# Open Information Extraction on German Wikipedia Texts

Christian Klose[1], Zhou Gui[1] and Andreas Harth[1]

*[1] Friedrich-Alexander-Universität Erlangen-Nürnberg, Chair of Technical Information Systems, Lange Gasse 20, 90403 Nürnberg, Germany*

**Abstract**

Knowledge Graphs are becoming a fundamental building block for semantic search and voice assistants. This paper deals with the automated Knowledge Graph Construction from unstructured data. Predominantly, the focus is on Open Information Extraction (Open IE), an unsupervised learning approach that attempts to extract triples from plain text independent of their domain. Hence, it is the first step towards automated Knowledge Graph Construction. Previous work mainly applied Open IE to English texts. In this paper, the focus is on German texts. Due to the lack of German Open Information Extraction datasets, a dataset on the basis of Wikipedia is created. Two Open Information Extraction Systems for German are introduced. Finally, the performance of the systems are evaluated.

**Keywords**

Open Information Extraction, Natural Language Processing, Knowledge Graph Construction

## 1. Introduction

In his vision of the Semantic Web [1], Tim Berners-Lee described a change from the Web of documents for and by people to a Web of information. According to his vision, information on the Web should not only be manipulable by humans, but also by machines. Most documents in the World Wide Web consist to a large extent of text and are still difficult for machines to process today. For this reason, the W3C[1] has developed a universal language Resource Description Framework (RDF), which makes information for machines on the Web accessible. Information in RDF can be serialized in multiple formats. One common format is Turtle. Turtle is a text representation of an RDF Graph which allows to store RDF triples in a compact and human readable form. A large collection of RDF Graphs in a specific domain can construct a Knowledge Graph (KG). Virtual assistants in particular can make use of facts, events and abstract concepts stored in Knowledge Graphs to bring insights to people during semantic search or question answering.

In order to build Knowledge Graphs, knowledge can be extracted in the form of triples from documents that are available in natural language. The transformation from text into a machine readable form is, therefore, a core task for building Knowledge Graphs. It can be broadly summarized as the goal of Machine Reading [2]. In the field of AI, Machine Reading is a long

*Text2KG 2022: International Workshop on Knowledge Graph Generation from Text, Co-located with the ESWC 2022, May 05-30-2022, Crete, Hersonissos, Greece*

christian.klose@fau.de (C. Klose); zhou.gui@fau.de (Z. Gui); andreas.harth@fau.de (A. Harth)



[1]World Wide Web Consortium https://www.w3.org/

standing goal and is discussed in the research community under the term Information Extraction (IE). IE includes downstream tasks such as Named Entity Recognition (NER), Relation Extraction (RE) or Entity Linking (EL). In recent years, an unsupervised approach to Relation Extraction, namely Open Information Extraction, has shown promising results and is therefore, the main subjective of this paper. The paper is structured as follows: In Section 2 the previous work is outlined. Thereafter, in Section 3, the scientific approach applied for our research is described. In Section 4 the results are presented and discussed. Finally, in Section 5 a conclusion is drawn.

