# Entity Extraction in the Ecological Domain — A practical guide

Vladimir Udovenko[1] Alsayed Algergawy[2]

**Abstract:** Scientific information comes in many shapes: As data in databases or spreadsheets, but also as textual information in papers and books. In order to exploit all this information and integrate all the knowledge that is available regarding a specific entity, it is necessary to identify entities and their relationships. In this paper, we provide a guideline to setting up a pipeline that supports entity and relationship extraction from scientific publications from the ecological domain.

**Keywords:** Information integration; Entity extraction; Relation extraction

## 1 Introduction

Research builds on knowledge gained from earlier resources. Such knowledge is encoded in data stored in various data sources as well as text. An essential step for integrating data from these heterogeneous data sources is to identify similar entities represented in different sources as well as their relations[DHI12]. However, the majority of scientific data are represented in unstructured formats, i.e. information and knowledge of interest are still hidden mostly in data sets without any formalized schema. It maybe scientific publications. They may contain tables or pictures, but mostly text data. The main objective is to make such information stored in text accessible for further data processing, such as integration and analysis [YB18].

Extracting information of interest from scientific publications in general and ecological in specific including entities and relations is a critical challenge to support the automation of integrating structured and unstructured data [KN04]. Suppose that we have this sample of a scientific publication "N2O contributes to the destruction of ozone layer", it becomes more difficult to identify and recognize named entities in such a domain specific scenario. Compared to the personal domain, which is well-established, it is hard to create and/or get annotations for such named entities. This requires the need to prepare training datasets that can be used either in learning-based approaches or as a list of domain-specific entities in the rule-based approaches. In both cases, the preparation process includes the collection and organization of domain specific information resources, such as ontologies.

[1] Friedrich-Schiller University of Jena, Heinz Nixdorf Chair for Distributed Information Systems, Germany

[2] Friedrich-Schiller University of Jena, Heinz Nixdorf Chair for Distributed Information Systems, Germany alsayed.algergawy@uni-jena.de

doi:10.18420/btw2019-ws-16

To this end, in this paper, we describe how existing building blocks can be combined to create a framework that supports in the identification and extraction of soil-related entities from scientific publications belonging to the Biodiversity Exploratory[3]. The extracted set of entities are then annotated by domain specific resources, which support the identification of relations across the entities. The proposed approach is implemented and validated using more than 100 publications and the preliminary experiments demonstrate encourage results.

## 2 Related work

The main goal of information extraction is the organization and structure of hidden knowledge in textual data that makes it accessible for other applications, e.g. as part of joint data integration systems [Ho02, Go18, Ch06]. In general, three main steps are needed for information extraction, namely; *text preprocessing*, *named entity recognition*, and *entity linking* (relationships between named entities). Named entity ($NE$) recognition is the task of identifying and classifying predefined types of named entities, like persons, location, etc. [YB18, BKL09, NS07]. In general, there are two approaches of named entity recognition [NS07]: *rule-based* and *statistical-based* approaches. In the case of rule-based approaches, manually constructed rules like regular grammars are used. Gazetteer-based annotation technique (string matching) is also an element of the rule-based toolkit. Using statistical methods of named entity recognition makes it possible to derive such rules based on *training data*. Such statistical models are general applications of *machine learning*. In these approaches, text chunk labeling is considered as a classification task and several algorithms can be used for this task, such as conditional random fields, supervised learning techniques like SVM [CL11] and deep learning [Sh17].

## 3 Proposed framework: An overview

To deal with the extraction of entities and relations between entities from the ecological domain, we propose a new approach. The main idea of the proposed approach is to exploit semantic information represented in domain-specific ontologies. The main components of proposed approach are depicted in Fig. 1. The figure shows that the framework has two main components to extract entities and relations as well as necessary preprocessing steps. In the following and for the space limitation, we are going to focus on the entity extraction and recognition component.

**Term extraction**: As Figure1 shows the proposed framework accepts three kinds of inputs: (i) text from where entities should be identified and classified, (ii) domain information resources: gazetteers or ontologies, and optional (iii) domain expert knowledge. First, the proposed framework accepts a text corpus and applies a preprocessing step, i.e. tokenization, sentence splitting, and POS-tagging. This functionality is implemented as elements of

