Identifying Localization in Peer Reviews of Argument Diagrams

Huy V. Nguyen and Diane J. Litman

University of Pittsburgh, Pittsburgh, PA, 15260
{huynv,litman}@cs.pitt.edu

Abstract. Peer-review systems such as SWoRD lack intelligence for detecting and responding to problems with students' reviewing performance. While prior work has demonstrated the feasibility of automatically identifying desirable feedback features in free-text reviews of student papers, similar methods have not yet been developed for feedback regarding argument diagrams. One desirable feedback feature is problem localization, which has been shown to positively correlate with feedback implementation in both student papers and argument diagrams. In this paper we demonstrate that features previously developed for identifying localization in paper reviews do not work well when applied to peer reviews of argument diagrams. We develop a novel algorithm tailored for reviews of argument diagrams, and demonstrate significant performance improvements in identifying problem localization in an experimental evaluation.

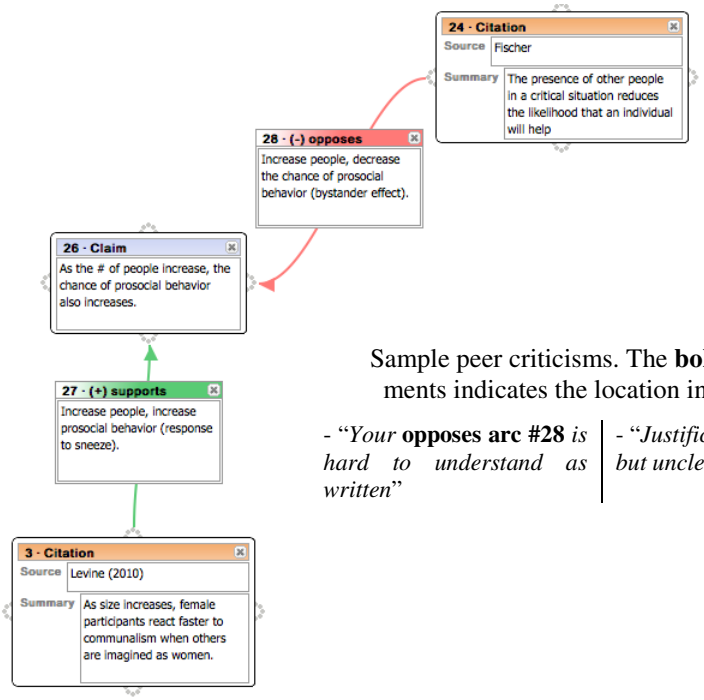
Keywords: peer review, argument diagrams, localization, localization pattern algorithm, natural language processing, SWoRD, LASAD.

1 Introduction

To facilitate writing and reviewing practices for students, web-based reciprocal peer-review systems such as SWoRD [3] have been built to manage typical activity cycles[1] such as writing, reviewing, back-evaluating, and rewriting. While some features of SWoRD are aimed at reducing potential drawbacks of novice reviewing (e.g., displaying review rating reliability indices, asking authors' to back-evaluate peer reviews), SWoRD does not automatically detect problems with student feedback, which in turn could be used to intelligently scaffold and tutor students to write better reviews. Prior work has shown that localization, which refers to pinpointing the source or location of a problem and/or solution, was one desirable feature of feedback regarding student writing, as it was significantly related to feedback implementation [5]. As the first step towards enriching SWoRD with such an automated assessment of student reviewing performance, Xiong and Litman [8] demonstrated the feasibility of using natural language processing (NLP) and machine learning to automatically predict localization in free-text feedback to student papers. In this paper we have a similar

[1] A basic function of SWoRD is to automatically distribute papers to reviewers and reviews back to authors given an instructor-defined number of reviews that each paper will receive.

K. Yacef et al. (Eds.): AIED 2013, LNAI 7926, pp. 91–100, 2013.



Sample peer criticisms. The **bold text** in comments indicates the location information.

- "*Your* **opposes arc #28** *is hard to understand as written*"	- "*Justification is sufficient but unclear in some parts.*"

Fig. 1. Excerpt from a student argument diagram, and samples of localized (left) and not localized (right) peer-review comments

interest in predicting localization, but in feedback regarding student *argument diagrams* rather than student papers.

