Supporting the Guide on the SIDE

Moonyoung Kang[1], Sourish Chaudhuri[1], Rohit Kumar[1], Yi-Chia Wang[1],
Eric R. Rosé, Yue Cui[1], Carolyn P. Rosé[1,2]

[1]Language Technologies Institute and [2]Human-Computer Interaction Institute,
Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213
{moonyoun,schaudhu,rohitk,yichiaw,cprose,ycui@cs.cmu.edu}

Abstract. We present SIDE[1] (the Summarization Integrated Development Environment), which is an infrastructure that facilitates the construction of reporting interfaces that support group learning facilitators in the task of getting a quick sense of the quality and effectiveness of a collaborative learning interaction. The SIDE framework offers flexibility in the specification of which conversational behavior to take note of as well as how noted behavior should be reported to instructors, making it a valuable research tool.

It is said that a facilitator of effective collaborative learning should be a "guide on the side" rather than a "sage on the stage". In classrooms, instructors roam around the room, listen to conversations, and jump in at key moments to offer guidance. In on-line settings, group learning facilitators are responsible for a larger number of groups. Listening in on conversations involves reading a high volume of text based chat, often paired with data streams in other modalities such as a digital whiteboard. The challenge is to enable group learning facilitators to quickly get a sense of the collaborative interaction so his resources can be strategically invested. In this poster, we present SIDE (the Summarization Integrated Development Environment), an infrastructure that supports group learning facilitators in quickly getting a sense of the quality and effectiveness of a collaborative learning interaction so that the instructor is better equipped to carry out this challenging task.

It has been found that effective learning in collaborative groups is linked to the process by which learners work on the task together, how they construct arguments, and how they build on the contributions of their learning partners, otherwise known as transactivity. Earlier research suggests that it is more effective to judge the quality of an interaction for learning when transactivity based conversational contributions are flagged [2]. The goal of our proposed reporting interface is to track these qualities of collaborative discourse as is becoming more practical because of projects such as TagHelper tools [1] that are capable of automatic collaborative learning process analysis. However, investigating the patterns of conversational behavior that are most indicative of the quality of an interaction or indicative of trouble in an interaction, is still an active area of research. Open questions related to which types of

[1] This research was supported by Office of Naval Research, Cognitive and Neural Sciences Division, grant N00014-05-1-0043.

information and how much of it is ideal to present to group learning facilitators, and in what form, are still at the frontier of this important area of research. Thus, rather than present a specific reporting tool designed to do a single type of analysis and present reports in one way, we present a framework that facilitates rapid prototyping of such reporting interfaces so that research in this area can move forward at an accelerated rate.

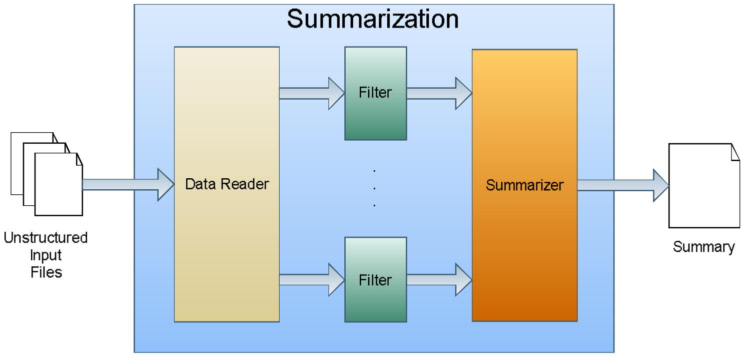
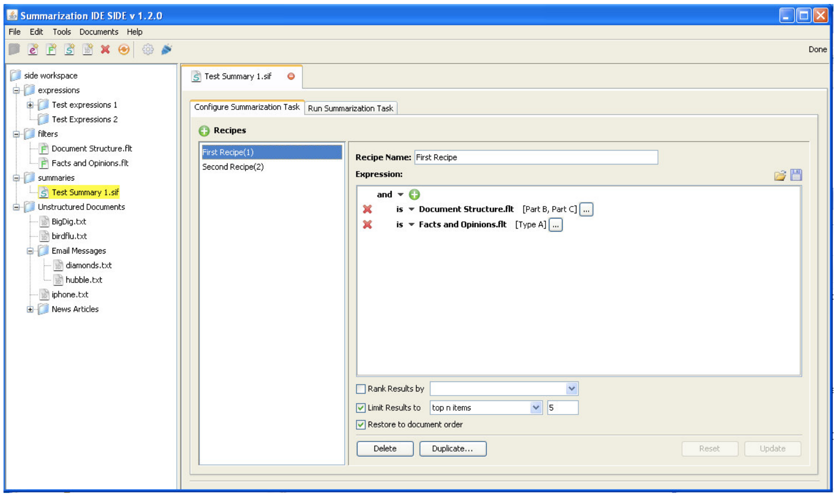


