

ArgRewrite: A Web-based Revision Assistant for Argumentative Writings

Fan Zhang Rebecca Hwa Diane Litman Homa B. Hashemi

University of Pittsburgh

Pittsburgh, PA, 15260

{zhangfan,hwa,litman,hashemi}@cs.pitt.edu

Abstract

While intelligent writing assistants have become more common, they typically have little support for revision behavior. We present ArgRewrite, a novel web-based revision assistant that focus on rewriting analysis. The system supports two major functionalities: 1) to assist students as they revise, the system automatically extracts and analyzes revisions; 2) to assist teachers, the system provides an overview of students' revisions and allows teachers to correct the automatically analyzed results, ensuring that students get the correct feedback.

1 Introduction

Making revisions is central to improving a student's writings, especially when there is a helpful instructor to offer detailed feedback between drafts. However, it is not practical for instructors to provide feedback on every change every time. While multiple intelligent writing assistants have been developed (Writelab, 2015; Draft, 2015; Turnitin, 2016), they typically focus on the quality of the current essay instead of the revisions that have been made. For example, Turnitin identifies weak points of the essay and gives suggestions on how to improve them; it also assigns an overall score to the essay so students can get a coarse-grained feedback on whether they are making progress in their revisions. However, without explicit feedback on each change, students may inefficiently search for a way to optimize the automatic score rather than actively making the existing revisions "better". Moreover, because students are the target users of these systems, instructors typically can neither correct the errors made by

the automatic analysis nor observe/assess the students' revision efforts.

We argue that an intelligent writing assistant ought to be aware of the revision process; it should: 1) identify all significant changes made by a writer between the essay drafts, 2) automatically determine the purposes of these changes, 3) provide the writer the means to compare between drafts in an easy to understand visualization, and 4) support instructor monitoring and corrections in the revision process as well. In our previous work (Zhang and Litman, 2014; Zhang and Litman, 2015), we focused on 1) and 2), the automatic extraction and classification of revisions for argumentative writings. In this work, we extend our framework to integrate the automatic analyzer with a web-based interface to support *student argumentative writings*. The purpose of each change between revisions is demonstrated to the writer as a kind of feedback. If the author's revision purpose is not correctly recognized, it indicates that the effect of the writer's change might have not met the writer's expectation, which suggests that the writer should revise their revisions. The framework also connects the automatic analyzer with an interface for the instructor to manually correct the analysis results. As a side benefit, it also sets up an annotation pipeline to collect further data to improve the underlying automatic analyzer.

2 System Overview

The design of ArgRewrite aims to encourage students to concentrate on revision improvement: to iteratively refine the essay based on the feedback of the automatic system or the writing instructor.

Our framework consists of three components, arranged in a server client model. On the server side, the **automatic analysis** component extracts revision changes by aligning sentences across drafts and infers the purposes of the extracted revisions; this may reduce the writing instructor’s workload. On the client side, a web-based **rewriting assistant interface**¹ allows the student to retrieve the feedback to their revisions from the server, make changes to the essay and submit the modified essay to the server for another round of analysis. The interface is also accessible to the writing instructor and allows the instructor to have a quick overview of the students’ revision efforts. Another client side interface is a Java-based **revision correction** component², which allows the writing instructors to override the results of the automatic analysis and upload the corrected feedback to the server.

As demonstrated in Figure 1, the complete process of the student’s writing using our system starts with the student’s rewriting and submission of the essay. The student writes the first draft of the essay before using our system and then modifies the original draft in our rewriting assistant interface. The submitted writings are automatically analyzed immediately after the receipt of the student’s submission. Afterwards the instructor can manually correct the analysis results if necessary. The student can choose to view the analysis results immediately after the completion of automatic revision analysis or wait until the analysis results were corrected by the instructor. After receiving the analysis feedback, the student can choose to continue with the cycle of essay revising until the revisions are satisfactory.