[3] https://www.biodiversity-exploratories.de/startseite/

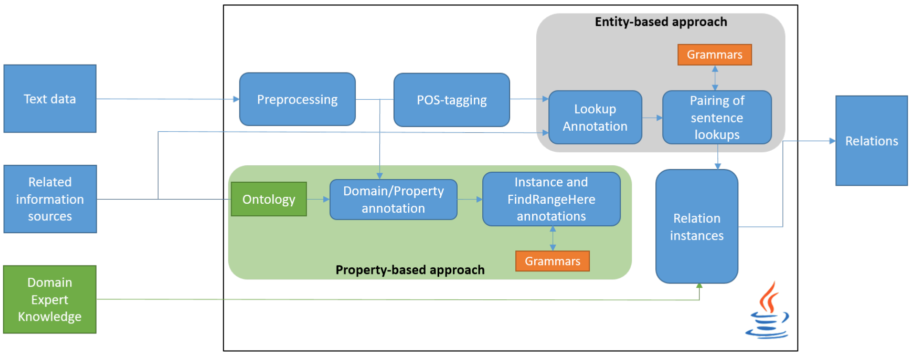

Fig. 1: Proposed framework

GATE corpus processing pipelines. Due to the size of the annotated XML files, we cannot store the whole text corpus in the main memory, as thus leads to its overflow even in the case of relatively small text set. To make this framework able to work with text corpus of any size, each document will be processed separately with new initialization of GATE resources. After preparing the input text, the next task is to extract terminologies related to the domain of interest. The term extraction process is a pattern matching problem. It may be solved using a *part of speech tagger* and a set of strict grammar rules in order to reflect the context. Following the definition in the book by [Ki14], we see that a term may be a noun phrase or a single word and also may be composed of nouns, adjectives, and prepositions. Therefore, a method to assign part-of-speech (POS) tags to tokens in textual data is used. The deployed method creates *term* annotations based on POS-tags using the grammar-based pattern matching. The expected result is a set of named entities with *term* label extracted from the given text corpus.

**Keyness ranking**. Once having the initial list of extracted terms, the next step is to filter it, because it will be a mix of general terms (non-specific) and domain specific terms which are needed. For each term, it is possible to compute a appearance frequency, but in the case of non-specific terms it will be always bigger than domain specific terms frequencies. Keyness ranking [Ki14] probably provides a solution for this problem. The main idea is relatively simple, we compute the frequencies of previously extracted terms in some general text corpus, also called *reference corpus*, $keyness_{term} = \frac{fpm_{focus}+n}{fpm_{ref}+n}$, where, $fpm_{focus}$ is the term frequency normalized per million in focus corpus (our normal working corpus), $fpm_{ref}$ is the term frequency normalized per million in reference corpus and $n$ - smoothing parameter, by default $n = 1$. An excerpt of the result is shown Table 1.

Tab. 1: A sample of extracted terms along their keyness values

| **Term** | beech | grassland | Microbiol | decomposer | fine root | rRNA | Soil |
|---|---|---|---|---|---|---|---|
| **Keyness** | 653.6 | 641.8 | 564.35 | 525.03 | 421.25 | 351.73 | 344.34 |

**Validation and filtering.** Keyness ranking provides a good filtering already and helps with the selection of needed terms. But it is semantically blind, some of the extracted terms are still non-domain terms. To deal with this issue, we make use of the services provided by

BioPortal[4], which support the possibility to retrieve the definition for each term. Bioportal provides a comfortable terminology search API and returns requested information as JSON data. From such responses, we extract the definition of each term as well as the ontology that contains it. For example, here is the search for the term *soil structure*

```
   {
   "prefLabel": "structure of soil",
    - "synonym": [
      "soil structure"
   ],
    - "definition": [
      "The structure of some soil."
   ],
   "links": {"ontology":
   "http://data.bioontology.org/ontologies/AGRO",
...}
```

List. 1: Example of JSON response

This method allows us to get a list of domain ontologies for a given text corpus. Additionally, an automated semantic terminology validation/filtering may be implemented based on keywords in definitions. The appearance of word *soil* in retrieved definition speaks in favor that this term is related to the domain of soil science and so on.

**Entity annotation.** After preparing the input text for processing (token and sentence annotations), the next step is identify and recognize named entity. In the current implementation, we make use of two different schemes: (i) using classical gazetteers: we have a list of entities and search for them in given text data. Additionally, it may be improved with fuzzy string matching techniques. (ii) using ontology as an information resource, which requires some preparation before usage. To implement ontology-based annotation of named entities, we construct a little bit tricky architecture for GATE processing application. After the ontology is loaded as language resource, we construct two processing pipelines: one for ontology resource pre-processing (*RootFinder*) and another corpus pipeline to create annotations. RootFinder pipeline is here to prepare ontology-resources (related human-readable lexicalizations). The result set is stored in OntoRoot gazetteer module and then forwarded into Flexible gazetteer in corpus pipeline to make annotations based on extracted ontology resources.

## 4 Experimental evaluation

The proposed approach has been developed and implemented using Java 8 utilizing GATE 8.4.1[5] with embedded JAPE- for text annotation and grammars over annotations and Apache Jena 3.9.0- for ontology processing. To validate the performance of the approach, we carried out a set of experiments utilizing a corpus of 112 scientific works (articles, publications,

[4] http://bioportal.bioontology.org/

[5] https://gate.ac.uk/

theses, etc) from the ecological and environmental domains obtained from the Biodiversity Exploratory publication list. Originally they are in PDF format, and thus text data extraction was needed for next steps of work. Preprocessing like tokenization and sentence splitting are implemented as a part of GATE pipelines.

To evaluate the quality of the term extraction component, we asked domain experts from the soil from different scientific groups. We first run the term extraction process, selected the top-1000 terms and split them into four different sets, allowing overlap between sets. Then we asked domain experts to validate the set of extracted terms. Computing the precision of the available evaluations we get a precision of $precision_1 = 0.607$ for the first group, while the second group scores with a precision of 0.846. We believe that this initial and preliminary results are encouraging especially for this specific domain.