There is increasing interest in developing software tools such as LASAD [6, 7] to support the learning of argumentation skills through graphical representations (see O. Scheuer at el. 2010 [7] for a recent review). In graphical argumentation, students create argument diagrams in which boxes represent statements and links represent argumentative or rhetorical relations between statements. Figure 1 shows an example LASAD diagram excerpt from our corpus. Recently, the idea of combining such graphical argumentation systems with peer-review systems has been proposed [1]. In such a combined system, student authors use argument diagramming to prepare or summarize their arguments; student argument diagrams are then distributed through a peer-review system to student reviewers for comment. Two example review comments associated with the LASAD argument diagram are also shown in Figure 1. Lippman et al. [4] studied such peer-review feedback comments to student argument diagrams, and showed that as with paper reviews, the presence of localization in feedback comments is strongly related to student implementation of peer feedback.

In this paper we present a new localization identification algorithm tailored to identifying localization in free-text peer feedback comments[2] to student argument diagrams. Experimental results show that when testing on a corpus of argument diagram reviews, our proposed algorithm outperforms a prior algorithm designed for feedback to student papers [8].

Section 2 introduces the corpus of argument diagrams and associated free-text review comments used in our study. Section 3 reviews the prior algorithm for identifying localization in paper reviews. Sections 4 and 5 next motivate and formalize our new algorithm for identifying localization in argument diagram reviews. Section 6 evaluates our algorithm. Finally, Sections 7 and 8 summarize our contributions and discuss future research.

2 Argument Diagram Review Corpus

Our corpus of peer-review textual feedback comments to student argument diagrams was collected in a Research Methods Lab at the University of Pittsburgh during Fall 2011. The Lab provided students with an opportunity to conduct psychological research and to write associated papers. To help students organize their thinking and create effective arguments, students were asked to create argument diagrams justifying their hypotheses using LASAD. LASAD argument diagrams consist of nodes and arcs from an instructor-defined ontology. The ontology for Research Methods consists of 4 node types (*current study, hypothesis, claim,* and *citation*) and 4 arc types (*comparison, undefined, supports,* and *opposes*). The diagram in Fig. 1, for example, contains three nodes (two citations and one claim) and 2 arcs (supports and opposes). Argument diagrams were later distributed via SWoRD to be reviewed by peer reviewers, using an instructor-defined rubric. Each student reviewer was asked to give textual feedback (the focus of our study), and to also grade the assigned diagrams on five dimensions using a 7-point scale. On average, each argument diagram was reviewed by 3 peers, with 19 textual comment units (defined below) per diagram.

The textual review feedback was segmented into 1104 comment units (defined as contiguous feedback referring to a single topic), then all comments were manually coded by two independent annotators (not the authors of this paper) for various coding schemes, two of which are relevant to our study. Each comment was first coded for the type of issue that it mentioned: *praise, summary, problem, solution, problem and solution (both), uncodeable*. Only comments having issue types of *problem,*

[2] SWoRD supports end-written comments as it is believed that a simple clicking interface that allows reviewers to point to a node/arc when providing a comment is too simple to address the localization issue. In diagram reviews, we have seen that reviewers may refer to more than one diagram component, or some missing node or arc. It is common in our corpus that reviewers mention groups of nodes and/or arcs when commenting on a line of argumentation. In such situations, reviewers may have trouble in pointing to the most appropriate node/arc expressing their comments. Moreover, click-to-point interfaces tend to lead reviewers to focus on low-level writing problems rather than evaluating the argumentation [5]. Due to such issues of direct annotations, we wish to support end-note written localizations.

solution, or *both* were further coded for localization; the localization values *yes* or *no* represented whether or not the exact location of the issue was mentioned in the comment. Inter-rater reliability for the two coding schemes is high with kappas of 0.87 for issue type and 0.84 for localization [4]. Our study focuses on the 590 comment units coded for localization (437 *yes*, 153 *no*). Fig. 1 shows an example localized comment (left) and an example not-localized comment (right).

In addition to the review comments, our corpus contains 56 student argument diagrams that were the targets of the 590 comments. While student papers were used to construct features for predicting localization in [8], we instead will extract features from student argument diagrams. In the next sections, we first review features used to predict localization in comments regarding papers [8], then describe our proposed algorithm that is tailored for predicting localization in reviews of argument diagrams.

3 Predicting Localization in Peer Reviews of Student Papers

Xiong and Litman [8] used NLP to develop features for predicting localization in peer-review comments of student papers. The class label was actually named pLocalization as it was coded for presence of problem localization in criticism feedback. Since this approach will serve as a baseline for evaluating our proposed algorithm, here we briefly describe this feature set.

