# CLASSLA-STANZA: THE NEXT STEP FOR LINGUISTIC PROCESSING OF SOUTH SLAVIC LANGUAGES

Nikola LJUBEŠIĆ,[1] Luka TERČON,[2] Kaja DOBROVOLJC[1,2]
[1]Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[2]Filozofska fakulteta, Univerza v Ljubljani, Aškerčeva cesta 2, 1000 Ljubljana, Slovenia

We present CLASSLA-Stanza, a pipeline for automatic linguistic annotation of South Slavic languages, which is based on the Stanza natural language processing pipeline. We describe the main improvements in CLASSLA-Stanza with respect to Stanza, and give a detailed description of the model training process for the latest 2.1 release of the pipeline. We also report performance scores produced by the pipeline for different languages and varieties. CLASSLA-Stanza exhibits consistently high performance across all the supported languages and outperforms or expands its parent pipeline Stanza at all the supported tasks. We also present the pipeline's new functionality enabling efficient processing of web data and the reasons that led to its implementation.

**Keywords:** South Slavic languages, automatic linguistic processing, annotation pipeline, linguistic annotation

## 1 INTRODUCTION

The South Slavic languages make up one of the three major branches of the Slavic language family. Despite their widespread use, many members of this group remain relatively low-resourced and under-represented in the field of natural language processing. Goldhahn et al. (2016) include Macedonian and Bosnian in their list of languages that are significantly under-resourced despite having more than 1 million speakers.

Although much additional work is required if South Slavic languages are ever to become capable of competing with linguistic giants such as English, steps have already been taken towards establishing common platforms for supporting the development of new resources and tools for these languages. The CLARIN Knowledge centre for South Slavic languages (CLASSLA[1]), was established

[1]https://www.clarin.si/info/k-centre/

as a result of prior cooperation in the development of language resources for Slovenian, Croatian, and Serbian, and currently acts as a platform providing expertise and support for developing language resources for South Slavic languages (Ljubešić et al., 2022). The efforts of the knowledge centre gave rise to the CLASSLA-Stanza[2] pipeline for linguistic processing, which arose as a fork of the Stanza neural pipeline (Qi et al., 2020). CLASSLA-Stanza was created with the aim of providing state-of-the-art automatic linguistic processing for South Slavic languages (Ljubešić & Dobrovoljc, 2019), and currently supports Slovenian, Croatian, Serbian, Macedonian, and Bulgarian. Additionally, Slovenian, Croatian, and Serbian have support for both standard and non-standard, Internet varieties. In comparison to its Stanza parent pipeline, CLASSLA-Stanza expands to cover the standard Macedonian language, as well as the non-standard, Internet varieties of Slovenian, Croatian and Serbian. Beside the expanded coverage of languages and varieties, CLASSLA-Stanza shows improvements in performance on all presented levels.

The aim of this paper is to provide both a systematic overview of the differences that CLASSLA-Stanza has to the official Stanza pipeline and a description of the model training procedure which was adopted when training models for the latest 2.1 release. The description of the training procedure is intended to serve as the main reference for future releases as well as for anyone using the CLASSA-Stanza tool to produce their own models for linguistic annotation.

In accordance with this aim, we first describe the differences between CLASSLA-Stanza and Stanza in section 2. Afterwards, section 3 introduces the datasets used for training the models. Section 4 then gives a general description of the model training process, which is followed by an analysis of the results produced by the newly-trained models in section 5.

At present, the CLASSLA-Stanza annotation tool supports a total of six tasks: tokenization, morphosyntactic annotation, lemmatization, dependency parsing, semantic role labeling, and named-entity recognition. Tokenization is handled by one of two external rule-based tokenizers included in CLASSLA-Stanza, either the Obeliks tokenizer[3] for standard Slovenian (Grčar et al., 2012) or the ReLDI

[2]https://github.com/clarinsi/classla
[3]https://github.com/clarinsi/obeliks

tokenizer[4] for non-standard Slovenian and all other languages (Samardžić et al., 2015). While the basic tasks of tokenization, morphosyntactic annotation, lemmatization and dependency parsing are covered at least for some languages in the parent Stanza pipeline, semantic role labeling and named entity recognition for South Slavic languages are available only in CLASSLA-Stanza.

The current version of the models was trained on data that are annotated accord-ing to three separate systems for morphosyntactic annotation: the universal part-of-speech tags and the universal morphosyntactic features tags—which are both part of the Universal Dependencies framework for grammatical anno-tation (de Marneffe et al., 2021) and will henceforth be referred to as UPOS and UFeats—and the MULTEXT-East V6 specifications for morphosyntactic annota-tion (Erjavec, 2012), which are implemented as the language-specific XPOS tags in the CoNLL-U file format,[5] the central file format used by CLASSLA-Stanza. For dependency parsing, the Universal Dependencies system for syntactic depen-dency annotation was used, as well as the JOS syntactic dependencies system for Slovenian (Erjavec, Fišer, et al., 2010). Additionally, the annotation schema described in Krek et al. (2016) was used for semantic role label annotation.

The outline of the model training process given in this paper describes all six tasks supported by CLASSLA-Stanza, however it must be noted that not all tasks are available for every supported language and variety. Dependency parsing has dedicated models for the standard variety of every language except Macedonian. Named entity recognition is also not supported for Macedonian. Processing of the non-standard variety is available only for Slovenian, Croatian and Serbian, while it is not available for Macedonian and Bulgarian. Semantic role label-ing currently relies on the JOS annotation system for dependency parsing of Slovenian and is thus only available for annotation of Slovenian datasets, but should become available for Croatian in the future, as there are training data available (Ljubešić & Samardžić, 2023). Table 1 provides an overview of every language variety and the tasks it supports.

