Easily Accessible Language Technologies for Slovene, Croatian and Serbian

Nikola Ljubešić,[†‡] Tomaž Erjavec,[†] Darja Fišer,[*†] Tanja Samardžić,[♣] Maja Miličević,[♠] Filip Klubička,[‡] Filip Petkovski[♢]

[†]Department of Knowledge Technologies, Jožef Stefan Institute
tomaz.erjavec@ijs.si

[‡]Faculty of Humanities and Social Sciences
nljubesi@ffzg.hr, fklubicka@ffzg.hr

[*]Faculty of Arts, University of Ljubljana
darja.fiser@ff.uni-lj.si

[♣]CorpusLab, University of Zürich
tanja.samardzic@uzh.ch

[♠]Faculty of Philology, University of Belgrade
m.milicevic@fil.bg.ac.rs

[♢]Freelance developer, Macedonia
filip.petkovsky@gmail.com

Abstract

In this paper we present the pipeline of recently developed language technology tools for Slovene, Croatian and Serbian. They currently cover text segmentation, text normalisation, part-of-speech tagging, lemmatisation and inflectional lexicon lookup. Most rely on machine learning approaches, such as statistical machine translation and conditional random fields, capable of producing high-quality models for the phenomenon covered. Special emphasis is put on easy accessibility of these tools by offering them and the trained models for all three languages as (1) open source via public git repositories and (2) online in the form of web applications and web services.

1. Introduction

With the increasing availability of language technologies for various languages, different scientific areas, including those of social sciences and humanities (SSH), have started to perceive the usefulness of such technologies for their own research. Given the lower level of technical competence of most researchers in SSH in comparison to the areas language technologies are developed in, a significant technological gap has to be filled, which would enable SSH scholars to include the developed technologies in their own research.

This paper presents a joint effort to make language technology for three western South Slavic languages – Slovene, Croatian and Serbian – more widely accessible. For Slovene there are already tools available for tagging and lemmatisation in form of web applications, such as ToTaLe[1] (Erjavec et al., 2005) and Obeliks[2] (Grčar et al., 2012), but of lower quality than the one presented in this paper. For Croatian there was a web application available hosting tools trained on the SETimes.HR corpus (Agić and Ljubešić, 2014), but given the superior quality of the tools presented in this paper, this web application is currently forwarding requests to the new solution. For Serbian there were no technologies available up to this point.

While many toolchains already exist, e.g. Gate (Cunningham et al., 2011), FreeLing (Padró and Stanilovsky, 2012), OpenNLP (Apache Software Foundation, 2014), there are two main reasons why they do not suit our needs. First, the choice of technology in existing toolchains is mostly oriented toward the major world languages. Subsequently, for part-of-speech tagging HMMs are used which, in case of more complex inflectional languages such as the Slavic ones, do not yield the best results. The other reason is that most of the toolchains only cover basic tasks like part-of-speech tagging, parsing and named entity recognition while our toolchain has already touched on more specific tasks like non-standard language normalisation.

Furthermore, we put special emphasis on bridging the aforementioned technology gap by offering three modes of using the developed technologies: (1) as open source programs available from the public GitHub repository, (2) as RESTful web services and (3) through a web application. The first is intended for technically experienced people who are capable of installing the tools and their dependencies and want to process large amounts of data, as well as control input and output formats. The latter two are better suited for those who are either processing smaller datasets or do not have the knowledge or hardware capabilities to install and run the tools locally. The web services can be used from code either directly as JSON-based RESTful services or through an available Python library. The developed web application is primarily intended for teaching purposes, trying out the technologies, debugging or processing only a handful of documents.

The tools are, for the most part, based on the machine

[1] http://nl.ijs.si/tei/convert/

[2] http://eng.slovenscina.eu/tehnologije/oznacevalnik

learning paradigm, and comprise the learning and execution components as well as models for Slovene, Croatian and Serbian developed by training the tool on the best resources available for the task.

The paper is structured as follows: the following section gives a short overview of the developed technologies, Section 3 describes the available modes of using them, while the last section gives a short-term plan of future developments.