**Keyness ranking.** Here we consider the computation of keyness score on an example of the keyword *soil*. Before all, it is necessary to get normalized per million frequencies of the extracted keyword. In our working (focus) corpus there are about 2149404 tokens and *soil* occurs 16499 times. Hence, the normalized frequency is: $fpm_{focus} = \frac{16499 \cdot 1000000}{2149404} = 7676.08$. In the reference corpus, we may find the same term *soil* 3489 times, or 28.38 per million. In this regard, keyness score will be computed as follows: $keyness_{term} = \frac{fpm_{focus}+n}{fpm_{ref}+n} = \frac{7676.08+1}{28.38+1} = 261.302$. Here $n = 1$ is a smoothing parameter used to prevent division by zero if some term was not found in the reference corpus. A larger value of the keyness score (in comparison with other terms) speaks in favor that this term is domain specific. Using these computed values, we construct a terminology ranking table.

**Search for domain-relevant ontologies.** To achieve this task, we make use of BioPortal, which provides access to 774 ontologies (as of 28.01.2019). To find domain knowledge resources we used the top-2000 list of keyness-ranked domain terms. Technically it was relatively difficult to apply search API to the whole list, instead we have drawn three random samples with about 100 terms in each one. In settings of search process we established exact matching. Based on this, we create ranked lists of ontologies for each prepared terminology sample. Table 2 illustrates an example of this ranking. By using three samples we compute the average score and use it as a criterion for ontology selection. Note that very big ontologies like IOBC are less applicable in the context of this work due to technical limitations.

Tab. 2: Occurrence-based ranking of domain ontologies.

| Ontology | Terminology samples | | | Average |
|---|---|---|---|---|
| | Sample 1 | Sample 2 | Sample 3 | |
| IOBC | 23 | 21 | 31 | 25,0 |
| NCIT | 13 | 15 | 19 | 15,7 |
| NIFSTD | 13 | 11 | 20 | 14,7 |
| SNOMEDCT | 13 | 7 | 15 | 11,7 |
| CHEAR | 10 | 11 | 12 | 11,0 |
| NBO | 8 | 9 | 11 | 9,3 |
| AGRO | 9 | 6 | 12 | 9,0 |
| CRISP | 8 | 10 | 8 | 8,7 |
| ENVO | 6 | 10 | 9 | 8,3 |
| ECSO | 6 | 9 | 8 | 7,7 |
| MESH | 6 | 8 | 9 | 7,7 |

## 5 Conclusion and future work

Many applications need extraction of named entities. To this end, we presented a framework that identifies and extracts entities from scientific publications from the ecological domain. The next step is to find not only named entities but also relations among entities. We have preliminary work on that.

## 6 Acknowledgements

We would like to thank A. Hildebrandt, FSU Jena and I. Schoening and T. Kloetzing, Max Planck Institute for Biogeochemistry, for evaluating extracted terms as domain experts. Also, we thank Andreas Ostrowski, FSU Jena, for providing the text corpus. This work has been mostly funded by the *Deutsche Forschungsgemeinschaft (DFG)* as part of CRC 1076 AquaDiva.

## References

[BKL09] Bird, S.; Klein, E.; Loper, E.: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, 2009.

[Ch06] Chang, Chia-Hui; Kayed, Mohammed; Girgis, Moheb R; Shaalan, Khaled F: A survey of web information extraction systems. IEEE TKDE, 18(10):1411–1428, 2006.

[CL11] Chang, Chih-Chung; Lin, Chih-Jen: LIBSVM: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST), 2(3):27, 2011.

[DHI12] Doan, AnHai; Halevy, Alon; Ives, Zachary: Principles of data integration. Elsevier, 2012.

[Go18] Golshan, Parisa Naderi; Dashti, HosseinAli Rahmani; Azizi, Shahrzad; Safari, Leila: A Study of Recent Contributions on Information Extraction. CoRR, abs/1803.05667, 2018.

[Ho02] Hobbs, Jerry R.: Information extraction from biomedical text. Journal of Biomedical Informatics, 35(4):260–264, 2002.

[Ki14] Kilgarriff, Adam; Baisa, Vít; Bušta, Jan; Jakubíček, Miloš; Kovář, Vojtěch; Michelfeit, Jan; Rychlý, Pavel; Suchomel, Vít: The Sketch Engine: ten years on, volume 1. Jul 2014.

[KN04] Krauthammer, Michael; Nenadic, Goran: Term identification in the biomedical literature. Journal of Biomedical Informatics, 37(6):512–526, 2004.

[NS07] Nadeau, David; Sekine, Satoshi: A survey of named entity recognition and classification. Investigationes, 30(1):W3–W26, 2007.

[Sh17] Shen, Yanyao; Yun, Hyokun; Lipton, Zachary Chase; Kronrod, Yakov; Anandkumar, Animashree: Deep Active Learning for Named Entity Recognition. In: Rep4NLP@ACL. pp. 252–256, 2017.

[YB18] Yadav, Vikas; Bethard, Steven: A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. In: COLING 2018. pp. 2145–2158, 2018.