# Final Project: Training and Evaluating a NER model for the Serbian Language in spaCy using an Old Serbian Novel

**Milan Miletić**
University of Tübingen
Computational Models for Named Entity Recognition, WS20/21
milan.miletic@student.uni-tuebingen.de

## Abstract

The following paper is a supplement to my final project submission for the course Computational Models for Named Entity Recognition, held at the University of Tübingen in the winter semester 2020/21. It starts with describing the process of creating annotated data using the predefined annotation guidelines. The annotated data is then used for training a NER model in Python with the help of the spaCy library. The final and perhaps the most interesting part of this paper deals with the evaluation of the model using both standard and novel approaches. The project was realized in cooperation with the *Serbian Association for Language Resources and Technologies (JeRTeh)* [1].

## 1 Overview

*The Serbian Association for Language Resources and Technologies* is an active participant in numerous COST[2] actions. A relatively recent one is the *CA16204 - Distant Reading for European Literary History*[3] which has, as the main objective, collecting and annotating a corpus of old novels (prior to 1920s) from more than 10 European languages. The task that I had been given was to, first of all, annotate one such novel with named entities - *Hadži-Đera*, by Dragutin Ilić (1904). The annotation was carried out through the BRAT[4] platform, according to a predefined annotation scheme, which is explained in detail in Section 2. After successful annotation and necessary corrections, I exploited the functionalities of the spaCy[5] module to train a NER model on this data. Further notes on this can be found in Section 3. Lastly, I used different methods for evaluating the best model and dedicated Section 4 for reporting and discussing the results. Additionally, I compared the performance of my model to that of a pretrained spaCy's model for Serbian and used a couple of tools for interesting visualization (more on that in the Appendix).

## 2 Annotation

The annotation guidelines included a total of 7 different NE categories that had to be labeled: PERS, ROLE, DEMO, ORG, LOC, WORK, EVENT (for detailed explanation see Table 1). The annotation was done with the BRAT annotation tool [6]. The novel was divided into chapters, each represented by a separate file with some prior labels already present. Therefore, the annotation process consisted of both labeling new entities and examining and potentially modifying the existing labels. For each chapter I kept a record of all the changes I made in a spreadsheet with the following columns:

- Added annotations (new entity labeled)
- Moved annotations (changed the span of the annotation)
- Changed annotations (changed the label of a previously marked entity)
- Problematic cases (unclear cases left as questions)

After completing a chapter, I would send the corresponding spreadsheet to JeRTeh for evaluation. After receiving the feedback, I would make the necessary changes to my annotations. Eventually, I would end up with a completely annotated novel that could be used as the train data in the next step (and also as gold data in the evaluation process).

[1] http://jerteh.rs/
[2] https://www.cost.eu/
[3] https://www.cost.eu/actions/CA16204
[4] https://brat.nlplab.org/
[5] version 2.0 was still used at the time
[6] http://brat.jerteh.rs/

| Label | Description | Explanation | Examples |
|---|---|---|---|
| **PERS** | names of people | Includes first names, last names and nicknames; Includes names of both real and fictional characters (thus including gods and saints); Roles such as *Dr.*, *Mr.*, *queen* etc. are NOT considered as a part of the PERS label (*see ROLE label*); Possessive adjectives derived from personal names are NOT marked as PERS (*e.g.*, Anna's) | Anna; Anna Smith; God; Jesus; St. Patrick; |
| **ROLE** | occupations, roles, responsibilities | Includes occupations, roles and responsibilities, regardless of whether they appear next to a proper name or not; can consist of multiple words/tokens | doctor; cobbler; king; general; manager; peasant; high-school teacher |
| **ORG** | organizations, institutions, associations | Includes, but not limited to, names of companies, political parties, educational institutions, sports teams, hospitals, museums, libraries, hotels, pubs, churches and sanctuaries | National Theatre; Socialistic party; St. Sava's Temple |
| **LOC** | geo-political entities | Includes, but not limited to, continents, countries, regions, cities, streets, mountains, islands, bodies of water, celestial bodies etc. | Balkan; Danube; New York; Adriatic sea; 5th Avenue |
| **WORK** | works of art | Includes, but not limited to, book titles, poems, songs, musical pieces, paintings, sculptures, newspapers etc. | Mona Lisa; Romeo and Juliet; Turkish March |
| **EVENT** | events | Includes names of events that occur regularly or that occurred once but have a special name - natural disasters, revolutions, battles, wars, concerts, sporting events etc. | Christmas; World War II; October Revolution |
| **DEMO** | nationalities and places of residence | Includes nationality, citizenship, residents of cities or regions, ethnic groups; Also includes adjectives derived from the aforementioned | Frenchman; New Yorker; Balkan (adj.) |