Regular expression (**reg**) is a Boolean feature that indicates whether any of a predefined set of regular expressions are matched in a given comment. The regular expressions were manually created to match the structure of student papers, e.g. `on page 5`, `the section about`.

Domain word count (**dw_cnt**) is a numerical feature indicating the number of domain words present in a given comment, where the dictionary of domain words is automatically extracted from the set of papers being reviewed using statistical NLP techniques [8]. For our argument diagram review corpus, the domain words will instead be extracted from the textual content associated with the nodes and arcs in the set of student argument diagrams, e.g. `As the # of people increase, the chance of prosocial behavior also increases`, in the claim node of Fig. 1.

Syntactic properties of a comment are represented using two features. The Boolean feature **so_domain** indicates whether any domain word occurs between the subject and object of any sentence in the comment. **Det_count** indicates the number of demonstrative determiners (*this, that, these,* and *those*) in the comment.

Finally, the numerical features window size (**wnd_size**) and number of overlapped words (**overlap_num**) are constructed using an overlapping window algorithm for searching for the common text span between a comment and a student paper. The algorithm iteratively searches through the paper for the referred windows of the most likely text span in the comment, and merges any two windows that are found to overlap. The algorithm returns the length of the maximal window and the number of window's words present in the comment.

We use the original code developed in [8] to compute features from our corpus without any modification. It is likely that the regular expressions defined in [8] will

not be particularly applicable to our corpus of argument diagram reviews. However, all features are extracted automatically from data and we can easily compute them using our corpus (substituting the text extracted from the argument diagrams wherever the student paper text was previously used). We will thus examine the predictive utility of our new algorithm both in isolation, as well as in conjunction with the original feature set.

4 Patterns of Localization in Argument Diagram Reviews

Obviously, inherent differences in the structure of papers and argument diagrams makes the problem of identifying localization in diagram reviews different than identifying localization in paper reviews. For example, we observe that the graph structure of argument diagrams seems to make it more convenient for reviewers to include location information in their comments. In the paper review corpus studied in [8], only 53% of the review comments were coded as localized. In our diagram review corpus, in contrast, 74% of the comments are labeled as localized. Not only does the frequency of localization differ, but the way that localization is realized in review text differs when commenting on diagrams rather than papers. We hypothesize that a model tailored to the following observations regarding localization in argument diagram review will work better than simply applying the features in [8] to our corpus.

Pattern 1: Numbered Ontology Type. Every node or arc that is added to a LASAD argument diagram must have a header consisting of both a numerical ID, and a node/arc type from the ontology (headers are visually displayed in the colored bars in Fig. 1). It is very common in our corpus that reviewers identify a diagram component by referring to its node/arc type followed by its ID number, e.g. `hypothesis 1`, `claim 4`, `supports arc 27`.

Pattern 2: Textual Component Content. As the diagram is a summarized graphical representation of an argument, students usually make the text in the node and arc bodies very concise. Reviewers often use this text in conjunction with node and arc types to identify specific diagram components, e.g. `claim that women are more polite than men`, `gender hypothesis`, `your Levine citation`.

Pattern 3: Unique Component. Because a localized comment must be tied to a particular node or arc in the argument diagram, when there is a unique node or arc of a given type, localization can be done using a definite noun phrase expressing the node/arc type, e.g. `the opposing arc` (assuming there is only one opposes arc).

Pattern 4: Connected Component. It is possible to localize a component in a diagram by expressing its connection to another component, e.g. `support for the time of day hypothesis` (as the mentioned support node can be located accurately), `claim node in between the opposes and support arcs 28 and 27`.

Pattern 5: Typical Numerical Regular Expression. Due to the fact that all nodes and arcs are numbered, there are typical numerical expressions used by reviewers to express localization, e.g. `the first hypothesis`, `H1` (hypothesis 1), `[14]` (node or arc 14), `#28` (node or arc 28).

5 The Localization Pattern Algorithm (LPA)

The basic idea of our algorithm is that if **location information** expressed in a peer comment helps the author of an argument diagram pinpoint a unique part of the diagram, then that location information is a possible signal that the review comment is localized. Patterns for detecting such location information involve a **diagram component keyword** surrounded by **supporting word(s)**.

