Organic Visualization of Document Evolution

Ignacio Perez-Messina  
Computer Science Dept.  
Universidad de Chile  
Santiago, Chile  
iperez@dcc.uchile.cl

Claudio Gutierrez  
Computer Science Dept.  
Universidad de Chile  
Santiago, Chile  
cgutierrez@dcc.uchile.cl

Eduardo Graells-Garrido  
Data Science Institute  
Universidad del Desarrollo  
Santiago, Chile  
egraells@udd.cl

ABSTRACT
Recent availability of data about writing processes at keystroke-granularity has enabled research on the evolution of document writing. A natural task is to develop systems that can actually show this data, that is, user interfaces that transform the data of the process of writing –today a black box– into intelligible forms. On this line, we propose a data structure that captures a document’s fine-grained history and an organic visualization that serves as an interface to it. We evaluate a proof-of-concept implementation of the system through a pilot study using documents written by students at a public university. Our results are promising and reveal facets such as general strategies adopted, local edition density and hierarchical structure of the final text.

INTRODUCTION
Writing is an old and relevant human skill whose standard product is text. Due to the widespread use of information technologies, most text available today is stored in digital platforms, such as the Web. Although static in their final form, documents –collaborative ones in particular– are works-in-progress, meaning they are still subject to their writing process, and thanks to versioning systems, they are being stored as such. Currently, we do not fully understand the importance of this data, even though the way we write is tied to how we learn and structure our knowledge [2]. This may be in part because, as stated by Grésillon and Perrin in the Handbook of Writing and Text Production, “The written (the product) aims at overcoming the writing (the process)” [6], which means that the better the quality of a text, the less we know about its development.

This lack of understanding, and of research on this topic, may be due to the fact that data was not easily available. But today, Web services such as Google Drive keep records of document changes at keystroke-level, so as data has become widely available, a new avenue of research has opened that could have a great impact on natural language processing and generation, namely the study of text in its temporal, mutable form.

A vast work on text visualization is available, mainly addressing text corpora exploration and classification while there is little research on individual documents and their production process (the most extensive survey on text visualization [11] does not consider versioned text as a specific data source). On the latter, most work focuses on collaboration at the coarse granularity of versioned text (e.g., Wikipedia content) and is not suited for research on individual writing.

In this paper, we propose an interactive visualization method in the largely unexplored field of fine-grained text production data. With an organic design that mimics plant metabolism and behavior, the proposed visualization shows the whole fine-grained history of a document in one image (a plant) and displays its development in time with animation (its growth). It also provides access to its textual content interactively, through which a naturally occurring segmentation of the text (the plant’s branches) can be produced. By allowing the complex behavior of producing text to visually emerge, it fosters exploration of its structure and evolution through time, which could reveal the information hidden in fine-grained versioning data.

We evaluated our interactive visualization through a pilot study, where we visualized and analyzed documents written by engineering students. The results show different characteristics of the writing process that emerged from the visualization: general strategies adopted, local edition density, and global text structure. These results evoke interesting applications for automated text analysis such as new text segmentation algorithms and document classifiers based on writing process rather than word choice. We conclude that the study and visualization of fine-grained text data enables a deeper understanding of documents, as it permits to augment the final product with the trace of the decisions performed during its production.

BACKGROUND
Currently most user interfaces for text visualization focus on finished text, i.e., the final product of the writing process of an agent, or a corpus of such products (visual text summarization [7], topic modeling [15], mapping content structure [18], recommender systems [5], among others [16]).

There is little work on visualization of the process itself that generates a document, particularly in the case of the human writing process [6]. The challenge here is to understand the structure and evolution of text according to several updates, each of which may add new content and/or delete prior content.
Currently, there are two prominent sources for this kind of visual interaction: collaborative user-generated content and individual research-generated.

Regarding collaborative content, the most common data source is Wikipedia, with tools like History Flow [19], that visualizes the revision history of Wikipedia articles, and the Notabilia project [17], which visualizes collective deliberation. DocuViz [20] applied the History Flow approach to collaborative documents in Google Docs and Kim et al. [8] proposed using only document deltas in this same line of visualization.

There are studies that focus on individual writing process [12]. Perrin and Wildi developed a statistical method to infer writing phases using cursor movement data [14]. Caporossi and Leblay [1] showed a graph-based visualization of the writing of a paragraph with data from ScriptLog (a keystroke logging program), where nodes represent operations; and edges their topological and temporal relations.

Evolution of single documents has thus been researched either from a collaborative, large-scale perspective using coarse data or from an individual, fine-grained one but only at very small scale. To the best of our knowledge, there is no visualization in between that encompasses these dimensions as a whole, therefore uniting writing process research. The system we describe next aims at filling this gap.