Table 1: Label descriptions

## 3 Training

Annotations created using the BRAT annotation tool are stored in a standoff format[7]. More detailed information can be found by following the link in the footnote, but below you can find a brief summary of the specific standoff flavor used by BRAT:

- **Text files** (`.txt`) - containing only the text (without annotations) in XML format
- **Annotation files** (`.ann`) - containing the entities with their labels and positions in the text (start, end)

This is, obviously, different from the format required by spaCy[8], so the first part of this task was to convert the data to the accepted format. Once the conversion is done, we can add the patterns to the `EntityRuler`[9] and split the data into train and test set (by default, 80% of the data is used for training and the remaining 20% for evaluation). Finally, I implemented a standard routine for training a NER model with spaCy (and later saving it to disk), making use of the following hyperparametres:

- **n_iter** - number of iterations
- **drop** - dropout rate

[7] https://brat.nlplab.org/standoff.html

[8] https://spacy.io/usage/rule-based-matching#entityruler-patterns

[9] https://spacy.io/api/entityruler

- **batch_from** - initial batch size
- **batch_to** - final batch size
- **batch_compound** - rate of batch size acceleration

Numbers are shown in Table 2 and Table 3 that should give a better insight into what the data looked like.

| | **sentences** | **entities** |
|---|---|---|
| **train** | 1736 | 2762 |
| **test** | 435 | 681 |
| | **2171** | **3443** |

Table 2: Number of sentences and entities across train and test set

| | **train** | **test** | |
|---|---|---|---|
| **PERS** | 1547 | 391 | 1938 |
| **ROLE** | 789 | 188 | 977 |
| **ORG** | 0 | 1 | 1 |
| **LOC** | 181 | 39 | 220 |
| **WORK** | 2 | 1 | 3 |
| **EVENT** | 6 | 1 | 7 |
| **DEMO** | 237 | 60 | 297 |
| | **2762** (80.2%) | **681** (19.8%) | **3443** |

Table 3: Number of entities by type across the train and test set

## 4 Evaluation

### 4.1 Standard metrics and different evaluation schemata

I conducted multiple methods for evaluating the previously trained model in order to get a better understanding of the model's performance. Naturally, I started with the standard metrics used in NER tasks (as described in Batista, 2018) that include *precision*, *recall*, and *F1-score*. I implemented a function that can return the mentioned scores for the whole document, as well as for individual labels. After tuning the hyperparametres (on a subset of the training data), I reached the F1-score of 97.63 for the whole document. See Table 4 and 5 for more details about hyperparametres and scores. It should be noted that the scores were calculated according to the CoNLL evaluation schema (Tjong Kim Sang and De Meulder, 2003), thus they were *strict* scores, namely, to cite the authors, "*a named entity is correct only if it is an exact match of the corresponding entity in the data file*". There are more fine-grained schemata, such as that of MUC (Chinchor and Sundheim, 1993) that includes a category for partial matches, or SemEval (described in Batista, 2018) that introduced different ways to measure P/R/F1 scores depending on the degree of overlap between the predicted and gold-standard entities.

| **Hyperparametres** | |
|---|---|
| `n_iter` | 30 |
| `drop` | 0.5 |
| `batch_from` | 4.0 |
| `batch_to` | 32.0 |
| `batch_compound` | 1.001 |

Table 4: Hyperparameter values for the best model

| **F1-Scores** | |
|---|---|
| **PERS** | 97.02 |
| **ROLE** | 98.68 |
| **ORG** | 100.0 |
| **LOC** | 100.0 |
| **WORK** | 100.0 |
| **EVENT** | 100.0 |
| **DEMO** | 96.77 |
| **DOCUMENT** | **97.63** |

Table 5: F1-scores per label and for the whole document

### 4.2 Novel approaches in NER evaluation

I did not make use of such fine-grained schemata, however, as I had decided to follow a recent novel approach described in the following paper: Fu et al. (2020). They authors argue that the standard evaluation metrics are becoming insufficiently informative given the abundance of models for NLP tasks today - "*[...] differences between holistic metrics such as [...] F1 do not tell us why or how particular methods perform differently and how diverse datasets influencethe model design choices*". What they suggest instead is to divide the data into buckets of entities based on different *attributes* and then evaluate the model on each of these buckets separately. This way, they argue, it is easier to locate the factors that improve or worsen the performance of the model.

