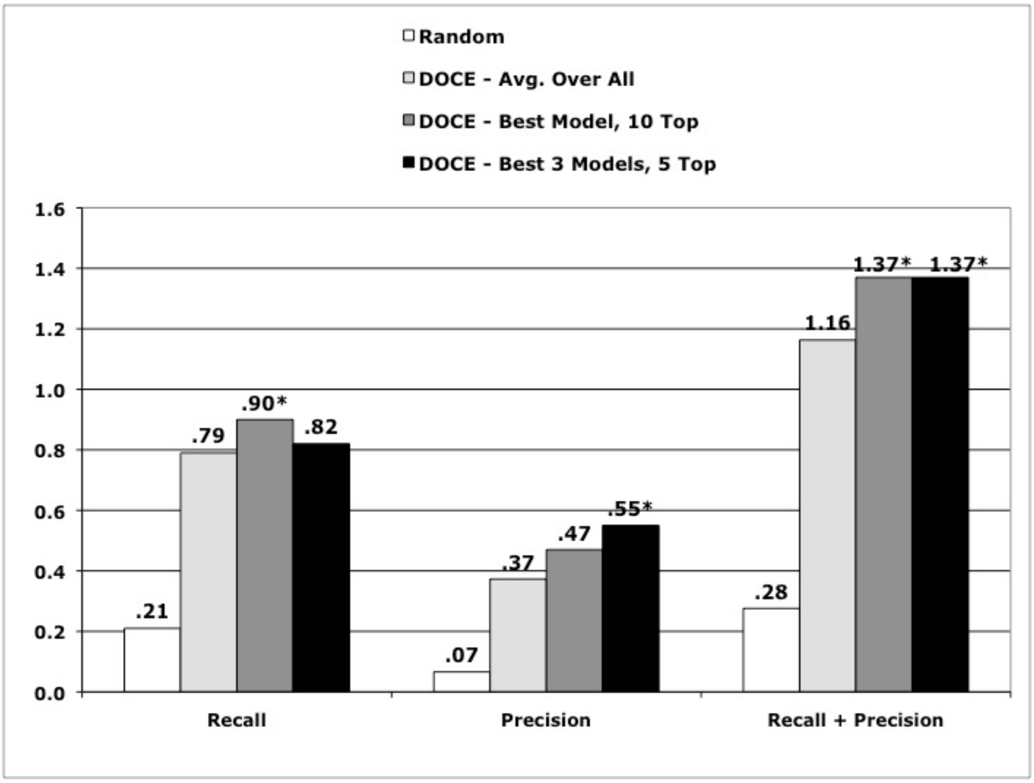


Fig. 10. Results of Cluster Experiment #1. The results are averaged across all three Corpus-Israel cluster types (i.e., *Argument + Evaluation*, *Chain of Opposition*, and *Clarification of Opinion Following Feedback*).

Cluster Experiment #2 Results

The results of Cluster Experiment #2 were first reported in McLaren et al. (2009) and are summarized in figures 9 and 10. In this later experiment, the DOCE algorithm was not compared to the random algorithm, since the results of Cluster Experiment #1 clearly showed the superiority of DOCE over a random clustering algorithm. In these figures, we compare (a) the average of *all* of the human-annotated models provided as input models to DOCE, (b) the single *best* input model of this cluster type, and (c) the best 3 input models of this cluster type, combined via the algorithm discussed above. Note, that the best results for *Deepening* and *Widening* are quite reasonable (the middle bar for recall, precision, and recall+precision in each of the figures), especially for recall, the most important metric. For instance, notice that the best *Deepening* model graph led to a recall of 0.80 and precision of 0.52 (the middle bar in each of the first two sets of three metrics in Figure 11). The average results, calculated across *all* of the annotated clusters, are not very good, considerably less than what was achieved in Cluster Experiment #1 (the leftmost bar for recall (0.42), precision (0.27), and recall+precision (0.69) in each of the figures). However, focusing on the best results is more important because, by design of the DOCE algorithm, only the best examples of *Deepening* and *Widening* will be subsequently used as model graph inputs to DOCE. That is, once one finds the best model for a particular cluster type – or the best set of models – that model (or models) will then be used as a "search probe" for all subsequent searches.

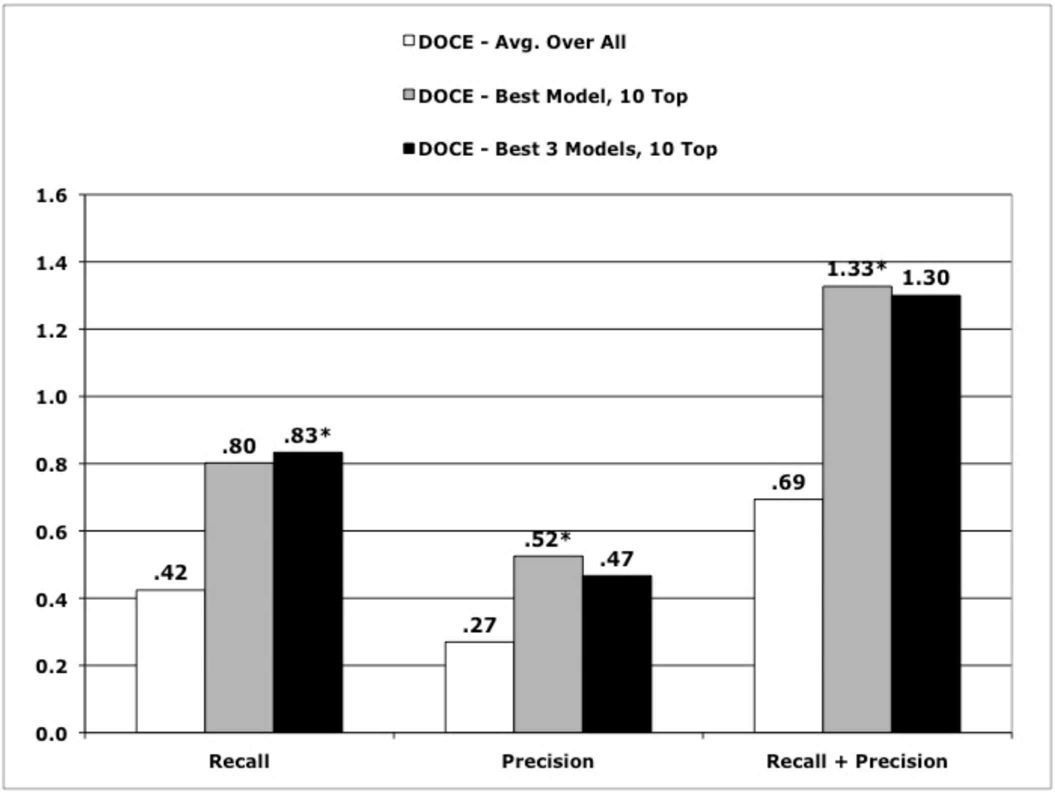


Fig. 11. Results of Cluster Experiment #2 – Part 1. Results of finding *Deepening* clusters.

As in Cluster Experiment #1, we also tested whether combining the results of multiple input models given to DOCE could further improve the results. The third bar in each set of three bars in Figures 9 and 10 depicts those results. Notice that for the *Deepening* clusters (Figure 11) the combination approach did marginally worse (i.e., recall+precision = 1.30 for the combination approach compared to 1.33 for the single best model), but for *Widening* clusters shown (Figure 12), the combination approach led to slightly better results (i.e., recall+precision = 1.49 for the combination approach compared to 1.42 for the single best model). Note also that DOCE did better in identifying *Widening* clusters, which, as discussed earlier, are a hallmark of creative reasoning, than in identifying *Deepening* clusters. In particular, note that the best *Widening* recall (0.93), precision (0.59), and recall+precision (1.49) in Figure 12 improves upon the best *Deepening* recall (0.83), precision (0.52), and recall+precision (1.33) from Figure 11.

