Sentence-Centric Modeling of the Writing Process

Malgorzata Anna Ulasik[1,2], Cerstin Mahlow[1] & Michael Piotrowski[2]

[1] ZHAW Zurich University of Applied Sciences, Winterthur | Switzerland.

[2] University of Lausanne, Lausanne | Switzerland

Abstract: Linguistic modeling of the writing process has gained in importance in recent years. Existing models, both from a linguistic perspective focusing on syntactic analyses as used in natural language processing and from writing research, are insufficient to actually linguistically explain what authors do when writing and revising. Writing is linear in time, but writers are free to move to any point in the text produced so far whenever they want, thus producing specific parts (e.g., sentences) in a non-linear fashion. However, the final product is a linear sequence of sentences. We therefore can interpret writing texts as a sentence-driven process. In this new framework, this article proposes a model of the production of sentences during writing. This sentence-centric model builds on existing considerations of transforming sequences, bursts and revisions, and takes into account aspects of linearity and non-linearity on the sentence level. We present a working implementation (available as open source software) and show which information can be gained by the resulting analyses in a small case study.

Keywords: writing process, sentence-driven, sentence-centric, writing model, keystroke logging

journal of
WRITING RESEARCH

Ulasik, M.A., Mahlow,C., & Piotrowski, M. (2025). Sentence-centric modeling of the writing process. *Journal of Writing Research*, *16*(3), 463-498. https://doi.org/10.17239/jowr-2025.16.03.05

Contact: Cerstin Mahlow, School of Applied Linguistics, ZHAW Zurich University of Applied Sciences, Theaterstrasse 15c, 840 Winterthur | Switzerland - cerstin.mahlow@zhaw.ch - https://orcid.org/0000-0003-0215-5551

1. Introduction

With the widespread availability of computers usable for writing in the 1980s, writing research started to investigate how word processing supports or hinders specific strategies for writing in general and for revising in particular, compared to writing on paper (e.g., Faigley & Witte, 1981; Collier, 1983; Fitzgerald, 1987; Haas, 1989; Lutz, 1983; Piolat, 1991). Daiute and Taylor (1981) and Bolter (1989) argued that computers would allow writers to freely alter linguistic units instead of being forced to type one character after the other as when using a typewriter. Several decades into writing research, as a field we still struggle to systematically explore how writers actually tackle linguistic units like phrases and sentences.

On paper, revision traces give indications as to which parts were revised and how (e.g., Mahrer & Zuccarino, 2025), but the *sequence* of comments and edits cannot be reconstructed. When using a word processor, keystroke logging makes it possible to track and thus to reconstruct all of the writer's actions. S-notation (Kollberg, 1998) is now widely used as a standard way for (automatically) annotating higher level actions (delete, insert) beyond single characters. More importantly, it provides the sequence of these actions performed by the writer to arrive at the resulting product, the final text.

With S-notation available, the question arose as to whether writing should be considered a linear or non-linear activity. (Severinson Eklundh, 1994) raises some research questions, in particular concerning linearity and non-linearity as characteristics of local *vs.* global revisions, but the article is today mostly regarded as merely an early proposal for S-notation.

The problem, however, is the rather informal definition of *non-linearity* as "repeated insertions of sizable chunks of texts at a large distance from the current point of inscription" (Severinson Eklundh, 1994, p. 212), and of "linearity" as the composition of the text "in the order of its final presentation" (Severinson Eklundh, 1994, p. 203). Since the latter is unrealistic, it is admitted that typically "some revision occurs during the composing session" (Severinson Eklundh, 1994, p. 203).

Kollberg and Severinson Eklundh (2002) expected the automatic identification of revision episodes based on S-notation to become a way of "gaining access to complex revising patterns which may relate to task constraints and individual differences between writers" (Kollberg & Severinson Eklundh, 2002, p. 103f), but they stressed that it "needs to be combined with other information sources to lead to an understanding of composing processes." While promising at the time, the approach fails to abstract from low-level operations, and complementary high-level interpretations are difficult to formalize.

As far as time is concerned, writing is always linear: one can only perform one action after another—be it the production or deletion of a character or a larger unit. It is not possible to go back in time or to skip some minutes. However, if we consider the document or the writing space, writing can be considered non-linear, as the writer is free to go back and revise what they have already produced. Though temporally, these actions take place one after the other. Cislaru and Olive (2018) distinguish chronological and spatial linearity and interpret revision as a general activity that maintains chronological linearity but disrupts spatial linearity. So writing is linear in time and non-linear in space.

Linearity and non-linearity are thus in some sense a false dichotomy, and Severinson Eklundh (1994, p. 204) notes that in fact, "it seems reasonable to view the property of linearity on a continuous scale along which writing sessions may be placed, assuming that a variety of factors may influence the order of text production." Discovering writing strategies or revision patterns with a focus on the whole text (or the text-produced-so-far at specific points during writing) is difficult and has not yet produced satisfactory results in terms of linearity and non-linearity, as Buschenhenke et al. (2023) show.

From another perspective: some specific units, such as single sentences, can be produced both rather linearly—all revisions take place continuously—or non-linearly, when the writer moves the point of inscription elsewhere between revisions of one sentence and then returns later. Producing a specific ***burst***—what could be seen as ***chunk of text***—, respects both linearities, time and space (Cislaru & Olive, 2018). So non-linearity of writing should be defined as ***discontinuous*** writing of a ***unit***—i.e., a burst or chunk or sequence of characters, a sentence. Additionally, "sizable chunks of texts" as used by Severinson Eklundh (1994) is a rather vague concept that cannot be properly operationalized for use in analyses or models. ***Sentence***, however, is an established term in linguistics and is also used in natural language processing (NLP)—i.e., it can be operationalized. Therefore, we use ***sentences*** and parts of sentences as reference points.

Gardiner (1922) remarks: "If we were in the habit of thinking in finished sentences, surely the difficulty which is often found in formulating a thought would not be experienced at all." We apparently do not think in finished (or syntactically complete) sentences, but do we write this way?

According to Gardiner (1922), the sentence "always seems in a certain measure 'satisfactory' – satisfactory, that is to say, inasmuch as it is self-sufficient and complete psychologically and socially." Following this definition, ending a sentence may be interpreted as a result of achieving a sense of completeness by the writer. It is thus the writer's decision that we can track on the surface of the evolving text. The sense of completeness might be temporary, but at the moment of typing the final punctuation mark, it is present and can be recorded.

Research on bursts has a long tradition, starting with Kaufer et al. (1986), and there exist several elaborate models and taxonomies of bursts (Baaijen & Galbraith, 2018; Conijn et al., 2021, e.g.; Hayes, 2009). Already Hayes (2009) uses a combination of temporal and production mode information to distinguish bursts. Conijn et al. (2021) propose to look both at pausing behavior and specific actions before and after a pause. Baaijen and Galbraith (2018) differentiates between bursts that start and end with a pause (P-bursts), bursts terminated by revision activities (R-bursts), and bursts starting with moving the point of inscription (I-bursts). These concepts coincide partly with the concept of transforming sequences based on versions triggered by change in production mode or point of inscription: if temporal aspects are considered, a transforming sequence can be interrupted by pauses and split into sub-sequences of the same type (append, deletion, insertion, etc.).

Bringing both perspectives together—writing as producing sentences and writing as a sequence of bursts—, we observe that: (1) ***writing bursts*** (Kaufer et al., 1986) and ***revision episodes*** (Kollberg & Severinson Eklundh, 2002) can be "interrupted" by final punctuation marks signaling a certain completeness of thoughts within a burst or episode; and (2) ***producing a single sentence*** can be interrupted by pauses and revisions signaling cognitive activities such as planning or evaluating.

Producing sentences as a specific linguistic unit can thus be interpreted as ***one*** layer of the writing process, while bursts and revisions can be seen as ***another*** layer. Observations collected through sentence-centric modeling of writing—i.e., focusing on one layer—can be mapped onto writing bursts and revision episodes—i.e., the other layer—, aligned by a common timeline. This mapping or projection can in turn facilitate the identification of syntactic structures within bursts and revisions as units embedded in the context of whole sentences and this way lead to a better understanding of linguistic production. Gaining insights into text production at the sentence level allows us to measure low-level fluency of writing and this way helps diagnose issues and improve writing feedback (Dux Speltz & Chukharev-Hudilainen, 2021). It also creates an opportunity to investigate the relationship between linguistic production and writing quality and development (Crossley, 2020).

In this article we present a new approach for sentence-centric analysis and modeling of writing based on keystroke logging data:

1. We analyze text production and revisions as sentence-driven processes, i.e., each action performed by the writer is interpreted and presented as a step towards realizing one or more complete sentences. The proposed framework combines versions of evolving texts and their differences as constituted by a change in production mode or displacement of the current point of inscription, explicit information on sentence fragments and their status during writing, and temporal aspects and bursts. This combined perspective reflects how the sentence production process is interrupted by writing mode switches and pauses, and impacted by non-linearity.
2. Focusing on single linguistic units of the overall text—i.e., sentences—and their evolution during writing, we propose the ***sentence production cycle***. This view follows from and is rooted in the interpretation of writing as a sentence-driven process. We model production and revision of single sentences, allowing us to track the evolution of an idea into a complete sentence.

The rest of the article is structured as follows: First we report on related work concerning modeling of writing with a focus on linguistic structures in section *2* and then outline our theoretical reflections that result in the proposal of sentence-centric modeling of writing in section *3*. In section *4*, we present an implementation of sentence-centric modeling of writing processes as open-source tool by extending the existing application THEtool (Mahlow et al., 2024) and perform an evaluation of the implementation in section *5*. As a proof of concept, we report on a use case in section *6*, performed on keystroke logging data collected over several writing sessions of bachelor students writing in German. The article ends with conclusions and an outlook for future work and applications in section *7*.

2. Related work

During writing, sentences are typically assembled from ***sentence segments***, as described by Kaufer et al. (1986). The question about the role of sentences as linguistic units during writing has not been conclusively answered so far, although there have been attempts to transform writing process data into linguistic units for further research purposes.