[4] https://github.com/clarinsi/reldi-tokeniser
[5] https://universaldependencies.org/format.html

Table 1: Tasks supported by CLASSLA-Stanza for every language and variety. The abbreviations for each task are as follows: Tok – tokenization, Morph - morphosyntactic tagging, Lemma - lemmatization, Depparse - dependency parsing, NER - named entity recognition, SRL - semantic role labeling.

| Language | Variety | Tok | Morph | Lemma | Depparse | NER | SRL |
|---|---|---|---|---|---|---|---|
| Slovenian | standard | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | nonstandard | ✓ | ✓ | ✓ | X | ✓ | X |
| Croatian | standard | ✓ | ✓ | ✓ | ✓ | ✓ | X |
| | nonstandard | ✓ | ✓ | ✓ | X | ✓ | X |
| Serbian | standard | ✓ | ✓ | ✓ | ✓ | ✓ | X |
| | nonstandard | ✓ | ✓ | ✓ | X | ✓ | X |
| Bulgarian | standard | ✓ | ✓ | ✓ | ✓ | ✓ | X |
| | nonstandard | X | X | X | X | X | X |
| Macedonian | standard | ✓ | ✓ | ✓ | X | X | X |
| | nonstandard | X | X | X | X | X | X |

## 2 DIFFERENCES BETWEEN CLASSLA-STANZA AND STANZA

The Stanza neural pipeline is centered around a bidirectional long short-term memory (Bi-LSTM) network architecture (Qi et al., 2020). CLASSLA-Stanza largely preserves the design of Stanza, except in some cases, such as tokenization, where a completely different model architecture is used. CLASSLA-Stanza also expands upon the original design with specific additions that help boost model performance for the South Slavic languages. This section thus lists the main differences between the two pipelines, and in the end provides an overview of the difference in the results produced by the models for one of the supported languages.

On the level of tokenization and sentence segmentation, Stanza uses a joint tokenization and sentence segmentation model based on machine learning. We generally view such learned tokenizers as suboptimal, since training data for the two tasks is always limited in size and thus too few tokenization and sentence-splitting phenomena can be learned by the model during the training process. Due to this drawback, CLASSLA-Stanza implements rule-based tokenizers, which handle both the task of tokenization as well as sentence

segmentation. As stated in the introduction, the two tokenizers used are the Obeliks tokenizer for standard Slovenian (Grčar et al., 2012) and the ReLDI tokenizer for non-standard Slovenian and all other languages (Samardžić et al., 2015).[6]

CLASSLA-Stanza also adds support for the use of external inflectional lexicons, which is not present in Stanza. For morphologically rich languages, applying this resource to the annotation process usually significantly increases the performance of the model (Ljubešić & Dobrovoljc, 2019). The South Slavic languages all have quite rich inflectional paradigms, which is why support for inflectional lexicons is present for almost all supported languages in the pipeline.

Most languages support an external lexicon usage only during lemmatization, except for Slovenian, which supports lexicon use also during morphosyntactic tagging. In that case, the lexicon is put into operation during the tag prediction phase, when the model limits the possible predictions to only those tags that are present in the inflectional lexicon for the specific token. Lexicon usage during lemmatization is similar in both Stanza and CLASSLA-Stanza, the main difference being that Stanza builds a lexicon only from the Universal Dependencies training data, while CLASSLA-Stanza additionally exploits an inflectional lexicon. Both Stanza and CLASSLA-Stanza use the lexicon for an initial lemma lookup, and fall back to predicting the lemma only in case that the form with the corresponding tag is not present in the lexicon. One important difference in the lexicon lookup in CLASSLA-Stanza is that the lookup uses XPOS tags that contain the full morphosynctactic information, while Stanza uses the UPOS tag, which is not enough for an accurate lemma lookup in morphologically rich languages.

When training models, Stanza uses a Universal Dependencies dataset as training data for training all the tasks in the pipeline and thus does not enable the user to train models on additional datasets. For certain layers, however, such as lemmatization and morphosyntactic tagging, the South Slavic languages often have more training data available than for dependency parsing, which is exploited by CLASSLA-Stanza. Thus, for example, instead of using only the 210

[6]The Obeliks tokenizer, featuring an extensive set of linguistically informed rules, is the de facto standard for Slovene text tokenization. It has been used in tokenizing the majority of reference Slovene corpora and thus facilitates direct comparisons of newly tokenized data to established corpora.

thousand tokens of data that are used for training the dependency parser, the latest set of standard Croatian models in CLASSLA-Stanza includes mophosyntactic tagging and lemmatization models which were trained on additional 290 thousand tokens, which were manually annotated only on these two levels of annotation.

CLASSLA-Stanza also has a special way of handling "closed-class" words. Closed-class control is a feature of the tokenizers and ensures that punctuation and symbols are assigned appropriate morphosyntactic tags and lemmas. It also restricts the set of possible tokens that can be assigned morphosyntactic tags and lemmas for punctuation and symbols to only those tokens that are defined as such in the tokenizer. In addition to punctuation and symbols, the Slovenian package also includes closed-class control for pronouns, determiners, adpositions, particles, and coordinating and subordinating conjunctions. These additional closed classes are controlled during the morphosyntactic tagging phase using the inflectional lexicon as a reference, disallowing for any token to be labeled with a closed class label if this token was not defined as such in the lexicon.[7]

The Stanza pipeline expects pretrained word embeddings as input. While it uses embedding collections based on Wikipedia data, CLASSLA-Stanza does the extra mile by using the CLARIN.SI embeddings (Terčon et al., 2023; Terčon & Ljubešić, 2023b, 2023b, 2023d, 2023c, 2023a), which are skipgram-based embeddings of 100 dimensions, trained with the fastText tool. These embeddings were primarily prepared for CLASSLA-Stanza, but are useful for other tasks as well. They were trained on multiple times larger text collections than Wikipedia, obtained through web crawling (Bañón et al., 2022), which ensures drastically more diverse word embeddings and thereby also better unseen word handling.

When working with Slovenian, Croatian, or Serbian, the pipeline can be set to any of three predetermined settings, which are used for processing different varieties of the same language. These settings are called *types* and can be either *standard*, *nonstandard*, or *web*. The processing types determine which model is used on every level of annotation (either standard or nonstandard) and are all associated with their respective language varieties: the *standard* type is used

[7] In-depth instructions on how to use the closed-class control functionality are included in the GitHub repository: https://github.com/clarinsi/classla/blob/master/README.closed_classes.md.

for processing standard language, the *nonstandard* type is used for processing nonstandard Internet language, and the *web* type is used for processing texts obtained from the web. The reasons for introducing a separate processing type for web texts are described in section 5.2. Below is an overview showing which model is used on every layer for every type:

Table 2: Overview of processing types in CLASSLA-Stanza.

| **Processing type** | Tokenizer | Morphosyntactic tagger | Lemmatizer | dependency parser |
|---|---|---|---|---|
| **standard** | standard | standard | standard | standard |
| **nonstandard** | nonstandard | nonstandard | nonstandard | standard |
| **web** | standard | nonstandard | nonstandard | standard |

The reason why the nonstandard and the web processing type use the standard dependency parsing model is primarily the lack of training data for training a model beyond standard text. The lack of motivation for building a dataset for parsing non-standard text lies in the fact that the parsing model has upstream lemma and morphosyntactic information at its disposal, therefore requires dedicated training data to a much lesser extent than those upstream processes.

