Impact of Annotation Difficulty on Automatically Detecting Problem Localization of Peer-Review Feedback

Wenting Xiong[1], Diane Litman[1,2] and Christian Schunn[2]

[1] Department of Computer Science, University of Pittsburgh
[2] Learning Research and Development Center, University of Pittsburgh

Abstract. We believe that providing assessment on students' reviewing performance will enable students to improve the quality of their peer reviews. We focus on assessing one particular aspect of the textual feedback contained in a peer review – the presence or absence of problem localization; feedback containing problem localization has been shown to be associated with increased understanding and implementation of the feedback. While in prior work we demonstrated the feasibility of learning to predict problem localization using linguistic features automatically extracted from textual feedback, we hypothesize that inter-annotator disagreement on labeling problem localization might impact both the accuracy and the content of the predictive models. To test this hypothesis, we compare the use of feedback examples where problem localization is labeled with differing levels of annotator agreement, for both training and testing our models. Our results show that when models are trained and tested using only feedback where annotators agree on problem localization, the models both perform with high accuracy, and contain rules involving just two simple linguistic features. In contrast, when training and testing using feedback examples where annotators both agree and disagree, the model performance slightly drops, but the learned rules capture more subtle patterns of problem localization.

Keywords problem localization in text comments, data mining of peer reviews, inter-annotator agreement, natural language

1 Introduction

Computer-supported peer-review has become more and more popular in various classroom settings. Peer-review is helpful not only because it provides learning opportunities as students play the role of reviewers, but also because it typically generates feedback from multiple reviewers for each student, which is likely to cover broader aspects of both domain knowledge and writing issues compared with a single instructor review. However, the quality of peer-review feedback varies with students' reviewing skills [1]. To better control the quality of peer-review feedback, we hypothesize that it would be beneficial to assess students'

reviewing performance and provide proper guidance for students to give more helpful feedback for their peers. To date no peer-review software is capable of responding to students' reviewing performance. Instead, most software is used as a document management and work allocation tool; for example, the SWoRD system[2] collects students' essays, automatically assigns each student's essay to multiple students for reviewing, and then collects feedback and gives feedback back to the corresponding students.

[3] studies how feedback features affect the helpfulness of peer-review feedback in terms of feedback implementation, and finds that *problem localization* is the feature (of those studied) that most significantly correlates with the likelihood of implementation. Problem localization means pinpointing the source of the problem and/or solution with respect to the associated essay (in other words, whether students can find the problem in their essay based on the feedback). One example of problem-localized peer-review feedback is given below:

> **Example of problem-localized peer-review feedback:** *On the 5th page, when the author is talking about the European immigrants who spoke out in favor of the principles of the US, it would be nice for the author to name a few.*

"On the 5th page" gives the explicit location of the problem, and "European immigrants who spoke out in favor of the principles of the US" provides the context for locating the problem. Localizing the problem in this way rather than simply saying "It would be nice for the author to give some examples when making the arguments." makes the problem easier to identify, and thus helps feedback receivers understand the feedback.[3]

Our ultimate goal is to automatically predict peer-review feedback quality so that software such as SWoRD could be able to provide assessment and tutoring on students' reviewing performance. Currently we focus on predicting the particular feedback feature "problem localization" that was shown to be most important in [3]. In previous work, we demonstrated the feasibility of using supervised machine learning and natural language processing to develop a classification model that predicts problem localization from linguistic features automatically extracted from textual feedback [4, 5]. Our feature set included regular expression features (RegularTag) to capture the usage of canonical location phrases such as "On the 5th page" in the example above. We also used domain lexicon features (e.g. #DomainWords), and syntactic features (e.g. SO_domain – whether there is a domain word between the subject and the object, #demDeterminers – the number of demonstrative determiners) to identify other patterns of problem localization such as indirect quotation. In the example above, domain words and phrases such as "European immigrants" and "US" are buried in a prepositional

[3] One solution to supporting localization is to allow direct annotation in the margins rather than use the more typical end-node approach. However, marginalia tends to lead reviewers to focus on low-level writing problems. Therefore, we wish to support written localizations.

phrase that serves as the object of the verb "talk", which is an opinion verb of the subject "author".

