Actes de CORIA-TALN-RJCRI-RECITAL 2025
Marseille, France, 30 juin au 4 juillet 2025

In Pursuit of Sentences: Methods for Processing Dynamic Data to Trace Sentence Production

Malgorzata Anna Ulasik[1, 2] Cerstin Mahlow[2]
(1) Université de Lausanne, Section des sciences du langage et de l'information, Lausanne, Switzerland
(2) Zurich University of Applied Sciences, School of Applied Linguistics, Winterthur, Switzerland
ulas@zhaw.ch, cerstin.mahlow@zhaw.ch

RÉSUMÉ

Nous présentons des méthodes de traitement des données dynamiques permettant de retracer le processus de production de phrases. En tant qu'activité incrémentielle et non linéaire, l'écriture produit des versions intermédiaires incomplètes ou mal formées qui évoluent au fil de fréquentes révisions. À l'aide d'outils d'enregistrement des frappes et de traitement du langage naturel (TALN), nous proposons un cadre permettant de reconstruire automatiquement l'historique des phrases. De plus, nous implémentons dans THEtool un modèle qui synchronise l'historique des phrases avec les événements de révision et les *patterns* de pause. Cette représentation multicouche facilite la compréhension détaillée des aspects cognitifs et linguistiques de la construction des phrases.

ABSTRACT

In Pursuit of Sentences: Methods for Processing Dynamic Data to Trace Sentence Production

We present methods for processing dynamic data to trace the sentence production process. As an incremental, nonlinear activity, writing produces incomplete or ill-formed intermediate text versions that evolve through frequent revisions. Using keystroke logging and natural language processing (NLP) tools, we propose a framework for automatically reconstructing sentence histories. Additionally, we implement a layer-based model in THEtool that synchronizes sentence histories with revision events and pause patterns. This multilayered representation facilitates detailed insights into the cognitive and linguistic aspects of sentence construction.

MOTS-CLÉS : Historique des phrases, production de phrases, traitement du langage naturel.

KEYWORDS: Sentence History, Sentence Production, Natural Language Processing.

ARTICLE : **Accepté à** DYN-TAL Traitement de données langagières dynamiques par les outils et méthodes du TAL.

1 Introduction

Keystroke logging is an established method to record writing processes. For each action performed by a writer, the relevant information on this event is stored: which key has been pressed and released at which point in time, its effect on the text, and the position of the cursor in the document. Such process data is incremental, since events are executed in chronological order. However, it is not linear: writers can move back and forth in the text. Two chronologically sequential events can take place at non-adjacent positions in the text. Thus, writing is the incremental, nonlinear production of sentences.

Actes de l'atelier Traitement de données langagières dynamiques
par les outils et méthodes du TAL 2025 (DYN-TAL), pages 1–6.

The text under production is often linguistically incomplete or ill-formed. Additionally, sentence production is unpredictable: A syntactically complete sentence can be transformed back to an incomplete sentence or to a single letter at any point in time. It can be merged with an adjacent sentence or split into phrases. It can remain unfinished until the completion of the text, or it can be entirely deleted. Some sentences might be produced without any revisions. To gain a comprehensive picture of the sentence production process, we also need to consider the time aspect. For a particular sentence, the creation may be interrupted by multiple pauses. Only when keeping track of both the revisions and the pausing behavior, we can achieve a comprehensive understanding of the flow of sentence production.

In this paper, we focus on methods to address challenges emerging from fully automatically handling writing process data characterized by incrementality, nonlinearity, contingency, incompleteness, and multidimensionality by leveraging existing natural language processing (NLP) tools. By developing new or refining existing concepts and definitions, and by creating specialized algorithms, we aim to provide insights into the intricate and multilayered nature of the sentence production process.

2 Related work

Understanding the dynamic nature of language production, particularly in writing, has gained increasing attention. The inherent dynamics of incremental events unfolding nonlinearly presents challenges for traditional linguistic models, which usually focus on the static final product.

The analysis of keystroke logs has focused mostly on pauses as potential indicators of the underlying cognitive processes (Alves *et al.*, 2007; Olive, 2012; Immonen & Mäkisalo, 2017). Pauses are often used to segment the writing stream into production units, so-called bursts (Chenoweth & Hayes, 2001). Investigations of the syntactic structure of these bursts suggest that they often do not align neatly with traditional syntactic units (Gilquin, 2020; Feltgen *et al.*, 2023) and are often incomplete (Cislaru & Olive, 2018).

However, keystroke logging data also enable the reconstruction of a text's evolution through intermediate versions determined by changes in production mode (Mahlow, 2015). More recent approaches focus on modeling the writing process by tracking changes across intermediate text versions (Miletic *et al.*, 2022). The concept of *text history* captures the sequence of all intermediate texts produced so far (TPSF). Analyzing the transformations between these versions (*transforming sequences*) allows a detailed look at editing operations, as proposed by (Mahlow *et al.*, 2024; Ulasik & Miletić, 2024; Ulasik *et al.*, 2025). The Text History Extraction tool (THEtool) (Ulasik, 2022) was developed to automate the extraction and analysis of text and sentence histories from keystroke logs, facilitating larger-scale studies into the dynamic, sentence-driven nature of writing.

