Phase to Phase
Developing an Automated Procedure to Identify and Visualize Phases in Writing Sessions Using Keystroke Data

Rianne Conijn[1,*], Alessandra Rossetti[2], Nina Vandermeulen[3] & Luuk Van Waes[3]

[1.] University of Technology, Eindhoven | The Netherlands

[2.] Vrije Universiteit Brussel | Belgium

[3.] University of Antwerp | Belgium

Abstract: Understanding the temporal organization of writing is key to studying writing processes. Existing methods to segment writing into phases often rely on arbitrary rules, extensive manual annotation, or focus on numerous transitions. This study aimed to develop an automated segmentation method to detect distinctive transition in the dominant writing process, particularly the transition from first draft to revision. For this, keystroke data (source-based L1 writing (N = 80) and text simplification in L2 (N = 88)) were manually annotated. The BEAST algorithm was applied for Bayesian change point detection, based on five characteristics derived from the annotation criteria: (1) percentage of the final text written so far, (2) distance between typed and remaining characters, (3) relative cursor position, (4) source use, and (5) pause timings. The first three features proved most effective in identifying change points. A rule-based approach was further applied to select one final change point, which resulted in mediocre accuracy ranging from 31% exact agreement to 49% agreement within 60 seconds. To conclude, the BEAST algorithm is useful in detecting a variety of change points in writing processes, yet connecting them to meaningful phases is still quite complex.

Keywords: Keystroke logging, writing process, change point detection, revision phase, drafting

journal of WRITING RESEARCH

Conijn, R., Rossetti, A., Vandermeulen N., & Van Waes, L. (2025). Phase to phase: Developing an automated procedure to identify and visualize phases in writing sessions using keystroke data. *Journal of Writing Research, 17*(2)*,* 339-369. DOI: https://doi.org/10.17239/jowr-2025.17.02.06

Contact: Rianne Conijn, Human-Technology Interaction Group, Eindhoven University of Technology, PO Box 513 5600 MB Eindhoven | The Netherlands – m.a.conijn@tue.nl.

1. Introduction

In the analysis of writing, the temporal organization of the writing process and sub processes — such as planning, drafting, revising, or consulting external sources — has only received limited attention. Van den Bergh & Rijlaarsdam (2001) were among the first researchers to clearly explain and demonstrate the importance of a more time-based approach to writing process research. Their focus was motivated by the fact that the task situation gradually changes as the writing process evolves, with the introduction and revision of new ideas becoming progressively less frequent as writers move towards a final version (Lo Sardo et al., 2023). This also has an effect on the amount of text added and/or revised over time, especially when a composition task is divided over multiple sessions (Bowen & Van Waes, 2020). Additionally, the temporal organization of writing is influenced by the writers' profiles, i.e. depending on the writer's preferences and experiences, writers will distribute cognitive activities differently (Van den Bergh & Rijlaarsdam, 2001; Van Waes & Schellens, 2003). Examining the temporal organization of the writing process can also provide key insights into self-regulation mechanisms (Saqr et al., 2021) and metacognitive strategies (Huang & Zhang, 2022). Furthermore, certain temporal distributions of tasks in the writing process are also predictive of text quality levels (Xu, 2018).

An often used method to examine the temporal organization of the writing process is to divide the writing process into phases, stages, segments, or episodes (e.g., Leijten et al., 2014; Sala-Bubaré et al., 2021; Xu & Xia, 2021). It is important to note here that this temporal approach to segmenting the writing process does not imply that composing texts is a linear process. Despite the continuous non-linearity that characterizes writing, a temporal approach adds an important perspective to the analysis of writing processes. This perspective is exemplified in the study by Xu & Xia (2021). They showed that overall writing time in second language writing is not affected by writing expertise. However, when the writing process was divided into three traditional sequential phases: prewriting/planning, formulation, and revising/reviewing, differences were found: Novice writers mainly focus on formulation, limiting the other phases, while more proficient writers distribute the time spent on the different phases more equally.

For examining the temporal organization of writing, logging tools (e.g., keystroke logging and/or eye-tracking) are key as they allow for the collection of temporal data on mental activities that might not be (fully) available through introspection (Torrance & Conijn, 2024). Although a variety of approaches exist into segmenting keystroke log data, the operationalizations are often relatively arbitrary (e.g., segment the writing into three segments of equal duration) or require extensive manual labelling (S. Li & Yu, 2024). Accordingly, in this study we aim to develop an automated segmentation of writing processes that focuses on a distinctive transition in the dominant writing processes, based on keystroke data.

2. Related work

The operationalization of time-based segmentation approaches differs across studies, with respect to the level of focus and the level of automatization. In addition, some approaches are more suitable for multi-session processes, while others are solely built for single-session processes. In the following, we distinguish four different types of segmentation approaches: time-based, content-based, version-based, and function-based.

2.1 Time-based segmentation

As stated above, Van den Bergh and Rijlaarsdam (2001) were among the first researchers who stressed the importance of a temporal approach to studying the writing process. In their thinking-aloud study, they modeled the occurrence of cognitive activities (orientation and planning activities versus formulating activities) as a function of the time elapsed since the start of the task. Including the total time elapsed so far or temporal location has subsequently been the approach in a variety of keystroke logging studies (see e.g., Zhang et al., 2016).

Rather than using time directly into modelling the writing process, time-based approaches have been used to segment the keystroke log into several time-based segments. The most common approach is to divide the process into three segments of equal length (see e.g., De Lario et al., 2006; Tarchi et al., 2023), arguably because it can be easily referred to as the 'start', 'middle', and 'end' of the writing process. Other studies have looked into more segments, including five (Leijten et al., 2019) and ten equal time intervals (Van Waes & Leijten, 2015). In the five intervals study, the authors used the first interval as a proxy for the initial planning phase, and the last interval as a proxy for the second draft, revision phase (Leijten et al., 2019).

The main advantage of this interval approach is that it is intuitive and easy to automate. Inputlog (Leijten & Van Waes, 2013), a commonly used program to collect keystroke data, already provides a default option to split the writing process into intervals of equal length, where the number of segments can be defined by the user. Moreover, the approach facilitates the comparison of writing processes of different lengths. A downside of this approach is that time segments of different total lengths are compared. In addition, there is no clear consensus or rationale for an 'optimal' number of segments, and the question is if such an optimum would even exist. Regardless of the number of chosen segments, the distinction can be considered quite arbitrary, as it does not take into account the different activities performed within the writing process (e.g., a segment might consist of no keystrokes), different functions (e.g., whether the dominant process was drafting an outline or making post-draft revision), or the content written (cf. Xu & Xia, 2021).

2.2 Content-based segmentation

Rather than segmenting the writing process based on time, others have segmented the process based on the content written. Similar to including the total time elapsed so far, the total amount of characters produced so far can be used as a function to model the temporal aspect of the keystroke log. Some content-based segmentation approaches go beyond the number of characters produced towards the actual content produced. For example, Sala-

Bubaré et al. (2021) manually distinguished different sections of an extended abstract in doctoral writing: title, introduction, objective, method, results, discussion, and sources. Segments that included two or more sections were labeled as global. These segments were then connected to activities in the writing process. Content-based segmentation is more common in analyses focusing on the writing product (linguistic analyses) rather than the writing process. For example, Crossley et al. (2022) used a content-based segmentation, in which they extracted different argumentation elements, such as primary claim, final claim, counterclaim, rebuttal, data, and concluding summary. Here they connected the incidence of these argumentative features with writing quality.

The advantage of these content-based approaches is that it connects the writing process directly with the writing product. In addition, some of the segmentation might be done automatically by using natural language processing approaches. However, the disadvantage of these approaches is that they are very time-intensive if done manually. Both automated (with NLP) and manual approaches are especially difficult if many revisions are made during the writing process, as it is not always clear what writer is writing about if only part of a sentence is written and then revised (cf. Mahlow et al., 2022).

2.3 Version-based segmentation

Related to content-based approaches are the version-based segmentations. These types of segmentations explicitly distinguish between different versions created by intermediately saving the text-produced-so far or by ending/starting a new writing session. A version has been defined as: "a point in the production history of a text that is deemed relevant based on particular criteria, a version is thus a specific text-produced-so-far" (Mahlow et al., 2022, p. 450). A version can be initiated by the writer or by the system (auto-save). For example, some studies have used automated versioning built into existing editors (see e.g., Lo Sardo et al., 2023). Here the authors used two different types of text editors, with Google docs creating a version every minute (if any writing activity took place), and WeWrite which created a new version every 3 minutes.

In writer-based versioning, the writer actively saves a version themselves (see e.g., the study on Wikipedia revision by Daxenberger & Gurevych, 2013). For example, Leijten et al. (2014) segmented the writing process of a single author in professional communication into five different segments, based on the five separate writing sessions the author employed. In a comparable way, Bowen & Van Waes (2020) created segments based on different sessions when students worked on their essays. In this qualitative study two essays were finalized after seven sessions; and one essay after five sessions. Interestingly, the version-based approaches was not the endpoint in both studies: Leijten et al. (2014) further divided the sessions into beginning, middle, and end of the writing process (time-based segmentation), while Bowen & Van Waes (2020) decided to further segment the sessions using a time-based approach based on the temporal ordering of the revisions. Rather than subdividing the sessions, Buschenhenke et al. (2023) chose to combine sessions, using clustering. In their study, 386

writing sessions were identified in the writing of a complete novel, which were subsequently combined into nine clusters based on the non-linearity characteristics within the sessions.

