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

Bursted! Un outil d'agrégation des keystrokes

Caroline Bordes[1] Thierry Olive[1] Georgeta Cislaru[2]
(1) CeRCA, Université de Poitiers & CNRS, 5 rue T. Lefebvre, 86000 Poitiers, France
(2) CLESTHIA, Sorbonne Nouvelle, Université Sorbonne Nouvelle, Maison de la recherche, 4, rue des Irlandais, 75005 PARIS

caroline.bordes@univ-poitiers.fr, thierry.olive@univ-poitiers.fr, georgeta.cislaru@sorbonne-nouvelle.fr

RESUME

Bursted! est un outil qui permet d'analyser les jets textuels, c'est-à-dire dire les segments de textes produits sans interruption lors d'une ou plusieurs sessions d'écriture. Il analyse les fichiers d'enregistrement des frappes au clavier (*keylogging*) fournis par les logiciels comme Inputlog. Ce travail s'inscrit dans le cadre théorique proposé par Cislaru et Olive (2018) pour étudier le processus de textualisation. L'application Bursted! automatise l'extraction des jets textuels et des variables associées et fournit un fichier au format '.csv' prêt pour des traitements ultérieurs.

ABSTRACT

Bursted! is a tool that allows analyzing bursts of writing, i.e. the text segments produced without interruption during one or several writing sessions from writing keylogging files provided by software such as Inputlog or Scriptlog. This work is part of the theoretical framework proposed by Cislaru and Olive (2018) to study the process of textualization. Bursted! automates the extraction of bursts of writing and associated variables, and provides a '.csv' file ready for further processing.

MOTS-CLES : enregistrement de l'écriture au clavier, jets textuels, pauses d'écriture.
KEYWORDS: keystroke logging, bursts of writing, writing pauses.

ARTICLE : Traitement de données langagières dynamiques par les outils et méthodes du TAL

1 Introduction

Nous présentons une application qui, à partir de données d'enregistrement des frappes au clavier (*keylogging* ou *keystroke logging*) lors de l'écriture, automatise l'extraction des jets textuels et des variables associées pour des traitements ultérieurs. L'enregistrement des frappes au clavier est une technique répandue pour étudier l'écriture sur ordinateur et sa dynamique (Lindgren & Sullivan, 2019). Les logiciels de *keylogging* enregistrent l'ensemble des frappes sur le clavier et les mouvements de souris en même temps que leur chronologie. Ces logiciels proposent également des préanalyses des données. Ils permettent aussi d'analyser la dynamique de l'écriture. Par

Actes de l'atelier Traitement de données langagières dynamiques
par les outils et méthodes du TAL 2025 (DYN-TAL), pages 12–16.
Cette œuvre est mise à disposition sous licence Attribution 4.0 International.

exemple, *Inputlog* (Leijten & van Waes, 2013) enregistre toutes les frappes au clavier et les actions de la souris, avec leur horodatage, pour fournir des résumés statistiques sur les pauses et les opérations de révisions (voir aussi *Scriptlog*, Wengelin et al., 2009).

Théoriquement, ce travail s'inscrit dans le cadre théorique proposé par Cislaru et Olive (2018) pour étudier le processus de textualisation, c'est-à-dire la construction progressive d'un texte. Cislaru et Olive (2018) proposent d'étudier les jets textuels, c'est-dire les segments de textes qui sont produits sans être interrompus par des pauses, que ces segments aient été gardés ou non dans le texte final. De cette façon, la construction progressive du texte final peut-être retracée. Cislaru et Olive ont souligné l'importance d'étudier la textualisation en couplant l'analyse linguistiques des jets textuels à l'analyse des variables temporelles.

Problématique

Bursted! a été développé dans la cadre du projet PRO-Text pour analyser les jets textuels d'un corpus qui contient plusieurs centaines d'enregistrements de textes rédigés en une ou plusieurs sessions. Parmi les analyses proposées par défaut par les logiciels de *keylogging,* aucune ne permet d'analyser les jets textuels. Il a donc été nécessaire de développer un outil automatisant l'extraction de ces jets et des variables associées. Bursted! a été développé en python et est une application autonome qui ne nécessite aucune autre installation ni connexion internet pour l'exécuter.

