Supporting CSCL with Automatic Corpus Analysis Technology

Pinar Doenmez, Carolyn Rose, Karsten Stegmann, Armin Weinberger, Frank Fischer

Supporting CSCL with Automatic Corpus Analysis Technology

Pinar Dönmez, Carolyn Rosé
Language Technologies Institute
Carnegie Mellon University
pinard,cprose@cs.cmu.edu

Karsten Stegmann
Department for Applied Cognitive Psychology and Media Psychology, University of Tuebingen
k.stegmann @iwm-kmrc.de

Armin Weinberger, Frank Fischer
Knowledge Media Research Center Tuebingen
a.weinberger, f.fischer@iwm-kmrc.de

Abstract. Process analyses are becoming more and more standard in research on computer-supported collaborative learning. This paper presents the rational as well as results of an evaluation of a tool called TagHelper, designed for streamlining the process of multi-dimensional analysis of the collaborative learning process. In comparison with a hand-coded corpus coded with a 7 dimensional coding scheme, TagHelper is able to achieve an acceptable level of agreement (Cohen's Kappa of .7 or more) along 6 out of 7 of the dimensions when we commit only to the portion of the corpus where the predictor has the highest certainty. In 5 of those cases, the percentage of the corpus where the predictor is confident enough to commit a code is at least 88% of the corpus. Consequences for theory-building with respect to automatic corpus analysis are formulated. Potential applications as a support tool for process analyses, as real-time support for facilitators of on-line discussions, and for the development of more adaptive instructional support for computer-supported collaboration are discussed.

Keywords: Corpus analysis, automatic text processing techniques, argumentation

PROBLEM BACKGROUND

Increasingly, research in CSCL addresses quantitative process analysis through multi-dimensional coding schemes (e.g., Fischer, Bruhn, Gräsel, & Mandl, 2002; Lally & De Laat, 2002). The process of collaboration is seen as a mediator between the computer-supported instructional settings and cognitive processes. Often only detailed process analyses reveal plausible interpretations of the effects of CSCL environments (Weinberger, 2003). Conducting detailed process analyses involves applying categorical coding schemes along multiple dimensions, each of which indicate something different about the text segment's function within the collaborative discourse. For example, Lally and De Laat (2002) code for activities along six dimensions including cognitive, meta-cognitive, affective, design, discourse maintenance, and direct instruction. Multi-dimensional coding schemes like these encode much more information than frameworks in which each text segment is coded with a single category. However, while single dimensional analyses can be expedited by requiring participants to select contribution openers that are indicative of contribution function, this is not practical with multi-dimensional coding. Furthermore, applying multi-dimensional categorical coding schemes by hand is extremely time intensive for three reasons. First, developing the coding schemes themselves in such a way that human coders can apply them reliably is a lengthy process requiring much iteration. Second, sophisticated coding schemes may require a high skill level and intensive training before coders can apply a well-designed coding scheme with high reliability. Thus, training time for learning a new coding scheme is another source of time expense involved in this type of research. Finally, applying coding schemes as part of the analysis process itself is a tedious and time consuming process. Surprisingly, although structured editors often support this work, other times it is done by pen and paper. We therefore conducted a study to find out the degree to which automatic classification technology can be successfully used to automate the challenging task of multi-dimensional quantitative process analysis.

In this paper we present results of an evaluation study of the TagHelper technology for supporting and streamlining the process of multi-dimensional analysis of the collaborative learning process. We begin by contextualizing our technological explorations within a high profile CSCL environment. We then review related work and explain how our work is unique and complementary to previous automatic analysis work within the CSCL community. We then describe our exploration process and the details of our evaluation. We conclude with discussion and current directions.

MOTIVATION

The main question addressed in this paper is the extent to which automatic classification technology can be used to automate the task of multi-dimensional quantitative process analysis. Addressing this question, we first present a promising approach to this challenging task - TagHelper technology. Then we report on major results of an evaluation study of TagHelper in the context of a high profile CSCL project. In this project, a multi-dimensional coding scheme is applied to massive amounts of discourse data in order to examine the process of collaboration under different instructional conditions.

Within the context of this project, a series of experimental studies were conducted that aimed to address the question of how computer-supported collaboration scripts could foster argumentative knowledge construction in online discussions. Argumentative knowledge construction is based on the perspective of cognitive elaboration, the idea that learners acquire knowledge through argumentation with one or more learning partners (Baker, 2003; Dillenbourg, 2004). Computer-supported collaboration scripts apply on specific dimensions of argumentative knowledge construction, e.g., a script for argument construction could support learners to ground and warrant their claims (Kollar, Fischer, & Hesse, 2003; Stegmann, Weinberger, Fischer, & Mandl, 2004) or a social collaboration script can support conflict orientation (Weinberger, 2003). These and other computer-supported collaboration scripts were varied experimentally (see Stegmann et al., 2004; Weinberger, 2003; Weinberger, Fischer, & Mandl, submitted for more detailed process analyses). These studies were conducted in three waves. The first wave took place in the winter of 2000/2001, the second in the winter of 2002/2003, and the third in the winter of 2003/2004. The complete process analysis comprises about 200 discussions of about 600 participants with altogether more than 17,000 coded text segments. Trained coders categorized each segment using a multi-dimensional coding scheme (see below).