The advantage of focusing on saved versions or sessions is that it does not rely on keystroke logging, hence it can be implemented relatively easily. A downside of automated versioning is that automated versioning does not reflect the writer's process management or their decision to segment it. Further, both automated versioning and writer-based versioning might not necessarily fit the researcher's aim. In addition, the writing sessions might differ substantially in total duration of the sessions and in the content created within the sessions. For example, Buschenhenke et al. (2023) found sessions ranging from a couple of minutes to over three hours. For some of the sessions, the time between sessions was very short, indicating that some of these sessions might be combined. In addition, some of the longer sessions included substantial idle time (over 30 minutes), indicating that these sessions might be split. The fact that merely distinguishing the writing process based on sessions is not always enough is also visible from the fact that these sessions have been further subdivided (Van Waes et al., 2014), grouped (Bowen & Van Waes, 2020; Buschenhenke et al., 2023).

Accordingly, some researchers did not follow the authors' versioning, but focused on researcher-based versioning created post-hoc, often based on some additional process characteristics obtained from keystroke logging. For example, Cislaru & Olive (2018) identified the start of a new version based on a long pause, while Mahlow (2015) automatically defined versions based on changes in the production. A new version was identified if the writer switched from continuous writing to continuous deletion or insertion. The latter could already be seen as a more function-based approach, distinguishing between text production and revision.

2.4 Function-based segmentation

Function-based segmentations indicate a segmentation based on a dominant (cognitive) writing process or activity. A common distinction is between the three most-cited writing (sub)processes planning, translating, and reviewing (Flower & Hayes, 1981). This is often based on manual annotation, where process visualizations such as Inputlog's process graph (Leijten & Van Waes, 2013), are utilized as a resource to aid the annotation. For an example of the process graph, see **Fout! Verwijzingsbron niet gevonden.**, p. 348. Xu & Xia (2021) manually divided the writing process of L2 writers into these three subprocesses. Prewriting/planning was defined as: "the temporal span from the beginning of a writing event upon topic assignment till the commencement of continuous textual output, featured by the flat product line (the solid line) following the beginning of a writing event" (p. 592; see also Figure 1 and related process graph description). Formulation was defined as "the process of continuous textual output, featured by steeply climbing process and product lines" (p. 592). Finally, reviewing/revising was defined as "the process temporally following the formulation process, with the cursor position (the dotted line) being moved to the beginning part of the text for reviewing and revising till task completion" (p. 592). This final reviewing/revising phase was similar to the revision phase in their previous work, which was operationalized as

the last revision(s) away from the point of inscription, whose termination was not followed by further production at the end of the text (Xu, 2018).

Similarly, Hall et al. (2022) manually distinguished explicit pre-planning and post-draft revisions. However, as the participants were asked to write spontaneously or plan on paper (depending on the condition), no planning other than writing the title was identified. Post-draft revisions were defined as text production after the writer decided to close their essay. Here the authors included both textual factors (whether the writer used words like 'to conclude' or 'finally'), as well as process factors (whether the writer moved away from the leading edge to start making edits from 'top-to-bottom'). This notion of top-to-bottom revision was also mentioned in Xu & Xia (2021)'s definition above, focusing on the cursor being moved to the beginning. Baaijen et al. (2012), first isolated the initial prewriting/planning phase as part of the task instruction since they asked the participants to plan their text with pen and paper for five minutes. Next they distinguished the other phases manually. First they categorized continued planning on the computer separately from text production. Then, they manually annotated text produced during an initial draft from text produced during a revision draft or the final revision phase (Baaijen et al., 2012). The final revision phase was defined as the revisions made outside the final paragraph, when the writer was working on the final paragraph (similarly to Hall et al.'s, 2022, focus on textual factors). In total, this final revision phase was identified in 65% of the writing processes.

The advantage of these function-based approaches is that they might be considered less arbitrary than time-based or version-based approaches, as they are directly related to functions or dominant writing processes involved. However, as could be seen from these examples, the segmentation is usually done based on manual inspection of progress graphs, keystroke logs, and replays of the writing process. The rules for manual annotation vary across studies often focus on single-session composition processes, and are arguably not always easily implemented (interestingly most articles do not mention any coding difficulties or inter-rater reliability, but see Xu, 2018). Therefore, in the current paper we aim to develop an automated function-based segmentation that overcomes these limitations.

2.5 Going beyond sequential phases

It is well-known that cognitive processes in writing are non-linear rather than sequential. The planning, translating, and reviewing processes are inter-related and interact. Accordingly, rather than segmenting the writing process into a specific number of consecutive segments, some researchers have also looked at dividing the keystroke log into specific micro-processes or activities, where activities can have different lengths and are recurring over time. These activities have been related to the cognitive subprocesses of writing and sometimes include behavioral metrics rather than cognitive constructs, such as typing versus reading sources. Exploring the sequences and combinations of these micro-processes or activities can further shed light on the non-linearity of the writing process.

For example, Kruse (2024) distinguished text production from source-use, based on the automated detection of focus shifts away from the written text. Guo et al. (2019)

automatically segmented the writing process into three different activities including: text production, editing (out of order insertion or deletion), and long pauses (pauses which are longer than four times the median pause length thus far). Conijn et al. (2024) further extended on the notion of editing by – as argued by the authors – focusing on the *full* cognitive process of making a revision, so for example also including the replacement text after an insertion. They used a rule-based algorithm to automatically distinguish revision activities from non-revision activities based on the number of deletions and the cursor position relative to the leading edge. Lo Sardo et al. (2023) also focused on the automatic extraction of sub-cycles of planning and translation using edit distances between different versions of the text. Specifically, planning or *exploration* was operationalized as moments where the edit difference between the first and last version was larger compared to the edit distances between the current version and first and last version combined (larger difference means more exploration). Translation in turn was identified when the writer more or less fluidly translated their ideas, deterministically reducing the distance between the first and last version.

Other researchers have opted for a broader segmentation, including a wider range of activities. For example, Sala-Bubaré et al. (2021) used manual annotation to identify seven different types of activities: text production, interaction with sources, editing (surface-level revisions), revising (deep-level revisions), reading text written so far, deleting (without inserting new text), recursive reformulations (changes at the point of inscription). These segmentations were done on top of the content-based segmentation of the different sections written as described above. Similarly, De Smedt and colleagues (*under review*), divided the keystroke log into automatically identified segments, including several activities related to accessing/reading resources, text production, navigation, and revising (including immediate and distant insertions and deletions).

The advantage of these approaches focusing on microprocesses is that they are more detailed and follow more closely the fact that writing is a non-linear process. In addition, these types of segmentation are usually well-suited for process mining methods, where the different activities can be seen as different 'states' and the changes between activities as 'transitions' (see e.g., De Smedt et al., *under review*). This type of analysis can also be interpreted in function of time, providing a specific focus on the distribution of a particular micro-process. A downside of this approach is, however, that by focusing on specific micro-processes there will be a lot within and between-writer variation and noise, making it harder to interpret the findings in terms of overall writing development or to improve writing instruction. This is also why these types of analyses are often supplemented with additional analyses such as clustering methods, to aid the interpretation.

2.6 Current approach

In this study we aim to develop an automated function-based segmentation of single-session processes that, rather than using a micro-approach focusing on a large number of transitions, focuses on a macro-approach identifying distinctive transition of the dominant process (e.g.,

in this case from predominantly writing to predominantly revising), based on keystroke data. Such a dominant switch in keystroke dynamics can also be seen as 'points-of-interest' as referred to by Leijten et al. (2014). In particular, our procedure consists of two steps. First, we aim to detect change points in the keystroke data. Thereafter, we aim to automatically select the change point that is indicative of the transition or change point from the production of a first draft of a text, with a focus on planning and the production of new content, towards a *second phase* in which writers revise and finalize their first (intermediate) draft. The full procedure is evaluated using manual coding of a set of process graphs obtained via the keystroke logging tool Inputlog. In particular, we aim to answer the following research questions:

1) To what extent is it possible to automatically detect change points in the writing process?
    a. To what extent do automatically identified change points overlap with the manual annotation of the start of the second phase?
    b. Which variables – inductively identified from human coding – are the best predictor(s) to detect change points in writing processes?
2) To what extent is it possible to automatically select the change point that indicates the transition to the second phase?

3. Method

3.1 Datasets

For this study, we used two datasets consisting of keystroke data collected via Inputlog (Leijten & Van Waes, 2013). The first dataset came from the project *PLanTra* (Plain Language for Financial Content: Assessing the Impact of Training on Students' Revisions and Readers' Comprehension) and the second dataset came from the project *LIFT* (Improving Pre-university Students' Performance in Academic Synthesis Tasks with Level-up Instructions and Feedback Tool). We particularly chose these two datasets as they are representative of different writing tasks, which presumably increases the variability in how a second phase might be initiated, thus allowing us to test our approach in a broader range of contexts.

The PLanTra project involved the collection of keystroke data from 47 Dutch-speaking university students, writing in English (L2). In a pre-test session, al students were assigned an extract of a corporate report dealing with sustainability and were instructed to revise it to simplify the text for a lay audience. Subsequently, half of the students received training on how to apply plain language principles to sustainability content, while the other half received training exclusively on the topic of sustainability. During a post-test session, both groups were instructed to revise a second extract of a corporate sustainability report. Not all participants participated in both the pre- and post-test, resulting in a total of 88 sessions being logged. A more detailed description of the methodology in the PLanTra project can be found in Rossetti & Van Waes (2022b). The dataset for the PLanTra project is published in Rossetti & Van Waes (2022a).

The LIFT project' s goal was to provide feedback and instruction to students' synthesis writing in the Netherlands, based on national baseline data. Various types of data were gathered to create the baseline including keystroke data. The baseline consists of a large and representative sample of 658 students from 43 schools. Participants were upper-secondary students from grades 10, 11 and 12. The students wrote multiple texts in two genres (argumentative and informative) of source-based writing in Dutch (L1). For the current study, a subset of writing processes was selected from the baseline: 40 processes of argumentative tasks and 40 processes of informative tasks. The processes selected for the subset cover a wide range of performance levels. For details on the methodology of the LIFT project, please consult Vandermeulen, De Maeyer, et al. (2020). The dataset for the LIFT project is published in Vandermeulen, Van Steendam, et al. (2020).