## 2. Related Work

Open Information Extraction is about extracting all possible triples from text, without knowing the relations or entities occurring in it a priori. The first Open IE system ever created is *Text Runner* [3], a learning-based system developed by a group of researchers at the University of Washington. In the years to follow, other systems were introduced, each attempting to improve on the results of the state-of-the-art by overcoming identified weaknesses and flaws in the systems. In addition, other types than learning-based system emerged, namely rule-based, *clause-based* and systems making use of *inter-proposition relationships* [4]. Rule-based systems entirely depend on hand-crafted rules or patterns. Systems that make use of this approach are, for example, *KrakeN* [5] or *Exemplar* [6]. In order to improve the precision of the systems described above, the idea of breaking down complex sentences into smaller components (clauses) came up. Two Clause-based Systems in particular are worth mentioning: *ClausIE* [7] and *Stanford Open IE* [8]. The system types mentioned so far have one common weakness. None of them is using the context and, therefore, a correct extraction cannot be guaranteed. Inter-Proposition-based Systems are trying to bridge this gap. Systems of this category, for example, are *RelNoun* [9], *OpenIE4* [10], *NestIE* [11] and *MinIE* [12].
One of the first to apply neural networks to Open IE were [13] with *RnnOIE*. The scientists formulated the problem as a sequence labeling task. Recently proposed models that follow a sequence labeling approach are *SenseOIE* [14], *SpanOIE* [15] and *iRankOIE* [16]. A downside of this discovered by [17] is, however, that sequence labeling models are not able to change the sentence structure or use new auxiliary words in the extraction. [18] used a different neural approach called sequence generation to develop *CopyAttention* and overcome that downside. Furthermore, [19, 20, 17] describe end-to-end approaches using seq2seq models based on the encoder-decoder framework alleviating the downsides and the need for hand-crafted patterns.

## 3. Research Method

### 3.1. Dataset Creation

One of the main challenges within Open IE is to verify the quality of extractions made by the system. A solution to this is to have a dataset, that is to find qualitative training and testing data where triples are mapped to sentences. Currently, two approaches to automatically generate training data are considered particularly useful among researchers. First, the *infobox-matching approach* [21] where Wikipedia infobox values are linked to sentences in the corpus and second,

the *distantly supervised approach* [22] where existing knowledge bases are used to heuristically map triples to sentences.
For the creation of the German Wikipedia dataset, the infobox-matching approach is used. In total, 5 steps were executed to create a clean dataset including 1) finding and downloading a Wikipedia dataset 2) prepossessing and cleaning the text 3) matching all infobox triples to the correspond page text, 4) matching the triples on a sentence level and 5) filtering out noisy training examples. After the last step, a number of $6,453$ triples mapped to $5,372$ sentences containing $1,324$ relation types was derived. Furthermore, the average number of words used in each part was calculated. On average, the subject has 2.0, the predicate 1.0 and the object 1.5 words. Last but not least, the average length of a sentence was computed and amounts to 22.8 words. The dataset and the code used to create the dataset are published and freely accessible.[2]

## 3.2. Open IE Systems

In total, two systems were implemented and used for our research. The first system is turCy and we implemented turCy as a spaCy[3] pipeline component to leverage the POS Tagger and Dependency Parser (DP). The second system uses an encoder-decoder seq2seq neural model that we call NeuralGerOIE.

### 3.2.1. TurCy

Research by [5] and [23] implies that with a decent amount of POS and DP patterns, a large variety of triples can be extracted - independently of any other constraints. TurCy is following a similar approach. In fact, it is a pattern learning system for binary extractions that is assembled of two essential components. The first is the *Pattern Builder*, the second is the *Triple Extractor*. A pattern consists of nodes that represent the POS-tags of a sentence. The relations between the nodes reflect the dependencies parse tree. A pattern with respect to a sentence can be used to represent exactly one triple. The pattern itself consists of subpatterns. Each subpattern represents a node in a tree and maps a word with left and right child nodes. For the sentence: "*Im Jahr 2019 zählte Nürnberg 518370 Bewohner.*" the POS and dependency tree is shown in Figure 1.

The Triple Extractor of turCy is - at its core - a recursive sub-tree search using the Pattern List. The algorithm starts at the root node, traverses each path up to the leaf of the tree and checks whether a node and its edge match a sub-pattern of a respective pattern. If all sub-patterns match, the triple is assembled and stored with respect to a sentence. Notice, a match implies that at last one token of all three parts (subject, predicate, object) of a triple were found during the recursive sub-tree search in the sentence. However, the true nature of the algorithm is more complicated. If the interested reader wants to fully understand the working mechanism, we recommend diving into the code. Therefore, and also to ensure full transparency of our research, turCy has been packaged as a python library and is released under an open-source license.[4]

