# NLMaps Web: A natural language interface to OpenStreetMap

Simon Will[1],*

[1] Department of Computational Linguistics, Heidelberg University, Heidelberg, Germany; simon.will@gorgor.de

* Author to whom correspondence should be addressed.

This abstract was accepted to the Academic Track of the State of the Map 2021 Conference after peer-review.

Nominatim [1] and Overpass [2] are powerful ways of querying OSM, but the Overpass Query Language is somewhat impractical for quick queries for unfamiliar users. In order to query OSM using natural language (NL) queries such as “Show me where I can find drinking water within 500m of the Louvre in Paris”, Lawrence and Riezler [3] created the first NLMaps dataset mapping NL queries to a custom machine-readable language (MRL), which can then be used to retrieve the answer from OSM via a combination of queries to Nominatim and Overpass. They extended their dataset in a subsequent work by auto-generating synthetic queries from a table mapping NL terms to OSM tags – calling the combined dataset NLMaps v2. [4] The proposed purpose of these datasets is training a parser that can parse NL queries into their MRL representation, as done in [4–7].

The main aim of this work was to build a web-based NLMaps interface that can be used to issue queries and to view the result. In addition, the web interface should enable the user to give feedback on the returned, either by simply marking the parser-produced MRL query as correct or incorrect, or by explicitly correcting it with the help of a web form. This feedback should be directly used to improve the parser by training it in an asynchronous online learning procedure.

After observing that parsers trained on NLMaps v2 perform poorly on new queries, an investigation into the causes for this revealed several shortcomings in NLMaps v2, mainly: (1) Train and test split are extremely similar limiting the informativeness of evaluating on the test split. (2) Various inconsistencies exist mapping from NL terms to OSM tags (e.g. “forest” sometimes mapping to natural=wood, sometimes to landuse=forest). (3) The NL queries’ linguistic diversity is limited since most of them were generated with a very simple templating procedure, which leads to parsers trained on the data not being very robust to new wordings of a query. (4) In a similar vein, there is only a small amount of different area names in NLMaps v2 with the names “Paris”, “Heidelberg” and “Edinburgh” being so dominant that parsers are biased towards producing them. (5) Some generated NL queries are not a good representation of natural language, which makes them counter-productive learning examples. (6) Usage of OSM tags is sometimes incorrect, which affects the usefulness of produced parses.

The detailed analysis is used to eliminate some of the shortcomings – such as incorrect tag usage – from NLMaps v2. Additionally, a new approach of auto-generating

Will, S. (2021). NLMaps web: A natural language interface to OpenStreetMap
In: Minghini, M., Ludwing, C., Anderson, J., Mooney, P., Grinberger, A.Y. (Eds.). Proceedings of the Academic Track at the State of the Map 2021 Online Conference, July 09-11 2021, 13-15. Available at https://zenodo.org/communities/sotm-2021
DOI: 10.5281/zenodo.5112227

NL-MRL pairs with probabilistic templates is used to create a dataset of synthetic queries that features a significantly higher linguistic diversity and a large set of different area names. The combination of the improved NLMaps v2 and the new synthetic queries is called NLMaps v3.

A character-based GRU encoder-decoder model with attention [8] is used for parsing NL queries into MRL queries using the configuration that performed best in previous work [7]. This model is trained on NLMaps v3 and used as the parser in the newly developed web interface. Mainly through advertising on the OSM talk list and the OSM subreddit, 12 annotators were hired from all over the world to use the web interface to issue new NL queries and to correct the parser-produced MRL query if it is incorrect. They were required to complete a tutorial before the annotation job and received help compiled from taginfo [9], TagFinder [10] and custom suggestions for difficult tag combinations. The collected dataset contains 3773 NL-MRL pairs and is called NLMaps v4.

With the help of NLMaps v4, an informative evaluation can be performed revealing that a parser trained on NLMaps v2 data achieves an exact match accuracy of 5.2% on the MRL queries of the test split of NLMaps v4 while a parser trained on NLMaps v3 performs significantly better with 28.9%. Pre-training on NLMaps v3 and fine-tuning on NLMaps v4 achieves an accuracy of 58.8%.

Since the goal of this work is to deliver an online learning system – i.e. a system that updates the parser directly after receiving feedback in the form of an NL-MRL pair –, various online learning simulations are conducted in order to find the best setup. In all cases, the parser is pre-trained on NLMaps v3 and then receives the NL-MRL pairs in NLMaps v4 one by one, updating the model after each step. The most simple variant of the experiment uses only the one NL-MRL pair for the update, another variant adds NL-MRL pairs from NLMaps v3 to the minibatch and a third variant additionally adds further “memorized” NL-MRL pairs from previously given feedback to the minibatch. The main findings of the simulation are that all variants improve performance on NLMaps v4 with respect to the pre-trained parser, but with some of them the performance on NLMaps v3 degrades. The simple variant that updates only on the one NL-MRL pair is particularly unstable, while adding NLMaps v3 instances stabilizes the performance on NLMaps v3 and improves the performance on NLMaps v4. Adding the instances from memorized feedback further improves the performance to an accuracy of 53.0%, which is still lower than the offline batch learning fine-tuning mentioned in the previous paragraph.

In conclusion, this work improves the existing NLMaps dataset and contributes two new datasets – one of which is especially valuable since it consists of real user queries – laying the groundwork necessary for further enhancing NLMaps parsers. The current parser – achieving an accuracy of 58.8% – can be used by OSM users via the new web interface [11] for issuing queries and also for correcting incorrect ones. Future work will concentrate on improving the web interface’s UX and enhancing the parser’s performance in terms of speed and accuracy.

## References

[1] Hoffmann, S., Metten, M. T., & Quinion, B. (2021). Nominatim: Open-source geocoding with OpenStreetMap data. Retrieved from https://nominatim.org

[2] Olbricht, R. (2021). Overpass API. Retrieved from https://overpass-api.de

[3] Haas, C., & Riezler, S. (2016). A corpus and semantic parser for multilingual natural language querying of OpenStreetMap. In: *Proceedings of the 2016 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 740-750.
[4] Lawrence, C., & Riezler, S. (2018). Improving a neural semantic parser by counterfactual learning from human bandit feedback. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 1820-1830.
[5] Lawrence, C., & Riezler, S. (2016). NLmaps: A natural language interface to query OpenStreetMap. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*: System Demonstrations, 6-10.
[6] Will, S. (2018). Parsing NLmaps queries using adversarial neural machine translation. Unpublished bachelor's thesis, Heidelberg University.
[7] Staniek, M. (2020). Towards error-aware interactive semantic parsing. Unpublished master's thesis, Heidelberg University.
[8] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint*, arXiv:1409.0473.
[9] Topf, J. (2021) Taginfo. Retrieved from https://taginfo.openstreetmap.org
[10] Gwerder, S. (2014). Tag-Suchmaschine und Thesaurus für OpenStreetMa. Student research project, HSR University for Applied Sciences Rapperswil.
[11] Will, S. (2021) NLMaps Web. Retrieved from https://nlmaps.gorgor.de