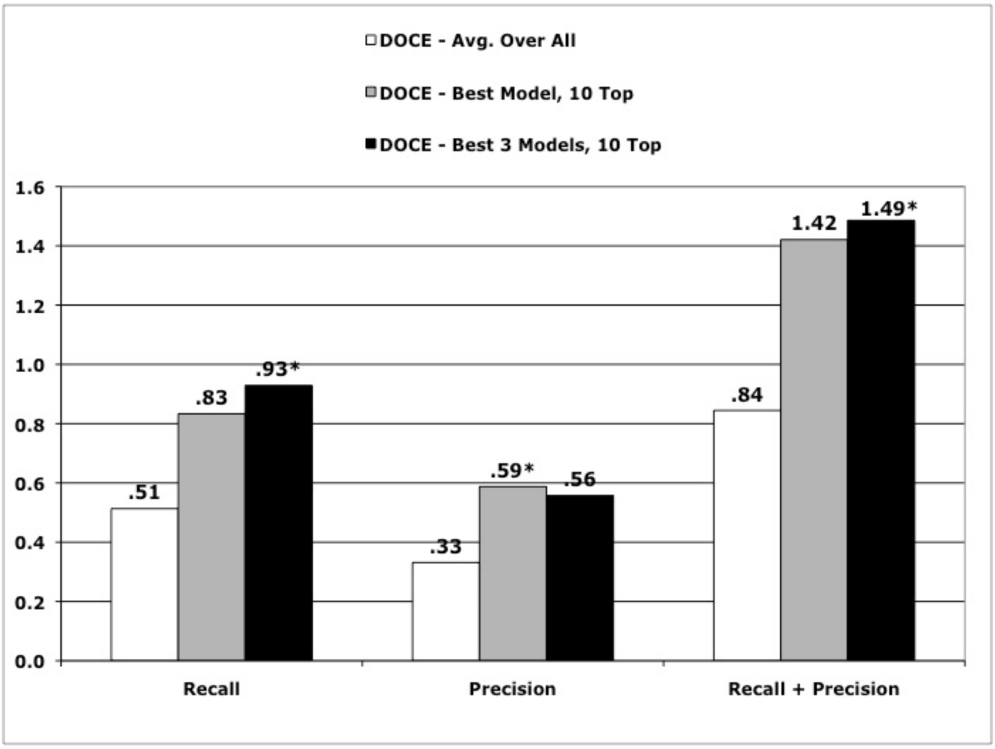


Fig. 12. Results of Cluster Experiment #2 – Part 2. Results of finding *Widening* clusters.

Interestingly, the results we achieved with Corpus-UK in the second experiment were better than most of the results with Corpus-Israel in the first experiment. Most significantly, the DOCE algorithm performed better in finding snippets of the creative reasoning represented by *Widening* clusters than it did in finding more standard argumentation structures, such as *Chain of Opposition* (i.e., a chain of contributions by students in which they go back in forth in argue for and against a given issue) and *Argument + Evaluation* (i.e., a student makes an argument which is then evaluated by another student).

Computation and Usability Analysis of DOCE

The run-time of optimal graph-matching algorithms such as DOCE (i.e., an algorithm that always finds the best solution independent of the size of search space) is known to be NP-Complete (Garey & Johnson, 1979). Thus, there is a potential that the algorithm could be unusable. In computational theoretic terms, this means that in the worst case, the run-time of the algorithm grows exponentially with the input size, resulting in non-acceptable run-time behavior even for moderately sized input. However, such algorithms, when applied in practical contexts in limited ways, are often usable without ever bumping into the theoretical limits. To explore this, we analyzed the run-time characteristics of DOCE as it was run against the data from our 14 discussion maps in Cluster Experiment #2.

Not surprisingly, the empirical run-time analysis showed a linear relationship between the text length of the input graph and the pre-processing time (the time needed to extract attributes from the input and model graph). On average, the pre-processing took about 10 seconds per map and peaked at roughly 21 seconds for very large maps (> 60 contributions) with plenty of text. The search time (finding the most similar matches) was highly influenced by the size of the model graphs: Models of size three were in all cases unproblematic with all search times below five seconds. For models of size four the maximum search times rose to 34 seconds. For models of size five we saw a couple of cases with run-times above one hour, confirming the theoretical assumption of an exponential run-time growth; on the other hand, in the vast majority of cases the run-time was still at an acceptable level.

In summary, it is clear that very large discussion maps can be a problem for DOCE, especially when a teacher uses the algorithm in real-time fashion. On the other hand, the Corpus-UK discussion maps were quite large, created over several days of classroom use and are almost certainly at (or above) the upper limit of practical map size. Model graphs did not lead to any excessive search times, as long as the models did not exceed four nodes in size. Model graphs that reached five nodes in size led to some extreme cases, but it should be noted that these cases only occurred when a five-node model graph was used to search very large discussion maps. Generally speaking, as long as model graphs do not exceed five nodes they are practically usable, especially when discussion maps are reasonably sized. Furthermore, to improve run-time performance, it is possible to use application-specific heuristics that restrict the set of target nodes mapped to a source node according to application heuristics. For instance, instead of DOCE evaluating the match of a model graph node to *every* node of an input graph, it could consider only nodes that are linked to an already-matched node, since our emphasis is on highly linked sets of nodes. However, such heuristics can miss the optimal solution, if one exists.

Integrating the Cluster Classifiers with the Classifier Web Service

Because the six cluster types experimented with in Cluster Experiments #1 and #2 led to classifiers that performed well enough, and the practical run-time characteristics were also acceptable, they were integrated into the Classifier Web Service. Figure 13 shows a screen shot of the results of one of the classifiers, *Argument + Evaluation*, applied to a live discussion map. Similar to Figure 3, this figure shows the state of the Moderator's Interface after a teacher has selected a single *Argument + Evaluation* cluster, as well as showing other clusters that were found by the DOCE algorithm. The other clusters are indicated by small circles shown on the geometric centers of the cluster. (For a two-shape cluster, such as *Argument + Evaluation*, the geometric center is the middle of the link between the shapes.)

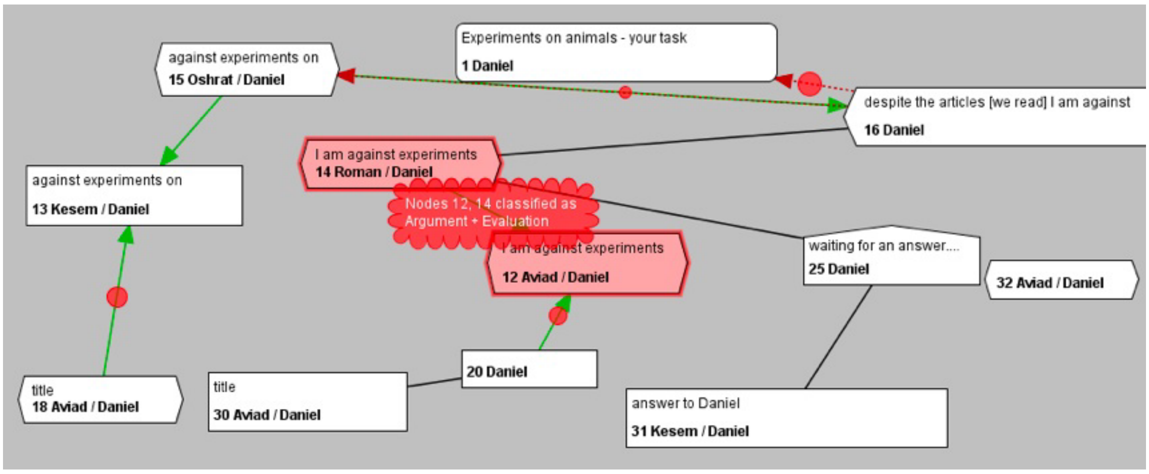


Fig. 13. Graph display of *Argument* + *Evaluation* clusters, with one match highlighted.