To illustrate the performance of CLASSLA-Stanza, Table 3 provides a comparison of the results produced by both Stanza and CLASSLA-Stanza when generating predictions on the SloBENCH evaluation dataset. SloBENCH[8] (Žitnik & Dragar, 2021) is a platform for benchmarking various natural language processing tasks for Slovenian, which includes also a dataset for evaluating the tasks supported by CLASSLA-Stanza. The performance scores are presented in the form of micro-F1 scores, while the relative error reduction between the scores of the pipelines is presented in percentages.

## 3 DATASETS

The latest models included in the 2.1 release of CLASSLA-Stanza were trained on a variety of datasets in five different languages: Slovenian, Croatian, Serbian, Macedonian, and Bulgarian. Slovenian, Croatian, and Serbian were all associated with two training datasets—one consisting of standard-language texts and one consisting of non-standard texts, while Bulgarian and Macedonian only had a standard-language training dataset available.

[8]https://slobench.cjvt.si/

Table 3: Comparison of performance on the SloBENCH evaluation dataset by both pipelines. Metrics are micro-F1 scores. Downstream tasks use upstream predictions, not gold labels.

| Task | Stanza | CLASSLA-Stanza | Rel. error reduction |
|---|---|---|---|
| Sentence segmentation | 0.819 | 0.997 | 98% |
| Tokenization | 0.998 | 0.999 | 50% |
| Lemmatization | 0.974 | 0.992 | 69% |
| Morphosyntactic tagging - XPOS | 0.951 | 0.983 | 65% |
| Dependency parsing LAS | 0.865 | 0.911 | 34% |

Slovenian standard language models were trained using the SUK training corpus (Arhar Holdt et al., 2022). It contains approximately 1 million tokens of text manually annotated on the levels of tokenization, sentence segmentation, morphosyntactic tagging, and lemmatization. Some subsets also contain syntactic dependency, named entity, multi-word expression, coreference, and semantic role labeling annotations. The corpus is a continuation of the ssj500k Slovene training corpus (Krek et al., 2021). Non-standard models were trained on a combination of the standard training corpus and the non-standard Janes-Tag training corpus (Lenardič et al., 2022), which consists of tweets, blogs, forums, and news comments, and is approximately 218 thousand tokens in size. It contains manually curated annotations on the levels of tokenization, sentence segmentation, word normalization, morphosyntactic tagging, lemmatization, and named entity annotation.

Croatian standard language models were trained on the hr500k training corpus (Ljubešić & Samardžić, 2023), which consists of about 500 thousand tokens and is manually annotated on the levels of tokenization, sentence segmentation, morphosyntactic tagging, lemmatization, and named entities. Portions of the corpus also contain manual syntactic dependency, multi-word expression, and semantic role labeling annotations. Croatian non-standard models were trained on a combination of the standard training corpus and the non-standard ReLDI-NormTagNER-hr training corpus (Ljubešić et al., 2023a). The ReLDI-NormTagNER-hr corpus contains about 90 thousand tokens of non-standard Croatian text from tweets and is manually annotated on the levels of tokenization, sentence segmentation, word normalization, morphosyntactic tagging, lemmatization, and named entity recognition.

Serbian standard models were trained on the Serbian portion of the SETimes corpus (Batanović et al., 2023), which contains about 97 thousand tokens of news articles manually annotated on the levels of tokenization, sentence segmentation, morphosyntactic tagging, lemmatization, and dependency parsing. Serbian non-standard models were trained, similar to the previously introduced languages, on a combination of the standard dataset and the non-standard ReLDI-NormTagNER-sr training corpus (Ljubešić et al., 2023b). ReLDI-NormTagNER-sr consists of about 90 thousand tokens of Serbian tweets manually annotated on the levels of tokenization, sentence segmentation, word normalization, morphosyntactic tagging, lemmatization, and named entity recognition.

Macedonian standard models were trained on a corpus made up of the Macedonian version of the MULTEXT-East "1984" corpus (Erjavec et al., 2010) and the Macedonian SETimes.MK corpus. The MULTEXT-East "1984" corpus consists of the novel *1984* by George Orwell in approximately 113 thousand tokens, while the SETimes.MK corpus in its 0.1 version is made up of 13,310 to- kens of news articles (Ljubešić & Stojanovska, 2023). At the time of writing this paper, only the SETimes.MK corpus has been made publicly available, while the "1984" corpus is still awaiting to being published by its authors. Both corpora are manually annotated on the levels of tokenization, sentence segmentation, morphosyntactic tagging, and lemmatization. The combining of the corpus was performed in the following way: the 1984 corpus was first split into three parts to obtain the training, validation and testing data splits, after which only the training data split was enriched with three repetitions of the SETimes corpus to ensure a sensible combination of literary and newspaper data in the training subset.

Bulgarian standard models were trained on the BulTreeBank training corpus (Osenova & Simov, 2015), which consists of approximately 253 thousand tokens manually annotated on the levels of tokenization, sentence segmentation, morphosyntactic tagging, and lemmatization. About 60% of the dataset also contains manual dependency parsing annotations.

Table 4 provides an overview of dataset sizes for every language, variety, and annotation layer.

