Title	Overview of tools for text normalization
Version	1.0
Author(s)	Darja Fišer, Jakob Lenardič
Date	12-12-2019
Status	For distribution
Distribution	BoD, NCF, UI
ID	CE-2019-1556

Table of contents

1. Introduction

In this CLARIN Resource and Tool Families report, we present an overview of language tools dedicated to text normalization. Text normalization is the process of transforming parts of a text into a single canonical form. It represents one of the key stages of linguistic processing for texts in which spelling variation abounds or deviates from the contemporary norm, such as in texts published in historical documents or on social media. After text normalization, standard tools for all further stages of text processing can be used. Another important advantage of text normalization is improved search which can be performed with querying a single, standard variant but takes into account all its spelling variants, be it historical, dialectal, colloquial or slang.

Our starting point for this overview is a survey of NLP Tools for Historical Documents, whose data were contributed by the participants of the CLARIN workshop: LP Tools for Historical

Documents organised by Martin Wynne, Christian Thomas and Bryan Jurish.[1] This survey presents a list of roughly 30 NLP tools of various functionalities, such as lemmatizers, named entity recognizers, as well as normalizers, which are aimed at processing historical texts. From this survey, we extracted all the tools dedicated to text normalization (8 in total). We then cross-referenced the 8 normalizers from the survey with the Virtual Language Observatory (VLO), the CLARIN Language Switchboard, and the websites of CLARIN consortia in order to determine the degree of their integration in the CLARIN infrastructure but also to add tools that were not part of the original survey.

Ultimately, we were able to collect a list of 14 tools dedicated to text normalization. Our primary aim was to evaluate the presentation of their metadata, paying particular attention to those tools that have VLO entries. Since this is the first tool overview conducted under the CRF initiative, our aim is also to present a standardized format for presenting the tool families that will be convertible to html tables for the online overviews on the CLARIN Resources Families page.

2. Tools for normalization in the CLARIN infrastructure

Table 1 lists 14 tools for text normalization. The tools are described with the following metadata:

(i) Functionality (e.g., normalization, named entity recognition)
(ii) Research domain (e.g., historical texts, social media texts)
(iii) Platform (e.g., Linux, OSX, Windows, cross-platform)
(iv) Licence
(v) Availability
(vi) Input and output formats of annotation
(vii) Publication

In Section 3, the metadata are summarized, focusing on issues with respect to their VLO presentation.

Table 1: Overview of text normalization tools, sorted by language

Tool	Languages	Description
Text Tonsorium **Functionality:** tokenization, segmentation, lemmatization, PoS-tagging, normalization, syntax analysis, NER, format transformations **Domain:** independent **Licence:** GPL	Afrikaans, Albanian, Armenian, Basque, Bosnian, Breton, Bulgarian, Catalan, Chinese, Corsican, Croatian, Czech, Danish, Dutch, English, Esperanto, Estonian, Faroese, Finnish,	Automatic construction and execution of several NLP workflows, which include normalisation. • **Availability:** download, web application • **Platform:** Ubuntu

[1] We would especially like to thank Bryan Jurish for his exhaustive comments on a previous version of this report.

	French, Galician, Georgian, German, Greek, Middle Low German, Haitian, Hindi, Hungarian, Icelandic, Indonesian, Inuktitut, Irish, Italian, Javanese, Kannada, Kurdish, Latin, Latvian, Lithuanian, Luxembourgish, Macedonian, Malay, Malayalam, Maltese, Norwegian, Occitan, Persian, Polish, Portuguese, Romanian, Russian, Serbian, Slovak, Slovenian, Spanish, Swahili, Swedish, Tamil, Turkish, Ukranian, Uzbek, Vietnamese, Welsh, Yiddish	
FoLiA-wordtranslate **Functionality:** normalization **Domain:** historical texts **Licence:** GNU Public License v3	Dutch	This tool does word-by-word lookups in a bilingual lexicon, applies some transformation rules and additionally it may consult the INT Historical Lexicon (to be obtained separately due to licensing restrictions). The aim is to use the modernisation layer to do further linguistic enrichment using contemporary models. This tool is part of the FoLiA-Utils collection as it operates on documents in the FoLiA format. Standalone it is only of very limited interest to others. • **Availability:** download • **Platform:** Linux/POSIX (C++) • **Input format:** plain text, FoLiA-XML • **Output format:** FoLiA-XML, plaintext