3 Design of Components

3.1 Automatic analysis

Revision extraction. Following our prior work, we extracted revisions at the level of sentences by aligning sentences across drafts. An added sentence or a deleted sentence is treated as aligned to null. The

¹rewriting assistant interface: www.cs.pitt.edu/~zhangfan/argrewrite now supported on chrome and firefox browser only

²revision correction component: www.cs.pitt.edu/~zhangfan/revisionCorrection.jar. Tutorial to the web and java interface: www.cs.pitt.edu/~zhangfan/argrewrite/tutorial.pdf

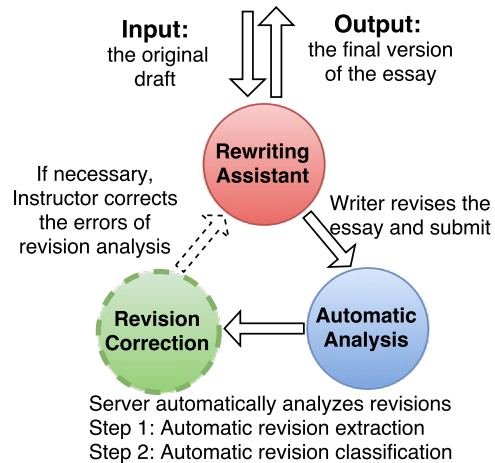


Figure 1: System structure of our rewriting assistance system.

aligned pairs where the sentences in the pair are not identical are extracted as revisions. We first use the Stanford Parser (Klein and Manning, 2003) to break the original text into sentences and then align the sentences using the algorithm in our prior work (Zhang and Litman, 2014) which considers both sentence similarity (calculated using TF*IDF score) and the global context of sentences.

Revision classification. Following the argumentative revision definition in our prior work (Zhang and Litman, 2015), revisions are first categorized to *Content (Text-based)* and *Surface*³ according to whether the revision changed the meaning of the essay or not. The *Text-based* revisions include *Thesis/Ideas (Claim)*, *Rebuttal*, *Reasoning (Warrant)*, *Evidence*, and *Other content changes (General Content)*. The *Surface* revisions include *Fluency (Word-usage/Clarity)*, *Reordering (Organization)* and *Errors (Conventions/Grammar/Spelling)*. On the basis of later work, the system includes the two new categories *Precision*⁴ and *Unknown*⁵. Using the corpora and features defined in our prior work, a multi-class Random Forest classifier was trained to automatically predict the revision purpose type for each extracted revision.

³The two types are defined in (Faigley and Witte, 1981).

⁴Revisions that make the essay more precise/accurate, e.g. “In the past” modified to “In the past 5 years”.

⁵For cases where the purpose of the revision is unrecognizable to the instructor, used when the instructor disagrees with the prediction of the automatic analysis component but cannot categorize the purpose to existing categories.

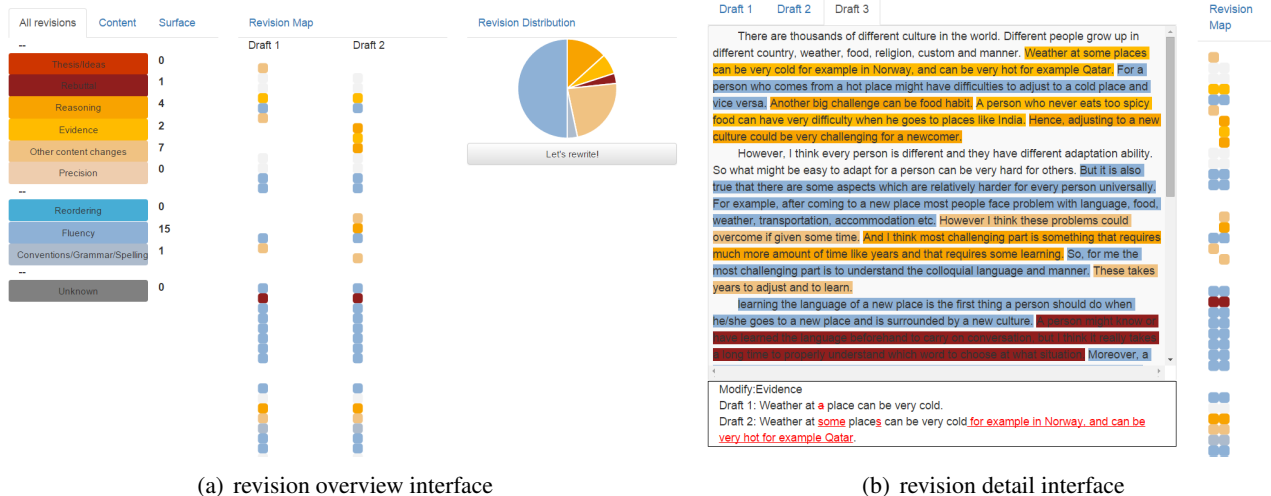


Figure 2: Screenshot of the web interface, which includes (a) the *revision overview* interface with the *revision statistics* (the numbers indicate the numbers of specified revision purposes) region, the *revision map* region and the *revision distribution* region, (b) the *revision detail* interface with the *revision text area* region and the *revision map* region (from left to right).