While our learned classification model predicted problem localization with nearly 80% accuracy, misclassification errors still remain. However, judging problem localization is not easy even for humans, given the fact that the inter-annotator agreement between two trained annotators as measured by Kappa is only 0.64 (Section 3). We thus hypothesize that the performance of the classifier might not only have been limited by the features it used, but also by the disagreement between the two annotators. Therefore, before developing more sophisticated features to cover more problem localization patterns, we would like to investigate the types of text feedback that human annotators find difficult to annotate, and the impact of including such difficult to annotate feedback examples when learning predictive models. Our results show that when models are trained and tested using only feedback where annotators agree on problem localization, the models both perform with high accuracy, and contain rules involving just two simple linguistic features. In contrast, when training and testing using feedback examples where annotators both agree and disagree, the model performance slightly drops, but the learned rules capture more subtle linguistic patterns of problem localization. Our quantitative results shed light on how annotation disagreement impacts the automatic prediction of problem localization, while our qualitative results suggest how localization is signaled linguistically, and which signals are more straightforward for human coders to use.

2 Related Work

Since problem localization is defined as a binary feature that is true when the problem/solution in the feedback can be easily located in the relevant essays with the given feedback, by nature identifying problem localization involves detecting reference information from one document to another, such as indirect quotation of certain content of the associated essay. One area of related work to our task is identifying quotation from reference works in primary materials for digital libraries [6]. This work is similar to ours since it also considers quotation as reference from one document to the other, although in their case most quotations are direct while quotations in the peer-review feedback are more likely to be indirect. They proposed an overlapping-window algorithm that searches for the most likely referred window of words through all possible primary materials to match a possible citation in a reference work. We applied this algorithm for our purpose, and developed features from information on the window that the algorithm retrieved (e.g. window size, the number of overlapped items).

There is some other related work on peer-review corpora that tries to predict certain feedback features automatically. [7] worked on a corpus that SWoRD collected, and compared three supervised learning algorithms in classifying peer-review comments with respect to several feedback issues (not including problem localization) separately, by means of features obtained using a text classification system which is based on non-semantic information in text. We instead only

focus on problem localization, and explore features that capture both syntactic and semantic information of text. Also they did not consider the impact of annotation disagreement, which is focused of in this paper.

The reliability of inter-annotator agreement is a common issue that has been analyzed in various domains. [8] evaluated an annotation scheme with respect to both inter-annotator agreement and the use of the annotations for training predictive models, by performing the same machine learning experiment using both agreed and consensus data. Agreed data contained only the examples where both annotators originally agreed on the annotation [9]. Consensus data in addition contained the originally disagreed examples; however, these examples were relabeled by the original annotators with a single consensus label after discussion [10]. The use of consensus data not only increased the amount of training data for machine learning relative to just using original agreement cases, but also provided more subtle training examples. Our analysis of inter-annotator disagreement will similarly compare the use of (double-coded) agreed and consensus data in machine learning experiments, and in addition will examine the use of single-coded data (labeled by only one annotator).

3 Data and Methods

In this work, we use the same corpus examined in our prior study of problem localization [4, 5]. This corpus was collected using SWoRD in a college level history introductory class, and first used to study the relationship between feedback features and helpfulness [3]. In [3], all the textual reviews were manually segmented into 1045 idea-units (defined as contiguous feedback referring to a single topic). These units were then annotated by two independent annotators for various feedback features. For our work, we try to automatically predict the particular feedback feature that was found to be most predictive of helpfulness – problem localization (*pLocalization*). According to the coding scheme, pLocalization is only applicable for criticism feedback, which constitutes 875 of the 1045 feedback idea-units. Among the 875 criticism feedback (for the rest of the paper, feedback is used to refer to the feedback idea-unit), 52.8% were annotated as "True" for pLocalization (the majority class).

Since not all feedback were double-coded by the two annotators, [3] analyzed the reliability of the annotations on the subset of data that were annotated by both annotators, in which 338 of them are criticism feedback. Within the 338 double-coded criticism feedback, both annotators agreed on the label of pLocalization for 277 cases. For the remaining 61 disagreed cases, the annotators later determined a consensus annotation. The rest of the annotations remained singly annotated. In their study of feedback helpfulness [3], all 875 criticism feedback were used by combining the consensus labels of the double-coded subset with the rest of the labels on the single-coded subset; we will refer to this set as the "combined" criticism feedback. Relevant statistics are listed in Table 1.