Table 4: Overview of the number of tokens annotated on every annotation layer for all training datasets used. The abbreviations for each task are as follows: Morph -morphosyntactic tagging, Lemma - lemmatization, Depparse - dependency parsing, SRL - semantic role labeling.

| Language | Variety | Morph | Lemma | Depparse | SRL |
|---|---|---|---|---|---|
| Slovenian | standard | 1,025,639 | 1,025,639 | 267,097 | 209,791 |
| | nonstandard | 222,132 | 222,132 | n/a | n/a |
| Croatian | standard | 499,635 | 499,635 | 199,409 | n/a |
| | nonstandard | 89,855 | 89,855 | n/a | n/a |
| Serbian | standard | 97,673 | 97,673 | 97,673 | n/a |
| | nonstandard | 92,271 | 92,271 | n/a | n/a |
| Bulgarian | standard | 253,018 | 253,018 | 156,149 | n/a |
| Macedonian | standard | 153,091 | 153,091 | n/a | n/a |

## 4 MODEL TRAINING PROCESS

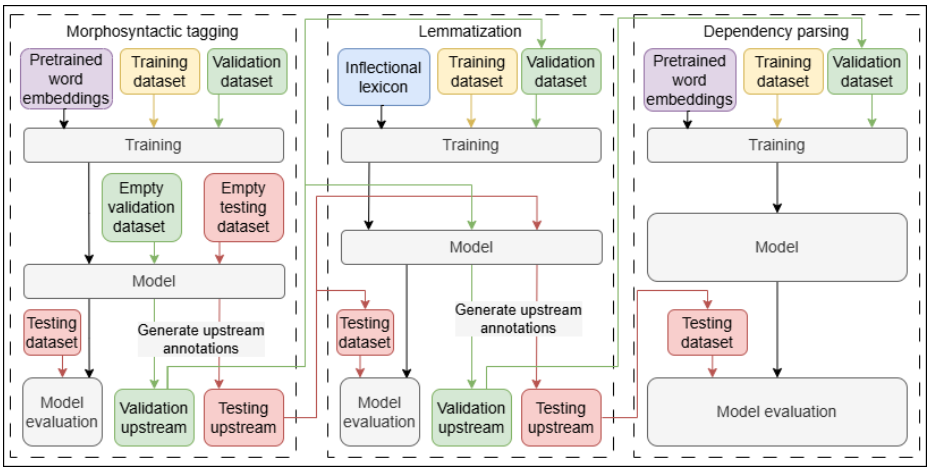
In this section, the model training process is described in detail. Only a descriptive account of the process is provided here. For a list of the specific commands and oversampling scripts used, refer to the GitHub repository of the training procedure.[9]

In this paper we give the general overview of the process which is common to all supported languages. For the specific steps that are unique to each language, please refer to the CLASSLA-Stanza technical report, a longer and older version of this paper available on arXiv (Terčon & Ljubešić, 2023). The language-specific steps were necessary due to some features and levels of annotation (semantic role labeling, oversampling of the training data, etc.) being unique to only certain languages, while all languages at the same time share the common steps described below.

An illustration of the basic procedure that was used to train standard models for the levels of morphosyntactic tagging, lemmatization, and dependency parsing for the latest release of CLASSLA-Stanza is shown in Figure 1.

[9] https://github.com/clarinsi/classla-training

Figure 1: Diagram of the basic model training process for standard morphosyntactic tagging, lemmatization and dependency parsing models.




As stated in the introduction, all tokenizers used by CLASSLA-Stanza are rule-based and thus do not need to be trained. Model training is thus performed on pretokenized data, typically beginning on the level of morphosyntactic tagging and continuing on through the subsequent annotation layers.

To ensure realistic evaluation results, automatically generated upstream annotations, rather than manually assigned annotations, were used as validation and test dataset inputs on each layer. For this, empty validation and test datasets first had to be generated by stripping all annotations from the test and validation datasets on all levels except for tokenization. These empty files were filled with model-generated annotations on each level, so that validation and model evaluation on subsequent layers could be performed on automatically generated upstream labels. Training datasets were not annotated with automatically generated upstream labels, since it is unclear whether this would lead to any performance gains and would require a more complicated type of cross-validation method such as jackknifing (splitting data into N bins, training a model on N-1 bins and annotating the N-th bin, repeating the process N times).

For each language, standard models were first trained. For morphosyntactic tagging training, the training and validation datasets from the prepared three-

way data split along with the pretrained word embeddings were used as inputs to the tagger module. After training, the tagger was used in predict mode to generate predictions on the empty test dataset and evaluate the performance of the tagger. After predictions were made for the test set, predictions were generated in the empty validation dataset as well, so as to produce a validation file with automatically generated morphosyntactic labels, that can be used later during training of subsequent annotation layer models, such as those for lemmatization and dependency parsing.

Once morphosyntactic predictions and evaluation results were obtained, the lemmatizer was trained. The validation and training datasets were used as inputs. In addition, for most languages, the inflectional lexicon is also provided to the lemmatizer as one of the inputs. During training, the lexicon is stored in the lemmatization model file to act as an additional controlling element during lemmatization. After training, the lemmatizer was run in predict mode to obtain evaluation results and add lemma predictions to the validation and test datasets for the training of the dependency layer model.

The dependency parser module was trained after lemmatizer training was finalized. CLASSLA-Stanza currently supports two types of annotation systems for syntactic dependency annotation: the UD dependency parsing annotation system, which is available for all supported languages except Macedonian, and the JOS parsing system, which is only available for Slovenian.[10] The parser was run in training mode using the training and validation datasets[11] as inputs along with the pretrained word embeddings. After training, the parser was run in predict mode to obtain evaluation results.

For this latest release of CLASSLA-Stanza, no new models for named entity recognition were trained. However, the process for training models for named-entity recognition is quite similar to the other tasks. The tagger trainer for this task accepts pretrained word embeddings and training and validation datasets

[10] In comparison to UD, JOS parsing system features a more concise set of dependency relations focusing on core syntactic constructs, and has thus been preferred over UD in some specific applications.

[11] For most languages, only a portion of the original datasets contained dependency parsing annotations. In these cases, a separate set of training, validation, and test datasets consisting of only this portion of the original data had to be extracted.

as inputs. After training, the named entity recognition tagger can be run in predict mode to obtain evaluation results.

The non-standard models were trained using the same process as the standard models, with a few exceptions. Firstly, no syntactic dependency annotations are present in the non-standard datasets. As a result, no non-standard dependency parsing models were trained.