[2] https://github.com/ChrisDelClea/WikiGerman4OIE
[3] Library for advanced natural language processing: https://spacy.io
[4] https://github.com/ChrisDelClea/turCy

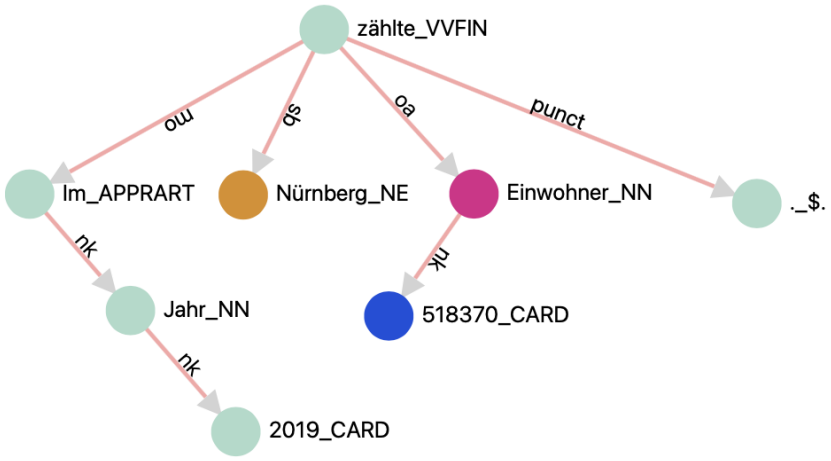


**Figure 1:** POS-DP-Tree or pattern representation of a simple sentence. The colored nodes are the parts of the triple in the pattern, with orange being the subject(s), purple being the predicate(s), and blue being the object(s). All other nodes are colored in green.

### 3.2.2. NeuralGerOIE

[5] The latest state-of-the-art Open IE systems use seq2seq neural networks. Therefore, the second system developed follows the *sequence generation approach.* and was created using the *Simple Transformers*[6] library. For training of the model, the WikiGerman4OIE dataset (3.1) was utilized. In addition, a pre-trained BART model for German was used.[7] It is important to mention that the model was trained to output multiple extractions within the scope of one subject.

We fine-tuned the model for 10 epochs using a batch size of 8. The maximum length of the input sentence was set to 300, i.e. all words thereafter were truncated. A difference between the seq2seq models discussed earlier and the approach described here, is the type of separator used. While [18] used start and end tags for each part of a triple (<arg1> Deep Learning </arg1><rel> is a subfield of </rel><arg2> Machine Learning </arg2>), we found that one token before each part along with a final end token was sufficient. In addition, the model struggled to output the same separator token multiple times within a sequence. Therefore, we added a number to each separator token. Lastly, we noticed that the names of the separator tokens affected the quality of the outputs. We proceeded with the following triple input: <sub> Deep Learning <rel0> is a subfield of <obj0> Machine Learning <end>. In total, the fine-tuning took about 2 hours 25 minutes on a *Nvidia Tesla V100 32GB GPU* to complete. The results are discussed in the subsequent section.

[5] https://github.com/ChrisDelClea/NeuralGermanOIE
[6] https://simpletransformers.ai/
[7] https://huggingface.co/Shahm/bart-german

## 4. Results

Each of the Open IE systems is evaluated against a gold dataset. The reason for high-quality annotations originates from the need to obtain more accurate insights regarding the quality of the extractions. Therefore, a subset of the WikiGerman4OIE dataset (3.1) was annotated by two of the authors. In total, the gold dataset consists of 47 sentences and 175 triples. On average, a sentence contains 3.8 triples. This subset is the basis for the evaluation process. Regarding the quantitative metrics, precision, recall, and $F_1$-score is computed using the token-based evaluation method introduced by [24] in a slightly adjusted manner to fit the systems outputs. The evaluation process is as follows: First, the full WikiGermanOIE dataset for turCy-large and the gold dataset for turCy-small was used to create the patterns with the Pattern Builder. The result were two pattern lists with sizes of 6,453 and 175 patterns, respectively, corresponding to the number of triples in each dataset. Second, the 47 sentences were fed as the only input into the Triple Extractor and the NeuralGerOIE prediction function. Lastly, precision, recall and $F_1$-score were calculated.