Présentation de Bursted!

L'outil logiciel Bursted! automatise les traitements préliminaires des fichiers enregistrés par les logiciels de keylogging Inputlog ou Scriptlog. Précisément, il regroupe les différents événements enregistrées (frappes, mouvements de souris…) en jets d'écriture et mesure des variables associées à ces jets. Pour cela, deux étapes sont nécessaires : la première prépare et « nettoie » les keylogs et la seconde agrège les événements conservés en jets d'écriture. Chaque étape donne lieu à un fichier de sortie.

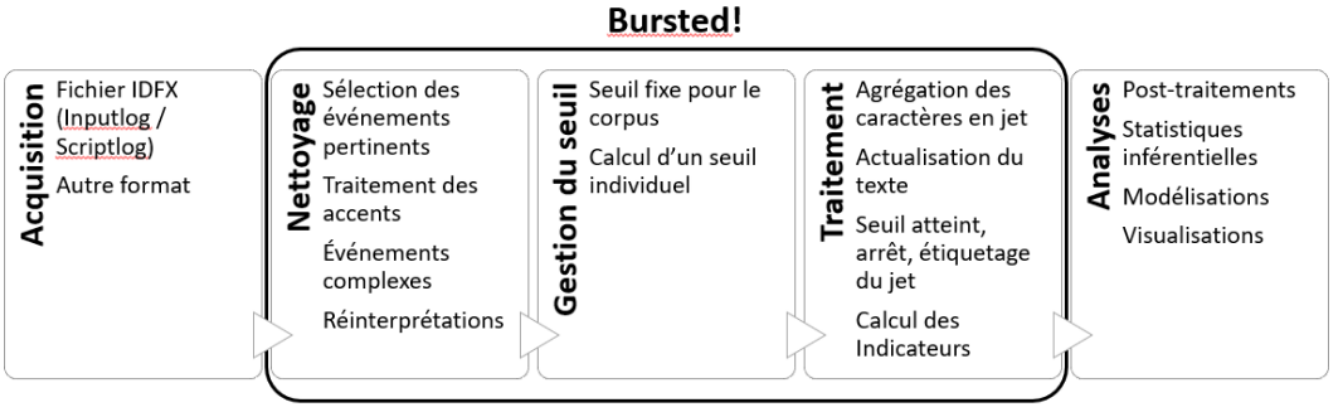

FIGURE 1: Étapes de traitement de Bursted!

1.1 Nettoyage des données

Le nettoyage se base sur des fichiers de type '.idfx', format natif d'Inputlog. En effet, la provenance du fichier (e.g, Inputlog ou Scriptlog) peut impacter la façon dont les événements

d'écriture sont enregistrés. Le logiciel détecte donc l'origine du fichier et traite automatiquement les différences entre les deux types de données.

Chaque événement est traité séquentiellement dans l'ordre chronologique. Dans un premier temps, Bursted! évalue si l'événement est complet et renseigne une activité relative à l'écriture du texte. Sont donc retirés les actions à la souris qui ne peuvent être reliés à des zones précise du texte, et tout événement se produisant hors de la feuille de texte. Les événements de frappes au clavier ne sont conservés que s'ils ajoutent ou retirent un caractère au texte. Les accents détachés (e.g. accent circonflexe ; accent grave pour les claviers QWERTY) sont par exemple considérés comme des événements d'écriture même s'ils n'apparaissent qu'une fois le caractère accentué inséré, tandis que les recours aux touches directionnelles ne modifient pas le texte, et ne constituent pas une action d'écriture. Pour chaque événement gardé, Bursted! consigne également les temps de début et de fin associés, la position dans le texte et la longueur du texte initial.