Three groups of about six coders, one group for each wave, were trained to apply the coding scheme to the collected corpus. One and the same trainer advised the analysts during all of the three waves. Each coder received a booklet with a detailed description of the coding scheme including all coding rules and examples for each category to ensure coding reliability. The training consisted of a combination of group meetings, dyadic practice, and individual practice. At regular intervals the reliability of the coding was computed by means of Cohen's Kappa. Discrepancies were then discussed and resolved. *Between the training and the coding itself, one quarter of the total duration of the research project was used for the coding of collaborative processes.* In particular, the training for each group of coders requires about several weeks, or about 500 working hours completely dedicated to the training process. The coding itself took about one month per wave, or about 1200 working hours.

Obviously a fully-automatic or even semi-automatic system, which could support coding of natural language corpus data, e.g., from computer-supported text-based communication, would facilitate and potentially improve quantitative process analyses in multiple ways. First of all, the number of working hours could be dramatically reduced for both training and coding. The role of the analysts could be reduced to simply checking the automatic coding and making corrections if necessary. Thus, the level of expertise of the coders could potentially be reduced, which would further reduce the cost. The coding itself would be faster. As learning processes could be analyzed promptly, even on the fly, facilitators could quickly identify specific deficits of collaborative learners as they are interacting and offer specific instructional support at key points.

OVERVIEW OF EXISTING TECHNOLOGY

Richards (1999), Soller & Lesgold (2000) and Goodman et al. (to appear) present work on automatically modeling the process of collaborative learning by detecting sequences of speech acts that indicate either success or failure in the collaborative process. The automatic analysis presented in this previous CSCL work builds upon an already completed categorical analysis of the text. These analyses can be thought of as meta-analyses with respect to the type of analysis we speak of. In contrast, the analysis that we present in this paper is based on the raw text contributed by the participants in the collaborative learning scenarios. What is different about our approach is that we start with the raw text and detect features within the text itself that are diagnostic of different local aspects of the collaboration. Thus, rather than presenting a competing approach, we present an approach that is complementary to that presented in prior work.

Currently there is a wide range of corpus analysis tools used to support corpus analysis work either at a very low level (e.g., word frequency statistics, collocational analyses, etc.) or at a high level (e.g., exploratory sequential data analysis once a corpus has been coded with a categorical coding scheme), but no tools to support the time consuming task of doing the categorical behavioral coding or content analysis, although much applicable technology developed in the language technologies community is already in existence. Content analysis includes both categorical analyses as well as more detailed, bottom-up analyses where spontaneous, informal observations about verbal behavior are recorded. In this paper we address the problem of streamlining the categorical type of protocol analysis.

	Components	Existing Technology	Existing Tools
Low Level Analysis	Word frequencies, word counting, morphsyntactic processing, collocation analysis	Tokenizers, Morphologicial analyzers, some shallow syntactic parsing	CLAN, SHAPA, TraSA, MultiTool, etc.
Medium Level Analysis	Various sentence level and segment level labeling tasks	LSA, CarmelTC, various dialogue act tagging approaches	
High Level Analysis	Exploratory sequential data analysis, Educational Data Mining	Statistics, data base queries	CLAN, SHAPA, ELAG, etc.

Figure 1. Abbreviated overview of some existing corpus analysis tools and technology

Currently, the only existing tools to support categorical content analysis are structured editors similar to Nb (Flammia & Zue, 1995) and MATE (McKelvie et al., 2000) or a wide variety of XML editors. We are exploring the application of state-of-the-art dialogue act tagging and text classification technology to enable fully and semi-automatic coding.

Applying Language Technology to a Previously Unexplored Application

Applying a categorical coding scheme can be thought of as a text classification problem where a computer decides which code to assign to a text based on a model that it has built based on regularities found from examining "training examples" that were coded by hand and provided to it. A number of such statistical classification and machine learning techniques have been applied to text categorization, including regression models (Yang & Pedersen, 1997), nearest neighbor classifiers (Yang & Pedersen, 1997), decision trees (Lewis & Ringuette), Bayesian classifiers (Dumais et al., 1998), Support Vector Machines (Joachims, 1998), rule learning algorithms (Cohen & Singer, 1996), relevance feedback (Rocchio, 1971), voted classification (Weiss et al., 1999), and neural networks (Wiener et al., 1993). While these approaches are different in many technical respects that are beyond the scope of this paper to describe, they are all used in the same way. A wide range of such machine learning algorithms are available in the Minorthird text-learning toolkit (Cohen et al, 2004), which we use as a resource for the work reported here. Minorthird is a software package that includes a wide range of configurable machine learning algorithms that can be used for text classification experimentation.