Leijten et al. (2012), Leijten et al. (2015), and Leijten et al. (2019) aggregated process data from the keystroke level to the word level and presented a module for analyzing writing process data with natural language processing (NLP) tools. Tracking revisions at the word level only can already yield important insights on the writing process (see, e.g., Serbina et al., 2017, on word class changes during production). Mahlow et al. (2024) and Miletic et al. (2022) investigated linguistic structures beyond words, i.e., sentences. Miletic et al. (2022) proposed a methodology for semi-automatic keystroke log annotation, which relies on reconstructing and annotating intermediate versions of the text. While this approach does provide a wider linguistic context, basing the methodology on the whole text as a unit of analysis makes it more difficult to identify evolution within individual sentences across different text versions. Mahlow et al. (2024) presented a method and a tool for fully automatic analysis of keystroke logging data and extracted intermediate text versions aggregated as text history. From this, they extracted all intermediate versions of all sentences and aggregated them in corresponding sentence histories. All intermediate drafts of each sentence can be parsed with NLP tools to investigate the evolvement of the text on the syntactic level.

Keystroke logs have been extensively explored focusing on the role of pauses in writing (Alves et al., 2007; Foulin, 1995; e.g., Matsuhashi, 1981). Pauses are used to segment writing process data into ***bursts*** (Chenoweth & Hayes, 2001). There is evidence that pauses are cognitively motivated (Olive, 2012) and that there are specific relations between distribution and duration of pauses defining bursts on the one hand and the linguistic content—and thus syntactic structure—that is being produced on the other hand (Immonen & MΔkisalo, 2017; Medimorec & Risko, 2017). Kaufer et al. (1986), Hayes (2009), Olive and Cislaru (2015), and Cislaru and Olive (2018) investigated the syntactic structure of bursts as a mapping of production data in form of bursts and linguistic structure. Ivaska et al. (2025) focused on pauses in and between words as indicator for proficiency when writing in an L1 and L2.

Cislaru and Olive (2018) manually examined syntactic properties of bursts and showed that bursts do *not* generally coincide with traditionally accepted syntactic structures. The bursts they observed are highly heterogeneous and more than 50% are syntactically incomplete. Some of these incomplete structures are identified as recurrent and potentially having specific functions in the writing process. The authors argued for the status of bursts as *units of linguistic production* as opposed to units of linguistic reception (Cislaru & Olive, 2018). Gilquin (2020) examined bursts through the lens of Construction Grammar and found that some—but not all—bursts correspond to units considered *constructions* in linguistics. Using robust statistical analysis, Feltgen et al. (2022) and Feltgen et al. (2023) showed that there is a relation between the linguistic content being produced and the segmentation of the writing process data into bursts.

Note, however, that both Cislaru and Olive (2018) and Gilquin (2020) based their work on a manual examination of a relatively small set of data. Feltgen et al. (2022), Feltgen et al. (2023), and Feltgen & Lefeuvre (2025) focused on phenomena that are represented by simple wordlists (the conjunction *et* ('and') and the clitic subject in French, respectively) and therefore relatively easy to track in a corpus.

We extend those ideas to develop sentence-centric modeling of the writing process as presented in the following section.

3. Sentence-centric models of writing

Modeling writing from a *sentence-centric* perspective interprets writing as a process of producing sentences that together form a text. Each action performed by the writer is a step towards realizing one or more complete sentences. Modeling this process means investigating each transformation of the text for its impact on sentences.

In this article, we present a sentence-centric model of writing: We understand writing a text as a *sentence-driven process*. In this framework, we propose a model to analyze the production of individual sentences in a *sentence production cycle*. This cycle covers the transcription of an initial idea (or parts of it) through completion and potential revision of the sentence and intermediate fragments to its eventual realization as present in the final product, including its complete removal.

This model allows us to consider specific isolated parts of the writing process: an individual sentence or events during a shorter period of time. The sentence production cycle does not account for relations between sentences but handles every sentence individually. Modeling the *entire* writing process as sentence-driven takes into account relations between sentences and their position in the text—both in the text produced so far (TPSF) during writing as well as in the final product. The model focuses exclusively on behavioral and thus directly observable actions and data: acts performed by the writer with their writing tool (e.g., keyboard and mouse) and visible effects of these acts to the text (e.g., on the screen).

In the following sections, we first explain basic concepts used as foundations (section *3.1*) and then explain the model in detail (sections *3.2* and *3.3*). In section *4* we report on the first attempt to implement the model computationally.

3.1 Basic concepts

Text history and transforming sequence

Following Mahlow et al. (2024), we use the concept of *text history*—extracted and reconstructed from writing process data—as basic source of information about the evolution of the whole text during writing. The text history contains all intermediate *versions* of a text. We follow Mahlow (2015) and define a version as the current TPSF when a change in production mode (append, delete, insert, paste, replace) occurs, or when the writer moves the point of inscription; for example, systematically going through a TPSF and inserting a comma before "and" will result in several versions at each new instance, even though the action is always *insertion*—i.e., no change in production mode—and could thus be understood as linear revision.

Two adjacent versions differ on the surface: the current version is shorter or longer than the previous one as the writer removed or added at least one character. This visible difference results from preceding actions carried out by the writer and can be recorded as process data. Both elements—the difference as observable on screen, as well as the sequence of keystrokes involved—constitute the *transforming sequence*, which is detected through the analysis of writing process data. It also includes temporal information and could be enriched with additional information from other sources.

Sentence produced so far (SPSF)

As proposed by Ulasik and Miletić (2024), each text version—i.e., each TPSF—can be segmented into one or more *sentences produced so far* (SPSF). Like the whole text, each sentence of this text has a distinct length and content at a given moment in time. A TPSF is thus an ordered sequence of SPSFs. We distinguish two types of SPSFs: (1) *sentences* (SEN), i.e., complete and correct sentences, and (2) *sentence candidates* (SEC), i.e., fragmentary or incorrect sentences (for details see Ulasik & Miletić, 2024). The distinction is based on both observable behavioral data (recorded as keystrokes) and the text itself as the visible result of the writing process at a given moment.

In the remainder of this article, we will use the abbreviation *SEN* to denote a sequence of characters fulfilling the specific sentence definition by Ulasik and Miletić (2024), notably correctness and completeness. We use the term *sentence* in the wider sense of a unit of meaning produced during writing in general.

Sentence histories

All versions of a particular sentence (i.e., all detectable intermediate SPSFs) constitute its *sentence history* (Mahlow et al., 2024). Sentence histories are extracted from text histories and can be stored as hierarchical data structures and further enriched with, e.g., POS information or aggregated time. As text histories and sentence histories are stored in a similar data structure, we can perform similar analyses for SPSF as for TPSF: The difference between two SPSF can be extracted on the surface level (product data), and the corresponding writer's actions (process data) are available, too. We can refer to those as ***sentence transforming sequences***. Note that actions may include the complete removal of this sentence, so that it will not be visible in the final product.

Additionally, note that ***versions of sentences*** mean that if $V_i(S_n)$ is a particular version V_i of the sentence S_n, and V_{i-1} and V_{i+1} are the preceding and following versions, respectively. Contrary to the versions of the text as stored in the text history, the sequence of versions is not strictly contiguous: $V_2(S_n)$ always follows $V_1(S_n)$, but the writer might have produced or revised other sentences in between. Therefore, each sentence history consists of a set of sentence transforming sequences and contains a unique sentence ID and the specific TPSF ID for each stored SPSF.

3.2 Modeling writing as a sentence-driven process

Sentences are typically constructed "from proposed sentence parts in a complex activity involving idea generation, evaluation, planning, and reading the text produced so far" Kaufer et al. (1986). Modeling writing as a ***sentence-driven process*** thus allows us to ***trace*** and ***track*** this process on the syntactic level. Each action performed by the writer is interpreted as a step towards realizing one or more complete sentences. The model comprises three layers: the ***transformation layer***, the ***sentence layer***, and the ***burst layer***, which can be mapped or projected onto each other (see Figure 1).

The starting point is the text history extracted from writing process data. Two adjacent versions of the TPSF differ by a transforming sequence (see ***3.1***). These transforming sequences constitute the transformation layer. The derivation of more abstract and intentional (or semantic) editing operations can be done by ***interpreting*** annotated process data, i.e., by analyzing the transformation layer. One example are ***revision episodes*** as defined by Kollberg and Severinson Eklundh (2002): They distinguish three classes of episodes of revision based on the location of revision and the interleaving of revisions: (a) episodes with multiple revisions at the same cursor position, (b) revisions that are interrupted by another revision, and (c) interruption of the current writing process to edit a passage that has already been written and then continue writing at the interruption point (Kollberg & Severinson Eklundh, 2002, p. 94f).

Each text version in the text history can be segmented into sentences produced so far (SPSF). This constitutes the sentence layer. From another perspective, the writing process and thus the keystroke logging data can be structured into bursts. Bursts thus form another layer, the burst layer.

Both the burst and the sentence layers can be overlaid on the transformation layer, i.e., the transforming sequence is split into complete and correct sentences (SENs) and sentence candidates (SECs), ***and*** into bursts. This mapping yields a combined view, at the same time accounting for changes in production mode ***and*** for temporal aspects. In the following we look at each mapping individually in more detail.