L'enregistrement de fichiers ifdx repose sur des codes claviers standard. De ce fait, Bursted! peut traiter des données provenant d'un grand nombre de claviers alphanumériques (qwerty, azerty…). Certaines langues présentent toutefois des spécificités avec par exemple des caractères qui ne peuvent être saisis en une frappe unique. C'est le cas par exemple des accents circonflexes en français. Ce type d'événement n'est pas toujours correctement enregistré, notamment lorsqu'il n'est pas rattaché à une voyelle ou qu'il se retrouve supprimé. Cela entraîne des décalages dans le texte et peut donner lieu à de faux inserts de caractères pour rattraper ce retard. Bursted! est donc doté de fonctions qui permettent le retraitement de ces événements et leur correction.

Bursted! prend en compte un grand nombre de situations complexes à traiter pour se rapprocher au plus près du texte final. On peut énumérer 3 types d'événements complexes : la modification de plusieurs caractères à la fois, les accents en français par exemple, et les événements erronés. Cette vérification est accompagnée d'une réécriture sous forme de table, enregistré sous le nom de clean_events.csv. C'est ce format qui sert de base à la seconde partie du traitement.

1.2 Création des jets d'écriture

Un enjeu majeur pour étudier les jets textuels processus est le seuil de pause qui est choisi pour segmenter l'activité d'écriture : des seuils différents produisent en effet des jets textuels spécifiques à ces seuils. Presque toutes les études qui ont analysé les périodes d'exécution et les jets textuels ont utilisé un seuil de pause de 2 secondes. Cela est problématique car les rédacteurs ayant des compétences et des âges différents, ou qui utilisent des outils d'écriture différents n'abordent pas le processus d'écriture de la même manière Chenu et al., 2014). Bursted! permet de déterminer un seuil unique pour tous les rédacteurs ou un seuil individualisé à partir des enregistrements collectés.

Le paramétrage de ce seuil est obligatoire au lancement du logiciel. Le seuil peut être soit unique pour tous les fichiers d'un même corpus, soit individualisée. Pour ce dernier cas, la procédure adoptée est celle de Bouriga (2019 ; voir aussi Bouriga, Olive, Bordes & Chesnet, soumis) qui garde cependant une référence au seuil de 2 secondes. Précisément, le seuil s d'un rédacteur sera, parmi l'ensemble de ses n durées de pauses inter-frappes, la valeur du centile dans lequel se situe, sur l'ensemble N des pauses inter-frappes de tout le corpus, la valeur seuil origine S. Si un rédacteur produit plusieurs textes ou écrit un même texte sur plusieurs sessions, le seuil sera établi sur la distribution de l'ensemble des pauses de toutes ses sessions d'écriture. Chaque fichier idfx

contenant des centaines de pauses, une seule session suffit pour calculer un seuil de pause individuel.

Extractions des jets textuels

A partir des fichiers *clean_events.csv*, Bursted! reconstitue l'écriture du texte événement par événement. Dès qu'une pause entre deux événements est supérieure au seuil, le jet textuel est enregistré dans sa version finale et dans sa version brute, c'est-à-dire avec toutes les frappes effacées pendant la rédaction.

Classification et indicateurs

Bursted ! catégorise les jets textuels selon leur fonction textuelle : les jets dits de production incrémentent le texte sur son bord droit, et ceux dits de révision interviennent sur le texte déjà produit. Il est possible de distinguer deux types de jets de révision : ceux qui portent sur le segment textuel produit lors de période d'exécution précédente, il s'agit alors d'un jet de révision immédiate, et ceux qui exigent un retour sur le texte au-delà de la dernière période d'exécution, on parle alors de jets de révision distante.

Concrètement, un jet textuel est étiqueté comme un jet de production ou de révision distante ou immédiate à partir de 2 critères : le type d'événement en début de jet (ajout / suppression) et la localisation dans le texte déjà écrit (dans le jet du bord droit du texte/ avant). Les jets textuels et les variables qui leur sont associées sont consignées dans le fichier bursts_[id].csv. La liste des indicateurs calculés est présentée dans le tableau 1 ci-dessous.