Within the computational linguistics community, a very common type of categorical coding scheme applied to text is that of speech acts or dialogue acts (Chu-Caroll, 1998; Reithinger & Klessen, 1997). Classifying spoken utterances into dialogue acts or speech acts has been a common way of characterizing utterance function since the 1960s. We argue that the same basic technology has the potential to achieve a much broader impact by becoming more accessible outside the computational linguistics community as well as using a broader range of coding schemes. One example of a community where this technology could have a major impact is the CSCL research community where large quantities of natural language data are being collected and analyzed painstakingly by hand.

Unfortunately, existing text classification technology is largely inaccessible to CSCL researchers who need and want semi-automatic tagging support because they do not have the background to apply it effectively to their analysis tasks. They are largely unaware of the wide range of alternative text classification techniques that are available, and furthermore, they do not possess the technical skills required to predict which available approaches are likely to be most appropriate for their task or to tune an appropriate technique once selected.

Bridging the Gap Between Language Technology and CSCL Research

The goal of our current work is to bridge the gap found in existing corpus analysis tools used by CSCL researchers for analyzing corpus data. In this paper we focus on the highly accurate text classification technology that enables some categorical corpus analysis work to be done totally automatically. In other work we have developed and tested an easy-to-use adaptive coding interface (Rosé et al., submitted). The easy-to-use TagHelper interface displays its automatic predictions about the analysis of each span of text to the analyst in the form of an adaptive menu-based interface. The system's predictions are visible to the analyst as he scans the page and modifies only the codes that he disagrees with by making an alternative selection.

Rosé et al. (submitted) have evaluated TagHelper's novel adaptive interface for facilitating content analysis of corpus data in comparison with an otherwise identical non-adaptive interface in terms of speed, validity, and reliability of coding. Since deciding to disagree with a predicted code and then choosing a new code takes longer than selecting a code from scratch, the advantage in coding speed for automatic predictions depends upon the accuracy with which predictions can be made. In order to break even with speed, a prediction accuracy of at least 50% is required. 50% prediction accuracy leads to an increase in reliability and validity of coding. In an evaluation with novice analysts in (Rosé et al., submitted), the top 30% of novice coders working with the automatic predictions achieved an average pairwise Kappa agreement measure of .71 in comparison with .54 in the unsupported coding condition ($P < .05$). Novice agreement with a gold standard was marginally higher ($P < .1$) across the whole population of coders. A gold standard corpus is a corpus that has been coded with a coding scheme, and the codes have been verified to be reliable. Thus, using automatic coding support, acceptable reliability and validity of coding can be achieved with novice coders using very little training. TagHelper can be quickly adapted for a new coding scheme and domain by providing only a small corpus of example texts encoded in XML and a simple specification of the structure of the coding scheme.

METHOD

In this paper, we examine the feasibility of TagHelper for supporting fully automatic analyses of the processes of argumentative collaborative knowledge construction. In this work, a human was required to optimize the selection and tuning of an appropriate machine learning algorithm. However, once a model was trained on the data using the selected technique, TagHelper was used to code data in a fully-automatic way.

Coding scheme for argumentative knowledge construction

In this section we describe a coding scheme that was applied in a project with more than 600 students of Educational Science at the Ludwig-Maximillians university of Munich, who participated in groups of three in multiple studies. Students in all experimental conditions had to work together in applying theoretical concepts to three case problems and jointly prepare an analysis for each case by communicating via web-based discussion boards. They were asked to discuss the three cases against the background of attribution theory (Weiner, 1985) and to jointly compose at least one final analysis for each case, i.e. they usually drafted initial analyses, discussed them, and wrote a final analysis. The cases portrayed typical attribution problems of university students, e.g., a student interpreting his failure on an important test. All groups collaborated in three discussion boards – one for each case. The discussion boards provided a main page with an overview of all message headers, which were graphically represented in a discussion thread structure. Learners could read the full text of all messages, reply to the messages, or compose and post new messages. In the replies, the original messages were quoted with ">" as in standard newsreaders and e-mail programs.

The purpose of our analysis was to model the process of argumentative knowledge construction. Argumentative knowledge construction must be evaluated on multiple process dimensions (Weinberger & Fischer, in press). These dimensions are derived from different theoretical approaches and focus on different concepts of argumentative knowledge construction. The main concepts are (1) epistemic activity, formal quality of argumentation, which includes (2) microlevel and (3) macrolevel, and (4) social modes of interaction (with a sub-dimension for (5) reaction). In accordance with the theoretical approach, the number of categories differs between dimensions from 2 (e.g., reaction) to 35 (e.g., epistemic). For experimental reasons, there is also a (6) treatment check dimension and a (7) quoted dimension.

On the *(1) epistemic dimension* (see table 1), argumentative knowledge construction processes are to be analyzed with respect to the questions of how learners work on the learning task, e.g., what content they are referring to or applying. One important distinction on the epistemic process dimension is to what extent learners work on the task or digress off task (Cohen, 1994). In order to solve a problem, learners may need to construct a problem space, construct a conceptual space, and construct relations between the conceptual and problem spaces. With the *construction of the problem space*, learners are to acquire an understanding of the problem they are supposed to work on. Therefore, learners select and relate individual components of the problem case information. The *construction of the conceptual space* serves to communicate an understanding of a theory. Learners connect individual theoretical concepts or distinguish them from another. The *construction of relations between conceptual and problem space* indicates to what extent learners are able to apply theoretical concepts adequately. In particular, learners may apply theoretical concepts that are to be learned, apply concepts stemming from prior knowledge or also apply wrong concepts.