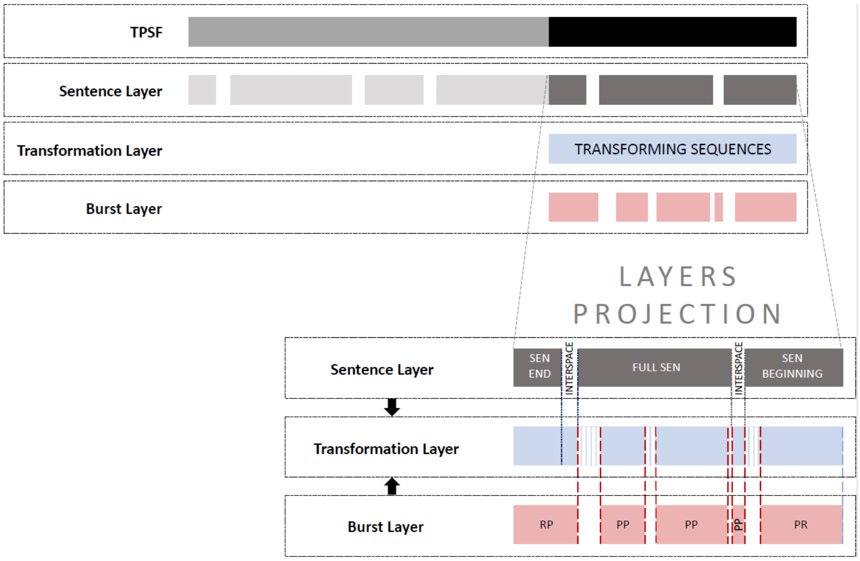


Figure 1. Visualization of the concept of layers and their projection on each other.
In the burst layer: RP refers to bursts starting with a revision and
ending with a pause; PR refers to burst starting with a pause and ending with a revision;
PP refers to bursts starting and ending with a pause.

3.2.1. Mapping sentence layer and transformation layer

Mapping the sentence layer onto the transformation layer means identifying sentences and sentence fragments in transforming sequences. This allows us to identify (1) ***how many*** and (2) ***which*** sentence parts (SPSFs) are impacted by a specific transforming sequence.

Depending on the extent of the impacted SPSFs, a particular transforming sequence can have one of the four ***scopes***: (1) in-sentence, (2) uni-sentence, (3) cross-sentence, or (4) multi-sentence. An ***in-sentence transforming sequence*** impacts exactly one SPSF. It consists in either ***producing*** a SEC (sentence candidate, i.e., only a part of a sentence) or alternatively, ***revising*** an existing SEC or an existing SEN (a complete and correct sentence). A ***uni-sentence transforming sequence*** results in ***producing*** a new SEN from scratch. The remaining two classes always impact more than one SPSF: ***cross-sentence transforming sequences*** affect parts of exactly two SPSFs. A ***multi-sentence transformation*** impacts at least three SPSFs.

Additionally, it is determined which sentence parts are impacted by a given transforming sequence. We use the term ***sentence segment*** for denoting a sequence of words ***within*** a sentence and distinguish between three categories of ***sentence segments*** based on their position in relation to a given sentence:

1. ***sentence beginning*** (B): a sequence of arbitrary length that starts with a sentence start but is not terminated by sentence-final punctuation,
2. ***sentence middle*** (M): a sequence of arbitrary length that do neither contain a sentence start which contains a sentence start nor a sentence end,
3. ***sentence end*** (E): a sequence of arbitrary length that ends with sentence-final punctuation but does not contain a sentence start. Additionally, a sequence containing both sentence beginning and sentence end is classified as a complete sentence (C).

Table ***1*** gives a combined overview of patterns of sentence segments and the scope of the corresponding transforming sequence. Note that for the scopes in-sentence, uni-sentence, and cross-sentence all possibilities are listed. The shortest possible multi-sentence transforming sequences—i.e., sequences containing more than one complete sentence—are E-C-C, C-C-B, E-C-C-B, and C-C-C. Longer ones contain additional complete sentences.

Table 1: Overview of all possible patterns of sentence segment sequences and the scope of the corresponding transforming sequence.

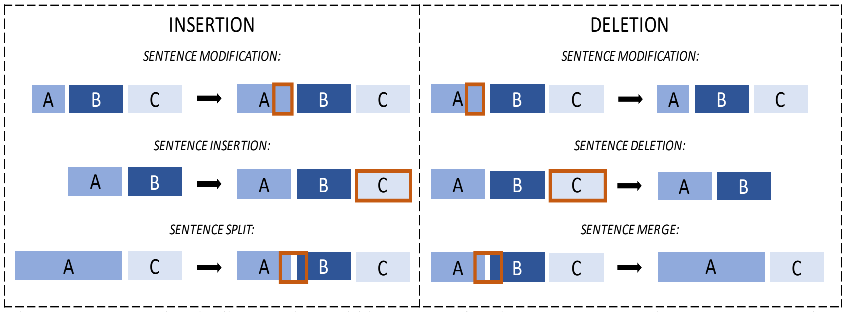
Label	**Sentence segments in TS**	**Scope of TS**
M	sentence middle	in-sentence
B	sentence beginning	in-sentence
C	complete sentence	uni-sentence
C-B	complete sentence + sentence beginning	cross-sentence
C-C	complete sentence + complete sentence	cross-sentence
C-C-B	complete sentence + complete sentence + sentence beginning	multi-sentence
C-C-C	complete sentence + complete sentence + complete sentence	multi-sentence
E	sentence end	in-sentence
E-B	sentence end + sentence beginning	cross-sentence
E-C	sentence end + complete sentence	cross-sentence
E-C-C	sentence end + complete sentence + complete sentence	multi-sentence
E-C-B	sentence end + complete sentence + sentence beginning	multi-sentence

The following restrictions apply:

1. In-sentence and cross-sentence transforming sequences always mean that only sentence segments rather than complete sentences are produced or removed.
2. Uni-sentence transforming sequences never impact sentence segments, they always produce or delete one single SEN.

Multi-sentence transforming sequences may contain both SENs and sentence segments. Following Mahlow et al. (2024), transforming sequences ***affect*** sentences by (1) modification, (2) deletion, (3) insertion, (4) split, (5) merge, and any combination thereof, see figures *2* and *3*. We call these possibilities the ***effect of the transforming sequence***. Note that the ***type*** of a transforming sequence (append, insert, delete, paste, replace) is different from its ***effect***: as shown in Figure *2*, an insertion can modify a sentence, insert a sentence, or split a sentence.

To summarize, mapping the sentence layer onto the transformation layer allows us to classify all transforming sequences according to the number of sentences they impact, and breaking them down into sentence segments.



*F*igure 2. Sentence-level edit operations within one transforming sequence targeting one sentence. The transforming sequence is represented by an orange box. The left column illustrates insertions of transforming sequences; the right one shows deletions. The impact of the transforming sequence depends on its position and content.

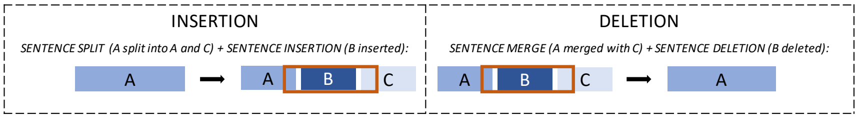

INSERTION
SENTENCE SPLIT (A split into A and C) + SENTENCE INSERTION (B inserted):
A
B
C
DELETION
SENTENCE MERGE (A merged with C) + SENTENCE DELETION (B deleted):

Figure 3. Sentence-level edit operations resulting from one transforming sequence containing more than one sentence.

Thus, transforming sequences can be classified on three levels with respect to sentences:

1. the type of the transforming sequence (append, delete, insert, paste, replace),
2. the scope of the transforming sequence (in-sentence, uni-sentence, cross-sentence, multi-sentence), also considering the sentence segment category (beginning, middle, end),
3. the effect of the transforming sequence (modification, deletion, etc.) on the sentence level.

As a result, we can model the exact impact of each transformation on sentences under production.

3.2.2. Mapping burst layer and transformation layer

Mapping the burst layer onto the transformation layer means identifying and categorizing bursts in transforming sequences. Defining and analyzing bursts using temporal criteria involves detecting specific pauses during writing. These pauses are characterized by length and location. As we aim at a sentence-centric model of writing, we assign a special role to pauses directly preceding and following an SPSF. We call the former a ***pre-sentential pause***, marking a pause made before a sentence beginning. We call the latter a ***post-sentential pause***, marking a pause made after the sentence-final punctuation was produced. Generally, all pauses between sentences might systematically have the dual status of pre- and post-sentential pauses. Ulasik and Miletić (2024) propose to use the production of specific characters (e.g., newline, space after punctuation marks) signaling the start of a sentence to distinguish post- from pre-sentential pauses.

A transforming sequence is by definition a sequence of characters produced between switches of production mode, thus it always begins with a revision burst (R-burst) and also always ends with a revision burst. It may be interrupted by pauses suitable to distinguish pause bursts (P-burst). Additionally, it is possible that a switch in production mode overlaps with a pause. In such a case, the given sub-sequence belongs to both categories.

For example, the action of a writer who inserts a longer part in the middle of the TPSF over several seconds or minutes, interrupted by several pauses without a change in production mode, and who then moves the cursor elsewhere, will be considered a sequence of insertions all belonging to a single transforming sequence: inserting text. Looking at those sequences as instances of bursts, we can classify the first as RP-burst (a burst starting with a revision and ending with a pause), the last one as PR-burst (a burst starting with a pause and ending with a revision), and the bursts in between as PP-bursts (a burst starting and ending with a pause).

So far, we have only distinguished bursts due to pauses of a specific length and unspecified revisions. The length of pauses used in analyses depends on research purposes or information needs; by default, we follow Van Waes and Leijten (2015) and interpret—for the purpose of classifying time-based bursts—interruptions shorter than 2 seconds as related to transcription activities rather than to higher level cognitive activities during writing. For revision bursts, we make use of available information for the transforming sequence, in particular the ***type*** (append, insert, delete, paste, replace). This allows us to incorporate definitions and distinctions as proposed and used by Baaijen and Galbraith (2018) or Conijn et al. (2021).

