From keystrokes to annotated process data: Enriching the output of Inputlog with linguistic information

Lieve Macken[1], Veronique Hoste[1], Mariëlle Leijten[2,3], Luuk Van Waes[2]

LT3, Language and Translation Technology Team – University College Ghent and Ghent University, Belgium (1)

Department of Management, University of Antwerp, Belgium (2)

Flanders Research Foundation (3)

E-mail: lieve.macken@hogent.be, veronique.hoste@hogent.be, marielle.leijten@ua.ac.be, luuk.vanwaes@ua.ac.be

Abstract

Keystroke logging tools are a valuable aid to monitor written language production. These tools record all keystrokes, including backspaces and deletions together with timing information. In this paper we report on an extension to the keystroke logging program Inputlog in which we aggregate the logged process data from the keystroke (character) level to the word level. The logged process data are further enriched with different kinds of linguistic information: part-of-speech tags, lemmata, chunk boundaries, syllable boundaries and word frequency. A dedicated parser has been developed that distils from the logged process data word-level revisions, deleted fragments and final product data. The linguistically-annotated output will facilitate the linguistic analysis of the logged data and will provide a valuable basis for more linguistically-oriented writing process research. The set-up of the extension to Inputlog is largely language-independent. As proof-of-concept, the extension has been developed for English and Dutch. Inputlog is freely available for research purposes.

Keywords: keystroke logging, writing process, linguistic annotation

1. Introduction

Keystroke logging is an unobtrusive way to monitor written language production. The method is well established (Sullivan & Lindgren, 2006) and is applied to collect writing process data to study a wide range of topics from a cognitive, strategic or developmental perspective a.o. professional writing in educational settings (Van Waes, Leijten, & Van Weijen, 2009), second language writing (Miller, Lindgren, & Sullivan, 2008), spelling errors (Grabowski, 2008), revision strategies (Lindgren & Sullivan, 2006), and translation processes (Ehrensberger-Dow & Perrin, 2009; Jakobsen, 2005).

Various keystroke-logging programs have been developed, e.g. Scriptlog (Strömqvist, Holmqvist, Johansson, Karlsson, & Wengelin, 2006), Inputlog (Leijten & Van Waes, 2006), Translog (Jakobsen, 2006), and EyeWrite (Wengelin et al., 2009), each with a different focus[1]. The programs differ in the events that are logged (keyboard and/or mouse), in the environment that is logged (a program-specific text editor, MS Word or all Windows-based applications), in their combination with other logging tools (speech recognition and/or eye tracking) and in the depth of analysis of the output files.

The work described in this paper is based on the output of Inputlog[2], but can also be applied to other keystroke-logging programs. Inputlog is a word-processor independent keystroke-logging program that not only registers keystrokes, mouse movements, clicks and pauses in MS Word, but also in any other Windows-based application. Moreover, also speech input via Dragon Naturally Speaking (Nuance) can be logged. Inputlog is freely available for research purposes,

To open the way for more linguistically-oriented writing process research, we enhanced Inputlog by aggregating the logged process data from the character level (keystroke) to the word level. We further enriched the logged process data with different kinds of linguistic information: part-of-speech tags, lemmata, chunk boundaries, syllable boundaries, and word frequency. The extension can only be used for text produced in MS Word.

The enriched process data can be combined with temporal information (time stamps and pauses) and will facilitate the analysis of the logged data e.g. in view of the following research questions: *Do high frequency words contain less or more typos than low frequency words? To what extent does the syllable structure influence the pause time between bigrams? Do translation segments correspond to linguistic units? Are high frequency words replaced by lower frequency near-synonyms in the text revision process?*

Although the set-up of the extension to Inputlog is largely language-independent, some language-dependent resources are used. As proof-of-concept, we focussed in a first phase on English and Dutch.