On the formal dimension of argumentation, the processes of argumentative knowledge construction can be examined on both a micro- and a macrolevel of representation that indicate how learners construct single arguments and how learners connect arguments into sequences. In contrast to the epistemic dimension, the formal dimension of argumentative knowledge construction is not as concerned with what learners are contributing, but how they construct arguments and argumentation sequences in order to make their point.

Table 1: Categories of epistemic dimension of argumentative knowledge construction

Category	**Description**
Construction of problem space	Retelling or rephrasing of the problem that the learners work on. Learners relate case information to case information. Aims to foster understanding of particularities of the problem.
Construction of conceptual space	Retelling or rephrasing the theory learners are supposed to apply. Learners relate theoretical concepts and explain theoretical principles to foster understanding of a theory.
Construction of adequate relations between conceptual and problem space	Applying the relevant theoretical concepts adequately to solve a problem. Learners relate theoretical concepts to case information. A number of concept-case-relations may need to be constructed to adequately solve a complex problem (ca. 30 concept-case-relations for each case problem of the Munich study)
Construction of inadequate relations between conceptual and problem space	Applying theoretical concepts inadequately to the case problem. Learners may select the wrong concepts or may not apply the concepts according to the principles of the given theory.
Construction of relations between prior knowledge and problem space	Applying concepts that stem from prior knowledge rather than the new theoretical concepts that are to be learned.
Non-epistemic activities	Digressing off-topic.

On the (2) *microlevel*, an individual argument consists of a claim, which can be grounded with a warrant and/or specified by a qualifier (Toulmin, 1958; Toulmin, Rieke, & Janik, 1984). The warrant contains a justification for the claim based on grounds. The qualifier limits the validity of the statement and can be sometimes represented implicitly in the structure of an argument, e.g., indicated by "perhaps". We regard the frequent use of warrants and qualifiers in an argument as an indicator for high argumentative skill (see table 2).

On the (3) *macrolevel*, argumentation sequences can be examined with respect to how learners connect single arguments and create an argumentation pattern together (Leitão, 2000). The analysis typically focuses on the rhetorical function of individual expressions in a sequence of contributions. Central concepts are argument, counterargument and reply/integration (see table 3).

Table 2: Categories of microlevel of formal dimension of argumentative knowledge construction

Category	**Explanation**
Simple claim	Expressing a claim without qualifying the claim or providing grounds that warrant the claim.
Qualified claim	Expressing a claim without giving grounds, but limiting the validity of the claim (with qualifier).
Grounded claim	Explaining a claim without limiting its validity, but providing grounds that warrant the claim.
Grounded and qualified claim	Expressing a claim and grounds that warrant the claim as well as limiting the validity of the claim.

Table 3: Categories of macrolevel of formal dimension of argumentative knowledge construction

Category	**Description**
Argument	Statement put forward in favor of a specific proposition.
Counterargument	An argument opposing a preceding argument, favoring an opposite proposition.
Integration (reply)	Statement that aims to balance a preceding argument and counterargument.
Question (non argumentative)	Seeking information.
Planning (non argumentative)	Coordinating technical moves within the CSCL environment..
Evaluation (non argumentative)	Assessing the value of arguments or the group work.

The *(4) social modes dimension* (see table 4) indicates to what degree or in what ways learners refer to the contributions of their learning partners. On this dimension, a number of social modes of co-construction and their

relations to individual knowledge construction have been identified (Fischer et al., 2002). Learners may explicate their knowledge, e.g., by contributing a new analysis of a problem case. *Externalizations* are discourse moves that neither refer to preceding contributions of peers nor aim to elicit information from the learning partners. Learners may use the learning partner as resource and seek information (*elicitation*) in discourse from the learning partners in order to solve a problem case. Learners need to build at least a minimum consensus regarding the learning task in a process of negotiation in order to improve collaboration (Clark & Brennan, 1991). There are different styles of reaching consensus, however. *Quick consensus building* means that learners accept the contributions of their learning partners not in terms of taking over his or her perspective, but in order to be able to continue the discourse (Clark & Brennan, 1991). Recent approaches towards collaborative learning stress that collaborative learners may eventually establish and maintain shared conceptions of a subject matter (*integration-oriented consensus building*). Learners approximate and integrate each other's perspective, synthesize their ideas, and jointly try to make sense of a task (Nastasi & Clements, 1992). *Conflict-oriented consensus building* has been considered an important component in the socio-cognitive perspective upon collaborative learning (Doise & Mugny, 1984; Teasley, 1997). By facing a critique, learners may be pushed to test multiple perspectives or find more and better arguments for their positions (Chan, Burtis, & Bereiter, 1997).