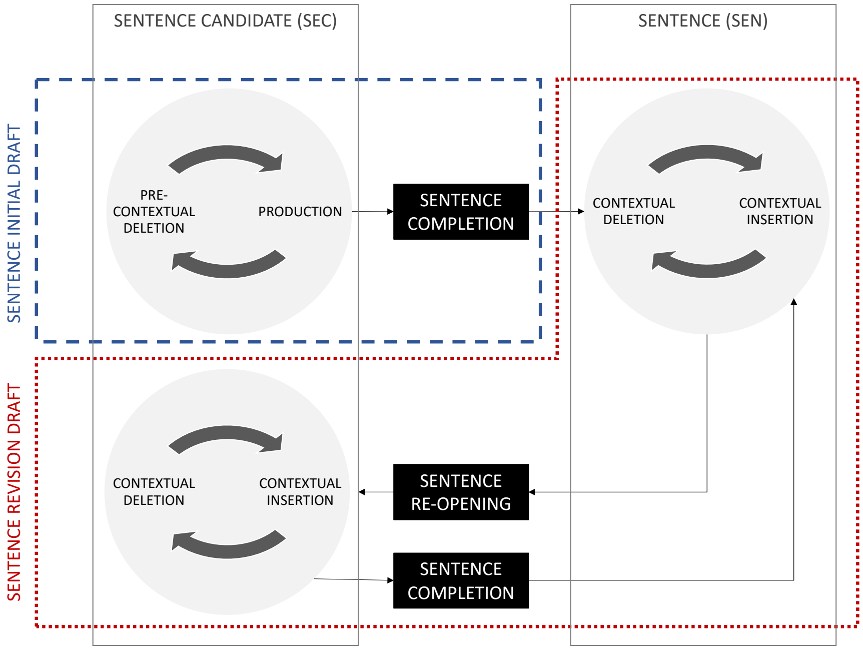


Figure 4. Sentence production cycle.

3.3 Modeling the sentence production cycle

Mapping the sentence layer onto the transformation layer allows us to draw conclusions about how writers work on sentences in general, as explained in section *3.2.1*, i.e., we explore the relation from the perspective of individual transformation sequences. This mapping also allows us to investigate how individual sentences are produced and revised, i.e., we explore the relation using the sentence level as reference point. We propose to model this view as ***sentence production cycle***. This model permits tracking the evolution of an idea into a complete sentence, focusing on individual sentences and their aggregated sentence histories extracted from text histories (see *3.1*).

We use the differentiation of ***sentence stages*** as basis. Following Baaijen et al. (2012), we distinguish ***sentence initial draft*** and ***sentence revision draft***. This distinction also reflects the semantic aspect of sentence definitions proposed in the literature and the notion of sentence completeness: a sentence is seen as a "meaning unit" (Вьhler, 1918) "capable of expressing a complete thought" (Noreen, 1903), "self-sufficient and complete psychologically" (Gardiner, 1922).

A ***sentence initial draft*** is the result of producing a first draft of a sentence. It is finished as soon as the writer enters the final punctuation mark at the end of the sentence for the first time. We interpret this as a signal of the writer achieving a sense of completeness. The sentence initial draft encompasses linear production as well as deletions and insertions—which can be referred to as pre-contextual revisions, i.e., all "changes made at the leading edge of the text before a full textual context has been externalised" (Lindgren et al., 2019, p. 349). In our terminology, a sentence initial draft comprises all SEC versions until a sequence of words achieves the status of a SEN for the first time. The corresponding *sentence transforming sequences* types are all pre-contextual only.

A ***sentence revision draft*** contains any changes applied to a sentence at some time ***after*** it achieved SEN status, including its complete deletion. We consider the sentence revision draft a result of *contextual revisions* (see Lindgren et al., 2019). Contextual revision might downgrade a SEN into a SEC if the sentence is no longer complete or correct.

Thus we can distinguish five types of sentence-transforming sequences (STS): (1) production, (2) pre-contextual deletion, (3) pre-contextual insertion, (4) contextual deletion, (5) contextual insertion.

Sentence histories containing all sentence transforming sequences of all sentences of a text may be of different length and complexity. In the simplest case, a sentence is produced immediately as a SEN within one uni-sentence or multi-sentence transforming sequence and never revised. The corresponding sentence history would then contain only one sentence transforming sequence; there is only a sentence initial draft. In more complex cases, the sentence is first produced completely and then revised once or several times; the sentence histories contain both sentence initial draft(s) and sentence revision draft(s). Revisions may take place immediately after the SEN is completed and before next sentences are produced, or much later. Figure *4* provides an overview of all possible steps in the sentence production cycle.

Tables *2*, *3*, *4*, and *5* show histories of four sentences as examples of different sentence production cycles. Table *2* shows a sentence history without revision draft: the SEN is accomplished as a result of three edits and also appears in this version in the final text. In the second history (table *3*), the sentence is produced in TPSF 85 as a result of a cross-sentence transforming sequence and then appears as revised in TPSF 98 and 99 (i.e., after other writing activities at other positions in the text) as a result of contextual deletion and insertion. In the third history (table *4*), the sentence production is interrupted by two pre-contextual deletions before the sequence achieves SEN status. Immediately after that, the writer revises the SEN by replacing the word *also* (contextual deletion) with the word *somit* (contextual insertion). The revision takes place within the sentence frame (between the sentence-initial capital letter and the sentence-final punctuation), and the sentence is still syntactically correct; hence the sequence never loses SEN status. Table ***5*** shows an excerpt from another sentence history where production is interrupted by pre-contextual deletion. In TPSF 141, the SEC has been completed into a SEN, and the writer continues text production. They produce another 137 text versions until they re-open the sentence again by performing contextual deletion and removing the whole subordinate clause (*, die dieser Studiengang ermöglicht, sind genau das, was ich im Leben machen möchte.*) resulting in TPSF 279. The status of the SPSF returns to SEC, but all following revisions are still considered contextual. In the TPSF 282, the sentence achieves again SEN status.

Table 2: A sentence history example comprising only sentence-initial drafts. SEC stands for sentence candidate and SEN for sentence. As soon as the SPSF achieves SEN status, its production is finished, and the writer never comes back to revise the sentence. "TPSF ID" stands for text version number, "Pos in text" is the position of the sentence in the corresponding sentence version (the 19th sentence in the final text). "SPSF" is the sentence version, and "ST" is sentence stage: the segment of transforming sequence which impacts the given sentence. The first version can be translated as "In my opinion, the bachelor's degree program takes all components into account in exactly the right way␣", i.e., with a trailing space. The writer then deletes the space and continues with ", so that they ultimately form a whole."

TPSF ID	**Pos in text**	**SPSF**	**STS**	**Operation**	**SPSF type**
83	19	Der Bachelor Studiengang berьcksichtigt in meinen Augen all die wichtigen Komponenten genau richtig ␣	Der Bachelor Studiengang berьcksichtigt in meinen Augen all die wichtigen Komponenten genau richtig␣	Production	SEC
84	19	Der Bachelor Studiengang berьcksichtigt in meinen Augen all die wichtigen Komponenten genau richtigX	␣	Pre-contextual deletion	SEC
85	19	Der Bachelor Studiengang berьcksichtigt in meinen Augen all die wichtigen Komponenten genau richtig, sodass sie letztendlich ein Ganzes ergeben.	, sodass sie letztendlich ein Ganzes ergeben.	Production	SEN

Table 3: A sentence history example. See table 2 for abbreviations. The first version can be translated as "Theory and people are in the foreground."; "in the foreground." is then deleted, and instead "are not neglected." is inserted, so that TPSF ID 99 reads: "Theory and people are not neglected."

TPSF ID	Pos in text	SPSF	STS	Operation	SPSF type
85	20	Die Theorie sowie der Mensch stehen im Vordergrund.	Die Theorie sowie der Mensch stehen im Vordergrund.	Production	SEN
98	20	Die Theorie sowie der Mensch X.	stehen im Vordergrund.	Contextual Deletion	SEN
99	20	Die Theorie sowie der Mensch kommen nicht zu kurz.	kommen nicht zu kurz.	Contextual Insertion	SEN

Table 4: A sentence history example comprising both sentence initial and sentence revision draft (TPSF IDs 16-20 und TPSF IDs 21-22 respectively). See table 2 for abbreviations. The first SEN (ID 20) reads: "I stand thus in direct relation to two languages, two countries, two cultures, and two societies.". IDs 17 and 19 are corrections of typos in 16 and 18. In ID 21 also 'thus' is deleted and in ID 22 the synonymous, but more formal adverb somit is inserted in its place.

TPSF ID	Pos in text	SPSF	STS	Operation	SPSF type
16	7	Ich stee a	Ich stee a	Production	SEC
17	7	Ich stX	ee a	Pre-contextual deletion	SEC
18	7	Ich stehe also in direkter Rea	ehe also in direkter Rea	Production	SEC
19	7	Ich stehe also in direkter ReX	a	Pre-contextual deletion	SEC
20	7	Ich stehe also in direkter Relation zu zwei Sprachen, zwei LΔndern, zwei Kulturen und zwei Gesellschaften.	lation zu zwei Sprachen, zwei LΔndern, zwei Kulturen und zwei Gesellschaften.	Production	**SEN**
21	7	Ich stehe X in direkter Relation zu zwei Sprachen, zwei	also	Contextual Deletion	SEN

TPSF ID	Pos in text	SPSF	STS	Operation	SPSF type
		LΔndern, zwei Kulturen und zwei Gesellschaften.			
22	7	Ich stehe somit in direkter Relation zu zwei Sprachen, zwei LΔndern, zwei Kulturen und zwei Gesellschaften.	somit	Contextual insertion	SEN

Table 5: Excerpt from a sentence history comprising both sentence initial and sentence revision draft (TPSF IDs 139-141 und TPSF IDs 279-282 respectively). SEC stands for sentence candidate and SEN for sentence. In TPSF 279 the SEN is transformed back into a SEC as a result of contextual deletion. "TPSF ID" stands for text version number, "Pos in text" is the position of the sentence in the corresponding sentence version, "SPSF" is the sentence version, and "ST" is sentence stage: the segment of transforming sequence which impacts the given sentence. The first SEN (ID 141) can be translated as: "I have always enjoyed passing on knowledge and the career prospects that this degree program offers are exactly what I want to do in life." ID 140 corrects a typo in the unfinished ID 139. ID 279 removes the relative clause, so that only "I have always enjoyed passing on knowledge and the career prospects that this degree program offers" remains. After a false start to complete the sentence, the final version (ID 282) reads: "I have always enjoyed passing on knowledge and the career prospects that this degree program offers are very appealing."