3 Handling Dynamic, Multidimensional Writing Process Data

3.1 Managing Incompleteness and Contingency with Text Unit Categorization

When writers start producing a sequence of characters, we do not know if it will result in a complete sentence, if it is just a new beginning for the following sentence, or if it will be removed with the next keystroke. Sentence boundaries are changing due to merges and splits. An incomplete sentence may remain incomplete for a long time and then finally get removed. Due to this incompleteness and ill-formedness, segmenting intermediate text versions into sentences is a challenging task. For

sentence boundaries detection, we apply the statistical DependencyParser from spaCy, an open-source Python library for advanced NLP (Montani *et al.*, 2023). According to the spaCy documentation, among the tools offered by spaCy 10, the dependency parser provides the most accurate sentence boundaries, which is why we have selected it for our implementation. Our tests have shown that it performs indeed very well on complete sentences, but it does not always provide correct results for incomplete and ill-formed sentences (Mahlow *et al.*, 2024; Ulasik & Miletić, 2024).

For tracking the content of each text version focusing on sentences we propose the concept of a *text unit*. The TPSF at any given point in time can be split into text units in such a way that each character produced, including whitespace, belongs to exactly one text unit. We distinguish two main types of text units: *SPSF* (sentence produced so far, in analogy to TPSF) (Ulasik & Miletić, 2024)—which holds the textual content—and *interspace*—which is used to separate SPSFs. Interspaces contain spaces, newlines, and indentation characters.

Keystroke logs do not provide information on writers' mental representation of the sentence. The only information available is the behavioral data—the keys the writer presses—and the text itself. In order to distinguish full-fledged sentences from sequences of characters that do not (yet) meet the sentencehood criteria (see section 3.2), we distinguish two types of SPSFs: *sentences* (SEN) and *sentence candidates* (SEC). For a simplified operationalization, we use the writer's behavior: the start of a sentence is indicated by capitalizing its first letter and its end is indicated by entering a final punctuation mark. This definition provides satisfactory results for automatically handling sentences incompleteness as shown in the evaluation in Ulasik & Miletić (2024).

A SEN is thus a sequence of characters that starts with a capital letter and ends with sentence-final punctuation. Once a sequence of characters has been identified as SEN, its status remains unchanged as long as the writer does not clearly signal a revision of the sentence scope by removing the capitalization of the initial letter, adjusting the final punctuation mark, or both.

A SEC is a sequence of characters that does *not* start with a capital letter and/or does *not* end with sentence-final punctuation. It is a container for content outside of SENs—whose further evolvement we cannot anticipate.

3.2 Approximating the Degree of Sentencehood

To capture and describe the evolution of sentences, we mark the status of a given SPSF in the formal, semantic, and pragmatic dimensions following the notions of completeness and correctness by Matthews (1993). We consider *completeness* (mechanical, syntactical, and conceptual) and *correctness* (mechanical and grammatical) as sentencehood criteria.

Mechanical completeness relies on our simplified definition of a sentence: it starts with a capital letter and ends with a final punctuation mark. An SPSF is classified as conceptually complete if it is mechanically complete and stays unaffected by the subsequent revision. To check for syntactic completeness, we apply an external Python library: each SPSF is parsed with the spaCy dependency parser (Montani *et al.*, 2023).

For checking correctness, we use the open-source proofreading tool LanguageTool (Naber, 2003). LanguageTool is based on human-curated rules for multiple languages. A mechanically correct SPSF is free from spelling, punctuation, or capitalization errors. If LanguageTool does not detect any grammatical errors, the SPSF is marked as a grammatically correct SEN.

The sentencehood degree reflects sentence incompleteness and ill-formedness. It can be used for

further analysis of sentence production and for understanding issues related to processing SPSFs with NLP tools—the lower the sentencehood degree, the less reliable the output of automated processing.

3.3 Recording Incrementality and Nonlinearity with Sentence Histories

Tracking sentences involves tracing and reconstructing all versions of all sentences via their sentence histories (Mahlow *et al.*, 2024). The sentence histories are derived from the intermediate text versions. THEtool iterates over the text versions segmented into SENs and SECs and retrieves new, modified, deleted, and unchanged SPSFs. Each new SPSF discovered receives a unique ID. Modified, deleted, and unchanged SPSFs are appended to the existing sentence lists. For modified SPSFs, the previous version is used for matching against the SPSF already stored in the sentence history.

All versions of a particular sentence are provided in chronological order. The versions do not necessarily belong to successive text versions, as a writer might have come back to this sentence several times after writing or modifying other sentences in between. The sentence history keeps track of incrementality of nonlinear sentence production.