Before training the non-standard models, approximately 20% of diacritics were removed from the training datasets in order to ensure that the models will learn to effectively handle dediacritized forms, which occur prominently in online communication.

It is important to note that the non-standard models were regularly trained on a mixture of standard and non-standard data for best possible performance, while still informing models of non-standard linguistic features. For that reason, non-standard training data were regularly oversampled so that their combination with the standard data would not make non-standard data much less represented, which would hinder learning the non-standard linguistic features.

## 5 MODEL PERFORMANCE ANALYSIS

We know, as noted in section 2, that CLASSLA-Stanza significantly outperforms Stanza on the Slovenian benchmark, with error reduction between 34% and 98%, depending on the processing layer.

However, in order to fully assess the performance of the newly-trained models, we perform in this section a series of additional performance analyses. In Section 5.1 we give a detailed rundown of the performance of the models for each UPOS and each UD label for each language. In Section 5.2 we continue with a more qualitative investigation of the performance of the models on web-specific data.

### 5.1 Model performance on UPOS and UD labels

To obtain a sense of which specific categories a model struggles with and which ones it handles with particular ease, model predictions for specific UPOS and UD syntactic relations were inspected. An accuracy score was calculated for all

17 UPOS labels and the 12 most frequent UD syntactic relations in the Croatian hr500k training corpus. The accuracy score was obtained by taking the number of correct predictions for a single label in the test dataset and dividing it by the total number of occurrences of that label in the test dataset. The resulting accuracies for all the UPOS tags are contained in Table 5, while Table 6 contains accuracies for each UD dependency relation.

Table 5: Table of per-relation accuracies for all UPOS tags. The language abbreviations are followed by either “st” for *standard* or “nonst” for *non-standard*.

| UPOS tag | Accuracy | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | sl-st | sl-nonst | hr-st | hr-nonst | sr-st | sr-nonst | mk-st | bg-st | Average |
| ADJ | 99.31 | 90.71 | 97.93 | 92.27 | 99.27 | 94.58 | 97.74 | 98.28 | 96.26 |
| ADP | 99.90 | 98.54 | 99.96 | 99.82 | 100.00 | 99.84 | 99.75 | 99.92 | 99.72 |
| ADV | 95.98 | 91.89 | 95.35 | 91.59 | 95.42 | 87.93 | 95.14 | 97.60 | 93.86 |
| AUX | 98.62 | 96.31 | 99.60 | 99.59 | 100.00 | 98.81 | 99.50 | 92.75 | 98.15 |
| CCONJ | 98.01 | 97.03 | 96.53 | 97.21 | 98.95 | 97.21 | 97.94 | 97.87 | 97.59 |
| DET | 99.29 | 93.29 | 95.68 | 94.08 | 98.88 | 96.74 | 100.00 | 87.79 | 95.72 |
| INTJ | 80.00 | 75.82 | 71.43 | 90.22 | n/a | 87.65 | 71.43 | 47.58 | 74.88 |
| NOUN | 98.88 | 93.75 | 98.33 | 93.98 | 99.23 | 97.66 | 99.55 | 98.53 | 97.49 |
| NUM | 99.74 | 98.41 | 98.87 | 100.00 | 98.71 | 100.00 | 100.00 | 98.17 | 99.24 |
| PART | 99.46 | 95.12 | 85.16 | 90.64 | 94.12 | 89.39 | 90.16 | 79.94 | 90.50 |
| PRON | 99.47 | 97.25 | 98.68 | 98.19 | 97.64 | 98.47 | 98.84 | 99.15 | 98.46 |
| PROPN | 98.71 | 78.23 | 93.65 | 77.81 | 97.31 | 83.68 | 97.97 | 98.14 | 90.69 |
| PUNCT | 100.00 | 99.79 | 100.00 | 99.73 | 100.00 | 99.82 | 100.00 | 100.00 | 99.92 |
| SCONJ | 99.78 | 97.99 | 95.72 | 94.79 | 99.52 | 98.25 | 94.70 | 99.61 | 97.55 |
| SYM | 100.00 | 99.85 | 90.91 | 99.10 | 100.00 | 99.38 | n/a | n/a | 98.21 |
| VERB | 97.05 | 94.12 | 99.30 | 97.84 | 99.18 | 98.76 | 99.74 | 96.79 | 97.85 |
| X | 59.13 | 75.67 | 77.15 | 80.10 | 43.33 | 62.86 | n/a | 0.00 | 56.89 |

The highest accuracies among UPOS tags are generally found with tags that represent function word classes, such as **AUX** (auxiliaries), **ADP** (adpositions), and **PRON** (pronouns), and closed-class tags, such as **PUNCT** (punctuation) and **SYM** (symbols), which are handled by the pipeline, inter alia, through rules in the tokenizer, as described in section 2. Conversely, the lowest accuracies are found with the infrequent **INTJ** tag (interjections)—of which there were only 5 instances in total in the Slovenian standard test dataset and no instances at all in the Serbian standard test dataset—and the loosely delineated **X** tag, which is used for abbreviations, URLs, foreign language tokens, and everything else that does not fit into any of the other categories.

Table 6: Table of per-relation accuracies for all UD relations.

| UD relation | Accuracy | | | | |
|---|---|---|---|---|---|
| | sl | hr | sr | bg | Average |
| punct | 100.00 | 100.00 | 100.00 | 99.91 | 99.98 |
| amod | 98.61 | 95.97 | 97.38 | 98.66 | 97.66 |
| case | 99.63 | 99.32 | 99.21 | 99.86 | 99.51 |
| nmod | 92.74 | 91.22 | 90.99 | 91.49 | 91.61 |
| nsubj | 90.49 | 93.39 | 94.30 | 91.10 | 92.32 |
| obl | 91.99 | 85.31 | 87.24 | 77.17 | 85.43 |
| conj | 92.51 | 90.92 | 93.06 | 93.95 | 92.61 |
| root | 93.14 | 94.98 | 95.77 | 95.97 | 94.97 |
| obj | 93.33 | 82.84 | 91.39 | 90.18 | 89.44 |
| aux | 99.48 | 97.88 | 97.57 | 90.46 | 96.35 |
| cc | 97.83 | 97.63 | 97.96 | 99.14 | 98.14 |
| advmod | 96.74 | 93.58 | 91.82 | 97.91 | 95.01 |