DISCUSSION

So how have we fared regarding our primary research question? More specifically, is it possible to automate the identification of salient contributions and patterns in student e-discussions? To tackle this question, we have taken a systematic approach to developing and empirically evaluating AI-based classifiers, as described in detail above. Our approach started with the generation of machine-learned classifiers of individual e-discussion contributions, moved to the creation of machine-learned classifiers of pairs of contributions, and, finally, led to the development of a novel AI-based graph-matching algorithm that classifies arbitrarily sized clusters of contributions. The classifiers, in all cases, are designed to account for the structural, textual, and sequential aspects of the graphical e-discussions. At each of these levels of analysis, we have run systematic evaluations of the resultant classifiers using actual classroom data, comparing the classifiers to human coding of the same data. Our evaluations have led to satisfactory (or better) results for many of the classifiers and have eliminated others. In summary, we have at least preliminarily answered our primary research question: it does appear that, through a combination of AI techniques, it is possible to automate the identification of key contributions and patterns in e-discussions.

An important contribution of our work has been the emphasis on accounting for structural, textual, *and* temporal aspects of the contributions made by students to e-discussions within graphical tools such as Digalo and FreeStyler. To our knowledge, no prior research on automated analysis has attempted to cover *all* of these aspects of online discussions – and particularly not with an emphasis on real-world use to support a teacher in the classroom, as we have on ARGUNAUT – and thus our research makes a novel contribution to both the fields of artificial intelligence in education and computer-supported collaborative learning.

Of course, there are still issues to address. First, while we achieved very promising results, all levels of automated analysis were done on relatively small datasets, especially the cluster level, so the results must be interpreted with caution. The total number of examples used for computing machine-learned classifiers is not really small in absolute numbers (> 1000 shapes, > 700 paired shapes), but for text categorization tasks, much larger document sets are often used to capture variety and rule out the ambiguities in natural language use (Sebastiani, 2002). On the other hand, the prior work of Dönmez et al. (2005), in which machine-learned classifiers were successfully derived from a corpus size comparable to ours (approximately 1250 instances) for a similar classification task (i.e., analysis of relatively short texts, data from pedagogical discussions, categories of argumentative structures), provides at least some indication that our corpus may have been large enough. Our approach also uses structural and temporal attributes that are potentially more predictive than text attributes alone. In addition, Joachims (1998) argues that support vector machines (SVMs), the machine learning algorithm that emerged as the best at the shape level in our experiments, are a good choice for text categorization tasks due to the high dimensional input space, few irrelevant attributes, sparse attribute vectors, and the fact that most text categorization problems are linearly separable. In addition, boosted algorithms, which were the best algorithms for most of the categories at the paired-shape level, have also been successfully applied to text categorization and/or with large attribute sets (e.g., Schapire & Singer, 2000). Thus, the results of our machine learning experiments are in line with prior research.

With respect to DOCE, we acknowledge that our data sets are, for practical reasons, quite small. As previously discussed, the difficulty of annotating discussion maps, due to having to find and code arbitrarily-sized clusters, hampered our ability to evaluate the algorithm. This raises a second issue related to the first: the DOCE algorithm has so far only been able to emulate the classifications of individual coders, since, at the cluster level, coding has been quite difficult and good interrater reliability elusive. Despite this shortcoming, we argue that the evaluation reported in this paper is meaningful because DOCE detects clusters that are similar to provided model graphs; in other words, the algorithm adapts to the "style" of the annotator. Note also that many annotations were marked by the coders as "borderline" examples that could have negatively influenced the results, yet we kept and used these annotations and still obtained arguably good results. In practical terms, it seems unlikely that we will obtain a high level of interrater reliability for the arduous and inexact task of identifying meaningful clusters, at least not without much more detailed specifications and extensive training of coders. Thus, while one of the next steps of our research will be to collect a larger set of cluster examples to more fully test DOCE, it may simply be that DOCE's real contribution, and, we argue, a valuable one, is as an emulator of particular coders.

A third issue to address, also related to the DOCE algorithm, is to extend the algorithm to "know" when its results are weak and inform the teacher of this. Currently, the algorithm always returns the best N matches when given a model graph to run against a set of input graphs. But what if there are simply no (or very few) instances of the example cluster type in the input graphs? In such a case, one would like the algorithm to have the capability to tell the teacher that there are no (or very few) matches, and not just blindly return its top N matches. One approach that is worth investigating in this vein is the use of meta-rules to determine whether to venture advice in specific graph-matching cases (McLaren & Ashley, 2001). For instance, a software layer on top of DOCE might apply rules (or simple thresholds) to evaluate specific attributes of the match, such as how well the language attributes match between the source and target, and use the results to decide whether the match results are good enough to report, as well as how many of the top N matches are acceptable.

A fourth issue for further investigation is this: to what extent are our classifiers special purpose versus general purpose in their behavior? The classifiers may have incorporated idiosyncrasies of the particular data set that we used. These particularities include: a limited set of discussion topics, a particular cultural background (Israel and the U.K.), specific (and possibly varying) instructions the students received before the discussions started, and the age of the participants. Would the classifiers function equally well when used, for instance, in a school in the United States with completely different discussion topics? Further research is necessary to investigate whether, and to what extent, the classifiers are transferable to different contexts. A related issue concerns the language of the contributions. Recall that language translation was done for the Hebrew discussion maps collected in Israel. It is, in general, necessary to have caution regarding conclusions based too strongly on translated text. The general, unanswered question is this: how would the system behave with classifiers trained (or given examples) in one language in classifying e-discussions in another language? It just so happened that the initial and best source of data came from Israel, even though our first objective was to standardize and evaluate English-language e-discussions. Testing the classifiers generated from the Israeli data against the maps collected in U.K. classrooms is in order, as well as developing classifiers from native English e-discussions.

Fifth, we were unsuccessful in our plan to use the results of the simpler levels of the discussion graphs for classifications at the more complex levels. At the paired-shape level, we conducted some preliminary experiments using shape-level annotations as attributes for learning paired-shape classifiers: in some cases, we saw small improvements, in other cases no improvement at all. These experiments simulate a best-case scenario in which shape-level classifications are (virtually) 100% correct, since we used the human annotations rather than the automated classifications. There are several reasons why we did not follow up on this: (1) given that we only realized small improvements using "perfect" shape classifications we could not expect improvement using fallible real classifications, (2) since we always achieved better results for paired-shape classifiers than for shape classifiers, it seems unlikely we could have improved the better classifiers through the bootstrapping of worse ones, (3) paired-shape classifiers are based on the same data as shape-level classifiers, so it seems unlikely that reusing the same data in a different way would result in classification improvement, and (4) even if we could slightly improve the paired-shape classifiers' accuracies using shape-level classifiers, we would have to pay with worse run-time performances, since running the more complex classifiers would always then require running of the less complex classifiers. At the cluster level, we also tried to use the results of the shape and paired-shape level within the cluster classifiers, but the overall results differed by a negligible amount in a series of preliminary experiments (+/-1% change in precision and recall (Mikšátko, 2007)). It is worth noting that the cluster level would also be subject to the additional computational overhead discussed above (i.e., the execution time of the cluster classifiers would increase if shape and paired-shape classifications would be required as part of cluster classification). Our general conclusion is that the idea of incorporating lower-level classifications as attributes of the higher-level classifications is not likely to improve results – and it may actually introduce problems such as the run-time issue; thus, we will abandon this direction in future work.