3.2 Rewriting assistant interface

Our rewriting assistant interface is designed with several principles in mind. 1) Because the revision classification taxonomy goes beyond the binary textual versus surface distinction, we want to make sure that users don't get lost distinguishing different categories; 2) We want to encourage users to think about their revisions holistically, not always just focusing on low-level details; 3) We want to encourage users to continuously re-evaluate whether they succeeded in making changes between drafts (rather than focusing on generating new contents). Thus, we have designed an interface that offers multiple views of the revision changes. As demonstrated in Figure 2, the interface includes a *revision overview* interface for the overview of the authors' revisions and a *revision detail* interface that allows the author to access the details of their essays and revisions.

Inspired by works on learning analytics (Liu et al., 2013; Verbert et al., 2013), we design the *revision overview* interface which displays the statistics of the revisions. Following design principle #1, the revision purposes are color coded and each purpose corresponds to a specific color. Our prior work (Zhang and Litman, 2015) demonstrates that only *Text-based* revisions are significantly correlated with the writing improvement. To inspire the writers to focus more on the important *Text-based* revisions, cold colors are chosen for the *Surface* revisions and

warm colors are chosen for the *Text-based* revisions. The statistics and the pie chart provide a quantitative summary of the writer's revision efforts. For example, in Figure 2, the writer makes many changes on the *Fluency* (15) of sentences but makes no change on the *Thesis/Ideas* (0). To allow the users to concentrate on improving one revision type at a time, the interface allows the user to click on a single revision purpose type and view only the specified revisions.

Following our design principle #2, the *revision map* in both interfaces presents an at-a-glance visual representation of the revision. This design is inspired by (Southavilay et al., 2013). Each sentence is represented as a square in the map. The left column of the map represents the sentences in the first draft and the right column represents the sentences in the second draft. The paragraphs within one draft are segmented by blanks in the map. The aligned sentences appear in the same row. The added/deleted sentences would be aligned to blank in the map. The revision map allows a user (either an instructor or a student) to view the structure of the essay and identify the locations of all the changes at once. For example, in Figure 2, the user can quickly identify that the writer aims at improving the clarity and soundness of the third paragraph by making a *Rebuttal* modification on the second sentence and *Fluency* modifications on all other sentences. The