SYSTEM DESIGN
Following Wise’s ecological [21] and Fry’s organic information design [4] approaches, we created a plant-like structure of interdependent units. With this design, a document is not shown as something linear and static, but rather as emergent and dynamic; it grows and dies as it is produced. We implemented our prototype using the Processing language [3] as it is commonly used for organic visualization systems and data from Google Docs.

Here we describe the different stages of the pipeline needed to arrive at such depiction: the definition and grouping of text operations, the data structure holding those operations, and the visual design that shows the data structure. We denote by agent the writing agent and leave the word user for the user of the system.

Operation Grouping and Locality Criteria
We define a document as a chain of atomic (distinguishable in time) operations (insertions and deletions) and group these operations according to a locality criteria. As the retrieved operations usually correspond to single keystrokes, we group adjacent ones into the largest locality-preserving chain (i.e., in such a way that no voluntary change in cursor position takes place between any two of these operations), thus resulting in larger operations we call bursts. Bursts are more coherent and significant than single keystrokes because they allow undoing as long as the agent does not actively change cursor position, so correction of typographical errors and low-level information of the writing process [6] are filtered out. Bursts are then rearranged spatially according to their insertion position in the document (in the same way as in Kollberg’s $S$-notation [10]).

Data Structure for Text Evolution
We store the operations and their spatial relations as a Planar Directed Acyclic Graph in which internal nodes represent operations, leaves represent Places of Insertion (POIs, which are the points between characters and elements where the blinking cursor can be), and edges are topological relations between them and direction follows the arrow of time (see Fig. 1). An empty document starts as a (root) node leading to a single leaf, at which point only an insertion can take place (as a deletion needs more than one POI to take place). When the agent types, an operation node containing the inserted string replaces the null node at the leaf, from which $n + 1$ edges emerge, where $n$ is the number of characters inserted. Each of these edges contains a new POI and the POI where the insertion took place is no longer reachable since it becomes an internal node.

The aforementioned process goes recursively, always maintaining a tree structure. This, however, does not hold when considering deletions. In the case of a deletion of $m$ characters, the $m + 1$ corresponding adjacent edges are bundled together into a single operation node with one leaf. Note that this radically changes the original insertions-only tree structure since a deletion may encompass many levels of the hierarchy.

Visual Representation
We used a glyph-based approach to visualize the aforementioned data structure, in which glyphs (the visual unit that conveys meaning) act as interdependent units and build upon each other as stems of a plant. Intuitively, an insertion “opens up” space in the document, by splitting one leaf into many, while deletions “close” it, by joining many leaves back into one. The glyph designed to represent insertion nodes is, therefore, a stylized multiplexer. Deletion nodes, on the other hand, do not have their own glyph but retroactively affect insertion glyphs (changing the look of glyphs that were already placed). Figure 2 illustrates this.

Seeing the visualization as a mapping from the data structure to the visual space, the rules that define this mapping are:

1. For each insertion node, there is one glyph that represents it and its first-level out-edges.
2. An edge leading from an insertion node to another means that the correspondent glyphs are related, precisely the latter is placed on top of the former, at the position corresponding to its relative POI within its parent.

3. An edge leading to a deletion node changes the glyph as shown in Figure 2.

**Aesthetics.** Preliminary prototypes showed the need for decreasing glyph overlap: to avoid, at the same time, straight and spiraling branches. Thus, a “phototropism factor” is applied to the growth of the tree, mimicking the plant behavior of growing towards a light source. Time is represented using a cyclical eight-color categorical palette (see Figure 2, right): nodes are colored according to the session (considered here as a day of writing) in which they were added. The radius of the arc doubles in case its center angle was to surpass π.

**Interaction.** “Phototropism” and arc length-node size ratio can be dynamically manipulated to globally change the shape of the tree and improve visibility. When a glyph is selected, the textual contents of its branch (deleted and active children) are displayed on the screen in a notation similar to S-notation [13]. Left-clicking hides/shows the branch.

**PILOT STUDY**

We performed a pilot study, where computer science students from a public university were asked to share their documents written in Google Drive. In total, we obtained 60 documents of different lengths (from a few paragraphs to full-length articles) and purposes (though they were all course assignments). Most of them were ruled out before visual analysis due to being incomplete or too complex. The five documents shown here were selected due to their complementariness (see Table 1). For each one, we identified the visualization’s branching structure, which leads to a hierarchic segmentation of the tree. Then, we inspected each branch’s content and identified which part of the document corresponded to the branch. We also took note of branch length and breadth, and important deletions, which we interpreted in the context of each document.