TPSF ID	Pos in text	SPSF	ST	Operation	SPSF type
139	9	Ich habe schon immer gern Wissen weitergegeben und die beruflichen Perspektiven, die dieser Studiengang ermцglicht, sind genau das, was ich im Leben machen mцct	m Leben machen mцct	Production	SEC
140	9	Ich habe schon immer gern Wissen weitergegeben und die beruflichen Perspektiven, die dieser Studiengang ermцglicht, sind genau das, was ich im Leben machen mцcX	t	Pre-contextual deletion	SEC

TPSF ID	Pos in text	SPSF	ST	Operation	SPSF type
141	9	Ich habe schon immer gern Wissen weitergegeben und die beruflichen Perspektiven, die dieser Studiengang ermuglicht, sind genau das, was ich im Leben machen muchte.	hte.	Production	SEN
279	9	Ich habe schon immer gern Wissen weitergegeben und die beruflichen PerspektivenX	, die dieser Studiengang ermuglicht, sind genau das, was ich im Leben machen muchte.	Contextual deletion	SEC
280	9	Ich habe schon immer gern Wissen weitergegeben und die beruflichen Perspektiven f	f	Contextual insertion	SEC
281	9	Ich habe schon immer gern Wissen weitergegeben und die beruflichen Perspektiven X	f	Contextual deletion	SEC
282	9	Ich habe schon immer gern Wissen weitergegeben und die beruflichen Perspektiven gefallen mir sehr gut.	gefallen mir sehr gut.	Contextual insertion	SEN

4. A first implementation

The analyses presented in the examples above can be carried out manually by carefully annotating and aggregating keystroke logging data. However, this would be both error-prone and restricted to rather small amounts of data. Following Weizenbaum (1976), who showed the connection between theory, model, and code, we provide a working implementation to test the general power of our models and the theoretical reflections outlined above.

We have implemented the model proposed in section *3* as a software component—called SCM (Sentence-Centric Modeling) component—that parses and analyzes keystroke logging data. It serves as a step towards developing and testing writing models in the form of computer programs operating on writing process data, as suggested by Hayes (2012). The component is an extension to THEtool (Text History Extraction Tool, available at *https://github.com/mulasik/wta*), an open-source application implemented in Python for parsing raw keystroke logging data to generate text and sentence histories (Mahlow et al., 2024).

4.1 Extended text history: writing as sentence-driven process

The starting point for automatically modeling writing as a sentence-driven process are text histories. We use the data stored in text histories generated by THEtool to model the three layers: transformation, sentence, and burst layer. THEtool first parses raw keystroke logging data and aggregates it into transforming sequences. This data, stored in hierarchical data structures, is then used as input for further processing steps.
To create the text history of a writing session, THEtool collects the following information: the content of the transforming sequence (sequence of characters constituting the difference to the previous version), the operation performed (whether the sequence was produced or removed), the start and end positions, the start and end times, as well as pauses preceding each character in the transforming sequence. Transforming sequences at consecutive positions in the text are aggregated to build a linear transforming sequence. Also included is temporal information, which can be further aggregated and analyzed for building the burst layer. As described by Mahlow et al. (2024), this data is enriched with additional information, e.g., part of speech (POS) information produced by natural language processing (NLP) tools (we currently use SpaCy).

The transformation layer is implemented as *extended text history* (ETH). All further layers are then mapped onto the ETH by storing the respective information as attributes. This makes the ETH a very rich hierarchical data structure that can be used for further analyses and visualizations.

The sentence layer results from linguistic processing of the data derived from text histories. The SCM component segments each text version (stored as a sequence of characters) into a sequence of sentences produced so far (SPSFs) and interspaces, i.e., SECs, SENs, SINs and PINs (see section *3.1* and Ulasik and Miletić (2024) for more details). Next, it detects which SPSFs were impacted by the given transforming sequence and labels them as either modified, new, or deleted. The sequence of segmented and labeled SPSFs provided by the SCM component serves as input for the subsequent mapping step. The SPSF labels allows us to determine how many SPSFs were impacted by the transforming sequence and correspondingly determine the scope of the transforming sequence as in-sentence, uni-sentence, cross-sentence, or multi-sentence. In the next processing step, the SCM component analyzes the impacted SPSFs to detect which of their parts were changed, deleted or are new. It distinguishes between sentence beginning, middle, and end, as well as a complete sentence. After this step, the transforming sequence is enriched with new information: the list of sentence segments which it affects. This information is then stored as attributes of the ETH for the particular writing session.

The burst layer is created from information on pause duration preceding each transforming sequence and within that transforming sequence. The duration of a pause considered relevant for bursts depends on the purpose of the respective analysis, hence our implementation allows for setting different thresholds. Based on this setting, each transforming sequence is segmented into bursts. The resulting information on bursts for each transforming sequence is stored as attributes of the ETH.

As writing is non-linear in space, THEtool tracks the position of each transforming sequence in relation to the preceding one to aid with reconstructing and understanding the process. The implementation of this functionality is limited in the current version of the SCM component and only distinguishes between the end of the text and any position in the middle of the text. Thus, actions performed at the end of the text can be aggregated into a linear transforming sequence. This additional information is also stored as attributes of the ETH.

4.2 Extended sentence histories: the sentence production cycle

The implementation of the sentence production cycle model outlined in section *3.3* relies on existing functionality of THEtool for generating sentence histories from the text history of a writing session. This step can be performed on the original text history or on the extended text history (ETH) containing additional information as described in section *4.1*. All additional information from modeling the sentence production cycle of each sentence is stored as attributes to its sentence history in the *extended sentence history* (ESH). Note that this structure is different from the sentence layer stored as attribute to the text history in the ETH.

We have extended THEtool to categorize each sentence transforming sequence by using general information from the corresponding transforming sequence (TS) at the text level as one of (1) production (TS type append or paste at sentence end), (2) pre-contextual deletion (TS type delete at sentence end), (3) pre-contextual insertion (TS type insert or paste at sentence middle), (4) contextual deletion (TS type delete), or (5) contextual insertion (TS type append, insert, or paste). The SCM component also assigns the appropriate sentence stage (sentence initial draft or sentence revision draft) to each SPSF in the sentence history (see section *3.3*).

As long as the sentence history does not contain a SEN, the sentence transforming sequence belongs to one of the first three categories and the SPSF is a sentence initial draft. Sentence transforming sequences *after* the first SEN was produced are all classified as categories 4 or 5: these SPSFs are considered sentence revision drafts. The category is stored as attribute of a sentence transforming sequence.

4.3 Limitations and challenges

While the implementation of the SCM component closely follows the theoretical approach outlined above, there are still some limitations and parts that have not yet been implemented.

One of the limitations concerns the method for SPSF identification. The distinction between SEN and SEC is currently based on strictly formal criteria: a SEN starts with a capital letter and ends with sentence-final punctuation. All other strings of characters are considered to be SECs. From a linguistic point of view, this definition of a sentence is oversimplified compared to sentence definitions in the literature, and it is obviously not applicable to languages using unicase alphabets. However, this simplification is necessitated by the absence of other data on which the TPSF segmentation could be based: since we do not have access to writers' intentions, we can only make assumptions based on behavioral data—i.e., keystroke logs—and formal properties of the produced text. The evaluation in Ulasik and Miletić (2024) indicates that this approach nevertheless yields good results and provides a solid basis for more advanced analyses.

A further limitation concerns the definition of sentence histories. We assume that a sequence of characters belongs to the same sentence history as the previous one if at least one of the characters from the original sequence remains. If the initial sequence is completely deleted, the new sequence will be considered as the beginning of a new sentence history, which can lead to starting new sentence histories even though an SPSF is actually a realization of the same idea the writer had transcribed in the deleted sentence. A potential solution could be to apply the approach proposed by Conijn et al. (2021) for the automatic extraction of full revision events from keystroke logs. If we could integrate a mechanism for detecting deletions and character productions that build parts of the same revision, we could potentially introduce a more advanced definition of a sentence history in our framework.

A complete extraction of revisions could also contribute to the improvement of extended text histories. It would allow us to eliminate deletion-insertion sequences which form a replace operation and should be stored as one transforming sequence—i.e., *replace*—in the text history. This way, transforming sequence would better approximate the writer's real intention.

Two aspects of the framework have not yet been implemented in the current version of the software. First, our application is not capable of detecting linear transforming sequences for edits performed in the middle of the text. As a result, a full representation reflecting text production non-linearity is not yet available: we can only model transforming sequences that are performed at the end of the TPSF. Second, the current architecture of our implementation does not allow for projecting bursts onto sentence segments. Our software detects bursts within transforming sequences and splits transforming sequences into sentence segments, but the linking between pauses and sentence segments is not available yet.

However, we are continuously extending THEtool; the two missing functionalities are planned to be included in the forthcoming release of the application.

5. Evaluation of algorithms and implementation

To verify the reliability of the implementation of our model, we performed an evaluation of three core algorithms: (1) classification of transforming sequences; (2) extraction of sentence segments from transforming sequences; (3) detection of categories of sentence transforming sequence.

5.1 Evaluation corpus

The evaluation corpus is derived from keystroke logs from 4 randomly selected writing sessions (named 1A1, 2A1, 4A1, 5A1) from THEcorpus (see section *6.1* for more details on the corpus). The evaluation corpus contains in total 535 transforming sequences, 93 sentence histories, and a total of 591 SPSFs. Table *6* provides statistics for each of the four texts.

Table 6: Statistics for the texts constituting the evaluation corpus.