3.2 Development of manual annotation criteria

As a first step in the coding, we built a set of criteria that could be used to identify the change point between the initial planning and production of new content, and a second phase in which writers revise and finalize their first (intermediate) draft. The initial step in criteria development involved an inductive analysis of the writing process, for which we used Inputlog's process graph (Leijten & Van Waes, 2013; see also Figure 1). Inputlog's process graph has been used to aid manual annotations (e.g., Xu & Xia, 2021 as described in the introduction). The process graph is a visualization aid that has been used to visualize the writing process based on keystroke data. The characteristics shown in the graph is based on an extended period of writing process research (E. Lindgren & Sullivan, 2019; Van Waes & Schellens, 2003), which allow for inspection and analysis of pausing behavior (Van Hell et al., 2008), writing fluency (Feltgen & Cislaru, 2025), (non-)linearity (Buschenhenke et al., 2023), revisions (E. Lindgren & Sullivan, 2006), and interactions with sources (Tarchi et al., 2023). The process graph in particular visualizes five different characteristics of the writing process: (1) product development (number of characters in the product so far), (2) process development (number of characters produced so far), (3) cursor position (at that moment in time), (4) pause distribution and pause length, and (5) interaction with sources.

Based on Inputlog's process graph, two researchers independently identified the beginning of the second phase on a subset of 24 cases from the PlanTra dataset. After discussion, the researchers agreed on three criteria for the annotation. The first criterion was the movement of the cursor towards – or close to – the start of the document, indicating that the writer had completed a first draft and was prepared for a whole-text reading/revision. The researchers also agreed that, when the cursor moved to the start of the text too early in the writing process (e.g. in the first quarter the process), or when this movement was fast and immediately followed by a repositioning of the cursor to the point of utterance, the cursor movement should be disregarded. A second criterion was the deletion of unnecessary content, as indicated by a sudden and substantial split between process line and product line. In the PlanTra dataset this was often the case when students opted for rewriting a new text from scratch and subsequently deleted the assigned text once their new draft was completed.

The third criterion was the flattening of the product line, indicating that no substantial content was being added anymore (i.e. the first draft was produced).

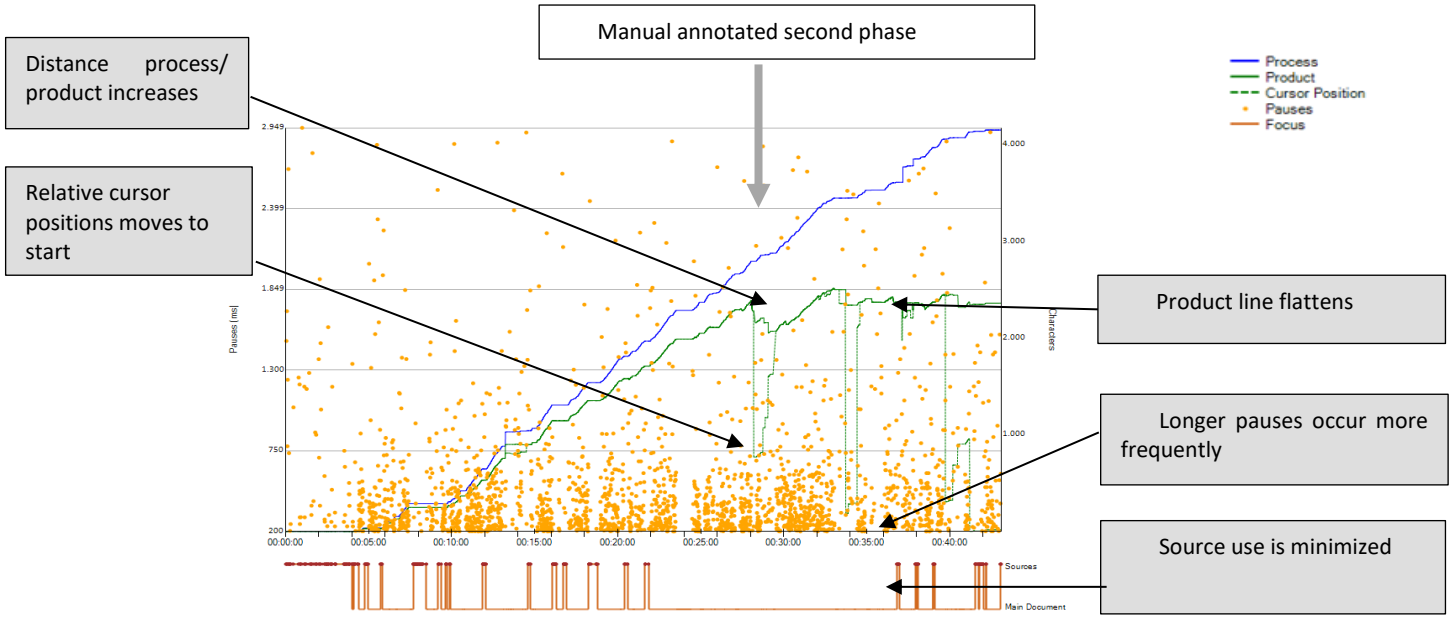

Figure 1. Inputlog process graph, including the five criteria for the second phase.

For a detailed description of the criteria, see Table 1.

Table 1. Description of indicators and annotation criteria.

Indicator	Criteria	Explanation
Document length	Product line flattens	When the product line flattens at a certain moment, this indicates that text production slows down or (temporarily) stops while rereading, and (low-level) insertions and deletions throughout the text follow up on each other, keeping the total length of the text more or less constant. So, this change in slope often marks the start of revision and rereading during the second phase.
Distance between product and process line	Distance between product and process line increases	Those writers that prefer rewriting the text by producing a new text (and do not revise in the source text itself) usually delete the original text at the end of the first phase. Some writers also delete the source text paragraph by paragraph. In this case, deletions are recursive and we take as change point — marking the start of the second phase — the end of the last recursive movement. Substantial deletions are always represented by a considerable drop in document length and, in turn, by greater distance between process and product line.
Relative cursor position	Point of utterance changes to start of the text	A moving cursor line towards the beginning of the text indicates that the writers' focus is changing from the end of the file towards the beginning of the text produced so far. Sometimes this happens in longer cycles that follow up on each other. This repositioning is often characterized by an initial longer reading pause indicating the start of a reflective revision phase (i.e. a second phase).
Length of source use	Interaction with sources is minimized	A drop in the interaction with sources often indicates a shift in the writers' focus from external documentation to the text itself, where revision becomes the central activity.
Pause length	Longer pauses occur more frequently	At the start and during the second phase writers tend to pause longer. We see a change in the pausing pattern in which the clusters of short pauses close to the x-axis are opened, and longer pause plots higher up in the graph are observed. Most of these longer pauses are likely related to reading and to evaluating the text produced so far.

Following the identification of three criteria, the two researchers applied these criteria again to the entire PlanTra dataset. The researchers had an initial agreement in 73% of cases, which – after discussion – resulted in full agreement. The researchers also observed that the change

points were usually preceded or followed by longer pauses (marking the beginning of a reflective reading/revision second phase) and by a change in the pattern of the interaction with sources (usually source use was minimized, indicating that external content was less frequently used once their first draft was completed). Based on these observations, the pause length and interaction with sources were added as supplementary criteria. During this stage of criteria development, the researchers also decided that the pause preceding the second phase would also be included, as this long pause could already indicate the start of a reading/revision phase, even though no external edits or actions are visible yet. The detailed description of the five criteria is shown in Table 1.

In the next stage, the two researchers recoded the same subset (n = 24) of the PLanTra process graphs using the new descriptions of the five criteria. Here, the researchers agreed in 83% of cases, and reached agreement on 100% of cases following discussion. In addition, this second round of coding confirmed that pause length and interaction with sources (i.e. the newly added criteria) were relevant for the identification of change points, especially when no clear change point was found using the first three indicators. Interestingly, for 75% of process graphs, the researchers identified more than one criterion that indicated the change point. Moreover, these criteria also allowed the researchers to identify writing processes (six out of the total) that did *not* involve a second phase.

3.3 Manual annotation

In the third round, three researchers applied the five criteria to the process graphs from the LIFT dataset. We believed that the indicators could be translated from the PlanTra to the LIFT dataset, as the datasets shared the characteristics of being both single-session source-based writing tasks (cf. Future work infra). For the informative texts, all three authors agreed in 68% of cases. Pairwise consensus (i.e. between two annotators) was reached on average in 78% of cases, but full agreement on all process graphs was reached after discussion. For the argumentative texts, agreement between the three researchers increased to 81% (87% for pairwise consensus), possibly as a result of coding practice. Full agreement was again reached following discussion. In addition, the three researchers also reached agreement on 24 process graphs that did not show a second phase. In total, there are 6 (PlanTra) + 24 (LIFT) = 30 sessions without an annotated change point, and 82 (PlanTra) + 56 (LIFT) = 138 sessions with an annotated change point.

3.4 Automated change point detection

The automated identification of the change point included three steps: (1) pre-processing, (2) automated detection of change points in the keystroke data, (2) automated selection of the change point between the first and second writing phase. All steps were performed using R, and the code can be found at https://github.com/RConijn/KeystrokeChangePoints.