In addition, any segment following an elicitation from another learning partner was coded on an explicit dichotomous *(5) sub-dimension of reaction* (no reaction vs. reaction). If a learner responded to an elicitation, e.g., by answering to a question, this response has been coded as reaction

Table 4: Categories of social modes dimension of argumentative knowledge construction (SOC)

Category	**Description**
Externalisation	Articulating thoughts to the group.
Elicitation	Questioning the learning partner or provoking a reaction from the learning partner.
Quick consensus building	Accepting the contributions of the learning partners in order to move on with the task.
Integration-oriented consensus building	Taking over, integrating and applying the perspectives of the learning partners.
Conflict-oriented consensus building	Disagreeing, modifying or replacing the perspectives of the learning partners.

The *(6) treatment check dimension* indicates how learners interact with the instructional design. The computer-supported collaboration script approach is often implemented with the help of prompts. These prompts support collaboration of learners and become part of the corpus data. This dimension considers how learners make use of prompts. Learners could *use the prompts in the intended manner*, e.g., write a counterargument when they are asked to write a counterargument. But learners could also *ignore the prompt*, i.e., write nothing in response to the prompt. If learners are prompted to write a counterargument but wrote an argument, it would be an *unintended use of prompt*. Obviously, this dimension could only be applied if prompts are part of the instruction. Prompts within the corpus data will be only analyzed on this single dimension.

Table 5: Categories of treatment check dimension of argumentative knowledge construction

Category	**Description**
Intended use of prompt	Reacting to this prompt like intended.
Ignoring prompt	Ignoring prompt. The action isn't connected with the prompt.
Unintended use of prompt	Using prompt, but not like intended.

The dichotomous *(7) quoted dimension* is a primary technical dimension (not quoted vs. quoted). As already mentioned before, in the replies, the original messages were quoted with ">" as in standard newsreaders and e-mail programs. Quoted text within the corpus data then will be only analyzed on this single dimension.

Experimental Process

We used the Minorthird text-learning toolkit (Cohen et al, 2004), which contains a large collection of configurable machine learning algorithms that can be applied to text classification tasks, as a framework in which to conduct our research. Because Minorthird includes a wide range of text classification algorithms that all operate over text coded in the same format, it is a convenient test environment for experimentation. We used as a gold-standard corpus as set of 1255 separate text segments coded with the multi-dimensional coding scheme described in the previous section. As described above, the coding scheme is composed of 7 dimensions, named epistemic, microlevel of argumentation, macrolevel of argumentation, social modes, reaction, treatment check, and quoted respectively. Each of these dimensions has a set of 2 or more categories associated with it. For example,

macrolevel of argumentation has 7 (six theoretical and one "rest" category) such categories, whereas microlevel of argumentation has 5 (four theoretical and one "rest" category), and epistemic has 35 (thirty-four theoretical and one "rest" category). The "rest" categories comprise prompts and quoted text. Every text segment in the gold standard corpus is labeled with a category for each of the 7 dimensions. Our experimentation followed a typical pattern for corpus based research, which we describe in this section. In other words, we form hypotheses about what might work based on our understanding of the coding scheme and our experience with the machine learning algorithms. We then run experiments with those algorithms and use the results to deepen our understanding of the representation and the interaction between the machine learning techniques and the data. We then revise our hypotheses and run additional experiments. We experimented with a range of techniques in a semi-directed manner. It is this semi-directed experimentation process that we are working towards automating in our continued research. We believe that if we could automate this process, we will have found the final piece of the puzzle that is required to make this technology fully accessible to CSCL researchers so that it could be applied to new problems without the aid of an experienced computational linguist.

We began our experimentation by testing a non-binary classifier called K-Nearest Neighbors to assign a category to each text for each of the seven dimensions. The difference between a binary classifier and a non-binary classifier is that binary classifiers can only distinguish between two categories (i.e., positive examples versus negative examples), a non-binary classifier can in theory make any number of distinctions (e.g., the 35 types of epistemological categories). Since the majority of the 7 dimensions that are part of our coding scheme contain more than two distinctions, a non-binary classifier was the most straightforward approach to use as a baseline. We tested this approach using what is called a cross-validation evaluation methodology. What this means is that we divided our gold-standard corpus into 10 equal subsets of coded spans of text. For each of these 10 subsets of data, we trained a model from the other nine subsets and tested on the selected subset so that we were always testing on a different set of data than what we trained on. Each of these rounds of training and testing are referred to as an iteration. So there are 10 iterations of training and testing for a 10-fold cross-validation evaluation such as this. This process is important for obtaining an accurate measure of how well a trained model will perform on additional data since it keeps a separation between data used for training the model and data used for testing the model. Once we had a measure of performance over each of the 10 subsets of data, we averaged those in order to obtain an estimate for the whole set. Cross-validation evaluations are standard practice in machine learning research. We went through this process separately for each of the 7 dimensions. The results are presented in Table 5. The non-binary classifier only achieved an acceptable level of agreement with the gold standard in the case of reaction, achieving a Kappa of .81.

