Shibani A. (2020) Constructing Automated Revision Graphs: A Novel Visualization Technique to Study Student Writing. In: Bittencourt I., Cukurova M., Muldner K., Luckin R., Millán E. (eds) Artificial Intelligence in Education. AIED 2020. Lecture Notes in Computer Science, vol 12164. Springer, Cham.
https://doi.org/10.1007/978-3-030-52240-7_52

Constructing Automated Revision Graphs: A novel visualization technique to study student writing

Antonette Shibani

University of Technology Sydney, NSW, Australia
Antonette.Shibani@uts.edu.au

Abstract. This paper introduces a novel technique of constructing *Automated Revision Graphs (ARG)* to facilitate the study of revisions in writing. ARG plots sentences of a written text as nodes, and their similarities to sentences from its previous draft as edges to visualize text as graph. Implemented in two forms: simple and multi-stage, the graphs demonstrate how sentence-level differences can be visualized in short texts to study revision products, processes, and student interaction with feedback in student writing.

Keywords: Natural Language Processing, Text Analysis, Writing, Revision, Automated Feedback, Automated Revision Graphs, Visualization

1 Introduction

With data and analytics permeating many aspects of teaching and learning, one area that increasingly uses its capabilities is writing. Writing Analytics makes use of natural language processing and machine learning techniques to assess, provide automated feedback and study student writing [1, 2]. One particular interest in writing analytics is in the study of revision to understand the written products and processes of students. Revision is an important process that contributes to the outcome of the writing by playing a recursive role of reworking and improving the writer's thoughts and ideas [3, 4]. Resource intensive manual observation and coding are now enhanced with advanced data collection and analytics techniques to seamlessly study this revision process. This is seen in recent automation efforts including the study of linguistic properties [5, 6] and visualizing revisions in student writing [7, 8].

However, there is a gap in existing methods to study revised texts and stages of revision in writing. Document-level metrics (such as cohesion, and other linguistic measures) [5] do not distinguish slight changes made to a base text, and require finer grained measures for shorter texts. On the other hand, key strokes and character editing in writing which are used to visualize and study patterns of revision [9-11] are too fine-grained to qualitatively study the actual changes made to the text. To meaningfully interpret what changes a student made to a given short text as a result of an intervention/ instruction, the need for automated visualizations to represent the process of drafting and revision at the sentence level arises. This need was identified from our research context where students engaged in a revision task using automated feedback from

AcaWriter [12] and provided consent for the use of their data as part of a writing intervention [13, 14]). The paper introduces a novel technique for visualizing text as graph called ‘Automated Revision Graphs’ (ARG) to study revisions at a sentence level for short texts, automating a previous manual prototype [8, 15]. It provides preliminary evidence to demonstrate its usage by generating two forms of ARG: 1) Simple revision graph, which compares two texts to visualize the differences, and 2) Multi-stage revision graph, which visualizes the evolution of a given text over its many drafts.

2 Simple Revision Graph

The first ARG form is a Simple Revision Graph comparing any two short texts (text 1 and text 2, both containing less than 15 sentences each) to visualize the differences between them at a sentence level. The *nodes* of the graph represented as circles denote individual sentences, and are displayed in their order of occurrence in the texts (e.g. Sentence 1, 2, 3,.., expanding downwards). The color of the node represents the text feature we are interested in, and can be adapted to suit different requirements. In the current research context, the node color signifies the number of rhetorical moves in the sentence as students receive automated feedback on this feature. A brown node indicates no rhetorical move made in that sentence, a blue node indicates one rhetorical move, and a green node indicates two or more rhetorical moves. The colored *edges* connecting two nodes in text 1 and text 2 show the similarity/ dissimilarity between the sentences represented by them. If there is a yellow edge between two nodes (sentences), it means that the two nodes are the same (no difference between the sentences). If the edge is teal colored, it indicates high similarity between the sentences (minor differences). A purple edge denotes medium similarity or major differences between the texts. A very small similarity means that the sentences are not related and very different (few or no common words between the sentences), and have no edges drawn between them. A sample simple revision graph with descriptions is provided in Fig 1a.

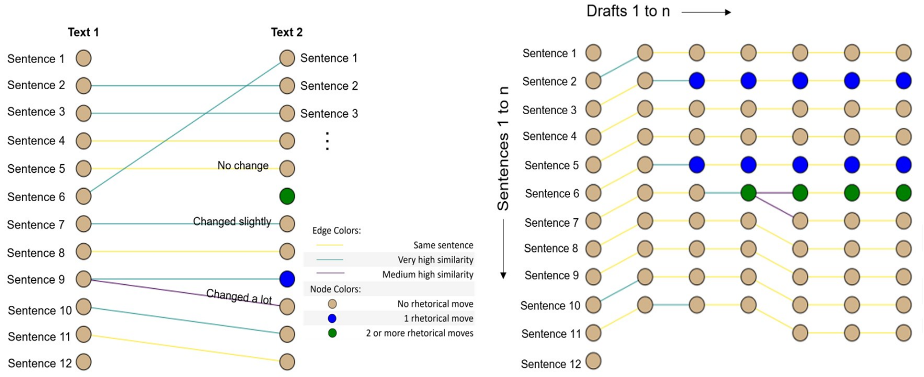

Fig 1: a) A simple revision graph example and **b)** Sample multi-stage revision graph with iterative changes.

