A User Interface for the Exploration of Manually and Automatically Coded Scientific Reasoning and Argumentation

Patrick Lerner, Johannes Daxenberger, Lucie Flekova, and Iryna Gurevych
patricklerner@me.com, daxenberger@ukp.informatik.tu-darmstadt.de, flekova@ukp.informatik.tu-darmstadt.de, gurevych@ukp.informatik.tu-darmstadt.de
Ubiquitous Knowledge Processing Lab, Technische Universität Darmstadt

Andras Csanadi, Christian Ghanem, Ingo Kollar, and Frank Fischer
Andras.Csanadi@psy.lmu.de, christian.ghanem@psy.lmu.de, ingo.kollar@phil.uni-augsburg.de, frank.fischer@psy.lmu.de
Munich Center of the Learning Sciences, LMU Munich

Abstract: Scientific reasoning and argumentation (SRA) is a complex process. Thus, analyzing the quality of learners' SRA and presenting SRA outcomes are important research problems. This study attempts to account for these problems by developing a user interface that facilitates learning scientists to analyze SRA, enabling them to evaluate the performance of an automated coding algorithm and supporting them to recognize patterns in the coded data using flexible visualizations. The usability and the comprehensibility of the user interface have been tested by means of a user study in an online performance test. The results indicate that the user interface effectively supports the investigation of SRA.

Keywords: scientific reasoning, user interface, automated coding, visualization, epistemic activities

Introduction

Scientific reasoning and argumentation (SRA) is a complex process, both for learners to engage in and for teachers or researchers to evaluate SRA. For example, in psychological research, the hand-coding of verbal data and the interpretation of manually and automatically coded outcomes can be very demanding. Recent developments have given rise to computer-supported scientific reasoning assessments (e.g., Gobert et al., 2013). Automated text classification techniques developed by computer scientists in collaboration with domain experts may facilitate the process of discourse analysis (Rosé et al., 2008). However, it has yet to be fully understood how technology should present the outcomes of the automated analyses to domain experts. Suitable user interfaces (UIs) shall support reflection on the reliability of automated coding outcomes and to discover previously unnoticed patterns and relations within the discourse data (Suthers, 2001). The aim of this study is thus to develop a UI to aid scientists with no computer science background by fostering their understanding of and work with both machine and hand-coded process data.

The empirical work applied in this study builds on a novel framework of SRA (Fischer et al., 2014), describing it as a process of inquiry. This model identifies eight epistemic activities relevant to SRA: problem identification (PI), questioning (Q), hypothesis generation (HG), constructing and redesign of artefacts (CA), evidence generation (EG), evidence evaluation (EE), drawing conclusions (DC) and communicating and scrutinizing (CS). Previous studies developed a coding scheme to capture epistemic activities of SRA in two professional domains. Their findings in the domain of teacher education (Csanadi, Kollar, & Fischer, 2015) and social work (Ghanem, Pankofer, Kollar, Fischer, & Lawson, 2015) support the notion that the SRA framework may be fruitfully applied in practical domains. In both studies, the researchers realized the need for a tool that offers visual support for coding interpretation. Thus, a UI that provides support in visualizing patterns within manually and automatically coded data has been developed in the present study. An online demonstration of this UI is available at http://reason.ukp.informatik.tu-darmstadt.de:9000.

Key requirements for visualizing epistemic activities

To create a suitable UI, which helps learners and domain experts to visually analyze manually and automatically coded data, we formulated the following requirements in accordance with its target users:

R1. The UI must be accessible and usable without any complex installation.

R2. Terminology should be aimed at its target users, unfamiliar terms and visualizations must be explained.

R3. To support the analysis of SRA, it must be possible to visualize both manually and automatically coded data of a sequential nature.

R4. Users must be able to look at the information used by the automatic classification algorithm to understand its relation to the codes better.

R5. Users must be informed about the performance of the automatic classification for given codes.