For the pre-processing, we first translated the five criteria as close as possible into indicators that could be retrieved from the keystroke data. Specifically, for criterion 1 (*product*

line flattens) the product line was measured using the product length, relative to the final product length; for criterion 2 (*distance between product and process line increases*) the distance between the product and process line was measured using the difference between the number of characters produced and the product length; for criterion 3 (*point of utterance changes to start of the text*) the relative position was calculated by the cursor location, divided by the product length; for criterion 4 (*interaction with sources is minimized*) the length of source use was calculated by the log cumulative sum of the time the writer spent in sources outside the main document; and finally for criterion 5 (*longer pauses occur more frequently*) pause length was measured using the cumulative mean of pauses, where pauses were log-transformed and trimmed to the 99% percentile. There was a low to moderate correlation between the criteria, ranging from $|r|$ = 0.01 to 0.43. The strongest correlations were observed between criterion 1 and 4 (r = -0.43, $p < 0.001$) and criterion 1 and 2 (r = 0.41, $p <$ 0.001).

For the change point detection algorithm, the keystroke data needed to be transformed into a time series that provided a value for each of the five indicators per x seconds. Given the fact that keystroke data can be relatively noisy, with quick jumps back and forth in the text, we decided to summarize the log file in multiple ways: per 1, 5, and 10 seconds. This was done to determine which timeframe would most effectively get rid of this noise, without overgeneralizing too much. In addition, given the fact that the start of the writing process is often messy, and usually not of interest for the detection of writing phases, we summarized the full writing process as well as the writing process with the first 10% of the time excluded (no trimming, versus 10% trimming). This resulted in a total of 3 (time) * 2 (trimming) = 6 different time series.

After the pre-processing, we first automatically identified change points in the keystroke data, based on each of the five indicators. For example, for the first indicator, one might see an increase in product length (steep product line), followed by a relative stable product length (product line flattens). A large change in the slope of the product length would then be indicated as a change point. The change points were identified using a Bayesian ensemble change-detection algorithm for time series, called the Bayesian Estimator of Abrupt change, Seasonality, and Trend (BEAST) from the Rpackage 'Rbeast' (Hu et al., 2021; Zhao et al., 2013, 2019). BEAST is a statistical algorithm that breaks a time series *Y(t)* into trends, seasonal variability, abrupt changes, and noise. It is an ensemble algorithm, which means that it combines multiple weaker models into one stronger model, using Bayesian model averaging. In addition, given that it is a Bayesian algorithm, rather than providing a point estimate for the change points, BEAST estimates the probability of the change points, and provides a credible interval for the location of the change points. For more information on the BEAST algorithm, see Zhao et al. (2019).

For the current problem, seasonal variability was not modeled because keystroke data do not exhibit periodic patterns (such as regular cycles seen in weather data, like daily or annual fluctuations). This resulted in the following model of the keystroke time series:

$$Y(t) = T(\theta_t) + \varepsilon$$

where T is the term for the trend component, which is modeled as a piecewise linear function with an unknown number of change points. Here, θ_t specifies the number and location of the change points in the trend component, and ε is the Gaussian random error term *N(0, δ^2)* with an unknown variance δ^2. The posterior probability distribution of θ_t and δ^2 are simulated using Markov Chain Monte Carlo (MCMC) sampling. For more information on the Bayesian MCMC scheme used, see Zhao et al. (2019). In our analysis, we opted for the default MCMC sampling settings, using three parallel chains with 8000 samples each. The first 1500 samples per chain were discarded as burn-in, and only every 5th sample was retained (thinning factor = 5). Weakly informative priors were used for the trend estimations T, with trend orders limited to 0 (flat) or 1 (linear), and the number of changepoints bounded between 0 and the maximum number of changepoints to be estimated. A non-informative uniform prior was used for the precision (i.e. the inverse variance: $1/\delta^2$). Model convergence was checked manually using trace plots for a subset of the models. In addition, robustness of the changepoints was further enforced by using only changepoints with a high probability and a narrow credible interval when selecting the final changepoint of interest (for details see below). This approach has been previously proven as a useful way to eliminate false change points (J. Li et al., 2022).
The BEAST algorithm was run for each participant, on each of the six time series, for each of the five indicators (univariate models), with a varying amount of maximum change points to be estimated (1, 3, 5, 10, or 20). The minimum distance between two consecutive change points was set to 10 seconds. In addition to the univariate models, multivariate models were estimated, with the first three indicators combined. However, it should be noted that these multivariate models are for experimental use only and still under development (see Rbeast documentation). The BEAST algorithm was evaluated in two ways. First, the goodness-of-fit of the algorithm was determined using the adjusted R^2 (only available for the univariate models). Second, we identified whether the manual annotated change point was among one of the BEAST-detected change points or fell into the 95% credible interval of the BEAST-detected change points.

Thereafter, the best performing BEAST algorithm was selected for each of the indicators. As these algorithms provided multiple change points for the indicators (often close to the set value of the maximum number of change points), we still needed to identify which of the change points would most likely be the change point of interest. To select the final change point, a rule-based algorithm was used, including several overarching rules. The change point should be after 1/3rd of the process; the majority of the document needs to be written (70%); the change point should have a high probability (>70%); and the change point should have a relatively narrow credible interval (< 60 seconds). In addition to the general rules, several indicator-specific rules were used, based on the increase or decrease in the intercept or slope of the indicator, following the manual annotation rules (top 3 flattest product line, that is slope of the segment close to 0; top 3 largest abrupt change in distance product versus process line; top 3 largest abrupt change in cursor location). Based on the final set of change point candidates left, a rule-based algorithm was created to pick the final change point (as shown in the results section).

4. Results

4.1 Automated detection of change points

The BEAST algorithm was used to determine the change points for each of the five indicators for all sessions. The full results of all estimated change points overlaying Inputlog's process graph (including the different summarization methods and trimming) can be obtained from the interactive Shiny application: https://rianneconijn.shinyapps.io/PhaseAnalysis/. It was found that trimming versus no trimming had limited effects on the final results. As expected, higher-level aggregation (per 5 or 10 seconds), resulted in slightly higher accuracy, compared to aggregation per 1 second, with limited difference between aggregation per 5 or 10 seconds. Therefore, below we only report the results on the models summarized per 5 seconds, without trimming.

A sample of the estimated change points for the first indicator, document length, can be found in Figure 2. In this sample, it can be seen that out of the five maximum change points, all five change points were estimated for all four participants. Three of the four change points are relatively accurate: three of the change points are relatively close (participant 1, 3, and 4), and two of those fall within the 95% credible interval (participant 1 and 4). For participant two, a maximum of five change points seem too little (note that only four change points are identified here in the algorithm), or the indicator (document length) might have been suboptimal to identify this change point.

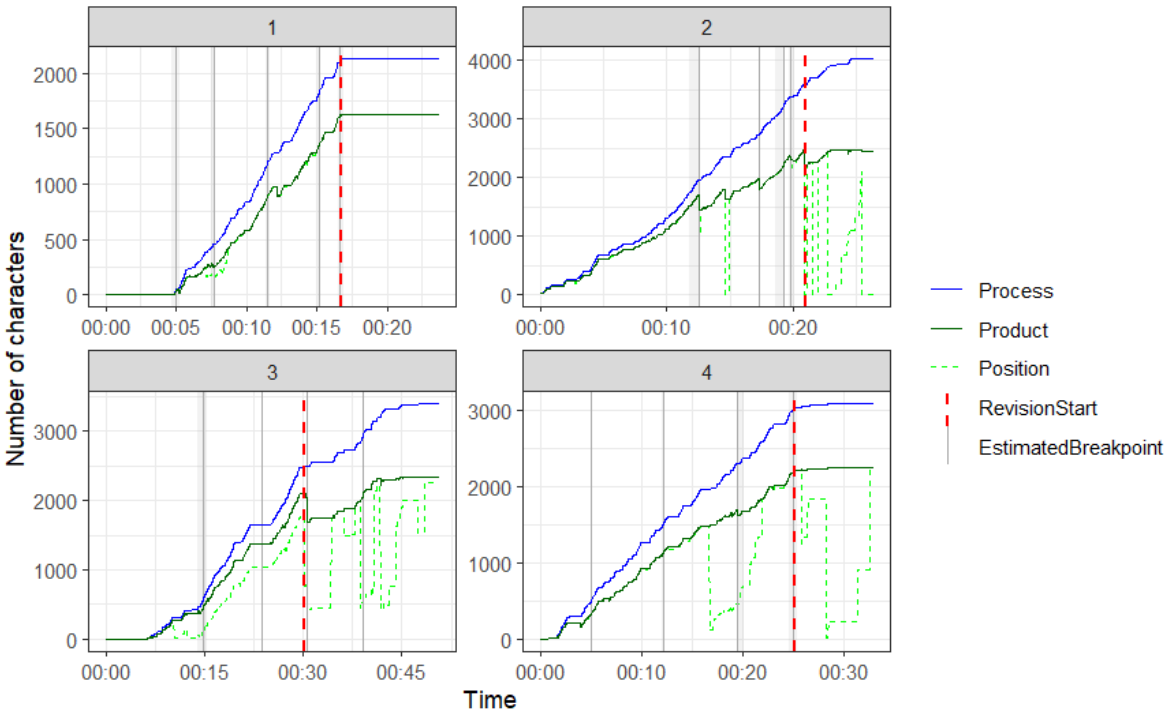

Figure 2. Estimated change points for document length (indicator 1) using the log file summarized per 5 seconds, with no trimming, and 5 maximum breakpoints. Results are shown for a subset of 4 participants (all from the LIFT dataset).

Note. Results for the other indicators, log file summarizations, and participants, can be obtained from https://rianneconijn.shinyapps.io/PhaseAnalysis/.