Table 5: Performance of Non-binary classifier over data

Name of Dimension	**Number of Categories**	**Kappa**
epistemic	35	.51
microlevel of argumentation	4	.54
macrolevel of argumentation	7	.54
Social modes	21	.35
reaction	3	.81
Treatment check	4	0
Quoted	2	.63

To assess the learnability of each of the categories along the 7 dimensions, we then began to experiment with binary classifiers. There is a much wider range of non-binary classifiers to choose from. For each category along each dimension we computed a Kappa value for a wide range of binary classifiers, each of which was given the task if distinguishing example texts that are assigned the corresponding category along its associated dimension and those that are not. We noticed that some categories were much easier to predict than others. Normally, it was the categories for which there were more than 25 examples in the corpus. Thus, we hypothesized that an approach where we cascaded the binary classifiers so that we first applied the most accurate classifiers and then the less accurate classifiers only if the accurate ones did not predict a positive match would be more accurate.

Again we adopted a cross-validation methodology. This time it was necessary to select on each iteration of the 10-fold cross-validation evaluation, not only a testing set, but also a validation set on which to determine the rank ordering of the individual binary classifiers. This is so that the set used for rank ordering the binary classifiers is not either the same set that they were trained over, nor the same set they will be tested over. This ensures both optimal training and most accurate testing. Thus, on each iteration, we trained a separate binary classifier for each category associated with each dimension over 8 subsets of data. We then tested the accuracy of these classifiers on the validation set. For each dimension, we rank ordered the binary classifiers according to their accuracy over the validation set. We then applied them in rank order over the test set, selecting as an assigned code the first binary classifier that indicated a positive match for an example text. We computed the accuracy of the cascaded classifier over each of the 10 test sets using this approach and then averaged the

results as in the first experiment with non-binary classifiers. The assumption here is that if one classifier gives a higher Kappa value over the validation set, then it will most likely be more reliable in terms of predicting correct labels over the testing set, hence it is more probable that its prediction is correct instead of the classifier with a lower Kappa. The best results we obtained were with the Voted Perceptron Learning algorithm, which gives better results with our data in general than the other classification techniques such as DecisionTrees, NaiveBayes approach, SVM Learning, etc. In the next section we present our current best results.

OUTCOMES

Since the results for the reaction dimension were already acceptable with non-binary classification, we restricted our experimentation to the remaining 6 dimensions. In all cases we achieved a significant increase over the non-binary classification result except in the case of the epistemic dimension. We first present the Kappa we achieve over the whole corpus using the cascaded approach. We then present the Kappa we achieve if we use a more conservative approach, only assigning a category to the portion of the corpus where our performance over the validation set was highest. The task was accomplished by eliminating the least accurate binary classifiers from the cascaded model one by one until an acceptable Kappa was achieved. In that column we present the best Kappa we were able to achieve and the percentage of the corpus it was computed over. For example, for the macrolevel of argumentation we are able to achieve a Kappa of .83 over 92% of the corpus, leaving 8% of the corpus uncoded. In the case where this conservative classifier is used, a human coder only needs to code 8% of the corpus by hand since the accuracy over the automatically coded portion of the corpus is acceptable.

Table 7: The table compares the accuracy computed in terms of Cohen's Kappa between the gold standard codes and 3 approaches to automatic classification

Name of Dimension	Kappa for Non-binary Classification	Kappa for Cascaded Binary Classification Over Whole Set	Kappa for Cascaded Binary Classification Over Partial Set
Epistemic	.51	.49	.52 (43% of corpus)
Microlevel of argumentation	.54	.76	.83 (92% of corpus)
Macrolevel of argumentation	.54	.67	.7 (88% of corpus)
Social modes	.35	.55	.68 (50% of corpus), .75 (25% of corpus)
Treatment check	0	.73	.85 (97% of corpus)
Quoted	.63	.98	.98 (100% of corpus)

Although the knowledge that is brought to bear on the coding process for the 7 different dimensions has different requirements (for example, in terms of how much context is required or what the distinctions mean about the student's contribution), in all cases except the epistemic dimension the same procedure lead to a classifier that achieved a significantly higher level of agreement with the gold standard than the non-binary classifier. Thus, this evaluation demonstrates that the cascaded binary classifier has some generality.

We plan to continue experimenting with alternative classification approaches for the social modes and epistemic dimensions. Similar to our previous explorations where we clustered examples according to similarity of coding across the 7 dimensions of our coding scheme, we are now exploring the possibility of clustering the coded text segments according to similarity of vocabulary distributions within text segments. We predict that within clusters of similar texts, there will be a smaller number of categories for each dimension than over the whole set. Thus, we predict that training a classifier over just the examples within clusters will be more accurate.

DISCUSSION

We have presented and evaluated technology for streamlining the process of multi-dimensional analysis of the collaborative learning data. We have argued that such technology could potentially have a tremendous impact on this increasingly important part of CSCL research. Beyond this community a wide range of other behavioral researchers including social scientists, psychologists, and other learning scientists and education researchers collect, code, and analyze large quantities of natural language corpus data as an important part of their research.