Text	keystrokes	transforming sequences	sentence histories	total sentence versions
1A1	2706	97	29	111
2A1	3455	195	21	205
4A1	3254	140	25	159
5A1	2927	103	18	116
Total	12,342	535	93	591

The basis for evaluating the classification of transforming sequences and extracting sentence segments are the text histories generated by THEtool. We manually assigned each transforming sequence from the text history to a corresponding scope: in-sentence, uni-sentence, cross-sentence, or multi-sentence (see section *4.1*). We then detected and categorized the sentence segments that make up each transforming sequence, distinguishing four categories: sentence beginning, sentence middle, sentence end, and complete sentence (see section *4.1*). The algorithm for detecting categories of sentence transforming sequences was also evaluated on the basis of sentence histories extracted with THEtool. We annotated each SPSF with one of the categories: production, pre-contextual deletion, pre-contextual insertion, contextual deletion, and contextual insertion. Table *7* provides an example of the manual annotation of a sentence history.

Table 7: Annotation of one sentence history from the evaluation corpus with the scope of the transforming sequences (Scope), the sentence segments, the transforming sequence, and the category of the sentence transforming sequence (STS). B stands for sentence beginning, M is sentence middle, and E denotes sentence end. Note that in this example, the transforming sequence is

identical with the sentence transforming sequence as the TS affects just this one sentence. The complete sentence can be translated as "When learning a new language, the language and the environment inevitable run into each other."

SPSF	Scope	Sentence Segment	Transforming Sequence	Category of STS
Bei der Aneignung einer neuen SS	in-sentence	B	Bei der Aneignung einer neuen SS	production
Bei der Aneignung einer neuen S	in-sentence	M	S	pre-contextual deletion
Bei der Aneignung einer neuen Sprache fliessen die Sprache und das Umfeldd	in-sentence	M	prache fliessen die Sprache und das Umfeldd	production
Bei der Aneignung einer neuen Sprache fliessen die Sprache und das Umfeld	in-sentence	M	d	pre-contextual deletion
Bei der Aneignung einer neuen Sprache fliessen die Sprache und das Umfeld unvermeindilch ineinander.	in-sentence	E	unvermeindilch ineinander.	production

5.2 Evaluation methods

We evaluate all three algorithms by considering them as classification tasks. For transforming sequences, we use definitions from our framework in sections ***3.1*** and ***3.2.1***. For evaluating the extraction of sentence segments, we consider each sequence of sentence segments a class as shown in table ***1***.

The evaluation of the detection of categories of sentence transforming sequences is based on the classes derived from our theoretical framework as presented in section ***3.3***. Each sentence transforming sequence can belong to one of the following categories: production, pre-contextual deletion, pre-contextual insertion, contextual deletion, and contextual insertion.

For all the three classification tasks, we use the typical metrics for evaluating classification tasks, i.e., accuracy, precision, and recall.

5.3 Evaluation results and discussion

The evaluation corpus used for evaluating the classification of transforming sequences contains 55 cross-sentence transforming sequences, 477 in-sentence transforming sequences, 7 multi-sentence transforming sequences, and 2 uni-sentence transforming sequences.

The SCM component performs very well on this task: the precision, recall, and accuracy scores for all texts are between 0.99 and 1.00. This shows that the algorithm for classifying transforming sequences provide solid results and can be reliably applied for automated processing of writing data.

We evaluate the algorithm for sentence segments extraction based on 480 sentence segments. All manually annotated patterns with the number of occurrences are provided in Table *8*.

Table 8: Results of evaluation of algorithm for extracting sentence segments from transforming sequences. B stands for sentence beginning, M is sentence middle, and E denotes sentence end.

Sentence segment pattern	# occurrences
M	376
B	25
C	2
C-B	2
C-C-C	1
E	15
E-B	53
E-C-B	4
E-C-C	2

The SCM component performs well on this task: for texts 4A1 and 5A1 the accuracy scores are between 0.95 and 0.99. For texts 1A1 and 2A1 the results are slightly worse but still above 0.9; we detected 17 incorrectly classified sentence segments in these two texts. The manual analysis of the errors shows that most of them (11 of 17) result from the software misinterpreting white space at the beginning of the sentence and classifying the sequence as sentence end.

Another issue concerns edits outside of sentences, e.g., when white space between sentences is deleted, which we cannot handle at the moment. In total, the SCM component misclassifies 28 out of 480 sentence segments, which is slightly less accurate than the classification of transforming sequences, but the algorithm is nevertheless sufficiently reliable.

The last evaluation relates to detection of categories of sentence transforming sequence. It is based on 589 sentence transforming sequences: 331 productions, 225 pre-contextual deletions, 16 contextual deletions, and 17 contextual insertions.

The SCM component performs very well on texts 2A1 and 5A1, with accuracy scores of 0.98 and 0.97, respectively. The performance on texts 1A1 and 4A1 is worse, in particular for the latter (accuracy 0.79). The manual investigation of the errors allowed us to discover a weakness of the algorithm in the interpretation of replacement events in the keystroke logs, which leads to most of the misclassifications.

5.4 Summary

Overall, the SCM component demonstrates robust performance in classifying transforming sequences. Accuracy scores ranging between 0.99 and 1.00 indicate high reliability of the algorithm. The extraction of sentence segments from transforming sequences also proves to be very accurate (> 0.9). The accuracy for the detection of categories of sentence transforming sequences is somewhat less accurate, but mostly due to a single issue.

6. Testing the explanatory power of the model

In section *5* we evaluated the algorithms derived from the model and their implementation in the SCM component. In this section, we evaluate the explanatory power and report on an exemplary application to real-world writing process data. As proof of concept, we analyze a small set of keystroke logs with respect to two questions: (1) Can writing be understood as sentence-driven transformations? (2) What are observable steps in the process of translating ideas into sentences? We seek to answer the first question with a model of writing as sentence-driven process and apply the sentence production cycle to provide an answer to the second question.

6.1 Data: THEcorpus

For this evaluation we use 12 sets of keystroke logs from writing sessions with 6 young adults writing 2 blog posts each. The writers are students of a bachelor program and write texts in their first language, German. The writing processes were recorded with ScriptLog (Johansson et al., 2018) and exported in IDFX-format. The data is collected to be used for verifying the concepts and models implemented in THEtool and for evaluating the performance of the software on real-life data. It is available as ***THEcorpus*** (Text History Extraction Corpus) for similar purposes by other researchers or our own future developments in the GitHub repository of THEtool (***https://github.com/mulasik/wta***).

6.2 Analyzing writing as sentence-driven process

If actions executed by writers are actually sentence-driven operations, all transforming sequences detected by the SCM component should be correctly assigned one of the transforming sequence scopes.

We extracted and investigated 14 text transformation histories with a total of 2525 transforming sequences. This yields instances of all possible patterns of all scopes except for a transforming sequence comprising at least two whole SENs followed by a SEC.

Table *9* provides an overview of sentence segment sequences occurring in the extracted transforming sequences.

We can observe that 75% of transforming sequences (1898) occur within the sentence frame, i.e., they impact neither beginning nor end of a sentence (sentence segment pattern M). The rest of the cases can be interpreted by focusing on the start or the end of a transforming sequence, which results in different groupings of the sentence segment patterns as listed in table *9* (we disregard the 15 various other cases that only occur once).

In 153 cases, the *start* of the transforming sequence coincides with the *beginning* of a sentence; of these, 139 impact just the beginning of a sentence (B); 5 result in one or two complete sentences (C and C-C); and 9 (C-B) in a complete sentence and the beginning of the next sentence. In 214 cases, transforming sequences start at the *end* of one sentence and continue to the *beginning* of the next one.

This includes the 194 simple cases of E-B, but also 4 cases where the following sentence is completed (E-C) and 16 cases where the end of the sentence is followed by one or two complete sentences and the transforming sequence continued to the beginning of a next sentence.

Table 9: Sequences of sentence segments impacted by text transformations in the sample data from THEcorpus. B stands for sentence beginning, M is sentence middle, and E denotes sentence end.

Sentence segment pattern	**# occurrences**
M	1898
B	139
C	2
C-B	9
C-C	3
E	89
E-B	194
E-C	4
E-C-B	14
E-C-C-B	2
other patterns (each occurring once)	15

In 98 cases, the *end of the transforming sequence* coincides with the end of a sentence: in addition to the 89 cases that impact just the end of a sentence (E), this includes the cases where a complete sentence is produced after the end of a sentence or just complete sentences (4 Ч E-C, 3 Ч C-C, 2 Ч C).

Due to the small sample size, we cannot draw definitive conclusions yet. The results also underscore the need to also include the burst layer, which might allow us to gain more insights into the sentence production process. For example, it seems likely that most of the mid-sentence transforming sequences are minor adjustments. Since transforming sequences otherwise do tend to align with sentence beginnings and end, this would confirm the intuition that the production of sentences is often interrupted by minor adjustments in the middle of the sentence, e.g., for spelling corrections. We will have a closer look at this hypothesis by applying the notion of the sentence production cycle to all sentence histories.

6.3 Analyzing the sentence production cycle

We examine the sentence production process by investigating sentence histories generated on the basis of the keystroke logs constituting THEcorpus. We look for answers to the following questions: (1) What are observable steps in the process of producing sentences? Can we discover repeating patterns? (2) Can we detect patterns in the process of assembling sentences from sentence segments? (3) Can we observe any typical properties of sentence segments from the particular categories (sentence beginning, middle, and end)?

We apply the sentence production cycle on our sample of 279 sentence histories and 1919 SPSFs (i.e., SECs and SENs) from THEcorpus as extracted previously. 11% of the sentence histories are not finished, i.e., none of the SPSFs produced during the whole sentence production has the properties of a SEN: the writer started producing a sentence but removed it entirely. These histories of unfinished sentence are excluded from our analysis. This reduces the number of sentence histories to 247 and the number of SPSFs to 1851.