Table 1. Descriptive statistics of annotations for pLocalization in different data sets

pLocalization	True	False	Total	True %
Double-coded agreed	154	123	277	55.6%
Double-coded consensus	182	156	338	53.8%
Combined criticism	462	413	875	52.8%

Table 2. Confusion matrix of the double-coded pLocalization. $Kappa = 0.64$

		Coder 2		Total
		True	False	
Coder 1	True	154	15	169
	False	46	123	169
Total		200	138	338

In this paper, we mainly focus on the 338 double-coded criticism feedback for our analysis of the impact of annotation disagreement. The confusion matrix summarizing the inter-annotator agreement is presented in Table 2 (Kappa = 0.64). To illustrate the subtleties between agreed and disagreed annotations for pLocalization, we present 3 examples as follows.

1. **Agreed: pLocalization = True**
   On the 5th page, when the author is talking about the European immigrants who spoke out in favor of the principles of the US, it would be nice for the author to name a few.
2. **Disagreed: pLocalization = True or False**
   Just when bringing up each topic try to bring up things that one would not normally think about, make your essay a little more interesting by doing this, but besides that, great job!
3. **Agreed: pLocalization = False**
   Again, be careful about time frames and about well-articulated introductions to and interconnections of arguments to reach maximum clarity for all aspects of your paper, including complex and significant historical and contemporary insight.

Since our goal is to analyze the impact of annotation disagreement on learning to predict problem localization, we will use the same feature set and learning algorithm as in our prior work [4]. As discussed in Section 1 (and detailed in [4]), our feature set includes regular expression features, domain lexicon features, syntactic features, and overlapping-window features. For supervised machine learning we use the Decision Tree (J48) algorithm provided by WEKA.[4]

4 Experiments

We conducted several experiments to investigate the impact of coders' annotation disagreement on both the predictive power and the content of the learned

[4] http://www.cs.waikato.ax.nz/ml/weka/

models. We first compare the quantitative impact of using the double-coded **agreed** data, the double-coded **consensus** data, and the **combined** data (Table 1), by training and testing on each data set separately and comparing model performance using standard metrics. Next, we evaluate how the model learned from the agreed data (which performed best in the first experiment) generalizes when tested using noisier examples from the consensus and combined data. Finally, we present a qualitative analysis of our learning experiments, by comparing how the linguistic features used to predict problem localization change when the model is trained from agreed versus consensus versus combined data.

4.1 Learning using Agreed vs. Consensus vs. Combined Labels

In this experiment we use supervised machine learning to predict problem localization (pLocalization) from each of our data sets separately. For all learned models, we perform 10-fold cross validation to evaluate performance. We also compare each model against a majority class baseline (always predict the class that constitutes the major part of the data) with respect to the data from which the model is learned. Our first learned model (agreed model) was trained and tested on the 277 examples where both annotators originally agreed on the label of pLocalization. The second model (consensus model) was trained and tested using the consensus labels of all 338 double-coded instances (the 277 originally agreed examples, and the 61 originally disagreed examples). Results are presented in Table 3. To save space we abbreviate precision of the true class of problem localization to be P_true, and the corresponding recall and F-measure to be R_true and F_true, respectively. We also do the same for the false class.

Table 3. Performance of agreed and consensus learned models (cross validation)

Model	No.	Accuracy	P_true	R_true	F_True	P_false	R_false	T_false	Kappa
agreed baseline	277	55.6%	0.56	1	0.69	0	0	0	0
agreed model	277	81.6%	0.97	0.69	0.81	0.71	0.98	0.83	0.64
consensus baseline	338	53.8%	0.58	1	0.70	0	0	0	0
consensus model	338	74.6%	0.79	0.72	0.75	0.70	0.78	0.74	0.49

Comparing each model against its baseline, Table 3 shows that both models outperform the corresponding baseline with respect to the two overall performance metrics, namely Accuracy and Kappa ($p < 0.05$). Comparing the agreed and consensus models,[5] Table 3 in addition shows that the agreed model outperforms the consensus model for these two metrics, although the differences are not statistically significant. We also see that for the metrics that evaluate

[5] Since the baselines of the agreed and consensus models differ, we also compared each model's relative improvement over its baseline error. Since results using normalized versus absolute values were the same, we only discuss the absolute values here.

each type of prediction separately, the learned models outperform the baselines, and the agreed model outperforms the consensus model, for all metrics except R_true. In sum, our results suggest that for identifying problem localization overall, models that are learned from agreed data are most accurate. However, when evaluating each class separately, while the agreed model has better results for predicting "false" and better precision for predicting "true", the consensus model has better recall for the "true" class. This suggests that consensus models might capture more information about the problem localization that is introduced by the originally disagreed cases (the agreed data is a subset of the consensus data).