Nederlab Pipeline **Functionality:** modernisation, normalisation, tokenisation, conversion, PoS-tagging, lemmatisation, NER with entity linking (all functionality is derived from the individual parts rather than an inherent part of the workflow) **Domain:** independent **Licence:** GNU Public License v3	Dutch	This is a linguistic enrichment pipeline for Historical Dutch as developed for and used in the Nederlab project. This workflow, powered by Nextflow, invokes various tools, including Frog and FoLiA-wordtranslate , as well as other tools such as ucto (a tokeniser), folialangid (language identification), tei2folia (conversion from a subset of TEI to FoLiA, which serves as the exchange format for all our tooling, as well as the final corpus format for Nederlab). Due to the high complexity in tooling, this workflow and all dependencies are distributed as part of the LaMachine distribution. • **Availability:** download (GitHub) • **Platform:** Linux/POSIX (workflow itself runs on JVM, underlying components are mostly implemented in C++ and Python) • **Input format:** FoLiA XML • **Output format:** FoLiA XML • **Publication:** Brugman et al. (2016)
TiCClops **Functionality:** corpus processing, normalization **Domain:** independent	Dutch	This tool is designed to search a corpus for all existing variants of (potentially) all words occurring in the corpus. This corpus can be one text, or several, in one or more directories, located on one or more machines. TICCL creates word frequency lists, listing for each word type how often the word occurs in the corpus. These frequencies of the normalized word forms are the sum of the frequencies of the actual word forms found in the corpus. TICCL is a system that is intended to detect and correct typographical errors (misprints) and OCR errors (optical character recognition) in texts. When books or other texts are scanned from paper by a machine, that then turns these scans, i.e. images, into digital text

		files, errors occur. For instance, the letter combination `in' can be read as `m', and so the word `regeering' is incorrectly reproduced as `regeermg'. TICCL can be used to detect these errors and to suggest a correct form. • **Availability:** web application (CLAM interface), web application (PHILOSTEI interface) • **Platform:** cross-platform • **Input format:** images (tiff, djvu), plain text, xml, csv • **Output formal:** xml • **Publication:** Reynaert (2010)
@Philostei **Functionality:** corpus processing, normalization **Domain:** independent	Dutch, English, Finnish, French, German, German (Fraktur), Classical Greek, Modern Greek, Icelandic, Italian, Latin, Polish, Portuguese, Russian, Spanish, Swedish	This tool uses a combination of an Tesseract webservice for text layout analysis and OCR and a multilingual version of TICCL for normalization. • **Availability:** web application (outdated "page-not-found-link) • **Platform:** cross-platform • **Input format:** images (tiff, djvu), plain text, XML, csv • **Output formal:** XML • **Related publication:** Betti, Reynaert and van den Berg (2017)
PICCL: Philosophical Integrator of Computational and Corpus Libraries **Functionality:** OCR, normalization, tokenisation, dependency parsing, shallow parsing, lemmatization, morphological analysis, NER, PoS-tagging **Domain:** independent **Licence:** GNU GPL	Dutch, Swedish, Russian, Spanish, Portuguese, English, German, French, Italian, Finnish, Modern Greek, Classical Greek, Icelandic, German (Fraktur), Latin, Romanian	This is a set of workflows for corpus building through OCR, post-correction, modernization of historic language and Natural Language Processing. It combines Tesseract Optical Character Recognition, TICCL and FROG functionality in a single pipeline. • **Availability:** download (github) • **Platform:** cross-platform • **Input format:** images (tiff, vnd.djvu), plain text, xml • **Output formal:** FoLiA XML • **Publication:** Reynaert et al. (2015)