**Table 1**
System's accuracy analysis using default metrics.

| System | # Patterns | # Extractions | # Matches | Prec. | Recall | F1 |
|---|---|---|---|---|---|---|
| turCy-large | 6,453 | 29 | 21/175 | 0.71 | 0.11 | 0.19 |
| turCy-small | 175 | 114 | 94/175 | 0.83 | 0.52 | 0.64 |
| NeuralGerOIE | - | 52 | 40/175 | 0.72 | 0.30 | 0.43 |

In general, we found that turCy-small achieves a better $F_1$-score as NeuralGerOIE due to its ability to extract many triples. At the same time, we noticed that the NeuralGerOIE obtained a very high precision for sentences annotated with a single triple.
A comparison between turCy-small and turCy-large (the only difference is in the number of patterns and their dataset of origin) indicates that, the quality of the automatically generated dataset is lower than initially expected. The reason for this assumption is that while trucy-large contains a high number of patterns build from the automatically generated dataset, only 29 triples were extracted. TurCy-small, on the other hand, yielded significantly more extractions with a lower number of patterns created from the gold dataset. In fact, the result is very counter-intuitive, as one would expect the number of extractions to be linearly correlated with the number of patterns. Moreover, when comparing turCy and NeuralGerOIE, we found that while turCy can only output words from the text in the extractions, the neural model can learn a direct representation between the words used in the text and the corresponding words in the gold dataset.

In addition to the quality analysis, we examined the run-time as Open IE systems are intended to process large amounts of text data at rapid pace. In order to make a judgment about the run-time, basically two metrics are taken into consideration. First, is the number sentences processed per second (# sent./sec.). Second, is the number of triples yielded per second (# triples/sec.).

Furthermore, the ratio of these two metrics with respect to the number of stored patterns is of interest. Table 2 shows that turCy-small is the best performing system, followed by NeuralGerOIE. As expected, the number of patterns has a major impact on the run-time - as we can see with turCy-large - leading to the conclusion that one of the main objectives of rule-based Open IE systems is to keep the number of patterns as small as possible, but as large as necessary to maximize the number of extractions.

**Table 2**
Run-time comparison between RE and Open IE.

| # System | # Patterns | # sent. /sec. | # triples /sec. |
|---|---|---|---|
| turCy-large | 6,453 | 0.47 | 0.71 |
| turCy-small | 175 | 1.12 | 2.7 |
| neuralGerOIE | - | 1.14 | 1.14 |

## 5. Conclusion & Outlook

In this paper the contribution is twofold. First, a dataset for Open Information Extraction based on German Wikipedia texts were created and published. Second, two different approaches for Open IE were implemented and evaluated. Several interesting research directions for future works can be recommended. We firmly believe that there is still potential for improvement in terms of dataset quality and quantity. For instance, in order to improve the quality, a crowd-sourcing platform such as Amazon Mturk could be leveraged. In addition, the distantly supervised approach for automated training data generation can be explored. In doing so, it would also help to determine what impact the applied approach for automated training data generation has on the quality of extractions made by the Open IE system.
Finally, the two Open IE systems can be further optimized to yield better results. For example, for the rule-based system turCy, tree pruning approaches can be explored to reduce the overall number of patterns and therefore, improve the extraction speed. Regarding NeuralGerOIE multi-subject extractions, different architectures and the utilization of more recent language models might be considered.

## 6. Acknowledgments

## Acknowledgments

This research paper was created within the scope of the project: Software Campus 2.0 (FAU) Grant number 01IS17045. The project was funded by the German government, therefore, we would kindly thank them for their sponsorship.