There are three dimensions of document evolution that, according to the analysis, are well captured by our system:

- **The internal organization of the text and its hierarchic structure** (Fig. 3). We observed that branches of a tree mostly correspond to the global text structure, which is a direct consequence of the proposed data structure. In Cases A and C, branches match paragraph divisions, as they have no other hierarchical level. Cases B and E have a typical hierarchical organization (cover information, sections, and bibliography) which is perfectly matched by the relations of the correspondent branches. Case D has also no more structure than paragraph-level as can be intuited by its “one big branch” appearance.

- **Some patterns and strategies adopted by agents** (Fig. 4). Thanks to the particular data structure and the visual representation of deletions, the strategies used to write the document can be observed. Cases B and E show a well-defined hierarchical structure, meaning their writing bore the final structure in mind from the beginning, something that can be expected in a course assignment. Case C shows a draft that was rewritten and erased (similarly, D shows the rewriting of pasted text), while A was written almost linearly, without important deletions.

- **The amount of work put into the document and its different parts** (Fig. 5). This dimension emerges from the color scheme and the grouping of operations into bursts. Case studies A, B, C and E have branches of only one color, meaning they were introduced during one session with no later rewriting, whereas D has branches showing many appendices of different colors, meaning they were reread and edited in posterior sessions. Moreover, the highest edition

![Figure 2. Glyph scheme of two related nodes (left) and cyclic color palette (right). A glyph is composed by an arc (b, f), which is composed by the node’s out-edges (making its length proportional to the characters inserted), and a support line (a, e). When a string is deleted, the correspondent part of the arc loses opacity and falls toward the center (c). Children nodes are placed as coming out from the POI they originated from (d). In this example, an insertion of size 15 was followed by a deletion of size 3 at position 10, and then writing was resumed at the end of the document.](image)

![Figure 3. Analysis of document B. From the raw visualization (a) we identified its branching structure (b. The colors were manually added). Inspecting each branch’s content produces a correspondent segmentation of the text (c. Highlights were manually added to match correspondent colors in b). This shows that the branching structure of the tree is the same as the hierarchic structure of the document: the cover title splits into two, the bibliography and the body, which splits also into three sections.](image)

**Table 1. Description of document case studies.**

| Doc | Description                      | Words | Operations |
|-----|----------------------------------|-------|------------|
| A   | Two-item summary                 | 312   | 1307       |
| B   | Three-question assignment        | 1567  | 7136       |
| C   | One-question assignment          | 657   | 3015       |
| D   | Unstructured essay               | 5242  | 15411      |
| E   | Structured assignment            | 1135  | 4350       |
Figure 4. Visualization of documents A, C and E. Note that the branching structure in each case was manually highlighted. A shows two branches, corresponding to its two paragraphs. C features four main branches, one of which (the faint-looking one) was completely deleted and from which the other branches arise. E is more complex, each branch mapping to a section of the text (same as case B). These cases show different writing strategies: almost linear (A), draft and rewrite (C), and hierarchically structured (E).

Figure 5. Analysis of D. Very long arcs (a) denote copy/pastes. Branches extending from these arcs (b) are their rewriting (note that the largest arc was almost completely deleted) and have the highest density of changes. Little branches of different colors (c) are later additions.

density is concentrated around the deletion of a large piece of text that was pasted from another document.

In summary, we observed that the system captures important components of the writing process, mainly through its underlying data structure and its decentralized depiction of operations rather than whole versions.

DISCUSSION AND CONCLUSIONS
Our results shed light on the dynamic origins of text and the structures underlying the process of writing. These findings could be useful in natural language processing, e.g., by including human-writing processes into automated text generation, topic segmentation or document summarization. A direct application of our system is a real-time writing-aid in document writing tools, which returned to the document its heterogeneity, for example, showing the relative age of parts of a text, their need for updating and the thread they belong to.

Scope and Future Work. A rightful critique is that, owing to its lack of a different glyph for deletion nodes, the visualization captures only a subset of the data structure, i.e., it is only a spanning tree of the whole graph, which leads to the non-uniqueness of a document’s representation: a design fault because it forces a degree of freedom not present in the data [9]. Future work, then, should include the design of deletion nodes so that they play a structural role. Also, having a way to differentiate agents would capture on-line collaborative behavior, which is a very appealing line of study. Displacing text (cut and paste actions inside the document) is not yet captured by the system due to that cutting/pasting is not a specific type of operation in the data, so further preprocessing is required to recognize this behavior. Finally, the pilot study showed that coloring a tree by its branching structure is important for analysis, so a useful feature would be the automatic segmentation of text by branching structure and the consequent visual mapping from the tree to the document.