Fig. 1. SIDE Architecture

The architectural diagram in Figure 1 illustrates the two stage process of building a report from a discussion log. Consider the example of a report that charts how the consensus building style of different participants in an on-line discussion shifts over time [3]. The documents input to the process in Figure 1 are individual contributions to the discussion. A Data Reader reads in those texts and stores them in a flat representation. The Filters each encode a type of analysis that can be applied to the unstructured conversational data- e.g. one Filter might be responsible for applying a coding scheme that labels argumentation acts such as Claim, Data, Qualifier, or Warrant, while another flags segments at a higher level of argumentation structure such as Argument, Counter-Argument, or Integration. Thus, each Filter encapsulates the analysis functionality currently included in tool sets like TagHelper tools [1]. Once the Filters are applied to unstructured data, it then becomes structured data. The segmentation and labeling facilitate navigation through the data in a principled fashion. It is then possible to determine the most common type of contribution within a region of a conversation, or the last time someone contributed a Counter-Argument. The Summarizer in Figure 1 uses the structure applied by the Filters to select a subset of the conversation and presents either the texts themselves or some aggregation of the texts as the summary, for example by plotting the concentration of Counter-Arguments within 10 minute intervals over the course of a 2 hour discussion.

The interface for the Summarizer is displayed in Figure 2. A separate interface is provided for defining Filters, which is not shown. The Summarizer interface allows a summary to be defined as a sequence of Recipes. Each Recipe consists of a Selector, a Ranker, a Limiter, a Sequencer, and a Display. The Selector uses the analysis applied by the Filters to select a subset of segments of data that satisfy a Boolean expression- e.g. the expression might specify that all Counter Arguments and Integrations should be selected. The Ranker specifies criteria used to rate selected segments with respect to its importance for the summary. The Limiter specifies what proportion of the segments that were selected by the Selector should be retained and presented in the report- if 50% is selected, then the segments will be sorted based on the Ranker criteria, and the top half will be retained. These segments are then ordered by the Sequencer in the order in which they were contributed to the conversation. Finally, the sequenced list of segments are passed to a Display module, which might simply present the text from each segment in order, or might present some summary of what was found, like the ratio of Counter Arguments to Integrations, etc.



Recipe Name: First Recipe

Expression:

and
is Document Structure.flt [Part B, Part C]
is Facts and Opinions.flt [Type A]

□ Rank Results by

☑ Limit Results to top n items 5

☑ Restore to document order

Fig. 2. SIDE Architecture

We are currently pilot testing SIDE in a graduate course on Summarization and Personal Information Management where students are using it to develop prototype reporting tools, so that we can determine what additional functionality is desirable.

References

1. Rosé, C. P., 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, submitted to the International Journal of Computer Supported Collaborative Learning

2. Joshi, M. & Rosé, C. P.(2007), Using Transactivity in Conversation Summarization in Educational Dialog; SLaTE Workshop on Speech and Language Technology in Education
3. Weinberger, A. & Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. Computers & Education.