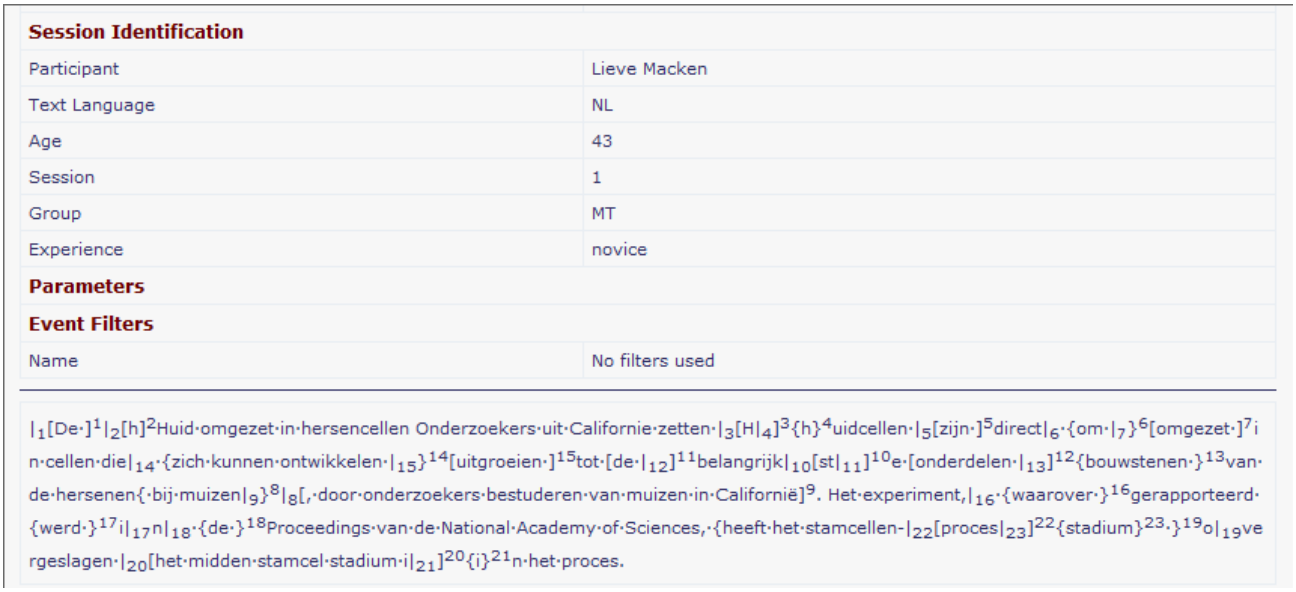
The remainder of this paper is structured as follows. Section 2 describes how the output of the keystroke-logging program was parsed, and section 3 discusses the linguistic annotations. In Section 4 we present a more elaborate example and Section 5 ends with some concluding remarks and directions for future work.

2. Output of Inputlog

Keystroke-logging programs store in essence the complete sequence of keyboard and/or mouse events in a chronological order.

[1] See http://www.writingpro.eu/logging_programs.php for an overview of available keystroke logging programs.

[2] http://www.inputlog.net/



Session Identification	
Participant	Lieve Macken
Text Language	NL
Age	43
Session	1
Group	MT
Experience	novice
Parameters	
Event Filters	
Name	No filters used

|1[De·]1|2[h]2Huid·omgezet·in·hersencellen Onderzoekers·uit·Californie·zetten·|3[H|4]3{h}4uidcellen·|5[zijn·]5direct|6·{om·|7}6[omgezet·]7in·cellen·die|14·{zich·kunnen·ontwikkelen·|15}14[uitgroeien·]15tot·[de·|12]11belangrijk|10[st|11]10e·[onderdelen·|13]12{bouwstenen·}13van·de·hersenen{·bij·muizen|9}8|8[,·door·onderzoekers·bestuderen·van·muizen·in·Californië]9. Het·experiment,|16·{waarover·}16gerapporteerd·{werd·}17i|17n|18·{de·}18Proceedings·van·de·National·Academy·of·Sciences,·{heeft·het·stamcellen·|22[proces|23]22{stadium}23.}19o|19vergeslagen·|20[het·midden·stamcel·stadium·i|21]20{i}21n·het·proces.

Figure 1: S-notation analysis of an Inputlog session

A more reader-friendly way of representing writing process data is to display all revisions at their positions in the text. The S-notation (Kollberg & Severinson Eklundh, 2002) contains information about the type of the revisions (insertion or deletion), the order of the revisions and the place in the text where the writing process was interrupted. The S-notation can be automatically generated from the keystroke loggings and has become a standard in the representation of the non-linearity in writing processes.

After the writing session has been recorded Inputlog can generate different data files from the source logging, a.o. a general analysis, a pause analysis or a revision analysis file. All analysis files are stored in XML format and also contain the session identification information. Figure 1 shows an example of writing process data represented as an S-notation. Figure 2 contains a smaller example, which will be used to explain how the S-notation is further processed and enriched with linguistic annotations.