A simple revision graph helps in studying the different kinds of changes students make at a sentence level on any given base essay. It can visually represent and quantify revision actions such as minor changes, major changes, additions and deletions made in the sentences of the given text, and the presence of rhetorical moves in the revised texts.

3 Multi-stage Revision Graph

The second ARG form of Multi-stage Revision Graph is similar to the simple revision graph described earlier, but extends over multiple text iterations. It is used to study the stages in the revision process over time by comparing one draft to its previous draft. A sample multi-stage revision graph is provided in Fig 1b, the student has removed the first and the last sentence from the given essay in the first draft requesting feedback (sentence 1 and sentence 12), depicted by missing outgoing edges. In the next draft, the student has introduced a rhetorical move represented by the blue colored node in sentences 2 and 5, with 2 or more rhetorical moves introduced in the subsequent draft in sentence 6 (represented by the green node). No major revisions have been made in the last two drafts as depicted by the unchanged graph structure towards the right end of the multi-stage revision graph.

The multi-stage revision graphs can be used to study the evolution of drafts in the revision process that led to the final product and student interaction with automated feedback based on the frequency of requests. They illuminate the underlying processes involved in the stages of revision after receiving automated feedback. These internal processes show how students apply the feedback on to their writing to revise the given text in different ways, which can be studied in relation to improvements in text quality.

4 Technical Implementation

Construction of ARG involved several steps, making use of Natural Language Processing (NLP) and graphical visualization packages in a Python Jupyter notebook. The code is released open source at https://github.com/AntonetteShibani/AutomatedRevisionGraphs for further development. An overview of steps is provided below:

- **Pre-processing the input text files:** The pre-processing step involved converting the input html files to extract the written text. The cleaned text was parsed to sentences using the TAP API [1], that provides NLP services such as sentence parsing, text metrics, and detection of rhetorical moves in text (More details at [12, 16]).
- **Getting rhetorical moves for all sentences:** The next step invoked Athanor from TAP to identify the rhetorical moves based on a concept-matching framework [17] (http://heta.io/online-training-in-rhetorical-parsing).
- **Creating the nodes from sentences:** The next step was to generate nodes for every sentence in the text and set its colour based on the number of rhetorical moves in

[1] GraphQL interface of the Text Analytics Pipeline (TAP): https://github.com/heta-io/tap hosted by Connected Intelligence Centre, University of Technology Sydney, Australia.

it. To do this, a nodes csv was created with an index for each node, its actual text (to display while hovering over), and the node category for defining its color.

- **Creating text vectors and calculating similarity scores between sentences:** Next, the edges were generated based on how similar the sentence in the revised text was, to sentences in the previous text, using a cosine similarity score. With no need for semantic similarity measures in the current context (as students were only asked to make structural changes, and not content changes), cosine similarity worked best.
- **Creating the edges based on similarities:** Based on the similarity scores calculated above, edges for the revision graph were created between the nodes of the given text and the revised text using set thresholds. If the similarity score was equal to or greater than the highest similarity threshold (>0.99 for the same sentence, >0.8 for highly similar sentences, >0.6 for medium similarity nodes), an edge was added between the nodes of the two sentences with the corresponding weight. The edges csv consisted of three columns: startnode, endnode and weight, appended for each edge.
- **Rendering the revision graphs:** The next step was to create and render the interactive ARG using the nodes and the edges csv created earlier. This was done using network graphs from a python library called HoloViews[2] with interactive exploration of nodes and edges facilitated by the Bokeh plotting interface[3]. The rendered revision graphs were saved as html files in the specified output folder.
- **Calculating metrics:** An optional step after generating the ARG is to collect quantifiable metrics from the network graph such as the number of nodes with a rhetorical move, number of edges showing absolute similarity with no changes etc.

5 Conclusion

This paper introduced a novel visualization technique of constructing Automated Revision Graphs (ARG) with open-source code to study revisions in student writing in two forms: simple and multi-stage. This visual representation can be used to examine the differences between short texts at a sentence level along with quantifiable metrics, and to study patterns of activities such as addition, deletion and re-organization of sentences in the revision of a given text (for validations with empirical student data, see [18]). In addition, they can be used to study the effects of automated writing feedback on students' revisions at iterative drafting stages by recognizing individual differences in the feedback literacy [19] of students. It can further inform research on the quality of revisions made by students in writing tasks [20] and influence design choices in writing tool development based on user engagement. Future work with improvements made to visual aspects and usability in this preliminary research form of ARG can potentially aid its usage among students and educators for reflecting on revision practices.