VARD2 **Functionality:** normalization **Domain:** historical texts **Licence:** CC-BY-NC-SA 2.0	English	This tool performs manual and automatic spelling normalisation based on letter replacement rules, phonetic matching (extended Soundex), edit distance, and variant mappings. • **Availability:** download • **Platform:** cross-platform (java) • **Input format:** plain text, rtf, SGML, XML • **Output format:** XML • **Publications:** see here
CAB historical text analysis **Functionality:** normalisation, PoS-tagging, lemmatisation **Domain:** historical texts	German	This tool is a WebLicht stub for the DTA::CAB service that provides orthographic normalisation, PoS-tagging and lemmatization for historical German. • **Availability:** web application • **Platform:** cross-platform • **Input format:** plain text, XML • **Output format:** stts tagset for PoS
CAB orthographic canonicalizer **Functionality:** normalisation **Domain:** historical texts	German	This tool is a WebLicht stub for the DTA::CAB service that provides orthographic normalization for historical German. • **Availability:** web application • **Platform:** cross-platform • **Input format:** plain text, XML • **Output format:** unspecified
DTA::CAB **Functionality:** lemmatization, PoS-tagging, normalization **Domain:** historical texts **Licence:** see here	German	This is an abstract framework for robust linguistic annotation, with public web-service including normalization and lemmatization for historical German • **Availability:** download, web application • **Platform:** cross-platform • **Input format:** various • **Output format:** various • **Publication:** Jurish (2012)
Normo **Functionality:** normalization	Hungarian	This tool is an automatic pre-normalizer for Middle Hungarian Bible translations. It employs a memory-based and a rule-

Domain: historical texts		based module, which consists of character- and token level rewrite rules. The tool was used for building the Old Hungarian Corpus. • **Availability:** unavailable • **Input format:** unspecified • **Output format:** unspecified • **Publication:** Vadász and Simon (2018)
Skrambi **Functionality:** OCR, normalization **Domain:** historical texts	Icelandic	This tool is a spell-checking application based on a noisy channel model, which can be used to achieve a true copy of the original spelling of historical OCR texts, and to produce a parallel text with modern spelling. • **Availability:** unavailable[2] • **Input format:** unspecified • **Output format:** unspecified
CSMTiser **Functionality:** normalization **Domain:** social media **Licence:** GNU Lesser General Public License v3.0	Slovenian	This is a trainable tool for text normalisation, based on Moses. • **Availability:** download • **Platform:** Linux • **Input format:** unspecified • **Output format:** unspecified • **Publication:** Ljubešić et al. (2016).
Turkish Natural Language Processing Pipeline **Functionality:** tokenisation, sentence splitting, normalisation,de-asciification, vowelisation, spelling correction, morphological analysis/disambiguation, named entity recognition, dependency parsing **Domain:** independent	Turkish	This is a pipeline of state-of-the-art Turkish NLP tools. • **Availability:** web application, web API • **Platform:** cross-platform • **Input format:** plain text • **Output format:** plain text • **Publication:** Eryiğit (2014)

[2] The normalizer currently isn't publicly available, but the related tools for OCR and spellchecking are available as web application.

3. Overview of the tools for text normalization

3.1. Identification

Only 4 out of the 14 normalization tools are listed in the VLO.

(1) CAB orthographic canonicalizer
(2) CAB historical text analysis
(3) CSMTiser
(4) Turkish Natural Language Processing Pipeline

Finding tools in the VLO is generally hard. This seems primarily to be due to two reasons, both of which are tied to the fact that "there are no facets [in the VLO] dedicated to software to refine one's search" (Odijk 2019: 121).

The Resource Type facet does not demarcate tools from language resources, which results in resource values such as *corpus, collection, lexicalResource* on the one hand and values for tools such as *web service* being listed together.