Previously developed tools, that support researchers' understanding of results from automatically coded data (e.g., MINERVA; Stoffel, Flekova, Oelke, Gurevych, & Keim, 2015), exhibit limitations of usability due to their exclusive focus on users with expert knowledge in automatic coding. Other systems for visualizing text classification components (Heimerl, Jochim, Koch, & Ertl, 2012; Lamprecht, Hautli, Rohrdantz, & Bogel, 2013) and automated coding tools that enable text classification for large amounts of data (e.g., LightSide, Mayfield & Rosé, 2013) are not suited for sequential data. The present UI is different from the above tools because it fills a specific gap and targets learning sciences students and domain experts with little or no background in computer science.

Developing the UI

Step 1: Manual and automatic coding of SRA

The development of this UI relied on two datasets from earlier studies (Csanadi et al., 2015; Ghanem et al., 2015). These studies developed a segmentation and coding scheme on epistemic activities of SRA (Fischer et al., 2014) for professional problem solving. Think aloud protocols and discourse data were segmented and coded for further analysis. Their segmentation into propositional units reached reliable proportions of agreement between 80% and 85%. The coding scheme proved to be reliable at $\kappa = 0.69$ and $\kappa = 0.68$. Results for the automatic coding of SRA were gathered from Spari (2015). Here, DKPro TC (Daxenberger, Ferschke, Gurevych, & Zesch, 2014) was used to perform machine learning based on features (see Requirement 4). A wide range of features were extracted, e.g., average word length and number of pronouns for each segment. Codes were then automatically assigned based on the features and the sequence of segments.

Step 2: Design of the user interface

While existing tools focus on the selection of appropriate features such as n-grams and syntactical information for a given coding task, the present UI aims to support learners and domain experts in understanding a) how SRA (and related sequential datasets) is manually coded, and b) how an automatic algorithm codes the same data. This requires suitable and flexible visualizations for sequential data. The present UI addresses a) in three interfaces: a segment listing, a label distribution histogram, and a progression graph visualization. In the segment listing, all segments of a transcript are shown in chronological order, with both manual and automatically assigned codes. The label distribution intends to give a quick overview of the frequencies of different codes. The progression graph (cf. Fig. 1 left) deals with the sequential nature of SRA by displaying a graph that shows the frequency of each code for a transcript over time. To do so, the occurrences of all codes are computed for different, flexibly configurable time intervals of a text.

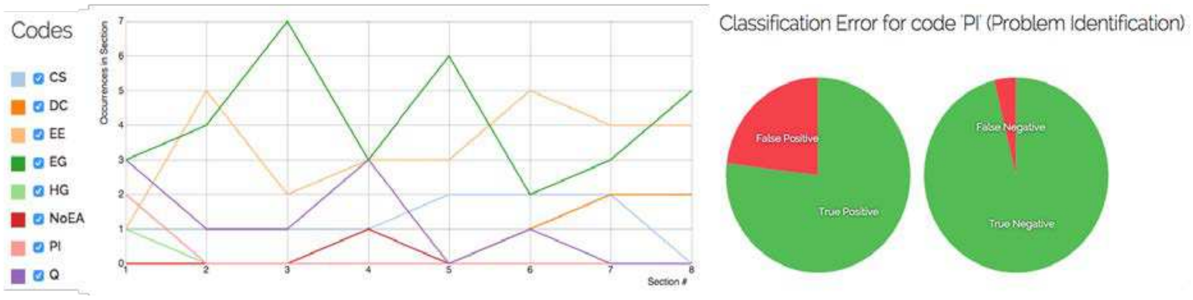

Figure 1. Parts of the UI: The progression graph (left) and classification error visualization (right).

Two further visualizations address b): feature value distribution and the classification of error types (cf. Fig. 1 right). The former shows the values for each of the features that have been used by the automatic coding algorithm to determine the code for a segment. Users can inspect the distribution of values for a given feature and code. The error classification visualization provides an insight into the reliability of the automatic coding by allowing for a comparison of the accuracy between them, as well as by listing absolute and percentage number of samples for each error type (e.g., false positives). In order to make visualizations accessible to users with little or no knowledge of automatic coding, simple representations have been preferred over more complex ones (Guyon & Elisseeff, 2003).