One important outcome from this research is that even sophisticated coding schemes such as the 7 dimensional coding scheme discussed here that requires several weeks of intensive training for a human to apply reliably can be largely automated. 4 of the 7 dimensions (i.e., macrolevel of argumentation, reaction, treatment check, and quoted) can be applied fully automatically with an acceptable level of accuracy, as measured using a cross validation methodology over our gold standard coded corpus. Significant portions of the additional two

dimensions (microlevel of argumentation and social modes) can be applied fully automatically to a significant portion of the data, thus cutting down the number of examples that must be coded by a human (an 88% reduction in the case of microlevel of argumentation dimension and a 25% reduction in the case of social modes dimension). While the results with epistemic dimension were lower, and the Kappa value over the whole set of data was only .51, the percent agreement was 80% over the portion of the corpus that received a committed code. This is 30% higher than the break even point for time savings with checking and correcting automatically coded examples according to Rosé and colleagues (submitted). Thus, even with this level of accuracy, the automatic category predictions can lead to a significant reduction in coding time on the epistemic dimension.

Another important outcome from this research is that the cascaded binary classification approach, which we explore, has some generality across multiple dimensions of our coding scheme although they are quite different in terms of the types and numbers of distinctions that must be made. Thus, it is an approach that is likely to be reused successfully with other coding schemes and eventually be part of an eventual approach to automatic selection and tuning of machine learning approaches to applying categorical coding schemes.

Beyond improvements to the data analysis that is central to our process, automatic coding technology would also enable new kinds of instructional interventions. For example, automatic on-line analysis of chat interactions could provide instructors with the capability to monitor the progress of multiple interactions occurring in parallel, indicating where the instructor's intervention is most needed, and even what the specific needs are that should be addressed. Further ahead, a fully automatic system could also enable automatic adaptive interventions for collaborative learning. Those interventions would be more flexible/adaptive than current static interventions. For example, a collaboration script for argument construction could be strategically applied when learners do not ground and warrant their claims and it could be faded out carefully when learners develop internal cognitive scripts that guide their argumentative knowledge construction. Such a system could prevent effects like over-scripting (Dillenbourg, 2004) or negative interaction effects between scripts (Kollar & Fischer, 2004).

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation grant number SBE0354420 and the Deutsche Forschungsgemeinschaft and KALEIDOSCOPE - a European network of excellence.

REFERENCES

Baker, M. (2003). Computer-mediated argumentative interactions for the co-elaboration of scientific notions. In J. Andriessen, M. Baker & D. Suthers (Eds.), *Arguing to learn: confronting cognitions in computer-supported collaborative learning environments* (Vol. 1, pp. 1-25). Dordrecht: Kluwer Academic Publishers.

Chan, C. K. K., Burtis, P. J., & Bereiter, C. (1997). Knowledge building as a mediator of conflict in conceptual change. Cognition and Instruction, 15(1), 1-40.

Chu-Carroll, J. A statistical model for discourse act recognition in dialogue interactions. In Applying Machine Learning to Discourse Processing: 1998 AAAI Spring Symposium, 12-17.

Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In S. D. Teasley (Ed.), Perspectives on socially shared cognition (pp. 127-149). Washington: American Psychologist Association.

Cohen, W. (2004). Minorthird: Methods for Identifying Names and Ontological Relations in Text using Heuristics for Inducing Regularities from Data, http://minorthird.sourceforge.net.

Cohen, W. and Singer, Y. (1996). Context-sentsitive learning methods for text categorization, In SIGIR'96: Proc. 19th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 307-315.

Dillenbourg, P. (2004). "Split Where Interaction Should Happen", a model for designing CSCL scripts. In P. Gerjets, P. A. Kirschner, J. Elen & R. Joiner (Eds.), Instructional design for effective and enjoyable computer- supported learning. Proceedings of the first joint meeting of the EARLI SIGs Instructional Design and Learning and Instruction with Computers (CD-ROM). Tuebingen: Knowledge Media Reseach Center.

Doise, W., & Mugny, G. (1984). The Social Development of the Intellect. Oxford: Pergamon.

Dumais, S., Platt, J., Heckerman, D. and Sahami, M. (1998). Inductive Learning Algorithms and Representations for Text Categorization, Technical Report, Microsoft Research.

Fischer, F., Bruhn, J., Gräsel, C., & Mandl, H. (2002). Fostering collaborative knowledge construction with visualization tools. Learning and Instruction, 12, 213-232.

Flammia, G., Zue, V. Empirical evaluation of human performance and agreement in parsing discourse constituents in spoken dialogue. Proc. Eurospeech-95, 1965-1968.

Goodman, B., Linton, F., Gaimari, R., Hitzeman, J., Ross, H., & Zarrella, J. (to appear). Using Dialogue Features to Predict Trouble During Collaborative Learning, to appear in the Journal of User Modeling and User Adapted Interaction.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features, In Proc. 10th European Conference on Machine Learning (ECML), Springer Verlag, 1998.

