Task: Improved linguistic modules for Frog, ongoing maintenance and redesign for speed

ID: T139

Author: Iris Hendrickx, Antal van den Bosch, Maarten van Gompel

What exists: Frog is an integration of memory-based natural language processing (NLP) modules developed for Dutch. Frog performs tokenization, part-of-speech tagging, lemmatization and morphological segmentation of word tokens. At the sentence level Frog identifies non-embedded phrase chunks in the sentence, recognizes named entities and assigns a dependency parse graph.

What must be adapted / extended / created anew: Some components of the toolkit are already 25 years old; the more recent components are nearly one decade old. Some of the modules need to be updated to state of the art techniques and all sub modules can be upgraded by training them on newly available data sets. The POS-tagger could be retrained using modern algorithms. The dependency parser in Frog is intended to be a lean and fast alternative to the more precise but slow Alpino parser - it was already reimplemented completely and made faster in CLARIAH WP3 and can now be trained on significantly more training data (training software for Frog was explicitly developed in the scope of CLARIAH), achieving scores closer to that of Alpino (Canisius and Van den Bosch, 2007). In addition, this task provides for the continued maintenance of Frog, as a popular software package for which users often come to us with questions, bug reports, and feature requests.

Furthermore, we plan a fundamental re-design of Frog to:

- reduce complexity
- increase maintainability
- be more flexible
- speed things up The main aspect to consider: At the moment Frog uses FoLiA as it's internal data-structure. That seemed a good plan once, but with growing data-sets this now became a memory hog. Also it has some nasty MultiThreading issue sand makes processing line-by-line almost impossible, as the whole input file is stuffed into one FoLiA document. Redesigning the internal process to handle smaller chunks would speed up the process, and will deliver output at a more constant pace. (See frog#54)

Why important for CLARIAH (scientific impact): Updating and maintaining a toolkit that is already part of Clariah and a toolkit that is used often.

Targeted/Actual users: End-users/developers/researchers

Actual use (quantify!):

The paper describing the core tool and algorithms from 2007 (Van den Bosch et al, 2007) has been cited 127 times in other works. Frog is widely used in the Dutch NLP community. Exact figures are hard to track but an indication is given by visits to its website and github page (see our report).

Social Impact (concrete examples):

Satisfied users due to better quality of linguistic enrichment; improved text analytics further down the pipeline (e.g. when Frog is used for preprocessing in other information systems), handling huge data files

Lead: Ko van der Sloot

Proposed PM estimation (try to justify): 12 PM = retraining pos-tagger (2PM) + reimplementations and maintenance (3PM) + retraining and upgrading dependency tagger (3PM) + redesign(4PM)

Proposed Participants + PMs: Ko van der Sloot, Iris Hendrickx, Maarten van Gompel

Allocated PMs: 6PM

Deliverables

1. Software: Frog
   - Resolution of frog#54
2. Documentation

Milestones

1. v2.0 - Early 2019 - Outcome of the redesign effort

References

- Canisius, S., and Van den Bosch, A. (2007). Recompiling a knowledge-based dependency parser into memory. In Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP-2007, pp. 104-108. Borovets, Bulgaria.
- Van den Bosch, A., Busser, G.J., Canisius, S., and Daelemans, W. (2007). An efficient memory-based morphosyntactic tagger and parser for Dutch. In P. Dirix, I. Schuurman, V. Vandeghinste, and F. Van Eynde (Eds.), Computational Linguistics in the Netherlands 2006: Selected Papers of the Seventeenth CLIN Meeting. Utrecht: LOT, pp. 191-206.