Finally, the ultimate question we would like to answer is our second research question: do automatically identified contributions and patterns in student e-discussions help teachers in e-moderating simultaneous, synchronous e-discussions? That is, will the ARGUNAUT system, and more specifically its deep alerts, *really* help teachers e-moderate multiple collaborating groups simultaneously? Put another way: are the deep alerts relevant and reliable enough to support teachers? While this is admittedly still an open question, there is at least some preliminary evidence that the ARGUNAUT system and approach will be successful. For instance, in a within-subject, counterbalanced study with two teachers, Wichmann et al. (2009) compared moderation with and without the support of ARGUNAUT's basic awareness tools (i.e., the shallow alerts, e.g., the number and type of contributions made by students). Each teacher was asked to moderate three synchronous e-discussions simultaneously, with as many as 4 students in each discussion, during each of their moderation periods with and without support. They found that both teachers performed better on 60 given tasks (e.g., who is the strongest collaborator in the discussion?) when given access to the awareness tools than when not given access to the tools. Schwarz and Asterhan (in press), in a design research study involving three moderators asked to simultaneously moderate as many as four groups of three students, evaluated the ability of the moderators to apply sophisticated e-moderation strategies (e.g., encouraging groups to open new perspectives) while dealing with the multiple, synchronous discussions. They were able to show, through protocol analysis, that it is possible for moderators to employ complex e-moderation strategies with the help of ARGUNAUT's basic awareness tools. These results provide at least some indication that ARGUNAUT can support teachers as they e-moderate synchronous discussion groups in a classroom.

But what about the added benefit of the deep alerts? As an adjunct to the Wichmann et al. (2009) study, an additional small study was conducted to explore this issue (Asterhan et al., 2008). A pre-service teacher was asked to moderate a session consisting of two groups of six students each using a version of ARGUNAUT with no alerts (*No-Alert* condition) followed by an identical session using a version of ARGUNAUT with deep alerts (*With-Alert* condition). While knowledge-gain and task-performance dependent measures did not vary across conditions, the teacher did spend less time finding answers in the *With-Alert* condition. This is important, since quick, real-time moderation is a fundamental goal of ARGUNAUT and just-in-time feedback is necessary to provide effective support for the learner. An additional illuminating observation was that in the *With-Alert* condition, the teacher initially insisted that students had not engaged in off-topic conversation, even though they had; it was not until the teacher saw the results of the Topic-Focus deep alert (see Table 1) that he realized off-task conversation had, in fact, occurred. Taken together, these preliminary results suggest that (a) the ARGUNAUT system can be effective in supporting a teacher in e-moderating multiple, synchronous discussions, and (b) the deep alerts (i.e., the automated analysis) can be of additional benefit to the teacher in this task. Of course, these findings must be taken as preliminary and tentative, given the small size of the studies done so far, but the results are at least suggestive that the answer to the second research question will be affirmative.

On the other hand, we have already anticipated some shortcomings in the deep alerts and have some clear ideas about how they might be extended to better support teachers (Scheuer & McLaren, 2008). Most of the patterns found by the classifiers, especially shape and paired-shape classifications, are more descriptive than alerting in nature, that is, they point to general discourse patterns whose occurrence is not problematic *per se*. For instance, while teachers generally want their students to stay on-topic, should they intervene on every off-topic contribution? Such an intervention pattern would probably interrupt the flow of the discussion and cause more harm than good. But what if eight of ten total contributions are off-topic? In such a discussion something clearly is wrong and the teacher should probably intervene. These examples show that fine-grained patterns in isolation may have limited significance. However their *aggregation* may point to critical general situations, suggesting the need for teacher action. We have incorporated some simple aggregating indicators in the Moderators' Interface to help with this; for instance, the Moderator's Interface displays the total number of positive classifications per discussion, such that teachers can identify undesirable overall discussions at a glance (e.g., a discussion containing predominantly off-topic contributions). We envision going beyond simple aggregations, developing models on top of the classifications. The goal is to provide teachers with a clearer and more global picture by qualitatively *summarizing* the discussions. Such a model might define numeric conditions such as "If more than 30% of all links are part of argument-counterargument clusters then the discussion qualifies as *controversial.*" A similar approach is taken in the Group Leader Tutor (Israel & Aiken, 2007) that classifies the collaborative effort of a group along three dimensions by aggregating and combining counts of collaborative skills that have been displayed. Alternatively, one could use inductive inference. Labels might be assigned to complete discussions, and a machine-learned model computed that infers discussion-level classifications from shape and paired-shape classifications.

Although simple numeric aggregations might be indicative of positive and negative classroom situations, they might also be misleading, since such approaches tend to lose the dynamic and fine-grained aspects of discussion. For instance, a discussion may contain many *Reasoned Claims*, but on closer inspection there may be no interaction between students, that is, students may only respond to the original question shape (a case described in Scheuer & McLaren, 2008). One way to detect larger patterns, without using a simple aggregation approach, is a knowledge-engineering approach, in which complex graphical patterns are explicitly pre-defined based on shape and paired-shape classifications. A Graph Grammar formalism, such as described in Pinkwart, Ashley, Lynch, & Aleven (2008), may be an option here. Finally, we do not expect to detect all imaginable types of clusters automatically. In the end, the identification of some complex patterns will still be up to the teacher, who can qualitatively inspect the discussion maps with the aid of the highlighted alerts.

RELATED WORK

The closest research to ours is the work of Rosé and colleagues, who developed the text analysis tool, TagHelper (Rosé et al., 2008), also used in our work. Originally, they aimed at freeing corpus analysts from the tedious task of manually coding large amounts of data, rather than focusing on our objective: analyzing online discussions. In one application (Dönmez et al., 2005), they analyzed a corpus of 1,250 coded text segments along multiple dimensions of argumentation in order to derive machine-learned classifiers. Some of the phenomena in their work, like argument-counterargument chains and grounded claims, are quite similar to the categories we are interested in and have developed classifiers for. They achieved acceptable Kappa values of 0.7 or higher for six of seven dimensions. More recently, they developed an approach to providing dynamic support to dyads collaborating on a problem-solving task (Kumar, Rosé, Wang, Joshi, & Robinson, 2007). Similar to our approach, they perform online analysis of textual communication data, in their case, chat data. In contrast to our approach, their analysis results are not displayed to human teachers but are instead used to trigger automatic interventions, sent directly to students. An empirical study showed significant learning benefits in terms of analytical knowledge and conceptual understanding when the dynamic support was provided. The Rosé et al. work also differs from ours in that their automated analyses do not take into account structural or temporal data from the rich semantics of graphical discussions (i.e., typed shapes and links). They do, however, account for the more primitive structure and sequence of threaded discussions (Rosé et al., 2008).

