Cutter – a Universal Multilingual Tokenizer

Johannes Graën
Universität Zürich
graen@cl.uzh.ch

Martin Volk
Universität Zürich
volk@cl.uzh.ch

Mara Bertamini
Universität Zürich
bertaminimara@gmail.com

We present Cutter, a rule-based tokenizer currently available for 17 languages. The rules, which are derived from annotation guidelines for human annotators, overlap to a great extent and are thus mainly language-independent. Both this property as well as the modular architecture of our rule system and our test-driven development approach render it possible to easily adapt the tokenizer to other languages and domains, genres or historical text variants, which do still not dispose of reasonable tokenization guidelines and/or tokenizing tools. Cutter can also be used as a web service and thus easily be integrated into any NLP pipeline.

Cutter consists of the two tools Cover and Knife, which are executed consecutively. Given at least one abbreviation list, Cover looks for potential abbreviations in the input text and subsequently looks them up in the abbreviation list. If the potential abbreviation is found in the list, it is isolated and exempted from further tokenization; abbreviations which can also be normal words are not being isolated and are thus subject to the applied rules. In a second step, Knife performs the actual tokenization by applying the rules in the order provided and thereby identifying tokens by patterns: Once a pattern matched a token, the whole input is split into token and non-token parts. Subsequently, the same patterns are applied to the non-token parts; when no further pattern is applicable, the leaves of the token tree correspond to the token sequence. The rules are organized in sets by 10 layers, from the most specific to the most general ones, thereby interweaving common with language-dependent rules.

A collection of unit tests covering most tokenization rules assures that new rules in form of token defining patterns do not conflict with existing ones, and thus guarantees consistency.