Method and design of the user performance test

An online performance test was developed to test the usability and comprehensibility of the UI. The sample (N = 20) for this test included members of an interdisciplinary research group dealing with SRA (n = 13), students of a Learning Sciences M.A. Program (n = 4), and postgraduates of a computer science laboratory (n = 3). All participants were students or employees at German universities.

The UI was demonstrated using two transcripts of think aloud protocols of pre-service teachers solving an authentic problem from a study conducted by Csanadi et al. (2015). The results of both automatic and manual coding were displayed in the UI. The test consisted of five sections, as follows:

Distribution of codes: We asked for the most frequent code in a given protocol, both for the automatic and the manual coding.

Domain-specific questions: We asked participants to decide whether a protocol indicates hypothetico-deductive or inductive reasoning.

Features: Participants had to compare the predictive power of three features for several codes.

Appraisal of automatically coded results: Using the classification error visualization, participants had to evaluate the accuracy of automatic coding. We asked for the highest number of 'True Positive' and 'False Positive' classifications, as well as for the highest accuracy.

Usability judgement: We asked participants to rate the usability of the UI in general and in particular, its usefulness for present/future research.

In addition to these five-point Likert-scale items, we asked participants to provide an overall feedback.

Findings

Table 1 shows the number of participants who answered correctly, incorrectly, or indicated an inability to answer, to each question type. For the domain-specific questions, participants were additionally asked to specify the visualization they used to find the answer. While the task was specifically designed towards the progression graph visualization and ten domain experts used it to solve the task, it could also be solved using the simple listing of segments with their codes, as the remaining three domain experts and two additional participants did. Only two participants incorrectly tried to solve these tasks using the distribution of codes, which was not possible as the task relied on the sequence of codes. On average, the participants rated the usefulness of the UI (1 = Not useful; 5 = Very useful) at 3.45 (domain experts: 3.23). The applicability of the UI to their present and future research was rated on average at 3.15 (domain experts: 3.08). The UI's ease of use (1 = Very hard/confusing; 5 = Very easy) was rated on average at 3.1 (domain experts: 3.15).

Table 1: Performance Test Outcomes: Number of Participants (Including Domain Experts)

		Participants (Domain experts)		
		Correct	Incorrect	Unclear
Evaluation of Descriptives	Distribution of codes	20 (13)	0 (0)	0 (0)
	Features	16 (11)	2 (0)	2 (2)
Domain-specific questions: Identify reasoning types	Identification of first text (Hypothetico-deductive)	16 (13)	1 (0)	3 (0)
	Identification of second text (Inductive)	10 (8)	2 (1)	8 (4)
Appraisal of automatically coded results	Compare accuracy of two codes	14 (9)	6 (2)	0 (0)
	Find correct statement about the performance of a feature	14 (9)	6 (4)	0 (0)

Conclusions and implications

Overall, this study demonstrates that the UI represents progress towards developing a comprehensive and effective tool for domain experts who are not overly familiar with computer science technology. R1 (easy accessibility) was addressed by implementing the system as an online interface that can be accessed on any modern web browser. R2 (user-friendliness) was met using "familiar" terminology and textual cues. The majority of users reported little to no difficulty understanding the interface or using it to solve the presented tasks, which was also shown by the high rate of correctly answered questions (cf. Table 1; for the easier questions, 100% correct answers were obtained). R3 (visualization of process data) was addressed by the segment listing, label distribution histogram, and progression graph interfaces. Here, the users experienced the most difficulty in a) identifying the reasoning type (this might be the result of a lack of knowledge in the domain of SRA) and b) understanding the performance of the automatic coding (possibly due to unfamiliarity with machine learning). R4 and R5 (feature value distribution and error classification visualization) were each addressed by their own interfaces. The average ratings from the related questions demonstrate that these visualizations tended to be more helpful than confusing. Inquiries about the generalizability of the UI in the research of the participants were also slightly more positive than negative. The same was recorded regarding the UI's ease of use. This feedback will be taken as encouragement to invest further effort in the development of the UI.