Goodman and colleagues (2005) also have applied a machine-learning approach to support collaborative problem solving in the EPSILON system. Peer groups work together on a problem in the domain of object modeling techniques (OMT). Their collaboration takes place within a shared whiteboard, similar to the shared workspaces of ARGUNAUT, in which diagrams (e.g., class diagrams) are constructed. Peers communicate via a text chat with a sentence opener interface; an agenda tool supports task management. In contrast to ARGUNAUT, dialogues are focused rather narrowly on a single domain (OMT) and on the coordination of task-related activities. The system evaluates aspects concerning domain (e.g., domain knowledge of peers), task (e.g., progress in solving the task) and, similar to our objectives, possible problems in the collaboration process (e.g., unanswered questions). The sentence opener interface plays a critical role; it is used to automatically assign a dialogue act classification to each chat contribution. These dialogue acts are used as a meta-level description of the discourse and serve as attributes for machine-learning analyses, bypassing the complicated task of natural language (or text) processing that we tackle in our work. The analysis approach of Goodman et al. (2005) differs to ours in other respects: our analysis is based on graphical argument diagrams, in which the relations between contributions are made explicit, that is, we know existing relations and their types; they analyze sequences of classified chat messages in which neither relations nor their types are explicitly represented. We apply automated shallow language processing techniques to approximate the content of messages; for the same purpose, they use simple human-defined keyword lists. Another difference is how the analysis results are used: We support students indirectly via a moderator, whereas they follow more of an ITS-style approach: Some of the results are displayed immediately to the peers via meters, while direct support is provided by means of an artificial peer agent that verbally interacts with the participants.

Soller's work, also in the context of EPSILON, was concerned with the analysis of chat conversations that accompanied activities in a problem-solving environment (Soller, 2004). The chat tool was enhanced by a sentence-openers interface to structure users' communication (the same one used by Goodman et al.) and to make automated analysis feasible, similar in some respects to how the pre-defined shapes and link types of ARGUNAUT make automated analysis tractable. The analysis of EPSILON aimed at identifying episodes in which students communicated their knowledge to their peers ("knowledge sharing episodes") and episodes in which they failed to do so ("knowledge sharing breakdowns"). She computed machine-learned classifiers, more specifically, Hidden Markov Models (HMMs), from data annotated as knowledge sharing episodes and knowledge sharing breakdown episodes. In a second step, she investigated reasons for knowledge sharing breakdowns using multidimensional scaling (MDS) and clustering techniques. In this way she identified different patterns (clusters) of successful knowledge sharing and knowledge sharing breakdowns. The key differences to our work are: (1) Soller analyzes sequences of actions, whereas we analyze sub-graphs within a discussion map that are sometimes sequential, sometimes parallel, (2) the *textual* content of contributions is not analyzed in Soller's work; it relies exclusively on dialogue acts, which are automatically inferred from the selected sentence openers, (3) she uses a sequential learning approach that can be applied to sequences of arbitrary length whereas our ML approach is currently restricted to single shapes and pairs, (4) a sequential learning approach captures dependencies between contributions naturally (because it assumes by design a sequential dependencies); our ML approach with flat representations requires a pre-processing step to encode sequential (and other) dependencies explicitly in its attribute space. On the other hand HMMs cannot be used to analyze graphical structures, such as the ones we are working with in ARGUNAUT. Also, Soller's approach requires a pre-segmentation of knowledge-sharing sequences; a restriction that introduces a new potential source of error in actual practice. Furthermore, our machine learned classifiers are more reliable: Soller reports an accuracy of 74%, which we re-calculated to a Kappa of 0.47, considerably lower than our best six machine-learned classifiers.

Ravi and Kim (2007) analyzed threads in a technical discussion board in order to call an instructor's attention to threads that contain unanswered questions. They developed two classifiers (linear SVMs), one for detecting questions (QC), and a second for detecting answers (AC). They achieved accuracies of 88% (QC) and 73% (AC). Similar to our approach, they used shallow language attributes, although somewhat more elaborated ones than the ones we used (e.g., in addition to uni- and bigrams they also use tri- and quadrograms). In addition to the classifiers they implemented a rule-based thread profiler that assigns one of four typical profiles to threads (accuracies vary between 70% and 93%). A follow-up to this work, Kim et al. (2008) describe PedaBot, a threaded discussion system that scaffolds students' discussions by retrieving messages from past discussions that are possibly relevant to the current context. Discussions are about technical topics (e.g., operating systems). The system is based on a text corpus (extracted from two text books), which has been analyzed to (1) determine relevant topics (corresponding to book chapters and sections), (2) determine relevant technical terms (corresponding to glossary entries) and (3) develop a topic profiler (a classic information retrieval term frequency–inverse document frequency (tf-idf) classifier that assigns topics to messages based on technical terms contained within them). The topic profiler, in conjunction with a further application of tf-idf term weighting, can then be used to fetch possibly relevant messages from prior discussions. The key differences between our approach and theirs are: (1) their analytical procedure is enhanced with domain knowledge (topics and technical terms) and hence is very domain-specific; our approach, on the other hand, while likely tied somewhat to our "domain" is less strictly so, (2) the output of PedaBot and its predecessor is also more specific (i.e., a concrete message instead of an abstract classification) and (3) the development of the topic profiler did not require annotation efforts. Finally, as with the other prior work already mentioned, the Kim et al. work does not specifically take into account structure and temporal attributes to support its classification approach, as do both our machine learning and DOCE approaches.

Jeong (2003; 2005; 2006) has developed a software tool called the Discussion Analysis Tool (DAT) that uses sequential analysis to capture and model sequences of speech acts. DAT models an online threaded conversation as a network of transitional probabilities, called a transitional state diagram, building the diagram from pre-labeled data. For example, in one diagram generated by DAT from real data a "challenge" act occurred with a probability of 0.52 after an "argument" was made, while an "explanation" followed an "argument" with a probability of 0.08. DAT has been used, for instance, to evaluate the interactions that are most likely to promote critical thinking (Jeong, 2003) and the effects of supportive language (e.g., I agree, ask questions) on subsequent group interactions (Jeong, 2006). DAT can also be used to evaluate whether threaded conversations deviate from a norm; it creates a Z-score matrix to show probabilities that were significantly higher or lower than expected in one state diagram compared to another. While Jeong shares our goal of analyzing collaborative, and argumentative, discussions, DAT is intended more as a post-hoc analysis tool for supporting various types of group interaction studies, and not for online "live" analysis. In particular, Jeong's system does no language analysis; it depends on human post-hoc coding (or real-time labeling) to identify the individual acts.

Sequential pattern mining algorithms (Agrawal & Srikant, 1995), which discover frequently occurring sequences of activities in a dataset, are an intriguing possibility for finding salient patterns of student collaboration. Such algorithms have, for instance, been used to account for both language and contextual attributes – to a reasonable degree of accuracy – in classifying email messages (Carvalho & Cohen, 2005). Sequential pattern mining algorithms have also been tried in collaborative learning scenarios. For instance, Kay, Maisonneuve, Yacef, and Zaïane (2006) used a variant of the Generalized Sequential Pattern Algorithm (GSP) (Srikant & Agrawal, 1996) to identify common interaction patterns in a source repository and Wiki log data from student software development projects. The authors' goal was to build tools that can flag student interactions that are indicative of problems, so that the tools can later be used to assist students in recognizing problems in new interactions. However, in contrast to log records, which contain simple events such as "file X was modified," the ARGUNAUT discussion maps contain natural language and highly complex graph structures, thus making the Srikant and Agrawal approach far less likely to succeed. Furthermore, the Srikant and Agrawal approach is an unsupervised learning approach, searching for frequently occurring patterns, whereas our classifiers target predefined and pedagogically relevant categories. Furthermore, Rosé et al. (2008) were able to show that designing or selecting appropriate linguistic and contextual features may be more important than using sophisticated supervised sequential learning algorithms, such as those used in approaches like Carvalho and Cohen (2005).