2. Language Technology Tools

2.1. Inflectional Lexicons

Slavic languages in general have a complex inflectional system and lexicons covering this layer of language are important for almost any task of automatic language processing. All three languages of interest now have available large inflectional lexicons, in particular:

- the Slovene Sloleks lexicon (Dobrovoljc et al., 2015), 100,805 lexemes in size;
- the Croatian hrLex lexicon (Ljubešić, 2016a), 99,680 lexemes in size;
- the Serbian srLex lexicon (Ljubešić, 2016b), 105,358 lexemes in size.

The entry in each lexicon consists of the lemma and its complete inflectional paradigm comprising the word forms, their morphosyntactic descriptions and their corpus frequencies.

Through our web services and application we give a unified interface to all three resources.

2.2. Diacritic Restoration Tool

In computer-mediated communication, such as emails, instant messages, tweets etc. users of Latin-based scripts often replace characters with diacritics with their ASCII equivalents for ergonomic reasons, especially when typing on tablets and smartphones. Such text is typically easily understandable to humans but very difficult for computational processing because many words without the diacritics become ambiguous or unknown. At the same time, computer-mediated communication has become a hot topic of research and application, which is why high-quality processing of such language is in high demand.

We have developed a diacritic restoration tool called REDI (Ljubešić et al., 2016a)[3] with models covering all three languages of interest. The tool is trained on large corpora and consists of two components: the translation model (the probability of a standard word given its dediacritised version) and the language model (the probability of the standard word given its context). For estimating the token translation probability we use the maximum likelihood estimate of a diacritised form given the dediacritised one, while for estimating the context probability we use KenLM (Heafield, 2011) with the default parameters. These two components are combined with a log-linear model.

The token-level accuracy of the tool is around 99.5% on standard text and around 99.2% on non-standard text (Ljubešić et al., 2016a). The tool significantly outperforms charlifter,[4] so far the only open source tool available for this task on the target languages, which achieves around 97% accuracy on standard and around 94% on non-standard text.

2.3. Non-Standard Text Normalisation

Computer-mediated communication is often written in non-standard language, where users are either not acquainted with the language norm or, more often, intentionally use phonetic and dialectal spelling. Similarly, historical texts are also written in language which is significantly different from the contemporary standard. However, annotation tools, such as PoS taggers and lemmatisers, are typically trained on standard language and perform poorly on non-standard texts. As developing new text tools for each language variety is very time consuming and expensive, a typical approach is to first standardise the spelling of words and only then apply further processing on them.

For normalising words in user-generated content we use character-level statistical machine translation (CSMT), the Slovene variant of which, applied both to computer-mediated communication and historical texts, is presented in Ljubešić et al. (2016). The technology is based on the well-known SMT system Moses (Koehn et al., 2007), which is trained on a manually normalised collection of tweets split into sequences of characters. For all three languages the training set comprises 80,000 tokens.

The last experiments on Slovene show that for less standard tweets the error reduction obtained when applying CSMT is $\sim$70% while for more standard tweets it is $\sim$50%. When comparing the CSMT systems to a baseline which applies the most probable token transformation as estimated on the same training data, the error reduction on less standard tweets is $\sim$35% and on more standard tweets $\sim$45% (Ljubešić et al., 2016).

2.4. Morphosyntactic Annotation and Lemmatisation

For Slavic languages morphosyntactic tagging is probably the most important step in text annotation, and is still an interesting topic of research. Such languages with their large tagsets of morphosyntactic descriptions (MSDs) and often limited training data still offer significant room for improvement in tagging accuracy. Similar points hold for lemmatisation, the process of assigning the base form to a word form in running text. On one hand, the rules for predicting the lemma of a word form are complex and have many exceptions, while, on the other, the word forms are often ambiguous and their MSD tag is needed to correctly determine the lemma.

We recently developed a new tagger combined with a lemmatiser, explicitly developed for high-quality processing of the languages of interest (Ljubešić and Erjavec, 2016; Ljubešić et al., 2016b).[5] The tagger follows the approach by Grčar et al. (2012) but replacing their instance classifier (SVM) with a sequential one (CRF) and re-engineering the optimal features given the different nature

