UNIVERSITY OF GOTHENBURG

# Methods and tools for automating language engineering

Grégoire Détrez

Thesis submitted for the degree of Doctor of Philosophy in Computer Science at the Department of Computer Science and Engineering, Chalmers University of Technology & University of Gothenburg, Göteborg, Sweden

To be defended in public, 10.00 am, 2nd June, 2016 in room EA, Hörsalsvägen 11, Göteborg

Faculty opponent
Assistant Professor Mans Hulden
Department of Linguistics
University of Colorado
U.S.A.

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
SE-412 96 Göteborg, Sweden
Telephone + 46 (0)31–772 1000

UNIVERSITY OF GOTHENBURG

Methods and tools for automating language engineering
Thesis for the degree of Doctor of Philosophy in Computer Science
GRÉGOIRE DÉTREZ
Department of Computer Science and Engineering
Chalmers University of Technology & University of Gothenburg

# Abstract

Language-processing software is becoming increasingly present in our society. Making such tools available to the greater number is not just a question of access to technology but also a question of language as they need to be adapted, or localized, to each linguistic community. It is thus important to make the tools necessary to the engineering of language-processing systems as accessible as possible, for instance through automation. Not so much to help the traditional software creators but more importantly to enable communities to bring their language use into the digital world on their own terms.

Smart paradigms are created in the hope that they can decrease the amount of work for the lexicographer who wishes to create or update a morphological lexicon. In the first paper, we evaluate smart paradigms implemented in GF. How good are they to guess the correct inflection tables? How much information is required? How good are they at compressing the lexicon?

In the second paper, we take some distance from the smart paradigms, although they have been used in this work, they are not the main focus of the study. Instead, we compare two rule-based machine translation systems based on different translation models and try to determine the potential of a possible hybridization.

In the third paper we come back to the smart paradigms. If they can reduce the work of the lexicographer, someone still needs to create the smart paradigms in the first place. In this paper we explore the possibility of automatically creating smart paradigms based on existing traditional paradigms using machine-learning techniques.

Finally, the last paper presents a collection of tools meant to help grammar engineering work in the Grammatical Framework community: a tokenizer; a library to embedded grammars in Java applications; a build server; a document translator and a kernel to Jupyter notebooks.

Keywords: Natural language processing, Language Engineering, Morphology, Lexicon, Complexity