A diagram component keyword can be the word *node*, *arc*, or any of the words defining the node and arc types from the diagram ontology. Recall that ontologies are defined by instructors, and may differ across courses. For our corpus, the keywords from the ontology include the node and arc types introduced in Section 2: *current study, hypothesis, claim, citation, comparison, undefined, supports,* and *opposes*. Our algorithm has been implemented to extract such keywords automatically by parsing the ontology.

In general, supporting word(s) are one or more words in proximity of a keyword, that help readers locate the diagram component(s) mentioned in a review comment. For example, the noun phrase `gender hypothesis` has the word `hypothesis` as its keyword; the word `gender` plays a supporting role when it distinguishes the mentioned hypothesis from other hypotheses that may exist in the diagram. For the noun phrase `gender hypothesis` to express location information in a peer comment, there must be a hypothesis node in the diagram and that node must have gender in its textual content.

To search for location information using patterns, we first segment peer-review comments into sentences, remove stop-words, and extract the keywords in each sentence. For each keyword found in a sentence, we collect all remaining non-keywords in the sentence that also appear in the text of a node or arc that is consistent with the keyword. We note that all keywords and content words are stemmed before being fed to a word matching procedure. To determine whether such words are supporting words that indicate localization, we then apply rules representing the 5 types of localization patterns noted above.

For the first pattern, we define supporting words as a number or list of numbers occurring right after the keyword, where the numbers match diagram component IDs.

The second pattern involves two cases. First, supporting words must occur before the keyword, e.g. `gender hypothesis`. This case requires that the nearest supporting word is right before the keyword. Second, supporting words can be after the keyword, e.g. `claim that women are more polite than men`. This case requires that the nearest supporting word must have distance less than 3 from the keyword, and the number of supporting words is at least 3.

For pattern 3, we count the number of nodes and arcs of each type when parsing the argument diagram, to easily determine whether or not the found keyword refers to a unique component of the diagram.

Pattern 4 can be addressed by doing reference resolution in the argument diagram. For each node and arc of the diagram, we extend its original textual content by adding sections that contain exactly the text of the node and/or arc to which it connects. While searching for common words between a review sentence and a diagram node/arc, we tag a matching phrase as support if it is in the added sections of the component. The rule is that the matching phrase in the original text must be a keyword, and the matching phrase in added sections must be location information.

Finally, pattern 5 was created by looking for typical regular expressions seen in the held-out set of development data to be described next.

As our localization pattern algorithm is rule-based, it was important to have development data to learn the localization patterns and create the rules for identifying those patterns. Fortunately, there was a data segment from the Fall 2011 Research Methods Lab which was not coded for localization, and was thus not included in our testing corpus. The first author collected 200 phrases[3] representing references to locations from that data segment. Those 200 localized phrases were used to learn the patterns and refine the parameters for the localization pattern algorithm. Note that the localization annotation described in Section 2 required comments to have an issue type of only problem, solution, or both; annotators were also instructed to look at the target diagram to verify location information. The first author did not follow those instructions, and collected location information from comments of all issue types, without the diagrams.

6 Experimental Results

We evaluate the predictive performance of two models that use LPA to identify localization in peer reviews of student argument diagrams, by comparing their performance to two baselines: a model (pLocalization) learned using only the paper review features [8] described in Section 3, and a model (Majority) that simply determines the most common class (localized) in the data and assigns every instance that class label. Our first proposed model directly uses LPA as the classifier for localization; if LPA can extract location information from a comment by matching at least one of its patterns, then the comment is classified as localized, otherwise it is classified as not-localized. Our second proposed model (Combined) adds the binary value returned by LPA as an additional feature to the original pLocalization feature set.

Table 1. Performance of 4 models for identifying localization. * denotes significantly better than the majority baseline with $p < 0.05$.

Metric	Majority	pLocalization	LPA	Combined
Accuracy (%)	74.07	73.98	80.34 *	83.78 *
Kappa	0	< 0.01	0.54 *	0.56 *
Weighted Precision	0.55	0.55	0.83 *	0.84 *
Weighted Recall	0.74	0.74	0.80 *	0.84 *

Table 1 shows the predictive performance for these 4 localization classifiers. To make the experiment consistent with [8], models involving pLocalization features are learned using the WEKA[4] J48 decision tree algorithm; testing with other algorithms (e.g. SVM and Logistic) did not yield significantly different results. All models are evaluated via 10-fold cross validation. Our results show that while the pLocalization

[3] Some phrases are used as examples in Section 4.