Recall that in our prior work [4], our model was learned from the combined labels of all 875 criticism feedback (combined model). We can compare the performance of this combined model (again evaluated by 10-fold cross validation, results in Table 4) with the agreed and consensus models. Comparing Tables 3 and 4 shows that for Accuracy and Kappa (and for most of the other metrics), the agreed model still performs better than the combined model, despite the much smaller number of training examples in the agreed data set. This further confirms that training from agreed data yields higher predictive accuracy compared to training from consensus or combined data. In contrast, the combined model outperforms the consensus model for all metrics, which suggests that when noisier training data is involved, we can continue to improve our model's performance by increasing the number of training examples.

Table 4. Performance of combined learned model (cross validation)

Model	No.	Accuracy	P_true	R_true	F_True	P_false	R_false	T_false	Kappa
combined baseline	875	52.9%	0.53	1	0.69	0	0	0	0
combined model	875	78.5%	0.84	0.73	0.78	0.74	0.85	0.79	0.57

Finally, it is important to point out that the Kappa of the combined model (0.57) is close to what human annotators achieved (Kappa = 0.64 in Table 2).[6] This implies that our model is not so far off from the upperbound of human annotations. On the other hand, since human agreement is relatively low, the annotation scheme could be improved to increase reliability.

4.2 Training on Agreed, Testing on Consensus vs. Combined

In this experiment we investigate whether the high predictive accuracy of the model learned from the agreed data in the prior section generalizes when tested on more difficult examples Our first testing set (disagreed set) contains the consensus labeled 61 double-coded criticism feedback, where the two annotators originally disagreed on pLocalization. The second testing set (combined set) was

[6] The human Kappa should be compared to the combined or consensus Kappas, but not the agreed Kappa, because the human annotators agreed on all instances in the agreed data set (thus for agreed, human Kappa = 1).

constructed to be of the same size as the disagreed set, by randomly selecting 61 instances from the 875 examples in the combined set, after removing the 277 agreed examples used for training the agreed model. Note that since the combined test set could be a mixed set of originally disagreed as well as single-coded feedback, it might have higher inter-annotator agreement (however, we cannot verify this given the single-coding).

Table 5. Testing the agreed model on consensus and combined data (held-out)

Testing Set	No.	Accuracy	P_true	R_true	F_True	P_false	R_false	T_false	Kappa
disagreed set	61	55.7%	0.52	0.43	0.47	0.58	0.67	0.62	0.10
combined set	61	72.1%	0.81	0.48	0.61	0..69	0.91	0.79	0.41

Experimental results presented in Table 5 show that the agreed model performed better (for all metrics) on the combined set, which might potentially have higher inter-annotator agreement. It is reasonable that the disagreed set is harder to predict in the supervised learning scheme, since the labels of the disagreed set were all originally disagreed by human annotators. Comparing Table 5 with the agreed model entry in Table 3 shows that the agreed model performed much better when tested on the agreed data (although it should be noted that Table 3 results were obtained using 10-fold cross validation while Table 5 results were obtained using held-out test sets). Our interpretation is that the agreed model was trained from only the agreed data, which is likely to only capture simple patterns of problem localization that humans can easily agree on; thus the agreed model is not able to recognize the other underlying patterns that are not as straightforward as what the agreed model is trained from. When there are easier cases in the testing set such as in the agreed set and likely in the combined set, the agreed model predicted more accurately in general. These results further support our hypothesis that annotation disagreement does indeed impact the predictive power of learned models.

4.3 Analysis of the Learned Trees

Interestingly, there is a great difference in the corresponding decision trees of the models that are learned from different data sets. Fig. 1 contains the decision trees of the agreed and consensus models, while Fig. 2 is the decision tree of the combined model. For prediction, the model will go through one particular path of the tree from the root to the leaves, where each node corresponds to one feature in the data set. For example, the model in part (a) of Fig. 1 first looks at whether any regular expression matches the feedback: if yes, it stops and predicts the feedback as problem-localized ("True"); if no, it continues and looks at the second feature – #DomainWords. If the number of domain words is greater than the threshold of 8, the model predicts the feedback as problem-localized, otherwise it predicts not problem-localized ("False"). The numbers (x/y) following the

decision conditions represent that there are x instances predicted at this node while y of x are predicted wrong. Values in Fig. 1 and Fig. 2 are the results when models are tested on themselves, thus these values represent the fitness of the learned decision trees with respect to the corresponding training data sets.