The values for different types of tool categories are not curated. On 8 November 2019 when the VLO was surveyed, the Resource Type facet listed at least the following values, where the main issue is the synonymity of several values (e.g. *Tool service* vs. *tool service*):

- *tool service* (50),
- *Tools* (29),
- *Tool service* (1),
- *tool service* (50)
- *software* (1487),
- *Software, multimedia* (1378)
- *software, webservice* (618)
- *web service* (513)
- *webservice* (183)
- *Web service* (1)
- *Applications* (41)
- *web application* (7)

In the case of the identified normalizers, (1) CAB orthographic canonicalizer, (2) CAB historical text analysis, and (3) CSMTiser are listed under *web service* and *webservice*. In this respect, our suggestion is that the Resource Type facet needs to be curated to at least reduce the number of labels to a semi-controlled vocabulary. This will reduce category labels that differ from one another only because of different spelling.

By contrast, (4) Turkish Natural Language Processing Pipeline is classified as both *software* and *webservice*. This is a to a degree redundant, since *software* is a hypernym of *webservice* and the latter category more or less entails the former. The need for such a double classification at different taxonomic levels could be eliminated if the VLO employed a separate facet – *e.g. Tool category* – dedicated only to software, which would then list tool categories at roughly the same taxonomic level (e.g., *web application* vs. *desktop application*).

The second reason for difficult findability relates to the fact that, as already pointed out by Odijk (2018: 124–125), many of the current CMDI profiles used to describe tools in the VLO do

not distinguish tool category (e.g., *web application* vs. *desktop application*) from tool functionality (e.g. *lemmatizer*, *normalizer*, etc.), and the latter information can only be discerned indirectly from e.g. the description field. This is indeed the case for the normalizers in VLO listed in (1)–(4) and is a problem for findability since the user mostly has to resort to trying out different search terms in the VLO for finding relevant tools.

For instance, the search term *normalizer* yields none of the tools in (1)–(4), while *normalization* does (but with low ranking for certain results like (3) *CSMTiser*), although this seems to be solely due to the fact that the term *normalization* appears in the description field. We therefore believe that introducing a facet in the VLO dedicated to searching by tool functionality would significantly improve visibility and findability.

3.2. Availability

In this section we list the availability of the tools and highlight those tools that seem to be unavailable because of outdated information in their repository entries.

3.2.1. Web application and for download

(1) DTA::CAB
(2) Text Tonsorium

3.2.2. For download

(1) VARD2
(2) CSMTiser
(3) FoLiA-wordtranslate
(4) PICCL: Philosophical Integrator of Computational and Corpus Libraries
(5) Nederlab Pipeline

3.2.3. Web application

(1) Turkish Natural Language Processing Pipeline
(2) CAB orthographic canonicalizer
(3) CAB historical text analysis

3.2.4. Unavailable

(1) Normo
(2) Skrambi
(3) @Philostei
(4) TiCClops

In sum, 2 tools are available both as web applications and for download, 5 tools are available only for download, and 3 tools are available as web applications, while 4 tools are unavailable.

There are two reasons for the unavailability. The tools (1) Normo and (2) Skrambi (the component for normalization) have not yet been published. By contrast, for (3) @Philostei and (4) TiCClops, the hyperlinks to their respective web applications yield a "page-not-found" error.

3.3. Metadata

3.3.1. Language

11 tools are aimed at processing a single language:

(1) Dutch (3 tools)
(2) English (1 tool)
(3) German (3 tools)
(4) Hungarian (1 tool)
(5) Icelandic (1 tool)
(6) Slovenian (1 tool)
(7) Turkish (1 tool)

3 tools are aimed at processing multiple languages:

(1) @Philostei (Dutch, English, Finnish, French, German, German (Fraktur), Classical Greek, Modern Greek, Icelandic, Italian, Latin, Polish, Portuguese, Russian, Spanish, Swedish)
(2) PICCL: Philosophical Integrator of Computational and Corpus Libraries (Dutch, Swedish, Russian, Spanish, Portuguese, English, German, French, Italian, Finnish, Modern Greek, Classical Greek, Icelandic, German (Fraktur), Latin, Romanian)
(3) Text Tonsorium (around 20 languages, see list here)