A similar trend is found among the UD syntactic relations. Relations such as **case** (which usually connects nominal heads with adpositions), **cc** (connects conjunct heads with coordinating conjunctions), and **aux** (connects verbal heads with auxiliary verbs) are used for fixed grammatical patterns that permit little variation. These display consistently high accuracies across all languages. Somewhat lower accuracies are displayed by the **obl** relation, mostly used for oblique nominal arguments, which play a less central role in the sentence structure than the core verbal arguments. It has been found that previous versions of dependency parsing models for CLASSLA-Stanza often incorrectly assigned the **obj** relation (used for direct objects) to instances which should receive the **obl** relation and vice versa (Dobrovoljc et al., 2022). Upon inspection of the outputs produced by the newly-trained Slovenian and Croatian parsers it was found that this error persists also in the current version, which is also a likely reason for the performance drops of the **obl** and **obj** relations in other languages as well.

### 5.2 Model performance on Web Data

The model evaluations described in the previous subsection provide a good summary of how well the CLASSLA-Stanza pipeline performs on both purely standard and purely non-standard data. However, modern corpus construction techniques—especially for low-resource languages—often rely on crawling data from online conversations, articles, blogs, etc. (Goldhahn et al., 2016), which typically consists of a mixture of different language styles and varieties. To illustrate how well the new CLASSLA-Stanza models handle language originating from the internet, this section provides a brief manual qualitative analysis of their performance on a corpus of web data.

The CLASSLA-Stanza tool was used with the newly-trained models to add linguistic annotations to the CLASSLA-web corpora, which consist of texts crawled from the internet domains of the corresponding languages (Ban et al., 2023b, 2023a). In preparation for the annotation process, a short test was conducted with the goal of determining which of the two sets of models—the standard or the non-standard—is best suited to be used for annotating the CLASSLA-web corpora. Shorter portions of the corpora were annotated on the levels of tokenization, sentence segmentation, morphosyntactic tagging and lemmatization, once using the standard and once using the non-standard models. The two outputs were then compared and a qualitative analysis of the differences was conducted.

Quite a few of the analyzed differences in the model outputs were connected to the processes of sentence segmentation and tokenization. In the CLASSLA-Stanza annotation pipeline, both of these processes are controlled by the tokenizer. As stated in section 2, the pipeline uses two different tokenizers depending on the language and the annotation type used. [12] The analysis showed that sentence segmentation was performed much more accurately by Obeliks and the standard mode of the ReLDI tokenizer. The non-standard mode of the ReLDI tokenizer appears to have a tendency towards producing shorter segments, since it is optimized for processing social media texts such as tweets. Thus, the non-standard tokenizer very consistently produces a new sentence after periods, question marks, exclamation marks, and other punctuation, even when

[12] The ReLDI tokenizer can be used in two different settings: standard and non-standard. The Obeliks tokenizer, on the other hand, only supports tokenization of standard text.

these characters do not signify the end of a segment. The following Croatian example in a simplified CoNLL-U format shows one such case of incorrect sentence segmentation, due to the use of reported speech. The original string *„ Svaku našu riječ treba da čuvamo kao najveće blago.“* was split into two segments - the first ending on the period character, while the quotation mark was moved to a separate sentence:

```
# newpar id = 76
# sent_id = 76.1
# text = „ Svaku našu riječ treba da čuvamo kao
najveće blago.
1 „
2 Svaku
3 našu
4 riječ
5 treba
6 da
7 čuvamo
8 kao
9 najveće
10 blago
11 .

# sent_id = 76.2
# text = “
1 “
```

Besides sentence segmentation issues, the standard models also performed better than the non-standard models when assigning certain types of grammatical features, such as with disambiguating between the UD part-of-speech labels AUX and VERB for the verb *biti* (Eng. “to be”). However, the difference between the two model outputs for these grammatical features was not as noticeable as on the levels of tokenization and sentence segmentation.

The non-standard models, on the other hand, handled non-standard word forms quite a bit better than the standard models. Particularly problematic for the standard Slovenian models were forms with missing diacritics, such as “sel”

instead of *šel*, “cist” instead of *čisto*, “hoce” instead of *hoče*, and “clovek” instead of *človek*. These were often assigned incorrect lemmas and morphosyntactic tags. An example of the standard lemmatizer output for the word form “hoce” (which corresponds to *hoče* in standard Slovene (Eng. “he/she/it wants”)) is displayed below. The model invents a nonexistent lemma “hocati”, while the correct form should be the standard Slovenian *hoteti*:

```
# sent_id = 53.1
# text = lev je lev pa naj govori kar kdo hoce
1 lev lev
2 je biti
3 lev lev
4 pa pa
5 naj naj
6 govori govoriti
7 kar kar
8 kdo kdo
9 hoce hocati
```

Non-standard forms which do not differ much from their standard counterparts, such as “zdej” as opposed to “zdaj” and “morš” as opposed to “moraš”, were generally handled well by both sets of models and did not cause many discrepancies in the outputs.

The analysis of such differences in the model outputs showed that the best results for the web corpus were achieved on the one hand by the standard tokenizer, and on the other by the non-standard models for all subsequent levels of annotation. In light of this, a new *web* type was implemented for the CLASSLA-Stanza pipeline. This new type combines the standard tokenizer and non-standard models for the other layers in a single package and is intended specifically for the annotation of texts originating on the Internet.

## 6 CONCLUSION

In this paper, we provided an overview of the CLASSLA-Stanza pipeline for linguistic processing of the South Slavic languages and described the training

process for the models included in the latest release of the pipeline. We described the main design differences to the Stanza neural pipeline, from which CLASSLA-Stanza arose as a forked project. We provided a summary of the model training process, while for a more detailed description of the training process for each language the technical documentation (Terčon & Ljubešić, 2023) should be consulted. We also presented per-label performance scores for UPOS labels from standard and non-standard models, and most frequent UD labels from standard models.