Table 10 presents the patterns detected in the production of sentences contained in these 247 sentence histories. 83% of the sentences were completed within the initial draft and never revised. Out of them, 35 sentences were produced directly within one transforming sequence as SENs. The remaining 43 sentences were produced as result of both initial and revision draft. For 32 sentence histories the revision stayed within the sentence frame—i.e., once a SEN was produced, all later revisions were made between the capital letter opening it and the final punctuation mark. The SPSF never lost the property of being a SEN (an example of this phenomenon is shown in table *4*). In contrast, 11 sentence histories included revisions consisting in re-opening the sentence. In all those 11 cases, the SPSF lost the property of a SEN and became a SEC again. See Table *5* for an example of a sentence history where a SEN is transformed back into a SEC.

Table 10: The patterns detected in the sentence production process. "SEN" represents a step consisting in producing or modifying a SEN, and "SEC" denotes a step resulting in producing or modifying a SEC. "+" indicates that the particular step was executed once or more, resulting in the

same type of SPSF (e.g., "SEC+ → SEN → SEN+" describes step sequences such as "SEC → SEC → SEN → SEN → SEN → SEN").

Pattern	**# Sentence histories**
ONLY INITIAL DRAFT	
SEC+ → SEN	169
SEN	35
Total	204
INITIAL AND REVISION DRAFT	
Revision within sentence frame	
SEN → SEN+	2
SEC+ → SEN → SEN+	30
Total	32
Revision with sentence re-opening	
SEC+ → SEN → SEC → SEN → SEC → SEN+	2
SEC+ → SEN → SEC+ → SEN+	6
SEN → SEC+ → SEN	1
SEC+ → SEN → SEN → SEC+ → SEN	1
SEN → SEC → SEN+ → SEC+ → SEN	1
Total	11

For the largest group of sentences presented above—i.e., sentences produced within the initial draft but not as a complete sentence in one operation—we investigated the sentence assembly process. Our goal was to examine how sentences are composed from sentence segments. Table ***11*** presents the results of this analysis. Most sentences whose production was accomplished within the initial draft were produced as a result of producing the sentence beginning and leaving it unchanged, then revising the middle of the sentence multiple times, and finally producing the sentence end. Only for 4 sentences the beginning of the sentence was revised. In most sentence histories (165), the revisions impacted merely the middle of the sentence.

However, when interpreting the results, there are two aspects that must be taken into consideration: (1) if the complete sentence beginning is removed without leaving any character, it is interpreted as start of a new sentence history. Only in case the beginning is modified by leaving subsequent tokens unchanged, the automatic analysis currently considers it modifying the sentence beginning. This limitation distorts the statistics (see section ***4.3*** for a justification for this approach). (2) The initial draft of the sentence by definition cannot contain edits of the sentence end.

Table 11: Most common patterns in the process of assembling sentences from sentence segments on the example of 247 sentence histories from THEcorpus. B stands for sentence beginning, M is sentence middle and E denotes sentence end.

Pattern	# Occurrences	Avg. final sentence length (in words)
Only initial draft		
B M+ E	125	18.5
B M E	39	15.0
F	35	11.2
Other patterns	5	—
Total	204	
Initial and revision draft		
B M+ E M+	14	17.9
B M E M+	6	12.2
B M+ E M	6	22.2
Other patterns	17	—
Total	43	

Deleting a sentence end would lead to re-opening the sentence and thus be a starting point of a revision draft (see part "INITIAL AND REVISION DRAFT" of Table *10*). For this reason, a pattern containing multiple transformations of a sentence end cannot occur within the initial draft.

Finally, we investigated the same group of sentences from yet another perspective: we analyzed the properties of sentence segments from the particular categories (sentence beginning, middle, and end). These statistics are presented in table *12*. Over 70% of edits in the middle of a sentence impact a sequence of one or two tokens, i.e., complete word forms or fragments of word forms (non-finished word forms). Close to 40% comprise only one or two characters. When investigating the data manually, we observed that the latter are mostly corrections of typos or spelling corrections that interrupt the sentence production.

Table 12: Sentence segments statistics with regards to number of occurrences (n) and lengths per segment.

Segment	*n*	Avg. length (in chars)	Avg. length (in words)	<3 chars (in %)	<3 tokens (in %)	>10 tokens (in %)
sen beg	201	39.20	6.52	2.49	20.90	19.90
sen mid	1197	12.05	2.68	39.68	70.26	2.59
sen end	196	34.97	4.90	13.27	38.78	10.20

Table *13* shows statistics with regards to operations performed on the sentence segments. Here, we can also see that the majority of operations performed in the sentence middle are pre-contextual deletions (600 operations). This means most operations consist in deleting a short sequence at the end of the SPSF before continuing the sentence production process. This confirms the observation concluding section *6.2*.

Table 13: Number of sentence operations performed per sentence segment type.

Segment	production	pre-con del	pre-con ins	pre-con rev
sen beg	170	0	30	1
sen mid	455	600	63	7
sen end	191	1	0	4

6.4 Summary

We investigated 14 text histories with a total of 2525 transforming sequences. We observed that the majority of transforming sequences happen partially or fully within sentence boundaries. Looking at the production of individual sentences by applying the sentence production cycle, we investigated 247 sentence histories containing 1851 SPSFs in total. We could observe that the majority of sentence were produced within the initial draft. This detailed analysis of those sentences led to the conclusion that in the majority of cases, the beginning of the sentence stays unchanged. We also discovered that most edits are in the middle of a sentence with a very small scope: between 1 and 2 tokens (complete word forms or fragments of word forms) in 70% of the cases. Manual analysis revealed that most edits are to correct typos and spelling mistakes.

This small study showed some example applications of the model as implemented by THEtool with the SCM component and some conclusions that can be drawn from a small set of real-world keystroke logs.

7. Conclusion and outlook

In this article, we have introduced a sentence-centric model of writing, i.e., we understand writing a text as a ***sentence-driven process***. This framework is based on the notion of ***text history*** and captures different dimensions of writing: producing a text by transforming it from one version to another, producing sentences one by one, pausing, and going back in text to perform revisions. By combining these different perspectives, it reflects how the sentence production process is punctuated by writing mode switches and bursts, and impacted by non-linearity.

Within this framework, we have proposed a model to analyze the production of individual sentences in a ***sentence production cycle***. Based on the corresponding notion of ***sentence history***, this cycle covers the transcription of an initial idea (or parts of it) through completion and potential revision of the sentence and intermediate fragments to its eventual realization as present in the final product, including its complete removal. It thus permits tracking the evolution of a sentence from a first, incomplete version to a complete sentence.

We have presented a working implementation of this approach in the form of an extension to THEtool, an open-source application for extracting text and sentence histories. In order to verify the reliability of the implementation, we have performed an evaluation of the three algorithms that constitute the foundation for our modeling. The evaluation was conducted on real-life data and demonstrated that the results are solid and can be applied for automated processing of writing data. We have also performed a small case study and shown some illustrative applications of the model along with some possible conclusions that can be drawn by analyzing real-life data with our software.

The theoretical framework and its current implementation presented in this article are first attempts at sentence-centric modeling of writing. The models are multidimensional and contain a large amount of data related to writers' behavior and the evolution of the text and its sentences on the surface. Currently, the models contain more information than can be actually processed and interpreted by THEtool. We intend to continue integrating further functionalities into the application to extract more information from the available data and investigate relations between different phenomena. Our objective is to create opportunities for both a more profound and a broader or more diversified sentence-focused analysis of the writing process.

Certain extensions to the models seem particularly interesting and relevant in this regard. The first is a deeper investigation of the mapping of the three layers (transformation layer, sentence layer, burst layer), which would allow us to link pauses and bursts to syntactic structures affected by transforming sequences; this would help us to identify the relations between syntactic structures and the cognitive effort needed to produce and revise them.

A further goal is to find visualizations to help with the interpretation of the complex relations, especially the mapping of the layers. Currently, the SCM component generates output files containing the data in different formats suitable for further processing, but it does not yet provide any visualizations.

With regards to model enhancements, two considerations seem particularly important. Firstly, we would like to integrate a module for the automatic detection of revisions as proposed by Conijn et al. (2021). Secondly, we intend to incorporate a detection mechanism for typos and spelling corrections. This would allow us to exclude transformations not relevant for content generation and revision.

The proposed sentence-centric modeling of writing not only opens a new perspective on text production but also, through its implementation, enables automated large-scale analysis of writing process data with the focus on sentences. To support future research in this area, we make the SCM module publicly available in the GitHub repository (*https://github.com/mulasik/wta*) as a component of the open-source application THEtool and invite collaboration on its further development.

Author note

The corpus used for evaluating the models as well as for performing the case study (THEcorpus) is available in the public repository of THEtool on github: *https://github.com/mulasik/wta*. We thank Brigitte Gerber and Nalini Bussinger for help with evaluating the implementation and the explanatory power of the model. The work of Malgorzata Anna Ulasik has been supported by ZHAW DIZH Fellowship Call 2021. We thank the anonymous reviewers for their helpful comments and questions.