A more manual approach to pattern mining was also evaluated on the ARGUNAUT project (Harrer et al., 2007). Their tool was designed for mining user-specified sequences of actions in the discussions (i.e., pattern rules), such as "create shape"/"add link"/"modify text." The tool was able to detect some commonly occurring patterns. However, this algorithm, which does exact-matching, was unable to detect patterns that differed in subtle and imprecise ways from one another. A key objective of our work on the DOCE algorithm was to develop a tool that could perform such inexact matching.

CONCLUSION

Students in classrooms around the world are starting to use visual argumentation tools for e-discussions. A key goal of such tools is to help students learn to discuss and argue in a well-founded, rationale manner. In order to effectively e-moderate multiple, synchronous discussions, teachers need a tool that provides summarized feedback regarding the on-going e-discussions. The ARGUNAUT system represents an important step toward this goal; it is designed to support a teacher in e-moderating multiple conversations through "alerts" that display interesting and critical events in the discussion, such as off-topic contributions, question-answer pairs and chains of opposition. The more advanced of ARGUNAUT's alerts, based on advanced AI techniques, leverage the structure (e.g., the types of shapes used by students, the links between shapes), textual contributions made by students, and the temporal sequence of the contributions.

While this project was driven by the real requirements of a new and emerging classroom paradigm, it was also guided by a basic research question: is it possible to automate the identification of salient contributions and patterns in student e-discussions? To address this research question, we undertook a systematic and empirical approach, described in full detail in this paper. First, we developed machine-learned classifiers that analyze structure, natural language text, and the temporal sequence of graphical discussions in order to classify single contributions and pairs of related contributions. After extensive experimentation, we were able to determine that six of the classifiers work successfully, closely emulating the annotations of humans who had earlier coded the e-discussions. Since the classical supervised machine learning approach was not applicable to the variable nature of clusters of student contributions (i.e., arbitrary numbers of contributions, as in, for instance, a chain of opposition), we next developed a more flexible graph-matching algorithm, called DOCE, to find and identify clusters of variable size, given example clusters as input. Our initial empirical results at the cluster level also showed the feasibility and potential of this new approach: five of the clusters of interest achieved good enough results to be included in the live ARGUNAUT system. Our overall results, while obtained with relatively small sample sets and thus not providing a conclusive answer to the main research question, indicate that it is very likely that automated analysis of contributions and patterns in student e-discussions *is* possible.

Due to the positive results, we integrated all of the most successful classifiers, a total of eleven, via Web Service technology, into the overall ARGUNAUT system to provide teachers with tools to monitor and analyze discussions at different levels of granularity (shapes, paired-shapes, clusters). It is important to note that the classifiers are not bound to the ARGUNAUT system; the Web Services architecture allows us to use the classifiers in other e-discussion contexts. Next steps will include testing our classifiers more exhaustively, with more data, and in particular investigating accuracy in contexts in which the discussion data differs more significantly from our training data. Most critically, we need to determine how useful the provided alerts are for teachers in actual, fast-paced moderation scenarios. In particular, while we have started to address our second research question – do automatically identified contributions and patterns in student e-discussions help teachers in e-moderating simultaneous, synchronous e-discussions? – there is still much work to do in this direction. Are the specific categories helpful enough for a teacher to improve his or her moderation? Are the predictions accurate enough? Is the summarization of activities at the right level to support fast-paced e-moderation? We suspect that the current alerts may identify too fine-grained patterns. As discussed above, we will next explore ways to derive higher-level characterizations based on aggregation and combinations of fine-grained patterns that highlight student problems more clearly.

ACKNOWLEDGEMENTS

This work would not have been possible without project partners Rakheli Hever, Reuma De Groot, Maarten De Laat, Matthias Krauß, Adam Giemza, Rupert Wegerif, Ulrich Hoppe, Raul Drachman, Astrid Wichmann, and Christa Asterhan, as well as other members of the ARGUNAUT project team. The 6th Framework Program of the European Community, Proposal/Contract No. 027728, sponsored this research.

REFERENCES

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. In P.S. Yu & A.L.P. Chen (Eds.) *Proceedings of the 11th International Conference on Data Engineering* (ICDE '95), (pp. 3–14). Washington, D.C.: IEEE Computer Society.

Andriessen, J.E.B., & Schwarz, B.B. (2009). Argumentative design. In N. Muller Mirza and A.-N. Perret Clermont (Eds.) *Argumentation and Education: Theoretical Foundations and Practices.* (pp. 145-174). Dordrecht, Heidelberg, London, New York: Springer.

Aslam, J.A., & Montague, M. (2001). Models for metasearch. In D.H. Craft, W.B. Croft, D.J. Harper, & J. Zobel (Eds.) *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. (pp. 276-284). New York: ACM Press.

Asterhan, C.S.C., Wichmann, A., Mansour, N., Wegerif, R., Hever, R., Schwarz, B.B., & Williams, M. (2008). Argunaut deliverable D6.3: Evaluation report on the pedagogical content of the Argunaut system (revised). Unpublished Manuscript. http://www.argunaut.org/ARGUNAUT_-_D6-3.zip

Baker, M., Andriessen, J., Lund, K., van Amelsvoort, M., & Quignard, M. (2007). Rainbow: A framework for analyzing computer-mediated pedagogical debates. *International Journal of Computer-Supported Collaborative Learning*, 2, 315-357.

Ben-David, A. (2006) What's wrong with hit ratio? *IEEE Intelligent Systems*, 21(6), 68-70.

Carvalho, V.R., & Cohen, W.W. (2005). On the collective classification of email “speech acts.” In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 345–352). New York: ACM Press.

Coghlan, M. (2001). eMODERATION – Managing a new language? Net*Working 2001 Conference – from Virtual to Reality, Brisbane, October 2001. http://michaelcoghlan.net/mc/ESL_WORK/ARTICLES_PRESENTATIONS/nw2001/emod_newlang.htm.

Cohen, E.G. (1994). Restructuring the classroom: Conditions for productive small groups. *Review of Educational Research*, 64(1), 1-35.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement,* 20(1), 37-46.

Constantino-Gonzalez, M.A., & Suthers, D. (2002). Coaching collaboration in a computer-mediated learning environment. In G. Stahl (Ed.) *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community* (CSCL-02). (pp. 583-584). Mahwah, NJ: Lawrence Erlbaum Associates.

De Groot, R., Drachman, R., Hever, R., Schwarz, B., Hoppe, U., Harrer, A., De Laat, M., Wegerif, R., McLaren, B.M., & Baurens, B. (2007). Computer supported moderation of e-Discussions: The ARGUNAUT approach. In C. A. Chinn, G. Erkens & S. Puntambekar (Eds.), *Mice, Minds and Society, Proceedings of the Conference on Computer Supported Collaborative Learning* (CSCL-07), Vol 8 (pp. 165-167). International Society of the Learning Sciences, Inc. ISSN 1819-0146.

De Laat, M., Chamrada, M., & Wegerif, R. (2008). Facilitate the facilitator: Awareness tools to support the moderator to facilitate online discussions for networked learning. In *Proceedings of the 6th International Conference on Networked Learning* (pp. 80-86) Lancaster: University of Lancaster.

De Laat, M., & Wegerif, R. (2007). Perspectives/rules to evaluate discussions. Argunaut public deliverable D5.1. http://www.argunaut.org/.

Dillenbourg, P., Baker, M., Blaye, A., & O'Malley, C. (1995). The evolution of research on collaborative learning. In P. Reimann & H. Spada (Eds.) *Learning in humans and machines: Towards an interdisciplinary learning science* (pp. 189–211). Oxford: Elsevier/Pergamon.