Conclusions. We have presented a novel visualization design for document evolution, which combines an operational view of the document with an organic visual scheme, and shown that it renders visible some facets of the complex behavior in the process of writing. It can be used, for example, to get an overview of the whole of a document’s history in a single image, which is enough to give an idea of the amount of work put into it and the general strategy adopted, and as an interface to its content. Examples of such strategies are rewriting from a draft, writing with a structure in mind, one-vs. many-session writing, etc. These features are something that, for a single session or single agent document and at this level of granularity, to the best of our knowledge, available systems cannot provide. Also, with its interactive functions, the system can be used to produce a segmentation of a document, which in some cases coincides with its hierarchical structure, but in any case is a naturally occurring segmentation which follows the thread of thought of the agent. We present this approach and system to provide an integration of computer-aided writing research by proposing a clear focus on the document as a well-defined temporal object. Text analysis should not be abstracted from a document’s history when possible, and this approach proves a fair candidate for a first step towards, what we’d like to call, diachronic text analysis.

ACKNOWLEDGEMENTS
The authors would like to thank Tobias Schreck for a fruitful discussion about the visual representation, and James Somers for the knowledge on how to download data from Google Docs. Claudio Gutierrez also thanks CIWS.
REFERENCES
1. Gilles Caporossi and Christophe Leblay. 2011. Online writing data representation: a graph theory approach. In International Symposium on Intelligent Data Analysis. Springer, 80–89.
2. Janet Emig. 1977. Writing as a mode of learning. College composition and communication 28, 2 (1977), 122–128.
3. Ben Fry. 2007. A Processing: Programming Handbook for Visual Designers and Artists. MIT Press.
4. Benjamin Jotham Fry. 2000. Organic information design. Ph.D. Dissertation. Massachusetts Institute of Technology.
5. Eduardo Graells-Garrido, Mounia Lalmas, and Ricardo Baeza-Yates. 2016. Data portraits and intermediary topics: Encouraging exploration of politically diverse profiles. In Proceedings of the 21st International Conference on Intelligent User Interfaces. ACM, 228–240.
6. Almuth Grésillon and Daniel Perrin. 2014. Methodology: From speaking about writing to tracking text production. Handbook of writing and text production 10 (2014), 79–111.
7. Enamul Hoque and Giuseppe Carenini. 2015. Convisit: Interactive topic modeling for exploring asynchronous online conversations. In Proceedings of the 20th International Conference on Intelligent User Interfaces. ACM, 169–180.
8. Seungyeon Kim, Joshua V Dillon, and Guy Lebanon. 2012. Cumulative Revision Map. arXiv preprint arXiv:1205.3205 (2012).
9. Gordon Kindlmann and Carlos Scheidegger. 2014. An algebraic process for visualization design. IEEE transactions on visualization and computer graphics 20, 12 (2014), 2181–2190.
10. Py Kollberg and Kerstin Severinson Eklundh. 2002. STUDYING WRITERS’ REVISI NG PATTERNS WITH S-NOTATION ANALYSIS. Studies in Writing 10 (2002), 89–104.
11. Kostiantyn Kucher and Andreas Kerren. 2015. Text visualization techniques: Taxonomy, visual survey, and community insights. In Visualization Symposium (PacificVis), 2015 IEEE Pacific. IEEE, 117–121.
12. Muhammad M Abdel Latif. 2008. A state-of-the-art review of the real-time computer-aided study of the writing process. International Journal of English Studies 8, 1 (2008), 29–50.
13. Daniel Perrin. 2003. Progression analysis (PA): Investigating writing strategies at the workplace. Journal of Pragmatics 35, 6 (2003), 907–921.
14. Daniel Perrin and Marc Wildi. 2009. 28 Statistical modeling of writing processes. Traditions of writing research (2009), 378.
15. Carson Sievert and Kenneth E Shirley. 2014. LDAvis: A method for visualizing and interpreting topics. In Proceedings of the workshop on interactive language learning, visualization, and interfaces. 63–70.
16. Artur Šilić and Bojana Dalbelo Bašić. 2010. Visualization of text streams: A survey. In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Springer, 31–43.
17. Dario Taraborelli and Giovanni Luca Ciampaglia. 2010. Beyond notability. Collective deliberation on content inclusion in Wikipedia. In Self-Adaptive and Self-Organizing Systems Workshop (SASOW), 2010 Fourth IEEE International Conference on. IEEE, 122–125.
18. Frank Van Ham, Martin Wattenberg, and Fernanda B Viégas. 2009. Mapping text with phrase nets. IEEE transactions on visualization and computer graphics 15, 6 (2009).
19. Fernanda B Viégas, Martin Wattenberg, and Kushal Dave. 2004. Studying cooperation and conflict between authors with history flow visualizations. In Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 575–582.
20. Dakuo Wang, Judith S Olson, Jingwen Zhang, Trung Nguyen, and Gary M Olson. 2015. DocuViz: visualizing collaborative writing. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM, 1865–1874.
21. James A Wise. 1999. The ecological approach to text visualization. Journal of the Association for Information Science and Technology 50, 13 (1999), 1224.