3.3.2. Functionality

The tools in Table 1 can be split into two groups w.r.t. functionality. First, half of the identified tools only perform normalization:

(1) VARD2
(2) CSMTiser
(3) FoLiA-wordtranslate
(4) Normo
(5) @Philostei
(6) TiCClops
(7) CAB orthographic canonicalizer

The second half constitutes tools/pipelines that provide – aside from text normalization – other functionalities as well which we list here in brackets:

(1) DTA::CAB (PoS-tagging, lemmatization)

(2) Skrambi (OCR, spellchecking)
(3) Text Tonsorium (tokenization, segmentation, PoS-tagging, lemmatization, syntactic analysis, NER, format conversions)
(4) PICCL: Philosophical Integrator of Computational and Corpus Libraries (tokenization, OCR, dependency parsing, shallow parsing, PoS-tagging, lemmatization, morphological analysis, NER)
(5) Nederlab Pipeline (tokenization, PoS tagging, lemmatization, NER with entity linking)
(6) CAB historical text analysis (PoS-tagging, lemmatization)[3]
(7) Turkish Natural Language Processing Pipeline (tokenization, sentence splitting, normalization, de-asciification, vowelization, spelling correction, morphological analysis, NER, dependency parsing)

In relation to the 4 tools with VLO entries (CSMTiser, CAB orthographic canonicalizer, CAB historical text analysis, Turkish Natural Language Processing Pipeline), we believe that their functionality is inefficiently presented in their respective metadata descriptions.

For instance, the CAB historical text analysis tool, which we take as representative of the issue regarding metadata presentation in the VLO, performs 3 tasks: PoS-tagging, lemmatization and normalization. In the VLO entry, these 3 tasks are only referred to in the Description tag, whereas the metadata description lacks a category which would define the functionality (compare with the VLO entry for *Frog*, which has a ToolClassification category defining the tool as both a Lemmatizer and a Parser), and the 3 tasks can only be indirectly discerned from the *Output* category which is nested under the *Operations* category (Figure 1). For example, the fact that the tool performs PoS-tagging is inferred from the *postags.tagset* parameter, which is valued for the *stts* tagset.

[3] Note that CAB historical text analysis and CAB orthographic canonicalizer are essentially the same tools (i.e., WebLicht stubs for DTA::CAB), since the "CAB orthographic canonicalizer WebLicht stub performs exactly the same operations as the CAB historical text analysis stub performs, but suppresses the output of everything but normalized surface forms (in order to allow users to choose alternative lemmatizers, part-of-speech taggers, etc." (Bryan Jurish p.c.). They are listed as separate tools because each has its own VLO entry.

