The Journal of Open Source Software

UralicNLP: An NLP Library for Uralic Languages

Mika Hämäläinen[1]

1 Department of Digital Humanities, University of Helsinki

DOI: 10.21105/joss.01345

Software

- Review
- Repository
- Archive

Submitted: 09 March 2019
Published: 09 May 2019

Introduction

In the past years the natural language processing (NLP) tools and resources for the small Uralic languages have received a major uplift. The open-source infrastructure by Giellatekno (Moshagen, Pirinen, & Trosterud, 2013) has served a key role in gathering these tools and resources in an open environment for researchers to use.

However, the many of the crucially important NLP tools, such as FSTs (finite-state transducers) (cf. Beesley & Karttunen, 2003) for processing morphology and CGs (constraint grammars) (cf. Karlsson, Voutilainen, Heikkilä, & Anttila, 1995) for syntax, require specialized tools with a learning curve. Their use for a researcher who is not familiar with them can be challenging, and ultimately lead to simply ignoring the existence of the resources.

This paper presents UralicNLP, a Python library, the goal of which is to mask the actual implementation behind a Python interface. This not only lowers the threshold to use the tools provided in the Giellatekno infrastructure but also makes it easier to incorporate them as a part of research code written in Python.

Functionalities

This section describes the current functionalities of the Python library. At the time of writing, the library focuses on low-level NLP tasks. Additionally, semantic models are provided for a limited number of languages.

Morphology

The FST models provided in the Giellatekno infrastructure are built on HFST (Helsinki Finite-State Technology) (Lindén et al., 2013), which is an open-source tool for compiling and running scripts that follow the FST formalism. UralicNLP uses the compiled FST models available through the Online Dictionary of Uralic Languages (Hämäläinen & Rueter, 2018).

The library provides morphological analysis on a word level for all supported languages. This means that it will output all the possible morphological readings for an input word form. The morphological analyzers provide typically a lemma, part-of-speech tag and a list morphological tags such as the number and case of the word from. The list of possible readings may include weights indicating the probability of the analysis. However, these are not currently implemented in any of the FST models. For example, for the Finnish word *voit*, the analyzer gives readings *voi* (butter) as a noun in the plural of nominative and *voida* (can) as a verb in the second person of singular.

Given a lemma, part-of-speech tag and morphological tags separated by a plus sign, it is possible to use UralicNLP to generate word forms. This inflection mechanism can be useful in various natural language generation tasks. For instance, giving the Finnish word *kissa*, and the morphological tags *plural* and *genitive*, the library inflects the word as *kissojen*.

Disambiguation

Whereas the morphological functionality does the analysis only on the word level, the disambiguator applies CG rules to rule out the morphological readings that are not suitable in the context by using the VISL CG-3 tool (Bick & Didriksen, 2015). These CG rules originate from the Giellatekno repository, but they are downloaded through the Online Dictionary of Uralic Languages.

Depending on the language, the disambiguator can often output multiple readings because the rules are not sufficient to fully disambiguate the sentence. It is important to take this into account when using the functionality.

Lexical Lookup

The API of the Online Dictionary of Uralic Languages provides essentially the same data as in the Giellatekno multilingual XML dictionaries in a JSON format. The actual contents of the data depend on the language, but information such as semantic tags, URLs to audio files, example sentences and translations in multiple languages is oftentimes provided.

In order to use the lexical lookup, the ISO code of the minority language needs to be specified. This will limit the query into the dictionary of that language. Queries can be done either with a lemma or with an inflectional form. It is also possible to query in one of the languages the minority language words are translated to.

Semantics

UralicNLP provides an easy to use programmatic interface to SemFi and SemUr databases (Hämäläinen, 2018a). These databases contain semantic information of words given their syntactic relations. For example, the database can be used to list out all the verbs that can have *koira* (dog) as a subject together with the frequency of the co-occurrence of the verbs and the noun *koira* in a corpus. SemFi has previously been used in the computationally creative task of poem generation (Hämäläinen, 2018b).

SemUr consists of databases for endangered Uralic languages that have been translated automatically from SemFi. Both of SemFi and SemUr are structurally identical SQLite databases which makes it possible to query them with the same methods provided by UralicNLP.

Universal Dependency Parser

UralicNLP comes with functionality to parse Treebanks. The parsed Treebanks can be queried effectively with the different universal dependency annotations such as part-of-speech, dependency relation and lemma. The queries support regular expressions. This functionality is useful with the growing number of UD Treebanks available for Uralic languages.

Distribution

UralicNLP is distributed as an installable package through PyPi with the name uralicNLP[1]. The source code is released under the Apache open source license on GitHub.

References

Beesley, K. R., & Karttunen, L. (2003). Finite-state morphology. In (pp. 451–454). Stanford, CA: CSLI Publications.

Bick, E., & Didriksen, T. (2015). CG-3 — beyond classical constraint grammar. In *Proceedings of the 20th nordic conference of computational linguistics, NODALIDA 2015, may 11-13, 2015, vilnius, lithuania* (pp. 31–39). University of Southern Denmark, Denmark; Linköping University Electronic Press, Linköpings universitet.

Hämäläinen, M. (2018a). Extracting a semantic database with syntactic relations for Finnish to boost resources for endangered Uralic languages. In *Proceedings of the logic and engineering of natural language semantics 15 (LENLS15)*. Retrieved from https://helda.helsinki.fi/handle/10138/282733

Hämäläinen, M. (2018b). Harnessing NLG to create Finnish poetry automatically. In *Proceedings of the ninth international conference on computational creativity* (pp. 9–15).

Hämäläinen, M., & Rueter, J. (2018). Advances in synchronized XML-MediaWiki dictionary development in the context of endangered Uralic languages. In *Proceedings of the eighteenth EURALEX international congress* (pp. 967–978).

Karlsson, F., Voutilainen, A., Heikkilä, J., & Anttila, A. (1995). Constraint grammar: A language-independent system for parsing unrestricted text. Walter de Gruyter.

Lindén, K., Axelson, E., Drobac, S., Hardwick, S., Kuokkala, J., Niemi, J., Pirinen, T. A., et al. (2013). HFST a system for creating NLP tools. In *International workshop on systems and frameworks for computational morphology* (pp. 53–71). Springer. doi:10.1007/978-3-642-40486-3_4

Moshagen, S. N., Pirinen, T. A., & Trosterud, T. (2013). Building an open-source development infrastructure for language technology projects. In *Proceedings of the 19th nordic conference of computational linguistics (NODALIDA 2013); may 22-24; 2013; oslo university; norway. NEALT proceedings series 16* (pp. 343–352). University of Tromsø, Norway; Linköping University Electronic Press.

[1] pip install uralicNLP