Kollar, I., & Fischer, F. (2004). Internal and external cooperation scripts in web-based collaborative inquiry learning. In P. Gerjets, P. A. Kirschner, J. Elen & R. Joiner (Eds.), Instructional design for effective and enjoyable computer- supported learning. Proceedings of the first joint meeting of the EARLI SIGs Instructional Design and Learning and Instruction with Computers (CD-ROM). Tuebingen: Knowledge Media Research Center.

Kollar, I., Fischer, F., & Hesse, F. W. (submitted). Computer-supported cooperation scripts - a conceptual analysis.

Lally, Vic and De Laat, Maarten F. (2002) Cracking the code:Learning to collaborate and collaborating to learn in a networked environment. In, CSCL, University of Colorado, Boulder, USA, 7-11 January, 2002.

Leitão, S. (2000). The potential of argument in knowledge building. Human Development, 43, 332-360.

Lewis, D. and Ringuette, R. (1994). A Comparison of teo learning algorithms for text classification, In Third Annual Symposium on Document Analysis and Information Retrieval, pp. 81-93.

McKelvie, D., Isard, A., Mengel, A., Moller, M., Grosse, M., & Klein, M. The MATE Workbench – an annotation tool for XML coded speech corpora. Speech Communication, 33(1-2):97-112.

Nastasi, B. K., & Clements, D. H. (1992). Social-cognitive behaviors and higher-order thinking in educational computer environments. Learning and Instruction, 2, 215-238.

Reithinger, N., Klessen, M. Dialogue act classification using language models. Proc. EuroSpeech-97, 2235-2238.

Richards, L. (1999). Using NVivo in Qualitative Research. Bundoora, Victoria, Australia: Qualitative Solutions and Research.

Rocchio, J. (1971). Relevance feedback in information retrieval, In The SMART Retrival System: Experiments in Automatic Document Processing, pp. 313-323, Prentice Hall Inc.

Rosé, C. P., Gweon, G., Wittwer, J., Nueckles, M. (submitted). An Adaptive Interface that Facilitates Reliable Analysis of Corpus Data, submitted to INTERACT '05.

Soller, A. & Lesgold, A. (2000). Modeling the Process of Collaborative Learning, Proceedings of the International Workshop on New Technologies in Collaborative Learning, Awaiji-Yumebutai, Japan.

Stegmann, K., Weinberger, A., Fischer, F., & Mandl, H. (2004). Scripting Argumentation in computer-supported learning environments. In P. Gerjets, P. A. Kirschner, J. Elen & R. Joiner (Eds.), Instructional design for effective and enjoyable computer- supported learning. Proceedings of the first joint meeting of the EARLI SIGs Instructional Design and Learning and Instruction with Computers (CD-ROM) (pp. 320-330). Tuebingen: Knowledge Media Research Center.

Stegmann, K., Weinberger, A., Fischer, F., & Mandl, H. (2004, April). Can Computer-Supported Collaboration Scripts Facilitate Argumentative Knowledge Construction? Paper presented at the Annual Meeting of the American Educational Research Association 2004, San Diego.

Teasley, S. D. (1997). Talking about reasoning: How important is the peer in peer collaboration? In L. B. Resnick, R. Säljö, C. Pontecorvo & B. Burge (Eds.), Discourse, tools and reasoning: Essays on situated cognition (pp. 361-384). Berlin: Springer.

Toulmin, S. (1958). The uses of argument. Cambridge: Cambridge Universty Press.

Toulmin, S., Rieke, R., & Janik, A. (1984). An introduction to reasoning. New York: Macmillian Publishing.

Weinberger, A. (2003). Scripts for Computer-Supported Collaborative Learning Effects of social and epistemic cooperation scripts on collaborative knowledge construction, from http://edoc.ub.uni-muenchen.de/archive/00001120/01/Weinberger_Armin.pdf

Weinberger, A. & Fischer, F. (in press). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education*.

Weinberger, A., Fischer, F., & Mandl, H. (submitted). Collaboration scripts to facilitate knowledge convergence in computer-mediated learning environments.

Weinberger, A., Stegmann, K., Fischer, F., & Mandl, H. (in press). Problem-based collaborative knowledge construction online: Effects of multiple argumentative script components in text-based communication. In F. Fischer, H. Mandl, J. Haake & I. Kollar (Eds.), Scripting computer-supported communication of knowledge - cognitive, computational and educational perspectives.

Weiner, B. (1985). An attributional theory of achievement motivation and emotion. Psychological Review, 92, 548-573.

Wiener, E., Pedersen, J. and Weigend, S. (1993). A neural network approach to topic spotting, In Proc. 4th annual symposium on document analysis and information retrieval, pp. 22-34, 1993.

Weiss, S., Apte, C. and Damerau, F. (1999). Maximizing Text-Mining Performance, Proceedings of IEEE Intelligent Systems.

Yang, Y. and Pedersen, J. (1997). Feature selection in statistical learning of text categorization, In the 14th Int. Conf. on Machine Learning, pp 412-420.