[3] https://github.com/clarinsi/redi

[4] https://sourceforge.net/projects/lingala/files/charlifter/

[5] https://github.com/clarinsi/reldi-tagger

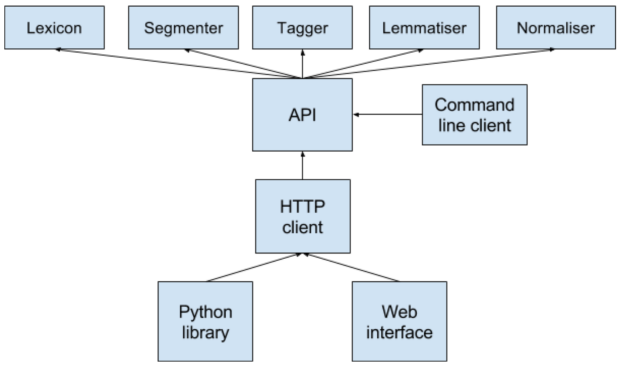

Figure 1: Architecture of the system exposing the developed language technologies.

of the classifier. With this we significantly improve their results, with an error reduction of ∼25% on both known and unknown words.

The Slovene model for our tagger is trained on the ssj500k corpus (Krek et al., 2015) and the Sloleks lexicon with the reported tagging accuracy at 94.27%. The Croatian and Serbian models are trained on the Croatian hr500k corpus (Ljubešić et al., 2016b), with the Croatian model using the hrLex lexicon (Ljubešić, 2016a) and the Serbian one the srLex lexicon (Ljubešić, 2016b). The reported tagging accuracy for Croatian is 92.53% and for Serbian 92.33%.

The currently used lemmatiser applies the mentioned lexicons in case the surface form and guessed MSD can be found in them. Otherwise, it uses supervised machine learning to predict the transformation of the surface form to the lemma. The predicted transformation is formalised as a 4-tuple (prefix length, prefix substitute, suffix length, suffix substitute); for example, in the transformation from *najvećih* to *velik* the 4-tuple is `(3,"",3,"lik")`. The features used for prediction are the suffixes of different length and the guessed MSD.

3. Accessibility

3.1. Open Source

Most of the tools described above are already available as open source distributions inside the CLARIN.SI GitHub organisation.[6] Git has become the most popular platform for (distributed) code development, and GitHub additionally offers a free platform for sharing the code, reporting bugs and requests for improvements, monitoring the activity of a project etc. It, of course, also offers the possibility for third party developers to post improvements to the code with a well-defined procedure for incorporating them into the master branch. For all technologies, we plan in the near future to deposit stable versions of the code to the repository of the Slovene research infrastructure CLARIN.SI,[7] as this frees us from the dependence on a U.S. based repository, and, more importantly, gives additional visibility and citability to the code. Namely, CLARIN.SI has an OAI-PMH endpoint, which enables it to expose the repository metadata to harvesting services, with the repository already being harvested by the European CLARIN Virtual Language Observatory. Furthermore, CLARIN.SI uses the Handle system for persistent identifiers and recommends the correct way of citing its items in publications, thus giving a better chance of acquiring citations for the tools in scientific publications.

3.2. Web application and services

The envisaged architecture of our system that joins the developed language technologies in one ecosystem is presented in Figure 1. There are three approaches to access our technologies: via a command line client, a web interface and a Python library. The latter two approaches access the technologies through the HTTP protocol and have no local requirements besides a browser and a Python interpreter. The command line client is planned for researchers who want to install all the technologies locally as a single package and this component of our system will be finished once all the intended technologies are added to the system. In the remainder of this subsection we describe the two HTTP-based access methods.

Access to both the web interface and to the API via a Python library requires authentication, in order to ensure the stability of the service. To obtain a user name and password one has to register at `http://nl.ijs.si/services/`. This URL is also the entry point to the web application.

3.2.1. Web Application

The technologies currently available through the web application are the lexicon, segmenter, tagger and lemmatiser. A screenshot of the interface to the tagger and lemmatiser is given in Figure 2. The interface enables either writing / pasting text into the form or uploading a text file, choosing the language, defining the input format (either plain text or the text corpus format TCF[8] and choosing the function one wants to run on the input data. Currently the available functions are "Tag", "Lemmatise" and "Tag + Lemmatise". Each of the functions also contains the preprocessing step of segmenting the input on sentences and tokens. For future versions of the technology a higher level of control is planned by allowing building custom pipelines for tasks like tagging already tokenised text, both normalising and tagging text etc.

The result of applying the function on the input data is presented on the right side of the screen in three different modes: as a table, as raw response from the web service and for download. The downloaded file contains either vertical text with tab-separated annotations if the input format was text, or a TCF file if such input format was given.