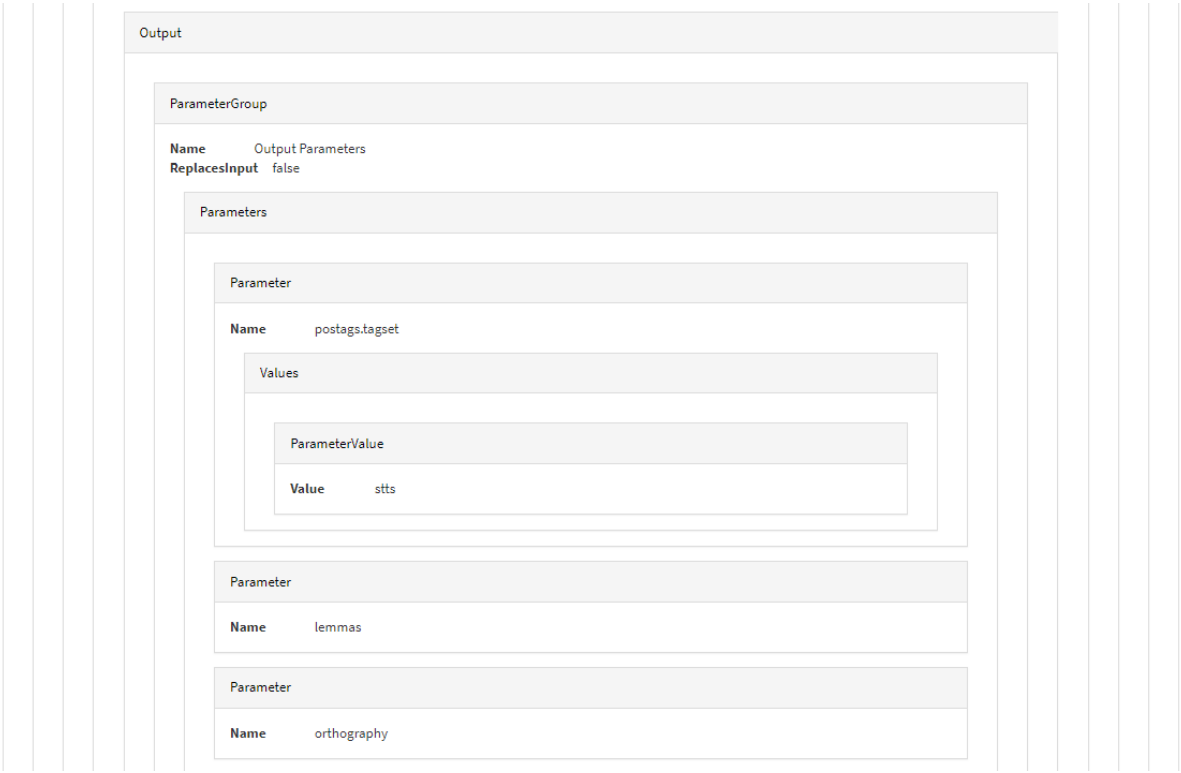

Figure 1: Metadata in the VLO for the CAB historical text analysis tool

In addition, the "Record details" tab of the VLO entry currently only displays metadata such as the collection type, resource type, data provider, as shown in Figure 2.

Figure 2: VLO entry for CAB orthographic canonicalizer

Our suggestion is that the functionality of tools should also be included with a clear presentation of at least the main functionality (e.g., *lemmatization* vs *normalization* vs *named entity recognition*) and some of the salient properties of the annotation performed, such as

the MIME type and the tagset. Essentially, we suggest that tools be described for the same types of fine-grained metadata that Odijk (2019: 121–122) suggests, and that the metadata that pertain to tool functionality should be explicitly presented in the Record details tab of the VLO. Note that the clarin:el repository already describes tools in a similar manner to what we are suggesting, as shown in Figure 3.