From a practical viewpoint, this tool is already able to support teachers when conducting spontaneous assessments of their students' SRA based on the visualization of the automatically coded texts. The UI might be further extended to provide automated feedback to students (e.g., about the type of reasoning they used), thus enabling both adaptive and adaptable support. Some limitations of the present study include the lack of a control group in the survey (i.e., a group of users who had to solve the same tasks/questions without the current UI or with another tool) or the inclusion of additional (potentially better) machine learning models, which might have increased the users' understanding of features and their meanings.

References

Chi, M. T. H. (2009). Active-Constructive-Interactive: A Conceptual Framework for Differentiating Learning Activities. *Topics in Cognitive Science, 1*(1), 73–105.

Csanadi, A., Kollar, I., & Fischer, F. (2015, August) *Internal scripts and social context as antecedents of teacher students' scientific reasoning*. Paper presented at the 16th Biennial Conference of the European Association for Research on Learning and Instruction (EARLI), Limassol, Cyprus.

Daxenberger, J., Ferschke, O., Gurevych, I., & Zesch, T. (2014, June). DKPro TC: A Java-based Framework for Supervised Learning Experiments on Textual Data. *Proceedings of the ACL*, Baltimore, MD, USA.

Fischer, F., Kollar, I., Ufer, S., Sodian, B., Hussmann, H., Pekrun, R.,. . . Eberle, J. (2014). Scientific Reasoning and Argumentation: Advancing an Interdisciplinary Research Agenda in Education. *Frontline Learning Research, 5*, 28–45.

Ghanem, C., Pankofer, S., Fischer, F., Kollar, I. & Lawson, T. R. (2015, April). *The Relation between Social Work Practice and Science - Analysis of Scientific Reasoning of Probation Officers and Social Work Students.* Paper presented at the European Conference for Social Work Research, Lubljana, Slovenia.

Gobert, J. D., Sao Pedro, M., Raziuddin, J., & Baker, R. S. (2013). From Log Files to Assessment Metrics: Measuring Students' Science Inquiry Skills Using Educational Data Mining. *Journal of the Learning Sciences*, *22*(4), 521–563.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*,1157–1182.

Heimerl, F., Jochim, C., Koch, S., & Ertl, T. (2012, December). FeatureForge: A Novel Tool for Visually Supported Feature Engineering and Corpus Revision. *Proceedings of COLING*, Mumbai, India.

Lamprecht, A., Hautli, A., Rohrdantz, C., & Bogel, T. (2013, August). A Visual Analytics System for Cluster Exploration. *Proceedings of the ACL*, Sofia, Bulgaria.

Mayfield, E., & Rosé, C. (2013). LightSIDE: Open Source Machine Learning for Text. In M. D. Shermis & J. Burstein (Eds.), *Handbook of automated essay evaluation*. Hoboken: Taylor and Francis.

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 computer-supported collaborative learning. *International Journal of Computer-Supported Collaborative Learning, 3*(3), 237-271.

Stoffel, F., Flekova, L., Oelke, D., Gurevych, I., & Keim, D. A. (2015, October). *Feature-Based Visual Exploration of Text Classification*. Symposium on Visualization in Data Science at IEEE VIS 2015.

Spari, M. (2015). *Automatic Text Classification for Recognizing Scientific Reasoning and Argumentation in German* (Unpublished bachelor's thesis). Technische Universität Darmstadt.

Suthers, D. D. (2001). Towards a Systematic Study of Representational Guidance for Collaborative Learning Discourse. *Journal of Universal Computer Science*, *7*(3), 254-277.

Acknowledgements

This research was supported by the Elite Network of Bavaria under Grant K-GS-2012-209, by the German Federal Ministry of Education and Research (BMBF) under the promotional reference 01UG1416B (CEDIFOR) and by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant № I/82806.