[2] HoloViews is an open-source Python library (https://github.com/pyviz/holoviews) to visualize graphs

[3] Bokeh is a Python visualization library to create interactive plots, dashboards, and data applications. More information at https://bokeh.pydata.org/en/latest/

Acknowledgements

An extended application of the work presented in this paper has been published in my doctoral thesis [18]. Thanks to Simon Buckingham Shum and Simon Knight for guiding the wider research project on automated writing feedback, which motivated the current work.

References

1. Shibani, A., Liu, M., Rapp, C., Knight, S.: Advances in Writing Analytics: Mapping the state of the field. In Companion Proceedings of the 9th International Conference on Learning Analytics & Knowledge (LAK19), Tempe, Arizona (2019)
2. Buckingham Shum, S., Knight, S., McNamara, D., Allen, L., Bektik, D., Crossley, S.: Critical perspectives on writing analytics. Workshop at the Sixth International Conference on Learning Analytics & Knowledge, pp. 481-483. ACM (2016)
3. Fitzgerald, J.: Research on revision in writing. Review of educational research 57, 481-506 (1987)
4. Flower, L., Hayes, J.R.: A cognitive process theory of writing. College composition and communication 32, 365-387 (1981)
5. McNamara, D.S., Graesser, A.C., McCarthy, P.M., Cai, Z.: Automated evaluation of text and discourse with Coh-Metrix. Cambridge University Press (2014)
6. McNamara, D.S., Crossley, S.A., McCarthy, P.M.: Linguistic features of writing quality. Written communication 27, 57-86 (2010)
7. Zhang, F., Hwa, R., Litman, D.J., Hashemi, H.B.: ArgRewrite: A Web-based Revision Assistant for Argumentative Writings. In: NAACL-HLT 2016 (Demonstrations), pp. 37-41. (2016)
8. Shibani, A., Knight, S., Buckingham Shum, S.: Understanding Students' Revisions in Writing: From Word Counts to the Revision Graph. Technical Report, Connected Intelligence Centre, University of Technology Sydney (2018)
9. Caporossi, G., Leblay, C.: Online writing data representation: a graph theory approach. In: International Symposium on Intelligent Data Analysis, pp. 80-89. Springer, (2011)
10. Southavilay, V., Yacef, K., Reimann, P., Calvo, R.A.: Analysis of collaborative writing processes using revision maps and probabilistic topic models. In: Proceedings of the third international conference on learning analytics and knowledge, pp. 38-47. ACM, (2013)
11. Wininger, M.: Measuring the evolution of a revised document. Journal of Writing Research 6, (2014)
12. Knight, S., Shibani, A., Abel, S., Gibson, A., Ryan, P., Sutton, N., White, R., Lucas, C., Sandor, A., Kitto, K., Liu, M., Mogarkar, R.V., Buckingham Shum, S.: AcaWriter: A Learning Analytics Tool for Formative Feedback on Academic Writing. Journal of Writing Research (2020, Forthcoming)
13. Shibani, A., Knight, S., Buckingham Shum, S., Ryan, P.: Design and Implementation of a Pedagogic Intervention Using Writing Analytics. In: Chen et al., W. (ed.) 25th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education, New Zealand (2017)
14. Shibani, A., Knight, S., Buckingham Shum, S.: Contextualizable Learning Analytics Design: A Generic Model, and Writing Analytics Evaluations. In Proceedings of the 9th International Conference on Learning Analytics and Knowledge (LAK'19). ACM, Tempe, Arizona (2019)
15. Shibani, A., Knight, S., Buckingham Shum, S.: Understanding Revisions in Student Writing through Revision Graphs. 19th International Conference on Artificial Intelligence in Education. Springer, London, UK (2018)

16. Shibani, A., Abel, S., Gibson, A., Knight, S.: Turning the TAP on Writing Analytics. In Companion Proceedings of the 8th International Conference on Learning Analytics and Knowledge. (2018)
17. Sándor, Á.: Modeling metadiscourse conveying the author's rhetorical strategy in biomedical research abstracts. Revue française de linguistique appliquée 12, 97-108 (2007)
18. Shibani, A.: Augmenting Pedagogic Writing Practice with Contextualizable Learning Analytics. PhD Thesis. Connected Intelligence Centre. University of Technology Sydney, Sydney, Australia (2019)
19. Carless, D., Boud, D.: The development of student feedback literacy: enabling uptake of feedback. Assessment & Evaluation in Higher Education 43, 1315-1325 (2018)
20. Afrin, T., Litman, D.: Annotation and classification of sentence-level revision improvement. arXiv preprint arXiv:1909.05309 (2019)