The algorithms were evaluated based on goodness-of-fit as well as the overlap with the manual annotated change point (that is, whether the indicated change point overlapped with the manual annotation). An overview of all findings can be found in

Table *2*. The Shiny dashboard allowed us to further explore and evaluate the algorithm's identification of change points at the case level. For the goodness-of-fit we found that when more change points were estimated, the models fit the data better. The distance between the product/process indicator resulted in the best models, with an average adjusted R^2 of 0.97-0.99 across all participants, regardless of the number of change points used. The length of source use shows similar high model fit, which is probably due to the fact that there was limited source use, resulting in a relatively flat curve, which was easy to predict by the model. The opposite is happening for the relative position, which showed the lowest model fit. For models with a single change point, the model does not seem to capture all fluctuations in the relative position, resulting in an average adjusted R^2 of 0.64. This adjusted R^2 increased to 0.96 for ten change points.

For the number of change points overlapping with the manual annotation, we see that one change point is not enough to correspond with the manual annotation (accuracies ranging from 0-11%, with one exception for relative position of 23%). We – not surprisingly – found that when we increased the maximum number of change points, there was also a higher chance that the manual annotated change point was amongst them, resulting in the highest accuracies for the models with 20 as the maximum number of change points. The model for the relative position indicator proved to be the most accurate model: in 70% of the cases, the manual annotated change point was exactly the same as one of the 20 suggested change points. For 71% of the cases the credible intervals of the change points included the manual annotated change point. Also the model with a maximum of 5 change points already scores above 50%. After the model for relative position, the document length and distance product/process indicators proved to be the best models. The length of source use and mean pause length proved to be insufficient indicators to detect the manually annotated change point, with correct change points only in 0-1% of the cases.

Table 2. Accuracies of the univariate change point detection using *BEAST*

Indicator	Maximum number of change points	Correct change point (%)	Correct change point (%) within 95% CI	Median difference (sec)	95% CI difference (sec)	Mean marginal likelihood	Mean adjusted R^2
Document length	1	10.9	14.5	425	[0;2182]	-570	0.929
	3	12.3	18.1	195	[0;1146]	-396	0.961
	5	23.2	33.3	38	[0;676]	-163	0.985
	10	25.4	42.8	15	[0;395]	77	0.993
	20	26.1	44.9	10	[0;183]	205	0.995
Distance product/ process	1	6.5	8.0	532	[0;2756]	-446	0.972
	3	13.8	17.4	185	[0;1319]	-182	0.990
	5	21.0	27.5	35	[0;720]	105	0.995
	10	23.9	34.1	20	[0;493]	371	0.996
	20	24.6	37.7	12	[0;277]	463	0.996
Relative position	1	22.5	22.5	370	[0;2228]	-1115	0.637
	3	44.2	44.2	25	[0;1244]	-951	0.797
	5	55.1	55.1	0	[0;573]	-736	0.904
	10	64.5	68.1	0	[0;380]	-433	0.964
	20	70.3	71.0	0	[0;183]	-229	0.976
Length of source use	1	0.0	0.0	1455	[796;2441]	-264	0.941
	3	0.0	7.9	982	[42;1959]	866	0.988
	5	0.0	9.5	600	[45;1880]	2396	0.994
	10	0.0	6.5	418	[42;3140]	3247	0.994
	20	0.0	5.8	452	[12;2508]	3255	0.994
Mean pause length	1	0.0	0.0	1445	[396;3473]	-747	0.849
	3	0.0	0.0	1165	[175;2752]	-537	0.921
	5	0.0	5.1	705	[25;2215]	-332	0.954
	10	0.7	13.0	400	[7;1781]	-195	0.980
	20	1.4	17.4	285	[5;1781]	-165	0.966

Note. Accuracies are provided for the change point closest to the manual annotation. CI = Credible interval.

This might indicate that the document length and distance product/process do not add much to the relative position indicator. However, it should be noted that the 95% credible interval of the difference between the predicted and manually annotated change point is much smaller. This indicates that although the multivariate algorithm does not necessarily pick the exact correct change point, it seems to be consistently closer to the actual change point, with the 95% credible interval being as narrow as between 0 (perfect overlap) and 36 seconds for the maximum of 20 change points.

Table 3. Accuracies of the multivariate change point detection using *BEAST*

Indicator	Maximum number of change points	Correct change point (%)	Correct change point (%) within 95% CI	Median difference (sec)	95% CI difference (sec)	Mean marginal likelihood
Document length,	1	11.7	13.9	495	[0;2373]	-2765
Distance product/	3	31.4	36.5	35	[0;951]	-2358
process, &	5	45.7	50.7	5	[0;429]	-1801
Relative position	10	57.2	68.1	0	[0;81]	-1014
	20	66.4	77.4	0	[0;36]	-292

Note. Accuracies are provided for the change point closest to the manual annotation. CI = Credible interval. Adjusted R^2 not available for this experimental multivariate analysis.

To further identify if similar change points were selected by the different univariate and multivariate models, we examined the overlap between the detected changepoints, as shown in

Table 4. Note that for clarity we only report the models with 10 maximum changepoints here. The results indicate that the models identified some overlapping change points, with overlap ranging from 17% to 30%. Among the univariate models, the relative position indicator appeared to detect more distinct change points compared to the document length and distance product/process indicators. As expected, the multivariate model showed the greatest overlap with each of the univariate models.

To conclude, we see that the length of source use model fits the data well but shows little information for predicting the change point. Relative position, followed by document length and distance product/process show to be more promising indicators, but each indicator seemed to point towards different change points (with some overlap). The multivariate model did not seem to outperform the relative position univariate model. Finally, we see that increasing the maximum number of change points largely improves the models. However, with more change points, it also becomes harder to pick the change point of interest. Therefore, the next section looks into selecting the actual change point.

Table 4. Overlap between the detected change points per indicator

Indicator	Mean (SD) change points detected	Percentage of overlapping* change points			
		Document length	Distance product/ process	Relative position	Document length, Distance product/ process, & Relative position
Document length	9.5 (1.1)	-	0.25	0.17	0.28
Distance product/ process	8.7 (2.0)		-	0.18	0.30
Relative position	9.2 (1.7)			-	0.29
Document length, Distance product/ process, & Relative position	9.9 (0.7)				-

Note. *Overlap is considered when the change point falls within the credible interval or within 10 seconds of the other change point. Values are provided for the model with 10 maximum change points.

4.2 Automated selection of change point

In the next step, based on all the change point candidates identified, we aimed to select the change point that indicates the transition to the second phase (i.e., revision). Based on the outcomes of the first stage, we selected the first three indicators (document length, distance product/process, and relative position). As the multivariate models are experimental, we only used the change points detected by the three univariate models. Further, we selected a maximum of 10 change points per indicator, to avoid having too many change point candidates, while still retaining reasonable accuracy.

First, four overarching rules were applied as detailed in the methods section to filter the change point candidates. Thereafter, change points that were considered overlapping (that is, change points that were within 10 seconds from each other or within the credible interval) were combined into one change point. After this initial filtering step, on average 4.8 (*SD* = 2.0, Min = 1, Max = 9) change point candidates were left per participant. Finally, a rule-based approach was used to select the final change point, including whether more than one indicator showed this change

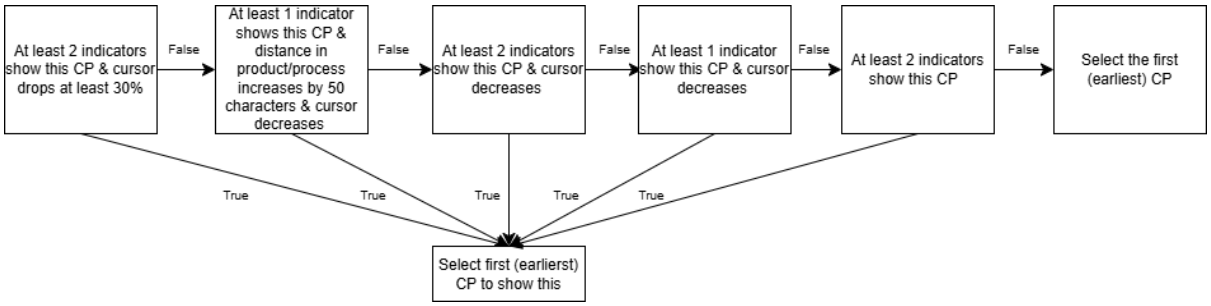

Figure 3. Rules to select the final change point among change point candidates. (CP = change point)

point, and whether a drop in cursor location was shown (for the full rules see Figure 3). An overview of the selected change point for each session (compared to the manual annotated change point) can be obtained from:
https://rianneconijn.shinyapps.io/PhaseAnalysis/.

The overall accuracy is low: Only in 31% of the sessions the selected change point is the same as the manual annotated change point. In addition, we see that some of the selected change points are only slightly off (36% correct within 10 seconds of the annotated change point, and 49% correct within 60 seconds). The accuracy did not seem to depend on session characteristics: although longer sessions showed somewhat lower accuracy in the overlap between the manual and selected changepoint, this was not found significant. The total number of characters typed and the total number of characters in the final product also did not influence the accuracy. For some writing sessions, the selected change point is far away from the annotated change point (e.g., in 27% of the cases the difference is larger than 5 minutes). A closer inspection into the erroneously predicted sessions showed a variety of reasons (see Figure 4 for a subset of sessions with low accuracy). For example, some of the authors (e.g., session 82) revised 'backwards', where the cursor started at the end of the text, and slowly moved upwards to the beginning. This slight change of cursor location is not detected by the algorithm. Another source of difficulty included sessions where the writer showed more than one phase in which they revised the first draft. The manual annotation guidelines detailed that in this case, the start of the first revision cycle needs to be selected. However, this was not always done by the algorithm: when the algorithm detected multiple cycles, sometimes either the first (e.g., session 77) or the second (e.g., session 135) was selected, which did not match the manual annotation.