Indicateurs calculées	**Explication de l'indicateur**
Burst	*Contenu du jet d'écriture*
Burst_type	*Type de jet (Production, Révision Immédiate, Révision Distante)*
Burst_start_time	*Marqueur de temps en début du jet (ms)*
Burst_duration	*Durée du jet (ms)*
Pause_duration	*Durée de la pause pré-jet (ms)*
Cycle_duration	*Durée du jet et de la pause qui le précède (cycle)*
Pause_percent	*% De temps de pause dans le cycle*
Burst_percent	*% De temps de jet dans le cycle*
Pause_burst_ratio	*Rapport du temps de pause sur temps de jet du cycle*
St_pos	*Position du curseur en début de jet*
End_pos	*Position du curseur en fin de jet*
Doc_len	*Taille du doc en fin de jet (nombre de caractères)*
Burst_len	*Nombre de caractères ajoutés par le jet*
N_chars	*Nombre de frappes pendant le jet (caractères / espaces / suppressions)*
Typing_speed	*Vitesse de frappe du jet (nchars/s)*
Raw_burst	*Contenu des frappes du jet (caractères / espaces / suppressions)*

TABLE 1 : Liste et nature des indicateurs calculées par Bursted!

2 Conclusion

Bursted! est une application qui facilite l'exploitation de fichiers d'enregistrement des frappes au clavier (*keylogging* ou *keystroke logging*) lors de la rédaction de textes en fournissant un fichier de jets textuels et de variables associées prêt à être utilisées pour de la visualisation, pour calculer des variables secondaires, pour préparer des traitements statistiques, ou pour l'analyse automatique des jets textuels.

Bursted peut être utilisé dans divers protocoles de recherche, de l'analyse de cas à une étude expérimentale, en passant par des corpus (quelle que soit leur taille) ; avec des adultes comme avec des enfants. Il a déjà été utilisé pour traiter des keylogs d'écriture dans des contextes écologiques, expérimentaux, avec des enfants, des adultes, pour, par exemple, des analyses linguistiques des jets textuels ou du clustering multivariables par des algorithmes d'intelligence artificielle (Cf. projet PRO-Text).

Son développement est en cours de finalisation. L'ajout de paramètres supplémentaires répond à une volonté d'élargir son champ d'utilisation (par exemple en important des fichiers intermédiaires déjà nettoyés au format csv) et, à terme, de l'intégrer en tant que module de traitement à la plateforme de visualisation du projet PRO-Text.

Remerciements

Le développement de Bursted! a été développé dans le cadre du Projet PRO-Text porté par G. Cislaru et soutenu par l'Agence Nationale de la Recherche, ANR-18-CE23-0024-02.

Références

BOURIGA S. (2020). *L'impact sur la dynamique des traitements cognitifs de la rédaction sur ordinateur*. Thèse de doctorat, Université de Poitiers. http://www.theses.fr/2020POIT5003/document

BOURIGA S., OLIVE T., BORDES C. & CHESNET D. (soumis). *Impact of the writing tool and of translating on bursts of writing: Comparing handwriting vs. typing in low and high demands of translating*.

CISLARU G. & OLIVE T. (2018). *Le processus de textualisation*. Bruxelles : De Boeck.

CHENU F., PELLEGRINO F., JISA H., & FAYOL M. (2014). Interword and intraword pause threshold in writing. *Frontiers in psychology*, 5, 182. https://doi.org/10.3389/fpsyg.2014.00182

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

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

WENGELIN Å., TORRANCE M., HOLMQVIST K., SIMPSON S., GALBRAITH D., JOHANSSON V. & JOHANSSON R. (2009). Combined eyetracking and keystroke-logging methods for studying cognitive processes in text production. *Behavior Research Methods*, 41, 337–351. https://doi.org/10.3758/BRM.41.2.337