#### 4.2.1 Attributes

The authors define attributes as *characterizations of different properties of spans or tokens of relevance*. They further divide them into **local attributes** (those that display the properties of spans/tokens themselves) and **aggregate attributes** (general properties that have to be calculated on the whole training corpus). In my project, I focused on incorporating some of the local attributes in the evaluation process. To be more precise, I used the following subset of attributes used by the authors:

- **sLen** – sentence length (in tokens)
- **eLen** – entity length (in tokens)
- **eDen** – entity density (number of entities in a sentence)

For each attribute I divided the data into three buckets depending on the value of the attribute. Sentences were divided into short (shorter than 5 tokens), middle (5-10 tokens), and long (longer than 5 tokens). Entity length buckets separated one-token, two-token, and multiple-token entities. Finally, the entity density was considered low if its value was less than $0.1$, moderate for values in range $[0.1\text{-}0.3]$, and high for values greater than $0.3$.

## 5 Observations

While the model had little trouble dealing with long sentences or sparse/dense sentences in regard to the number of entities ($F1 \approx 99$), an interesting observation that resulted from this evaluation method was the model's significant drop in performance for entities consisting of more than a single token ($F1 \approx 68$). The fact that the entity length has a huge influence on the performance of a model was also observed by the authors. They further went on to examine what modifications or additions could be made to a model in order to improve its performance with regards to entity length. For instance, they suggest that a significant improvement can be achieved by introducing a CRF layer. However, this is only applicable in case the dataset has a low *label consistency* - an aggregate attribute that represents a measure of versatility of an entity, which I didn't make use of for this project. This should be firstly examined in order to arrive to a more confident conclusion.


## Acknowledgments

I would like to thank the people from JeRTeh, most notably Dr. Branislava Šandrih, as well as Dr. Cvetana Krstev and Prof. Dr. Ranka Stanković for their supervision and guidance through the process of completing this project and gaining a valuable new experience.


## References


David S. Batista. 2018. Named-Entity evaluation metrics based on entity-level.

Nancy Chinchor and Beth Sundheim. 1993. MUC-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.

Jinlan Fu, Pengfei Liu, and Graham Neubig. 2020. *Interpretable Multi-dataset Evaluation for Named Entity Recognition*. Cornell University.

Cvetana Krstev, Ivan Obradović, and Duško Vitas. 2014. A System for Named Entity Recognition based on Local Grammars. *Journal of Logic and Computation, Oxford Journals*, 24(2):473–489.

Rohit Kumar Singh. 2021. A 2021 guide to named entity recognition.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. *COLING2018*.

Branislava Šandrih, Cvetana Krstev, and Ranka Stanković. 2019. Development and Evaluation of Three Named Entity Recognition Systems for Serbian - The Case of Personal Names. In *RANLP 2019: Recent Advances in Natural Language Processing*, pages 1060–1068.


## A Visualization

In addition to the evaluation methods I used above, I decided to compare the performance of my model with the pretrained NER model in spaCy for Serbian (described in Šandrih et al., 2019 and Krstev et al., 2014). As the mentioned model for Serbian can detect only those entities of type PERS, the comparison was made only with regard to this label. The F1 score calculated on the pretrained model is $0.59$ (compared to $97.2$ in my case). Although this looks like a significant difference, it

was somewhat expected given that the pretrained model was trained on a collection of newspaper articles from online portals. The language used in such documents differs notably when compared to the language used in a novel over a 100 years old. Nevertheless, it was an interesting comparison and I tried to make it more tangible by using the GEMINI[10] tool for visualization through the JeRTeh's web portal *NER & Beyond*[11] designed for the purposes of the COST action described at the beginning of this paper. I included the file to my project submission which displays the PERS entites recognized by the two models in the first chapter of the novel. Furthermore, I used displaCy[12] for displaying the output of my model on the test set.

[10] https://github.com/fyh828/gemini/
[11] http://nerbeyond.jerteh.rs/
[12] https://spacy.io/usage/visualizers