CLASSLA-Stanza gives consistent results across all supported languages and outperforms the Stanza pipeline on all supported NLP tasks, as illustrated in sections 2 and 4. However, overall low accuracies are still seen for infrequent labels and pairs of labels that are not so easily disambiguated. It remains to be seen whether larger and more diverse training datasets can contribute to improving model performance in these specific cases, or rather the move to contextual embeddings, i.e., transformer models. Additionally, when processing texts obtained from the Internet, special care must be taken to use the combination of models that is best suited for the task, which is why we also described the special *web* processing type implemented within CLASSLA-Stanza.

The release of a specialized pipeline for linguistic processing of South Slavic languages is an important new milestone in the development of digital resources and tools for this relatively under-resourced group of languages. However there is still much left to be achieved and improved upon. Full support for all annotation tasks, such as, for instance, semantic role labeling, which is currently only available for Slovenian, remains to be extended to other languages as well. As larger training datasets become available, more capable models can be trained for the currently supported languages. In addition, the aim is also to extend support to other members of the South Slavic language group, provided that training datasets of sufficient size are eventually produced for those languages as well. Finally, the performance of the CLASSLA-Stanza pipeline should also be compared to other recent state-of-the-art tools for automatic linguistic annotation, such as Trankit (Nguyen et al., 2021), which was shown to outperform Stanza over a large number of languages and datasets.

## 7 ACKNOWLEDGMENTS

The work described by this paper was made possible by the Development of Slovene in a Digital Environment project (Razvoj slovenščine v digitalnem okolju, project ID: C3340-20-278001), financed by the Ministry of Culture of the Republic of Slovenia and the European Regional Development Fund, the Language Resources and Technologies for Slovene research program (project ID: P6-0411), financed by the Slovenian Research Agency, the MEZZANINE project (Basic Research for the Development of Spoken Language Resources and Speech Technologies for the Slovenian Language, project ID: J7-4642) and the SPOT project (A Treebank-Driven Approach to the Study of Spoken Slovenian, Z6-4617), financed by the Slovenian Research Agency, and the CLARIN.SI research infrastructure.

## REFERENCES

Arhar Holdt, Š., Krek, S., Dobrovoljc, K., Erjavec, T., Gantar, P., Čibej, J, ... Zajc, A. (2022). *Training corpus SUK 1.0.* Slovenian language resource repository CLARIN.SI http://hdl.handle.net/11356/1747

Bañón, M., Chichirau, M., Esplà-Gomis, M., Forcada, M. L., Galiano-Jiménez, A., García-Romero, C., Kuzman, T., Ljubešić, N., van Noord, R., Pla Sempere, L., Ramírez-Sánchez, G., Rupnik, P., Suchomel, V., Toral, A., & Zaragoza-Bernabeu, J. (2023a). *Croatian web corpus MaCoCu-hr 2.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1806

Bañ, M., Chichirau, M., Espl-Gomis, M., Forcada, M. L., Galiano-Jimnez, A., Garca-Romero, C., Kuzman, T., Ljubešić, N., van Noord, R., Pla Sempere, L., Ramírez-Sánchez, G., Rupnik, P., Suchomel, V., Toral, A., & Zaragoza-Bernabeu, J. (2023b). *Slovene web corpus MaCoCu-sl 2.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1795

Bañón, M., Esplà-Gomis, M., Forcada, M., García-Romero, C., Kuzman, T., Ljubešić, N., van Noord, R., Pla Sempere, L., Ramírez-Sánchez, G., Rupnik, P., Suchomel, V., Toral, A., van der Werff, T., & Zaragoza, J. (2022). MaCoCu: Massive collection and curation of monolingual and bilingual data: focus on under-resourced languages. In L. Macken et al. (Eds.), *Eamt 2022 - proceedings of the 23rd annual conference of the european association for machine translation* (pp. 303–304). European Association for Machine Translation.

Batanović, V., Ljubešić, N., Samardžić, T., & Erjavec, T. (2023). *Serbian linguistic training corpus SETimes.SR 2.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1843

de Marneffe, M.-C., Manning, C. D., Nivre, J., & Zeman, D. (2021). Universal Dependencies. *Computational Linguistics*, *47*(2), 255-308. https://doi.org/10

.1162/coli_a_00402 doi: doi:10.1162/coli_a_00402

Dobrovoljc, K., Terčon, L., & Ljubešić, N. (2022). Universal Dependencies za slovenščino: nadgradnja smernic, učnih podatkov in razčlenjevalnega modela [Universal Dependencies for Slovenian: An Upgrade to the Guidelines, Annotated Data and Parsing Model]. In D. Fišer & T. Erjavec (Eds.), *Jezikovne tehnologije in digitalna humanistika: zbornik konference [language technology and digital humanities: Conference proceedings]* (pp.33–39). Inštitut za novejšo zgodovino [Institute of Contemporary History]. https://nl.ijs.si/jtdh22/pdf/JTDH2022_Proceedings.pdf

Erjavec, T. (2012). MULTEXT-East: morphosyntactic resources for Central and Eastern European languages. *Language Resources and Evaluation*, *46*(1), 131–142. Retrieved 2023-06-21, from http://www.jstor.org/stable/41486069

Erjavec, T., Barbu, A.-M., Derzhanski, I., Dimitrova, L., Garabík, R., Ide, N., Haalep, H., Kotsyba, N., Krstev, N., Oravecz, C., Petkevič, V., Priest-Dorman, G., QasemiZadeh, B., Radziszewski, A., Simov, K., Tufis, D., & Zdravkova, K. (2010).

*MULTEXT-East "1984" annotated corpus 4.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/ 11356/1043

Erjavec, T., Fišer, D., Krek, S., & Ledinek, N. (2010). The JOS Linguistically Tagged Corpus of Slovene. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10).* European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2010/pdf/139_Paper.pdf

Goldhahn, D., Janicki, M., & Quasthoff, U. (2016). Corpus collection for under-resourced languages with more than one million speakers. In *Workshop on Collaboration and Computing for Under-Resourced Languages (CCURL), LREC.*

Grčar, M., Krek, S., & Dobrovoljc, K. (2012). Obeliks: statisticni oblikoskladenjski oznacevalnik in lematizator za slovenski jezik [Obeliks: Statistical Morphosyntactic Tagger and Lemmatizer for Slovene]. In J. PBI G. T. Erjavec (Ed.), *Proceedings of the eighth language technologies conference.* Jožef Stefan Institute.