[4] `www.cs.waikato.ac.nz/ml/weka`. Algorithms in our experiments use parameters set to the defaults.

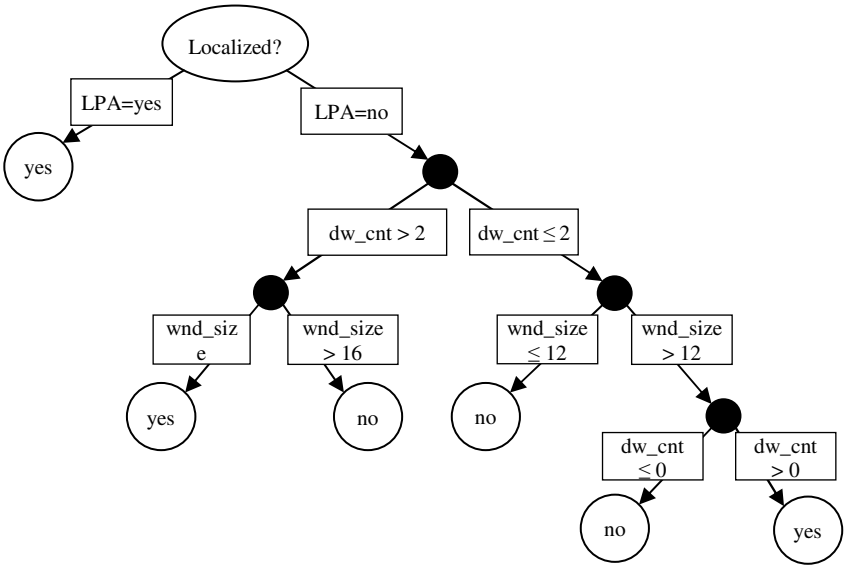


Fig. 2. Learned decision tree for predicting localization of argument-diagram reviews, leaves are prediction outputs, conditions are in rectangle boxes

model does not outperform Majority for any metric, LPA alone significantly outperforms Majority for all metrics. The significant improvement in precision, recall and kappa shows that LPA can predict efficiently the minor class which the baseline models fail to predict. Furthermore, the Combined model yields the best results of all, with accuracy and weighted recall values significantly better than LPA alone ($p < 0.05$).

Fig. 2 presents the decision tree learned for the Combined model. The LPA feature appears at the root, with comments classified as localized if LPA outputs *yes*. Two features from [8] (domain word count (dw_cnt) and window size (wnd_size)) are used to refine the cases in which LPA outputs *no*. Note that the regular expression feature (reg), which was the most predictive feature for paper reviews [8], is not predictive for diagram reviews. This result shows the advantage of diagram-tailored features.

7 Related Work

Research has been conducted to understand what type of feedback is the most helpful, and why it is helpful. Nelson and Schunn [5] studied relationships between feedback features, potential internal mediators and feedback helpfulness in terms of the likelihood of implementation. Their assumption was that feedback features may not directly affect implementation, but instead do so through internal mediators because of the complex nature of writing performance. The corpus consisted of peer reviews of student papers in a History class, which were coded for feedback features, e.g. localization. The authors' back-review regarding peers' comment were coded for internal

mediators, e.g. problem understanding. Nelson and Schunn found that localization in review was significantly related to problem understanding which is an effective mediator that significantly relates to implementation.

Unlike Nelson and Schunn's study on peer reviews of student papers [5], Lippman et al. [4] studied what influences the implementation of peer reviews of student argument diagrams. Peer reviews were collected from a Research Method Lab in which students were asked to give feedback, and rate argument diagrams of their peers. The authors coded peer feedback for various features, e.g. problem, solution, localization. Their finding was consistent with Nelson and Schunn [5] to an extent, and showed that issue type (problem, solution, or both) and localization have distinct, non-interacting influence on the implementation of peer feedback. In addition, results in [4] also suggested that location information helps student implement peer feedback when the focus of the critique is more complex as opposed to more superficial.