Diziol, D., Rummel, N., Spada, H., & McLaren, B.M. (2007). Promoting learning in mathematics: Script support for collaborative problem solving with the Cognitive Tutor Algebra. In C.A. Chinn, G. Erkens & S. Puntambekar (Eds.) *Mice, minds and society. Proceedings of the Computer Supported Collaborative Learning Conference* (pp. 39-41). International Society of the Learning Sciences, Inc.

Dönmez, P., Rosé, C., Stegmann, K., Weinberger, A., & Fischer, F. (2005). Supporting CSCL with automatic corpus analysis technology. In T. Koschmann, D.D. Suthers, T.-W. Chan, (Eds.) *The next 10 years! Proceedings of the Conference on Computer Supported Collaborative Learning* (CSCL-05) (pp. 125-134) Mahwah, NJ: Lawrence Erlbaum.

Finley, T., & Joachims, T. (2005). Supervised clustering with Support Vector Machines. In L. De Raedt & S. Wrobel (Eds.) *Proceedings of the 22nd International Conference on Machine Learning* (pp. 217-224). New York: ACM.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin,* 76(5), 378–382.

Forbus, K.D., Gentner D., & Law, K. (1994). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science,* 19, 141-205.

Garey, M.R., & Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: WH Freeman & Co.

Gil, J., Schwarz, B.B., & Asterhan, C.S.C. (2007). Intuitive moderation styles and beliefs of teachers in CSCL-based argumentation. In C.A. Chinn, G. Erkens & S. Puntambekar (Eds.) *Mice, minds and society. Proceedings of the Computer Supported Collaborative Learning Conference* (pp. 219-228). International Society of the Learning Sciences, Inc.

Goodman, B., Linton, F., Gaimari, R., Hitzeman, J., Ross, H., & Zarrella, G. (2005). Using dialogue features to predict trouble during collaborative learning. *User Modeling and User-Adapted Interaction,* 15, 85-134.

Gregory, L., & Kittler, J. (2002). Using graph search techniques for contextual colour retrieval. In T. Caelli, A. Amin, R.P.W. Duin, M.S. Kamel, & D. de Ridder (Eds.) *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition* (pp. 186-194). New York: Springer.

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann.

Harrer, A., Hever, R., & Ziebarth, S. (2007). Empowering researchers to detect interaction patterns in e-Collaboration. In R. Luckin, K.R. Koedinger, & J Greer (Eds.) *Proceedings of the 13th International Conference on Artificial Intelligence in Education* (AIED 2007) (pp. 503-510). Amsterdam: IOS Press.

Harrer, A., Ziebarth, S., Giemza, A., & Hoppe, H.U. (2008). A framework to support monitoring and moderation of e-discussions with heterogeneous discussion tools. In P. Díaz et al (Eds.) *Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies* (ICALT-2008) (pp. 41-45). Washington, D.C.: IEEE Computer Society.

Hever, R., De Groot, R., De Laat, M., Harrer, A., Hoppe, H.U., McLaren, B.M., & Scheuer, O. (2007). Combining structural, process-oriented and textual elements to generate alerts for graphical e-discussions. In C.A. Chinn, G. Erkens & S. Puntambekar (Eds.) *Mice, minds and society. Proceedings of the Computer Supported Collaborative Learning Conference* (pp. 286-288). International Society of the Learning Sciences, Inc.

Hoppe, H.U., & Gaßner, K. (2002). Integrating collaborative concept mapping tools with group memory and retrieval functions. In G. Stahl (Ed.) *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community* (pp. 716-725). Mahwah, NJ: Lawrence Erlbaum Associates.

Israel, J., & Aiken, R. (2007). Supporting collaborative learning with an intelligent web-based system. *International Journal of Artificial Intelligence in Education*, 17, 3-40.

Jain, A.K., Murty, M.N., & Flynn, P.J. (1999). Data clustering: A review. *ACM Computing Surveys (CSUR)*, 31 264-323.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem - A systematic study. *Intelligent Data Analysis,* 6, 429-450.

Jeong, A. (2003). The sequential analysis of group interaction and critical thinking in online threaded discussions. *American Journal of Distance Education,* 17 (1), 25-43.

Jeong, A. (2005). A guide to analyzing message-response sequences and group interaction patterns in computer-mediated communication. *Distance Education*, 26(3), 367-383.

Jeong, A. (2006). The effects of conversational styles of communication on group interaction patterns and argumentation in online discussions. *Instructional Science, 34*(5), 367-397.

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In C. Nedellec & C. Rouveirol (Eds.) *Proceedings of the 10th European Conference on Machine Learning (ECML-98)* (pp. 137-142). Berlin: Springer.

Kay, J., Maisonneuve, N., Yacef, K., & Zaïane, O. (2006). Mining patterns of events in students' teamwork data. In C. Heiner, R. Baker and K. Yacef (Eds.) *Proceedings of the Workshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems* (ITS 2006) (pp. 45–52). http://www.educationaldatamining.org/ITS2006EDM/EDMITS2006.html

Kim, J., Shaw, E., Ravi, S., Tavano, E., Arromratana, A., & Sarda, P. (2008). Scaffolding on-line discussions with past discussions: An analysis and pilot study of PedaBot. In B. Woolf, E. Aimeur, R. Nkambou, S. Lajoie (Eds.) *Proceedings of the 9th International Conference on Intelligent Tutoring Systems* (ITS-08) (pp. 343-352). Berlin: Springer.

Klein, P., Tirthapura, S., Sharvit, D., & Kimia, B. (2000) A tree-edit-distance algorithm for comparing simple, closed shapes. In D. Schmoys (Ed.) *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 696-704). San Francisco: Society for Industrial and Applied Mathematics

Kollar, I., Fischer, F., & Hesse, F. W. (2003). Cooperation scripts for computer-supported collaborative learning. In B. Wasson, R. Baggetun, H.U. Hoppe & S. Ludwigsen (Eds.) *Proceedings of the Computer Support for Collaborative Learning* (pp. 59-61). Bergen, Norway: InterMedia, University of Bergen.

Kolodner, J. (1993). *Case-Based Reasoning*. San Francisco: Morgan Kaufmann Publishers.

Krippendorff, K. (1980). *Content Analysis: An Introduction to its Methodology*. Thousand Oaks, CA: Sage Publications.

Kumar, R., Rosé, C.P., Wang, Y.-C., Joshi, M., & Robinson, A. (2007). Tutorial dialogue as adaptive collaborative learning support. In R. Luckin, K. R. Koedinger, & J. Greer (Eds.) *Proceedings of the 13th International Conference on Artificial Intelligence in Education* (AIED 2007) (pp. 383-390). Amsterdam: IOS Press.

Landis J.R., & Koch G.G. (1977). The measurement of observer agreement for categorical data. *Biometrics,* 33, 159-174.

Lingnau, A., Harrer, A., Kuhn, M., & Hoppe, H.U. (2007). Empowering teachers to evolve media enriched classroom scenarios. *Research and Practice in Technology Enhanced Learning,* 2, 105-129.

McAlister, S., Ravenscroft, A., & Scanlon, E. (2004). Combining interaction and context design to support collaborative argumentation using a tool for synchronous cmc. *Journal of Computer Assisted Learning,* 20 (3), 194–204.

McCallum, A., & Nigam K. (1998). A comparison of event models for Naïve Bayes text classification. In M. Sahami, M. Craven, T. Joachims, & A. McCallum (Eds.) AAAI-98 *Workshop on Learning for Text Categorization*, (pp. 41–48). Menlo Park, CA: AAAI Press. Also technical report WS-98-05, Carnegie Mellon University, Pittsburgh, USA.

McLaren, B.M. (2003). Extensionally defining principles and cases in ethics: An AI model. *Artificial Intelligence*, 150, 145-181.