## References


[1] T. Berners-Lee, J. HENDLER, O. LASSILA, The semantic web, Scientific American (2001) 34–43.

[2] O. Etzioni, M. Banko, M. Cafarella, Machine reading., 2007, pp. 1–5.

[3] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, O. Etzioni, Open information extraction from the web, IJCAI International Joint Conference on Artificial Intelligence (2007) 2670–2676.

[4] C. Niklaus, M. Cetto, A. Freitas, S. Handschuh, A survey on open information extraction, arXiv preprint arXiv:1806.05599 (2018).

[5] A. Akbik, A. Löser, KrakeN: N-ary facts in open information extraction, in: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX), 2012, pp. 52–56. URL: https://www.aclweb.org/anthology/W12-3010.

[6] F. Mesquita, J. Schmidek, D. Barbosa, Effectiveness and efficiency of open relation extraction (2013) 447–457. URL: https://www.aclweb.org/anthology/D13-1043.

[7] L. Del Corro, R. Gemulla, Clausie: clause-based open information extraction, in: Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 355–366.

[8] G. Angeli, M. J. J. Premkumar, C. D. Manning, Leveraging linguistic structure for open domain information extraction, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 344–354.

[9] H. Pal, et al., Demonyms and compound relational nouns in nominal open ie, in: Proceedings of the 5th Workshop on Automated Knowledge Base Construction, 2016, pp. 35–39.

[10] M. Mausam, Open information extraction systems and downstream applications, in: Proceedings of the twenty-fifth international joint conference on artificial intelligence, 2016, pp. 4074–4077.

[11] N. Bhutani, H. Jagadish, D. Radev, Nested propositions in open information extraction, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 55–64.

[12] K. Gashteovski, R. Gemulla, L. d. Corro, Minie: minimizing facts in open information extraction, Association for Computational Linguistics, 2017.

[13] G. Stanovsky, J. Michael, L. Zettlemoyer, I. Dagan, Supervised open information extraction, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 885–895.

[14] A. Roy, Y. Park, T. Lee, S. Pan, Supervising unsupervised open information extraction models, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 728–737.

[15] J. Zhan, H. Zhao, Span model for open information extraction on accurate corpus, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 9523–9530.

[16] Z. Jiang, P. Yin, G. Neubig, Improving open information extraction via iterative rank-aware

learning, arXiv preprint arXiv:1905.13413 (2019).
[17] K. Kolluru, S. Aggarwal, V. Rathore, S. Chakrabarti, et al., Imojie: Iterative memory-based joint open information extraction, arXiv preprint arXiv:2005.08178 (2020).
[18] L. Cui, F. Wei, M. Zhou, Neural open information extraction, arXiv preprint arXiv:1805.04270 (2018).
[19] P.-L. H. Cabot, R. Navigli, Rebel: Relation extraction by end-to-end language generation, in: Findings of the Association for Computational Linguistics: EMNLP 2021, 2021, pp. 2370–2381.
[20] K. Kolluru, V. Adlakha, S. Aggarwal, S. Chakrabarti, et al., Openie6: Iterative grid labeling and coordination analysis for open information extraction, arXiv preprint arXiv:2010.03147 (2020).
[21] F. Wu, D. S. Weld, Open information extraction using wikipedia, in: Proceedings of the 48th annual meeting of the association for computational linguistics, 2010, pp. 118–127.
[22] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2009, pp. 1003–1011. URL: https://www.aclweb.org/anthology/P09-1113.
[23] Mausam, M. Schmitz, S. Soderland, R. Bart, O. Etzioni, Open language learning for information extraction, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2012, pp. 523–534. URL: https://www.aclweb.org/anthology/D12-1048.
[24] W. Lechelle, F. Gotti, P. Langlais, Wire57: A fine-grained benchmark for open information extraction (2019) 6–15.