3.4 A Layer-Based Model for Systematizing Multidimensionality

Sentence production represents one dimension of writing: the transformation of ideas into sentences. Concurrently, the writer revises the previously produced content of sentences and makes pauses while creating it. This behavior signals underlying cognitive activities such as planning or revising of the sentences; considering it is essential for a comprehensive analysis of the sentence production process. To systematize this multidimensionality, THEtool implements a writing model comprising three layers: sentence layer, transformation layer, and burst layer.

The sentence layer contains the sentence histories as outlined above. The transformation layer reflects all revisions undertaken throughout the writing process. The burst layer provides information about duration and content of bursts, as well as the length of pauses.

To gain a holistic understanding of the sentence production process, we project the transformation and burst layers on the sentence layer. The three layers share the same timeline and take place within the space of the text.

4 Conclusion and Outlook

We have presented a comprehensive framework for analyzing keystroke logging data with a focus on sentence production. The concepts of sentence candidates, degree of sentencehood and text units introduced in Ulasik & Miletić (2024) and further developed in our recent work address incrementality, non-linearity, incompleteness and multidimensionality. Our approach implemented in THEtool, using existing NLP tools such as spaCy and LanguageTool, enables automatic extraction and tracking of sentence histories from the writing process data and has been shown to provide reliable results (Ulasik & Miletić, 2024). The introduction of a layer-based model supports a nuanced understanding of fundamental aspects of writing, and has been a starting point for the ongoing further development of THEtool.

Future work will focus on further refining sentencehood criteria, enhancing the robustness of our approach, and extending THEtool with syntactic parsing of SPSFs. Expanding the framework with further projections will allow us among others to investigate the syntactic structures within bursts.

References

ALVES R. A., CASTRO S. L., DE SOUSA L. & STRÖMQVIST S. (2007). Influence of typing skill on pause–execution cycles in written composition. In M. TORRANCE, L. VAN WAES & D. GALBRAITH, Éds., *Writing and Cognition*, p. 55–65. Brill. DOI : 10.1163/9781849508223_005.

CHENOWETH N. A. & HAYES J. R. (2001). Fluency in writing: Generating text in L1 and L2. *Written communication*, **18**(1), 80–98. DOI : 10.1177/074108830101800100.

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 Supérieur. DOI : 10.3917/dbu.cisla.2018.01.

FELTGEN Q., LEFEUVRE F. & LEGALLOIS D. (2023). Sujet clitique et dynamique de l'écrit: un éclairage par les jets textuels [the clitic subject and the dynamics of writing: a look at textual bursts]. *Discours. Revue de linguistique, psycholinguistique et informatique. A journal of linguistics, psycholinguistics and computational linguistics*, **32**, [no pagination]. DOI : 10.4000/discours.12509.

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

IMMONEN S. & MÄKISALO 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. DOI : 10.7146/hjlcb.v23i44.97266.

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, Éds., *DChanges 2015: Proceedings of the 3rd International Workshop on (Document) Changes: modeling, detection, storage and visualization*, p. 27–32: ACM. DOI : 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. DOI : 10.1007/s11145-021-10234-6.

MATTHEWS P. (1993). Central Concepts of Syntax. In J. JACOBS, A. VON STECHOW, W. STERNEFELD & T. VENNEMANN, Éds., *Syntax*, p. 89–117. Walter de Gruyter. DOI : 10.1515/9783110095869.1.1.89.

MILETIC A., BENZITOUN C., CISLARU G. & HERRERA-YANEZ S. (2022). Pro-TEXT: an annotated corpus of keystroke logs. In N. CALZOLARI, F. BÉCHET, P. BLACHE, K. CHOUKRI, C. CIERI, T. DECLERCK, S. GOGGI, H. ISAHARA, B. MAEGAARD, J. MARIANI, H. MAZO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, p. 1732–1739: European Language Resources Association.

MONTANI I., HONNIBAL M., BOYD A., VAN LANDEGHEM S. & PETERS H. (2023). explosion/spacy: v3. 7.2: Fixes for apis and requirements. *Zenodo https://doi. org/10.5281/zenodo*, **10009823**.

NABER D. (2003). LanguageTool: A rule-based style and grammar checker. Mémoire de master, University of Bielefeld.

OLIVE T. (2012). Writing and working memory: A summary of theories and of findings. In E. L. GRIGORENKO, E. MAMBRINO & D. D. PREISS, Éds., *Writing: A Mosaic of New Perspectives*, p. 120–136. Psychology Press. DOI : 10.4324/9780203808481.

ULASIK M. A. (2022). THEtool (software). DOI : 10.5281/zenodo.7941668.

ULASIK M. A., MAHLOW C. & PIOTROWSKI M. (2025). Sentence-centric modeling of the writing process. *Journal of Writing Research*, **16**(3), 463–498. DOI : 10.17239/jowr-2025.16.03.05.

ULASIK M. A. & MILETIĆ A. (2024). Automated extraction and analysis of sentences under production: A theoretical framework and its evaluation. *Languages*, **9**(3), 71. DOI : 10.3390/languages9030071.