S-Notation
Th[r|1]1e q{u}2ick|2 brown [dog]5{f}5|6{[i]7|8{o}8|9x}6|7 jumps over the [{old }4|5]11lazy [d|3]3[fox]9|10{dog}10|11.|4 The end[. |12]12!

Final text
The quick brown fox jumps over the lazy dog. The end!

Figure 2: writing process data represented as S-notation

The following conventions are used in the S-notation:

$\|_i$	A break in the writing process with sequential number *i*
{insertion}^{i}	An insertion occurring after break *i*
[deletion]^i	A deletion occurring after break *i*

As can be seen in Figure 2, the S-notation can become rather complicated as revisions can be embedded.

Two major obstacles need to be overcome in order to enrich the logged process data with different kinds of linguistic information. A first problem is that keystroke-logging programs basically log at the level of the character, while Natural Language Processing (NLP) tools work with sentences and words. A second problem is that keystroke-logging programs record process data (containing sentence fragments, unfinished sentences/ words and spelling errors), while NLP tools are typically designed for clean and grammatically correct text.

To tackle the first problem, the S-notation was segmented into sentences and tokenized. A dedicated sentence segmentation and tokenizer module was developed to take into account the S-notation conventions. To tackle the second problem, the S-notation was parsed and three types of data were extracted from the S-notation: word-level revisions, deleted fragments and the final writing product.

In theory, the word-level revisions can be extracted from the S-notation by retaining all words with word-internal square or curly brackets; the deleted fragments can be extracted from the S-notation by retaining only the words and phrases that are surrounded by word-external square brackets; and the final product data can be obtained by deleting everything in between square brackets from the S-notation. In practice, this process is more complex as the insertions and deletions are often nested. An example of the three different data types extracted from the tokenized S-notation is presented in Figure 3. To facilitate the readability of the resulting data, the indices are omitted.

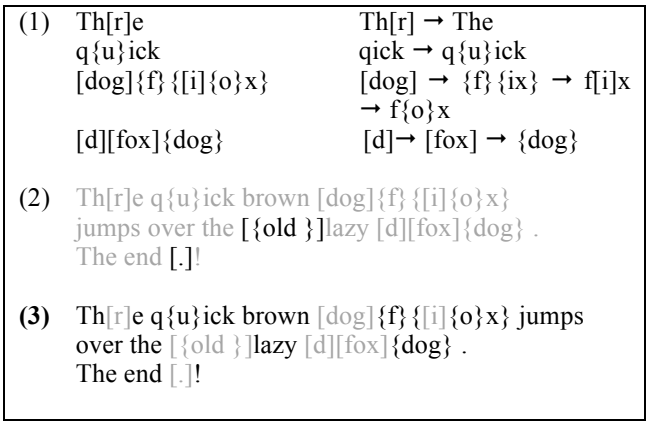


Figure 3: word-level revisions (1), deleted fragments (2) and the final writing product (3) extracted from the S-notation

3. Linguistic Annotations

We enriched the logged process data with different kinds of linguistic information: part-of-speech tags, lemmata, chunk boundaries, syllable boundaries, and word frequency.

As standard NLP tools are trained on clean data, these tools are not suited for processing input containing spelling errors. Therefore, we only enrich the final product data and the deletions with different kinds of linguistic annotations. As part-of-speech taggers typically use the surrounding local context to determine the proper part-of-speech tag for a given word (typically a window of two to three words and/or tags is used), the deletions in context are extracted from the S-notation to be processed by the part-of-speech tagger. The deleted fragments in context are retrieved from by S-notation by deleting all insertions. The contextual information is only used to optimize the results of the linguistic annotation.

For the shallow linguistic analysis, we used the tools suite developed by the Language and Translation Technology Team (LT3) of Ghent consisting of a part-of-speech tagger (LeTsTAG), a lemmatizer (LeTsLEMM) and a chunker (LeTsCHUNK). The LT3 tools are platform-independent and can thus be used in Windows and Unix environments. LeTsTAG and LeTsLEMM are trained with CRF++[3], an open source implementation of Conditional Random Fields (Lafferty, McCallum, & Pereira, 2001), which is a machine learning technique suited for labelling sequential data.

The English PoS tagger uses the Penn Treebank tag set, which contains 45 distinct tags. The Dutch part-of-speech tagger uses the CGN tag set codes (Van Eynde, Zavrel, & Daelemans, 2000), which is characterized by a high level of granularity. Apart from the word class, the CGN tag set codes a wide range of morpho-syntactic features as attributes to the word class. In total, 316 distinct tags are discerned.