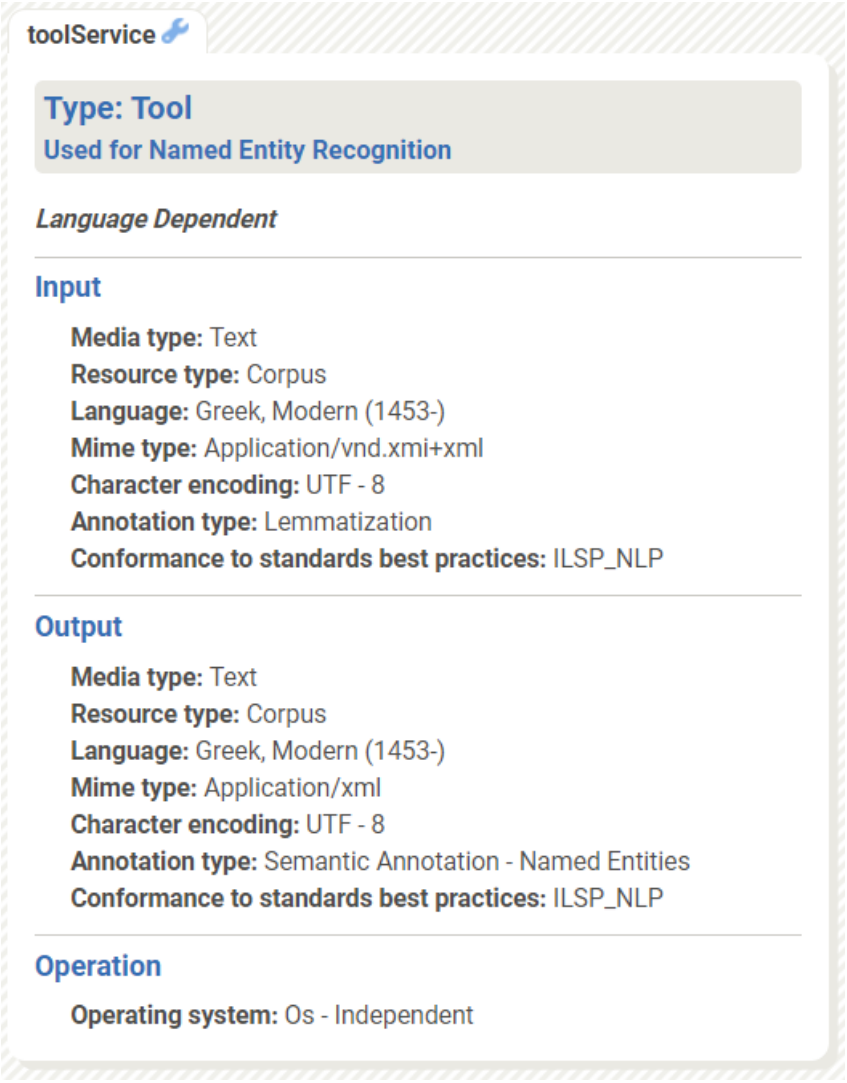

Figure 3: The description of GrNE-Tagger in clarin:el

3.3.3. Input and output types

The following 9 tools specify the types of the input and output files:

(1) VARD2
   a. **Input:** plain text, rtf, SGML, XML
   b. **Output:** XML

(2) DTA::CAB

    a. **Input:** various
    b. **Output:** various
(3) FoLiA-wordtranslate
    a. **Input:** plain text, FoLiA-XML
    b. **Output:** plain text, FoLiA-XML
(4) @Philostei
    a. **Input:** images (tiff, djvu), plain text, XML, csv
    b. **Output**: XML
(5) TiCClops
    a. **Input:** images (tiff, djvu), plain text, xml, csv
    b. **Output:** XML
(6) Text Tonsorium
    a. **Input:** plain text, RTF, PDF, html, doc, TEI5, and several others
    b. **Output:** plain text, RTF, PDF, html, doc, TEI5, and several others
(7) PICCL: Philosophical Integrator of Computational and Corpus Libraries
    a. **Input:** images (tiff, vnd.djvu), plain text, xml
    b. **Output:** FoLiA XML
(8) Nederlab Pipeline
    a. **Input:** FoLiA XML
    b. **Output:** FoLiA XML
(9) Turkish Natural Language Processing Pipeline
    a. **Input:** plain text
    b. **Output:** plain text

The following 5 tools do not specify the input and output types (which at least in the case of tools (2) and (3) is due to them not having been published yet) or do so only partially.

(1) CSMTiser
(2) Normo
(3) Skrambi
(4) CAB orthographic canonicalizer (partially specified for input but not for output)
(5) CAB historical text analysis (partially specified for input but not for output)

3.3.4. Licence

The following tools lack information on licence:[4]

(1) Normo
(2) Skrambi
(3) @Philostei
(4) TiCClops
(5) CAB orthographic canonicalizer
(6) CAB historical text analysis

[4] Note that tools (5) and (6) are missing licence only in their VLO entries. As Bryan Jurish (p.c.) points out, the licence of their underlying software is the same as that of the DTA::CAB service.

(7) Turkish Natural Language Processing Pipeline

Otherwise, the most common licence is the GNU General Public Licence (5 tools), while 1 tool is available under CC-BY.

4. Conclusion

We have provided an overview of 14 tools for text normalization and evaluated them from the perspective of VLO findability and availability, as well as from the perspective of the metadata describing functionality, input type, and licence.

