Year: 2008

Computational linguistics for word processing: opportunities and limits

Mahlow, C; Piotrowski, M

Computational linguistics for word processing: opportunities and limits

Abstract

In this paper we briefly outline editing functions which are aware of the structures of natural languages by using methods from computational linguistics. Such functions could reduce errors and better support writers in realizing their communicative goals. However, linguistic methods have limits, and there are various aspects software developers have to take into account to avoid creating a solution looking for a problem:Language-aware functions could be powerful tools for writers, but writers must not be forced to adapt to their tools.

Computational Linguistics for Word Processing: Opportunities and Limits

Cerstin Mahlow and **Michael Piotrowski**
Institute of Computational Linguistics
University of Zurich
Zurich, Switzerland
{mahlow, mxp}@cl.uzh.ch

Abstract

In this paper we briefly outline editing functions which use methods from computational linguistics and take the structures of natural languages into consideration. Such functions could reduce errors and better support writers in realizing their communicative goals. However, linguistic methods have limits, and there are various aspects software developers have to take into account to avoid creating a solution looking for a problem: Language-aware functions could be powerful tools for writers, but writers must not be forced to adapt to their tools.

1 Introduction

Writing is a daily task for a great number of people. However, today's word processors offer only limited support for writing and editing: Most functions are character-based and thus force writers to translate high-level goals into low-level functions of the editor. This causes typical errors, e.g., missing verbs, agreement errors, or wrong word order. Functions improving the "brain-to-hand-to-keyboard-to-screen-connection" (Taylor, 1987, p. 79) as proposed by Dale (1989; 1996) or Mahlow and Piotrowski (2008) could help avoid several types of errors. Additionally, as cognitive resources are limited (McCutchen, 1996; Allen and Scerbo, 1983), language aware functions could reduce the effort needed to deal with word processors and help writers concentrate on their actual goals and keep control of their text.

2 Language Awareness in Word Processing

Checkers for spelling, grammar, and style, which are nowadays available for various languages in most word processors already provide a certain level of "language awareness." However, regardless of their quality (Vernon, 2000; McGee and Ericsson, 2002), they are suitable only for post-writing and do not support writers *during* writing and editing.

Writers should also receive interactive support. We propose two types of functions operating on linguistic elements, such as words, phrases, or clauses. (1) *Information functions* for highlighting elements (known as *syntax highlighting* in programming editors), or for providing writers with information about certain aspects of the text, such as prepositions used, sentences without verbs, or variants of multi-word expressions. Writers can interpret the results themselves and decide how to make use of them. (2) *Operations* for reordering, modifying, or deleting linguistic elements. In order to reduce the cognitive load, the number of actions necessary to reach a specific goal should be reduced drastically by combining sequences of core operations into higher-level functions closer to writers' goals and their mental model of the task.

Both types of functions require *linguistic knowledge* and *linguistic resources*. Linguistic knowledge will influence the ideal combination of existing core operations into higher-level functions a user can call with one keystroke: Reordering conjuncts, for example, is a highly complex task if a writer has to find the sequence of core operations on their own; using *one* operation reduces the risk of producing ungrammatical conjuncts. Linguistic resources will be needed for operations that modify certain linguistic elements: Pluralization of entire phrases may serve as an example here.

2.1 Opportunities

Linguistic resources can be used in static and dynamic settings. Static resources are lists of prepositions or the like. Dynamic resources include components capable of analyzing and generating structures like wordforms or phrases. Changing tense and mood of a sentence or a whole text obviously requires tagging and morphological analysis as well as generation of wordforms. Operations using syntactical and morphological components could offer writers new ways of working creatively with their texts: With one click they could apply changes to their texts, inspect the results, undo them, or try a different change. They could concentrate on their goal, play with words and phrases, and would not have to care about how to realize these changes, would not have to worry about forgetting one occurence, and would not have to keep in mind that other locations may need changes because of the original change (e.g., pluralizing the subject of a sentence requires adjustment of the finite verb).

2.2 Limits

Today's computers are capable of performing analyses and generation of linguistic structures in a reasonable time to be suitable for interactive use. But linguistic components usually fail to produce results that are 100% correct in terms of precision and recall, and, what is worse, for most of these components we can not predict if the result will be correct. When using them as basis for language-aware functions in word processors, writers must be aware that they should not blindly trust the system to avoid frustrations similar to those often associated with checkers.

A second limit are cases where linguistic resources can deliver correct, but ambigous results, e.g., it may not be possible to determine the exact category of a wordform. The editing function then cannot be executed automatically but has to interact with the writer to resolve the ambiguity.

A third limit is the danger of concentrating on aspects of (computational) linguistics rather than on aspects of the writing process and on writers' needs. For example, at first glance, it seems to be obvious that only operations resulting in grammatically well-formed structures should be allowed. But, on the one hand, the structure may not (yet) be completed and therefore not well-formed *before* executing an operation (e.g., when pluralizing a phrase consisting only of a determiner and an adjective, and the noun is added only after pluralizing). On the other hand, the relevant operation may be used only as one step in a complex sequence: After executing this operation more changes will be applied, and the result is not the end result (e.g., a list of wordforms, clearly not a phrase, shall be pluralized, and some of these are then moved to other parts of the text).

3 Conclusion

We presented the concept of language-aware functions in word processors using methods and systems from computational linguistics. They represent opportunities for supporting the writing process, but developers should avoid concentrating on technical aspects, causing dissatisfaction in writers and forcing them to adapt to their tools. Clearly, the goal must be to support writers by lowering the cognitive effort for complex operations and at the same time allowing them to define *their* goals and to be in control of their texts. This principle has to direct the implementation with respect to technology and usability.

References

Robert B. Allen and M. W. Scerbo. 1983. Details of command-language keystrokes. *ACM Trans. Inf. Syst.*, 1(2):159–178, April.

Robert Dale and Shona Douglas. 1996. Two investigations into intelligent text processing. In Mike Sharples and Thea van der Geest, editors, *The New Writing Environment: Writers at Work in a World of Technology*, chapter 8, pages 123–145. Springer.

Robert Dale. 1989. Computer-based editorial aids. In Jeremy Peckham, editor, *Recent Developments and Applications of Natural Language Processing*, chapter 2, pages 8–22. Kogan Page Limited.

Cerstin Mahlow and Michael Piotrowski. 2008. Linguistic support for revising and editing. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 9th International Conference, CICLing 2008, Haifa, Israel, February 17–23, 2008. Proceedings*, pages 631–642, Heidelberg. Springer.

Deborah McCutchen. 1996. A capacity theory of writing: Working memory in composition. *Educational Psychology Review*, 8(3):299–325.

Tim McGee and Patricia Ericsson. 2002. The politics of the program: MS Word as the invisible grammarian. *Computers and Composition*, 19(4):453–470, December.

Lee R. Taylor. 1987. Software views: A fistful of word-processing programs. *Computers and Composition*, 5(1):79–90.

Alex Vernon. 2000. Computerized grammar checkers 2000: capabilities, limitations, and pedagogical possibilities. *Computers and Composition*, 17(3):329–349, December.