During lemmatization, for each orthographic token, the base form (lemma) is generated. For verbs, the base form is the infinitive; for most other words, this base form is the stem, i.e. the word form without inflectional affixes. The lemmatizers make use of the predicted PoS codes to disambiguate ambiguous word forms, e.g. Dutch *landen* can be an infinitive (base form *landen*) or plural form of a noun (base form *land*). The lemmatizers were trained on the English and Dutch parts of the Celex lexical database respectively (Baayen, Piepenbrock, & van Rijn, 1993).

During text chunking syntactically related consecutive words are combined into non-overlapping, non-recursive chunks on the basis of a fairly superficial analysis. The LT3 chunkers are rule-based and contain a small set of constituency and distituency rules. Constituency rules define the part-of-speech tag sequences that can occur within a constituent (such as preposition + noun), while distituency rules define the part-of-speech tag sequences that cannot be adjacent within a constituent (such as noun + preposition). The use of distituency rules in the task of constituent boundary parsing was introduced by Magerman and Marcus (1990). The chunks are represented by means of IOB-tags. In the IOB-tagging scheme, each token belongs to one of the following three types: I (inside), O (outside) and B (begin); the B- en I-tags are followed by the chunk type, e.g. B-VP, I-VP.

Apart from a shallow linguistic analysis, we further added some word-level annotations on the final writing product and the deletions, viz. syllable boundaries and word frequencies. Syllabification was approached as a classification task: a large instance base of syllabified data is presented to a classification algorithm, which automatically learns from it the patterns needed to syllabify unseen data. The syllabification tools were trained on Celex[4].

Word frequency information for English and Dutch is retrieved from frequency lists compiled on the basis of Wikipedia pages, which were extracted from the XML dump of the English and Dutch Wikipedia of December 2011. The Wikipedia Extractor of Medialab[5] was used to extract the text from the wiki files. Frequencies are presented both as absolute frequencies and as frequency ranks.

After annotation, the final writing product, deleted fragments and word-level corrections are aligned and the indices are restored. The resulting output is presented in a table format (see Figure 4) and will be rendered in XML format. In the example presented in Figure 4, the first column gives the number of revisions occurring at each token, the second column shows the revision numbers as they occurred in the writing process.

[3] http://crfpp.sourceforge.net

[4] Visit http://lt3.hogent.be/en/tools/timbl-syllabification for a demo of the syllabification tool

[5] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

# Rev.	Index	Product	Word level corrections	PoS	Lemma	Chunks	Deletions	PoS	Lemma	Chunks	Syllabification	Absolute freq	Freq rank
1	1	The	Th[r][1]e	DT	the	B-NP					the	22995878	1
1	2	quick	q{u}ick	JJ	quick	I-NP					quick	30549	1378
		brown		JJ	brown	I-NP					brown	22508	1907
4	5,6,7,8	fox	[dog][5]{f}[5]{[i][7]{o}[8]x}[6]	NN	fox	I-NP					fox	5678	6394
		jumps		VBZ	jump	B-VP					jumps	3714	8741
		over		IN	over	B-PP					o-ver	314614	123
		the		DT	the	B-NP					the	22995878	1
1	11						[old][11]	JJ	old	I-NP	old	190027	185
		lazy		JJ	lazy	I-NP					la-zy	3438	9253
3	3,9,10	dog	[d][3][fox][9]{dog}[10]	NN	dog	I-NP					dog	35880	1158
		.		.	.	O							
		The		DT	the	B-NP					the	22995878	1
		end		NN	end	I-NP					end	129052	297
1	12						[.][12]	.	.	O			
		!		.	!	O							

Figure 4: Final writing product, deleted fragments and word-level revisions enriched with linguistic annotations

4. Post-editing example

In this section, we present a more elaborate example of the use of Inputlog in a translation context, and more specifically to monitor the task of human translators post-editing machine translation output.