References

Alves, R. A., Castro, S. L., Sousa, L. de, & Strumqvist, S. (2007). Influence of typing skill on pause–execution cycles in written composition. In M. Torrance, L. van Waes, & D. Galbraith (Eds.), *Writing and cognition* (pp. 55–65). Brill. *https://doi.org/10.1163/9781849508223_005*

Baaijen, V. M., & Galbraith, D. (2018). Discovery through writing: Relationships with writing processes and text quality. *Cognition and Instruction*, *36*(3), 199–223. *https://doi.org/10.1080/07370008.2018.1456431*

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*

Bolter, J. D. (1989). Beyond word processing: The computer as a new writing space. *Language & Communication*, *9*(2–3), 129–142. *https://doi.org/10.1016/0271-5309(89)90014-1*

Bьhler, K. (1918). Kritische Musterung der neuen Theorien des Satzes [Critical examination of the new theories of the sentence]. *Indogermanisches Jahrbuch*, *6*, 1–20. *https://doi.org/10.1515/if-1927-0121*

Buschenhenke, F., Conijn, R., & Van Waes, L. (2023). Measuring non-linearity of multi-session writing processes. *Reading and Writing*, 511–537. *https://doi.org/10.1007/s11145-023-10449-9*

Chenoweth, N. A., & Hayes, J. R. (2001). Fluency in writing: Generating text in L1 and L2. *Written Communication*, *18*(1), 80–98. *https://doi.org/10.1177/0741088301018001004*

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

Collier, R. M. (1983). The Word Processor and Revision Strategies. *College Composition and Communication*, *34*(2), 149–155. *https://doi.org/10.2307/357402*

Conijn, R., Dux Speltz, E., & Chukharev-Hudilainen, E. (2021). Automated extraction of revision events from keystroke data. *Reading and Writing*, *37*(2), 483–508. *https://doi.org/10.1007/s11145-021-10222-w*

Crossley S. A. (2020). Linguistic features in writing quality and development: An overview. *Journal of Writing Research*, *11*(3), 415-443. *https://doi.org/10.17239/jowr-2020.11.03.01*

Daiute, C. A., & Taylor, R. (1981). Computers and the improvement of writing. *Proceedings of the ACM '81 Conference*, 83–88. *https://doi.org/10.1145/800175.809841*

Dux Speltz, E., & Chukharev-Hudilainen, E. (2021). The effect of automated fluency-focused feedback on text production. *Journal of Writing Research*, *13*(2), 231-255. *https://doi.org/10.17239/jowr-2021.13.02.02*

Faigley, L., & Witte, S. (1981). Analyzing revision. *College Composition and Communication*, *32*(4), 400–414. *https://doi.org/10.2307/356602*

Feltgen, Q., Cislaru, G., & Benzitoun, C. (2022). Йtude linguistique et statistique des unitйs de performance йcrite: le cas de *et* [Linguistic and statistical study of written performance units:

the case of *et*]. *8^e Congrès mondial de linguistique française*, *SHS Web of Conferences 138*, 10001. *https://doi.org/10.1051/shsconf/202213810001*

Feltgen, Q., Lefeuvre, F., & Legallois, D. (2023). Sujet clitique et dynamique de l'йcrit: un йclairage par les jets textuels [The clitic subject and the dynamics of writing: a look at textual bursts]. *Discours. Revue de linguistique, psycholinguistique et informatique*, *32.* https://doi.org/10.4000/discours.12509

Feltgen, Q, & Lefeuvre, F. (2025). Clitic subjects as landmarks in the writing production process: A study based on a keylog-derived corpus of writing bursts. *Journal of Writing Research*, *16*(3), 435-462. https://doi.org/10.17239/jowr-2025.16.03.04

Fitzgerald, J. (1987). Research on Revision in Writing. *Review of Educational Research*, *57*(4), 481–506. *https://doi.org/10.2307/1170433*

Foulin, J.-N. (1995). Pauses et dйbits: les indicateurs temporels de la production йcrite [Pauses and flows: the temporal indicators of written production]. *L'année psychologique*, *95*(3), 483–504. *https://doi.org/10.3406/psy.1995.28844*

Gardiner, A. H. (1922). The definition of the word and the sentence. *British Journal of Psychology: General Section*, *12*(4), 352–361. *https://doi.org/10.1111/j.2044-8295.1922. tb00067.x*

Gilquin, G. (2020). In search of constructions in writing process data. *Belgian Journal of Linguistics*, *34*(1), 99–109. *https://doi.org/10.1075/bjl.00038.gil*

Haas, C. (1989). How the writing medium shapes the writing process: Effects of word processing on planning. *Research in the Teaching of English*, *23*(2), 181–207. *http://www.jstor.org/stable/40171409 https://doi.org/10.58680/rte198915523*

Hayes, J. R. (2009). From idea to text. In R. Beard, D. Myhill, J. Riley, & M. Nystrand (Eds.), *The SAGE handbook of writing development* (pp. 65–79). SAGE. *https://doi.org/10.4135/9780857021069.n5*

Hayes, J. R. (2012). Modeling and remodeling writing. *Written Communication*, *29*(3), 369–388. *https://doi.org/10.1177/0741088312451260*

Immonen, S., & MΔkisalo, J. (2017). Pauses reflecting the processing of syntactic units in monolingual text production and translation. *HERMES – Journal of Language and Communication in Business*, *23*(44), 45–61. *https://doi.org/10.7146/hjlcb.v23i44.97266*

Ivaska, I., Toropainen, O., & Lahtinen, S. (2025). Pauses during a writing process in two typologically different languages. *Journal of Writing Research*, *16*(3), 407-433. https://doi.org/10.17239/jowr-2025.16.03.03

Johansson, V., Frid, J., & Wengelin, E. (2018). ScriptLog – an experimental keystroke logging tool. In R. A. Alves & A. Camacho (Eds.), *Proceedings of the 1st literacy summit* (p. 51).

Kaufer, D. S., Hayes, J. R., & Flower, L. (1986). Composing written sentences. *Research in the Teaching of English*, *20*(2), 121–140. *https://www.jstor.org/stable/40171073 https://doi.org/10.58680/rte198615612*

Kollberg, P. (1998). *S-notation – a Computer Based Method for Studying and Representing Text Composition* [Master's thesis]. Kungliga Tekniska Hugskolan.

Kollberg, P., & Severinson Eklundh, K. (2002). Studying writers' revising patterns with S-notation analysis. In T. Olive & C. M. Levy (Eds.), *Contemporary tools and techniques for studying writing* (Vol. 10, pp. 89–104). Kluwer. *https://doi.org/10.1007/978-94-010-0468-8_5*

Leijten, M., Macken, L., Hoste, V., Van Horenbeeck, E., & Van Waes, L. (2012). From character to word level: Enabling the linguistic analyses of Inputlog process data. In M. Piotrowski, C. Mahlow, & R. Dale (Eds.), *Proceedings of the second workshop on computational linguistics and writing (CL&w 2012): Linguistic and cognitive aspects of document creation and document engineering* (pp. 1–8). ACL. *https://aclanthology.org/W12-0301/*

Leijten, M., Van Horenbeeck, E., & Van Waes, L. (2019). Analysing keystroke logging data from a linguistic perspective. In E. Lindgren & K. Sullivan (Eds.), *Observing writing* (pp. 71–95). Brill. *https://doi.org/10.1163/9789004392526_005*

Leijten, M., Van Waes, L., & Van Horenbeeck, E. (2015). Analyzing writing process data: A linguistic perspective. In *Writing(s) at the crossroads: The process-product interface* (pp. 277–302). John Benjamins. *https://doi.org/10.1075/z.194.14lei*

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

Lutz, J. A. (1983). *A study of professional and experienced writers revising and editing at the computer and with pen and paper* [PhD thesis]. Rensselaer Polytechnic Institute.

Mahlow, C. (2015). A definition of "version" for text production data and natural language document drafts. In G. Barabucci, U. M. Borghoff, A. Di Iorio, S. Maier, & E. Munson (Eds.), *DChanges 2015: Proceedings of the 3rd international workshop on (document) changes: Modeling, detection, storage and visualization* (pp. 27–32). ACM. *https://doi.org/10.1145/2881631.2881638*

Mahlow, C., Ulasik, M. A., & Tuggener, D. (2024). Extraction of transforming sequences and sentence histories from writing process data: A first step towards linguistic modeling of writing. *Reading and Writing*, *37*, 443–482. *https://doi.org/10.1007/s11145-021-10234-6*

Mahrer, R., & Zuccarino, G. (2025). Units of linguistic analysis in written production: From the case of enunciative interruptions. *Journal of Writing Research*, *16(3)*, 521-535. https://doi.org/10.17239/jowr-2025.16.03.07

Matsuhashi, A. (1981). Pausing and planning: The tempo of written discourse production. *Research in the Teaching of English*, *15*(2), 113–134. *https://doi.org/10.58680/rte198115773*

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

Miletic, A., Benzitoun, C., Cislaru, G., & Herrera-Yanez, S. (2022). Pro-TEXT: An annotated corpus of keystroke logs. In N. Calzolari, F. Bйchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (pp. 1732–1739). European Language Resources Association. *https://aclanthology.org/2022.lrec-1.184*

Noreen, A. (1903). *Vårt språk: nysvensk grammatik i utförlig framställning [Our language: the new Swedish grammar presented in detail]* (Vol. 1). Gleerup.

Olive, T. (2012). Writing and working memory: A summary of theories and of findings. In E. L. Grigorenko, E. Mambrino, & D. D. Preiss (Eds.), *Writing: A mosaic of new perspectives* (pp. 120–136). Psychology Press. *https://doi.org/10.4324/9780203808481*

Olive, T., & Cislaru, G. (2015). Linguistic forms at the process-product interface: Analysing the linguistic content of bursts of production. In G. Cislaru (Ed.), *Writing(s) at the crossroads* (pp. 99–124). John Benjamins. *https://doi.org/10.1075/z.194.06oli*

Piolat, A. (1991). Effects of word processing on text revision. *Language and Education*, *5*(4), 255–272. *http://cogprints.org/3621/ https://doi.org/10.1080/09500789109541314*

Serbina, T., Hintzen, S., Niemietz, P., & Neumann, S. (2017). Changes of word class during translation – insights from a combined analysis of corpus, keystroke logging and eye-tracking data. In S. Hansen-Schirra, O. Czulo, & S. Hofmann (Eds.), *Empirical modelling of translation and interpreting* (pp. 177–208). Language Science Press. *https://doi.org/10.5281/zenodo.1090968*

Severinson Eklundh, K. (1994). Linear and nonlinear strategies in computer-based writing. *Computers and Composition*, *11*(3), 203–216. *https://doi.org/10.1016/8755-4615(94)90013-2*

Ulasik, M. A., & Miletić, A. (2024). Automated extraction and analysis of sentences under production: A theoretical framework and its evaluation. *Languages*, *9*(3), 71. *https://doi.org/10.3390/languages9030071*

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/*https://doi.org/10.1016/j.compcom.2015.09.012*

Weizenbaum, J. (1976). *Computer power and human reason: From judgment to calculation*. W. H. Freeman & Co.