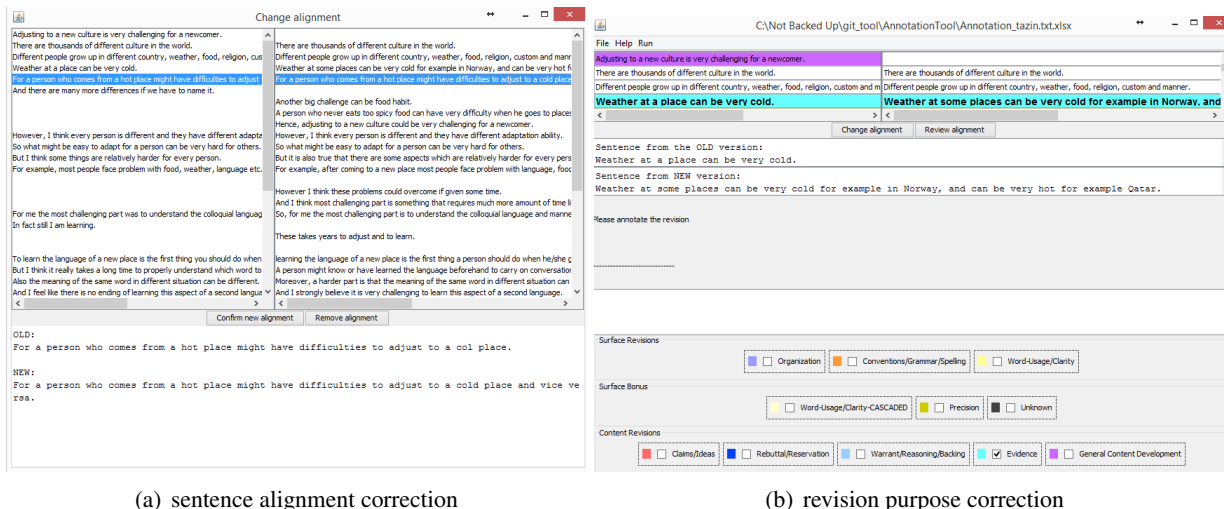


Figure 3: Screenshot of the correction interface, including the sentence alignment correction and revision purpose correction.

user can also click on the square to view the details of the revision in the *revision text area* region of the *revision detail interface*.

To encourage students to make revisions (design principle #3), in the *revision detail* interface the *revision text area* region highlights the revisions (color-coded by the revision categories) in the essay and allows the writer to modify it directly. The writer clicks on the text to read the revision and examine whether the revision purpose is recognized by the instructor/system. A character-level diff⁶ is done on the aligned sentences to help the writer identify the differences between two drafts. In the example the writer can see that their “Evidence” change is recognized, indicating that the revision effort is clear and effective. If the writer finds out that their real revision purpose is not recognized, they can modify the essay in the textbox directly and submit the essay to the server when all the edits are done.

3.3 Revision correction

The revision correction tool is developed for instructors only. The instructor loads the revision annotation files from the server, corrects the analysis results and uploads the corrections to the server. As demonstrated in Figure 3, the tool includes a *sentence alignment correction* interface and a *revision purpose correction* interface. The instructor first corrects the sentence alignment errors and then se-

lects the revision purposes for the re-aligned or mis-labeled sentence pairs. The correction actions of the instructors will be recorded and used to improve the analysis accuracy of the automatic analysis module.

4 Conclusion and Future Work

In this work we demonstrate a novel revision assistant for argumentative writings. Comparing to other assistants, the system focuses on inspiring writers to improve existing revisions instead of making new revisions. The system takes the writer’s drafts as the input and presents the revision purposes (analyzed manually or automatically) as the feedback. The writer revises iteratively until the purposes of the revisions are clear enough to be recognized.

In the future we plan to develop and incorporate the function of revision quality analysis, which not only recognizes the revision purpose, but also evaluates the quality of the revision (whether the revision weakly/strongly improves the essay). We are also about to begin a user study to evaluate the system.

Acknowledgments

We want to thank the members of the SWoRD and ITSpoke groups for their helpful feedback and all the anonymous reviewers for their suggestions.

This research is funded by NSF Award #1550635 and the Learning Research and Development Center of the University of Pittsburgh.

⁶google diff match: <https://code.google.com/archive/p/google-diff-match-patch/>

References

- Draft. 2015. Draft. <https://draftin.com/>. [Online; accessed 10-03-2015].
- Lester Faigley and Stephen Witte. 1981. Analyzing revision. *College composition and communication*, pages 400–414.
- Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.
- Ming Liu, Rafael A Calvo, and Abelardo Pardo. 2013. Tracer: A tool to measure and visualize student engagement in writing activities. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on*, pages 421–425. IEEE.
- Vilaythong Southavilay, Kalina Yacef, Peter Reimann, and Rafael A Calvo. 2013. Analysis of collaborative writing processes using revision maps and probabilistic topic models. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 38–47. ACM.
- Turnitin. 2016. Turnitin. http://turnitin.com/en_us/what-we-offer/revision-assistant. [Online; accessed 01-22-2016].
- Katrien Verbert, Erik Duval, Joris Klerkx, Sten Govaerts, and José Luis Santos. 2013. Learning analytics dashboard applications. *American Behavioral Scientist*, pages 1500–1509.
- Writelab. 2015. WriteLab. <http://home.writelab.com/>. [Online; accessed 10-03-2015].
- Fan Zhang and Diane Litman. 2014. Sentence-level rewriting detection. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–154, Baltimore, Maryland, June. Association for Computational Linguistics.
- Fan Zhang and Diane Litman. 2015. Annotation and classification of argumentative writing revisions. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 133–143, Denver, Colorado, June. Association for Computational Linguistics.