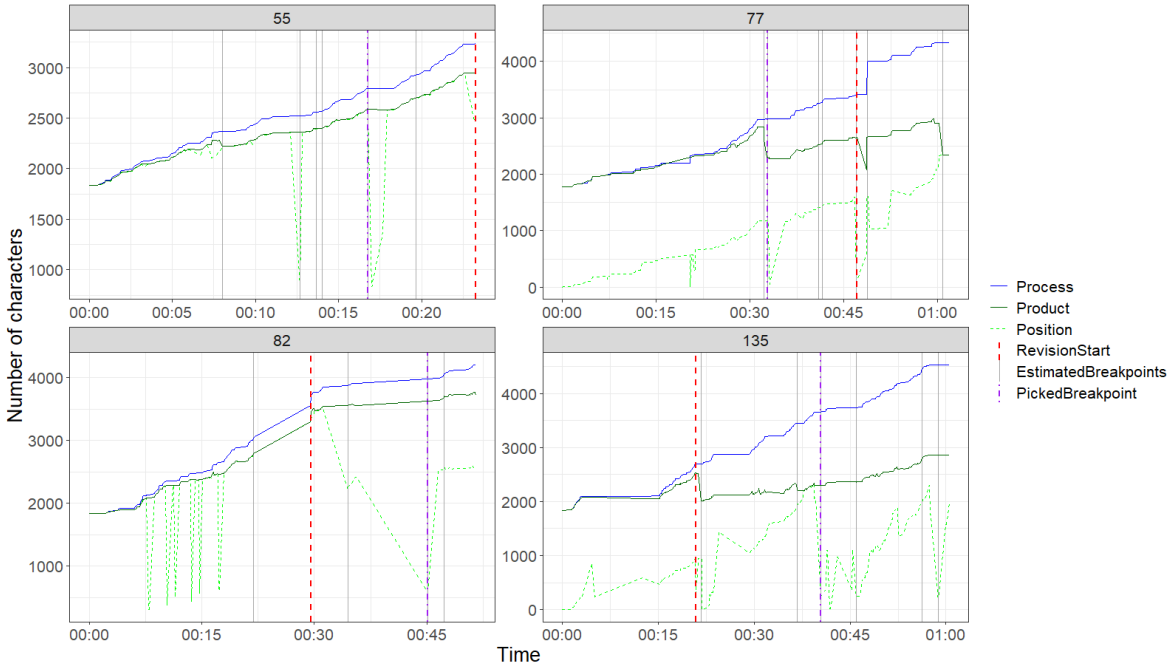

Figure 4. Selected change points versus manual annotated change point. Results are shown for a subset of 4 participants (all from the PlanTra dataset) showing low accuracy.

Note. Only the estimated breakpoints are shown that are left after the initial filtering step. Results for the other participants can be obtained from: https://rianneconijn.shinyapps.io/PhaseAnalysis/.

Finally, some sessions showed a very late manual annotated change point (e.g., session 55), where almost nothing happened after the change point, making it hard for the algorithm to detect a change in one of the indicators, hence resulting in the selection of a change point earlier on in the process.

4.3 No second phase present

In 30 of the sessions, no second (revision) phase was present, according to the manual annotation. As further proof of concept, we are interested to see how the change point detection algorithm performs in these cases. First, it was found that for almost all (29/30) of the sessions, at least one change point was identified after the initial filtering step. On average slightly less change point candidates were left compared to the sessions *with* a manually annotated second phase (M = 3.0, SD = 1.3, Min = 1, Max = 6 change point candidates left). This indicates that even though there is no final revision phase present, still some important change point candidates are identified. For all but one of the sessions, at least one change

point was related to indicator 1 (document length), while indicators 2 and 3 were less present (14/30 and 12/30 sessions, respectively). After running a decision tree with 10-fold cross-validation, these characteristics also showed up to be the main indicators: If at least one of the change points was related to indicator 3 (cursor position), there was a high chance of a second phase. In addition, if there was no change point related to indicator 3, but there were four or more change points detected, there was still a high change of a second phase.

5. Discussion

Previous work has used different ways of segmenting writing processes, to be able to analyze the temporal organization of the writing process. In this contribution, we distinguished four types of segmentation approaches: time-based, content-based, version-based, and function-based. We argue that the first three can be arbitrary, where the approach often focuses on the researcher's intuition rather than the writer's intent. The function-based approaches might be more promising, as they are more related to the (underlying cognitive) functions or the dominant writing sub processes involved. However, function-based approaches require time-intensive manual annotation. Therefore, in this paper, we aimed to develop an automated segmentation of writing processes that focuses on a distinctive transition in the dominant writing processes. The BEAST algorithm (Zhao et al., 2019) was used to automatically detect change points within keystroke data obtained from two different datasets (Rossetti & Van Waes, 2022b; Vandermeulen, De Maeyer, et al., 2020). A variety of change points were detected based on five indicators from the keystroke data: document length, distance between product and process, relative position, length of source use, and mean pause length. Thereafter, a rule-based approach was applied to select one change point which would be indicative of a shift from the first draft of the text to a second phase in which the writer revises and finalizes their first (intermediate) draft. The results of both steps are discussed below.

5.1 Automated detection of change points

First, it was found that all models showed high goodness-of-fit, indicating that the observed values are close to the expected values of the models. This shows that the time-series decomposition into trends, abrupt changes, and noise fit the keystroke data well. Intuitively, as the models became more complex (that is, a higher number of maximum change points was allowed), the model fit increased. The overlap with the manually annotated change point differed substantially per indicator: length of source use and mean pause length showed to be useless for detecting the manual annotated change point, while the relative cursor position showed to be the most promising indicator.

The fact that the length of source use was less useful might be explained by the fact that the interaction with sources was very limited in the current datasets, which might have resulted in the algorithm not being able to detect a substantial change in the length of source use. In the manual annotation, source use was often seen as an additional indicator, indicating that it might not be the best indicator for a univariate model. An alternative reason for the

low accuracy of the source use indicator could be that the algorithm looks into a specific point in time where the length of source use changes, while the manual annotators merely looked for an overarching pattern of source use. Hence, the length of source use might be better used to identify a time period where the shift to the second phase should take place, rather than a specific point in time. Future work should look into datasets with more extensive source use as well as focusing on a time range to test these hypotheses.

For the mean pause length the low accuracy might also be due to the fact that the pauses were often used as an additional indicator, rather than a stand-alone one. In addition, a longer pause might be indicative to various other activities rather than reading and evaluating the text produced so far, hence not necessarily be related to the start of a second phase. Pauses could also indicate other activities, such as reading sources, time off-task, planning for sentence production, irrelated to the start of a second phase (Medimorec & Risko, 2017). Future work should look into the added value of using eye-tracking to get a better indicator for reading and evaluation of the text.

Overall, document length, distance between product and process, and relative position, showed to have reasonable to high overlap with the manual annotated change point. Moreover, these indicators showed distinct change points indicating it might be useful to consider multiple measures simultaneously. However, interestingly the (experimental) multivariate model did not prove to outperform the univariate models. Models with a larger number of maximum change points were shown to have a higher accuracy in terms of the overlap with the manual annotated change point. However, this also means that it becomes more difficult to select the correct change point amongst the candidates, hinting at a trade-off between accuracy and interpretability. This further stressed the importance of evaluating both based on model fit as well as interpretability, in this case measured as overlap with human annotation.

5.2 Automated selection of change points

Given the amount of identified change point candidates, the next step of the analysis focused on selecting the correct change point overlapping the manual annotated change point. A combination of overarching filters and a rule-based model was used. Combined this resulted in relatively low accuracy: only 31% of the change points could be correctly identified. For 49% of the sessions, the selected change point was within 60 seconds of the manual annotated change point. So, for more than half of the sessions, the algorithm was unable to properly select the right candidate. This stresses the complexity of finding *one specific* change point. Manual inspection showed that there are quite some irregularities in the writing process that might be selected as change point candidate by the model. Some of the more common errors (e.g., the algorithm not picking up on writers revising backwards through their text), could be added to the rule-based algorithm. However, there is a trade-off between having an extensive set of rules (hence higher accuracy), versus interpretability and generalizability of the model (also known as 'the principle of parsimony'). Applying more rules will make the model harder to interpret for humans, and might result in overfitting of the data, which would reduce

generalizability. Accordingly, we did not further specify the rules. Future work could look into machine learned algorithms to identify (a) whether a change point is present, and (b) which of the candidates is the correct change point, with a larger sample size allowing for a test-train split and cross-validation to counter overfitting.

Finally, it should be noted that the manual annotators also showed difficulties in selecting the starting point of the second phase, which is a common issue in annotating keystroke data (Conijn et al., 2021; E. Lindgren et al., 2019). One might question whether the manual annotation correctly represents the ground truth. An alternative method would be to let the writers themselves indicate the change in phase *post-hoc*, which might be used to further train the model. Similarly, the writer might indicate a change in phase concurrently, closely resembling writer-based versioning. A tailored user interface, for example using specific tabs to separate phases of the writing process could help here (e.g., see the planner tool in Li et al., 2024). However, for writer-based versioning, the writer needs to be aware of these changes in processes, and for concurrent detection even be able to immediately identify such a change. In addition, identifying changes concurrently will influence the (mouse and) keystroke data, resulting in changes that might be more easily noticeable by an algorithm and hence might generalize less well across sessions without writer-indicated change points.

5.3 Limitations & Future work

This study is obviously limited by the specific type of algorithm and pre-processing chosen to automatically select the change point. Other algorithms could have resulted in higher accuracies. The advantage of using the BEAST algorithm is the fact that it is an ensemble algorithm, which combines multiple weaker models into one (Zhao et al., 2019). Experimental evidence has shown that ensemble algorithms outperform single-best-model approaches (Hastie et al., 2009). In addition, the model also outputs the probability of each detected change point, which was consequently used in filtering the change points. Finally, the algorithm explicitly models the uncertainty, by providing 95% credible intervals around the detected change point, which arguably works well for complex and noisy data such as keystroke data.