Recently, keystroke loggings have been used to study the process of post-editing automatically translated text by Koehn (2009) and Carl et al. (2011). As keystroke logging tools record the actual post-editing process, the process data combined with temporal information (pauses and time stamps) can shed light on the machine translation passages that were difficult to process. Moreover, as Inputlog also logs all Windows-based events, researchers can also keep track of the external sources that were consulted and the search queries that were formulated. The consultation of external sources can be regarded as an indicator of uncertainty during translation (Angelone 2010)

The enriched process data will enable researchers to examine e.g. the following research questions. *What kind of automatically generated translation suggestions are taken over by the post-editor. Are these mainly lexical elements? To what extent does the post-editor consult external sources to verify the automatically generated translation suggestions?*

We will present the output of Inputlog for a post-editing task by means of the example presented in Figure 5. The original English text was taken from a BBC article[6]. Please note that the S-notation representing the process data is displayed in Figure 1.

[6] http://www.bbc.co.uk/news/health-16788809

Part of the linguistically-enriched output of Inputlog is presented in Figure 6. In the example, two phases can be discerned in the post-editing process. In the first phase, the post-editor assembles a fluent translation on the basis of the lexical elements that were present in the automatic translation, i.e. the post-editor mainly restructures the sentence using the lexical items available in the automatic translation. In a second phase the post-editor consults external sources (e.g. searched for a synonym of "onderdelen" (En: component) in synoniemen.net and replaced it with the more specific word "bouwstenen " (En: building blocks), which is also reflected in the

Original English text:
Skin transformed into brain cells
Skin cells have been converted directly into cells which develop into the main components of the brain, by researchers studying mice in California. The experiment, reported in Proceedings of the National Academy of Sciences, skipped the middle stem cell stage in the process.

Google Translate (Dutch):
De huid omgezet in hersencellen
Huidcellen zijn direct omgezet in cellen die uitgroeien tot de belangrijkste onderdelen van de hersenen, door onderzoekers bestuderen van muizen in Californië. Het experiment, gerapporteerd in Proceedings van de National Academy of Sciences, overgeslagen het midden stamcel stadium in het proces.

Post-edited text (Dutch):
Huid omgezet in hersencellen
Onderzoekers uit Californië zetten huidcellen direct om in cellen die zich kunnen ontwikkelen tot belangrijke bouwstenen van de hersenen bij muizen.
Het experiment, waarover gerapporteerd werd in de Proceedings van de National Academy of Sciences, heeft het stamcellen- stadium overgeslagen in het proces.

Figure 5: Original English text, automatic translation (Dutch) and post-edited text (Dutch)

absolute frequency of both words. A similar process can be observed in the lexical replacement of the verb phrase "uitgroeien tot" (En: grow into), which has been replaced by "zich kunnen ontwikkelen tot" (En: develop).

The extension to Inputlog will not only facilitate the linguistic analysis of the logged process data, it will also allow us to align the final result of the post-editing process with the original machine translation output.

Product	Word level corrections	Deletions	AbsFreq
		[De][1]	5827958
Huid	[h][2]Huid		3890
omgezet			2756
in			2616374
hersencellen			36
Onderzoekers			2502
uit			415839
Californië			5637
zetten			17701
huidcellen	[H][3]{h}[4]uidcellen		41
		[zijn][5]	2124155
direct			14440
		[omgezet][7]	2756
{om}[6]			267953
in			2616374
cellen			4845
die			526829
		[uitgroeien][15]	2223
{zich			173643
kunnen			221064
ontwikkelen}[14]			21044
tot			334912
		[de][11]	5827958
belangrijke	belangrijk[st][10]e		53258
		[onderdelen][12]	19895
{bouwstenen}[13]			225
van			3049345
de			5827958
hersenen			2248
		[,	
		door	521384

Figure 6: Post-edited text, deleted fragments and word-level revisions

5. Conclusion and future work

In this paper we presented how writing process data can be enriched with linguistic information. The annotated output will facilitate the linguistic analysis of the logged data and will provide a valuable basis for more linguistically-oriented writing process research.

In a first phase we only focussed on English and Dutch, but the method can be easily applied to other languages as well provided that the linguistic tools are available for a Windows platform. For the moment, the linguistic annotations are limited to part-of-speech tags, lemmata, chunk information, syllabification and word frequency information, but can be extended, e.g. by n-gram frequencies to capture collocations.

By aggregating the logged process data from the character level (keystroke) to the word level, general statistics (e.g. total number of deleted or inserted words, pause length before nouns preceded by an adjective or not) can be generated easily from the output of Inputlog as well.

By combining the time information provided by Inputlog with the linguistic information, researchers can easily calculate different measures, e.g. mean pause time at chunk boundaries, pause time before and after verb phrases, pause time at conjunctions, etc.