Cho [2] further investigated the relationship between feedback features and feedback helpfulness, but using a machine-learning approach. Peer reviews were collected from a Physics class using SWoRD, and were human-coded for various issue types, e.g. problem detection, solution suggestion. Each review was then labeled as helpful or not helpful in terms of these issue types. Experimental results showed that peer reviews can be classified regarding helpfulness with accuracy up to 67% using simple NLP techniques. While Cho's work strengthened the understanding of some feedback features regarding peer review helpfulness, our work instead aims to automatically identify one important aspect, i.e. localization; we also focus on diagram reviews rather than paper reviews, and use different NLP techniques for feature construction.

Given findings of previous studies showing that localization is an important indicator of feedback helpfulness, Xiong and Litman [8] used NLP techniques and supervised machine learning to automatically identify the problem localization in peer feedback. Their work is different from ours firstly at the data domain. While Xiong and Litman studied peer reviews of student papers, the data domain in our study is peer reviews of student argument diagrams. The second difference between our work and [8] is at the syntactic level of features extracted from the textual content. Xiong and Litman proposed using features from the parsed dependency tree of the sentence to abstract their intuition regarding the structure of localized reviews. In this study, we however focus only on the word level by considering common words between peer reviews and student diagram. Our intuition regarding structure of localized reviews is formulated simply through the relative order between keywords and supporting words.

8 Conclusion and Future Work

This paper presents the LPA algorithm for identifying localization in peer reviews of argument diagrams. Experimental results show that LPA outperforms a model developed for student papers with respect to a number of evaluation metrics, and that combining the two approaches works best of all. The combined model has the LPA feature appear at the root of the learned decision tree. Even though the location patterns

were defined manually based on the development data, they show potential generality by yielding significantly high accuracy on the test data. Recall that the development data and test data are non-overlapping which means all reviewers in the development set are not those in the test set. Moreover, the only domain-specific features used in our combined model are keywords and domain-words lists which can be extracted automatically by parsing instructor-defined ontologies and student-generated diagrams. Therefore we expect the model will work well with new argument diagram reviews from other courses with different ontologies and content domains.

In future work, we aim to apply advanced learning techniques to automatically learn the type of rules and regular expressions used in LPA, rather than use our cur-rent hand-engineered approach. We also plan to evaluate the generality of our LPA and Combined models, by testing them on data currently being collected from courses with different argument diagram ontologies. In addition we are incorporating the Combined model into SWoRD and will be evaluating its use for intelligent scaffolding. Finally, we plan to adapt the lessons learned from developing LPA back to the area of paper reviews. It is more challenging to learn keywords and supporting words from paper comments, but we expect that the task will be feasible when localization patterns can be learned automatically.

Acknowledgements. This material is based upon work supported by the National Science Foundation under Grant No. 1122504. We are grateful to J. Lippman and our other colleagues for providing us with the annotated corpus. We thank members of both the ArgumentPeer and ITSPOKE projects for commenting on our research, W. Xiong and M. Lipschultz for providing feedback regarding this paper, and the reviewers for their many constructive comments.

References

1. Ashley, K.D., Goldin, I.M.: Toward AI-enhanced Computer-supported Peer Review in Legal Education. In: Proceedings of JURIX 2011, pp. 3–12 (2011)
2. Cho, K.: Machine classification of peer comments in physics. In: Proceedings of the Educational Data Mining 2008, pp. 192–196 (2008)
3. Cho, K., Schunn, C.D.: Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. Computers and Education 48(3), 409–426 (2007)
4. Lippman, J., Elfenbein, M., Diabes, M., Luchau, C., Lynch, C., Ashley, K.D., Schunn, C.D.: To Revise or Not To Revise: What Influences Undergrad Authors to Implement Peer Critiques of Their Argument Diagrams? In: ISPST 2012 Conf., poster (2012)
5. Nelson, M.M., Schunn, C.D.: The nature of feedback: How different types of peer feedback affect writing performance. Instructional Science 37(4), 375–401 (2009)
6. Scheuer, O., McLaren, B.M., Loll, F., Pinkwart, N.: An Analysis and Feedback Infrastructure for Argumentation Learning Systems. In: Proceedings of AIED 2009, pp. 629–631 (2009)
7. Scheuer, O., Loll, F., Pinkwart, N., McLaren, B.M.: Computer-supported argumentation: A review of the state of the art. International Journal of Computer-Supported Collaborative Learning 5(1), 43–102 (2010)
8. Xiong, W., Litman, D.: Identifying Problem Localization in Peer-Review Feedback. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010, Part II. LNCS, vol. 6095, pp. 429–431. Springer, Heidelberg (2010)