Of course, some improvements to the algorithm are possible. Additional features could have been passed to the BEAST algorithm or alternative rules could have been applied to select the final change point. For example, eye-tracking could have been included to more accurately identify sustained reading of the text written-so-far (as detailed above). Natural Language Processing could have been applied to more accurately detect if the majority of the text has been written e.g., by identifying whether a final concluding paragraph has been written or when the semantics do not change considerably anymore (e.g., see Tian et al., 2024). Finally, rather than allowing the change point to be possible at every 1 (or 5 or 10) seconds, it might have been worth it to constrain the location of the change point to previously studied boundaries in writing processes, such as after revision events (Conijn et al., 2024), after a break in linearity (Buschenhenke et al., 2023), or after a P-burst (Baaijen et al., 2012).

Further, some of the current indicators are based on characteristics of the final product, as they were relative to the final product length, or the total number of pauses. In this way, the analysis could not be used to provide real-time segmentation of the writing process. Hence, the algorithm cannot be used for real-time writing process interventions. Other metrics, such as text length relative to the amount of text produced so far, or the plain (log) cumulative number of insertions or deletions would have allowed for this real-time processing. However, it should be noted that it already proved difficult to identify change points at the end of the writing process, making it arguably more difficult to identify change points concurrently. In fact, the BEAST algorithm has lower robustness in identifying change points close to the end of a sequence, indicating that real-time segmentation of writing process would be difficult. Robust real-time segmentation might be necessarily done with a delay (e.g., a switch might only be noticed after x minutes) or might require alternative algorithms. Future work should examine to what extent the current approach – albeit with new indicators – would allow for real-time segmentation.

In addition, one could argue that the current macro approach focuses too much on the segmentation being discrete and sequential, which does not correspond with the complex non-linear nature of cognitive writing processes. Although our approach does give insight into recursiveness in revision, e.g., multiple revision phases, an activity-based segmentation (as in Sala-Bubaré et al., 2021 for example), could have resulted in a finer-grained insight into the recursiveness of the writing process. However, here, we were not interested in a large number of transitions, but rather focused on distinctive transitions in the writing process. We believe that these could be used as points-of-interest (Leijten et al., 2014), which could be further explored in future work.

Finally, although already two distinct genres of writing were selected (source-based writing and text simplification), it is unknown how well the algorithm generalizes to other datasets of the same genre or even other genres and contexts with less revising behavior, as for example found in novice writers. It is also uncertain whether algorithm performance varies across datasets, genres, or contexts. Future work should include additional datasets to better assess the algorithm's generalizability. For an overview on writing development of novice writers, see e.g., Beard et al. (2009) or Miller et al. (2018) or for examples of more complex writing and professional non-linear revising behavior, see e.g., multi-session writing, Buschenhenke et al. (2023) or Leijten et al. (2014).

5.4 Implications

The paper has several implications for writing research and practice, focusing on the change points themselves (as points-of-interest) or on the segments *between* the points of interest. First of all, the points of interest – in particular as shown on the phase analysis dashboard – can provide additional feedback to learners on their writing process, which may be used as an additional feature in a process report to reflect upon (Vandermeulen et al., 2022; Vandermeulen, Leijten, et al., 2020). In addition, this type of visualization can be used to improve the efficiency and accuracy of manual annotations of points-of-interest in keystroke

data. It should be noted here that the change points identified were directly related to indicators used in the manual annotation of revision phase, so they might not be applicable to other types of points-of-interest such as the end of an initial planning phase, or a distinctive shift in (non-)linearity. Yet, the indicators are relatively general (e.g., document length, cursor position) and focus on the main points presented in Inputlog's process graph, indicating their utility for writing practice and research. Future work should identify to what extent it is possible (and useful) to label the change points, and whether different variables could label these change points more accurately.

Second, one could use the change points to segment the writing process in different phases, allowing for analyses across all the different phases, rather than aggregating keystroke data over the full writing process (e.g., Xu & Xia, 2021). In addition, one could remove a specific phase, if it is out of interest for the specific analysis (e.g., removing the final revision phase as in Baaijen et al., 2012). Finally, one could test whether the type of processing is different in different segments, focusing for example on different pause patterns across the segments (cf. Roeser et al., 2025).

6. Conclusion

In this study we tried to explore the possibilities to automatically determine change points in writing processes, and use this as a basis to identify the change point that indicates the transition between the first (dominant focus on writing) and the second phase (dominant focus on rereading and revision) of a writing process. Our results showed that the BEAST algorithm was useful in detecting change points in keystroke data. In particular, relative position, followed by document length and distance product/process showed to be promising indicators to determine change points. Unfortunately, our results showed that it is still difficult to identify the change point connected to a second phase. Yet, we contend that our approach to identify change points is still useful to explore a specific subset of points-of-interest within the writing process or to divide the writing processes in specific segments with their own functionality.

Acknowledgements

The data collection for this study has received funding from the European Union's Horizon 2020 research and innovation programme (Marie Sklodowska-Curie grant agreement No 888918), and from a NWO National Grant (Dutch Research Council, Grant 405–14-301).

References

Baaijen, V. M., Galbraith, D., & de Glopper, K. (2012). Keystroke Analysis: Reflections on Procedures and Measures. *Written Communication*, *29*(3), 246–277. https://doi.org/10.1177/0741088312451108

Beard, R., Riley, J., & Myhill, D. (2009). *The SAGE Handbook of Writing Development*. 1–616. https://doi.org/10.4135/9780857021069

Bowen, N., & Van Waes, L. (2020). Exploring Revisions in Academic Text: Closing the Gap Between Process and Product Approaches in Digital Writing. *Written Communication*, *37*(3), 322–364. https://doi.org/10.1177/0741088320916508

Buschenhenke, F., Conijn, R., & Van Waes, L. (2023). Measuring non-linearity of multi-session writing processes. *Reading and Writing*, *37*(2), 511–537. https://doi.org/10.1007/S11145-023-10449-9/

Cislaru, G., & Olive, T. (2018). *Le processus de textualisation: analyse des unités linguistiques de performance écrite [The process of textualization: analysis of linguistic units in written performance]*. De Boeck Supérieur. https://doi.org/10.3917/dbu.cisla.2018.01

Conijn, R., Dux Speltz, E., & Chukharev-Hudilainen, E. (2024). Automated extraction of revision events from keystroke data. *Reading and Writing*, *37*, 483–508. https://doi.org/10.1007/S11145-021-10222-W/

Conijn, R., Dux Speltz, E., Zaanen, M. van, Van Waes, L., & Chukharev-Hudilainen, E. (2021). A Product- and Process-Oriented Tagset for Revisions in Writing. *Written Communication*, *39*(1), 97–128. https://doi.org/10.1177/07410883211052104

Crossley, S. A., Tian, Y., & Wan, Q. (2022). Argumentation features and essay quality: Exploring relationships and incidence counts. *Journal of Writing Research*, *14*(1), 1–34. https://doi.org/10.17239/JOWR-2022.14.01.01

Daxenberger, J., & Gurevych, I. (2013). Automatically classifying edit categories in Wikipedia revisions. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 578–589. https://doi.org/10.18653/v1/D13-1055

De Lario, J. R., Manchón, R. M., & Murphy, L. (2006). Generating Text in Native and Foreign Language Writing: A Temporal Analysis of ProblemSolving Formulation Processes. *The Modern Language Journal*, *90*(1), 100–114. https://doi.org/10.1111/J.1540-4781.2006.00387.X

Feltgen, Q., & Cislaru, G. (2025). The fluency vs. disfluency dichotomy in writing processes as reflected in the structure of the inter-key intervals empirical distribution. *Discourse Processes*, *1*(62), 16–2439. https://doi.org/10.1080/0163853X.2024.2417330

Flower, L., & Hayes, J. R. (1981). A Cognitive Process Theory of Writing. *College Composition and Communication*, *32*(4), 27–87. https://doi.org/10.58680/ccc198115885

Guo, H., Zhang, M., Deane, P., & Bennett, R. E. (2019). Writing Process Differences in Subgroups Reflected in Keystroke Logs. *Journal of Educational and Behavioral Statistics*, *44*(5), 571–596. https://doi.org/10.3102/1076998619856590

Hall, S., Baaijen, V. M., & Galbraith, D. (2022). Constructing theoretically informed measures of pause duration in experimentally manipulated writing. *Reading and Writing*, *37*(2), 329–357. https://doi.org/10.1007/S11145-022-10284-4/

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer New York, NY. https://doi.org/https://doi.org/10.1007/978-0-387-21606-5

Hu, T., Myers Toman, E., Chen, G., Shao, G., Zhou, Y., Li, Y., Zhao, K., & Feng, Y. (2021). Mapping fine-scale human disturbances in a working landscape with Landsat time series on Google Earth Engine. *ISPRS Journal of Photogrammetry and Remote Sensing*, *176*, 250–261. https://doi.org/10.1016/J.ISPRSJPRS.2021.04.008

Huang, Y., & Zhang, L. J. (2022). Facilitating L2 writers' metacognitive strategy use in argumentative writing using a process-genre approach. *Frontiers in Psychology*, *13*, 1036831. https://doi.org/10.3389/FPSYG.2022.1036831/BIBTEX

Kruse, M. (2024). Problem-solving activity during the foreign language writing process: A proposal for categorisation and visualisation of source use and a new take on fluency in. *Journal of Writing Research*, *1*(16), 129–161. https://doi.org/10.17239/jowr-2024.16.01.05

Leijten, M., & Van Waes, L. (2013). Keystroke Logging in Writing Research: Using Inputlog to Analyze and Visualize Writing Processes. *Written Communication*, *30*(3), 358–392. https://doi.org/10.1177/0741088313491692