6. Acknowledgements

This research was funded by FWO (Flanders Research Foundation).

7. References

Angelone, E. (2010). Uncertainty, uncertainty management and metacognitive problem solving in the translation task. In G. M. Shreve & E. Angelone (Eds.), Translation and Cognition (pp. 17-40). Amsterdam; Philadelphia: Benjamins.

Baayen, R. H., R. Piepenbrock, & H. van Rijn. (1993). The CELEX lexical database on CD-ROM. Philadelphia, PA: Linguistic Data Consortium.

Carl, M., Dragsted, B., Elming, J., Hardt, D., & Jakobsen, A. L. 2011. *The Process of Post-Editing: a Pilot Study.* Paper presented at the 8th international NLPSC workshop. Special theme: Human-machine interaction in translation, Copenhagen Business School, Denmark.

Ehrensberger-Dow, M., & D. Perrin. 2009. "Capturing translation processes to access metalinguistic awareness". *Across Languages and Cultures, 20*(2), 275-288.

Grabowski, J. 2008. "The internal structure of university students' keyboard skills". *Journal of Writing Research, 1*(1), 27-52.

Jakobsen, A. L. 2005. "Investigating expert translators' processing knowledge". In V. Dam, H., J. Engberg & H. Gerzymisch-Arbogast (eds.), *Knowledge systems and translation*. 173-189. Berlin: Walter de Gruyter.

Jakobsen, A. L. 2006. "Translog: Research methods in translation". In Sullivan, K. P. H. & E. Lindgren (eds.), *Computer Keystroke Logging and Writing: Methods and Applications*. 95-105. Oxford: Elsevier.

Koehn, P. (2009), A process study of computer-aided translation. In: Machine Translation, 23 (4), pp. 241-263

Kollberg, P., & K. Severinson Eklundh. 2002. "Studying writers' revising patterns with S-notation analysis". In Olive, T. & C. M. Levy (eds.), *Contemporary Tools and Techniques for Studying Writing*. 89-104. Dordrecht: Kluwer Academic Publishers.

Lafferty, J. D., A. McCallum, & F. Pereira. 2001. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In *Proceedings of the International Conference on Machine Learning (ICML)*. 282-289. Williamstown, MA, USA.

Leijten, M., & L. Van Waes. 2006. "Inputlog: New Perspectives on the Logging of On-Line Writing". In Sullivan, K. P. H. & E. Lindgren (eds.), *Computer Keystroke Logging and Writing: Methods and Applications*. 73-94. Oxford: Elsevier.

Lindgren, E., & K. P. H. Sullivan. 2006. "Analysing

Online Revision". In Sullivan, K. P. H. & E. Lindgren (eds.), *Computer Keystroke Logging and Writing: Methods and Applications*. 157-188. Oxford: Elsevier.

Magerman, D. M., & M. P. Marcus. 1990. "Parsing a Natural Language Using Mutual Information Statistics". In *Proceedings of the Proceedings of the eighth National Conference on Artificial Intelligence (AAAI-90)*. 984-989.

Miller, K. S., E. Lindgren, & K. P. H. Sullivan. 2008. "The Psycholinguistic Dimension in Second Language Writing: Opportunities for Research and Pedagogy Using Computer Keystroke Logging". *TESOL QUARTERLY, 42*(3), 433-454.

Strömqvist, S., K. Holmqvist, V. Johansson, H. Karlsson, & A. Wengelin. 2006. "What keystroke logging can reveal about writing". In Sullivan, K. P. H. & E. Lindgren (eds.), *Computer Keystroke Logging and Writing: Methods and Applications*. 45-71. Oxford: Elsevier

Van Eynde, F., J. Zavrel, & W. Daelemans. 2000. "Part of Speech Tagging and Lemmatisation for the Spoken Dutch Corpus". In *Proceedings of the Proceedings of the second International Conference on Language Resources and Evaluation (LREC)*. 1427-1433. Athens, Greece.

Van Waes, L., M. Leijten, & D. Van Weijen. 2009. "Keystroke logging in writing research: Observing writing processes with Inputlog". *German as a foreign language (GFL)*(2-3), 41-64.

Wengelin, A., M. Torrance, K. Holmqvist, S. Simpson, D. Galbraith, V. Johansson, et al. 2009. "Combined eyetracking and keystroke-logging methods for studying cognitive processes in text production.". *Behavior Research Methods, 41*(2), 337-351.