Since the decision tree algorithm embeds feature selection in its learning process, it will automatically select the most discriminative features regarding the training set. Thus, even though we extract the same features across training sets, the decision tree algorithm could selectively use different features, thus the learned tree structures could be quite different. Furthermore, the predictive power of a feature might not be the same even when it appears in different models, due to the differences in the training corpora. Note that features selected by the two trees in Fig. 1 are exactly the same, suggesting that when the training set is small, the usage of localization expressions (RegularTag) and domain lexicons (#DomainWords) are the main factors to separate problem-localized feedback from the others. But when the model needs to consider more examples as in Fig. 2, it has to rely on more sophisticated features (i.e. the syntactic features (SO_domain, #demDeterminers) and overlapping-window features (windowSize – the size of the overlapped window) introduced in Section 1) to detect more subtle expressions of problem localization.

```
RegularTag = False
|  #DomainWords  <= 8: FALSE (168.0/48.0)
|  #DomainWords  > 8: TRUE (38.0/3.0)
RegularTag = True: TRUE (71.0)
```

(a) the agreed model

```
RegularTag = False
|  #DomainWords  <= 8: FALSE (206.0/64.0)
|  #DomainWords  > 8: TRUE (51.0/7.0)
RegularTag = True: TRUE (81.0/7.0)
```

(b) the consensus model

Fig. 1. Decision trees of the agreed and consensus models.

```
RegularTag = False
|  #DomainWords <= 5: FALSE_N/A (468.0/124.0)
|  #DomainWords  > 5
|  |  SO_domain = True: TRUE (144.0/22.0)
|  |  SO_domain = False
|  |  |  #detDeterminers <= 0: TRUE (59.0/22.0)
|  |  |  #detDeterminers  > 0
|  |  |  |  windowSize <= 18: FALSE_N/A (7.0)
|  |  |  |  windowSize > 18: TRUE (2.0)
RegularTag = True: TRUE (195.0/18.0)
```

Fig. 2. Decision tree of the combined model.

5 Conclusion

In this paper we investigated whether and how the disagreement of annotations affects the models we build for identifying problem localization in peer-review feedback. The quantitative results as well as the learned models (decision trees) suggest that while agreed data can achieve higher predictive accuracy in general (Section 4.1), it cannot be generalized well to consensus data or combined data that might involve inter-annotator disagreement (Section 4.2). Furthermore, while the model learned from pure agreed data performs well using only regular expressions and domain lexicons for identifying simple localization patterns, more sophisticated features are necessary to capture more subtle expressions in general problem localization cases (Section 4.3). This might indirectly reflect that different localization patterns have different difficulty for annotators represented as inter-annotator disagreement. In sum, we have shown that annotation disagreement does exert an impact on the predictive power and content of the learned models. Our work also suggests how localization is signaled linguistically, and which signals are more easily used by human coders. We hope to use these insights in future work to refine our annotation manual, to improve both inter-annotator agreement and ultimately model performance.

References

1. Kluger, A.N., DeNisi, A.: The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory. Psychological Bulletin **119**(2) (1996) 254–284
2. Cho, K., Schunn, C.D.: Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. Computers and Education **48**(3) (2007) 409–426
3. Nelson, M.M., Schunn, C.D.: The nature of feedback: how different types of peer feedback affet writing performance. Instructional Science **37** (2009) 375–401
4. Xiong, W., Litman, D.: Identifying problem localization in peer-review feedback. In: Proc. of Tenth International conference on Intelligent Tutoring systems. (2010)
5. Xiong, W., Litman, D., Schunn, C.: Assessing reviewers performance based on mining problem localization in peer-review data. In: Proc. of Third International Conference on Educational Data Mining. (2010)
6. Ernst-Gerlach, A., Crane, G.: Identifying quotations in reference works and primary materials. Research and Advanced Technology for Digital Libraries (2008) 78–87
7. Cho, K.: Machine learning of peer comments in physics. In: Proc. of Educational Data Mining 2008. (2008)
8. Litman, D.J., Forbes-Riley, K.: Annotating student emotional states in spoken tutoring dialogues. In: Proc. of 5th SIGdial Workshop on Discourse and Dialogue. (2004)
9. Lee, C., Narayanan, S., Pieraccini, R.: Recognition of negative emotions from the speech signal. In: Proc. of ASRU. (2001)
10. Ang, J., Dhillon, R., Krupski, A., Shriberg, E., Stolcke, A.: Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In: in Proc. ICSLP 2002. (2002) 2037–2040