The main purposes of the presented web application are the following: (1) a first insight in the quality of the output of the language technologies, (2) an insight in the raw response given from the API and (3) a way to process smaller amounts of data, mostly present in form of text files. Uploading, processing and downloading a text file via this web

[6] `https://github.com/clarinsi`
[7] `http://www.clarin.si/`
[8] TCF if an XML-based format used by WebLicht `http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/The_TCF_Format`

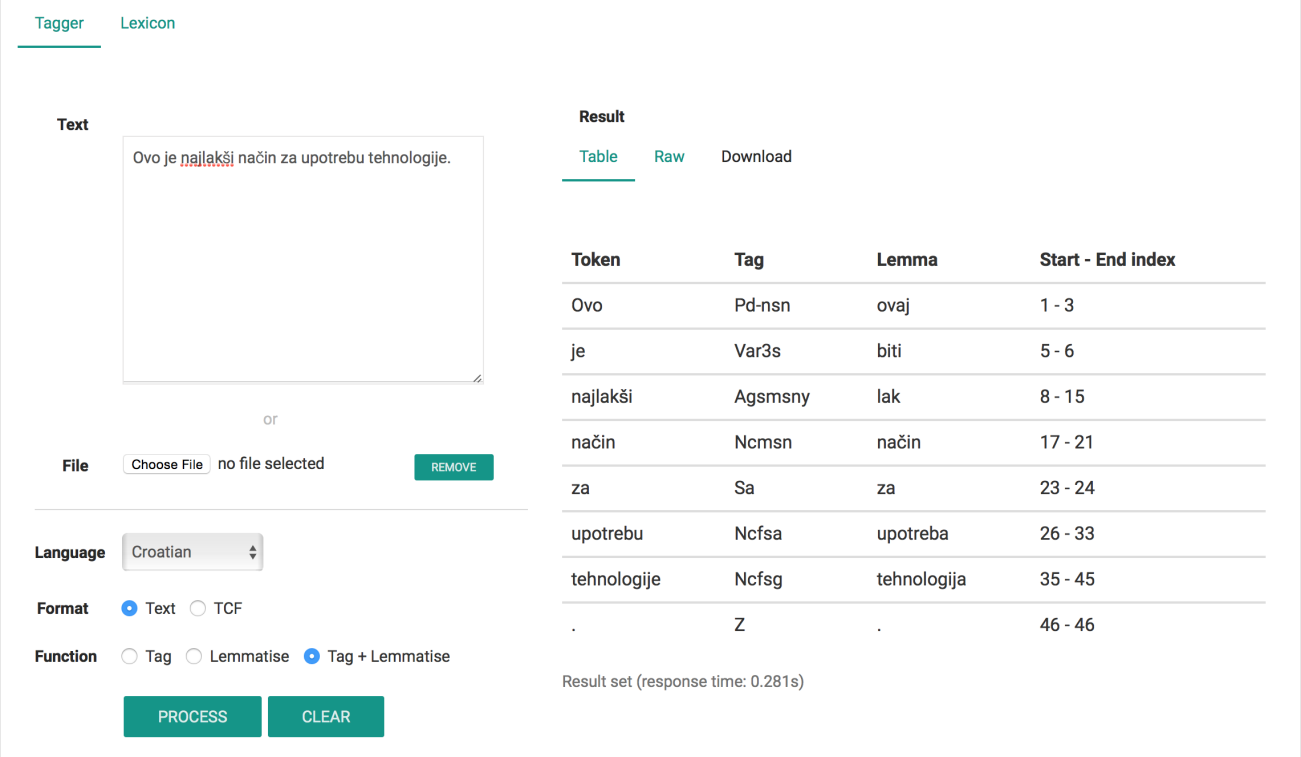

Figure 2: Screenshot of the web interface.

application should not take more than a minute of a user's time.

3.2.2. Web Services

The easiest way to use the language technologies from code is via the Python library which is available via PyPI[9] while the documentation on using the library is available from GitHub.[10] Currently all the developed technologies besides the CSMT text normaliser are available through this library. Here is a code snippet example of using the Python library for segmentation, tagging and lemmatisation:

```
from reldi.tagger import Tagger
from getpass import getpass
username="my_username"
passwd=getpass("Input password: ")
tagger = Tagger("hr")
tagger.authorize(username, passwd)
result=tagger.tagLemmatise("Obradi me.")
```

4. Future Developments

While a lot of work was already put into developing the presented technologies and ensuring their accessibility through a unified ecosystem, a lot is still to be done. Here we present the order of our planned activities.