In relation to findability, only 4 of the 14 (29%) tools are available in the VLO. Following Odijk (2019), we have argued that VLO findability is suboptimal since there are no VLO facets aimed at finding tools.

In relation to accessibility, 2 (14%) tools are available both as web applications and for download, 5 (36%) tools are available only for download, and 3 (21%) tools are available only as a web application, while 4 (29%) tools are unavailable. Crucially, unavailability is often due to outdated or broken links, which should thus be remedied by the responsible curators.

In relation to language, most (11/14, 79%) tools are aimed at normalizing texts within a single language (3 Dutch, 1 English, 3 German, 1 Hungarian, 1 Icelandic, 1 Slovenian, 1 Turkish), while the rest have a multilingual scope.

In terms of functionality, 7 (50%) tools are dedicated normalizers while 7 larger tool pipelines that apart from normalization also provide functionalities such as PoS-tagging, lemmatization, and named entity recognition. What is crucial is that, from the perspective of the 4 tools listed in the VLO, their functionality is unclearly presented, i.e., it is often listed only as part of the tool's general description. By contrast, our suggestion is that functionality and related properties thereof (e.g., types of output and input, tagsets) be clearly represented on the main tab of the VLO entry. Similarly, the type of input and output is unclearly defined for 5 (36%) of the tools

Finally, half of the tools are missing information on licence.

5. References

Betti, Arianna, Martin Reynaert, and Hein van den Berg. 2017. @PhilosTEI: Building Corpora for Philosophers. In *CLARIN in the Low Countries*, edited by Jan Odijk and Arjan van Hessen. London: Ubiquity Press. https://doi.org/10.5334/bbi.32.

Brugman, Hennie, Martin Reynaert, Nicoline van der Sijs, René van Stipriaan, Erik Tjong Kim Sang, and Antal van den Bosch. 2016. Nederlab: Towards a Single Portal and Research Environment for Diachronic Dutch Text Corpora. In *Proceedings of LREC 2016*, 1277–1281. http://www.lrec-conf.org/proceedings/lrec2016/pdf/471_Paper.pdf.

Eryiğit, Gülşen Cebiroğlu. 2014. ITU Turkish NLP Web Service. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*. https://web.itu.edu.tr/gulsenc/papers/itunlp.pdf.

Jurish, Bryan. 2012. *Finite-State Canonicalization Techniques for Historical German*. PhD dissertation. Universität Potsdam. http://kaskade.dwds.de/~moocow/mirror/pubs/jurish2012diss.pdf.

Ljubešić, Nikola, Katja Zupan, Darja Fišer, and Tomaž Erjavec. 2015. Normalising Slovene data: historical texts vs. user-generated content. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016*), 146–155. https://www.linguistics.rub.de/konvens16/pub/19_konvensproc.pdf.

Odijk, Jan. 2019. Discovering software resources in CLARIN. In *Selected papers from the CLARIN Annual Conference 2018*, 121–132. http://www.ep.liu.se/ecp/159/013/ecp18159013.pdf.

Reynaert, Martin, Maarten van Gompel, Ko van der Sloot, and Antal van den Bosch. 2015. PICCL: Philosophical Integrator of Computational and Corpus Libraries. http://www.nederlab.nl/cms/wp-content/uploads/2015/10/Reynaert_PICCL-Philosophical-Integrator-of-Computational-and-Corpus-Libraries.pdf.

Reynaert, Martin. 2010. Character confusion versus focus word-based correction of spelling and OCR variants in corpora. *International Journal on Document Analysis and Recognition* 14 (2): 173–187. https://doi.org/10.1007/s10032-010-0133-5.

Vadász, Noémi, and Eszter Simon. 2018. NORMO: An Automatic Normalization Tool for Middle Hungarian. In *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities*, 227–236. https://www.oeaw.ac.at/fileadmin/subsites/academiaecorpora/PDF/CRH2.pdf.