McLaren, B.M., & Ashley, K.D. (2001). Helping a CBR program know what it knows. In D.W. Aha and I. Watson (Eds.) *Proceedings of the Fourth International Conference on Case-Based Reasoning* (ICCBR-01) (pp. 377-391). Berlin: Springer-Verlag.

McLaren, B.M., Scheuer, O., De Laat, M., Hever, R., De Groot, R., & Rosé, C.P. (2007). Using machine learning techniques to analyze and support mediation of student e-discussions. In R. Luckin, K.R. Koedinger, & J. Greer J. (Eds.) *Proceedings of the 13th International Conference on Artificial Intelligence in Education* (AIED 2007) (pp. 141-147). Amsterdam: IOS Press.

McLaren, B.M., Wegerif, R., Mikšátko, J., Scheuer, O., Chamrada, M., & Mansour, N. (2009). Are your students working creatively together? Automatically recognizing creative turns in student e-discussions. In V. Dimitrova, R. Mizoguchi, B. du Boulay, & A. Graesser (Eds.) *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (AIED-09), (pp. 317-324). Amsterdam : IOS Press.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). YALE: Rapid prototyping for complex data mining tasks. In M. Craven & D. Gunopulos (Eds.) *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD 2006) (pp. 935-940). New York: ACM Press.

Mikšátko, J. (2007). Using machine learning techniques to analyze and recognize complex patterns of student e-discussions (M.Sc. Thesis). Charles University, Prague.

Mikšátko, J., & McLaren, B.M. (2008). What's in a cluster? Automatically detecting interesting interactions in student e-discussions. In B. Woolf, E. Aimeur, R. Nkambou, & S. Lajoie (Eds.), *Proceedings of the 9th International Conference on Intelligent Tutoring Systems* (pp. 333-342). Berlin: Springer.

O'Donnell, A.M. (1999). Structuring dyadic interaction through scripted cooperation. In A. M. O'Donnell & A. King (Eds.) *Cognitive Perspectives on Peer Learning* (pp. 179-196). Mahwah, NJ: Erlbaum.

Pinkwart, N. (2003). A plug-in architecture for graph based collaborative modelling systems. In H.U. Hoppe, F. Verdejo & J. Kay (Eds.) *Proceedings of the 11th International Conference on Artificial Intelligence in Education* (AIED 2003) (pp. 535-536). Amsterdam: IOS Press.

Pinkwart, N., Ashley, K.D., Lynch, C., & Aleven, V. (2008). Graph grammars: An ITS technology for diagram representations. In H. Chad Lane, & D. Wilson (Eds.) *Proceedings of the 21st International FLAIRS conference* (pp. 433-438). Menlo Park, CA: AAAI Press.

Ravi, S., & Kim, J. (2007). Profiling student interactions in threaded discussions with speech act classifiers. In R. Luckin, K.R. Koedinger, & J. Greer (Eds.) *Proceedings of the 13th International Conference on Artificial Intelligence in Education* (AIED 2007) (pp. 357-364). Amsterdam: IOS Press.

Rosé, C., Wang, Y.-C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., & Fischer, F. (2008). Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in CSCL. *International Journal of Computer-Supported Collaborative Learning*, 3(3), 237-271.

Salmon, G. (2004). *E-moderating: The Key to Teaching and Learning Online* (2nd Ed.). London: Routledge Falmer.

Salomon, G., & Globerson, T. (1989). When teams do not function the way they ought to. *International Journal of Educational Research*, 13, 89-100.

Schapire, R.E., & Singer, Y. (2000). BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39, 135-168.

Scheuer, O., Loll, F., Pinkwart, N., & McLaren, B.M. (2010). Computer-supported argumentation: A review of the state of the art. *International Journal of Computer-Supported Collaborative Learning.* 5(1), 43-102.

Scheuer, O., & McLaren, B.M. (2008). Helping teachers handle the flood of data in online student discussions. In B. Woolf, E. Aimeur, R. Nkambou, & S. Lajoie (Eds) *Proceedings of the 9th International Conference on Intelligent Tutoring Systems* (ITS-08) (pp. 323-332). Berlin: Springer.

Schwarz, B.B., & Asterhan, C.S.C. (in press). E-moderation of synchronous discussions in educational settings: A nascent practice. To appear in *The Journal of the Learning Sciences*.

Schwarz, B.B., & De Groot, R. (2007). Argumentation in a changing world. *International Journal of Computer-Supported Collaborative Learning*, 2, 297-313.

Schwarz, B.B., & Glassner, A. (2007). The role of floor control and of ontology in argumentative activities with discussion-based tools. *International Journal of Computer-Supported Collaborative Learning*, 2(4), 449–478.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34, 1-47.

Soller, A. (2004). Computational modeling and analysis of knowledge sharing in collaborative distance learning. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, 14, 351-381.

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In P.M.G. Apers, M. Bouzeghoub, & G. Gardarin (Eds.): *Advances in Database Technology - EDBT'96, Proceedings of the Fifth International Conference on Extending Database Technology*, Lecture Notes in Computer Science 1057 (pp. 3-17). Berlin: Springer.

Suthers, D.D., Connelly, J., Lesgold, A., Paolucci, M., Toth, E.E., Toth, J., & Weiner, A. (2001). Representational and advisory guidance for students learning scientific inquiry. In K. D. Forbus, & P. J. Feltovich (Eds.) *Smart Machines in Education: The Coming Revolution in Educational Technology* (pp. 7-35). Menlo Park, CA: AAAI/MIT Press.

Tsai, W.H., & Fu, K.S. (1979). Error-correcting isomorphisms of attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 9, 757-768.

Vizcaíno, A. (2005). A simulated student can improve collaborative learning. *International Journal of Artificial Intelligence in Education*, 15, 3-40.

Walker, E., McLaren, B.M., Rummel, N., & Koedinger, K.R. (2007). Who says three's a crowd? Using a Cognitive Tutor to support peer tutoring. In R. Luckin, K.R. Koedinger, & J. Greer (Eds.) *Proceedings of the 13th International Conference on Artificial Intelligence in Education* (AIED 2007) (pp. 399-406). Amsterdam: IOS Press.

Wegerif, R. (2006). A dialogic understanding of the relationship between CSCL and teaching thinking skills. *International Journal of Computer Supported Collaborative Learning*, 1(1), 143-157.

Wegerif, R. (2007). *Dialogic, Education and Technology: Expanding the Space of Learning*. New York, NY: Kluwer-Springer.

Weinberger, A., Ertl, B., Fischer, F., & Mandl, H. (2005). Epistemic and social scripts in computer-supported collaborative learning. *Instructional Science*, 33(1), 1-30.

Weiss, G.M. (2004). Mining with rarity: A unifying framework. *SIGKDD Explorations*, 6, 7-19.

Wichmann, A., Giemza, A., Krauß, M., & Hoppe, H.U. (2009). Effects of awareness support on moderating multiple parallel e-discussions. In C. O'Malley, D. Suthers D., P. Reimann, & A. Dimitracopoulou (Eds.) *Computer Supported Collaborative Learning Practices: Proceedings of the 9th International Conference* *on Computer Supported Collaborative Learning*(pp. 646-650). International Society of the Learning Sciences, Inc.

Witten, I.H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques (2^nd^ Ed.).* San Francisco: Morgan Kaufmann.

Wong, A.K.C., You, M., & Chan, S.C. (1990). An algorithm for graph optimal monomorphism. *IEEE Transactions on Systems, Man and Cyberne*tics, 20, 628-638.

Zloof, M.M. (1977). Query-by-example: A data base language. *IBM Systems Journal*, 16, 324-343.