We plan to develop additional annotation tools, namely a dependency parser and a named entity recogniser. While Slovene and Croatian are part of the Universal Dependencies (UD) project[11] (Nivre et al., 2016), we are working on adding Serbian to its repository by annotating the Serbian dataset corresponding to the Croatian SETimes.HR corpus (Agić and Ljubešić, 2014).

For named entity recognition we have a series of datasets already available and plan on expanding them and develop a CRF-based named entity recogniser.

Once a tool is developed, the procedure we follow is the following: (1) releasing it as open-source via GitHub, (2) including it in our API, (3) ensuring access to the API component through our Python library and (4) making the tool accessible via the web application, which is the final step of exposing a technology as it requires most work, the majority of which is related to the development of the user interface. We are currently working on including the CSMT normalisers of all three languages into the API and training models for UD parsers for Slovene and Croatian.

Once all the technologies have gone through the process of development, inclusion in the API, the Python library and the web application, we will deploy our whole ecosystem as a single package, enabling researchers with large data processing needs to seamlessly install and use the technologies on their own servers.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement PIAP-GA-2012-324414 (Abu-MaTran), the Slovenian Research Agency within the national basic research project "Resources, Tools and Methods for the Research of Nonstandard Internet Slovene" (J6-6842, 2014-2017) and the Swiss National Science Foundation grant no. IZ74Z0_160501 (ReLDI).

[9] https://pypi.python.org/pypi/reldi
[10] https://github.com/clarinsi/reldi-lib
[11] http://universaldependencies.org

5. References

Željko Agić and Nikola Ljubešić. 2014. The SETimes.HR linguistically annotated corpus of Croatian. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Apache Software Foundation. 2014. openNLP Natural Language Processing Library. http://opennlp.apache.org/.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters. 2011. *Text Processing with GATE (Version 6)*.

Kaja Dobrovoljc, Simon Krek, Peter Holozan, Tomaž Erjavec, and Miro Romih. 2015. *Morphological lexicon Sloleks 1.2*. Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1039.

Tomaž Erjavec, Camelia Ignat, Bruno Poliquen, and Ralf Steinberger. 2005. Massive multilingual corpus compilation: Acquis communautaire and totale. In *The 2nd Language & Technology Conference - Human Language Technologies as a Challenge for Computer Science and Linguistics*. Association for Computing Machinery (ACM) and UAM Fundacja.

Miha Grčar, Simon Krek, and Kaja Dobrovoljc. 2012. Obeliks: statistični oblikoskladenjski označevalnik in lematizator za slovenski jezik. In *Proceedings of the Eight Conference on Language Technologies*, Ljubljana, Slovenia.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *In Proc. of the Sixth Workshop on Statistical Machine Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Simon Krek, Kaja Dobrovoljc, Tomaž Erjavec, Sara Može, Nina Ledinek, and Nanika Holz. 2015. Training corpus ssj500k 1.4. Slovenian language resource repository CLARIN.SI.

Nikola Ljubešić and Tomaž Erjavec. 2016. Corpus vs. Lexicon Supervision in Morphosyntactic Tagging: The Case of Slovene. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA).

Nikola Ljubešić, Tomaž Erjavec, and Darja Fišer. 2016a. Corpus-Based Diacritic Restoration for South Slavic Languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA).

Nikola Ljubešić, Filip Klubička, Željko Agić, and Ivo-Pavao Jazbec. 2016b. New Inflectional Lexicons and Training Corpora for Improved Morphosyntactic Annotation of Croatian and Serbian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA).

Nikola Ljubešić. 2016a. *Inflectional lexicon hrLex 1.1*. Slovenian language resource repository CLARIN.SI. `http://hdl.handle.net/1135/1067`.

Nikola Ljubešić. 2016b. *Inflectional lexicon srLex 1.1*. Slovenian language resource repository CLARIN.SI. `http://hdl.handle.net/11356/1066`.

Nikola Ljubešić, Katja Zupan, Darja Fišer, and Tomaž Erjavec. 2016. Normalising Slovene data: historical texts vs. user-generated content. In *Proceedings of KONVENS 2016*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Lluís Padró and Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey, May. ELRA.