Leijten, M., Van Waes, L., Schrijver, I., Bernolet, S., & Vangehuchten, L. (2019). Mapping master's students' use of external sources in source-based writing in L1 and L2. *Studies in Second Language Acquisition*, *41*(3), 555–582. https://doi.org/10.1017/S0272263119000251

Leijten, M., Van Waes, L., Schriver, K., & Hayes, J. R. (2014). Writing in the workplace: Constructing documents using multiple digital sources. *Journal of Writing Research*, *5*(3), 285–337. https://doi.org/10.17239/jowr-2014.05.03.3

Li, J., Li, Z. L., Wu, H., & You, N. (2022). Trend, seasonality, and abrupt change detection method for land surface temperature time-series analysis: Evaluation and improvement. *Remote Sensing of Environment*, *280*, 113222. https://doi.org/10.1016/J.RSE.2022.113222

Li, S., & Yu, H. (2024). Effects of topic familiarity on L2 writing processes and behaviors. *International Journal of Applied Linguistics*, *34*(1), 348–366. https://doi.org/10.1111/IJAL.12497

Li, T., Fan, Y., Srivastava, N., Zeng, Z., Li, X., Khosravi, H., Lucia, S., Yi-Shan Tsai, A., Swiecki, Z., Gašević, D., & Tsai, Y.-S. (2024). Analytics of Planning Behaviours in Self-Regulated Learning: Links with Strategy Use and Prior Knowledge. *The 14th Learning Analytics and Knowledge Conference (LAK '24)*, 438–449. https://doi.org/10.1145/3636555.3636900

Lindgren, E., & Sullivan, K. (2019). *Observing Writing: Insights from Keystroke Logging and Handwriting*. Brill. https://doi.org/10.1163/9789004392526

Lindgren, E., & Sullivan, K. P. (2006). Writing and the analysis of revision: An overview. In K. P. Sullivan & E. Lindgren (Eds.), *Computer keystroke logging and writing: methods and applications (Studies in Writing)* (pp. 31–40). Elsevier.

Lindgren, E., Westum, A., Outakoski, H., & Sullivan, K. (2019). Revising at the leading edge: shaping ideas or clearing up noise. In E. Lindgren & K. Sullivan (Eds.), *Studies in writing: Vol. 38. Observing writing* (pp. 346–365). Brill. https://doi.org/10.1163/9789004392526_017

Lo Sardo, D. R., Gravino, P., Cuskley, C., & Loreto, V. (2023). Exploitation and exploration in text evolution. Quantifying planning and translation flows during writing. *PLOS ONE*, *18*(3), e0283628. https://doi.org/10.1371/JOURNAL.PONE.0283628

Mahlow, C. (2015). A definition of "version" for text production data and natural language document drafts. *ACM International Conference Proceeding Series*, 27–32. https://doi.org/10.1145/2881631.2881638

Mahlow, C., Ulasik, M. A., & Tuggener, D. (2022). Extraction of transforming sequences and sentence histories from writing process data: a first step towards linguistic modeling of writing. *Reading and Writing*, *37*(2), 443–482. https://doi.org/10.1007/S11145-021-10234-6/FIGURES/14

Medimorec, S., & Risko, E. F. (2017). Pauses in written composition: on the importance of where writers pause. *Reading and Writing*, *30*(6), 1267–1285. https://doi.org/10.1007/s11145-017-9723-7

Miller, B., McCardle, P., & Connelly, V. (2018). *Writing Development in Struggling Learners: Understanding the Needs of Writers across the Lifecourse*. Brill. https://doi.org/https://doi.org/10.1163/9789004346369

Roeser, J., Conijn, R., Chukharev, E., Ofstad, G. H., & Torrance, M. (2025). Typing in Tandem: Language Planning in Multisentence Text Production Is Fundamentally Parallel. *Journal of Experimental Psychology: General*. https://doi.org/10.1037/XGE0001759

Rossetti, A., & Van Waes, L. (2022a). *Dataset - Text simplification in second language: process and product data*. Zenodo. https://doi.org/10.5281/ZENODO.6720290

Rossetti, A., & Van Waes, L. (2022b). It's not just a phase: Investigating text simplification in a second language from a process and product perspective. *Frontiers in Artificial Intelligence*, *5*, 983008. https://doi.org/10.3389/FRAI.2022.983008

Sala-Bubaré, A., Castelló, M., & Rijlaarsdam, G. (2021). Writing processes as situated regulation processes: A context-based approach to doctoral writing. *Journal of Writing Research*, *13*(1), 1–30. https://doi.org/10.17239/jowr-2021.13.01.01

Saqr, M., Peeters, W., & Viberg, O. (2021). The relational, co-temporal, contemporaneous, and longitudinal dynamics of self-regulation for academic writing. *Research and Practice in Technology Enhanced Learning*, *16*(1), 1–22. https://doi.org/10.1186/S41039-021-00175-7

Tarchi, C., Villalón, R., Vandermeulen, N., Casado-Ledesma, L., & Fallaci, A. P. (2023). Recursivity in source-based writing: a process analysis. *Reading and Writing*, *37*, 2571–2593. https://doi.org/10.1007/S11145-023-10482-8

Tian, Y., Kim, M., & Crossley, S. (2024). Making sense of L2 written argumentation with keystroke logging. *Journal of Writing Research*, *15*(3), 435–461. https://doi.org/10.17239/JOWR-2024.15.03.01

Torrance, M., & Conijn, R. (2024). Methods for studying the writing time-course. *Reading and Writing*, *37*(2), 239–251. https://doi.org/10.1007/S11145-023-10490-8/METRICS

Van den Bergh, H., & Rijlaarsdam, G. (2001). Changes in Cognitive Activities During the Writing Process and Relationships with Text Quality. *Educational Psychology*, *21*(4), 373–385. https://doi.org/10.1080/01443410120090777

Van Hell, J. G., Verhoeven, L., & Van Beijsterveldt, L. M. (2008). Pause time patterns in writing narrative and expository texts by children and adults. *Discourse Processes*, *45*(4–5), 406–427. https://doi.org/10.1080/01638530802070080

Van Waes, L., & Leijten, M. (2015). Fluency in Writing: A Multidimensional Perspective on Writing Fluency Applied to L1 and L2. *Computers and Composition*, *38*, 79–95. https://doi.org/10.1016/j.compcom.2015.09.012

Van Waes, L., & Schellens, P. J. (2003). Writing profiles: The effect of the writing mode on pausing and revision patterns of experienced writers. *Journal of Pragmatics*, *35*(6), 829–853. https://doi.org/10.1016/S0378-2166(02)00121-2

Van Waes, L., van Weijen, D., & Leijten, M. (2014). Learning to write in an online writing center: The effect of learning styles on the writing process. *Computers & Education*, *73*, 60–71. https://doi.org/10.1016/j.compedu.2013.12.009

Vandermeulen, N., De Maeyer, S., Van Steendam, E., Lesterhuis, M., Van den Bergh, H., & Rijlaarsdam, G. (2020). Mapping synthesis writing in various levels of Dutch upper-secondary education: A national baseline study on text quality, writing process and students' perspectives on writing. *Pedagogische Studiën*, *97*(3), 187–236. https://psycnet.apa.org/record/2021-69640-001

Vandermeulen, N., Leijten, M., & Van Waes, L. (2020). Reporting writing process feedback in the classroom: Using keystroke logging data to reflect on writing processes. *Journal of Writing Research*, *12*(1), 109–140. https://doi.org/10.17239/JOWR-2020.12.01.05

Vandermeulen, N., Van Steendam, E., De Maeyer, S., & Rijlaarsdam, G. (2022). Writing Process Feedback Based on Keystroke Logging and Comparison With Exemplars: Effects on the Quality and Process of Synthesis Texts. *Written Communication*, *1*(40), 90–144. https://doi.org/10.1177/07410883221127998

Vandermeulen, N., Van Steendam, E., & Rijlaarsdam, G. (2020). *DATASET - Baseline data LIFT Synthesis Writing project*. Zenodo. https://doi.org/10.5281/ZENODO.3893538

Xu, C. (2018). Understanding online revisions in L2 writing: A computer keystroke-log perspective. *System*, *78*, 104–114. https://doi.org/10.1016/j.system.2018.08.007

Xu, C., & Xia, J. (2021). Scaffolding process knowledge in L2 writing development: insights from computer keystroke log and process graph. *Computer Assisted Language Learning*, *34*(4), 583–608. https://doi.org/10.1080/09588221.2019.1632901

Zhang, M., Hao, J., Li, C., & Deane, P. (2016). Classification of Writing Patterns Using Keystroke Logs. In L. A. van der Ark, D. M. Bolt, W.-C. Wang, J. A. Douglas, & M. Wiberg (Eds.), *Quantitative Psychology Research: The 80th Annual Meeting of the Psychometric Society, Beijing, 2015* (pp. 299–314). Springer. https://doi.org/10.1007/978-3-319-38759-8_23

Zhao, K., Valle, D., Popescu, S., Zhang, X., & Mallick, B. (2013). Hyperspectral remote sensing of plant biochemistry using Bayesian model averaging with variable and band selection. *Remote Sensing of Environment*, *132*, 102–119. https://doi.org/10.1016/J.RSE.2012.12.026

Zhao, K., Wulder, M. A., Hu, T., Bright, R., Wu, Q., Qin, H., Li, Y., Toman, E., Mallick, B., Zhang, X., & Brown, M. (2019). Detecting change-point, trend, and seasonality in satellite time series data to track abrupt changes and nonlinear dynamics: A Bayesian ensemble algorithm. *Remote Sensing of Environment*, *232*, 111181.https://doi.org/10.1016/J.RSE.2019.04.034S