Krek, S., Dobrovoljc, K., Erjavec, T., Može, S., Ledinek, N., Holz, N., Zupan, K., Gantar, P., Kuzman, T., Čibej, J., Arhar Holdt, Š., Kavčič, T., Škrjanec, I., Dafne, M., Jezeršek, L., & Zajc, A. (2021). *Train- ing corpus ssj500k 2.3.* Slovenian language resource repository CLARIN.SI http://hdl.handle.net/11356/1434

Krek, S., Gantar, P., Dobrovoljc, K., & Škrjanec, I. (2016). Označevanje udeleženskih vlog v učnem korpusu za slovenščino [Semantic Role Label- ing in the Training Corpus for Slovene]. In (pp. 106–110). Znanstvena založba Filozofske fakultete [Ljubljana University Press, Faculty of Arts]. Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1732

Dobrovoljc, K. (2022). *CMC training corpus Janes-Tag 3.0.* http://hdl.handle.net/11356/1732

Ljubešić, N., & Dobrovoljc, K. (2019). What does Neural Bring? Analysing Improvements in Morphosyntactic Annotation and Lemmatisation of Slovenian, Croatian and Serbian. In *Proceedings of the 7th workshop on balto-slavic natural language processing* (pp. 29–34). Association for Computational Linguistics. https://aclanthology.org/W19-3704

Ljubešić, N., Erjavec, T., Batanović, V., Miličević, M., & Samardžić, T. (2023a). *Croatian Twitter training corpus ReLDI-NormTagNER-hr 3.0.* http://hdl.handle.net/11356/ 1793 (Slovenian language resource repository CLARIN.SI)

Ljubešić, N., Erjavec, T., Batanović, V., Miličević, M., & Samardžić, T. (2023b). *Serbian Twitter training corpus ReLDI-NormTagNER-sr 3.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/ 1794

Ljubešić, N., & Samardžić, T. (2023). *Croatian linguistic training corpus hr500k 2.0.* http://hdl.handle.net/11356/1792 (Slovenian language resource repository CLARIN.SI)

Ljubešić, N., & Stojanovska, B. (2023). *Macedonian linguistic training corpus SETimes.MK 0.1.* http://hdl.handle.net/11356/1886 (Slovenian language resource repository CLARIN.SI)

Ljubešić, N., Erjavec, T., Petrović, M. M., & Samardžić, T. (2022). Together We Are Stronger: Bootstrapping Language Technology Infrastructure for South Slavic Languages with CLARIN.SI. In D. Fišer & A. Witt (Eds.), *Clarin. the infrastructure for language resources.* (pp. 429–456). Berlin, Boston: De Gruyter. Retrieved 2023-06-21, from https://doi.org/10.1515/9783110767377-017

Nguyen, M. V., Lai, V., Veyseh, A. P. B., & Nguyen, T. H. (2021). Trankit: A Light-Weight Transformer-based Toolkit for Multilingual Natural Language Processing. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: System demonstrations.*

Osenova, P., & Simov, K. (2015). Universalizing BulTreeBank: a Linguistic Tale about Glocalization. In The 5th workshop on Balto-Slavic natural language processing (pp. 81–89). INCOMA Ltd. Shoumen, BULGARIA. https://aclanthology.org/W15-5313

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th annual meeting of the association for computational linguistics: System demonstrations.* https://nlp.stanford.edu/pubs/qi2020stanza.pdf

Samardžić, T., Ljubešić, N., & Miličević, M. (2015). Regional Linguistic Data Initiative (ReLDI). In *The 5th workshop on Balto-Slavic natural language processing* (pp. 40–42). INCOMA Ltd. Shoumen. https://aclanthology.org/W15-5306

Terčon, L., & Ljubešić, N. (2023a). *Word embeddings CLARIN.SI-embed.bg 1.0.* http://hdl.handle.net/11356/1796 (Slovenian language resource repository CLARIN.SI)

Terčon, L., & Ljubešić, N. (2023b). *Word embeddings CLARIN.SI-embed.hr 2.0.* http://hdl.handle.net/11356/1790 (Slovenian language resource repository CLARIN.SI)

Terčon, L., & Ljubešić, N. (2023c). *Word embeddings CLARIN.SI-embed.mk 2.0.* http://hdl.handle.net/11356/1788 (Slovenian language resource repository CLARIN.SI)

Terčon, L., & Ljubešić, N. (2023d). *Word embeddings CLARIN.SI-embed.sr 2.0.* http://hdl.handle.net/11356/1789 (Slovenian language resource repository CLARIN.SI)

Terčon, L., Ljubešić, N., & Erjavec, T. (2023). *Word embeddings CLARIN.SI-embed.sl 2.0.* http://hdl.handle.net/11356/1791 (Slovenian language resource repository CLARIN.SI)

Terčon, L., & Ljubešić, N. (2023). *CLASSLA-Stanza: The Next Step for Linguistic Processing of South Slavic Languages.* https://doi.org/10.48550/arXiv.2308.04255

Žitnik, S., & Dragar, F. (2021). *SloBENCH evaluation framework.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1469

# CLASSLA-STANZA: NASLEDNJI KORAK ZA JEZIKOVNO PROCESIRANJE JUŽNOSLOVANSKIH JEZIKOV

V članku predstavljamo orodje CLASSLA-Stanza, cevovod za avtomatsko jezikovno označevanje južnoslovanskih jezikov, ki temelji na cevovodu za procesiranje naravnega jezika Stanza. Opisujemo vse glavne izboljšave, ki jih prinaša CLASSLA-Stanza v primerjavi s Stanzo in podamo podroben opis postopka učenja modelov v različici 2.1, najnovejši različici orodja. Obenem poročamo o rezultatih delovanja cevovoda za različne jezike in jezikovne zvrsti. CLASSLA-Stanza dosega konsistentno visoke rezultate za vse podprte jezike in preseže rezultate izvornega cevovoda Stanza pri vseh podprtih nalogah. Predstavimo tudi novo funkcijo cevovoda, ki omogoča učinkovito procesiranje spletnih besedil, in razloge za njeno implementacijo.

**Keywords:** južnoslovanski jeziki, avtomatsko procesiranje jezika, označevalni cevovod, jezikovno označevanje