Keystroke logging tools are widely used to monitor written language production. Basically, they record all keystrokes, including backspaces and deletions, together with timing information. To open the way for more linguistically-oriented writing process research, we enhanced the keystroke-logging program Inputlog by aggregating 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.

**Introduction**

Keystroke logging is an unobtrusive way to monitor written language production. The method is applied to study a wide range of topics from a cognitive, strategic or developmental perspective a.o. professional writing in educational settings, second language writing, spelling errors, revision strategies and translation processes.

Inputlog is 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 via www.inputlog.net

**S-notation**

The S-notation is the de facto standard for representing the non-linearity of the writing process. It displays all revisions at their positions in the text and can be automatically generated from the keystroke-logging data.

Conventions:

- A break in the writing process with sequential number $i$
- An insertion occurring after break $i$
- A deletion occurring after break $i$

**Post-editing example**

Use of Inputlog in a translation context
- To study post-editing of Machine Translations
- Temporal information can shed light on the MT passages that were difficult to process
- Logging of all windows-based events to study external sources that were consulted

A dedicated parser has been developed that distills from the logged process data word-level revisions, deleted fragments and final product data. The parser can handle nested revisions.

**Linguistic annotations**

Final product data and deletions are further enriched with linguistic annotations. To optimize the results of the shallow linguistic analysis, the deletions in context are analyzed.

**LT³ tools suite:**
- PoS tagger/lemmatizer/chunker
- Syllabification
- Word frequency information derived from Wikipedia

The resulting output is presented in a table format and will be rendered in XML format.

**Focus on English and Dutch as proof-of-concept**

**Acknowledgements**

This research was funded by Flanders Research Foundation (FWO)